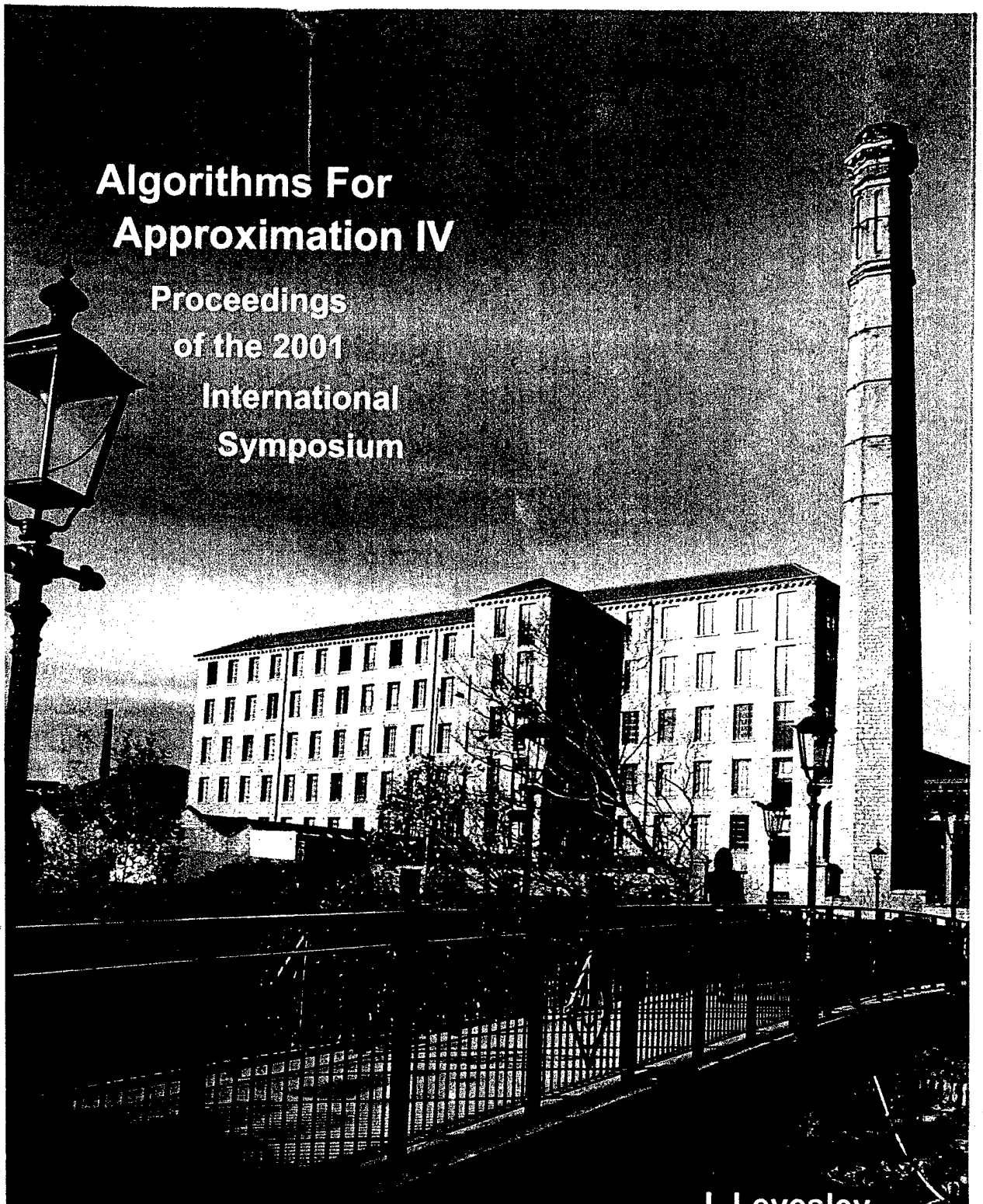
Algorithms For Approximation IV

Proceedings of the 2001 International Symposium



J. Levesley
I.J. Anderson
J.C. Mason
(Eds)

University of Huddersfield

Proceedings Published 2002

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
14-10-2002	Conference Proceedings	16 July 2001 - 20 July 2001

4. TITLE AND SUBTITLE

Algorithms for Approximation IV (A4A4)

5a. CONTRACT NUMBER
F61775-00-WF078

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Conference Committee
(Organizer, Professor John C Mason)

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Huddersfield
Queensgate
Huddersfield HD1 3DH
United Kingdom

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
PSC 802 BOX 14
FPO 09499-0014

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
CSP 00-5078

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Final Proceedings for Algorithms for Approximation IV (A4A4), 16 July 2001 - 20 July 2001, a multidisciplinary conference addressing many areas of interest to the Air Force. Of primary interest are the potential applications to Modeling and Simulation. Specifically, the topics to be covered include in the following four major areas: Algorithms, Efficiency, Software, and Applications. Each major topic is divided into subtopics as follows:
Algorithms - Approximation of Functions, Data Fitting, Geometric and Surface Modelling, Splines, Wavelets, Radial Basis Functions, Support Vector Machines, Norms and Metrics, Errors in Data, Uncertainty Estimation,
Efficiency - Numerical Analysis, Parallel Processing,
Software - Standards, Libraries, New Routines, World Wide Web,
Applications - Metrology (Science of Measurement), Data Fusion, Neural Networks and Intelligent Systems, Spherical Data and Geodetics, Medical Data.

15. SUBJECT TERMS

EOARD, Software, Mathematics, Intelligent Systems, Computational Mathematics

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18, NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Christopher Reuter, Ph. D.
UNCLAS	UNCLAS	UNCLAS	UL	492 (plus front matter)	19b. TELEPHONE NUMBER *(Include area code)* +44 (0)20 7514 4474

Algorithms for Approximation IV

The proceedings of the Fourth International Symposium on Algorithms for Approximation, held at the University of Huddersfield, July 2001.

Edited by

Jeremy Levesley
Department of Mathematics and Computer Science
University of Leicester
Leicester LE1 7RH, UK.

Iain Anderson
Analyticon Ltd
Elopak House
Rutherford Close
Meadway Technology Park
Stevenage, SG1 2EF, UK.

John C. Mason
School of Computing and Mathematics
The University of Huddersfield
Queensgate
Huddersfield, HD1 3DH, UK.

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

Published by The University of Huddersfield

20030319 033

AQ F03-04-0599

First published in 2002 by the University of Huddersfield, Queensgate, Huddersfield HD1 3DH.

Printed in Great Britain by The Charlesworth Group, 254, Deighton Road, Huddersfield HD2 1JJ, UK.

ISBN 186218 040 7

British Library in Publication Data
A catalogue record for this book is available from the British Library.



Contents

Contributors

Preface

Chapter 1 Computer Aided Geometric Design 1

An automatic control point choice in algebraic numerical grid generation 2
C. Conti, R. Morandi, and D. Scaramelli

Shape-measure method for introducing the nearly optimal domain 10
A. Fakharzadeh and J. E. Rubio

Convex combination maps 18
M. Floater

Shape preserving interpolation by curves 24
T. N. T. Goodman

CAGD techniques for differentiable manifolds 36
A. Lin and M. Walker

Parametric shape-preserving spatial interpolation and ν-splines 44
C. Manni

On the q-Bernstein polynomials 52
H. Oruç and N. Tuncer

Uniform Powell-Sabin splines for the polygonal hole problem 60
J. Windmolders and P. Dierckx

Chapter 2 Differential Equations 69

Iterative refinement schemes for an ill-conditioned transfer equation in astrophysics 70
M. Ahues, F. d'Almeida, A. Largillier, O. Titaud, and P. Vasconcelos

Geometric symmetry in the symmetric Galerkin BEM 78
A. Aimi and M. Diligenti

The numerical simulation of the qualitative behaviour of Volterra integro-differential equations 86
J. T. Edwards, N. J. Ford, and J. A. Roberts

Systems of delay equations with small solutions: a numerical approach 94
N. J. Ford and P. M. Lumb

On an adaptive mesh algorithm with minimal distance control 102
K. Shanazari and K. Chen

An alternative approach for solving Maxwell equations 110
W. Sproessig and E. Venturino

Chapter 3 Metrology 121

Orthogonal distance fitting of parametric curves and surfaces 122
S. J. Ahn, E. Westkämper, and W. Rauh

Template matching in the ℓ_1 norm 130
I. J. Anderson and C. Ross

A bootstrap method for mixture models and interval data in inter-comparisons 138
P. Ciarlini, G. Regoliosi, and F. Pavese

Efficient algorithms for structured self-calibration problems 146
A. Forbes

On measurement uncertainties derived from "metrological statistics" 154
M. Grabe

l_1 and l_∞ fitting of geometric elements 162
H.-P. Helfrich and D. S. Zwick

Evaluation of measurements by the method of least squares 170
L. Nielsen

An overview of the relationship between approximation theory and filtration 188
P. J. Scott, X. Q. Jiang, and L. A. Blunt

Chapter 4 Radial Basis Functions 197

Applications of radial basis functions: Sobolev-orthogonal functions, radial basis functions and spectral methods 198
M.D. Buhmann, A. Iserles, and S. P. Nørsett

Approximation with the radial basis functions of Lewitt 212
J. J. Green

Computing with radial basic functions the Beatson-Light way! 220
W. A. Light

Application of orthogonalisation procedures for Gaussian radial basis functions and Chebyshev polynomials 236
J. C. Mason and A. Crampton

Geometric knot selection for radial basis scattered data approximation 244
R. Morandi and A. Sestini

On the boundary over distance preconditioner for radial basis function interpolation 252
C. T. Mouat and R. K. Beatson

What are 'good' points for local interpolation by radial basis functions? 260
R. P. Tong, A. Crampton, and A. E. Trefethen

Chapter 5 Regression 269

Generalised Gauss-Markov regression 270
A. Forbes, P. M. Harris, and I. M. Smith

Nonparametric regression subject to a given number of local extreme values 278
A. Majidi and L. Davies

Model fitting using the least volume criterion 286
C. Tofallis

Some problems in orthogonal and non-orthogonal distance regression 294
G. A. Watson

Chapter 6 Splines and Wavelets 305

Nonlinear multiscale transformations: from synchronisation to error control 306
F. Arandiga and R. Donat

Splines: a new contribution to wavelet analysis 314
A. Z. Averbuch and V. A. Zheludev

Knot removal for tensor product splines 322
T. Brenna

Fixed- and free-knot univariate least-squares data approximation by polynomial splines 330
M. Cox, P. Harris, and P. Kenward

On the approximation power of local least squares polynomials 346
O. Davydov

A wavelet-based preconditioning method for dense matrices with block structure 354
J. M. Ford and K. Chen

Some properties of the perturbed Haar wavelets 362
A. L. Gonzalez and R. A. Zalik

An example concerning the L_p-stability of piecewise linear B-wavelets 370
P. Oja and E. Quak

How many holes can locally linearly independent refinable vector functions have? 378
G. Plonka

The correlation between the convergence of subdivision processes and solvability of refinement equations 394
V. Protasov

Accurate approximation of functions with discontinuities using low order Fourier coefficients 402
R. K. Wright

Chapter 7 General Approximation 411

Remarks on delay approximations based on feedback 412
A. Beghi, A. Lepschy, W. Krajewski, and U. Viaro

Point shifts in rational interpolation with optimized denominator 420
J.-P. Berrut and H. D. Mittelmann

An application of a mathematical blood flow model 428
M. Breuss, A. Meister, and B. Fischer

Zeros of the hypergeometric polynomial $F(-n, b; c; , z)$ 436
K. Driver and K. Jordaan

Approximation error maps 446
A. Gomide and J. Stolfi

Approximation by perceptron networks 454
V. Kůrková

Eye-ball rebuilding using splines with a view to refractive surgery simulation 462
M. Lamard, B. Cochener, and A. Le Méhauté

A robust algorithm for least absolute deviation curve fitting 470
D. Lei, I. J. Anderson, and M. G. Cox

Tomographic reconstruction using Cesaro-means and Newman-Shapiro operators 478
U. Maier

A unified approach to fast algorithms of discrete trigonometric transforms 486
M. Tasche and H. Zeuner

Contributors

Invited Speakers

Martin Buhmann	Lehrstuhl Numerische Mathematik, Mathematisches Institut, Justus-Liebig-University, 35392 Giessen, Germany.
Maurice Cox	National Physical Laboratory, Teddington, Middlesex, TW11 0LW, UK.
Kathy Driver	School of Mathematics, University of the Witwatersrand, Private Bag 3, WITS, 2050, South Africa.
Michael Floater	SINTEF, Applied Mathematics, P.O. Box 124, Blindern, 0314 Oslo, NORWAY.
Tim Goodman	Department of Mathematics, The University of Dundee, Dundee DD1 4HN, Scotland.
Will Light	Department of Mathematics, The University of Leicester, Leicester LE1 7RH, UK.
Lars Nielsen	Danish Institute of Fundamental Metrology DK-2800 Lyngby, Denmark.
Gerlind Plonka	Gerhard-Mercator-Universität Duisburg Institut fr Mathematik, D-47048 Duisburg, Germany.
Tomaso Poggio	Massachusetts Institute of Technology Department of Brain and Cognitive Sciences 77 Massachusetts Avenue, E25-406 Cambridge, MA 02139-4307, USA.
Larry Schumaker	Vanderbilt University, Department of Mathematics, 1326 Stevenson Center, Nashville TN 37240-0001, USA.
Alistair Watson	Department of Mathematics, The University of Dundee, Dundee DD1 4HN, Scotland.

Contributing Speakers

S. J. Ahn	Fraunhofer Institute for Manufacturing Engineering and Automation (IPA), 70569 Stuttgart, Germany.
A. Aimi	Department of Mathematics, University of Parma, Italy.
F. D. d'Almeida	University of Porto, Faculty of Engineering, 4200-468 Porto, Portugal.
I. J. Anderson	Analyticon Ltd, Elopak House, Meadway Technology Park, Stevenage, SG1 2EF, UK.
F. Arandiga	Dept. Matematica Aplicada, University of Valencia, Spain.
A. A. Badr	Alexandria University, Dept. of Mathematics, Faculty of Science, Alexandria, Egypt.
R. K. Beatson	Dept. of Mathematics and Statistics, Univ. of Canterbury, Christchurch, New Zealand.

E. Belinsky	University of the West Indies, Dept. of Computer Science, Maths and Physics, P O Box 64, Bridgetown, Barbados.
J.-P. Berrut	Dept. de Mathématiques, Université de Fribourg, Switzerland.
K. Bittner	University of Missouri - St Louis, Dept. of Maths and Computing Science, St Louis, MO63121, USA.
T. Brenna	Dept. of Informatics, University of Oslo, Oslo, Norway.
M. Breuss	Dept. of Mathematics, University of Hamburg, Hamburg, Germany.
C. Brezinski	University of Lille, Lille, France.
A. Chunovkina	VNIIM, St Petersburg, Russia.
P. Cross	University College London, Dept. of Geomatic Engineering, London WC1E 6BT, UK.
M. P. Dainton	National Physical Laboratory, Teddington, Middlesex, TW11 OLW, UK.
O. Davydov	Universität Giessen, Mathematisches Institut, D-35392 Giessen, Germany.
A. Fakharzadeh	Dept. of Mathematics, Shahid Chamran University of Ahvaz, Ahvaz, Iran.
A. B. Forbes	National Physical Laboratory, Middlesex TW11 0LW, UK.
J. M. Ford	Dept. of Mathematical Sciences, University of Liverpool, Liverpool L69 7ZL, UK.
N. Ford	Chester College, Parkgate Road, Chester, CH1 4BJ, UK.
D. J. Gavaghan	Oxford University, Computing Laboratory, Oxford, OX1 3QD, UK.
A. J. P. Gomes	University Beira Interior, Dept. Informatica, 6201-001 Covilha, Portugal.
A. Gomide	Institute of Computing, University of Campinas, Brazil.
M. Grabe	PTB, Am Hasselteich 5, 38104 Braunschweig, Germany.
P. R. Graves-Morris	University of Bradford, Dept. of Maths and Computing Science, Bradford, BD7 1DP, UK.
J. J. Green	University of Sheffield, Dept. of Applied Mathematics, Sheffield, UK.
H.-P. Helfrich	Mathematisches Seminar der Landwirtschaftlichen Fakultät der Universität Bonn, Bonn, Germany.
H. O. Kim	KAIST, Division of Applied Mathematics, Taejon, Korea.
W. Krajewski	Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland.
V. Kurkova	Academy of Sciences of the Czech Republic, Institute of Computer Science, PO Box 5, 182 07 Prague 8, Czech Republic.
M. Lamard	LATIM - INSERM ERM 0102, 29609 Brest, Cedex France.
D. Lei	School of Computing and Mathematics, University of Huddersfield, Huddersfield, UK.
S. Li	Southeastern Louisiana University, USA.
J. Lippus	Tallinn Tech. University, Institute of Cybernetics, 12618 Tallinn, Estonia.
P. M. Lumb	Chester College, Parkgate Road, Chester, CH1 4BJ, UK.
T. Lyche	University of Oslo Institute for Informatics, P O Box 1080, Blindern, 0316 Oslo, Norway.

U. Maier	Justus-Liebig Universität, Mathematisches Institut, D-35392 Giessen, Germany.
A. Majidi	Dept. of Mathematics and Computer Science, University of Essen, Germany.
C. Manni	Dept. of Mathematics, University of Torino, Italy.
J. C. Mason	School of Computing and Mathematics, University of Huddersfield, Huddersfield, UK.
G. W. Morgan	Numerical Algorithms Group, Oxford, UK.
A. Palomares	Universidad de Granada, Facultad de Ciencias, 18071 Granada, Spain.
F. Pavese	Istituto di Metrologia "G.Colonnetti", Torino, Italy.
M. J. D. Powell	University of Cambridge, DAMTP, Cambridge, CB3 9EW, UK.
A. Prymak	National Taras Shevchenko University of Kyiv, Mech-Math Faculty, Kyiv 01033, Ukraine.
E. Quak	SINTEF Applied Mathematics, P.O. Box 124 Blindern, 0314 Oslo, Norway.
M. Rogina	University of Zagreb, Dept. of Mathematics, 10002 Zagreb, Croatia.
C. Ross	School of Computing and Mathematics, University of Huddersfield, Huddersfield, UK.
D. Scaramelli	Dipartimento di Energetica, 50134 Firenze, Italy.
C. Schneider	Johannes Gutenberg Universität, FB 17, D-55099 Mainz, Germany.
P. J. Scott	Taylor Hobson Ltd, New Star Road, Leicester LE4 9JQ, UK.
S. Serra Capizzano	University Insubria Como, 22100 Como, Italy.
A. Sestini	Dipartimento di Energetica, Università di Firenze, Italy.
K. Shanazari	Dept. of Mathematical Sciences, The University of Liverpool, Liverpool L69 7ZL, UK.
I. M. Smith	National Physical Laboratory, Middlesex, TW11 OLW, UK.
A. Sommariva	Universita di Padova, Dipartimento di Matematica Pura e Applicada, Padova, Italy.
W. Sproessig	Freiberg University of Mining and Technology, 09596 Frieburg, Germany
K. Strøm	SimSurgery, Sognsveien 75B, N-0855 Oslo, Norway.
M. Tasche	University of Rostock, Dept. of Mathematics, D–18051 Rostock, Germany.
C. Tofallis	University of Hertfordshire Business School, Hertford, SG13 8QF, UK.
R. P. Tong	The Numerical Algorithms Group Ltd, Oxford, OX2 8DR, UK.
L. Traversoni	Universidad Autonoma Metropolitana, D.F. Mexico CP 09340.
N. Tuncer	Dept. of Mathematics, Dokuz Eylül University, 35160 Buca Izmir, Turkey
M. Walker	York University, Toronto M3J 1P3, Canada.
J. Windmolders	Dept. of Computer Sciences, Kath. University Leuven, Belgium
R. K. Wright	Dept. of Mathematics and Statistics, UVM, Burlington, VT, 05445 USA.
R. A. Zalik	Dept. of Mathematics, Auburn University, AL 36849–5310, USA.
V. A. Zheludev	School of Computer Science, Tel Aviv University, Israel.
D. S. Zwick	Wilcox Associates, Inc. Phoenix, AZ 85310 USA.

Chairs

CAGD	T. N. T. Goodman
Data Approximation	H.-P. Helfrich
Metrology	A B Forbes
Neural Networks	V. Kurkova
Orthogonal Polynomials and Pade Approximation	P. R. Graves-Morris
Radial Basis Functions	J. C. Mason
Radial Basis Functions and Wavelets	M J D Powell
Shape Preserving Methods	C. Manni
Spline Functions	C. Brezinski
Spline Functions	T. Morton

Preface

This book contains the proceedings of an International Symposium on Algorithms for Approximation Four (A4A4), held at University of Huddersfield from July 15th to 20th, 2001, and attended by 106 people from no less than 32 countries. The accommodation base was the attractive University Park at Storthes Hall, where social events were centred. There was a very friendly atmosphere, helped by the presence of a significant number of younger people to balance the stalwarts. Food was excellent and weather was generally good.

This was the fourth, after a pause of 9 years, in the series of "Algorithms for Approximation" meetings held before in Oxfordshire in 1985, 1988, 1992, and once again it was run under the sponsorship of US Air Force (European Office of Aerospace Research and Development) and this time with grants from London Mathematical Society and National Physical Laboratory (NPL) (Software Support for Metrology).

The Organising Committee consisted of Iain Anderson, John Mason, David Turner (Huddersfield) Maurice Cox and Alistair Forbes (NPL) and Jeremy Levesley and Will Light (Leicester). In addition to them, the Programme Committee included Claude Brezinski (Lille), Martin Buhmann (Giessen), Tim Goodman (Dundee), Tom Lyche (Oslo), Alistair Watson (Dundee) and Larry Schumaker (Vanderbilt). In support of the committee, the Symposium Secretary, Ros Hawkins was extremely efficient, and was helped by Karen Mitchell.

Moving to the academic programme, there were 11 invited speakers. From UK were Maurice Cox, Tim Goodman, Alistair Watson and Will Light; from other parts of Europe were Martin Buhmann (Giessen), Michael Floater (SINTEF Oslo), Lars Nielsen (Danish Institute of Fundamental Metrology) and Gerlind Plonka (Duisburg); from USA were Tomaso Poggio (MIT) and Larry Schumaker (Vanderbilt); and from South Africa, Kathy Driver (Witwatersrand).

In addition there were 74 submitted papers given at the meeting, of which a good proportion were offered in Special Sessions in Metrology-Maths (run by David Turner), Metrology-Stats (Alistair Forbes), Orthogonal Polynomials and Padé Approximation (Claude Brezinski and Peter Graves-Morris (Bradford)), Spline Functions (Tom Lyche), Mathematical Modelling in Medicine (Ewald Quak), Integrals and Integral Equations (Ezio Venturino (Torino)) and Wavelets (Richard Zalik (Auburn)).

The current volume contains a substantial portion of the papers from the conference, which were provided by the speakers, so that this is a solid

and broad contribution to the area. The book has been organised in topics to suit the final selection of papers.

All submitted papers were refereed and significant modifications were made to a number of papers. In general, there was a high standard of submissions.

We cannot conclude this preface without mentioning the celebration of three 60th birthdays of 2001 at the meeting, namely those of Claude Brezinski, Maurice Cox, and John Mason. All played major parts in the Symposium.

We must finish by offering thanks to all the staff at University of Huddersfield, NPL, USAF-EOARD, London Mathematical Society, and the publishers, who contributed to this most successful and memorable symposium.

Thanks also go to Jeremy Levesley and Iain Anderson and the publishers, who worked so hard on the proceedings, and to all authors without whom the volume would not exist.

John Mason
Huddersfield

Chapter 1

Computer Aided Geometric Design

An automatic control point choice in algebraic numerical grid generation

C. Conti, R. Morandi, and D. Scaramelli

Dipartimento di Energetica, via C. Lombroso 6/17, 50134 Firenze, Italy
costanza@sirio.de.unifi.it, morandi@de.unifi.it, scaramel@math.unipd.it

Abstract

A strategy to construct a grid conforming to the boundaries of a prescribed domain by using transfinite interpolation methods is discussed. A transfinite interpolation procedure is combined with a B-spline tensor product scheme defined by using suitable control points. Their choice is performed by taking into account a quality measure parameter based on the condition number of matrices linked to the covariant metric tensors.

1 Introduction

The algebraic grid generation approach relies on the construction of a coordinate transformation from the computational domain into the physical domain. In particular, this can be obtained through transfinite interpolating operators allowing us the generation of grids with boundary conformity. Furthermore, using a Hermite-type transfinite interpolating scheme we can obtain orthogonal grid lines emanating from the boundary. This can be very important for practical reasons since the grid point distribution in the immediate neighborhood of the boundaries has a strong influence on the accuracy of the numerical solution of partial differential equations [5]. Furthermore, in case a domain decomposition is necessary the orthogonality guarantees smoother grids. In order to obtain a grid with other specified properties, e.g. the control of the shape and position of the coordinate curves, transfinite interpolating methods can be combined with tensor product schemes using suitably chosen control points (see for instance [1, 2, 6, 7, 8]). Even though this type of algebraic method is computationally efficient, to define workable meshes, a significant amount of user interaction is required for the selection of the control points involved in the tensor product. To overcome this drawback, an automatic strategy for choosing the control points turns out to be desirable. Here, following the approach first discussed in [1], we present an algebraic Hermite-type transfinite method to construct a grid interpolating the boundary and its normal derivatives. In fact, given a "quadrilateral" domain $\Omega \subset \mathbb{R}^2$, a transformation $G: R = [0,1] \times [0,1] \to \Omega$ is defined as

$$G(s,t) := T_P(s,t) + (P_1 \oplus P_2)([\phi, \psi] - T_P)(s,t) \tag{1.1}$$

where T_P is a tensor product surface i.e. $T_P(s,t) := \sum_{i=1}^{m} \sum_{j=1}^{n} Q_{ij} B_{i,3}(s) B_{j,3}(t)$ with $B_{i,3}$ denoting the usual cubic B-spline, ϕ and ψ are boundary curves and $(P_1 \oplus P_2)$ is the

Boolean sum of Hermite-type blending function linear operators. The set $\mathcal{Q} = \{Q_{ij},\ i = 1,\dots,m,\ j = 1,\dots,n\}$ is the set of control points.

As already noted, the choice of the control points is a crucial matter. In this paper we take into account a grid quality measure parameter for their selection. In particular, the proposed automatic procedure relies on the fact that some grid properties can be described in terms of the condition number of matrices linked to the covariant metric tensors [4]. Therefore, the control points are chosen minimizing their condition number.

The outline of this paper is as follows. In Section 2, the transformation (1.1) is given in detail and its properties are investigated. In Section 3, a way for choosing the control points is proposed relying on a particular quality measure parameter. Finally, in Section 4 some numerical results are presented to illustrate the features of the proposed strategy.

2 The transformation

In this section the transformation (1.1) is characterized. Let us consider a "quadrilateral" domain $\Omega \subset \mathbb{R}^2$ such that $\partial\Omega = \cup_{i=1}^4 \partial\Omega_i$, with $\partial\Omega_1, \partial\Omega_2, \partial\Omega_3, \partial\Omega_4$ being the supports of four regular curves $\gamma_i : [0,1] \to \partial\Omega_i,\ i = 1,\dots,4$ taken counterclockwise. Furthermore, let us suppose that $\partial\Omega_1 \cap \partial\Omega_3 = \emptyset$ and $\partial\Omega_2 \cap \partial\Omega_4 = \emptyset$, with any other intersection occuring only at the end points of the boundary curves. In particular, the following compatibility conditions are assumed

$$\gamma_1(0) = \gamma_4(1),\ \gamma_1(1) = \gamma_2(0),\ \gamma_2(1) = \gamma_3(0),\ \gamma_4(0) = \gamma_3(1)\ .$$

For later convenience, we set $\phi_1(s) := \gamma_1(s),\ \phi_2(s) := \gamma_3(1-s)$ denoting by s the curve parameter running on $[0,1]$ and we set $\psi_1(t) := \gamma_4(1-t),\ \psi_2(t) := \gamma_2(t)$ denoting by t the curve parameter running on $[0,1]$. In addition, the components of the $\boldsymbol{\phi}$-curves and $\boldsymbol{\psi}$-curves are denoted by $\boldsymbol{\phi}^x, \boldsymbol{\phi}^y$ and $\boldsymbol{\psi}^x, \boldsymbol{\psi}^y$ respectively.

Next, we define four additional curves by computing the derivatives of the $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$-curves, i.e.,

$$\begin{aligned} \phi_{i+2}(s) &= \frac{\mathcal{C}}{||\phi_i'||_2}(-(\phi_i^y(s))',\ (\phi_i^x(s))')\ , \quad i = 1,2\ , \\ \psi_{j+2}(t) &= \frac{\mathcal{C}}{||\psi_j'||_2}(-(\psi_j^y(t))',\ (\psi_j^x(t))')\ , \quad j = 1,2\ , \end{aligned} \tag{2.1}$$

with $\mathcal{C}$ a constant value also depending on the curve orientations and with $||\cdot||_2$ the Euclidean norm. Then, we introduce the linear operators

$$\begin{aligned} &P_1[\boldsymbol{\phi}](s,t) := \textstyle\sum_{i=1}^4 \alpha_i(t)\phi_i(s)\ , \quad P_2[\boldsymbol{\psi}](s,t) := \textstyle\sum_{j=1}^4 \alpha_j(s)\psi_j(t)\ , \\ &P_1P_2[\boldsymbol{\phi},\boldsymbol{\psi}](s,t) := \textstyle\sum_{i=1}^2 \left(\alpha_i(t)P_2[\boldsymbol{\psi}](s,u_i) + \alpha_{i+2}(t)\frac{\partial P_2[\boldsymbol{\psi}](s,u_i)}{\partial t}\right)\ , \end{aligned} \tag{2.2}$$

where $u_1 = 0,\ u_2 = 1$. The functions $\alpha_i,\ i = 1,\dots,4$, are the dilated versions of the

classical Hermite bases with support on $[0,\bar{u}]$ and on $[1-\bar{u},1]$ being $0<\bar{u}<1$, i.e.

$$\alpha_1(s) := (1+2\tfrac{s}{\bar{u}})(1-\tfrac{s}{\bar{u}})^2 \ , \qquad \alpha_3(s) := s(1-\tfrac{s}{\bar{u}})^2 \ , \quad s\in[0,\bar{u}],$$

$$\alpha_2(s) := (3-2\tfrac{s+\bar{u}-1}{\bar{u}})(\tfrac{s+\bar{u}-1}{\bar{u}})^2 \ , \quad \alpha_4(s) := (s-1)(\tfrac{s+\bar{u}-1}{\bar{u}})^2 \ , \quad s\in[1-\bar{u},1]. \tag{2.3}$$

The Boolean sum operator $(P_1\oplus P_2) = P_1+P_2-P_1P_2$ provides the blending function surface

$$B(s,t) := (P_1\oplus P_2)[\phi,\psi](s,t) = P_1[\phi](s,t)+P_2[\psi](s,t)-P_1P_2[\phi,\psi](s,t)\ . \tag{2.4}$$

It is known that B satisfies

$$\begin{array}{ll} B(u_i,t)=\psi_i(t)\ ,\ i=1,2 & \frac{\partial B(u_i,t)}{\partial s}=\psi_i(t)\ ,\ i=3,4\ , \\ B(s,w_j)=\phi_j(s)\ ,\ j=1,2 & \frac{\partial B(s,w_j)}{\partial t}=\phi_j(s)\ ,\ j=3,4\ , \end{array} \tag{2.5}$$

where $u_1=u_3=0$, $u_2=u_4=1$ and $w_1=w_3=0$, $w_2=w_4=1$. It is worthwhile to remark that, as we are dealing with orthogonal grid lines emanating from the boundary of the domain, the intersecting boundary curves must be also orthogonal. Thus, the following additional conditions are assumed:

$$\begin{array}{llll} \phi_{i+2}(0) & =\psi_1'(w_i), & \phi_{i+2}(1) & =\psi_2'(w_i)\ , \\ \psi_{i+2}(0) & =\phi_1'(u_i), & \psi_{i+2}(1) & =\phi_2'(u_i)\ , \\ \phi_i''(0) & =\psi_1''(w_i), & \phi_i''(1) & =\psi_2''(w_i)\ , \end{array} \qquad i=1,2\ . \tag{2.6}$$

Now, in order to define a suitable grid, following the approach given in [1], we use the linear transformation G

$$G(s,t) := T_P(s,t)+(P_1\oplus P_2)\big([\phi,\psi]-T_P\big)(s,t) \tag{2.7}$$

where $T_P(s,t) := \sum_{i=1}^{m}\sum_{j=1}^{n} Q_{ij}B_{i,3}(s)B_{j,3}(t)$ with $B_{i,3}$ denoting the usual cubic B-splines with uniform knots. The set $\mathcal{Q}=\{Q_{ij},\ i=1,\dots,m,\ j=1,\dots,n\}$ is a suitable set of control points whose definition is discussed in Section 3. It should be noted that in (2.7) the Boolean sum operator is also acting on a surface $T_P(s,t)$. In this case (2.2) is used taking the eight boundary curves $T_P(0,t)$, $T_P(1,t)$, $T_P(s,0)$, $T_P(s,1)$, $\frac{\partial T_P(0,t)}{\partial s}$, $\frac{\partial T_P(1,t)}{\partial s}$, $\frac{\partial T_P(s,0)}{\partial t}$, $\frac{\partial T_P(s,1)}{\partial t}$.

It is easy to show that G still satisfies $G(u_i,t)=\psi_i(t)$, $i=1,2$, $\frac{\partial G(u_i,t)}{\partial s}=\psi_i(t)$, $i=3,4$, $G(s,w_j)=\phi_j(s)$, $j=1,2$ and $\frac{\partial G(s,w_j)}{\partial t}=\phi_j(s)$, $j=3,4$. Furthermore, because of the locality of the blending functions α_i, $i=1,\dots,4$, the control of the coordinate lines obtained by means of the evaluation of G over a parameter set in the interior of the domain is mainly based on the contribution of T_P. This fact and the use of B-splines ensures the convex-hull property in the interior of the domain. This property is of importance in numerical grid generation to locate the grid with respect to the position of control points.

3 Grid quality measure

It is well known that grid generation techniques sensible to grid quality features are particularly attractive. Thus, in this section, we discuss a strategy to choose the set $\mathcal{Q}$ of control points based on a suitable grid quality measure parameter.

Given a set of grid points $\mathcal{G} := \{G_{ij}\}_{i,j=1}^{M,N}$ defining the quadrilateral cells $\{C_{ij}\}_{i,j=1}^{M-1,N-1}$, quality measures can commonly include: grid "skewness", measuring the departure of C_{ij} from a rectangle, grid "aspect ratio", measuring the departure of C_{ij} from a rhombus or grid "conformality", measuring the departure of C_{ij} from a square (see for instance [5]).

Here, as done in [4] for the case of unstructured grids, we define a grid quality measure taking into account the condition number of particular matrices derived from the grid. As explained below, somehow this quality parameter measures the departure of C_{ij} from a square.

The strategy starts with a set $\mathcal{Q}^i$ of control points obtained by evaluating on a coarse parameter set $\mathcal{S}_c = \{(s_i, t_j)\}_{i,j=1}^{M_c,N_c}$ a Lagrange blending function surface (for detail related to Lagrange blending function methods we refer, for instance, to [3]) by working only with the four boundary curves of the given domain. Then, using $\mathcal{Q}^i$ a first grid is obtained by evaluating the surface G in (2.7) on a fine parameter set $\mathcal{S}_f = \{(s_i, t_j)\}_{i,j=1}^{M,N}$ obtaining the grid points

$$\mathcal{G} := \{G_{i,j} = (G_{i,j}^x, G_{i,j}^y) = G(s_i, t_j),\ i = 1, \ldots, M,\ j = 1, \ldots, N\}\ .$$

The set $\mathcal{G}$ is then used to define $(M-1) \times (N-1)$ bidimensional matrices associated with the $(M-1) \times (N-1)$ quadrilateral cells C_{ij}, $i = 1, \ldots, M-1$, $j = 1, \ldots, N-1$. These matrices are defined as

$$A_{i,j} := \begin{pmatrix} G_{i+1,j}^x - G_{i,j}^x & G_{i,j+1}^x - G_{i,j}^x \\ G_{i+1,j}^y - G_{i,j}^y & G_{i,j+1}^y - G_{i,j}^y \end{pmatrix},\ i = 1, \ldots, M-1,\ j = 1, \ldots, N-1 \quad (3.1)$$

and their condition number $K(A_{i,j})$ is related to the stretch of the cells. In fact, it is easy to prove that $K(A_{i,j}) := ||A_{i,j}||_2 \cdot ||A_{i,j}^{-1}||_2 = 1$ if and only if we are dealing with a cell C_{ij} where the three points $G_{i,j+1}, G_{i,j}, G_{i+1,j}$ generate half a square [9]. On the other hand, in order to involve all the grid points in the quality measure it is also convenient to define the boundary matrices

$$\begin{aligned} A_{i,N-1}^l &:= \begin{pmatrix} G_{i+1,N}^x - G_{i,N}^x & G_{i,N-1}^x - G_{i,N}^x \\ G_{i+1,N}^y - G_{i,N}^y & G_{i,N-1}^y - G_{i,N}^y \end{pmatrix},\ i = 1, \ldots, M-1, \\ A_{M-1,j}^r &:= \begin{pmatrix} G_{M,j+1}^x - G_{M,j}^x & G_{M-1,j}^x - G_{M,j}^x \\ G_{M,j+1}^y - G_{M,j}^y & G_{M-1,j}^y - G_{M,j}^y \end{pmatrix},\ j = 1, \ldots, N-1, \qquad (3.2) \\ A_{M-1,N-1}^u &:= \begin{pmatrix} G_{M,N}^x - G_{M-1,N}^x & G_{M,N}^x - G_{M,N-1}^x \\ G_{M,N}^y - G_{M-1,N}^y & G_{M,N}^y - G_{M,N-1}^y \end{pmatrix}, \end{aligned}$$

so that the boundary points are also taken into account.

Next, we modify the initial set $\mathcal{Q}^i$ of control points minimizing the following objective

function

$$f_{ob} = \frac{1}{MN}\left(\sum_{i=1}^{M-1}\sum_{j=1}^{N-1} K(A_{i,j}) + \sum_{i=1}^{M-1} K(A^l_{i,N-1}) + \sum_{j=1}^{N-1} K(A^r_{M-1,j}) + K(A^u_{M-1,N-1})\right). \tag{3.3}$$

The minimization is done with respect to the control points under suitable constraints on their coordinates depending on the geometry of the domain Ω. This is the only user interaction required.

Obviously, since ideal inner cells are characterized by an associated matrix $A_{i,j}$ having a condition number close to one, the optimal distribution of the control points should guarantee $\min_{\mathcal{Q}} f_{ob} \approx 1$. On the other hand, $\min_{\mathcal{Q}} f_{ob}$ strongly depends on the geometry of the domain (for example in case of a squared domain the optimal value is $\min_{\mathcal{Q}} f_{ob} = 1$ while, in general, this value is not reached).

Summary of the Method

(1) Compute the initial set of control points $\mathcal{Q}^i$ by means of a Lagrange blending function method using the four given boundary curves,
(2) Compute the initial grid $\mathcal{G}^i = \{G(s_i, t_j),\ i = 1, \ldots, M,\ j = 1, \ldots, N\}$ with G given in (2.7) by using the set of control points $\mathcal{Q}^i$,
(3) Minimize the objective function (3.3) so defining a new set of control points $\mathcal{Q}^f$,
(4) Compute the final grid $\mathcal{G}^f = \{G(\tilde{s}_i, \tilde{t}_j),\ i = 1, \ldots, \tilde{M},\ j = 1, \ldots, \tilde{N}\}$ with G given in (2.7) by using the set of control points $\mathcal{Q}^f$ with $\tilde{M} \gg M$, $\tilde{N} \gg N$.

Remark 3.1 *We note that, in order to reduce the computational cost of the minimization procedure, the integers M and N are chosen less than $\tilde{M}$ and $\tilde{N}$.*

4 Numerical Results

We conclude the paper giving some numerical results testing the properties of the transformation G and showing the performance of the proposed approach.

Three domains are considered. For each of them we present the initial grid obtained by the transformation G using the initial set of control points $\mathcal{Q}^i$ and the final grid obtained using the set of control points $\mathcal{Q}^f$ resulting from the minimization procedure. In all the figures the control points are denoted by the symbol '*'. The minimization problem is solved by using a sequential quadratic programming method i.e. by using the routine constr of the Optimization toolbox of the Matlab package. In the minimization procedure, the constraints on the control points $\mathcal{Q}^f$ are chosen so that some geometric properties of the domain, such as symmetry and convexity, are preserved. Furthermore, in all the examples M and N are equal to $\tilde{M}$ and to $\tilde{N}$. The values of the objective function before the minimization (f^i_{ob}) and after the minimization (f^f_{ob}) are also given in the figure captions.

The first and the second test display a "waterway" grid and a $\mathcal{H}$-shaped grid with their control points before and after the minimization procedure. The effectiveness of the method is evident.

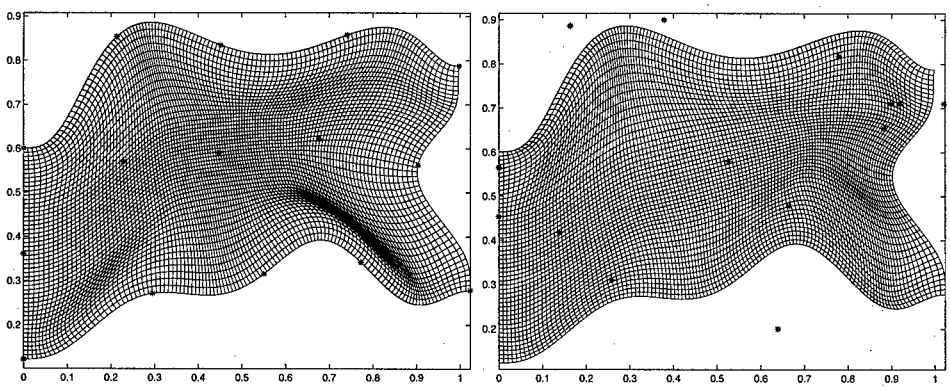


FIG. 1. Initial grid (left) and final grid (right), $f_{ob}^{i} = 3.74$, $f_{ob}^{f} = 1.65$.

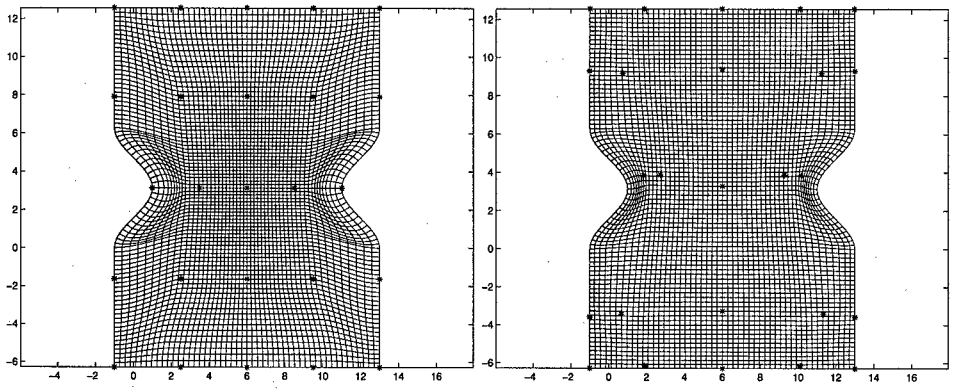


FIG. 2. Initial grid (left) and final grid (right), $f_{ob}^{i} = 1.45$, $f_{ob}^{f} = 1.22$.

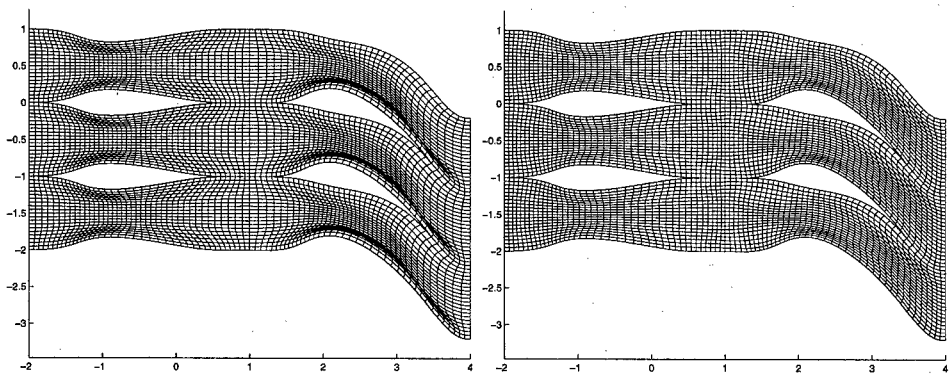


FIG. 3. Initial grid (left) and final grid (right), $f_{ob}^{i} = 1.89/6.36$, $f_{ob}^{f} = 1.59/2.20$.

Figure 3 shows a grid composed of six sub-grids, obtained via a domain decomposition approach. In this case, the Hermite-type interpolation method guarantees a C^1 connection among the patches. Here, the two values of f_{ob}^i and f_{ob}^f in the figure captions refer to the "horizontal" and "slanted" grids, respectively.

Bibliography

1. P. R. Eiseman, *High Level Continuity for Coordinate Generation with Precise Controls*, Journal of Computational Physics **47** (1982), 352–374.
2. P. R. Eiseman, *Control Point Grid Generation*, Computers Math. Applic. **5** (1992), 57–67.
3. W. J. Gordon and L. C. Thiel, *Transfinite Mappings and their Application to Grid Generation*, Appl. Math. and Comp. Vol. 10-11 on Numerical Grid Generation, J.F. Thompson ed., 171–192, 1982.
4. P. M. Knupp, *Matrix Norms & the Condition Number*, Proceedings, 8th International Meshing Roundtable, South Lake Tahoe, CA, U.S.A., 13-22, 1999.
5. V. D. Liseikin, *Grid Generation Methods*, Springer, 1999.
6. C. W. Mastin, *Three-dimensional Bezier interpolation in solid modeling and grid generation*, Comp. Aid. Geom. Des. **14** (1997), 797–805.
7. R. Morandi and A. Sestini, *Precise Controls in Numerical Grid Generation*, Advanced Topics in Multivariate Approximation, edited by F. Fontanella, K.Jetter, and P. J. Laurent, 243–258, 1996.
8. B. V. Saunders and P. W. Smith, *Grid generation and optimization using tensor product B-Splines*, Approx. Theory & its Appl., **3** (1987), 120–152.
9. D. Scaramelli, *Ph.D. Thesis*, in preparation.

Shape-measure method for introducing the nearly optimal domain

A. Fakharzadeh

Department of Mathematics, Shahid Chamran University of Ahvaz, Ahvaz, Iran.
a_fakharzadeh@hotmail.com

J. E. Rubio

Department of Applied Mathematical Studies, University of Leeds, Leeds, LS2 9JT, UK.

Abstract

We deal with introducing a new algorithm for solving the optimal shape problems in which they are defined with respect to a pair of geometrical elements. The problem is to find the optimal domain approximately for a given functional that is involved with the solution of a linear or nonlinear elliptic equation with a boundary condition over a domain. The Shape-Measure method, in Cartesian coordinates, will be used to find the nearly optimal solution in two steps. By transferring the problem into a measure-theoretical form, first we will find the solution of the elliptic problem for a given domain by using the embedding method. Then the Shape-Measure method will be applied to find the best domain approximately. An example will be given.

1 Introduction and Problem

Consider the optimal shape (optimal shape design) problems in which they are defined with respect to a pair of geometrical elements; this pair consists of a measurable set (in $I\!R^2$), which can be regarded as a domain, and a simple closed curve containing a given point, which is the boundary of the set. By considering the property for the desired curves to be simple, the problem depends on the geometry which is used. In polar coordinates, we solved the similar problem in [1]. But in Cartesian coordinates, it is difficult to introduce a linear condition which determines the property of a closed curve being simple. Thus here we consider some limitation on shape in order to make sure that it is simple. The problem will be solved in two stages. First, by use of measures, the value of the objective function will be calculated for any given domain. Then the optimal domain will be obtained by use of optimization techniques.

Let $D \subset I\!R^2$ be a bounded domain with a piecewise-smooth, closed and simple boundary ∂D. We assume that some part of ∂D is fixed and the rest, Γ, with the given initial and final points A and B respectively, is not fixed. Here we suppose that the fixed part of ∂D is made by three segments, parts of lines $y = 0, x = 0$ and $y = 1$ between points $A(1,0), (0,0), (0,1), B(1,1)$ (see Figure 1).

Thus Γ is chosen as an appropriate variable curve joining A and B so that D is well-defined. Let $u(X)$ $(X = (x,y) \in I\!R^2)$ be a bounded solution of the following elliptic

Preceding Page Blank

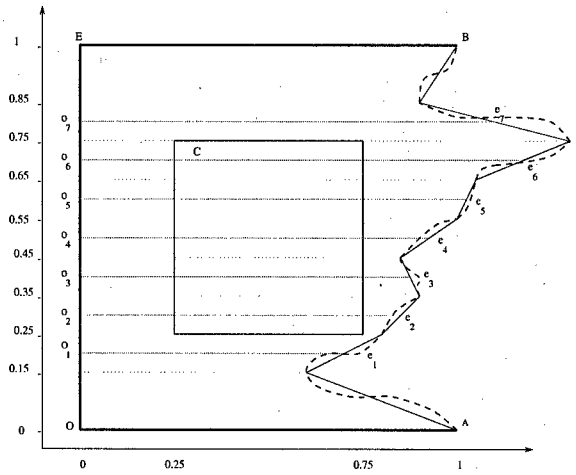


FIG. 1. An admissible domain D under the assumptions of the numerical work.

equation:

$$\Delta u(X) + f(X,u) = v(X)\ ,\ u_{|\partial D} = 0, \tag{1.1}$$

where $X \in D \to v(X) \in I\!R$ is a bounded real function (v also can be considered as a fixed control function); the function f is assumed to be a bounded and continuous real-valued function in $L_2(D \times I\!R)$. Moreover the above domain D is called an *admissible* if the equation (1.1) has a bounded solution on D; we denote by $\mathcal{D}$ as the set of all such admissible domains. We are going to solve the problem of minimizing the functional $\mathbf{I}(D) = \int_D f_\circ(X,u)\,dX$, on the set $\mathcal{D}$ where $f_\circ$ is a given continuous, nonnegative, real-valued function on $D \times I\!R$. To calculate the value of $\mathbf{I}(D)$ for a given domain D, it is necessary first to identify the solution of (1.1).

2 Weak solution and metamorphosis

In general, it is difficult to identify a classical solution for the problem like (1.1) and usually one tries to find a *weak* (generalized) solution of them. Hence the variational form of (1.1) is introduced in the following; we remind the reader that $H_0^1(D) = \{\psi \in H^1(D) : \psi_{|\partial D} = 0\}$, where $H^1(D)$ is the Sobolev space of order 1.

Proposition 2.1 *Let u be the classical solution of (1.1), then we have the following equality:*

$$\int_D (u\Delta\psi + \psi f)\,dX = \int_D \psi v\,dX\ ,\ \forall \psi \in H_0^1(D). \tag{2.1}$$

Proof: Multiplying (1.1) by the function $\psi \in H_0^1(D)$ and then integrating over D, with use of the Green's formula (see [3]) gives $\int_D (u\Delta\psi + \psi f - \psi v)\,dX = \int_{\partial D} (\psi \frac{\partial u}{\partial \mathbf{n}} - u\frac{\partial \psi}{\partial \mathbf{n}})\,dS$, where $\mathbf{n}$ is the unit vector normal to the boundary ∂D and directed outward with respect to D. Because $\psi_{|\partial D} = 0$ and $u_{|\partial D} = 0$, then (2.1) is satisfied. □

Definition 2.2 *A function $u \in H^1(D)$ is called a bounded weak solution of the problem (1.1) when it is bounded and satisfies the equality (2.1) for all $\psi \in H_0^1(D)$ (the conditions for existence of the weak solution of the problem (1.1) and also the boundedness property of it, have been considered in many references, like [3] and [2]).*

Now we apply our new way which is called the *Shape-Measure* method. Let $\Omega \equiv U \times \overline{D}$, where $U \subset I\!R$ is the smallest bounded set in which the bounded weak solution $u(\cdot)$ takes values. Then by applying the Riesz Representation Theorem ([6]), a bounded weak solution can be represented by a positive Radon measure; the proof of the following Proposition is similar to the Proposition 3.1 in [1].

Proposition 2.3 *Let $u(X)$ be a bounded generalized solution of (1.1); there exist a unique positive Radon measure, say μ_u, in $\mathcal{M}^+(\Omega)$ such that:*

$$\mu_u(F) \equiv \int_\Omega F \, d\mu_u = \int_D F(X,u)\, dX \; ; \; \forall F \in C(\Omega). \tag{2.2}$$

Thus the equality (2.1) can be changed to $\mu_u(F_\psi) = \gamma_\psi$, $\forall \psi \in H_0^1(D)$, where $F_\psi = u\Delta\psi + f\psi$ and $\gamma_\psi = \int_D \psi v \, dX$. Also, $\mathbf{I}(D) = \mu_u(f_\circ)$. Because the measure μ_u projects on the (x,y)-space as the respective Lebesgue measure, we should have $\mu_u(\xi) = a_\xi$, where $\xi : \Omega \to I\!R$ depends only on variable X (i.e. $\xi \in C_1(\Omega)$), and a_ξ is the Lebesgue integral of ξ over D. Therefore the original problem can be described as follows:

To find a measure $\mu_u \in \mathcal{M}^+(\Omega)$ so that it satisfies the following constraints:

$$\begin{aligned} \mu_u(F_\psi) &= \gamma_\psi, \qquad \forall \psi \in H_0^1(D); \\ \mu_u(\xi) &= a_\xi, \qquad \forall \xi \in C_1(\Omega). \end{aligned} \tag{2.3}$$

As Rubio did in [5], to be sure that we do not miss any solution, we extend the underlying space; instead of finding a measure $\mu_u \in \mathcal{M}^+(\Omega)$, introduced by Proposition 2.3 and equalities (2.3), we seek a measure $\mu \in \mathcal{M}^+(\Omega)$ which satisfies just the conditions:

$$\begin{aligned} \mu(F_\psi) &= \gamma_\psi, \qquad \forall \psi \in H_0^1(D); \\ \mu(\xi) &= a_\xi, \qquad \forall \xi \in C_1(\Omega). \end{aligned} \tag{2.4}$$

3 Approximation

The system (2.4) is linear because all the functions in the right-hand-side of equations are linear functions in their argument μ. But the number of equations and the underlying space are not finite. We shall develop this system by requiring that only a finite number of the constraints are satisfied. This will be achieved by choosing countable sets of functions whose linear combinations are dense in the appropriate spaces. But first we should approximate the unknown part of the boundary just by the finite number of its points. This idea comes from the approximation of a curve by broken lines. For the given D and hence for the given Γ, let $A_m = (x_m, y_m), m = 0, 1, 2, \ldots, M$, be a finite number of points on Γ (where $A_0 = A$). We link together each pair of consecutive points A_m and A_{m+1} for $m = 0, 1, \ldots, M-1$ and close this curve by joining the points A_M and B together. Now the resulted shape, which is denoted by ∂D_M, is an approximation for

∂D; we also call the domain which introduced by its boundary ∂D_M as D_M (see Figure 1).

It is possible that by increasing M, the curve ∂D_M will become closer and closer (in the Euclidean metric) to the curve ∂D, and hence one may conclude that the minimizer of $\mathbf{I}$ over $\mathcal{D}_M$, if it exists, tends to the minimizer of $\mathbf{I}$ over $\mathcal{D}$, if it exists. But some difficulties could arise (too oscillatory a curve may cause problems). Thus, we will fix the number of points. For a given M, let the value of the components $y_1, y_2, \ldots, y_M$, be fixed. Because x_m is a free term, the point A_m could be anywhere on the line $y = Y_m, x \geq 0$ for every m (see Figure 1). Therefore points A_m and A_{m+1} can be chosen so that they belong to Γ and hence the part of Γ between the lines $y = Y_m$ and $y = Y_{m+1}$ can be approximated by the segment $A_m A_{m+1}$. Hence, we do not lose generality. Thus, we fix the components $y_1, y_2, \ldots, y_M$ with the values $Y_1, Y_2, \ldots, Y_M$, respectively.

Now we introduce the set $\{\psi_i \in H_0^1(D) : i = 1, 2, \ldots\}$ so that the linear combinations of the functions $\{\psi_i\}$ are uniformly dense (that is, dense in the topology of the uniform convergence) in $H_0^1(D)$. We know that the vector space of polynomials with the variable x and y, $P(x, y)$, is dense in $C^\infty(\overline{D})$; therefore the set $P_0(x, y) = \{p(x, y) \in P(x, y) \mid p(x, y) = 0, \forall (x, y) \in \partial D\}$, is dense (uniformly) in $\{h \in C^\infty(\overline{D}) : h_{|\partial D} = 0\} \equiv C_0^\infty(\overline{D})\}$. Since the set

$$Q(x, y) = \{1, x, y, x^2, xy, y^2, x^3, x^2 y, xy^2, y^3, \ldots\}$$

is a countable base for the vector space $P(x, y)$, each elements of $P(x, y)$ and also $P_0(x, y)$, is a linear combination of the elements in $Q(x, y)$. By Theorem 3 of Mikhailov [3] page 131, the space $C^\infty(\overline{D})$ is dense in $H^1(D)$; thus the space $C_0^\infty(\overline{D})$ will be dense in $H_0^1(D)$. Consequently, the space $P_0(x, y)$ is uniformly dense in $H_0^1(D)$. We define

$$\psi_i(x, y) = xy(y - 1) \prod_{l=1}^{M} (x - x_l + y - Y_l) q_i(x, y), \tag{3.1}$$

where $q_i \in Q(x, y)$. Therefore $\psi_{|\Gamma} = 0$ and the set $\{\psi_i(x, y) : i = 1, 2, \ldots\}$, is total (dense in the topology of the uniform convergence) in $H_0^1(D)$.

For the second set of functions, let L be a given positive integer and divide D into L (not necessary equal) parts $D_1, D_2, \ldots, D_L$, so that by increasing L the area of $D_s, s = 1, 2, \ldots, L$, will be decreased. Then, for each s we define:

$$\xi_s(x, y, u) = \begin{cases} 1 & \text{if } (x, y) \in D_s, \\ 0 & \text{otherwise.} \end{cases}$$

These functions are not continuous, but each of them is the limit of an increasing sequence of positive continuous functions, $\{\xi_{s_k}\}$; then if μ is any positive Radon measure on Ω, $\mu(\xi_s) = \lim_{k \to \infty} \mu(\xi_{s_k})$. The linear combination of functions $\{\xi_j : j = 1, 2, \ldots, L\}$ for all positive integer L, can approximate a function in $C_1(\Omega)$ arbitrary well (see [5] Chapter 5).

By selecting just the finite number of functions in the mentioned spaces the problem (2.4) can be replaced by another one in which we are looking for the measure $\mu_{M_1, M_2} \in$

$\mathcal{M}^+(\Omega)$, so that it satisfies the following constraints:

$$\begin{aligned} \mu_{M_1,M_2}(F_i) &= \gamma_i, \qquad i = 1,2,\ldots,M_1; \\ \mu_{M_1,M_2}(\xi_j) &= a_j, \qquad j = 1,2,\ldots,M_2, \end{aligned} \tag{3.2}$$

where M_1 and M_2 are two positive integers and $F_i \equiv F_{\psi_i}$, $\gamma_i \equiv \gamma_{\psi_i}$, $a_j \equiv a_{\xi_j}$. If we denote by $Q(M_1, M_2)$ the set of positive Radon measures in $\mathcal{M}^+(\Omega)$ which satisfy equalities (3.2), and also denote by Q the set of positive Radon measures in $\mathcal{M}^+(\Omega)$ which satisfy equalities (2.4), one can easily prove the following Proposition by considering the proof of Proposition $III.1$ in [5].

Proposition 3.1 : *If $M_1, M_2 \longrightarrow \infty$ then $Q(M_1, M_2) \longrightarrow Q$; hence for the large enough numbers M_1 and M_2 the set Q can be identified by $Q(M_1, M_2)$.*

But even if the number of equations in (3.2) is finite, the underlying space $Q(M_1, M_2)$ is still infinite-dimensional. By Theorem $A.5$ in the Appendix of [5], μ_{M_1,M_2} in (3.2) can be characterized as $\mu_{M_1,M_2} = \sum_{n=1}^{M_1+M_2} \alpha_n \delta(Z_n)$, with triples $Z_n \in \Omega$ and the coefficients $\alpha_n \geq 0$ for $n = 1,2,\ldots,M_1 + M_2$, where $\delta(z) \in \mathcal{M}^+(\Omega)$ is supposed to be a unitary atomic measure with support the singleton set $\{z\}$. Thus the measure problem is equivalent to a nonlinear one in which the unknowns are the coefficients α_n and supports $\{Z_n\}$. Proposition $III.3$ of [5] Chapter 3, states that the measure μ_{M_1,M_2} has the following form

$$\mu_{M_1,M_2} = \sum_{n=1}^{N} \alpha_n \delta(Z_n), \tag{3.3}$$

where $Z_n, n = 1,2,\ldots,N$, belongs to a dense subset of Ω. Now let us put a discretization on Ω, with the nodes $Z_n = (x_n, y_n, u_n)$, in a dense subset of Ω; then we can set up the following linear system in which the unknowns are the coefficients α_n:

$$\begin{aligned} \alpha_n &\geq 0, \qquad n = 1,2,\ldots,N; \\ \sum_{n=1}^{N} \alpha_n F_i(Z_n) &= \gamma_i, \qquad i = 1,2,\ldots,M_1; \\ \sum_{n=1}^{N} \alpha_n \xi_j(Z_n) &= a_j, \qquad j = 1,2,\ldots,M_2. \end{aligned} \tag{3.4}$$

The solution of (3.4) is not necessary unique (even if the problem (1.1) satisfies the necessary conditions for having a unique bounded weak solution), because of the approximation scheme.

4 The optimal solution

The main aim of the present section is to find an optimal domain $D^* \in \mathcal{D}_M$ so that the value of $\mathbf{I}(D^*)$ will be the minimum on the set $\mathcal{D}_M$. By applying the result of the previous section, a solution of (1.1) can be found. Indeed, it is approximated by a solution of the linear system (3.4) according to the variables, $x_m, m = 1,2,\ldots,M$. As mentioned,

this solution is not necessary unique. Let us specify one by solving the following linear programming problem

$$\begin{aligned}
&\textbf{Minimize}: && \sum_{n=1}^{N} \alpha_n f_\circ(Z_n) && \\
&\textbf{Subject to}: && \alpha_n \geq 0, && n = 1, 2, \ldots, N; \\
& && \sum_{n=1}^{N} \alpha_n F_i(Z_n) = \gamma_i, && i = 1, 2, \ldots, M_1; \\
& && \sum_{n=1}^{N} \alpha_n \xi_j(Z_n) = a_j, && j = 1, 2, \ldots, M_2. \qquad (4.1)
\end{aligned}$$

Thus, for each D, the value $\mathbf{I}(D) = \int_D f_\circ(X,u)\, dX \equiv \mu(f_\circ) \simeq \mu_{M_1,M_2}(f_\circ)$, is defined uniquely in terms of the variables $x_m, m = 1, 2, \ldots, M$. So, we set up a function, $\mathbf{J}$, on $\mathcal{D}_M$ defined by $D \in \mathcal{D}_M \longrightarrow \mathbf{I}(D) \cong \mu_{M_1,M_2}(f_\circ)$ where $\mu_{M_1,M_2}(f_\circ) = \sum_{n=1}^{N} \alpha_n f_\circ(Z_n)$. Clearly $\mathbf{J}$ can be regarded as a vector function:

$$\mathbf{J} : (x_1, x_2, \ldots, x_M) \in I\!R^M \longrightarrow \mu_{M_1,M_2}(f_\circ) \in I\!R. \qquad (4.2)$$

Since $\mathbf{J}$ is a real-valued function which is bounded below, and is defined on a compact set (since constraints are to be put in the variables), it is possible to find a sequence of points so that the value of the function along the sequence tends to the (finite) infimum of the function. The coordinate values corresponding to the points in the sequence are of course finite. Now, suppose that $(x_1^*, x_2^*, \ldots, x_M^*)$ is the minimizer of the vector function $\mathbf{J}$; it can be identified by using one of the related minimization methods. The introduced domain by the minimizer is denoted by D^*. We assume in the following theoretical result that the minimization algorithm which is used, is perfect; that is, it comes out with the *global minimum* of J in its (compact) domain.

Theorem 4.1 : *Let M, M_1 and M_2 be the given positive integers which were defined in section 3, and D^* be the minimizer of (4.2) as mentioned above. Then D^* is the minimizer domain of the functional $\mathbf{I}$ over $\mathcal{D}_M$ and the value of $\mathbf{I}(D^*)$ can be approximated by $\mathbf{J}(D^*)$; moreover $\mathbf{J}(D^*) \longrightarrow \mathbf{I}(D^*)$ as M_1 and M_2 tend to infinity.*

Proof: Suppose D^* is not the minimizer of $\mathbf{I}$; hence there exists a domain, call D', in $\mathcal{D}_M$ so that $\mathbf{I}(D') < \mathbf{I}(D^*)$. Proposition 2.3 shows that there is a unique measure, call μ', in $\mathcal{M}^+(\Omega)$ so that $\mathbf{I}(D') = \mu'(f_\circ)$, and also Proposition 3.1 states that for sufficiently large numbers M_1 and M_2, $\mu'(f_\circ)$ can be approximated by $\mu'_{M_1,M_2}(f_\circ)$ in $Q(M_1, M_2)$. Thus, $\mathbf{I}(D') \cong \mu'_{M_1,M_2}(f_\circ) = \mathbf{J}(D')$. In the same way, one can show that $\mathbf{J}(D^*)$ approximates $\mathbf{I}(D^*)$; so $\mathbf{I}(D^*) \cong \mu^*_{M_1,M_2}(f_\circ) = \mathbf{J}(D^*)$. Hence $\mathbf{J}(D') < \mathbf{J}(D^*)$. which is contrary with the fact that D^* is the minimizer of $\mathbf{J}$. Moreover, from Proposition 3.1 it follows that $\mathbf{J}(D^*)$ tends to $\mathbf{I}(D^*)$ as $M_1, M_2 \longrightarrow \infty$. □

5 Numerical example

We consider the elliptic equations (1.1) with

$$v(x,y) = \begin{cases} 1 & \text{if } (x,y) \in D \cap C, \\ 0 & \text{otherwise,} \end{cases}$$

where C is the square $[\frac{1}{4},\frac{3}{4}]\times[\frac{1}{4},\frac{3}{4}]$ (see Figure 1). We also take $M = 8$, $M_1 = 10$, $M_2 = 8$, $N = 740$ (the 36 number of nodes are chosen so that $u_{|\partial D} = 0$) and suppose $Y_1, Y_2, \ldots, Y_8$ are $0.15, 0.25, \ldots, 0.85$, respectively. By extra constraints, $x_m \geq \frac{3}{4}$, $m = 2, 3, \ldots, 7$, the value of γ_i for any $D \in \mathcal{D}_M$ is defined as $\gamma_i = \int_{\frac{1}{4}}^{\frac{3}{4}} \int_{\frac{1}{4}}^{\frac{3}{4}} \psi_i(x,y)\, dxdy$, $i = 1, 2, \ldots, 10$. We also assume that the function u takes values in U =[-1,1], and consider the polynomials $q_i(x,y)$ as $1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3$. The function $f_\circ$ is chosen as $f_\circ = (u-0.1)^2$. This function can be considered as a distribution of heat in the surface for the system governed by an elliptic equations.

In minimization, we apply the Downhill Simplex Method in Multidimension by using the Subroutine *AMOEBA* (see [4]) and also consider an upper bound for variables (suppose they are not higher than 2). These conditions are applied by means of the penalty method (see [7]). Hence, for the nonlinear case of the partial differential equations (1.1), we have taken $f(x,y,u) = 0.25u^2$, and used the initial values as $X_m = 1.0, m = 1, 2, \ldots, 8$, and the stopping tolerance for the program (variable *ftol* in the Subroutine *AMOEBA*) as 10^{-7}. We remind the reader that the functions F_i and the values of γ_i, $i = 1, 2, \ldots, 10$, have been calculated by the package "*Maple V.3*". The results are: the optimal value of $\mathbf{I}$ = 0.45467920356379, the number of iterations = 502, the value of the variables in the final step are X_1 = 1.05019, X_2 = 1.08521, X_3 = 0.750001, X_4 = 0.768701, X_5 = 1.12986, X_6 = 1.13775 ,X_7 = 0.97783, X_8 = 1.61566, which represent the optimal domain, shown in the Figure 2.

Bibliography

1. A. Fakharzadeh J. and Rubio, J. E. *Shapes and Measures.* IMA Journal of Mathematical Control and Information, vol.16, p.207-220, 1999.
2. Ladyzhenskaya, O. A. and Ural'tseva, N. N. *Linear and Quasilinear Elliptic Equations.* vol.46, ACADEMIC PRESS, Mathematics in Science and Engineering, 1968.
3. Mikhailov, V. P. *Partial Differential Equation.* MIR Publisher, Moscow, 1978.
4. Press W. H., Flannery B. P., Teukolsky S. A. and Vetterling. W. T. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, 1986.
5. Rubio, J. E. *Control and Optimization: The Linear Treatment of Nonlinear Problems.* Manchester University Press, Manchester, 1986.
6. Rudin, W. *Real and Complex Analysis.* Tata McGraw-Hill Publishing Co.Ltd, New Delhi, second edition, 1983.
7. Walsh, G. R. *Method of Optimization.* John Wiley and Sons Ltd., 1975.

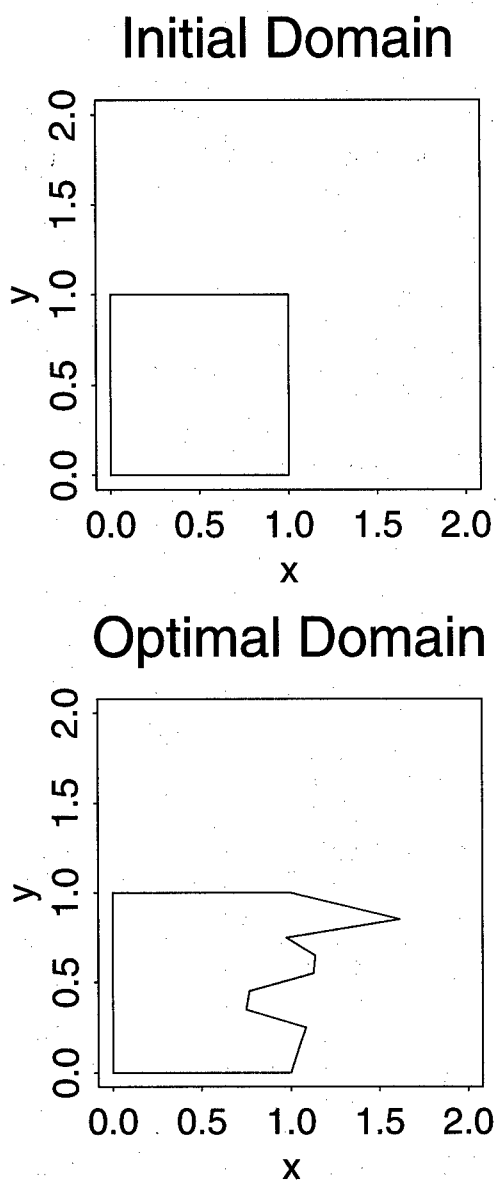


FIG. 2. The initial and the optimal domain for nonlinear case of elliptic equations.

Convex combination maps

Michael S. Floater

SINTEF, Postbox 124 Blindern, 0314 Oslo, NORWAY.
mif@math.sintef.no

Abstract

Piecewise linear maps over triangulations are used extensively in geometric modelling and computer graphics. This short note surveys recent progress on the important question of when such maps are one-to-one, central to which are *convex combination maps.*

1 Introduction

Piecewise linear maps over triangulations have several applications in geometric modelling and computer graphics. For example, Figure 1a shows a surface triangulation $\mathcal{T}$ of a set of points (x_i, y_i, z_i) sampled from some unknown surface in $\mathbb{R}^3$. A standard approach to fitting a smooth parametric surface $s(u, v)$ to these points is to first *parameterize* them, i.e., compute planar points (u_i, v_i) corresponding to the data points (x_i, y_i, z_i). Then using some scattered data method, we find a parametric surface $s : \Omega \to \mathbb{R}^3$, defined over some suitable domain Ω containing the points (u_i, v_i), such that

$$s(u_i, v_i) \approx (x_i, y_i, z_i).$$

A choice of parameterization is shown in Figure 1b and a least squares surface approximation using bicubic B-splines is shown in Figure 1c.

Notice that the choice of parameter points (u_i, v_i) uniquely determines a piecewise linear map $\phi : D_{\mathcal{T}} \to \mathbb{R}^2$, where $D_{\mathcal{T}}$ is the union of the triangles in $\mathcal{T}$. In practice, a necessary requirement on ϕ to ensure adequate quality of the subsequent surface approximation $s(u, v)$ is that ϕ should be injective. In Figure 1b the mapping ϕ was taken to be a so-called *convex combination map*, which, as we will see later, is guaranteed to be one-to-one since the boundary of $\mathcal{T}$ is mapped to a rectangle. Put another way, none of the triangles in Figure 1b are 'folded over'. In fact further properties of the map are important, such as linear precision, and this was achieved in Figure 1b by using the so-called *shape-preserving* weights (the coefficients in the convex combinations). For a discussion of that, see [3].

Another application of piecewise linear maps is to image morphing. Image morphing can be carried out by continuously transforming one planar triangulation $\mathcal{T}^0$ (whose vertices represent feature points in the image) to another, $\mathcal{T}^1$. Here we assume that there is a one-to-one correspondence between the vertices, edges, and triangles of $\mathcal{T}^0$ and $\mathcal{T}^1$. We can view each intermediate triangulation $\mathcal{T}(t)$, $0 \le t \le 1$, (where $\mathcal{T}(0) = \mathcal{T}^0$ and $\mathcal{T}(1) = \mathcal{T}^1$) as the image of a piecewise linear map $\phi(t) : D_{\mathcal{T}^0} \to D_{\mathcal{T}(t)}$. As with

parameterizations, it is again important that $\phi(t)$ is one-to-one. Figure 2 shows a so-called *convex combination morph* of [4] of two given planar triangulations: $\mathcal{T}^0$ appears on the left and $\mathcal{T}^1$ on the right. The two triangulations in the middle are $\mathcal{T}(1/3)$ and $\mathcal{T}(2/3)$. This morph ensures that $\phi(t)$ is one-to-one for all t in $[0,1]$ and therefore $\mathcal{T}(t)$ has no 'folded' triangles at any time instant t.

Piecewise linear maps also arise in: texture mapping; numerical grid generation; and in setting up multiresolution frameworks (nested spaces of piecewise linear functions) for manifold surface triangulations in computer graphics.

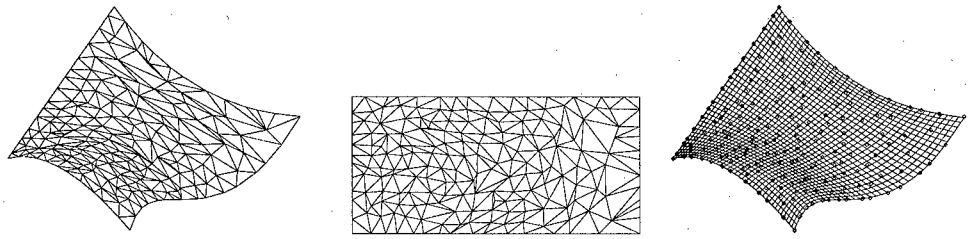


FIG. 1. Spatial triangulation (1a), Convex combination parameterization (1b), Bicubic spline approximation (1c).

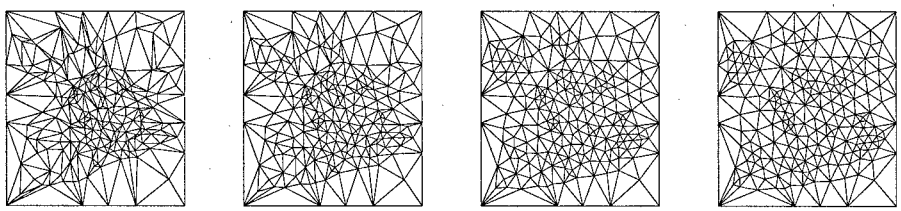


FIG. 2. Convex combination morph.

2 Convex combination maps

For the sake of simplicity we will only discuss convex combination maps defined over planar triangulations even though all the results hold equally well when the domain of the map is a spatial triangulation such as that in Figure 1a. Thus let $\mathcal{T} = \{T_1, \ldots, T_M\}$ be a simply-connected planar triangulation, with closed triangles T_i, and let $D_{\mathcal{T}} = \bigcup_{T \in \mathcal{T}} T$, as in Figure 3. We will call a mapping $\phi : D_{\mathcal{T}} \to \mathbb{R}^2$ a **convex combination map** if it is piecewise linear over $\mathcal{T}$ and, for every interior vertex v of $\mathcal{T}$, there exist weights $\lambda_{vw} > 0$, for $w \in N_v$, such that

$$\sum_{w \in N_v} \lambda_{vw} = 1,$$

and

$$\phi(v) = \sum_{w \in N_v} \lambda_{vw} \phi(w), \tag{1}$$

where N_v is the set of neighbours of v; see Figure 3.

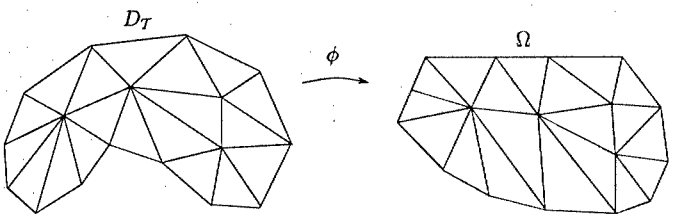


FIG. 3. Convex combination map.

In applications, the mapped boundary vertices $\phi(v)$ are chosen first. Then the weights λ_{vw} are all specified according to some chosen strategy. Then finally the mapped interior vertices are found by treating the equations in (1) as a linear system.

Example 2.1 If an interior vertex v of $\mathcal{T}$ has five neighbours $v_1, \ldots, v_5$, then we might set

$$\phi(v) = \frac{1}{4}\phi(v_1) + \frac{1}{8}\phi(v_2) + \frac{1}{8}\phi(v_3) + \frac{1}{4}\phi(v_4) + \frac{1}{4}\phi(v_5).$$

Until recently, the only theory behind convex combination maps was that of Tutte [8]. Working from a purely graph-theoretic point of view, Tutte proposed a so-called barycentric mapping for constructing straight line drawings of 3-connected graphs (which include triangulations). A barycentric mapping in our context is simply a convex combination map in which all the weights at each vertex are equal, i.e., $\lambda_{vw} = 1/d_v$, where d_v is the degree or valency of the vertex v. Thus for v in Example 1 we must have

$$\phi(v) = \frac{1}{5}\phi(v_1) + \frac{1}{5}\phi(v_2) + \frac{1}{5}\phi(v_3) + \frac{1}{5}\phi(v_4) + \frac{1}{5}\phi(v_5).$$

Tutte showed that a valid straight line drawing, i.e. one with no edge crossings, results from a barycentric mapping if the 'boundary' of the graph, a so-called 'cycle', is mapped to a convex polygon. However, as argued in [3], convex combination maps share all those properties of barycentric maps necessary for Tutte's proof. Thus when interpreted in the right way and suitably generalized, Tutte's theorem can be expressed in the following way.

Theorem 2.2 *Suppose $\phi : D_{\mathcal{T}} \to \mathbb{R}^2$ is a convex combination mapping which maps the n boundary vertices of $\mathcal{T}$ cyclically into the n vertices of some n-sided convex polygon in the plane. Then ϕ is one-to-one.*

Despite this generalization, however, there are still two aspects of it which need to be improved from the point of view of applications and future research.

The first is that we would like to extend the theorem so that we can allow some, and indeed many, of the mapped boundary vertices to be collinear. Indeed in the application

to parameterization for surface fitting, it might be convenient to map all the boundary vertices of the given triangulation into the four sides of a rectangle, as in Figure 1b. This is because tensor-product splines surfaces are defined over rectangular domains. Collinearity will also often be desirable in morphing, as in Figure 2, and in most other applications. Thus a drawback of Theorem 2.2 is that it does not allow collinear vertices in the image boundary.

The second aspect is that we would like to simplify the proof in order to have some hope of establishing the injectivity of piecewise linear maps in even more general situations, such as when mapping to non-convex regions, or when some of the mapped vertices are constrained, for example. The fact that Tutte's proof relies on the non-existence of the Kuratowski subgraphs K_5 and $K_{3,3}$ in a planar graph illustrates its complexity.

It is these two improvements that are the focus of [5]. The main idea of [5] is the observation that Theorem 2.2 is very similar to a theorem on harmonic maps, referred to by Duren and Hengartner [2] as the Radó-Kneser-Choquet thereom, which was established in [7, 6, 1]. Recall that a mapping $\phi : D \to \mathbb{R}^2$, with $D \subset \mathbb{R}^2$ and $\phi = (u, v)$, is **harmonic** if both its components $u(x, y)$ and $v(x, y)$ satisfy the Laplace equation in D, i.e.,

$$u_{xx} + u_{yy} = 0, \qquad v_{xx} + v_{yy} = 0;$$

see Figure 4.

Radó-Kneser-Choquet Theorem. *Suppose $\phi : D \to \mathbb{R}^2$ is a harmonic mapping which maps the boundary ∂D homeomorphically into the boundary $\partial\Omega$ of some convex region $\Omega \subset \mathbb{R}^2$. Then ϕ is one-to-one.*

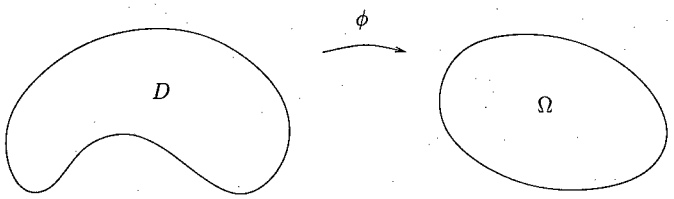


FIG. 4. Harmonic map.

This suggested that a proof of Theorem 2.2 might be based on a proof of the Radó-Kneser-Choquet theorem, in particular the short proof of Kneser [6]. Kneser's proof begins by showing that ϕ is locally one-to-one in the sense that the Jacobian of ϕ,

$$\begin{vmatrix} u_x & u_y \\ v_x & v_y \end{vmatrix}$$

never vanishes. Kneser establishes this by supposing that the Jacobian is zero at some point (x_0, y_0). In that case there must be a straight line $ax + by + c = 0$ passing through the point $\phi(x_0, y_0)$ such that both partial derivatives of the function $h(x, y) = au(x, y) + bv(x, y) + c$ are zero at (x_0, y_0). At the same time, the function $h : D \to \mathbb{R}$ is zero at (x_0, y_0) and has just two zeros along the boundary of D. Noting that $h(x, y)$ is a harmonic

function, Kneser then uses the Nodal Lines theorem of Courant to argue that there are at least four zero contours of h emanating from (x_0, y_0) and due to the maximum principle for h, these four curves can never self-intersect nor intersect one another. Therefore all four curves must reach the boundary of D which is a contradiction.

These ideas were used in [5] to establish a much simpler proof of Theorem 2.2 than that of Tutte. No graph theory is needed at all. Instead, the discrete maximum principle for convex combination functions plays the role of the maximum principle for harmonic functions. Similar to Kneser's proof we show first that ϕ is locally one-to-one, except that we understand this to mean that the restriction of ϕ to any quadrilateral in $\mathcal{T}$ is one-to-one, a quadrilateral being the union of two triangles sharing a common edge.

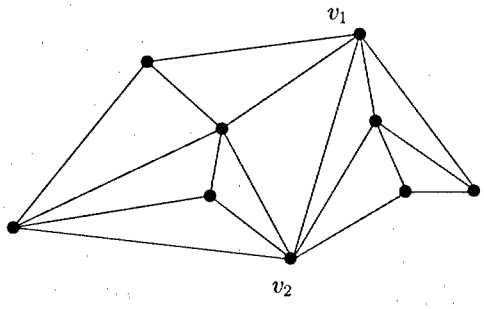


FIG. 5. Dividing edges.

Moreover, Theorem 2.2 is generalized in [5] to allow collinear mapped boundary vertices. We call an edge $[v, w]$ of $\mathcal{T}$ a *dividing edge* if both endpoints v and w are boundary vertices yet the edge $[v, w]$ itself is not contained in the boundary. For example in Figure 5, the only dividing edge in the triangulation is $[v_1, v_2]$. Dividing edges play a critical role because they partition the triangulation into subtriangulations $\mathcal{T}_i$, in each of which every convex combination function satisfies a discrete maximum principle in its strong form. The main result of [5] was the following.

Theorem 2.3 *Suppose $\mathcal{T}$ is any triangulation and that $\phi : D_{\mathcal{T}} \to \mathbb{R}^2$ is a convex combination mapping which maps $\partial D_{\mathcal{T}}$ homeomorphically into the boundary $\partial\Omega$ of some convex region $\Omega \subset \mathbb{R}^2$. Then ϕ is one-to-one if and only if no dividing edge $[v, w]$ of $\mathcal{T}$ is mapped by ϕ into $\partial\Omega$.*

3 Future research

Here is a list of topics for future research.

- A triangulation is a special (maximal) kind of planar graph. Can one extend Theorem 2.3 to other planar graphs, for example, rectangular grids? This is likely because Tutte's theory already holds for all 3-connected graphs.
- In what way can the theorem be extended from bivariate maps to trivariate ones?
- Can similar one-to-one maps be guaranteed when mapping closed surfaces of various topology? For example, we would like to map a closed manifold triangulation,

homeomorphic to a sphere, into a unit sphere injectively. Here each triangle in the triangulation would be mapped to a spherical triangle on the surface of the sphere.

- Can one find sufficient conditions for the injectivity of constrained maps, i.e., piecewise linear maps in which the image of certain interior points is specified in advance?
- Can one remove the requirement of having to map the boundary to a convex polygon and still ensure a one-to-one mapping under some weaker condition?
- Can the Radó-Kneser-Choquet thereom and Theorem 2.3 be combined as part of a single more general theorem?

Bibliography

1. G. Choquet, Sur un type de transformation analytique généralisant la représentation conforme et définé au moyen de fonctions harmoniques, Bull. Sci. Math. **69** (1945), 156–165.
2. P. Duren and W. Hengartner, Harmonic mappings of multiply connected domains, Pac. J. Math. **180** (1997), 201–220.
3. M. S. Floater, Parametrization and smooth approximation of surface triangulations, Comp. Aided Geom. Design **14** (1997), 231–250.
4. M. S. Floater and C. Gotsman, How to morph tilings injectively, J. Comp. Appl. Math. **101** (1999), 117–129.
5. M. S. Floater, One-to-one piecewise linear mappings over triangulations, to appear in Math. Comp.
6. H. Kneser, Lösung der Aufgabe 41, Jahresber. Deutsch. Math.-Verien. **35**, (1926), 123–124.
7. T. Radó, Aufgabe 41, Jahresber. Deutsch. Math.-Verien. **35**, (1926), 49.
8. W. T. Tutte, How to draw a graph, Proc. London Math. Soc. **13** (1963), 743–768.

Shape preserving interpolation by curves

T. N. T. Goodman

Department of Mathematics, University of Dundee, Dundee DD1 4HN.
tgoodman@maths.dundee.ac.uk

Abstract

A survey is given of algorithms for passing a curve through data points so as to preserve the shape of the data.

1 Introduction

We consider the problem of passing a curve through a finite sequence of points. We want the curve to preserve in some sense the shape of the data, i.e. the shape of the curve gained by joining the data by straight line segments (which we call the 'piecewise linear interpolant'). We do not consider the important problems of *approximating* the data by a curve, or of shape-preserving interpolation by a surface. The short length of the paper forces it to be selective. So we concentrate on actual algorithms for solving the problem rather than related theory. Also we consider only algorithms where the curve is defined explicitly, not implicitly either as the zero set of a function or as the limit of a subdivision process (though there are, to our knowledge, extremely few such implicit shape-preserving schemes).

In Section 2, we consider planar curves given by a function $y = f(x)$, often rather misleadingly referred to as 'functional interpolation'. There are numerous such schemes, dating from 1966, with most of them prior to 1990. Our treatment is therefore very selective. Section 3 deals with parametrically defined planar curves, for which the schemes are fewer and more recent. Finally, in Section 4, we consider curves in three dimensions, often called 'space curves'. Here the work is much more limited, dating only from 1997.

We note that in shape-preserving interpolation, the map from the data to the function describing the curve must be non-linear. In what we call 'tension methods' the curve can be constructed by a linear scheme for any choice of certain 'tension parameters'. These parameters are then varied so as to 'pull' the curve towards the piecewise linear interpolant until the shape criteria are satisfied. Though there are a few variations on this theme, there is generally a clear distinction between tension methods and other schemes, which we shall term 'direct methods'.

2 Functional interpolation

Given data

$$(x_i, y_i) \in R^2, \quad i = 0, \ldots, N, \quad x_0 < x_1 < \cdots < x_N, \tag{2.1}$$

we consider a function $f : [x_0, x_N] \to R$ satisfying

$$f(x_i) = y_i, \quad i = 0, \ldots, N. \tag{2.2}$$

For some reasons, perhaps the physical situation which f is intended to model, we may wish the graph of f to inherit certain shape properties of the data. We now describe these and other properties which it may be desirable for f to possess.

2.1 Desirable properties

Monotonicity. Here we require f to be increasing (respectively decreasing) if (y_i) is increasing (respectively decreasing). More generally we may require the scheme to be **'co-monotone'**, i.e. for $i = 0, \ldots, N-1$, f is increasing (decreasing) on $[x_i, x_{i+1}]$ if $y_i \leq y_{i+1}$ ($y_i \geq y_{i+1}$). Co-monotonicity has the consequence that the local extrema of f occur exactly at the local extrema of (y_i). Moreover if $y_i = y_{i+1}$, then f is constant on $[x_i, x_{i+1}]$. These properties may be too restrictive and a weaker alternative is what we call **'local monotonicity'**: for $i = 1, \ldots, N-2$, f is increasing on $[x_i, x_{i+1}]$ if $y_{i-1} \leq y_i \leq y_{i+1} \leq y_{i+2}$ (and similarly for decreasing). Although this is not generally stated, it is also desirable that for $i = 0, \ldots, N-1$, f has at most one local extremum on (x_i, x_{i+1}).

Convexity. Here we require f to be convex (concave) if the piecewise linear interpolant is convex (concave). More generally we call the scheme **'co-convex'** if for $i = 1, \ldots, N-2$, f is convex (concave) on $[x_i, x_{i+1}]$ if the piecewise linear interpolant is convex (concave) on $[x_{i-1}, x_{i+2}]$. It is also desirable in a co-convex scheme for f to have at most one inflection in (x_i, x_{i+1}), $0 \leq i \leq N-1$.

Smoothness. By definition, the piecewise linear interpolant is shape-preserving, and so the problem is trivial unless we require f to have greater smoothness than continuity, i.e. C^k for $k \geq 1$. Since all the schemes use piecewise analytic functions, the C^k condition needs to be checked only at a finite number of 'knots', which generally include the data points. We remark that smoothness and shape-preservation may not be compatible; e.g. if for $i = 0, \ldots, 4$, $x_i = i - 2$, $y_i = |x_i|$, and f is convex on $[x_0, x_4]$, then $f(x) = |x|$, $-2 \leq x \leq 2$, and so is not C^1 at 0.

Approximation order. It is generally supposed that the data arise as values of some unknown 'smooth' function g, i.e. $y_i = g(x_i)$, $i = 0, \ldots, N$. Then we can consider how fast the interpolant f converges to g as we increase the density of data values x_i in the fixed interval $[a, b]$. A scheme has approximation order $O(h^m)$ if $||f - g|| = O(h^m)$, where $h = \max\{x_{i+1} - x_i : i = 0, \ldots, N-1\}$ and the usual norm is $||F|| = \sup\{|F(x)| : a \leq x \leq b\}$.

Locality. In a 'global' scheme, the value $f(x)$, for any x, generally depends on all the data. In contrast, for a 'local' scheme, $f(x)$ depends on the data values (x_i, y_i) only for x_i 'near' x. There may be advantages in local schemes, e.g. when data are modified or inserted.

Fairness. It is often desirable that the curve is 'fair', i.e. pleasing to the eye, see Section 3.

Other desirable properties are invariance under scaling or reflection in x or y, and stability, i.e. small changes in the data produce small changes in f. There may also be other constraints on f, e.g. $f \geq 0$ when $y_i \geq 0$, $i = 0, \ldots, N$.

2.2 Tension methods

Many tension methods are a modification of cubic spline interpolation, which we now describe. Given data (2.1), there is a unique function f satisfying (2.2), where f is C^2, is a cubic polynomial on $[x_i, x_{i+1}]$, $i = 0, \ldots, N-1$, and satisfies suitable boundary conditions at x_0 and x_N. The function f minimises $\int_{x_0}^{x_N} (g'')^2$ over a suitable class of functions and this energy minimisation property is generally considered to give a fair curve. Determining f requires solving a global, strictly diagonally dominant tridiagonal system of linear equations.

Since cubic spline interpolation is not shape-preserving, in 1966 Schweikert [67] modified the scheme by replacing cubic polynomials on each interval $[x_i, x_{i+1}]$ by solutions of

$$f^{(4)} - \lambda_i f'' = 0,$$

where $\lambda_i \geq 0$. When $\lambda_i = 0$, f will reduce to a cubic, while as $\lambda_i \to \infty$, f approaches a linear polynomial. Thus λ_i acts as a tension parameter and by making appropriate choices of λ_i large enough the function will preserve monotonicity and/or convexity globally or locally.

Many papers have been written on Schweikert's tension splines giving, for example, ways of choosing the values of the tension parameters, e.g. [68,57,46,60]. However the fact that the method uses exponential functions can be seen as a drawback. An alternative was introduced by Nielson in 1974 [55] by adjusting the minimisation property of cubic splines to a minimisation problem involving also the first derivative. The resulting function, called a ν-spline, is also cubic on each interval $[x_i, x_{i+1}]$ but only C^1. However the form of the C^1 continuity gives extra 'smoothness' for parametrically defined curves and so we discuss ν-splines further in Section 3. By generalising the minimisation problem still further one can gain a C^1 piecewise cubic interpolant with further parameters for gaining shape properties [22].

The idea of using rational functions in tension methods was introduced by Späth [69], also in 1974, and put in a general setting of tension methods in [57]. From 1982–1988, Gregory and/or Delbourgo produced a series of algorithms using rational functions, e.g. [19,36,20,21,18]. We illustrate the ideas with an algorithm from [37]. Here f is C^2 and on each interval $[x_i, x_{i+1}]$ it has the form, for some a, b, c, d,

$$f(t) = \frac{a + bt + ct^2 + dt^3}{1 + \lambda_i t(1-t)}, \quad t = \frac{x - x_i}{x_{i+1} - x_i}.$$

For $\lambda_i > -1$, $i = 0, \ldots, N-1$, f can be determined as the solution of a strictly diagonally dominant tridiagonal linear system (and hence the scheme is global). When all $\lambda_i = 0$, f reduces to the usual cubic spline interpolant, while as $\lambda_i \to \infty$, f converges uniformly to the linear interpolant on $[x_i, x_{i+1}]$. In general the approximation order is $O(h^2)$ for

data from a C^4 function. In the special case of monotone data, choosing

$$\lambda_i = \mu_i + (f'(x_i) + f'(x_{i+1}))\frac{x_{i+1} - x_i}{y_{i+1} - y_i}, \quad \mu_i \geq -3, \quad i = 0, \ldots, N-1,$$

ensures that f is correspondingly monotone, and for the choice $\mu_i = -2$, f reduces to a rational quadratic which gives optimal approximation order $O(h^4)$. Similarly for convex data, f is also convex provided that each λ_i satisfies an inequality involving $f'(x_i)$, $f'(x_{i+1})$, and choosing λ_i appropriately (which requires solving a non-linear equation) further ensures approximation order $O(h^4)$.

There are some more recent methods involving rationals, e.g. [58].

The idea of using variable degree to preserve shape was introduced by McAllister, Passow and Roulier in 1977 [47,56]. They produce monotone, convex schemes of arbitrarily high smoothness by constructing a shape-preserving piecewise linear interpolant l with one knot between any two data points (and no knots at the data points) and then defining the final interpolant on each interval $[x_i, x_{i+1}]$ as the Bernstein polynomial of l of some degree m_i. The idea was extended from 1986 by Costantini [8–10]. For $k \geq 1$, $m_i \geq 2k+1$, $i = 0, \ldots, N-1$, he constructs a shape-preserving piecewise linear interpolant l with knots at $x_i + k(x_{i+1} - x_i)/m_i$ and $x_{i+1} - k(x_{i+1} - x_i)/m_i$, $i = 0, \ldots, N-1$. The final interpolant f coincides on each interval $[x_i, x_{i+1}]$ with the Bernstein polynomial of l of degree m_i and is hence C^k (with $f^{(j)}(x_i) = 0$, $j = 2, \ldots, k$). In [10] there is a co-monotone, co-convex scheme in which the degrees m_i can either be chosen a priori or computed automatically according to the data.

The above schemes using variable degree are not strictly tension schemes in our sense but in 1990, Kaklis and Pandelis [40] introduced a tension method by using the above form for $k = 1$, i.e. on each interval $[x_i, x_{i+1}]$ it has the form:

$$f(t) = f(x_i)(1-t) + f(x_{i+1})t + c_i t(1-t)^{m_i} + d_i t^{m_i}(1-t), \quad t = \frac{x - x_i}{x_{i+1} - x_i}.$$

Here $m_i \geq 2$ is an integer and for each choice of $m_0, \ldots, m_{N-1}$, the numbers c_i, d_i are chosen so that f is C^2, which requires the solution of a strictly diagonally dominant tridiagonal linear system. When all $m_i = 2$, this reduces to the usual cubic spline interpolant, while as $m_i \to \infty$, f converges uniformly to the linear interpolant on $[x_i, x_{i+1}]$ with order $O(m_i^{-1})$ (or $O(m_i^{-2})$ if m_{i-1}, m_{i+1} remain bounded). For further discussion of variable degree shape-preserving functional interpolation, see [11].

Our final type of tension method was introduced by Manni [50] in 1996. The general idea is to define f on $[x_i, x_{i+1}]$ as

$$f(x) = p_i(q_i^{-1}(x)),$$

where p_i, q_i are cubic polynomials on $[x_i, x_{i+1}]$ and q_i is strictly increasing from $[x_i, x_{i+1}]$ onto itself, so that the inverse q_i^{-1} is well-defined on $[x_i, x_{i+1}]$. For $f'(x_i) = d_i$, $i = 0, \ldots, N$, we require

$$p_i'(x_i) = \lambda_i d_i, \quad q_i'(x_i) = \lambda_i, \quad p_i'(x_{i+1}) = \mu_i d_{i+1}, \quad q_i'(x_{i+1}) = \mu_i,$$

for parameters $\lambda_i > 0$, $\mu_i > 0$. For $\lambda_i = \mu_i = 1$, we have $q_i(x) = x$ and f reduces to a cubic on $[x_i, x_{i+1}]$, while for $\lambda_i = \mu_i = 0$, f becomes linear on $[x_i, x_{i+1}]$.

In [50], the values $d_0, \ldots, d_N$ are assumed known (or estimated from the data values) and the scheme is local C^1, gives necessary and sufficient conditions for the values of the parameters λ_i, μ_i for co-monotonicity, and has approximation order $O(h^2)$ when g is C^2 and generally $O(h^4)$ when g is C^4.

Manni and co-workers have written a series of papers using the same idea, [51,53,54]. For example in [45], the values d_i are not assumed given but are chosen to ensure that the function is C^2, thus providing a locally monotone, co-convex global scheme which generalises usual cubic spline interpolation; while in [52] two further knots are inserted in each interval $[x_i, x_{i+1}]$ to produce a C^2, locally monotone, co-convex local scheme which interpolates values of $f^{(j)}(x_i)$, $j = 1, 2$, $i = 0, \ldots, N$.

2.3 Direct methods

In 1967, Young [71] considered shape-preserving interpolation by polynomials and a number of papers have appeared since on this topic, e.g. [59] gives a constructive proof of the existence of a co-monotone interpolant with an upper bound on the degree required. However for a practical algorithm, using a piecewise polynomial offers much more flexibility than a single polynomial. Numerous papers have been written using such polynomial splines and we mention briefly only a few.

By inserting extra knots between data points, a convexity preserving scheme with C^2 cubics was given by de Boor [4, p.303], and co-monotone, co-convex schemes with C^1 quadratics in [48,49,66]. C^1 cubic splines with knots at the data points are used for co-monotonicity in [25,5,24,70], (the last of these using a variational approach), and for both co-monotonicity and co-convexity in [16,17]. We also recall the methods using spline functions of variable degree with knots between the data points to obtain interpolants with arbitrarily high smoothness which were discussed under tension methods.

Finally we note that following the paper [62] which was as early as 1973, Schaback [63] gives a C^2 co-monotone, co-convex scheme which uses a cubic polynomial on any interval $[x_i, x_{i+1}]$ where an inflection is needed, and on other intervals employs a rational function of form quadratic/linear.

3 Planar curves

Given data

$$I_i \in R^2, \quad i = 0, \ldots, N,$$

we consider a curve $r : [a, b] \to R^2$ satisfying

$$r(t_i) = I_i, \quad i = 0, \ldots, N, \tag{3.1}$$

for values $a = t_0 < t_1 < \cdots < t_N = b$. For a closed curve the situation is extended periodically so that

$$I_{i+N} = I_0, \quad t_{i+N} = t_i, \quad i \in Z, \quad r(t + b - a) = r(t), \quad t \in R.$$

3.1 Desirable properties

Shape. For this case it is not usually relevant to consider preservation of monotonicity. We say a scheme is **'co-convex'** if the curve r has the minimum number of inflections

consistent with the data. In practice, schemes satisfy the somewhat stronger condition that for any $0 \leq i \leq j-2 \leq N-2$, r is positively (negatively) locally convex on $[t_{i+1}, t_{j-1}]$ if the polygonal arc joining $I_i, \ldots, I_j$ is positively (negatively) locally convex. For more details on this and other desirable properties, see [29].

Smoothness. We shall call the interpolating curve C^k for $k \geq 0$ if the function r is C^k. A C^0 curve r we shall call G^1 if the unit tangent vector is continuous, and G^2 if, in addition, the curvature is continuous. A C^k curve r is G^k, $k = 1, 2$, provided that the parameterisation is regular, i.e. $r'(t) \neq (0,0)$, which is generally desirable. It is usually sufficient to have G^k, rather than C^k, continuity if only the appearance of the curve is important and the choice of parameter t is not significant.

Fairness. Planar curves often arise in computer-aided design where it may be particularly important that the curve is pleasing to the eye. Though this is subjective, various criteria have been suggested to be relevant, such as magnitude, rate of change or monotonicity of the curvature. Some schemes include 'shape parameters' which can be manipulated by the designer to modify the shape of the curve.

Approximation order is not important in the context of design when the data are not considered to be taken from some unknown curve. Approximation order is related to reproduction of polynomial curves, and a related property for planar curves is reproduction of arcs of circles (or more generally conics); this cannot be done exactly by polynomials but it can be achieved by using rationals.

Locality and other desirable properties are similar to the functional case as described in Section 2.1, though it is generally more appropriate that the invariance is under a rotation and the same scaling in both x and y.

3.2 Tension methods

In Section 2.2 we mentioned Nielson's ν-splines [55]. Applying this scheme for both components of r gives a function r which is cubic on each interval $[t_i, t_{i+1}]$, is C^1 and satisfies

$$r''(t_i^+) = r''(t_i^-) + \nu_i r'(t_i), \quad i = 1, \ldots, N-1,$$

where $\nu_i \geq 0$. This condition is sufficient for G^2 continuity of r (assuming regular parameterisation). When all $\nu_i = 0$, r will reduce to the usual C^2 cubic spline interpolant. As $\nu_i \to \infty$, the curve is 'pulled tight' at I_i and as ν_i, $\nu_{i+1} \to \infty$, it approaches the linear interpolant on $[t_i, t_{i+1}]$.

The scheme in [37] by Gregory which was mentioned in Section 2.2 was adapted to the planar case in [38]. Other schemes using rationals were proposed by Clements in [6,7], where r is a C^2 curve which on each interval $[t_i, t_{i+1}]$ has the form, for some a, b, c, $d \in R^2$,

$$r(t) = \frac{a(1-s)^3}{w_i s + 1} + b(1-s) + cs + \frac{ds^3}{w_i(1-s)+1}, \quad s = \frac{t - t_i}{t_{i+1} - t_i},$$

where $w_i \geq 0$ are the tension parameters.

The variable degree tension method of [40], also mentioned in Section 2.2, was adapted to the planar case in [41], and extended in [27] to allow the designer to obtain a 'fair' curve by minimising the number of changes in the monotonicity of the curvature.

3.3 Direct methods

The papers [34,35,28,23] give local, G^2 co-convex schemes, e.g. in [28], a rational cubic/cubic is used on each interval $[t_i, t_{i+1}]$ and the tangent vectors and curvatures are stipulated by the algorithm to ensure that the convexity conditions are satisfied and circular arcs are reproduced, with the possibility of modifying the tangent vectors and curvatures further as shape parameters.

Following an earlier scheme in [64], Schaback in [65] gives a global G^2 co-convex scheme which uses a cubic polynomial on any interval $[t_i, t_{i+1}]$ where an inflection is needed, and on other intervals employs quadratic polynomials.

Sapidis and Kaklis [61] give a G^2 co-convex scheme by interpolating by a piecewise quintic curve tangent directions and curvatures gained by their tension method [41].

In [1] a local, co-convex G^2 scheme is given which uses polynomials of degree six and which attempts to obtain a fair curve by imposing conditions on the curvature to minimise measures of fairness. Finally we note that in [12] Costantini gives an abstract theory and general purpose code.

4 Space curves

Given data

$$I_i \in R^3, \quad i = 0, \ldots, N,$$

we consider a curve $r : [a, b] \to R^3$ satisfying condition (3.1) as before.

4.1 Desirable properties

What is meant by 'shape-preserving' is not so clear for space curves as for the planar case. Criteria were introduced by Kaklis and Karavelas [39] and extended by Ong and the author in [31]. We shall sketch these below. They are discussed in further detail in [30], where some further extensions are suggested. We write, for appropriate indices i:

$$L_i = I_{i+1} - I_i, \quad \Delta_i = \det[L_{i-1}, L_i, L_{i+1}], \quad N_i = L_{i-1} \times L_i.$$

Torsion. We ensure that the curve is 'twisting' in the same manner as the piecewise linear interpolant by requiring that if $\Delta_i \neq 0$, then the torsion of r has the same sign as Δ_i on (t_i, t_{i+1}).

Convexity. Let

$$K(t) = r'(t) \times r''(t), \quad a \leq t \leq b.$$

We require that for $1 \leq i \leq N-1$, $K(t_i).N_i > 0$, which means that the projection of the curve r onto the plane of I_{i-1}, I_i, I_{i+1}, has the same sign of local convexity at I_i as the polygonal arc $I_{i-1}I_iI_{i+1}$. Moreover if $N_i.N_{i+1} > 0$, we require

$$K(t).N_j > 0, \quad j = i, i+1, \quad t_i \leq t \leq t_{i+1},$$

which implies that the curve r has the same sign of local convexity on $[t_i, t_{i+1}]$ when projected in any direction $\lambda N_i + (1-\lambda)N_{i+1}$ for $0 \leq \lambda \leq 1$. Finally we require that if $N_i.N_{i+1} < 0$, then for $j = i,\ i+1$, $K(t).N_j$ has exactly one sign change in $[t_i, t_{i+1}]$, which imples that each of the above projections of r have just one inflection.

Smoothness. This is as for planar curves, except that we call the curve G^3 if it is G^2 and, in addition, the torsion is continuous. Other desirable properties are similar to the planar case.

4.2 Tension methods

Although interpolation by space curves with a special shape is considered in [44], the first specific shape-preserving interpolation scheme by space curves was due to Kaklis and Karavelas [39], who adapted the variable degree tension method of [40] to give a C^2 method which was also G^3, but at the expense of zero torsion at the data points. In [42] the same authors adapted Nielson's ν-splines to the three dimensional case to give a curve which is C^1 and G^2. The paper [14] also uses variable degree for tension parameters but gives a C^3 scheme in which the limiting curve as the tension goes to infinity is not the piecewise linear interpolant but the shape-preserving interpolant given by either of the above two schemes. In [15] a C^3 scheme is also given but here the components of r on each interval $[t_i, t_{i+1}]$ lie in the linear span of the functions

$$(1-u), u, (1-u)^{m_i}, (1-u)^{m_i-1}u, (1-u)u^{m_{i+1}-1}, u^{m_{i+1}}, \quad u = \frac{t-t_i}{t_{i+1}-t_i}.$$

When $m_i = m_{i+1} = 5$, this reduces to a quintic polynomial. As $m_i,\ m_{i+1} \to \infty$, it tends to a linear polynomial and then the curve r approaches the piecewise linear interpolant on $[t_i, t_{i+1}]$.

The paper [26] also uses variable degree splines with degree on each interval at least five, and the curve r also converges to the piecewise linear interpolant as the degrees go to infinity. However here the curve is C^4, which the authors feel may give extra fairness to the curve due, for example, to lowering the maximum absolute value of the curvature. Variable degree polynomial splines are also used in [13].

4.3 Direct methods

Following an earlier scheme in [31], Ong and the author gave a local G^2 scheme in [32] which employed a rational cubic/cubic between data points, extending the ideas of the planar scheme in [28]. This was further extended to a local G^3 scheme using a rational quartic/quartic in [43]. In [33], the degrees of freedom inherent in the scheme in [32] were used to optimise a fairness measure. Finally we mention the papers [2,3] which give local G^3 schemes using a piecewise polynomial of degree six, also allowing optimisation of a fairness measure.

It will be noted that many of the above papers are extremely recent and it is hoped that the unavoidable lack of detail here will serve to tantalise readers to discover for themselves more of this rapidly developing field.

Bibliography

1. S. Asaturyan, P. Costantini and C. Manni, G^2 shape preserving parametric planar curve interpolation, in Creating Fair and Shape-Preserving Curves and Surfaces, H. Nowacki, P. D. Kaklis (eds.), B. G. Teubner, Stuttgart (1998), 89–98.
2. S. Asaturyan, P. Costantini and C. Manni, Shape-preserving interpolating curves in R^3: a local approach, in Creating Fair and Shape-Preserving Curves and Surfaces, H. Nowacki, P. D. Kaklis (eds.), B. G. Teubner, Stuttgart (1998), 99–108.
3. S. Asaturyan, P. Costantini and C. Manni, Local shape-preserving interpolation by space curves, IMA J. Numer. Anal. **21** (2001), 301–325.
4. C. de Boor, A Practical Guide to Splines, Springer, New York (1978).
5. J. Butland, A method of interpolating reasonable-shaped curves through any data, Proc. Computer Graphics 80, Online Publ. Ltd., Northwood Hills, Middlesex, U.K. (1980), 409–422.
6. J. C. Clements, Convexity-preserving piecewise rational cubic interpolation, SIAM J. Numer. Anal. **27** (1990), 1016–1023.
7. J. C. Clements, A convexity-preserving C^2 parametric rational cubic interpolant, Numer. Math. **63** (1992), 165–171.
8. P. Costantini, On monotone and convex spline interpolation, Math. Comp. **46** (1986), 203–214.
9. P. Costantini, Co-monotone interpolating splines of arbitrary degree - a local approach, SIAM J. Sci. Stat. Comput. **8** (1987), 1026–1034.
10. P. Costantini, An algorithm for computing shape-preserving interpolating splines of arbitrary degree, J. Comput. Appl. Math. **22** (1988), 89–136.
11. P. Costantini, Abstract schemes for functional shape-preserving interpolation, in Advanced Course on Fairshape, J. Hoschek, P. Kaklis (eds.), B. G. Teubner, Stuttgart (1996), 185–199.
12. P. Costantini, Boundary-valued shape-preserving interpolating splines, ACM Trans. on Math. Software **23** (1997), 229–251.
13. P. Costantini, Curve and surface construction using variable degree polynomial splines, Computer Aided Geometric Design **17** (2000), 419–446.
14. P. Costantini, T. N. T. Goodman and C. Manni, Constructing C^3 shape preserving interpolating space curves, Advances Comp. Math. **14** (2001), 103–127.
15. P. Costantini and C. Manni, Shape-preserving C^3 interpolation: the curve case, to appear.
16. P. Costantini and R. Morandi, Monotone and convex cubic spline interpolation, Calcolo **21** (1984), 281–294.
17. P. Costantini and R. Morandi, An algorithm for computing shape-preserving cubic spline interpolation to data, Calcolo **21** (1984), 295–305.
18. R. Delbourgo, Shape preserving interpolation to convex data by rational functions with quadratic numerator and linear denominator, IMA J. Numer. Anal. **9** (1989), 123–136.
19. R. Delbourgo and J. A. Gregory, C^2 rational quadratic spline interpolation to mono-

tonic data, IMA J. Numer. Anal. **3** (1983), 141–152.
20. R. Delbourgo and J. A. Gregory, The determination of derivative parameters for a monotonic rational quadratic interpolant, IMA J. Numer. Anal. **5** (1985), 397–406.
21. R. Delbourgo and J. A. Gregory, Shape preserving piecewise rational interpolation, SIAM J. Sci. Stat. Comput. **6** (1985), 967–976.
22. T. A. Foley, A shape preserving interpolant with tension controls, Computer Aided Geometric Design **5** (1988).
23. T. A. Foley, T. N. T. Goodman and K. Unsworth, An algorithm for shape-preserving parametric interpolating curves with G^2 continuity, in Mathematical Methods in CAGD, T. Lyche, L. L. Schumaker (eds.), Academic Press, Boston (1989), 249–259.
24. F. N. Fritsch and J. Butland, A method for constructing local monotone piecewise cubic interpolants, SIAM J. Sci. Stat. Comput. **5** (1984), 300–304.
25. F. N. Fritsch and R. E. Carlson, Monotone piecewise cubic interpolation, SIAM J. Numer. Anal. **17** (1980), 238–246.
26. N. C. Gabrielides and P. D. Kaklis, C^4 interpolatory shape-preserving polynomial splines of variable degree, Computing **65** (2001), to appear.
27. A. Ginnis, P. Kaklis and N. S. Sapidis, Polynomial splines of non-uniform degree: controlling convexity and fairness, in Designing Fair Curves and Surfaces, N. S. Sapidis (ed.), SIAM Series on Geometric Design, Philadelphia (1994), Part 3, Chapter 10.
28. T. N. T. Goodman, Shape preserving interpolation by parametric rational cubic splines, in Numerical Mathematics Singapore 1988, R. P. Agarwal, Y. M. Chow, S. J. Wilson (eds.), International Series of Numerical Mathematics Vol. 86, Birkhauser Verlag, Basel (1988), 149–158.
29. T. N. T. Goodman, Shape preserving interpolation by planar curves, in Advanced Course on Fairshape, J. Hoschek, P. Kaklis (eds.), B. G. Teubner, Stuttgart (1996), 29–38.
30. T. N. T. Goodman and B. H. Ong, Shape preserving interpolation by curves in three dimensions, in Advanced Course on Fairshape, J. Hoschek, P. Kaklis (eds.), B. G. Teubner, Stuttgart (1996), 39–48.
31. T. N. T. Goodman and B. H. Ong, Shape preserving interpolation by space curves, Computer Aided Geometric Design **15** (1997), 1–17.
32. T. N. T. Goodman and B. H. Ong, Shape preserving interpolation by G^2 curves in three dimensions, in Curves and Surfaces with Applications in CAGD, A. LeMehauté, C. Rabut, L. L. Schumaker (eds.), Vanderbilt Univ. Press, Nashville (1997), 151–158.
33. T. N. T. Goodman, B. H. Ong and M. L. Sampoli, Automatic interpolation by fair, shape preserving, G^2 space curves, Computer-aided Design **30** (1998), 813–822.
34. T. N. T. Goodman and K. Unsworth, Shape preserving interpolation by parametrically defined curves, SIAM J. Numer. Anal. **25** (1988), 1453–1465.
35. T. N. T. Goodman and K. Unsworth, Shape preserving interpolation by curvature continuous parametric curves, Computer Aided Geometric Design **5** (1988), 323–

340.
36. J. A. Gregory, Shape preserving rational spline interpolation, in Rational Approximation and Interpolation, Graves-Morris, Saff and Varga (eds.), Springer-Verlag (1984), 431–441.
37. J. A. Gregory, Shape preserving spline interpolation, Computer-aided Design **18** (1986), 53–58.
38. J. A. Gregory and M. Sarfraz, A rational cubic spline with tension, Computer Aided Geometric Design **7** (1990), 1–13.
39. P. D. Kaklis and M. I. Karavelas, Shape preserving interpolation in R^3, IMA J. Numer. Anal. **17** (1997), 373–419.
40. P. D. Kaklis and D. G. Pandelis, Convexity-preserving polynomial splines of non-uniform degree, IMA J. Numer. Anal. **10** (1990), 223–234.
41. P. D. Kaklis and N. S. Sapidis, Convexity-preserving interpolating parametric splines of non-uniform polynomial degree, Computer Aided Geometric Design **12** (1995), 1–26.
42. M. I. Karavelas and P. D. Kaklis, Spatial shape-preserving interpolation using ν-splines, Numerical Algorithms **23** (2000), 217–250.
43. V. P. Kong and B. H. Ong, Shape preserving interpolation using Frenet frame continuous curves of order 3, to appear.
44. C. Labenski and B. Piper, Coils, Computer Aided Geometric Design **20** (1996), 1–29.
45. P. Lamberti and C. Manni, Shape preserving C^2 functional interpolation via parametric cubics, Numerical Algorithms, to appear.
46. R. W. Lynch, A method for choosing a tension factor for spline under tension interpolation, M.Sc. Thesis, Univ. of Texas at Austin (1982).
47. D. F. McAllister, E. Passow and J. A. Roulier, Algorithms for computing shape preserving spline interpolation to data, Math. Comp. **31** (1977), 717–725.
48. D. F. McAllister and J. A. Roulier, An algorithm for computing a shape preserving osculating quadratic spline, ACM Trans. Math. Software **7** (1981), 331–347.
49. D. F. McAllister and J. A. Roulier, Algorithm 574. Shape preserving osculating quadratic splines, ACM Trans. Math. Software **7** (1981), 384–386.
50. C. Manni, C^1 comonotone Hermite interpolation via parametric cubics, J. Comp. Appl. Math. **69** (1996), 143–157.
51. C. Manni, Parametric shape-preserving Hermite interpolation by piecewise quadratics, in Advanced Topics in Multivariate Approximation, F. Fontanella, K. Jetter, P. J. Laurent (eds.), World Scientific (1996), 211–228.
52. C. Manni, On shape preserving C^2 Hermite interpolation, BIT **14** (2001), 127–148.
53. C. Manni and P. Sablonnière, Monotone interpolation of order 3 by C^2 cubic splines, IMA J. Numer. Anal. **17** (1997), 305–320.
54. C. Manni and M. L. Sampoli, Comonotone parametric Hermite interpolation, in Mathematical Methods for Curves and Surfaces II, M. Daehlen, T. Lyche, L. L. Schumaker (eds.), Vanderbilt Univ. Press, Nashville (1998), 343–350.

55. G. M. Nielson, Some piecewise polynomial alternatives to splines under tension, in Computer Aided Geometric Design, R. E. Barnhill, R. F. Riesenfeld (eds.), Academic Press (1974), 209–235.
56. E. Passow and J. A. Roulier, Monotone and convex interpolation, SIAM J. Numer. Anal. **14** (1977), 904–909.
57. S. Pruess, Properties of splines in tension, J. Approx. Theory **17** (1976), 86–96.
58. R. Qu and M. Sarfraz, Efficient method for curve interpolation with monotonicity preservation and shape control, Neural, Parallel and Scientific Computations **5** (1997), 275–288.
59. L. Raymon, Piecewise monotone interpolation in polynomial type, SIAM J. Math. Anal. **12** (1981), 110–114.
60. N. S. Sapidis, P. D. Kaklis and T. A. Loukakis, A method for computing the tension parameters in convexity preserving spline-in-tension interpolation, Numer. Math. **54** (1988), 179–192.
61. N. S. Sapidis and P. D. Kaklis, A hybrid method for shape-preserving interpolation with curvature-continuous quintic splines, Computing Suppl. **10** (1995), 285–301.
62. R. Schaback, Spezielle rationale Splinefunktionen, J. Approx. Theory **7** (1973), 281–292.
63. R. Schaback, Adaptive rational splines, NAM-Bericht Nr. 60, Universität Göttingen (1988).
64. R. Schaback, Interpolation in R^2 by piecewise quadratic visually C^2 Bézier polynomials, Computer Aided Geometric Design **6** (1989), 219–233.
65. R. Schaback, On global GC^2 convexity preserving interpolation of planar curves by piecewise Bézier polynomials, in Mathematical Methods in CAGD, T. Lyche, L. L. Schumaker (eds.), Academic Press, Boston (1989), 539–548.
66. L. L. Schumaker, On shape preserving quadratic spline interpolation, SIAM J. Numer. Anal. **20** (1983), 854–864.
67. D. G. Schweikert, An interpolation curve using a spline in tension, J. Math. Phys. **45** (1966), 312–317.
68. H. Späth, Exponential spline interpolation, Computing **4** (1969), 225–233.
69. H. Späth, Spline algorithms for curves and surfaces, Utilitas Mathematica Pub. Inc., Winnipeg (1974).
70. F. I. Utreras and V. Celis, Piecewise cubic monotone interpolation: a variational approach, Departamento de Matematicas, Universidad de Chile, Tech. Report MA-83-B-281 (1983).
71. S. W. Young, Piecewise monotone polynomial interpolation, Bull. Amer. Math. Soc. **73** (1967), 642–643.

CAGD techniques for differentiable manifolds

Achan Lin and Marshall Walker

York University, Toronto M3J 1P3, Canada.
lin@yorku.ca, walker@yorku.ca

Abstract

The paper outlines procedures for extending the de Casteljau, de Boor and Aitken algorithms in such a way as to allow the construction on a Riemannian manifold of curves analogous to Bezier, B-spline, and Lagrange curves. These curves lie in the manifold and respect intrinsic geometry.

1 Introduction

Given a sequence of points in a Riemannian manifold M we describe methods for extending the de Casteljau, de Boor, and Aitken algorithms. These methods allow construction of corresponding interpolating or approximating curves that lie in the manifold and respect intrinsic geometry. In the case that the manifold is a sphere, opportunity for applications exist in the domain of geological and geographical mapping, for instance the creation of topographical contour lines or isotherms, and in the field of video production, where it is desirable to have smooth camera trajectories interpolating fixed camera positions. For higher dimensional manifolds there are applications in the field of data analysis. For the case of a sphere, there is an extensive literature dealing with the general problem of data fitting, and a superb review can be found in Fasshauer and Schumaker [2]. Shoemake [7] uses properties of quaternion arithmetic to describe curves on the unit quaternion sphere, and Levesley and Ragozin [4], using techniques different from those presented in this paper, describe methods for Lagrange interpolation in differentiable manifolds.

The techniques described in this paper come from the simple observation that in the de Casteljau, de Boor, and Aitken algorithms one may formally substitute appropriately parametrized geodesic arcs for straight line segments. These ideas are introduced in detail in the next section in the context of the blossoming paradigm, [6] and [3]. Unfortunately many of the useful properties of blossoms depend on the affine structure of Euclidean space which in general has no counter part in a Riemannian manifold. In particular, geodesic blossoms may be neither symmetric or multi-affine, and in general they do not possess uniqueness characteristics common to the Euclidean blossom.

For an arbitrary Riemannian manifold [1] or indeed an arbitrary differentiable 2-manifold embedded in $\mathbb{R}^3$, it may not be possible to construct unique shortest geodesic arcs between two points. However, if the manifold is compact or in the case that the two points lie in a sufficiently small neighborhood, such arcs are known to exist. But even

then, there appears to be no general method that allows explicit construction. So, the task of constructing geodesic blossoms becomes a study of special cases in which specific methods can be set forth. For the general case, a discrete variational method can be used to obtain good approximations.

In Section 3 a few specific examples are discussed. The case in which the manifold is a sphere is given special attention. There we introduce a variation which allows the discussion of Archimedian curves which are constructed by substituting Archimedian spirals for geodesics. This variation allows the natural construction of curves that lie off the sphere. Although the spherical geodesic blossoms are neither symmetric or multi-affine, a simple reparametrization of geodesic arcs results in spherical blossoms that have all desirable characteristics. Section 3 also contains a brief discussion of the problem of finding geodesics in developable surfaces and in surfaces of revolution.

2 Preliminaries

Let M be a C^∞ Riemannian manifold. There is the following theorem that guarantees the existence locally of geodesics.

Theorem 2.1 *If M is a Riemannian manifold, $x_0 \in M$. Then there exists a neighborhood V of x_0 and $\varepsilon > 0$ so that if $x \in V$ and v is a non-zero tangent vector at x and $\|v_x\| < \varepsilon$, then there is a unique C^∞ geodesic $\alpha : (-2, 2) \to M$ defined on the open interval $(-2, 2)$ such that $\alpha(0) = x$ and $\left(\frac{d\alpha}{dt}\right)_{t=0} = v_x$.*

For compact Riemannian manifolds there is the Hopf-Rinow theorem that tells us that points can be connected by geodesic arcs.

Theorem 2.2 *(Hopf and Rinow) If a connected Riemannian manifold M is compact, then any pair of points x and y may be joined by a geodesic whose length corresponds to the distance in the manifold from x to y.*

We also need the notion of geodesic convexity and the result of J. H. C. Whitehead that geodesically convex neighborhoods exist for all $x \in M$.

Definition 2.3 *Given a subset X of M and a point $x_0 \in X$, X is star shaped with respect to the point x_0, if for every $x \in X$ there is a unique shortest geodesic connecting x_0 with x which lies in X.*

Definition 2.4 *A subset X of M is geodesically convex if it is star shaped with respect to each of its points.*

Definition 2.5 *Given a subset A of a geodesically convex set X the geodesic convex hull of A is the smallest convex set which contains A.*

Theorem 2.6 *(J. H. C. Whitehead) Let V be an open subset of a Riemannian manifold M and let $x \in M$, then there is a geodesically convex open neighborhood U of x such that $U \subset V$.*

Let M be a Riemannian manifold and let X be a geodesically convex subset of M. Given points P_i in M we describe extensions of the de Castlejau, de Boor, and Aitken algorithms.

2.1 Riemannian Lagrange curves

Let M be a Riemannian manifold, and let $A = \{P_0, P_1, \cdots, P_n\}$ be a subset of a geodesically convex subset X. Given parameter points, $t_0 < t_1 < \cdots < t_n$, assume that A is contained in a sufficiently small neighborhood in which specified geodesics exist. For $0 \leq i \leq n-1$, define $\gamma_i^1 : [t_0, t_n] \to X$ to be the unique geodesic parametrized so that $\gamma_i^1(t_i) = P_i$ and $\gamma_i^1(t_{i+1}) = P_{i+1}$. For $1 < r \leq n$ and $0 \leq i \leq n-r$ define $\gamma_i^r : [t_0, t_n]^r \to X$ so that $\gamma_i^r(u_1, u_2, \cdots, u_{r-1}, \cdot)$ is the unique geodesic parametrized so that $\gamma_i^r(u_1, u_2, \cdots, u_{r-1}, t_i) = \gamma_i^{r-1}(u_1, u_2, \cdots, u_{r-1})$ and $\gamma_i^r(u_1, u_2, \cdots, u_{r-1}, t_{i+r}) = \gamma_{i+1}^{r-1}(u_1, u_2, \cdots, u_{r-1})$. The function $\gamma_0^n : [t_0, t_n]^n \to X$ is called the *geodesic Aitken blossom* associated with the points $P_i \in X$, $0 \leq i \leq n$ and the parameter points, $t_0 < t_1 < \cdots < t_n$. If $\Delta : [t_0, t_n] \to [t_0, t_n]^n$ is the *diagonal map* defined by $\Delta(u) = (\underbrace{u, u, \cdots, u}_{n})$, the *geodesic Lagrange curve* associated with X and the points P_i is the function $\Gamma_0^n = \gamma_0^n \circ \Delta$.

Theorem 2.7 *If $\Gamma_0^n : [t_0, t_n] \to M$ is the geodesic Lagrange curve associated with the points $P_i \in M$, $0 \leq i \leq n$, as defined above, then $\Gamma_0^n(t_i) = P_i$.*

Proof: Observe that for $1 \leq r \leq n$ and $0 \leq i \leq n-r$, γ_i^r depends for its definition only on the points, P_j, where $i \leq j \leq i+r$. If $n = 1$, and we are given points, P_0 and P_1, the result follows from the definition of γ_0^1 . Inductively assume it is true for $k < n$. For $k = n$, if $i = 0$, by definition

$$\Gamma_0^n(t_0) = \gamma_0^n(\underbrace{t_0, t_0, \cdots, t_0}_{n}) = \gamma_0^{n-1}(\underbrace{t_0, t_0, \cdots, t_0}_{n-1}) = \cdots = \gamma_0^1(t_0) = P_0$$

and likewise if $i = n$, $\Gamma_0^n(t_n) = \gamma_0^n(\underbrace{t_n, t_n, \cdots, t_n}_{n}) = \gamma_0^{n-1}(\underbrace{t_n, t_n, \cdots, t_n}_{n-1}) = \cdots = \gamma_0^1(t_n) = P_n$. For $i \neq 0$ and $i \neq n$, observe that the geodesics used in the construction of γ_0^{n-1} and γ_1^{n-1} may be restricted respectively to the intervals $[t_0, t_{n-1}]$ and $[t_1, t_n]$ so that γ_0^{n-1}becomes the geodesic Aitken blossom associated with the points $P_0, P_1, \cdots, P_{n-1}$and the parameter points $t_0 < t_1 < \cdots < t_{n-1}$, and γ_1^{n-1} becomes geodesic Aitken blossom associated with the points $P_1, P_2, \cdots, P_n$ and the parameter points $t_1 < t_2 < \cdots < t_n$. By the deductive assumption, $\gamma_0^{n-1}(\underbrace{t_i, t_i, \cdots, t_i}_{n-1}) = P_i = \gamma_1^{n-1}(\underbrace{t_i, t_i, \cdots, t_i}_{n-1})$, and consequently $\gamma_0^n(\underbrace{t_i, t_i, \cdots, t_i}_{n-1}, \cdot)$ is the geodesic connecting $\gamma_0^{n-1}(\underbrace{t_i, t_i, \cdots, t_i}_{n-1})$ with $\gamma_1^{n-1}(\underbrace{t_i, t_i, \cdots, t_i}_{n-1})$, and is thus the constant function, $\gamma_0^n(\underbrace{t_i, t_i, \cdots, t_i}_{n}, u) = P_i$ for all $u \in [t_0, t_n]$. Thus in particular, $\gamma_0^n(\underbrace{t_i, t_i, \cdots, t_i}_{n}) = \Gamma_0^n(t_i) = P_i$. □

2.2 Riemannian Bézier curves

Following the previous format we introduce a Riemannian version of the de Casteljau algorithm. Accordingly, let X be a geodesically convex subset of a Riemannian manifold M. Let $A = \{P_0, P_1, \cdots, P_n\}$ be a subset of X. Define $\gamma_i^0 : [0, 1] \to X$ by $\gamma_i^0(u) = P_i$. For $1 \leq r \leq n$ and $0 \leq i \leq n - r$ define $\gamma_i^r : [0, 1]^r \to X$ to be the unique geodesic with the property that $\gamma_i^r(u_1, u_2, \cdots, u_{r-1}, 0) = \gamma_i^{r-1}(u_1, u_2, \cdots, u_{r-1})$ and $\gamma_i^r(u_1, u_2, \cdots, u_{r-1}, 1) = \gamma_{i+1}^{r-1}(u_1, u_2, \cdots, u_{r-1})$. The function $\gamma_0^n : [0, 1]^n \to X$ is called the *geodesic de Casteljau blossom* associated with the set A. If $\Delta : [0, 1] \to [0, 1]^n$ is the diagonal map, the *geodesic Bézier curve* associated with X and the set A is the function $\Gamma_0^n = \gamma_0^n \circ \Delta$.

2.3 Riemannian B-Spline curves

Given $A = \{P_0, P_1, \cdots, P_n\}$ contained in a geodesically convex subset X of a Riemannian manifold M, and given knots $t_1 < t_2 < \cdots < t_{2n}$, define $\gamma_i^0 : [t_1, t_{2n}] \to X$ by $\gamma_i^0(t) = P_i$, for $0 \leq i \leq n$. For $1 \leq r \leq n$ and $r \leq i \leq n$, define $\gamma_i^r : [t_i, t_{i+n+1-r}]^r \to X$ to be the unique geodesic with the property that $\gamma_i^r(u_1, u_2, \cdots, u_{r-1}, t_i) = \gamma_{i-1}^{r-1}(u_1, u_2, \cdots, u_{r-1})$ and $\gamma_i^r(u_1, u_2, \cdots, u_{r-1}, t_{i+n+1-r}) = \gamma_i^{r-1}(u_1, u_2, \cdots, u_{r-1})$. The function $\gamma_n^n : [t_n, t_{n+1}]^n \to X$ is called the *geodesic de Boor blossom* associated the set A. If $\Delta : [t_n, t_{n+1}] \to [t_n, t_{n+1}]^n$ is the diagonal map, the *geodesic B-Spline curve* associated with X and the points P_i is the function $\Gamma_n^n = \gamma_n^n \circ \Delta$.

We have the following results, which follow from the fact that both the geodesic de Casteljau and the geodesic de Boor blossoms are constructed from successive geodesic combinations beginning with the set $A = \{P_0, P_1, \cdots, P_n\}$.

Theorem 2.8 *Given $A = \{P_0, P_1, \cdots, P_n\}$ contained in a geodesically convex subset of a Riemannian manifold, if $\gamma_0^n : [0, 1]^n \to X$ is the geodesic de Casteljau blossom of A, then $\gamma_0^n([0, 1]^n)$ is contained in the geodesic convex hull of the set A.*

Theorem 2.9 *Given $A = \{P_0, P_1, \cdots, P_n\}$ contained in a geodesically convex subset of a Riemannian manifold, if $\gamma_0^n : [t_n, t_{n+1}]^n \to X$ is the geodesic de Boor blossom of A relative to a knot sequence $t_1 < t_2 < \cdots < t_{2n}$, then $\gamma_0^n([t_n, t_{n+1}]^n)$ is contained in the geodesic convex hull of the set A.*

Since each of the three blossoms are constructed successively from C^∞ geodesics, it follows that the blossoms and their restrictions to the diagonal are also of class C^∞.

Theorem 2.10 *The geodesic Lagrange, Bézier, B-spline curves are of class C^∞ as are each of their corresponding blossoms.*

3 Examples

The impediments to implementation of these ideas depend on the manifold in question. In all cases it is necessary that the points P_i should lie in a region in which it is possible to construct geodesic arcs between points. The problem then reduces to that of finding methods for such constructions. Even in cases for which this is possible, there is the additional problem that many of the desirable properties associated with B-spline or Bézier curves in $\mathbb{R}^3$ may have no direct analogs. Many properties such as the ability

to subdivide a curve depend on the blossom being symmetric or multi-affine, and for the generalizations presented here, this is seldom true. For the case of an orientable 2-manifold embedded in $\mathbb{R}^3$, there are in many cases good solutions to the problem of finding geodesics, but different classes of surfaces lead to different solution. In this section we mention a few. In the case that the manifold M is the 2-sphere S^2 a preliminary version of our results is reported in [5].

3.1 The sphere

In the case that $M = S^2$, a small alteration to methods presented so far allows the consideration of curves that lie off the sphere. Given points P and Q that lie off the sphere consider radial projections to points $\tilde{P}$ and $\tilde{Q}$ and let $\tilde{\gamma} : [a,b] \to S^2$ be a geodesic with the property that $\tilde{\gamma}(a) = \tilde{P}$ and $\tilde{\gamma}(b) = \tilde{Q}$. The curve $\gamma : [a,b] \to \mathbb{R}^3$ defined by

$$\gamma(t) = \left(\tfrac{b-t}{b-a} \cdot \|P\| + \tfrac{t-a}{b-a} \cdot \|Q\|\right) \cdot \tilde{\gamma}(t)$$

is called the *Archimedian spiral* connecting the points P and Q. To explicitly describe the curve $\tilde{\gamma}$, set $\tilde{P} = v_1$, $\tilde{Q} = v_2$ and for simplicity consider the parameter interval $[a,b]$ to be the unit interval $[0,1]$. For $< \cdot, \cdot >$ the standard inner product on $\mathbb{R}^3$ set

$$v_3 = (< v_1, v_2 > v_1 - v_2)/(\|< v_1, v_2 > v_1 - v_2\|)$$

so that v_3 is orthogonal to v_1 and in the plane containing v_1 and v_2. Letting $\theta =< v_1, v_2 >$ denote the angle between v_1 and v_2, the geodesic $\tilde{\gamma}$ connecting v_1 with v_2 is defined by

$$\begin{aligned} \tilde{\gamma}(t) &= \cos(t\theta)v_1 + \sin(t\theta)v_3 \\ &= \left(\cos(t\theta) + \frac{\sin(t\theta) < v_1, v_2 >}{\|< v_1, v_2 > v_1 - v_2\|}\right) v_1 - \frac{\sin(t\theta)}{\|< v_1, v_2 > v_1 - v_2\|} v_2. \end{aligned}$$

The corresponding Archimedian Lagrange, Bézier and B-spline curves may now be constructed with the general algorithms of Section 2.

One of the difficulties that arise with Archimedian curves is that geodesic blossoms are not necessarily symmetric or multi-affine. It is even not clear what these concepts might mean in a geodesic context. Consequently, certain results that hold for normal Bézier or B-spline curves that depend on these properties are no longer valid. In particular analogs of the subdivision algorithms that allow one to determine control points of a portion of a given Bézier or B-spline are not valid. However, it can be shown that a simple non-linear change in the parametrization of the geodesic arcs, makes it possible to recapture most of what is needed.

Definition 3.1 *Given two points A and B on the sphere. Let C be the smaller arc of the spherical geodesic joining A with B. The barycentric parametrization of C on the parameter interval $[a,b]$ is the function $\alpha : [a,b] \to C$ defined by*

$$\alpha(t) = q(x(t)),$$

where $x(t) = \frac{(b-t)}{b-a}A + \frac{(t-a)}{b-a}B$ and $q : \mathbb{R}^3 \to S^2$ is the radial projection $q(x) = \dfrac{x}{\|x\|}$.

In the following we prove a spherical version of the Menelaus theorem.

Theorem 3.2 *Given 3 points P_0, P_1, P_2 on S^2 let $\gamma : [0,1]\times[0,1] \to \mathbb{R}^3$ be the geodesic de Casteljau blossom in which all geodesic arcs are given the barycentric parametrization. Then $\gamma(s,t) = \gamma(t,s)$.*

Proof: Observe that an elementary geometric argument tells us that:

$$\begin{aligned}\gamma(s,t) = \gamma_0^2(s,t) &= q\left((1-t)\gamma_0^1(s) + t\gamma_1^1(s)\right) \\ &= q(\ (1-t)[(1-s)P_0 + sP_1] + t[(1-s)P_1 + sP_2]\ )\end{aligned}$$

and

$$\begin{aligned}\gamma(t,s) = \gamma_0^2(t,s) &= q\left((1-s)\gamma_0^1(t) + s\gamma_1^1(t)\right) \\ &= q(\ (1-s)[(1-t)P_0 + tP_1] + s[(1-t)P_1 + tP_2]\ ).\end{aligned}$$

And the result follows from the affine properties of $\mathbb{R}^3$. □

As an immediate consequence we have

Theorem 3.3 *Given points $P_0, P_1, \cdots, P_n$ on S^2, the associated de Casteljau blossom, in which geodesic arcs are given barycentric parametrization, is symmetric.*

The conventional blossoming description of subdivision can now be employed. From the blossom construction we can conclude that $\gamma_0^n(0,0,\cdots,0,\underbrace{1,1,\cdots,1}_{i}) = P_i$. In particular, it follows that, for $0 < u < 1$, the points $Q_i = \gamma_0^n(0,0,\cdots,0,\underbrace{u,u,\cdots,u}_{i})$ describe a geodesic de Casteljau blossom which is parametrized to the interval $[0,u]$ and which, because of the uniqueness of geodesic arcs, equals the restriction of γ_0^n to $[0,u]^n$. Likewise, for the interval $[u,1]$ the points $R_i = \gamma_0^n(\underbrace{u,u,\cdots,u}_{i},\ 1,1,\cdots,1)$ determine a geodesic de Casteljau blossom which is parametrized to the interval $[u,1]$ and which equals the restriction of γ_0^n to $[u,1]^n$ Therefore, if $g : [0,1] \to S^2$ is the geodesic Bézier curve determined by $P_0, P_1, \cdots, P_n$ and if $g = \gamma_0^n \circ \Delta$, it follows that, $g\,|_{[0,u]} : t \mapsto \gamma_0^n(t,t,\cdots,t,\ \underbrace{u,u,\cdots,u}_{i})$ and $g\,|_{[1,u]} : t \mapsto \gamma_0^n(\underbrace{u,u,\cdots,u}_{i},\ t,t,\cdots,t)$, for $0 < u < 1$.

More generally and along the lines of the proof above, we have the following theorem which allows all familiar properties of both Bézier and B-spline curves which have descriptions in terms of their corresponding blossoms to carry over to the spherical case.

Theorem 3.4 *Let $f : [0,1]^n \to \mathbb{R}^3$ be the Euclidean blossom generated by the de Casteljau algorithm using points $P_i \in S^2, 0 \le i \le n$. Then $\gamma_0^n = q \circ f$.*

3.2 Other surfaces

We briefly discuss two examples in which explicit descriptions of geodesics between points are possible.

A developable surface S [4], described as the image of a function $f : U \to \mathbb{R}^3$ for U an open subset of $\mathbb{R}^2$, possess the characteristic, among others, that distances are

preserved by the function f. Therefore, a geodesic in the surface $f(U)$ may be considered as the image of a straight line in the plane. If $P_0, P_1, \cdots, P_n$ are points in S, let $Q_i = f^{-1}(P_i)$, $0 \leq i \leq n$. If $C \subset U$ is the Lagrange, Bézier, or B-spline curve obtained from the standard Euclidean versions of the algorithms, then it follows that $f(C)$ is the corresponding geodesic curve in S that would have been obtained using geodesic versions of the algorithms that we have described.

For surfaces of revolution the description of geodesics between two points is rather more involved. Let C be a curve in the yz-plane described implicitly by

$$\begin{cases} f(y) = z \\ x = 0 \end{cases},$$

for (y, z) belonging to some open set U contained in the upper half of the yz-plane. The surface S obtained by rotating C about the z-axis may be expressed as $g^{-1}(0)$ where $g : \mathbb{R} \times U \to \mathbb{R}$ is defined by $g(x, y, z) = f(\sqrt{x^2 + y^2}) - z = 0$. In polar coordinates letting $u = \sqrt{x^2 + y^2}$, we express S in the form

$$\begin{cases} x = u\cos\theta \\ y = u\sin\theta \\ z = f(u) \end{cases}.$$

Let $P = (u_1\cos\theta_1, u_1\sin\theta_1, f(u_1))$ and $Q = (u_2\cos\theta_2, u_2\sin\theta_2, f(u_2))$ be two points on S. Then it may be shown that the geodesic connecting P with Q is the function $\alpha : [u_1, u_2] \to S$ such that $\alpha(u) = (u\cos\theta(u), \sin\theta(u), f(u))$, where for fixed u_0,

$$\theta(u) = \int_{u_0}^{u} \sqrt{\frac{1 + (f'(t))^2}{\frac{1}{c^2}t^4 - t^2}}\, dt + c',$$

and constants c and c' satisfy the following equations:

$$\theta_2 - \theta_1 = \int_{u_1}^{u_2} \sqrt{\frac{1 + (f'(u))^2}{\frac{1}{c^2}u^4 - u^2}}\, du$$

$$c' = \theta_1 - \int_{u_0}^{u_1} \sqrt{\frac{1 + (f'(u))^2}{\frac{1}{c^2}u^4 - u^2}}\, du.$$

For complete details see [6].

4 Conclusion and future research

We have outlined a procedure by which conventional computer aided design constructions may be extended to arbitrary Riemannian manifolds. In practice, there are difficulties. In a given manifold points to be interpolated or approximated must lie in a region in which it is possible to construct necessary geodesic arcs. Supposing this the case, one then needs to find explicit descriptions of the geodesics. And then there is the question of the additional characteristics which the curves might possess. The paper raises more questions than it answers. In the case of a sphere, good results are obtained, and it

is also possible to add variation that allows consideration of curves off the sphere but which project radially to geodesic Lagrange, Bézier, or B-spline curves. It is also shown, in the spherical case, that a change parametrization of geodesics results in blossoms that retain the desirable characteristics associated with Euclidean blossoms. For surfaces of revolution and developable surfaces, we know that geodesics can be found between points so the geodesic blossom constructions will always exist. It is however unlikely that these blossoms will be either symmetric or multi-affine; these characteristics depend on the affine structure of $\mathbb{R}^3$. Thus, in the case of a general Riemannian manifold, although the constructions may be valid, it is not clear that we will be able to employ fundamental operations such as subdivision which depend on the symmetry of the blossom. We have outlined three different methods of blossom construction, one for each of the algorithms considered. In the Euclidean case, we know that there is a unique symmetric, multi-affine polynomial that restricts to a given polynomial on the diagonal. This may not be true in our more general setting.

Bibliography

1. Conlon, L., *Differential Manifolds, a First Course,* Birkhäuser, Boston, 1993.
2. Fasshauer, G. E. and Schumaker, L. L., Data Fitting on the Sphere, in *Mathematical Methods for Curves and Surfaces II,* Daehlen, M., Lyche, T., and Schumaker, L. L. (eds), Vanderbilt University Press, Nashville, 1998, 117–166.
3. Gallier, J., *Curves and Surfaces in Geometric Modeling, Theory and Applications,* Morgan Kaufman, San Francisco, 2000.
4. Opera, J., *Differential Geometry and its Applications,* Prentice Hall, Upper Saddle River, NJ, 1997.
5. Levesley, J., and Ragozin, D. L., Local Approximation on Manifolds Using Radial Basis Functions and Polynomials, in *Curve and Surface Fitting,* Cohen, A., Rabut, C.R., Schumaker, L.L. (eds), Vanderbilt University Press, Nashville, 2000, 291–301.
6. Lin, A., *Geodesics between points on surfaces of revolution,* Tech. Report, Dept. Mathematics, York University, Toronto, May 2001.
7. Ramshaw L., Blossoming: A Connect the Dots Approach to Splines, Digital Systems Research Center, Report 19, Palo Alto, CA, 1987.
8. Shoemake, K., Animating Rotation with Quaternion Curve, *ACM Proceedings*, San Francisco, July 22–26, **9,** 1985, 245–254.
9. Walker, M., Curves over a Sphere, preprint, 2000.

Parametric shape-preserving spatial interpolation and $\nu-$splines

Carla Manni

Department of Mathematics, University of Torino, Italy
manni@dm.unito.it

Abstract

In this paper we present a class of C^2 spatial interpolating curves depending on a set of tension parameters and we illustrate their ability to reproduce the shape of the data. The curves are constructed using cubic splines and basically reduce to classical ν-splines for particular values of the tension parameters.

1 Introduction

Shape-preserving interpolation via functional as well as parametric splines is a well studied topic for the planar case. On the other hand, shape-preserving interpolation for spaces curves is considerably more complex than for planar ones and the related literature is apparently limited. On this concern, a considerable part of the available schemes only ensures geometric continuity of the obtained curve (see [1, 8] and references quoted therein). Recently, C^2 and C^3 shape-preserving interpolating space curves have been obtained using polynomial splines of variable degree, [2, 3, 6]. However, working with low(fixed)-degree polynomial splines seems to be a standard choice in the CAD/CAM community. This motivates the careful investigation of shape preserving properties of cubic $\nu-$splines recently carried out in [7] and the present paper.

In this paper we present a method for constructing C^2 spatial interpolating curves reproducing the shape of the polygonal line which interpolates the given data. The curve is constructed via the so called "parametric approach", [10], using classical cubic splines. The shape of the curve is controlled by the amplitude of the tangent vectors at the data sites which play the role of tension parameters. It turns out that, for particular values of the tension parameters, the proposed scheme provides a new, geometrically evident, description of classical $C^1 - G^2$ cubic $\nu-$splines, [11]. Moreover, the method produces a suitable reparameterization for the above mentioned curves ensuring C^2 continuity. The reparameterization is a cubic polynomial involving the tension parameters (see (3.3)). Thus, the evaluation of the curve for a fixed value of the new parameter requires the solution of a cubic equation.

The geometric meaning of the tension parameters coupled with the powerful "shape-preserving" properties of the Bernstein-Bézier representation can be efficiently used to construct an iterative algorithm for C^2 shape-preserving interpolation. The algorithm

converges in a finite number of iterations and requires at each iteration the solution of a diagonally dominant linear system.

The paper is organized as follows. In Section 2 we state the problem. In Section 3 we describe the construction of the required interpolant and we illustrate its dependence on the tension parameters. The asymptotic behavior and the shape-preserving properties of the obtained curve are briefly discussed in Section 4. We conclude in Section 5 with a graphical example.

2 The problem

In this section we introduce the problem of *shape-preserving* interpolation by curves in $\mathbb{R}^3$. The adopted notion of shape-preserving follows the definitions of [2] and [6]. Let

$$\mathbf{I}_i \in \mathbb{R}^3,\ i = 0, \ldots, N,$$

be the interpolation points with $\mathbf{I}_i \neq \mathbf{I}_{i+1}$. Define, for all admissible indices,

$$\begin{aligned}
&\mathbf{L}_i := \mathbf{I}_{i+1} - \mathbf{I}_i,\\
&\mathbf{N}_i := \begin{cases} \frac{\mathbf{L}_{i-1}\times\mathbf{L}_i}{\|\mathbf{L}_{i-1}\times\mathbf{L}_i\|}, & \text{if } \|\mathbf{L}_{i-1}\times\mathbf{L}_i\| > 0,\\ \mathbf{0}, & \text{elsewhere,}\end{cases}\\
&\Delta_i := \begin{cases} \frac{|\mathbf{L}_{i-1}\ \ \mathbf{L}_i\ \ \mathbf{L}_{i+1}|}{\|\mathbf{L}_{i-1}\times\mathbf{L}_i\|\ \|\mathbf{L}_i\times\mathbf{L}_{i+1}\|}, & \text{if } \|\mathbf{L}_{i-1}\times\mathbf{L}_i\|\ \|\mathbf{L}_i\times\mathbf{L}_{i+1}\| > 0,\\ 0, & \text{elsewhere,}\end{cases}
\end{aligned}$$

where $|\mathbf{a}\ \mathbf{b}\ \mathbf{c}|$ denotes the determinant of the matrix with columns $\mathbf{a}, \mathbf{b}, \mathbf{c}$. The vectors $\mathbf{N}_i$ and the scalars Δ_i are, respectively, the discrete binormals and the discrete torsions of the data.

Let the parameter values $\sigma_i,\ i = 0, \ldots, N$, with $\sigma_i < \sigma_{i+1}$ be given, and let

$$h_i := \sigma_{i+1} - \sigma_i,\ i = 0, 1, \ldots, N-1$$

be the corresponding spacings. We wish to construct a curve $\mathbf{Q}(s),\ s \in [\sigma_0, \sigma_N]$, which interpolates the data, $\mathbf{Q}(\sigma_i) = \mathbf{I}_i,\ i = 0, \ldots, N$, such that $\mathbf{Q} \in C^2[\sigma_0, \sigma_N]$. In addition, we also require that $\mathbf{Q}(s)$ is shape-preserving, that is it reproduces the convexity and torsion of the polygonal line connecting the interpolation points. More specifically, denoting with dashes derivatives with respect to the parameter s, we define

$$\mathbf{K}(s) := \frac{\mathbf{Q}'(s)\times\mathbf{Q}''(s)}{\|\mathbf{Q}'(s)\|^3},\ \text{if}\ \ \mathbf{Q}'(s) \neq \mathbf{0},\ \tau(s) := \frac{|\mathbf{Q}'(s)\ \ \mathbf{Q}''(s)\ \ \mathbf{Q}'''(s)|}{\|\mathbf{Q}'(s)\times\mathbf{Q}''(s)\|^2},\ \ \text{if } \mathbf{K}(s) \neq \mathbf{0} \tag{2.1}$$

as the *curvature vector* and the *torsion* of the curve respectively. $\mathbf{Q}(s)$ is shape-preserving if it satisfies the following criteria ([2, 6, 7]).

(i) *Convexity criteria:*
- (i.1) if $\mathbf{N}_i \cdot \mathbf{N}_{i+1} > 0$, then $\mathbf{K}(s) \cdot \mathbf{N}_j > 0, j = i, i+1,\ s \in [\sigma_i, \sigma_{i+1}]$,
- (i.2) if $\mathbf{N}_i \cdot \mathbf{N}_{i+1} < 0$, then $\mathbf{K}(s) \cdot \mathbf{N}_j$, $j = i, i+1$, has one change in sign in $[\sigma_i, \sigma_{i+1}]$,
- (i.3) if $\mathbf{N}_i \cdot \mathbf{N}_j \neq 0$ then $(\mathbf{K}(\sigma_i) \cdot \mathbf{N}_j)(\mathbf{N}_i \cdot \mathbf{N}_j) > 0,\ j = i-1, i, i+1$.

(ii) *Torsion criteria:* if $\Delta_i \neq 0$ then $\tau(s)\Delta_i > 0,\ s \in [\sigma_i^+, \sigma_{i+1}^-]$.

For the sake of brevity we refer to [7] for the more technical *collinearity* and *coplanarity* criteria.

3 Constructing the interpolating curve

In order to construct the curve $\mathbf{Q}$ we consider, as a first step, a cubic curve $\mathbf{C}$ interpolating the data. We put

$$\mathbf{C}(t)_{|[\sigma_i,\sigma_{i+1}]} := \mathbf{C}_i(t;\lambda_i^{(0)},\lambda_i^{(1)}), \tag{3.1}$$

$$\mathbf{C}_i(t;\lambda_i^{(0)},\lambda_i^{(1)}) := \mathbf{I}_i H_0^{(0)}(u) + \mathbf{I}_{i+1} H_1^{(0)}(u) + \lambda_i^{(0)} h_i \mathbf{T}_i H_0^{(1)}(u) + \lambda_i^{(1)} h_i \mathbf{T}_{i+1} H_1^{(1)}(u),$$
$$t \in [\sigma_i,\sigma_{i+1}], \quad u := (t-\sigma_i)/h_i, \tag{3.2}$$

where $0 < \lambda_i^{(0)}, \lambda_i^{(1)} \leq 1$ are shape parameters, $\mathbf{T}_i$, $\mathbf{T}_{i+1}$ are vectors to be determined and $H_i^{(j)}(u)$ denote the elements of the cardinal basis for cubic Hermite interpolation, that is $H_i^{(j)}(u)$ are the polynomials of third degree such that

$$\frac{d^l H_i^{(j)}(r)}{du^l} = \delta_{lj}\delta_{ri}, \ r,l = 0,1.$$

One can immediately verify that the curve (3.2) interpolates the points $\mathbf{I}_i$, $\mathbf{I}_{i+1}$ at the extremes of the interval $[\sigma_i,\sigma_{i+1}]$ and has tangent vectors $\lambda_i^{(0)}\mathbf{T}_i$, $\lambda_i^{(1)}\mathbf{T}_{i+1}$ at the same extremes. The parameters $\lambda_i^{(0)},\lambda_i^{(1)}$ determine the amplitude of the tangent vectors of the curve at the two end points of the interval and they control the shape of the curve. To be more specific, since $H_0^{(0)}(u)+H_1^{(0)}(u) = 1$, we have that $\mathbf{C}_i(t;0,0)$ reduces to the line through $\mathbf{I}_i$, $\mathbf{I}_{i+1}$. Thus, the parameters $\lambda_i^{(0)},\lambda_i^{(1)}$ act as *tension parameters* stretching the curve from the classical Hermite cubic interpolating $\mathbf{I}_i$, $\mathbf{I}_{i+1}$ with tangents $\mathbf{T}_i$, $\mathbf{T}_{i+1}$ ($\lambda_i^{(0)},\lambda_i^{(1)} = 1$) to the line segment ($\lambda_i^{(0)},\lambda_i^{(1)} = 0$). The curve (3.1) turns out to be of class G^1.

Let us consider now the new global parameter

$$s(t)_{|[\sigma_i,\sigma_{i+1}]} := s_i(t;\lambda_i^{(0)},\lambda_i^{(1)}) := \sigma_i H_0^{(0)}(u) + \sigma_{i+1} H_1^{(0)}(u) + \lambda_i^{(0)} h_i H_0^{(1)}(u) + \lambda_i^{(1)} h_i H_1^{(1)}(u). \tag{3.3}$$

It is not difficult to see that, if

$$0 < \lambda_i^{(0)},\lambda_i^{(1)} \leq 1 \tag{3.4}$$

then

$$\frac{ds_i(t;\lambda_i^{(0)},\lambda_i^{(1)})}{dt} > 0, \quad t \in [\sigma_i,\sigma_{i+1}].$$

Thus (3.3) implicitly defines a function $t = t(s)$, which provides a reparameterization for (3.1). In the following we assume that conditions (3.4) hold and we define

$$\mathbf{Q}(s) := \mathbf{C}(t(s)). \tag{3.5}$$

Since $\mathbf{Q}'(\sigma_i) = \mathbf{T}_i$, $i = 0,\ldots,N$, $\mathbf{Q}$ is of class C^1. For each sequence of the tension

parameters $\lambda_i^{(0)}, \lambda_i^{(1)}$ we will determine the tangent vectors $\mathbf{T}_i, \mathbf{T}_{i+1}$ so that $\mathbf{Q}$ is also of class C^2. Let us denote by dots derivatives with respect to the local parameter u. Imposing continuity of $\mathbf{Q}''(s)$ at σ_i, $i = 1, \ldots, N-1$, from (3.3), (3.5) and from the chain rule for derivatives, we obtain

$$\frac{\ddot{\mathbf{C}}_{i-1}(1^-)h_{i-1}\lambda_{i-1}^{(1)} - \ddot{s}_{i-1}(1^-)h_{i-1}\lambda_{i-1}^{(1)}\mathbf{T}_i}{(h_{i-1}\lambda_{i-1}^{(1)})^3} = \frac{\ddot{\mathbf{C}}_i(0^+)h_i\lambda_i^{(0)} - \ddot{s}_i(0^+)h_i\lambda_i^{(0)}\mathbf{T}_i}{(h_i\lambda_i^{(0)})^3}. \tag{3.6}$$

Thus, after some manipulations, from (3.2) we have

$$u_i\mathbf{T}_{i-1} + \mathbf{T}_i + v_i\mathbf{T}_{i+1} = \mathbf{z}_i, \quad i = 1, \ldots, N-1, \tag{3.7}$$

$$\begin{aligned} u_i &= \frac{h_{i-1}\lambda_{i-1}^{(0)}(h_i\lambda_i^{(0)})^2}{w_i}, \\ v_i &= \frac{h_i\lambda_i^{(1)}(h_{i-1}\lambda_{i-1}^{(1)})^2}{w_i}, \\ w_i &= h_{i-1}(3-\lambda_{i-1}^{(0)})(h_i\lambda_i^{(0)})^2 + h_i(3-\lambda_i^{(1)})(h_{i-1}\lambda_{i-1}^{(1)})^2, \\ \mathbf{z}_i &= \frac{3}{w_i}\mathbf{L}_i(h_{i-1}\lambda_{i-1}^{(1)})^2 + \frac{3}{w_i}\mathbf{L}_{i-1}(h_i\lambda_i^{(0)})^2. \end{aligned} \tag{3.8}$$

In order to uniquely determine the vectors $\mathbf{T}_i$ we need two additional equations that will be obtained by imposing boundary conditions. Classical boundary conditions are *periodic conditions:*

$$u_0\mathbf{T}_{N-1} + \mathbf{T}_0 + v_0\mathbf{T}_1 = \mathbf{z}_0, \quad u_N\mathbf{T}_{N-1} + \mathbf{T}_N + v_N\mathbf{T}_1 = \mathbf{z}_N$$

(with $u_0, v_0, u_N, v_N, \mathbf{z}_0, \mathbf{z}_N$ defined according to (3.8) setting $h_{-1} = h_{N-1}$, $\lambda_{-1}^{(0)} = \lambda_{N-1}^{(0)}$, $\lambda_{-1}^{(1)} = \lambda_{N-1}^{(1)}$, $\mathbf{L}_{-1} = \mathbf{L}_{N-1}$, $h_N = h_0$, $\lambda_N^{(0)} = \lambda_0^{(0)}$, $\lambda_N^{(1)} = \lambda_0^{(1)}$, $\mathbf{L}_N = \mathbf{L}_0$) and *end tangent conditions*:

$$\mathbf{T}_0 = \mathbf{D}_0, \qquad \mathbf{T}_N = \mathbf{D}_N,$$

(where $\mathbf{D}_0, \mathbf{D}_N$ are given in input). In the following we will denote by $\mathcal{I}$ the set of indices $\{1, \ldots, N-1\}$ ($\{0, \ldots, N\}$) when end tangent (periodic) conditions are considered. It is not difficult to see that (3.7) for any choice of the above mentioned boundary conditions provide a diagonally dominant system

$$A\mathbf{T} = \mathbf{z}. \tag{3.9}$$

Thus we can state the following

Theorem 3.1 *For any sequence $\lambda_i^{(0)}, \lambda_i^{(1)}$, $i = 0, \ldots, N-1$, satisfying (3.4), there exists a unique $\mathbf{Q} \in C^2[\sigma_0, \sigma_N]$ defined via (3.1)-(3.3), (3.5) which interpolates the given data and satisfies periodic or end tangent conditions.*

We notice that for $\lambda_k^{(0)} = \lambda_k^{(1)} = 1$, system (3.9) reduces to the system for the computation of classical C^2 cubic splines. Moreover, if $\lambda_{k-1}^{(1)} = \lambda_k^{(0)} = \lambda_k$, $k \in \mathcal{I}$, the

curve $\mathbf{C}$ is of class C^1 and equation (3.6) reads

$$\frac{d^2}{dt^2}\mathbf{C}_i(\sigma_i^+) - \frac{d^2}{dt^2}\mathbf{C}_{i-i}(\sigma_i^-) = \frac{h_i^{-2}\ddot{s}_i(0^+) - h_{i-1}^{-2}\ddot{s}_{i-1}(1^-)}{\lambda_i}\frac{d}{dt}\mathbf{C}_i(\sigma_i^+).$$

Then (3.6) is equivalent to impose that the cubic curve (3.1) is a C^1–G^2 cubic $\nu-$*spline* [5, 7, 11] where, from (3.3), for $i \in \mathcal{I}$

$$\nu_i := \frac{h_i^{-2}\ddot{s}_i(0^+) - h_{i-1}^{-2}\ddot{s}_{i-1}(1^-)}{\lambda_i} = \frac{(6-4\lambda_i-2\lambda_{i+1})h_i^{-1} + (6-2\lambda_{i-1}-4\lambda_i)h_{i-1}^{-1}}{\lambda_i}. \tag{3.10}$$

4 Asymptotic behavior and shape-preservation

In this section we briefly discuss the asymptotic behavior and the resulting shape-preserving properties of the curve $\mathbf{Q}$, defined by (3.1)–(3.3), (3.5) and (3.9), as the tension parameters $\lambda_i^{(0)}, \lambda_i^{(1)}$ approach zero. The following lemma (see also [7]) concerns the asymptotic behavior of the tangents $\mathbf{T}_i$. We omit the details of the proof which are completely analogous to those of Theorem 3 in [9].

Lemma 4.1 *The vectors* $\mathbf{T}_i$, $i = 0, \ldots, N$, *obtained from (3.9) are bounded independently of* $\lambda_j^{(0)}, \lambda_j^{(1)}$, $j = 0, \ldots, N-1$. *Moreover,*

$$\begin{aligned}\lim_{\lambda_{i-1}^{(0)},\, \lambda_i^{(1)} \to 0} \mathbf{T}_i &= \frac{h_i(\lambda_i^{(0)})^2}{h_i(\lambda_i^{(0)})^2 + h_{i-1}(\lambda_{i-1}^{(1)})^2}\frac{\mathbf{L}_{i-1}}{h_{i-1}} + \frac{h_{i-1}(\lambda_{i-1}^{(1)})^2}{h_{i-1}(\lambda_{i-1}^{(1)})^2 + h_i(\lambda_i^{(0)})^2}\frac{\mathbf{L}_i}{h_i} \\ &=: (1-\alpha_i)\frac{\mathbf{L}_{i-1}}{h_{i-1}} + \alpha_i\frac{\mathbf{L}_i}{h_i}, \quad i \in \mathcal{I}. \end{aligned} \tag{4.1}$$

Since the tangents are bounded independently on the tension parameters, from the previous section we have that $\mathbf{Q}$ approaches the piecewise linear function interpolating the data as the tension parameters tend to zero. Moreover, each tangent $\mathbf{T}_i$ determined by (3.9) tends to a strictly convex combination of $\mathbf{L}_{i-1}/h_{i-1}$ and $\mathbf{L}_i/h_i$ as the tension parameters $\lambda_{i-1}^{(0)}, \lambda_i^{(1)}$ tend to zero while $\lambda_{i-1}^{(1)}/\lambda_i^{(0)}$ remains bounded and strictly positive. Due to these two main facts, we are able to easy control the shape of the curve $\mathbf{Q}$ and to ensure that it reproduces the shape of the data as the tension parameters approach zero as we will discuss briefly in the following.

Since $\mathbf{C}$ and $\mathbf{Q}$ only differ for a reparameterization they have the same image. Thus, as far as the shape-preserving properties are concerned, we can consider the expression of $\mathbf{C}$. As noticed in Section 3, if $\lambda_{i-1}^{(1)} = \lambda_i^{(0)}$, $i \in \mathcal{I}$, the curve $\mathbf{C}$ with $\mathbf{T}_j$ obtained by (3.9), is a C^1-G^2 cubic ν-spline. In such a case, using (3.10), the careful shape analysis carried out in [7] and the resulting algorithm can be considered. However, the simple geometric meaning of the tension parameters $\lambda_i^{(0)}, \lambda_i^{(1)}$ coupled with the "shape-preserving" properties of the Bézier-Bernstein representation, allow us to more easily establish the shape-preserving results also for completely general configurations of $\lambda_{i-1}^{(1)}$, $\lambda_i^{(0)}$. Thus, we express the

curve segment $\mathbf{C}_i(t;\lambda_i^{(0)},\lambda_i^{(1)})$ in Bézier-Bernstein form:

$$\mathbf{C}_i(t;\lambda_i^{(0)},\lambda_i^{(1)}) = \sum_{l=0}^{3}\mathbf{C}_{i,l}\binom{3}{l}t^l(1-t)^{3-l},$$

$$\mathbf{C}_{i,0} := \mathbf{I}_i,\ \mathbf{C}_{i,1} := \mathbf{I}_i + \tfrac{1}{3}h_i\lambda_i^{(0)}\mathbf{T}_i,\ \mathbf{C}_{i,2} := \mathbf{I}_{i+1} - \tfrac{1}{3}h_i\lambda_i^{(1)}\mathbf{T}_{i+1},\ \mathbf{C}_{i,3} := \mathbf{I}_{i+1}.$$

Let us consider at the beginning the convexity criteria.

Lemma 4.2 *If* $\mathbf{N}_i \cdot \mathbf{N}_j \neq 0$ *and* $\frac{\lambda_{i-1}^{(1)}}{\lambda_i^{(0)}} \to c > 0$, *then*

$$\lim_{\lambda_{i-1}^{(0)},\lambda_i^{(1)}\to 0}(\mathbf{K}(\sigma_i)\cdot\mathbf{N}_j)(\mathbf{N}_i\cdot\mathbf{N}_j) > 0.$$

Proof: From the properties of Bézier curves (see [5]) and from (2.1) and (3.5)

$$\begin{aligned}\operatorname{sgn}(\mathbf{K}(\sigma_i)\cdot\mathbf{N}_j) &= \operatorname{sgn}((\mathbf{C}_{i,1}-\mathbf{C}_{i,0})\times(\mathbf{C}_{i,2}-\mathbf{C}_{i,1}))\cdot\mathbf{N}_j \\ &\quad \operatorname{sgn}\left(\left[\mathbf{T}_i\times\left(\mathbf{L}_i - \frac{\lambda_i^{(0)}h_i}{3}\mathbf{T}_i - \frac{\lambda_i^{(1)}h_i}{3}\mathbf{T}_{i+1}\right)\right]\cdot\mathbf{N}_j\right)\end{aligned}$$

where $\operatorname{sgn}(y)$ denotes the sign of y. Moreover, from (4.1)

$$\lim_{\lambda_{i-1}^{(0)},\lambda_i^{(1)}\to 0}(\mathbf{T}_i\times\mathbf{L}_i)\cdot\mathbf{N}_j = \left(\alpha_i\frac{\mathbf{L}_i}{h_i}\times\mathbf{L}_i + (1-\alpha_i)\frac{\mathbf{L}_{i-1}}{h_{i-1}}\times\mathbf{L}_i\right)\cdot\mathbf{N}_j = \frac{(1-\alpha_i)}{h_{i-1}}\mathbf{N}_i\cdot\mathbf{N}_j.$$

Hence, we obtain the assertion if $\mathbf{N}_i\cdot\mathbf{N}_j \neq 0$. □

The previous lemma ensures that, if $\lambda_{i-1}^{(0)},\lambda_i^{(1)}$ are small enough the third convexity criterion, (i.3), stated in Section 2 is satisfied. In addition, the sign of $\mathbf{K}(\sigma_k)\cdot\mathbf{N}_j$, $k = i, i+1$ can be checked considering the Bézier coefficients $\mathbf{C}_{i,l}$, $l = 0,1,2,3$, of $\mathbf{C}_i$. Furthermore, thanks to the shape-preserving properties of totally positive bases, for small values of the tension parameters, (see [4]) the number of changes in sign of $\mathbf{K}(s)\cdot\mathbf{N}_j$, $s \in [\sigma_i,\sigma_{i+1}]$ is bounded by the number of changes of sign in the pair $\mathbf{K}(\sigma_k)\cdot\mathbf{N}_j, k = i, i+1$. Thus, also the first and the second convexity criteria (i.1) and (i.2) are satisfied if the tension parameters are small enough.

As far as the torsion is concerned, we recall that the sign of the torsion of a cubic curve coincides with the sign of the discrete torsion of its Bézier control polygon (see for example [5]) thus it is not difficult to obtain the following

Lemma 4.3 *If* $\Delta_i \neq 0$ *and* $\frac{\lambda_{j-1}^{(1)}}{\lambda_j^{(0)}} \to c > 0$, $j = i, i+1$, *then*

$$\lim_{\lambda_{i-1}^{(0)},\lambda_{i-1}^{(1)},\lambda_i^{(0)},\lambda_i^{(1)},\lambda_{i+1}^{(0)}\lambda_{i+1}^{(1)}\to 0}\tau(s)\Delta_i > 0,\ s \in [\sigma_i^+,\sigma_{i+1}^-].$$

With similar arguments it is not difficult to prove that also the collinearity and the coplanarity criteria stated in [7] are fulfilled as the tension parameters approach zero. We omit the details for the sake of brevity.

Summarizing, from the previous discussion it follows that if the tension parameters are small enough then the Bézier control polygon of $\mathbf{C}$ reproduces the shape of the data and

the curve $\mathbf{C}$ does the same thanks to the properties of Bézier-Bernstein representation. Thus, to obtain an automatic algorithm to compute the C^2 interpolant $\mathbf{Q}$ defined by (3.5), satisfying convexity and torsion criteria, basically we have to perform the following steps:

(a) for a given sequence of the tension parameters solve the system (3.9) and compute the Bézier coefficients of the resulting curve $\mathbf{C}$;

(b) check if the control polygon of each segment $\mathbf{C}_i$ satisfies the convexity and torsion criteria;

(c) if this is not the case reduce the values of the related tension parameters according to a given rule and go to step (a).

5 A graphical example

To illustrate the performance of the presented scheme we consider the data proposed in [7], Example 2, consisting of 20 points with uniform parameterization in $[0, 1]$. End tangent boundary conditions have been used (see Table 2 in [7]). Figures 1–3 show the behavior of the obtained C^2 curve $\mathbf{Q}$ compared with the classical C^2 cubic spline. The shape-preserving curve $\mathbf{Q}$ is defined by the following sequence of tension parameters

$$\lambda_i^{(0)}: \quad .6 \;\; .6 \;\; 1 \;\; .9 \;\; .9 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; .75 \;\; 1 \;\; 1 \;\; 1 \;\; 1$$
$$\lambda_i^{(1)}: \quad .9 \;\; .6 \;\; .6 \;\; 1 \;\; .9 \;\; .9 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; .75 \;\; 1.$$

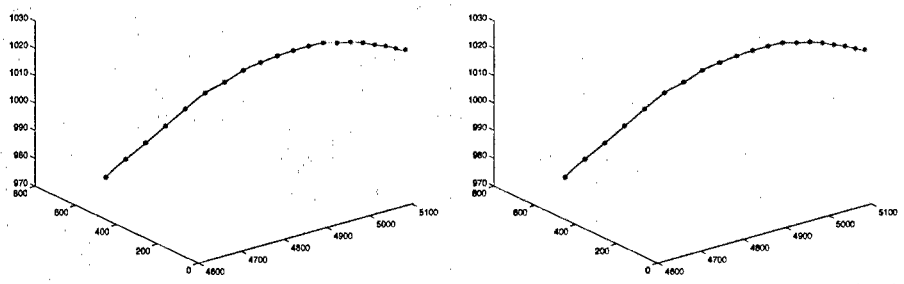


FIG. 1. C^2 cubic spline (left) and $\mathbf{Q}$ (right).

Bibliography

1. S. Asaturyan, P. Costantini and C. Manni, Local shape–preserving interpolation by space curves, *IMA J. Numer. Anal.* **21** (2001), 301–325.
2. P. Costantini, T. N. T. Goodman and C. Manni, Constructing C^3 shape–preserving interpolating space curves, *Adv. Comput. Math.* **14** (2001), 103–127.
3. P. Costantini and C. Manni, Shape-preserving C^3 interpolation: the curve case, *Adv. Comput. Math.* (2002) to appear.
4. T. N. T. Goodman, Total positivity and the shape of curves, in *Total Positivity and its Applications*, M. Gasca and C. A. Micchelli (eds), Kluwer, 1996, 157–186.

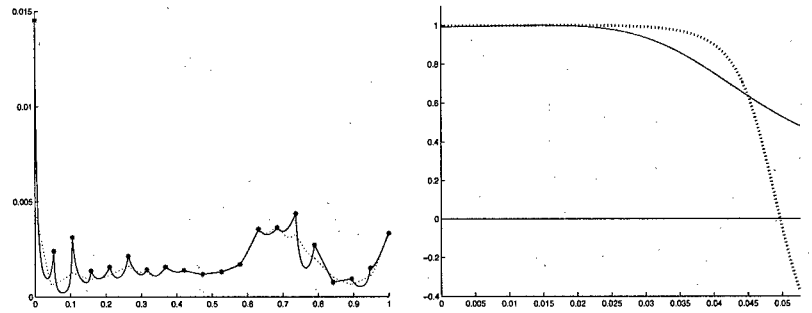


FIG. 2. Left: $\|\mathbf{K}(s)\|$ for the C^2 cubic spline (dotted line) and for $\mathbf{Q}$. Right: convexity ratio $\frac{\mathbf{K}(s)\cdot\mathbf{N}_0}{\|\mathbf{K}(s)\|}$ in $[\sigma_0, \sigma_1]$ (with $\mathbf{N}_0 := \frac{\mathbf{T}_0\times\mathbf{L}_0}{\|\mathbf{T}_0\times\mathbf{L}_0\|}$) for the C^2 cubic spline (dotted line) and for $\mathbf{Q}$.

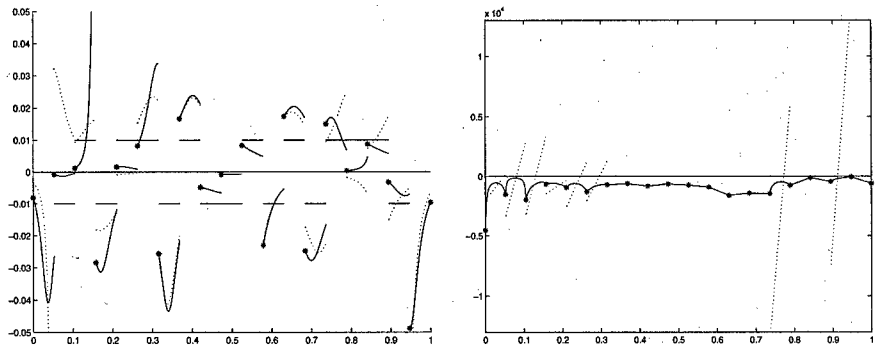


FIG. 3. Left: torsion of the C^2 cubic spline (dotted line) and of $\mathbf{Q}$ (the horizontal lines depict the sign of the discrete torsion). Right: first component of $d^2\mathbf{C}/dt^2$ (dotted line) and of $d^2\mathbf{Q}/ds^2$.

5. J. Hoschek and D. Lasser, *Fundamentals of Computer Aided Geometric Design*, A. K. Peters Ltd., 1993.
6. P. D. Kaklis and M. I. Karavelas, Shape preserving interpolation in $\mathcal{R}^3$, *IMA J. Numer. Anal.* **17** (1997), 373–419.
7. M. I. Karavelas and P. D. Kaklis, Spatial shape–preserving interpolation using ν–splines, *Numer. Algorithms* **23** (2000), 217–250.
8. V. P. Kong and B. H. Ong, Shape Preserving Interpolation using Frenet Frame Continuous Curve of Order 3, (2001) preprint.
9. P. Lamberti and C. Manni, Shape-preserving C^2 functional interpolation via parametric cubics, *Numer. Algorithms* **28** (2001), 229–254.
10. C. Manni, On Shape Preserving C^2 Hermite Interpolation, *BIT* **41** (2001), 127–148.
11. G. Nielson, Some piecewise polynomial alternative to spline under tension, in *Computer Aided Geometric Design,* R. E. Barnhill and R. F. Riesenfeld (eds) Academic Press, 1974, 209–235.

On the q-Bernstein polynomials

Halil Oruç and Necibe Tuncer

Department of Mathematics, Dokuz Eylül University, Tinaztepe Kampüsü
35160 Buca İzmir, Turkey
halil.oruc@deu.edu.tr, necibe.tuncer@deu.edu.tr

Abstract

We discuss here recent developments on the convergence of the q-Bernstein polynomials $B_n f$ which replaces the classical Bernstein polynomial with a one parameter family of polynomials. In addition, the convergence of iterates and iterated Boolean sum of q-Bernstein polynomial will be considered. Moreover a $q-$difference operator $\mathcal{D}_q f$ defined by $\mathcal{D}_q f = f[x, qx]$ is applied to q-Bernstein polynomials. This gives us some results which complement those concerning derivatives of Bernstein polynomials. It is shown that, with the parameter $0 < q \leq 1$, if $\Delta^k f_r \geq 0$ then $\mathcal{D}_q^k B_n f \geq 0$. If f is monotonic so is $\mathcal{D}_q B_n f$. If f is convex then $\mathcal{D}_q^2 B_n f \geq 0$.

1 Introduction

First we begin by introducing some notations to be used. For any fixed real number $q > 0$, the q-integer $[k]$ is defined as

$$[k] = \begin{cases} (1-q^k)/(1-q), & q \neq 1, \\ k, & q = 1, \end{cases}$$

for all positive integer k. The term Gaussian coefficient is also used, since they were first studied by Gauss (see Andrews [1]).

Let $p(N, M, n)$ denote the number of partitions of a positive integer n into at most M parts, each less than or equal to N. Then the Gaussian polynomial, $G(N, M, n)$, appears as the generating function

$$G(N, M, n) = \begin{bmatrix} N+M \\ M \end{bmatrix} = \sum_{n \geq 0} p(N, M, n) q^n.$$

Note that $\begin{bmatrix} n \\ k \end{bmatrix}$ defined by

$$\begin{bmatrix} n \\ k \end{bmatrix} = \begin{cases} \frac{[n]!}{[r]![n-k]!}, & n \geq k \geq 0, \\ 0, & \text{otherwise}, \end{cases}$$

where $[n]! = [n][n-1]\cdots[1]$ with $[0]! = 1$, is called Gaussian polynomial (or q-binomial coefficient) since it is a polynomial in q with the degree $(n-k)k$. The q-binomial coeffi-

cients satisfy the recurrence relations,

$$\begin{bmatrix} n+1 \\ k \end{bmatrix} = q^{n-k+1} \begin{bmatrix} n \\ k-1 \end{bmatrix} + \begin{bmatrix} n \\ k \end{bmatrix} \tag{1.1}$$

and

$$\begin{bmatrix} n+1 \\ k \end{bmatrix} = \begin{bmatrix} n \\ k-1 \end{bmatrix} + q^k \begin{bmatrix} n \\ k \end{bmatrix}. \tag{1.2}$$

The following Euler identity can be verified using the recurrence relation (1.1) by induction that

$$(1+x)(1+qx)\cdots(1+q^{k-1}x) = \sum_{r=0}^{k} q^{r(r-1)/2} \begin{bmatrix} k \\ r \end{bmatrix} x^r. \tag{1.3}$$

Phillips [8] introduced a generalization of Bernstein polynomials (q-Bernstein polynomials) in terms of q-integers

$$B_n(f;x) = \sum_{r=0}^{n} f_r \begin{bmatrix} n \\ r \end{bmatrix} x^r \prod_{s=0}^{n-r-1} (1-q^s x), \tag{1.4}$$

where $f_r = f\left(\frac{[r]}{[n]}\right)$ and an empty product denotes 1. When $q = 1$ the (1.4) reduces the classical Bernstein polynomials. The $B_n(f;x)$ generalizes many properties of classical Bernstein polynomials. Firstly, generalized Bernstein polynomials satisfy the end point interpolation

$$B_n(f;0) = f(0), \quad B_n(f;1) = f(1).$$

Phillips [8] also states the generalization of well known forward difference form (see Davis [3]) of the classical Bernstein polynomials by the following theorem.

Theorem 1.1 *The generalized Bernstein polynomial, defined by (1.4), may be expressed in the q-difference form*

$$B_n(f;x) = \sum_{r=0}^{n} \begin{bmatrix} n \\ r \end{bmatrix} \Delta^r f_0 x^r \tag{1.5}$$

where $\Delta^r f_i = \Delta^{r-1} f_{i+1} - q^{r-1}\Delta^{r-1} f_i$ *for* $r \geq 1$ *and* $\Delta^0 f_i = f_i$.

It is easily verified by induction that q-differences satisfy

$$\Delta^r f_i = \sum_{k=0}^{r} (-1)^k q^{k(k-1)/2} \begin{bmatrix} r \\ k \end{bmatrix} f_{r+i-k}. \tag{1.6}$$

Using the q-difference form of the q-Bernstein polynomials (1.5), one may show that q-Bernstein polynomials reproduce linear functions, since $B_n(1;x) = 1; \quad B_n(x;x) = x$.

2 Convergence

In the discussion of the uniform convergence of the q-Bernstein operator, the Bohman-Korovkin Theorem (see Cheney [2]) is used as in the classical case. The Bohman-Korovkin Theorem states that for a *linear monotone* operator $\mathcal{L}_n$, the convergence of

$\mathcal{L}_n f \to f$ for $f(x) = 1, x, x^2$ is sufficient for the sequence of operators $\mathcal{L}_n$ to have the uniform convergence property $\mathcal{L}_n f \to f$, $\forall f \in C[0,1]$. Observe that the q-Bernstein operator is a *monotone linear* operator for $0 < q \leq 1$. For a fixed value of q with $0 < q < 1$

$$[n] \to \frac{1}{1-q} \quad as \quad n \to \infty.$$

Notice that, since $B_n(x^2;x) = x^2 + \frac{x(1-x)}{[n]}$, $B_n(x^2;x)$ does not converge to x^2. Phillips [8] studies the uniform convergence of q-Bernstein polynomial.

Theorem 2.1 *Let $q = q_n$ satisfy $0 < q_n < 1$ and let $q_n \to 1$ as $n \to \infty$. Then,*

$$B_n(f;x) \to f(x), \quad \forall f(x) \in C[0,1].$$

The degree of q-Bernstein approximation to a bounded function on $[0,1]$ may be described in terms of the *modulus of continuity* with the following theorem.

Theorem 2.2 *If f is bounded on $[0,1]$ and $B_n f$ denotes the generalized Bernstein operator associated with f defined by (1.4), then*

$$\|f - B_n f\|_\infty \leq \frac{3}{2}\omega(1/[n]^{1/2}).$$

An error estimate for the convergence of q-Bernstein polynomials is given in Phillips [8] by the Voronvskaya type theorem.

Theorem 2.3 *Let f be bounded on $[0,1]$ and let x_0 be a point of $[0,1]$ at which $f''(x_0)$ exists. Further, let $q = q_n$ satisfy $0 < q_n < 1$ and let $q_n \to 1$ as $n \to \infty$. Then the rate of convergence of the sequence of generalized Bernstein polynomials is governed by*

$$\lim_{n\to\infty} [n](B_n(f;x_0) - f(x_0)) = \frac{1}{2}x_0(1-x_0)f''(x_0).$$

It is well known that the classical Bernstein polynomials $B_n f$ provide simultaneous approximation of the function and its derivatives. That is if $f \in C^p[0,1]$, then

$$\lim_{n\to\infty} B_n^{(p)}(f;x) = f^{(p)}(x)$$

uniformly on $[0,1]$. It is worthwhile to examine if this property hold for q-Bernstein polynomials. Phillips [7] proved that the p^{th} derivative of q-Bernstein polynomials converges uniformly on $[0,1]$ to the p^{th} derivative of f under some restrictions of the parameter q. This property results from the generalization of the following theorem.

Theorem 2.4 *Let $f \in C^1[0,1]$ and let the sequence (q_n) be chosen so that the sequence (ϵ_n) converges to zero from above faster than $(1/3^n)$, where*

$$\epsilon_n = \frac{n}{1 + q_n + q_n^2 + \cdots + q_n^{n-1}} - 1.$$

Then the sequence of derivatives of the generalized Bernstein polynomials, $B_n' f$, converges uniformly on $[0,1]$ to $f'(x)$.

Up to now the convergence of q-Bernstein polynomials is examined by taking a sequence $q = q_n$ such that $q_n \to 1$ as $n \to \infty$. In the recent developments, the convergence

of q-Bernstein polynomials is examined for fixed real q, $0 < q < 1$ and for $q \geq 1$. It is proved in Oruç and Tuncer [6] that for a fixed q, $0 < q < 1$, the uniform convergence holds if and only if f is linear on the interval $[0,1]$. Moreover, if $q \geq 1$, $B_n f \to f$ as $n \to \infty$ if f is a polynomial.

Theorem 2.5 *Let $q \geq 1$ be a fixed real number. Then, for any polynomial p,*

$$\lim_{n\to\infty} B_n(p;x) = p(x).$$

For any fixed integer i, the q-Bernstein polynomials of monomials (see Goodman *et.al.* [4]) can be written explicitly as

$$B_n(x^i;x) = \sum_{j=0}^{i} \lambda_j \, [n]^{j-i} S_q(i,j) x^j, \tag{2.1}$$

where

$$\lambda_j = \prod_{r=0}^{j-1} \left(1 - \frac{[r]}{[n]}\right),$$

an empty product denotes 1, and

$$S_q(i,j) = \frac{1}{[j]!\, q^{j(j-1)/2}} \sum_{r=0}^{j} (-1)^r q^{r(r-1)/2} \begin{bmatrix} j \\ r \end{bmatrix} [j-r]^i, \quad 0 \leq i \leq j, \tag{2.2}$$

is the Stirling polynomial of second kind. Thus for any polynomial p of degree m, one may write

$$B_n(p;x) = \mathbf{a}^T \mathbf{A} \mathbf{x}, \tag{2.3}$$

where $\mathbf{a}$ is the vector whose elements are the coefficients of p, $\mathbf{A}$ is an $(m+1) \times (m+1)$ lower triangular matrix with the elements

$$a_{i,j} = \begin{cases} \lambda_j [n]^{j-i} S_q(i,j), & 0 \leq j \leq i, \\ 0, & i < j, \end{cases} \tag{2.4}$$

and $\mathbf{x}$ is the vector whose elements form the standard basis for the space of polynomials P_m of degree m.

Lemma 2.1 *Let $0 < q < 1$ be a fixed real number. Then*

$$\lim_{n\to\infty} B_n(p;x) = p(x)$$

if and only if $p(x)$ is linear.

This lemma can be generalized for any function $f \in C[0,1]$.

Theorem 2.6 *Let $0 < q < 1$ be a fixed real number and $f \in C[0,1]$. Then*

$$\lim_{n\to\infty} B_n(f;x) = f(x)$$

if and only if $f(x)$ is linear.

3 The iterates

The iterates of classical Bernstein polynomials were first studied by Kelisky and Rivlin [5]. The authors proved that iterates of Bernstein polynomials converge to linear end point interpolants on $[0,1]$. Several generalization of the result due to Kelisky and Rivlin has been considered by many authors; see Sevy [9] and Wenz [10]. The recent result is the convergence of iterates of generalized Bernstein polynomials. It is proved in Oruç and Tuncer [6] that the q-Bernstein polynomials do preserve the convergence property of iterates of classical Bernstein polynomial. The iterates of generalized Bernstein polynomial are defined by

$$B_n^{M+1}(f;x) = B_n(B_n^M(f;x);x), \quad M = 1,2,\ldots, \tag{3.1}$$

where $B_n^1(f;x) = B_n(f;x)$.

Theorem 3.1 *Let $q \geq 0$ be a fixed real number. Then*

$$\lim_{M\to\infty} B_n^M(f;x) = f(0) + (f(1) - f(0))x. \tag{3.2}$$

Let $\mathcal{A}$ and B be operators then the Boolean sum of $\mathcal{A}$ and B is defined to be

$$\mathcal{A} \oplus B = \mathcal{A} + B - \mathcal{A} \circ B.$$

We will be concerned with iterated Boolean sums of the generalized Bernstein polynomials in the form $B_n \oplus B_n \oplus \cdots \oplus B_n$ and will denote such an M-fold Boolean sum of the generalized Bernstein operators by $\oplus^M B_n$. Sevy [9] and Wenz [10] proved that the limit of iterated Boolean sums of Bernstein polynomials is the interpolation polynomial with respect to the nodes $(\frac{i}{n}, f(\frac{i}{n}))$ $i = 0,\ldots,n$ as $M \to \infty$. The second theorem of this section will give a result for the convergence of iterates of Boolean sums of generalized Bernstein polynomials. It is proved in Oruç and Tuncer [6] that the iterates of Boolean sums of q-Bernstein polynomials converge to the interpolating polynomial at the nodes $\left(\frac{[i]}{[n]}, f\left(\frac{[i]}{[n]}\right)\right)$.

Theorem 3.2 *The iterated Boolean sum of the q-Bernstein operator $\oplus^M B_n(f;x)$ associated with the function $f(x) \in C[0,1]$ converges to the interpolating polynomial $L_n f$ of degree n of $f(x)$ at the points $x_i = [i]/[n]$, $i = 0,1,\ldots,n$.*

4 A difference operator $\mathcal{D}_q$ on generalized Bernstein polynomials

Given any function $f(x)$ and $q \in R$ we define the operator $\mathcal{D}_q$

$$\mathcal{D}_q f(x) = \frac{f(qx) - f(x)}{qx - x}. \tag{4.1}$$

Thus $\mathcal{D}_q f(x)$ is simply a divided difference, $\mathcal{D}_q f(x) = f[x, qx]$. Note that, for a function f and non-negative integer k

$$f[x, qx, \ldots, q^k x] = \frac{1}{[k]!}\mathcal{D}_q^k f(x).$$

Theorem 4.1 *For any integer* $0 \leq k \leq n$,

$$\mathcal{D}_q^k B_n(f;x) = [n]\cdots[n-k+1]\sum_{r=0}^{n-k}\Delta^k f_r \begin{bmatrix} n-k \\ r \end{bmatrix} x^r \prod_{s=k}^{n-r-1}(1-q^s x).$$

Proof: Recall the q-difference form of generalized Bernstein polynomials (1.5) and apply the operator $\mathcal{D}_q$ to $B_n(f;x)$ repeatedly k times to get,

$$\mathcal{D}_q^k B_n(f;x) = \sum_{r=0}^{n-k}\frac{[n]!}{[n-k-r]![r]!}\Delta^{k+r} f_0 x^r. \tag{4.2}$$

It will be useful to express Δ^{k+r} in terms of Δ^k. One may prove by induction on m that, for $0 \leq m \leq n-k$ we may write

$$\Delta^{m+k} f_i = \sum_{t=0}^{m}(-1)^t q^{t(t+2t-1)/2}\begin{bmatrix} m \\ t \end{bmatrix}\Delta^k f_{m+i-t}.$$

Now applying the latter identity to (4.2) gives

$$\mathcal{D}_q^k B_n(f;x) = \sum_{r=0}^{n-k}\sum_{t=0}^{r}(-1)^t q^{t(t+2k-1)/2}\frac{[n]!}{[n-k-r]![r]!}\begin{bmatrix} r \\ t \end{bmatrix}\Delta^k f_{r-t} x^r. \tag{4.3}$$

Writing $m = r - t$

$$\frac{[n]!}{[n-k-m-t]![m+t]!}\begin{bmatrix} m+t \\ t \end{bmatrix} = \frac{[n]!}{[n-k-m]![m]!}\begin{bmatrix} n-k-m \\ t \end{bmatrix} \tag{4.4}$$

and putting (4.4) in (4.3) we obtain

$$\mathcal{D}_q^k B_n(f;x) = \sum_{m=0}^{n-k}\frac{[n]!}{[n-k-m]![m]!}\Delta^k f_m x^m \sum_{t=0}^{n-k-m}(-1)^t q^{t(t+2k-1)/2}\begin{bmatrix} n-k-m \\ t \end{bmatrix} x^t.$$

Now, it can be easily derived from generalized binomial expansion (1.3), on replacing x by $q^k x$, that

$$\prod_{t=k}^{n-m-1}(1-q^t x) = \sum_{t=0}^{n-k-m}(-1)^t q^{t(t+2k-1)/2}\begin{bmatrix} n-k-m \\ t \end{bmatrix} x^t.$$

This completes the proof. □

From Theorem 4.1 we see that, with $0 < q \leq 1$, if $\Delta^k f_r \geq 0$ for $0 \leq r \leq n-k$ then $\mathcal{D}_q^k B_n(f;x) \geq 0$. If f is convex on $0 \leq x \leq 1$ then $\mathcal{D}_q^2 B_n(f;x) \geq 0$ for $0 < q \leq 1$. If f is increasing then $\mathcal{D}_q B_n(f;x) \geq 0$, for $0 < q \leq 1$.

Acknowledgment: The second author is supported from the Institute of Natural and Applied Sciences of D.E.U. and this research is partially supported by the grant AFS 0922.20.01.02.

Bibliography

1. G. E. Andrews, *The Theory of Partitions,* Cambridge University Press, Cambridge, 1998.
2. E. W. Cheney, *Introduction to Approximation Theory,* AMS Chelsea, Providence, 1981.
3. P. J. Davis, *Interpolation and Approximation,* Dover Publications, New York, 1975.
4. T. N. T. Goodman, H. Oruç, and G. M. Phillips, Convexity and generalized Bernstein polynomials, *Proc. Edin. Math. Soc.* **42** (1999) 179–190.
5. R. P Kelisky and T. J. Rivlin, Iterates of Bernstein polynomials, *Pacific J. Math.* **21** (1967), 511–520.
6. H. Oruç and N. Tuncer, On the convergence and iterates of q-Bernstein polynomials, *J. Approx. Theory,* to appear.
7. G. M. Phillips On generalized Bernstein polynomials, *Numerical Analysis,* D. Griffiths and G. Watson eds. (1996), 263–269.
8. G. M. Phillips, Bernstein polynomials based on the q-integers, The heritage of P. L. Chebyshev: a Festschrift in honor of the 70th birthday of T. J. Rivlin. *Ann. Numer. Math.* **4** (1997), 511–518.
9. J. C. Sevy, Lagrange and least-square polynomials as limits of linear combinations of iterates of Bernstein and Durrmeyer polynomials, *J. Approx. Theory* **80** (1995), 267–271.
10. H. J. Wenz, On the limits of (Linear combinations of) iterates of linear operators, *J. Approx. Theory* **89** (1997), 219–237.

Uniform Powell–Sabin splines for the polygonal hole problem

Joris Windmolders and Paul Dierckx

Department of Computer Sciences, Kath. University Leuven, Belgium.
Joris.Windmolders@cs.kuleuven.ac.be, Paul.Dierckx@cs.kuleuven.ac.be

Abstract

An algorithm is described for smoothly filling in a polygonal hole in a surface, with a parametric uniform Powell–Sabin spline surface patch. It uses interpolation and subdivision techniques for iteratively determining an approximating solution. No assumptions are made about the surrounding surface. The user has to provide routines for calculating the curve points and the unit surface normal along the edge, as well as the unit tangent vector of the edge curves, parametrized on the unit interval.

1 Introduction

A classical problem in CAGD is to fill in a hole, bounded by a set of surfaces. This problem has already been addressed in the literature (e.g. [1, 2, 4]). In most cases, assumptions are made on the bounding surfaces. In this paper, we present an algorithm for filling in a 3, 4, 5 or 6–sided hole that makes no assumptions on the surrounding surfaces, and therefore it is generally applicable. On the other hand, the filling patch will meet the given boundary curves approximately. The input of our algorithm (see Figure 1) consists of the boundary curves $\mathbf{p}$ which join at their endpoints. Furthermore, the user should provide the unit tangent vector $\vec{\gamma}$ to the boundary curves at any point, and the unit normal vector $\vec{n}$ to the surrounding surface at any curve point except the endpoints, where the tangent vectors of the joining curves are needed only (see Figure 1 again). For other (interior) curve points, our algorithm will calculate a unit vector $\vec{\delta} = \vec{n} \times \vec{\gamma}$, which will be called the (unit) cross–boundary tangent vector. It shall be referred to as if it were provided by the user. We will calculate a filling surface patch that interpolates the user supplied boundary curves and has the same surface normal in a number of points. This will leave us some degrees of freedom, which we will use to fit the curve and the cross–boundary tangent vector in between each pair of interpolation points. In section 2 we briefly recall the basic properties of uniform Powell–Sabin splines. Section 3 explains how we can benefit from these properties to use UPS–splines for the polygonal hole problem. Section 4 explains our algorithm in detail. Finally we remark that on the pictures, we will denote 2D and 3D entities interchangebly; therefore most pictures reflect the situation only schematically.

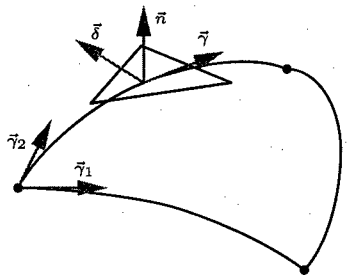


FIG. 1. User supplied data.

2 Uniform Powell-Sabin splines

This section recalls the main properties of Uniform Powell-Sabin splines. For details, we refer to the original papers [3, 5].

By $S_2^1(\Delta^*)$ we denote the linear space of uniform Powell–Sabin splines (in the sequel called UPS–splines), i.e., piecewise quadratic polynomials on a uniform triangulation Δ (which means that all triangles are equilateral and have the same size) of a polygon Ω, where Δ^* is a PS–refinement of Δ. The boundary of Ω will be called $\delta\Omega$, whereas the boundary of the triangulation will be referred to as $\delta\Delta$. The vertices of Δ are denoted $V_i, i = 1, \ldots, n$, and its triangles are $\rho_i, i = 1, \ldots, m$. These splines have global C^1–continuity on Δ^*. Any $\mathbf{s}(\mathbf{u}, \mathbf{v})$ has a unique B–spline representation

$$\mathbf{s}(\mathbf{u}, \mathbf{v}) = \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{3}} \mathbf{c_{i,j}} \mathbf{B_i^j}(\mathbf{u}, \mathbf{v}), \quad (\mathbf{u}, \mathbf{v}) \in \boldsymbol{\Omega}, \tag{2.1}$$

where the locally supported basis functions form a convex partition of unity and $\mathbf{c_{i,j}} \in \mathbf{R^3}$ are the control points. It follows that $\mathbf{s}(\mathbf{u}, \mathbf{v})$ belongs to the convex hull of $\{\mathbf{c_{i,j}}\}_{i,j}$. Furthermore, one can prove that the control triangles, being defined as $T_i(\mathbf{c_{i,1}}, \mathbf{c_{i,2}}, \mathbf{c_{i,3}})$, $i = 1, \ldots, n$, are tangent to the surface at $\mathbf{s}(\mathbf{V_i})$. Due to the local support of B_i^j, a change to $\mathbf{c_{i,j}}$ will only affect $\mathbf{s}(\mathbf{u}, \mathbf{v})|_{\mathbf{M_i}}$, i.e., the restriction of $\mathbf{s}(\mathbf{u}, \mathbf{v})$ to the molecule of V_i, being the set of triangles ρ_j that have V_i as a vertex. This indicates that we have a useful representation for C^1–continuous surfaces, without being restricted to a rectangular domain, and still enjoying the interesting features of the classical B–spline representation for tensor product splines.

2.1 Subdivision

In [5] we present a subdivision scheme for UPS–splines. Let Δ_r be a uniform refinement of Δ, obtained by midedge subdivision. For a given $\mathbf{s}(\mathbf{u}, \mathbf{v})$ on Δ, the representation (2.1) on Δ_r can be calculated using convex barycentric combinations of the control points only. First, a new control triangle along each edge V_iV_j is calculated as illustrated in

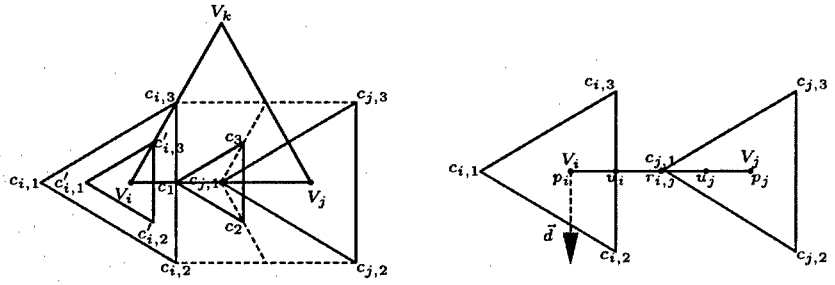


FIG. 2. Subdivision and Bézier points.

Figure 2, left, for the bottom edge of a triangle $\rho_l(V_i, V_j, V_k) \in \Delta$:

$$\begin{cases} \mathbf{c_1} &= \frac{1}{2}(\mathbf{c_{i,2}} + \mathbf{c_{i,3}}) \\ \mathbf{c_2} &= \frac{1}{2}\mathbf{c_{j,1}} + \frac{1}{4}(\mathbf{c_{1,2}} + \mathbf{c_{j,2}}) \\ \mathbf{c_3} &= \frac{1}{2}\mathbf{c_{j,1}} + \frac{1}{4}(\mathbf{c_{i,3}} + \mathbf{c_{j,3}}). \end{cases} \tag{2.2}$$

Next, the control triangles at the original vertices are rescaled: for example,

$$\begin{cases} \mathbf{c'_{i,1}} &= \frac{2}{3}\mathbf{c_{i,1}} + \frac{1}{6}(\mathbf{c_{i,2}} + \mathbf{c_{i,3}}) \\ \mathbf{c'_{i,2}} &= \frac{2}{3}\mathbf{c_{i,2}} + \frac{1}{6}(\mathbf{c_{i,3}} + \mathbf{c_{i,1}}) \\ \mathbf{c'_{i,3}} &= \frac{2}{3}\mathbf{c_{i,3}} + \frac{1}{6}(\mathbf{c_{i,1}} + \mathbf{c_{i,2}}). \end{cases} \tag{2.3}$$

They are still tangent to the surface at their barycenter, but their area is only a quarter that of the former control triangles. Therefore they connect tighter to the surface.

2.2 The piecewise Bézier representation

Another important property of the B–spline representation for UPS–splines, is that the piecewise Bézier representation can be calculated from (2.1) using simple convex barycentric combinations of the control points. In particular, focus an edge V_iV_j of Δ (see Figure 2, right). The Bézier points of the edge curve can be found from:

$$\mathbf{s(V_i)} = \mathbf{p_i} = \frac{1}{3}(\mathbf{c_{i,1}} + \mathbf{c_{i,2}} + \mathbf{c_{i,3}}), \quad \mathbf{s(V_j)} = \mathbf{p_j} = \frac{1}{3}(\mathbf{c_{j,1}} + \mathbf{c_{j,2}} + \mathbf{c_{j,3}}), \tag{2.4}$$

$$\mathbf{u_i} = \frac{1}{2}(\mathbf{c_{i,2}} + \mathbf{c_{i,3}}), \quad \mathbf{u_j} = \frac{2}{3}\mathbf{c_{j,1}} + \frac{1}{6}(\mathbf{c_{j,2}} + \mathbf{c_{j,3}}), \quad \mathbf{r_{i,j}} = \frac{1}{2}(\mathbf{u_i} + \mathbf{u_j}). \tag{2.5}$$

This is a piecewise quadratic Bézier curve, which means that $\mathbf{p_i}$, $\mathbf{r_{i,j}}$ and $\mathbf{p_j}$ are surface points, and that $\mathbf{u_i} - \mathbf{p_i}$ and $\mathbf{p_j} - \mathbf{u_j}$ are tangent to the surace at $\mathbf{p_i}$, resp. $\mathbf{p_j}$. Assuming a (counterclockwise) ordering of the boundary vertices $V_i \in \delta\Delta$, the edge curve from $\mathbf{s(V_i)}$ to the next adjacent point $\mathbf{s(V_j)}$ will be denoted $\mathbf{e_i(u, v)}$.

3 Application to the polygonal hole problem

Recall that our goal is to calculate a UPS–spline filling a hole in a surface, given by a set of bounding curves (denoted $\mathbf{p}$), their derivatives $\vec{\gamma}$ and the cross–boundary tangent vectors $\vec{\delta}$. The UPS–patch will fit these curves approximately along its boundary. In the first place, interpolation of the given data at the vertices $V_i \in \delta\Delta$ is achieved. This leaves

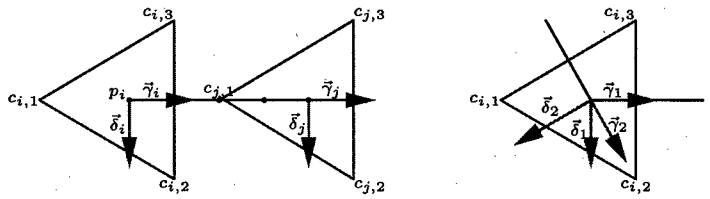


FIG. 3. Tangent and cross-boundary tangent vectors.

some degrees of freedom allowing to fit the given curves. In the sequel we shall denote the user supplied data, evaluated at V_i, by $(\mathbf{p_i}, \vec{\gamma}_\mathbf{i}, \vec{\delta}_\mathbf{i})$.

3.1 Interpolating UPS–splines and degrees of freedom

In order to obtain interpolation we determine a control triangle T_i in the tangent plane spanned by $\mathbf{p_i} + \epsilon\vec{\gamma}_\mathbf{i} + \nu\vec{\delta}_\mathbf{i}$, $\epsilon, \nu \in \mathbf{R}$, such that $\mathbf{s}(\mathbf{V_i}) = \mathbf{p_i}$. Curve point interpolation is simply expressed by (2.4). Furthermore, we let the tangent to $\mathbf{e_i}$ at V_i be parallel to $\vec{\gamma}_i$:

$$\mathbf{u_i} - \mathbf{p_i} = \frac{\mathbf{1}}{\mathbf{6}}(\mathbf{c_{i,2}} + \mathbf{c_{i,3}}) - \frac{\mathbf{1}}{\mathbf{3}}\mathbf{c_{i,1}} = \alpha_\mathbf{i}\vec{\gamma}_\mathbf{i}, \tag{3.1}$$

where α_i is a scaling factor. Next, we need the cross–boundary tangent vector of $\mathbf{s}(\mathbf{u}, \mathbf{v})$ at V_i to be parallel to $\vec{\delta}_i$. Mapping the cross–boundary vector $\vec{d}$ in the domain plane (see Figure 2, right) onto the control triangle yields a vector parallel with $\mathbf{c_{i,2}} - \mathbf{c_{i,3}}$:

$$\mathbf{c_{i,2}} - \mathbf{c_{i,3}} = \mathbf{2}\beta_\mathbf{i}\vec{\delta}_\mathbf{i}, \tag{3.2}$$

where β_i is again a scaling factor.

Solving (2.4), (3.1) and (3.2) to $\mathbf{c_{i,j}}$ in terms of the unknown α_i and β_i (further called the α– and β–factors) yields

$$\begin{cases} \mathbf{c_{i,1}} &= \mathbf{p_i} - \alpha_\mathbf{i}\vec{\gamma}_\mathbf{i} \\ \mathbf{c_{i,2}} &= \mathbf{p_i} + \frac{\alpha_\mathbf{i}}{\mathbf{2}}\vec{\gamma}_\mathbf{i} + \beta_\mathbf{i}\vec{\delta}_\mathbf{i} \\ \mathbf{c_{i,3}} &= \mathbf{p_i} + \frac{\alpha_\mathbf{i}}{\mathbf{2}}\vec{\gamma}_\mathbf{i} - \beta_\mathbf{i}\vec{\delta}_\mathbf{i}. \end{cases} \tag{3.3}$$

These equations ensure that $\mathbf{s}(\mathbf{u}, \mathbf{v})$ interpolates the given data at $V_i \in \delta\Delta$, and leaves us two degrees of freedom per vertex (α_i and β_i). These scaling factors are related to the size of the control triangle. For example, subdivision by (2.3) divides α_i and β_i by a factor of 2.

3.2 The fitting equations

We will now use these degrees of freedom to fit the user supplied data, in between each pair of adjacent interpolating vertices $V_i, V_j \in \delta\Delta$. First, the α–factors at V_i and V_j are determined by trying to interpolate the curve $\mathbf{p}$ at the edge midpoint $V_{i,j} = \frac{1}{2}(V_i + V_j)$. From Section 2.2, the interpolation condition reads $\mathbf{r_{i,j}} = \frac{\mathbf{1}}{\mathbf{2}}(\mathbf{u_i} + \mathbf{u_j}) = \mathbf{p_{i,j}}$, where $\mathbf{p_{i,j}}$ is the given curve point. Taking (2.5) and (3.3) into account, we have

$$\alpha_i\vec{\gamma}_i - \alpha_j\vec{\gamma}_j = 4\mathbf{p_{i,j}} - \mathbf{2}(\mathbf{p_i} + \mathbf{p_j}) = \mathbf{q_{i,j}}. \tag{3.4}$$

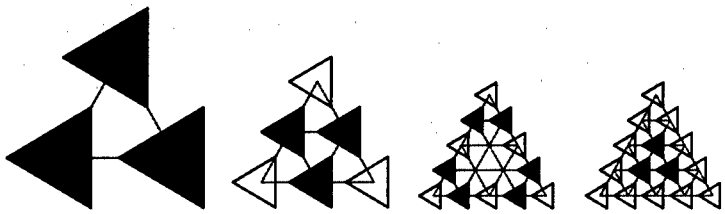


FIG. 4. Consecutive iteration steps.

This is a system of 3 equations with (at most) 2 unknowns. It can be solved in the least squares sense.

Next, the β–factors at V_i and V_j are obtained by fitting the cross–boundary tangent vector at $V_{i,j}$. First, we derive a subdivision rule for the β–factors at the vertices of Δ from (2.2) and (3.2):

$$\beta'_{i,j}\vec{\delta}'_{i,j} = \frac{1}{4}(\beta_i\vec{\delta}_i + \beta_j\vec{\delta}_j), \tag{3.5}$$

where $\vec{\delta}'_{i,j}$ is the cross–boundary tangent vector to $\mathbf{s}(\mathbf{u},\mathbf{v})$ at $V_{i,j}$. This $\beta'_{i,j}$–factor belongs to a finer subdivision level then β_i and β_j, so we have to scale it up by a factor of 2. The interpolation condition then is

$$\beta_{i,j}\vec{\delta}_{i,j} = \frac{1}{2}(\beta_i\vec{\delta}_i + \beta_j\vec{\delta}_j). \tag{3.6}$$

Note that $\vec{\delta}_{i,j}$ has been used instead of $\vec{\delta}'_{i,j}$. This is again an overdetermined system which can be solved in the least squares sense.

4 The algorithm

We will restrict the figures illustrating the algorithm to the case of a triangular hole, although the algorithm is immediately applicable to cases with 4, 5 and 6 boundary curves as well (see Section 4.4).

The idea is to calculate, during a pre–iteration step, an initial solution which is smooth, but in general not close enough, and to refine this approximation iteratively to obtain a better fit to the given curves until a certain stopping criterion is satisfied. Finally, during a post–iteration step, the interior control triangles are calculated, actually filling the hole. Figure 4 illustrates this: imagine a pre–iteration step, two refinement steps and a post–iteration step. The control triangles added during a particular step have been shaded.

4.1 An initial solution

The initial solution (Figure 4, leftmost) is easily obtained by solving (3.4) in the least squares sense for each edge V_iV_j. If we assume that $\vec{\gamma}_i \neq \vec{\gamma}_j$, then

$$\alpha_i = \frac{1}{D}\left((\vec{\gamma}_i \cdot \mathbf{q_{i,j}}) - (\vec{\gamma}_\mathbf{j} \cdot \mathbf{q_{i,j}})(\vec{\gamma}_\mathbf{i} \cdot \vec{\gamma}_\mathbf{j})\right), \tag{4.1}$$

$$\alpha_j \;=\; \frac{1}{D}\left(-(\vec{\gamma}_j \cdot \mathbf{q_{i,j}}) + (\vec{\gamma}_\mathbf{i} \cdot \mathbf{q_{i,j}})(\vec{\gamma}_\mathbf{i} \cdot \vec{\gamma}_\mathbf{j})\right), \tag{4.2}$$

where $D = 1-(\vec{\gamma}_i \cdot \vec{\gamma}_j)^2$. This yields two α–factors per vertex: one for each boundary edge being incident to that vertex. Therefore, T_i is completely determined. The β–factors can be calculated by writing (3.3) for both edges incident with the vertex and eliminating $\mathbf{c_2}$, respectively $\mathbf{c_1}$, e.g., for Figure 3, right,

$$\beta_1 = \alpha_2(\vec{\gamma}_2 \cdot \vec{\delta}_1), \quad \beta_2 = -\alpha_1(\vec{\gamma}_1 \cdot \vec{\delta}_2). \tag{4.3}$$

There exist pathological cases where $\vec{\gamma}_2 \perp \vec{\delta}_1$ or $\vec{\gamma}_1 \perp \vec{\delta}_2$. Our algorithm then sets $\beta_1 = \alpha_1$, resp. $\beta_2 = \alpha_2$. For the case $\vec{\gamma}_i = \vec{\gamma}_j$, (3.4) has no solution in the least–squares sense. Assuming that $\mathbf{s_i}$ is a straight line from $\mathbf{s(V_i)}$ to $\mathbf{s(V_j)}$, the α–factors can then be determined from the projection onto the domain plane, where the size of the so–called PS–triangles (the projections of the control triangles) is fixed. The reader can verify that this yields $\alpha_i = \alpha_j = \frac{1}{2}|V_iV_j|$.

4.2 The iteration step

First the control triangles from the previous steps are rescaled by subdivision. This is simply done by scaling down the α– and β–factors: $\alpha_i \leftarrow \frac{\alpha_i}{2}$ and $\beta_i \leftarrow \frac{\beta_i}{2}$, for each $V_i \in \delta\Delta$. Next, a new control triangle is created in between any two adjacent vertices at the coarser level. This situation is illustrated in Figure 5, left, where the darker triangles are known. We are looking for the α–and β–factors for the middle control polygon, which is tangent to the surface at $\mathbf{s(V_k)}$, $V_k = \frac{1}{2}(V_i + V_j)$. Consider the α–factor first. In order to obtain a better fit, we try to interpolate $\mathbf{p}$ at $V_{i,k} = \frac{1}{2}(V_i + V_k)$ and $V_{k,j} = \frac{1}{2}(V_k + V_j)$. This yields a set of fitting equations

$$\begin{cases} \alpha_i\vec{\gamma}_i - \alpha_k\vec{\gamma}_k &= \mathbf{q_{i,k}}, \\ \alpha_k\vec{\gamma}_k - \alpha_j\vec{\gamma}_j &= \mathbf{q_{k,j}}, \end{cases} \tag{4.4}$$

where α_i and α_j are known. Thus, α_k can be obtained as the least-squares solution of (4.4):

$$\alpha_k = \frac{1}{2}(\vec{\gamma}_k \cdot (\alpha_i\vec{\gamma}_i - \mathbf{q_{i,k}} + \mathbf{q_{k,j}} - \alpha_\mathbf{j}\vec{\gamma}_\mathbf{j})). \tag{4.5}$$

The β_k–factor is found by fitting the cross–boundary vectors at $V_{i,k}$ and $V_{k,j}$, i.e., by solving the following system in the least–squares sense:

$$\begin{cases} \beta_{i,k}\vec{\delta}_{i,k} &= \frac{1}{2}(\beta_i\vec{\delta}_i + \beta_k\vec{\delta}_k), \\ \beta_{k,j}\vec{\delta}_{k,j} &= \frac{1}{2}(\beta_k\vec{\delta}_k + \beta_j\vec{\delta}_j), \end{cases} \tag{4.6}$$

where β_i and β_j are known. If $\vec{\delta}_{i,k} = \vec{\delta}_k = \vec{\delta}_{k,j}$, as is always the case for a planar curve, this system has no solution in the least–squares sense. The β_k factor can then easily be obtained by equation (3.6), i.e., by subdivision and upscaling.

4.3 The interior control points

Finally, as soon as the user supplied edge curves have been approximated well enough, the interior control points at the eventual refinement level have to be calculated. We will

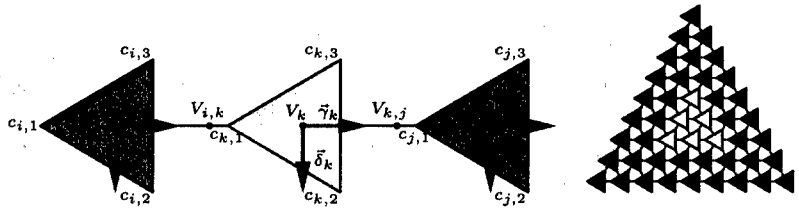


FIG. 5. The refinement and post-iteration steps.

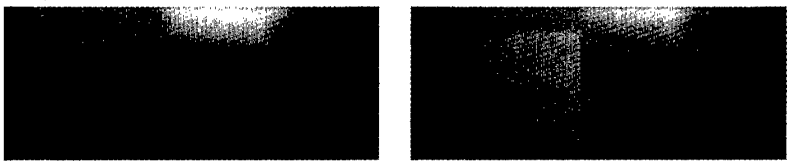


FIG. 6. The hole and the triangular patches.

discuss three possibilities by the help of an example; Figure 6 shows a hole (left) and two filling patches (right).

Copy From Initial. The interior control points are obtained directly from the initial solution by subdivision. This guarantees that the interior of the patch is smooth. A disadvantage is that the inner of the first approximation in general has no connection with the shape of the edge curves. This can cause unwanted artefacts near the boundary, after a few iterations (see Figure 7, left). The next option will therefore take edge features into account.

Averaging. We will fill the hole gradually by calculating a ring of control triangles during each pass, going from the edge towards the inner of the patch. Figure 5, right shows an example where each ring has a different shade of grey. At each step, a control triangle of the current ring is obtained by averaging six surrounding control triangles. These come from the initial solution, or, if possible, from a previously calculated ring. Edge features are now smoothed out towards the inner of the patch. However, there is a main disadvantage to this approach, if averaging is applied after the last iteration step: the unwanted artefacts mentioned before are now repeated for every ring, smoothed out towards the inner of the surface, as shown on Figure 7, middle.

Instant Update. A good compromise would be to take edge features into account before we finish iterating. This can be accomplished by subdividing the initial solution at each refinement step, but, we always overwrite its edge with the most recent boundary approximation. The results of this strategy are depicted in Figure 7, right.

In any case can the user change the interior control triangles, and still he has a C^1–continuous filling patch, fitting the specified edge curves with demanded precision.

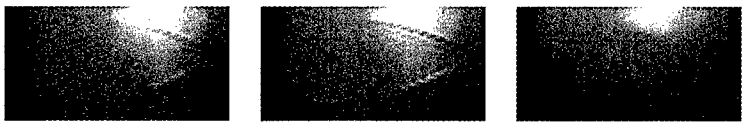


FIG. 7. Copy from initial solution and averaging (4 iterations); instant update (3 iterations).

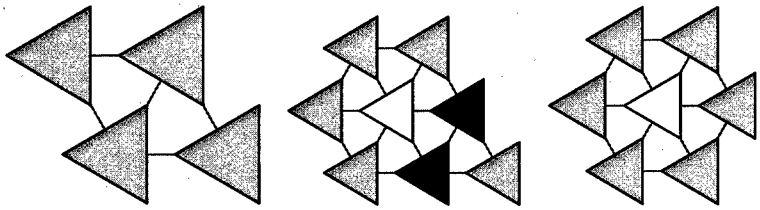


FIG. 8. Cases with 4, 5 and 6 boundary curves.

4.4 A note on the number of edges

The algorithm sketched in Section 4 is immediately applicable to problems with 4, 5 and 6 boundary curves as well. Figure 8 shows the configuration of the initial solution for each of these cases. If we are working with 5 edges, there are 2 edges having a control triangle at its midpoint (shaded darker). This requires a tiny modification to the calculation of the initial solution for those edges. The α–factors are obtained by solving (4.4) to the unknown α_i, α_j and α_k. The β–factors of the outer control poygons are obtained as usual; for the middle polygon one can apply (3.6). Also, for the cases of 5 and 6 boundary curves, an interior control triangle (unshaded) has to be calculated for the initial solution. This can be done by averaging the six surrounding control polygons.

Bibliography

1. Charrot, P. and A. Gregory, A pentagonal surface patch for computer aided geometric design, Computer Aided Design 1, pp 87–94.
2. Chui, C. K. and M.–J. Lai (2000), Filling polygonal holes using C^1 cubic triangular spline patches, Computer Aided Geometric Design 17, pp 297–307.
3. Dierckx, P. (1997), On calculating normalized Powell–Sabin B–splines, Computer Aided Geometric Design 15, pp 61–78.
4. Gregory, J.A., V. K. H. Lau, and J. M. Hahn (1993) , High order continuous polygonal patches, in Geometric Modelling, G. Farin, H. Hagen and H. Noltemeier (eds.), Springer–Verlag Wien.
5. Windmolders, J., Dierckx, P. (1999), Subdivision of Uniform Powell–Sabin splines, Computer Aided Geometric Design 16, 301-315.
6. Windmolders, J. and P. Dierckx, NURPS for Special Effects and Quadrics: Oslo 2000, Tom Lyche and L. L. Schumaker (eds.), Vanderbilt Press, Nashville 2001.

Chapter 2

Differential Equations

Iterative refinement schemes for an ill-conditioned transfer equation in astrophysics

Mario Ahues, Filomena d'Almeida, Alain Largillier, Olivier Titaud and Paulo Vasconcelos

Université de Saint Etienne, France and Universidade do Porto, Portugal

Abstract

Let $X := L^1([0,\tau_0])$, where τ_0 represents the optical depth of a stellar atmosphere. The weakly singular integral operator $T : X \to X$ defined by

$$(T\varphi)(\tau) = \tfrac{\varpi}{2} \int_0^{\tau_0} E_1(|\tau - \tau'|)\varphi(\tau')\, d\tau',$$

where $\varpi \in]0,1[$ is the albedo of the atmosphere and E_1 denotes the first exponential-integral function, is such that $\|T\|_1 = \varpi(1 - E_2(\tau_0/2))$, where E_2 denotes the second exponential-integral function. If ϖ is close to 1, and τ_0 is large, then $\|T\|_1$ is close to 1. In that case, the *transfer problem*

given $f \in X$, find $\varphi \in X$ such that $T\varphi = \varphi + f$

is ill-conditioned, and the convergence of the fixed-point iteration $\varphi_{k+1} = T\varphi_k - f$, which is commonly used by numerical astronomers, becomes prohibitively slow. The purposes of this work are to approximate φ through different sequences whose terms solve well-conditioned approximate equations, and to compare their efficiency and computational costs.

1 Introduction

For a given $\tau_0 > 0$, let g be a function defined on $]0,\tau_0]$ such that

$$\lim_{\tau \to 0^+} g(\tau) = +\infty, \tag{1.1}$$

$$g \in C^0(]0,\tau_0]) \cap L^1([0,\tau_0]), \tag{1.2}$$

$$g(\tau) \geq 0 \text{ for all } \tau \in]0,\tau_0], \tag{1.3}$$

$$g \text{ is a decreasing function on }]0,\tau_0]. \tag{1.4}$$

We consider the integral operator T defined by

$$(Tx)(\tau) := \int_0^{\tau_0} g(|\tau - \tau'|)x(\tau')\, d\tau'. \tag{1.5}$$

Theorem 1 *T is a linear compact operator in $L^1([0,\tau_0])$ and $\|T\|_1 = 2\int_0^{\tau_0/2} g(\tau)\, d\tau$.*

Proof: See [2]. □

For z in the resolvent set of T, we consider the Fredholm equation of the second kind

$$T\varphi = z\varphi + f. \tag{1.6}$$

Applications will concern the function $g :]0, \tau_0] \to \mathbb{R}$ given by

$$g(\tau) := \frac{\varpi}{2} E_1(\tau) \tag{1.7}$$

where $\varpi \in]0, 1[$ and E_1 is the exponential-integral function : $E_1(\tau) := \displaystyle\int_1^\infty \frac{\exp(-\tau\mu)}{\mu}\, d\mu$, $\tau > 0$. E_1 is the first function of the sequence $(E_\nu)_{\nu \geq 1}$, $E_\nu(\tau) := \displaystyle\int_1^\infty \frac{\exp(-\tau\mu)}{\mu^\nu}\, d\mu$, $\tau \geq 0$, $\nu \geq 2$, and it is the only one presenting a logarithmic singularity at $\tau = 0$. Following Theorem 1, when g is defined by (1.7), we have $\|T\|_1 = \varpi[1 - E_2(\tau_0/2)] < 1$.

We recall that a bounded linear finite rank operator T_n in a normed linear space X can be written as

$$T_n := \sum_{j=1}^{n} \langle \cdot\,, \ell_{n,j} \rangle e_{n,j} \tag{1.8}$$

where $n \in \mathbb{N}^*$, and, for $j \in [\![1, n]\!]$, $\ell_{n,j} \in X^*$, the topological adjoint space of X, and $e_{n,j} \in X$.

The resolution of the approximate equation

$$T_n\varphi_n = z\varphi_n + f, \tag{1.9}$$

where z belongs to the resolvent set of T_n, leads to an n-dimensional linear system

$$(\mathsf{A}_n - z\mathsf{I}_n)\mathsf{x}_n = \mathsf{b}_n \tag{1.10}$$

where I_n is the identity matrix of order n,

$$\mathsf{A}_n(i,j) := \langle e_{n,j}\,, \ell_{n,i} \rangle, \quad \mathsf{b}_n(i) := \langle f\,, \ell_{n,i} \rangle, \quad \mathsf{x}_n(j) := \langle \varphi_n\,, \ell_{n,j} \rangle. \tag{1.11}$$

Once this system is solved, the solution of (1.9) is given by

$$\varphi_n = \frac{1}{z} \left(\sum_{j=1}^{n} \mathsf{x}_n(j) e_{n,j} - f \right). \tag{1.12}$$

We are interested in refining approximations obtained with $T_n := \pi_n T$, where π_n is a sequence of projections with finite rank n. A bounded projection π_n of finite rank n is defined by $\pi_n x := \sum_{j=1}^{n} \langle x\,, e^*_{n,j} \rangle e_{n,j}$ for all $x \in X$, where $(e_{n,j})_{j=1}^n$ is an ordered basis of the range of π_n, and $(e^*_{n,j})_{j=1}^n$ is an adjoint basis of the former in X^*. Hence

$$T_n x := \sum_{j=1}^{n} \langle Tx\,, e^*_{n,j} \rangle e_{n,j}, \quad x \in X. \tag{1.13}$$

We suppose that π_n is pointwise convergent to the identity operator in the Banach X where the operator T is defined. Since T is compact, T_n converges to T in the operator

norm. Let $R(z) := (T - zI)^{-1}$ be the resolvent of T at z. Then $R_n(z) := (T_n - zI)^{-1}$ exists for n large enough and is uniformly bounded, that is, there exists n_0 such that

$$c_0(z) := \sup_{n>n_0} \|R_n(z)\| < +\infty. \tag{1.14}$$

We develop an application in the space $X := L^1([0,\tau_0])$. Let $(\tau_{n,j})_{j=0}^n$ be a grid on $[0,\tau_0]$ such that

$$0 =: \tau_{n,0} < \tau_{n,1} < \cdots < \tau_{n,n-1} < \tau_{n,n} := \tau_0, \tag{1.15}$$

and set

$$h_{n,j} := \tau_{n,j} - \tau_{n,j-1} \quad \text{for } j \in [1,\ldots,n]. \tag{1.16}$$

We define, for $\tau \in [0,\tau_0]$,

$$e_{n,j}(\tau) := \begin{cases} 1 & \text{if } \tau \in (\tau_{n,j-1}, \tau_{n,j}) \\ 0 & \text{otherwise} \end{cases} \tag{1.17}$$

and, for $x \in L^1([0,\tau_0])$,

$$\langle x\,,\, e^*_{n,j}\rangle := \frac{1}{h_{n,j}} \int_{\tau_{n,j-1}}^{\tau_{n,j}} x(\tau')\,d\tau'. \tag{1.18}$$

The product defined in (1.18) is a special case of the scalar product used in equation (1.8) when a grid such as (1.15) is set. In this case the operator in (1.13) is the operator in (1.8) if we choose $\ell_{n,j} = T^* e^*_{n,j}$. Let

$$\mu_n := \min\{h_{n,j} \,:\, j \in [1,\ldots,n]\}, \quad h_n := \max\{h_{n,j} \,:\, j \in [1,\ldots,n]\}, \quad q_n := \frac{\mu_n}{h_n}. \tag{1.19}$$

For quasi-uniform grids, there exists a constant q independent of n such that, for all n, $q \le q_n$. For uniform grids, $q_n = 1$ for all n.

Theorem 2 *Let $\varphi \neq 0$ be the solution of (1.6) with T defined by (1.5). Let φ_n be the solution of (1.9) with T_n defined by (1.8) and (1.15)–(1.17). Then, for n large enough,*

$$\frac{\|\varphi - \varphi_n\|_1}{\|\varphi\|_1} \le \frac{8c_0(z)}{q_n} \int_0^{h_n} g(\tau)\,d\tau, \tag{1.20}$$

where $c_0(z)$ is given by (1.14) and computed with the 1-norm.

Proof: See [2]. □

In the case (1.7), the matrix A_n of the linear system (1.10) has entries

$$\mathsf{A}_n(i,j) := \frac{\varpi}{2h_{n,i}} \int_{\tau_{n,i-1}}^{\tau_{n,i}} \int_0^{\tau_0} E_1(|\tau - \tau'|) e_{n,j}(\tau')\,d\tau'\,d\tau, \tag{1.21}$$

and the second member b_n has entries

$$\mathsf{b}_n(i) := \frac{\varpi}{2h_{n,i}} \int_{\tau_{n,i-1}}^{\tau_{n,i}} \int_0^{\tau_0} E_1(|\tau - \tau'|) f(\tau')\,d\tau'\,d\tau. \tag{1.22}$$

For more details, see [3]. An application to the transfer problem in astrophysics gives (1.6) with $z = 1$, and as free term,

$$f(\tau) := \begin{cases} -1 & \text{if } 0 \le \tau \le \tau_0/2, \\ 0 & \text{if } \tau_0/2 < \tau \le \tau_0, \end{cases} \tag{1.23}$$

which describes a sudden drop of the temperature on the $\tau = \tau_0/2$ layer of the atmosphere. For further details on the physical model, see [4].

2 Iterative refinement of approximate solutions

To attain a given precision on the approximate solution φ_n, it may be necessary that the largest grid step h_n be so small that the dimension of the corresponding linear system will be prohibitively large from a computational point of view. Not only the algorithm's stability becomes poor but also the condition number of the matrix may increase if its size increases. Refinement schemes allow us to attain iteratively the exact solution of a large scale linear system by means of the resolution of a sequence of linear systems of moderate fixed size. Let us consider the general framework of a complex Banach space X and a linear compact operator $T : X \to X$. If z is in the resolvent set of T, then $z \neq 0$. Let T_n be a sequence of linear bounded operators in X such that $\|T - T_n\| \to 0$ in the operator norm. Then, for n large enough, z belongs to the resolvent set of T_n and $R_n(z)$ is norm-convergent to $R(z)$.

The most elementary way to refine the approximate solution $\varphi_n := R_n(z)f$ is the following.

$$\text{Scheme A} \quad \begin{cases} x^{(0)} := \varphi_n, \\ x^{(k+1)} := x^{(k)} - R_n(z)(Tx^{(k)} - zx^{(k)} - f), \quad k \ge 0. \end{cases} \tag{2.1}$$

We can interpret $R_n(z)$ as an approximation of the inverse of the Fréchet derivative of the affine operator $x \mapsto (T - zI)x - f$, the exact one being $R(z)$. Since $R(z)$ satisfies the identities

$$R(z) = \frac{1}{z}(R(z)\,T - I) = \frac{1}{z}(T\,R(z) - I) \tag{2.2}$$

two new different approximations of $R(z)$ are thus motivated,

$$\widetilde{R}_n(z) := \frac{1}{z}(R_n(z)T - I), \quad \widehat{R}_n(z) := \frac{1}{z}(TR_n(z) - I). \tag{2.3}$$

These approximate resolvent operators lead to the following iterative refinement schemes,

$$\text{Scheme B} \quad \begin{cases} \widetilde{x}^{(0)} := \widetilde{R}_n(z)f, \\ \widetilde{x}^{(k+1)} := \widetilde{x}^{(k)} - \widetilde{R}_n(z)(T\widetilde{x}^{(k)} - z\widetilde{x}^{(k)} - f), \quad k \ge 0, \end{cases} \tag{2.4}$$

$$\text{Scheme C} \quad \begin{cases} \widehat{x}^{(0)} := \widehat{R}_n(z)f, \\ \widehat{x}^{(k+1)} := \widehat{x}^{(k)} - \widehat{R}_n(z)(T\widehat{x}^{(k)} - z\widehat{x}^{(k)} - f), \quad k \ge 0. \end{cases} \tag{2.5}$$

Since the computation of residuals which tend to zero, as well as the resolution of almost homogeneous linear systems may be unstable, the following theorems are interesting for algorithmic purposes.

Theorem 3 *In (2.1),* $x^{(k+1)} = x^{(0)} + R_n(z)(T_n - T)x^{(k)}$ *for* $k \geq 0$.

Theorem 4 *In (2.4),* $\widetilde{x}^{(k+1)} = \widetilde{x}^{(0)} + \dfrac{1}{z}R_n(z)(T_n - T)T\widetilde{x}^{(k)}$ *for* $k \geq 0$.

Theorem 5 *In (2.5),* $\widehat{x}^{(k+1)} = \widehat{x}^{(0)} + \dfrac{1}{z}TR_n(z)(T_n - T)\widehat{x}^{(k)}$ *for* $k \geq 0$.

Proof: For each $k \geq 0$, in (3),

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - R_n(z)(Tx^{(k)} - zx^{(k)} - f) \\ &= x^{(k)} - R_n(z)(T - T_n + T_n - zI)x^{(k)} + x^{(0)} \\ &= x^{(0)} + R_n(z)(T_n - T)x^{(k)} \end{aligned}$$

For (4) and (5), the proof follows the same idea but it is technically more complicated. □

In our application to the transfer equation in astrophysics, T is defined by (1.5) with g given by (1.7), and the equation (1.6) has $z = 1$.

3 Numerical computations

The iterative refinement schemes allow us to obtain the exact solution of a large scale linear system by solving a sequence of moderate fixed size ones. Each of the three iterative refinement schemes presented in this work are based on an approximation, say $G_n(z)$, of the resolvent operator $R(z)$. Their common structure is the following.

$$\begin{cases} \xi^{(0)} &:= G_n(z)f, \\ \xi^{(k+1)} &:= \xi^{(0)} + (I - G_n(z)(T - zI))\xi^{(k)}, \quad k \geq 0. \end{cases} \tag{3.1}$$

Theorem 6 *Let* $c_1(z) := 8c_0(z)\max\{1, \|T\|_1/|z|\}$, *and* $(\xi^{(k)})_{k\geq 0}$ *be any of the sequences (2.1), (2.4) or (2.5). Then*

$$\frac{\|\xi^{(k)} - \varphi\|_1}{\|\varphi\|_1} \leq \Big(\frac{c_1(z)}{q_n}\int_0^{h_n} g(\tau)\,d\tau\Big)^{k+1}, \quad k \geq 0.$$

Proof: Let us prove the bound for the sequence defined by (2.1). For the other two, the arguments are similar. Using Theorem 3, we have

$$\begin{aligned} x^{(k)} - \varphi &= \big(R_n(z)(T_n - T)\big)^k(x^{(0)} - \varphi), \\ x^{(0)} - \varphi &= R_n(z)(T - T_n)\varphi. \end{aligned}$$

Hence,

$$\|x^{(k)} - \varphi\|_1 \leq \|(R_n(z)(T - T_n))^{k+1}\|_1\|\varphi\|_1,$$

and, in [2], we have shown that $\|R_n(z)(T_n - T)\|_1 \leq \dfrac{8c_0(z)}{q_n}\displaystyle\int_0^{h_n} g(\tau)\,d\tau$. □

All the schemes need evaluations of T at some prescribed functions of X. In practice T is not used for this purpose but an operator T_m of the sequence $(T_\nu)_{\nu\geq 1}$ is used instead,

where $m > n$. We consider the kernel g defined by (1.7) and the free term f defined by (1.23). Table 1 gives the number of iterations performed by each scheme for several values of ϖ in order to obtain a first relative residual less than or equal to 10^{-12}, when a quasi-uniform grid $(\tau_{\nu,i})_{i=0}^{\nu}$ is built such that ν is a multiple of 10, $\tau_0 = 1000$,

$$n = 200, \quad m = 1000, \quad \text{and} \quad h_{\nu,i} := \begin{cases} \dfrac{\tau_0}{2\nu} & \text{if} \quad i \in [1, \ldots, \frac{\nu}{5}], \\ \dfrac{\tau_0}{5\nu} & \text{if} \quad i \in [\frac{\nu}{5} + 1, \ldots, \frac{\nu}{2}], \\ \dfrac{\tau_0}{2\nu} & \text{if} \quad i \in [\frac{\nu}{2} + 1, \ldots, \frac{9\nu}{10}], \\ \dfrac{4\tau_0}{\nu} & \text{if} \quad i \in [\frac{9\nu}{10} + 1, \ldots, \nu]. \end{cases} \tag{3.2}$$

Albedo ϖ	Scheme A (2.1)	Scheme B (2.4)	Scheme C (2.5)
0.750	29	15	14
0.990	46	27	26
0.999	385	196	195

TAB 1. Number of iterations.

Figures 1, 2 and 3 show the last iterate of all schemes, as well as the corresponding convergence histories, for $\varpi \in \{0.750, 0.990, 0.999\}$. As we can see, the schemes B and C are much faster than Atkinson's formula A, specially when the albedo is close to 1. In the latter situation a wider boundary layer arises at the left of the atmosphere, and the decay at the middle point takes place along a wider subinterval.

A survey on different discretization methods for integral operators can be found in [1], with special emphasis on spectral applications. In what concerns condition number of associated linear systems, the reader is refered to [7], [5] and [6].

Bibliography

1. M. Ahues, A. Largillier and B.V. Limaye, *Spectral Computations with Bounded Operators*, Chapman and Hall, Boca Raton, 2001.
2. M. Ahues, A. Largillier and O. Titaud, The roles of a weak singularity and the grid uniformity in the relative error bounds, Numer. Funct. Anal. and Optimiz. **22**, 789–814, 2001.
3. M. Ahues, F. D'Almeida, A. Largillier, O. Titaud and P. Vasconcelos, An L^1 Refined Projection approximate solution of the radiation transfer equation in stellar atmospheres, Journal of Computational and Applied Mathematics **140**, 13–26, 2002.
4. I. W. Busbridge, *The Mathematics of Radiative Transfer*, Cambridge University Press, 1960.

5. L. N. Desphande and B.V. Limaye, On the stability of singular finite-rank methods, SIAM J. Numer. Anal. **27**, 792–803, 1990.
6. A. Largillier and B.V. Limaye, Finite-rank methods and their stability for coupled systems of operator equations, SIAM J. Numer. Anal. **2**, 707–728, 1996.
7. R. Whitley, The stability of finite-rank methods with applications to integral equations, SIAM J. Numer. Anal. **23**, 118–134, 1986.

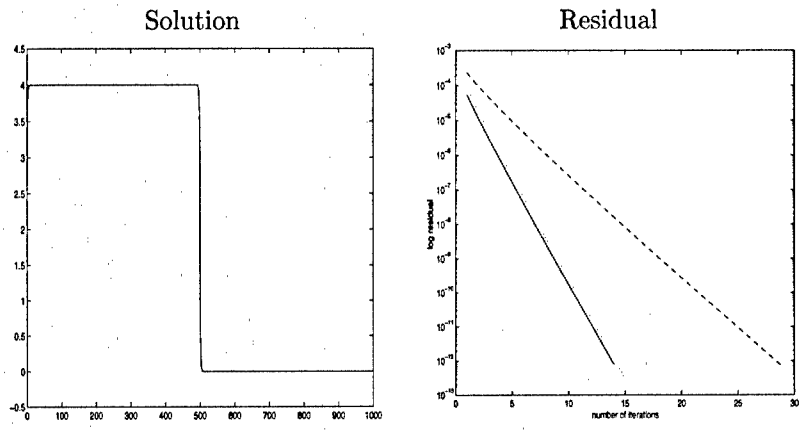


FIG. 1. Solution and convergence history for $\varpi = 0.750$: Scheme A — dashed line, Scheme B — dotted line, Scheme C — solid line.

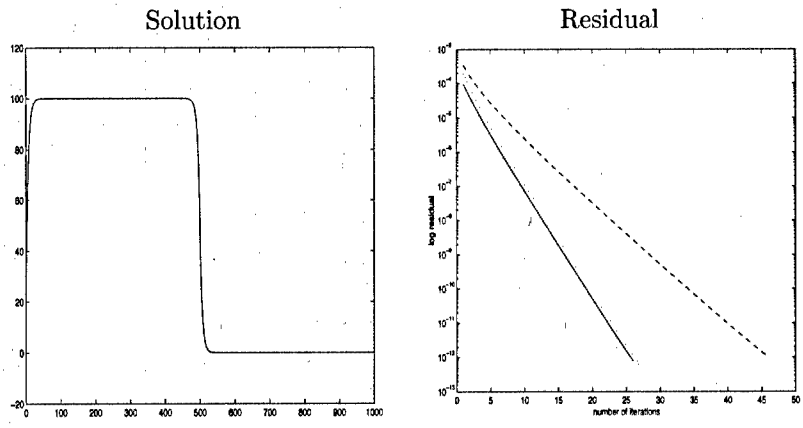


FIG. 2. Solution and convergence history for $\varpi = 0.990$: Scheme A — dashed line, Scheme B — dotted line, Scheme C — solid line.

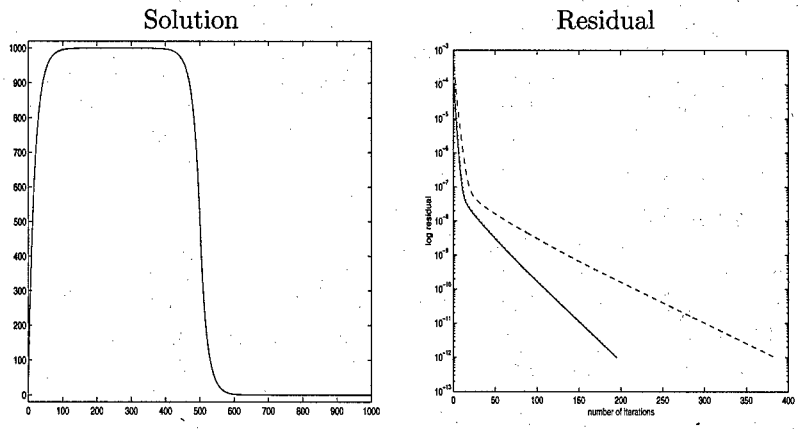


FIG. 3. Solution and convergence history for $\varpi = 0.999$: Scheme A — dashed line, Scheme B — dotted line, Scheme C — solid line.

Geometrical symmetry in symmetric Galerkin BEM

Alessandra Aimi and Mauro Diligenti

Department of Mathematics, University of Parma, Italy.
alessandra.aimi@unipr.it, mauro.diligenti@unipr.it

Abstract

We consider a symmetric boundary integral formulation associated with a mixed boundary value problem defined on a domain $\Omega \in \mathbb{R}^2$ with piecewise smooth boundary Γ. We assume that $\overline{\Omega}$ is mapped onto itself by a finite group $\mathcal{G}$ of congruences having at least two distinct elements. Hence, we can decompose the related symmetric Galerkin BEM problem into independent subproblems of reduced dimension with respect to the complete one. Shape functions for each subproblem can be obtained from classical BEM basis, ordered as a vector, applying suitable *restriction matrices* constructed starting from group representation theory.

1 Introduction

Let $\Omega \subset \mathbb{R}^2$, be a bounded domain with a piecewise smooth boundary Γ. The boundary Γ is partitioned into two non intersecting open subset Γ_1 and Γ_2, with $\Gamma = \overline{\Gamma}_1 \bigcup \overline{\Gamma}_2 = \bigcup_{j=1}^{J} \overline{\Gamma}^j$, Γ^j being an open strainght line segments. In the following we always assume $\operatorname{meas} \Gamma_1 > 0$. The solution of the mixed boundary value problem

$$L(x)\, u(x) = 0 \qquad \text{in } \Omega\,, \tag{1.1}$$

$$u(x) = u^*(x) \qquad \text{on } \Gamma_1\,, \qquad q(x) := \frac{\partial u}{\partial \mathbf{n}} = q^*(x) \qquad \text{on } \Gamma_2, \tag{1.2}$$

can be expressed by the representation formula

$$u(x) = \int_\Gamma U(x,y) q(y)\, \mathrm{d}y - \int_\Gamma \frac{\partial}{\partial \mathbf{n}_y} U(x,y) u(y)\, \mathrm{d}y, \qquad x \in \Omega\,. \tag{1.3}$$

In (1.1) $L(\cdot)$ is an elliptic partial differential operator of second order, $U(x,y)$ its fundamental solution (see [4] for a general discussion). In (1.2) $\frac{\partial u}{\partial \mathbf{n}}$ denotes the derivative with respect to the outher normal $\mathbf{n}$ to Γ, and u^* and q^* are given functions. Applications of (1.1)-(1.2) are, for instance, boundary value problems in potential theory and in elastostatic. From (1.3) it is clear that if we want to recover u in Ω we have firstly to know the remaining Cauchy data, since in (1.2) these functions are given only partially. Taking the limit of $u(x)$ for $x \in \Gamma_a$ and the normal derivative $\frac{\partial u}{\partial \mathbf{n}}(x)$ for $x \in \Gamma_2$ in this formula and using the jump relations, one finds the system [2]

$$\int_{\Gamma_1} U(x,y) q(y)\, \mathrm{d}y - \oint_{\Gamma_2} \frac{\partial}{\partial \mathbf{n}_y} U(x,y) u(y)\, \mathrm{d}y = f_1(x)\,, \qquad x \in \Gamma_1,$$

$$-\int\!\!\!\!\!\!-_{\Gamma_1} \frac{\partial}{\partial \mathbf{n}_x} U(x,y)q(y)\,dy + \int\!\!\!\!\!\!=_{\Gamma_2} \frac{\partial^2}{\partial \mathbf{n}_x \partial \mathbf{n}_y} U(x,y)u(y)\,dy = f_2(x)\,, \quad x \in \Gamma_2. \tag{1.4}$$

In order to perform the Galerkin method, we need a family of finite-dimensional subspaces $\{U_{h,p}(\Gamma)\}$ defined on Γ. Let us define a mesh Γ_h^j for each Γ^j: $\overline{\Gamma}^j = \bigcup_{i=1}^{N_h^j} \overline{\Gamma}_{h,i}^j$ such that $\Gamma_{h,i}^j$ is an open segment. We define for $p \geq 0$, $h > 0$, $U_{h,p}(\Gamma_1)$ to be the set of functions on Γ_1 whose restrictions to $\Gamma^j \subset \Gamma_1$ belong to the set of all polynomials of degree $\leq p$ on $\Gamma_{h,i}^j$. Moreover, for $p \geq 1$, $U_{h,p}^\circ(\Gamma_2)$ will denote those continuous functions on Γ_2 whose restrictions to $\Gamma^j \subset \Gamma_2$ belong to $C^\circ(\Gamma_2)$ and which vanish at the end points of Γ_2. The approximating boundary element shape functions of degree $p > 0$ are defined through the standard assembling of the local basis functions defined on each $\Gamma_{h,i}^j$. We then define

$$U_{h,p}(\Gamma) := \operatorname{span}\{(\varphi_i,\, \psi_\ell) \;:\; \varphi_i \in U_{h,p}^\circ(\Gamma_2),\; \psi_\ell \in U_{h,p}(\Gamma_1)\}. \tag{1.5}$$

The corresponding symmetric Galerkin boundary elements scheme for (1.4) leads to a linear system of the form

$$A\xi = b. \tag{1.6}$$

If the boundary Γ presents symmetry properties, we will exploit them to reduce the computational cost of the solution of (1.6), using a decomposition result for the Galerkin boundary element problem that we will introduce at the end of the next section.

2 Matrix representation of a finite group of congruences and projection operators

Let $\mathcal{G}$ be a finite group of t congruences ($t \geq 2$) of the Euclidean space $\mathbb{R}^m$ ($m = 2, 3$). The group $\mathcal{G}$ can be described by orthogonal matrices γ_i of order m. Let $\{\gamma_1, \ldots, \gamma_t\}$ be the elements of $\mathcal{G}$, γ_1 the identity matrix. From the theory of group representation [5] it follows that any finite group $\mathcal{G}$ admits a finite number q of unitary irreducible, pairwise inequivalent matrix representations

$$\{\omega^{(1)}(\gamma_i)\},\ \{\omega^{(2)}(\gamma_i)\}, \ldots,\ \{\omega^{(q)}(\gamma_i)\} \qquad (i = 1, \ldots, t). \tag{2.1}$$

Let d_ℓ be the order of the representation $\{\omega^{(\ell)}(\gamma_i)\}$, i.e., the order of the matrices $\omega^{(\ell)}(\gamma_i)$. The number q of the representations (2.1) and the orders $d_1, \ldots, d_q$ only depend on $\mathcal{G}$. Any representation $\{\omega^{(\ell)}(\gamma_i)\}$ of order $d_\ell \geq 2$, can be replaced, in the system (2.1), by an equivalent unitary representation. Representations of order 1 are univocally determined. We observe that, if γ_i and γ_j are two elements of $\mathcal{G}$, then $\omega^{(\ell)}(\gamma_i\gamma_j) = \omega^{(\ell)}(\gamma_i)\omega^{(\ell)}(\gamma_j)$, $\omega^{(\ell)}(\gamma_i^{-1}) = [\omega^{(\ell)}(\gamma_i)]^*$, where $[\omega^{(\ell)}(\gamma_i)]^*$ denote the transpose of the matrix $\omega^{(\ell)}(\gamma_i)$. Always from the theory of group representation it follows that $q \leq t$ and the relation $d_1^2 + d_2^2 + \cdots + d_q^2 = t$ holds. Furthermore, $q = t$ if and only if $d_1 = d_2 = \cdots = d_q = 1$. Having set $M = d_1 + d_2 + \cdots + d_q$, then $q \leq M \leq t$, and we have $q = M = t$ if and only if $\mathcal{G}$ is an abelian group.

Let Ω be a bounded domain in $\mathbb{R}^2$ with a piecewise smooth boundary Γ, *invariant* with respect to $\mathcal{G}$, i.e., sent onto itself by the congruences of $\mathcal{G}$. Also the boundary Γ is invariant with respect to $\mathcal{G}$, i.e., for any $\gamma_\ell \in \mathcal{G}$ and $x \in \Gamma$, $(\gamma_\ell x) \in \Gamma$.

Let $\mathcal{W}(\Gamma)$ be the real vector space of real functions defined on Γ. We can associate to any element γ_i of $\mathcal{G}$ a linear transformation T_i defined, for any $v \in \mathcal{W}(\Gamma)$, by

$$(T_i v)(x) := v(\gamma_i^{-1} x) \qquad x \in \Gamma\,, \tag{2.2}$$

where T_i is a linear, invertible transformation from $\mathcal{W}(\Gamma)$ onto $\mathcal{W}(\Gamma)$, and T_1 is the identity.

Definition 2.1 *A subset $\mathcal{V}(\Gamma)$ of $\mathcal{W}(\Gamma)$ is said to be invariant with respect to $\mathcal{G}$ (or $\mathcal{G}$-invariant) if for any $v \in \mathcal{V}(\Gamma)$ and any $\gamma_i \in \mathcal{G}$, $T_i v \in \mathcal{V}(\Gamma)$.*

Obviously if v is a function of $\mathcal{W}(\Gamma)$, not identically equal to zero, the set of functions $\{T_i v,\ i = 1, \ldots, t\}$ is invariant with respect to $\mathcal{G}$.

Definition 2.2 *Let $\mathcal{L}$ be a linear operator in $\mathcal{V}(\Gamma)$. We will say that $\mathcal{L}$ is invariant with respect to $\mathcal{G}$ if for any $u \in \mathcal{V}(\Gamma)$: $\mathcal{L} T_i u = T_i \mathcal{L} u$, $i = 1, \ldots, t$.*

Example 2.3 Let $\mathcal{V}(\Gamma)$ be a suitable Sobolev space and $(\mathcal{L}f)(x) := \int_\Gamma \mathcal{K}(x,y) f(y) \mathrm{d}\Gamma_y$ an integral operator defined on $\mathcal{V}(\Gamma)$, with kernel $\mathcal{K}(x,y)$.
We have: $T_i(\mathcal{L}f)(x) = \int_\Gamma \mathcal{K}(\gamma_i^{-1}x, y) f(y) \mathrm{d}\Gamma_y$; since $\gamma_i \in \mathcal{G}$ is an isometry, the mapping $y \to \gamma_i y$ preserves the differential element $\mathrm{d}\Gamma_y$. Thus

$$\mathcal{L}(T_i f)(x) = \int_\Gamma \mathcal{K}(x,y) f(\gamma_i^{-1} y) \mathrm{d}\Gamma_y = \int_\Gamma \mathcal{K}(x, \gamma_i y) f(y) \mathrm{d}\Gamma_y\,.$$

Then the integral operator $\mathcal{L}$ is $\mathcal{G}$-invariant if the kernel $\mathcal{K}(x,y)$ satisfies the condition $\mathcal{K}(x,y) = \mathcal{K}(\gamma_i x, \gamma_i y)$ for all $x,\ y \in \Gamma,\ i = 1, \ldots, t$.

Starting from the group $\mathcal{G}$, the system of representation (2.1) and the linear transformations T_i defined by (2.2), we can introduce M linear transformations of $\mathcal{W}(\Gamma)$,

$$P_{\ell k} = \frac{d_\ell}{t} \sum_{i=1}^{t} \omega_{kk}^{(\ell)}(\gamma_i) T_i \qquad (\ell = 1, \ldots, q;\ k = 1, \ldots, d_\ell)\,. \tag{2.3}$$

Owing to the property of the representations (2.1), there holds

$$P_{\ell k}^2 = P_{\ell k}, \qquad P_{\ell k} P_{\ell' k'} = 0 \quad \text{if } (\ell, k) \neq (\ell', k'), \qquad \sum_{\ell=1}^{q} \sum_{k=1}^{d_\ell} P_{\ell k} = T_1. \tag{2.4}$$

The linear transformations $P_{\ell k}$, which will be called *projection operators*, determine a decomposition of any vector space $\mathcal{V}(\Gamma) \subset \mathcal{W}(\Gamma)$ invariant with respect to $\mathcal{G}$, into a direct sum of M subspaces $\mathcal{V}_{\ell k}(\Gamma)$; $\mathcal{V}_{\ell k}(\Gamma)$ is the co-domain of $P_{\ell k}$, viewed as a linear transformation from $\mathcal{V}(\Gamma)$ onto itself.

If $\mathcal{G}$ is a non-abelian group, it is useful to consider in the space $\mathcal{W}(\Gamma)$ further linear transformations linked to the system (2.1). Let $\{\omega^{(\ell)}(\gamma_i)\}$ be a representation of $\mathcal{G}$ of order $d_\ell \geq 2$. Let us consider d_ℓ^2 linear transformations, already introduced in [1], defined as follows

$$A_{kr}^{(\ell)} = \frac{d_\ell}{t} \sum_{i=1}^{t} \omega_{kr}^{(\ell)}(\gamma_i) T_i\,, \qquad k, r = 1, \ldots, d_\ell\,. \tag{2.5}$$

If $k = r$, then $A_{kr}^{(\ell)} = P_{\ell k}$.

Definition 2.4 *Let $\mathcal{B}(\cdot,\cdot)$ be a bilinear form from $\mathcal{V}(\Gamma) \times \mathcal{V}(\Gamma)$ on $\mathbb{R}$. We will say that $\mathcal{B}(\cdot,\cdot)$ is $\mathcal{G}$-invariant if for, any $u, v \in \mathcal{V}(\Gamma)$,*

$$\mathcal{B}(T_i u, T_i v) = \mathcal{B}(u, v), \qquad i = 1, \ldots, t. \tag{2.6}$$

Let $\mathcal{V}(\Gamma)$ be a Hilbert space and let us consider the following problem

$$\textit{find} \qquad u \in \mathcal{V}(\Gamma) \ : \ \mathcal{B}(u, v) = \mathcal{F}(v) \qquad \text{for all } v \in \mathcal{V}(\Gamma), \tag{2.7}$$

where $\mathcal{B}(\cdot,\cdot)$ is continuous and coercive, and $\mathcal{F}(\cdot) : \mathcal{V}(\Gamma) \to \mathbb{R}$ a linear continuous functional. If Γ and $\mathcal{V}(\Gamma)$ are invariant with respect to $\mathcal{G}$, and $\mathcal{V}(\Gamma) = \oplus_{\ell=1}^{q} \oplus_{k=1}^{d_\ell} \mathcal{V}_{\ell k}(\Gamma)$ is the decomposition of $\mathcal{V}(\Gamma)$ defined by the projection operators (2.3) the following fundamental result holds.

Theorem 2.5 *If $\mathcal{B}(\cdot,\cdot)$ verifies the condition* (2.6) *and $P_{\ell k}$ are the projection operators defined in* (2.3)*, then the problem* (2.7) *can be decomposed into M independent problems; find $u_{\ell k} \in \mathcal{V}_{\ell k}(\Gamma)$ such that*

$$\mathcal{B}(u_{\ell k}, v_{\ell k}) = \mathcal{F}(v_{\ell k}) \quad \text{for all } v_{\ell k} \in \mathcal{V}_{\ell k}(\Gamma), \quad \ell = 1, \ldots, q;\ k = 1, \ldots, d_\ell\,. \tag{2.8}$$

The solution of (2.7) *can be recovered as $u = \oplus_{\ell=1}^{q} \oplus_{k=1}^{d_\ell} u_{\ell k}$.*

The above result can be applied, under the invariance hypothesis, in the discrete form to the symmetric Galerkin BEM scheme if we choose the finite dimensional subspace $U_{h,p}(\Gamma)$ defined in (1.5), to be $\mathcal{G}$-*invariant* too, and therefore decomposable as $U_{h,p}(\Gamma) = \oplus_{\ell=1}^{q} \oplus_{k=1}^{d_\ell} U_{h,p}^{\ell k}(\Gamma)$. Then the symmetric Galerkin boundary element problem can be decomposed into M independent problems which have reduced dimension with respect to the original one and which can be solved on parallel processors. Now one has to construct boundary element basis functions for each subspace $U_{h,p}^{\ell k}(\Gamma)$. With some simple geometries (and groups of congruences) this can be done directly, but in many cases this is a difficult task. We solve it here by applying *restriction matrices*, which we introduce in the next sections, to the basis of $U_{h,p}(\Gamma)$, ordered as a vector. Since there is a ono-to-one correspondence between the standard boundary element shape functions and the nodes of the mesh fixed on Γ, in the following we will work directly on the nodes of the boundary.

3 Elementary restriction matrices

In this section we introduce suitable matrices depending only on the group $\mathcal{G}$ and on the system of representations (2.1), which will be called *elementary restriction matrices*. In the following sections we will see how, starting from these, we can construct restriction matrices relative to a mesh defined on Γ. We fix a finite group $\mathcal{G} \equiv \{\gamma_1, \ldots, \gamma_t\}$ of congruences of $\mathbb{R}^m$ and a system (2.1) of orthogonal irreducible, pairwise inequivalent representations of $\mathcal{G}$. $\mathcal{G}$ always admits the representation $\{1, 1, \ldots, 1\}$ which we indicate by $\{\omega^{(1)}(\gamma_i)\}$; let us order the remaining representations (2.1) with increasing order d_ℓ; let $\{\omega^{(1)}(\gamma_i)\}, \ldots, \{\omega^{(s)}(\gamma_i)\}$ be the representations of order 1. If $\mathcal{G}$ is an abelian group one has $s = q = t$ and $d_1 = d_2 = \cdots = d_t = 1$. If $\mathcal{G}$ is a nonabelian group, it holds $s < q < t$ and therefore $d_1 = d_2 = \cdots = d_s = 1\,, \quad 2 \le d_{s+1} \le \cdots \le d_q$.

Let $\mathcal{G}$ be an abelian group. We will call *elementary restriction matrices* the following t matrices, with 1 row and t columns

$$R_{\ell 1} = \frac{1}{\sqrt{t}}\left(\omega^{(\ell)}(\gamma_1) \cdots \omega^{(\ell)}(\gamma_t)\right), \quad \ell = 1, \ldots, t. \tag{3.1}$$

Since representations $\{\omega^{(\ell)}(\gamma_i)\}$ are real, it follows that $\omega^{(\ell)}(\gamma_i) = \pm 1$, for ℓ, $i = 1, \ldots, t$.

Let $\mathcal{G}$ be a nonabelian group. Correspondingly to the representations $\{\omega^{(\ell)}(\gamma_i)\}$ of order 1 of the system (2.1), we introduce matrices $R_{\ell 1}$ with 1 row and t columns

$$R_{\ell 1} = \frac{1}{\sqrt{t}}\left(\omega^{(\ell)}(\gamma_1) \ldots \omega^{(\ell)}(\gamma_t)\right), \quad \ell = 1, \ldots, s. \tag{3.2}$$

We obtain, in this case, s matrices. Let now $\{\omega^{(\ell)}(\gamma_i)\}$ be a representation of the system (2.1) of order d_ℓ, with $d_\ell \geq 2$. With $k = 1, \ldots, d_\ell$ fixed, let us consider the following matrix, with d_ℓ rows and t columns

$$R_{\ell k} = \sqrt{\frac{d_\ell}{t}} \begin{pmatrix} \omega_{1k}^{(\ell)}(\gamma_1) & \omega_{1k}^{(\ell)}(\gamma_2) & \cdots & \omega_{1k}^{(\ell)}(\gamma_t) \\ \omega_{2k}^{(\ell)}(\gamma_1) & \omega_{2k}^{(\ell)}(\gamma_2) & \cdots & \omega_{2k}^{(\ell)}(\gamma_t) \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{d_\ell k}^{(\ell)}(\gamma_1) & \omega_{d_\ell k}^{(\ell)}(\gamma_2) & \cdots & \omega_{d_\ell k}^{(\ell)}(\gamma_t) \end{pmatrix}. \tag{3.3}$$

Due to the orthogonality properties of the representation $\{\omega^{(\ell)}(\gamma_i)\}$, matrix $R_{\ell k}$ has pairwise orthonormal rows. Therefore the rank of matrix $R_{\ell k}$ is d_ℓ. For any representation $\{\omega^{(\ell)}(\gamma_i)\}$ we obtain d_ℓ matrices $R_{\ell k}$ $(k = 1, \ldots, d_\ell)$. Matrices $R_{\ell k}$ $(\ell = 1, \ldots, q;\ k = 1, \ldots, d_\ell)$ defined in (3.2) and (3.3) will be called *elementary restriction matrices.* The total number of these matrices is M, with $M = d_1 + d_2 + \cdots + d_q$. The matrices defined in (3.1) or (3.2)-(3.3) satisfy some properties, easily deducible from orthogonality relations (2.4) and which we summarise in the following.

Theorem 3.1 ([1]) *The M elementary restriction matrices defined by* (3.1) *or* (3.2)-(3.3) *verify the relations*

$$R_{\ell k} R_{\ell k}^* = I_{d_\ell}, \qquad R_{\ell k} R_{\ell' k'}^* = 0 \text{ if } (\ell, k) \neq (\ell', k'), \qquad \sum_{\ell=1}^{q} \sum_{k=1}^{d_\ell} R_{\ell k}^* R_{\ell k} = I \tag{3.4}$$

where I_{d_ℓ}, I are identity matrices of order d_ℓ and t respectively.

4 $\mathcal{H}(\Sigma_a)$ spaces and elementary restriction matrices

Let Γ be the piecewise smooth boundary of Ω, invariant with respect to $\mathcal{G}$, and $a \in \Gamma$. Consider the ordered set

$$\Sigma_a = \{a, \gamma_2^{-1} a, \ldots, \gamma_t^{-1} a\}, \tag{4.1}$$

and the space $\mathcal{H}(\Sigma_a)$ of real functions defined in Σ_a. A natural basis B in $\mathcal{H}(\Sigma_a)$ is formed by functions having value 1 in a point of Σ_a and 0 in the remaining points. Having indicated with χ the function of B with value 1 in the point a, we obtain the

ordered basis $B \equiv \{\chi(x), \chi(\gamma_2 x), \ldots, \chi(\gamma_t x)\}$, such that, of course,

$$\mathcal{H}(\Sigma_a) = \text{span}\{\chi(x), \chi(\gamma_2 x), \ldots, \chi(\gamma_t x)\}. \tag{4.2}$$

$\mathcal{H}(\Sigma_a)$ is a vector space with finite dimension $n \leq t$, invariant with respect to $\mathcal{G}$ (since Σ_a is invariant with respect to $\mathcal{G}$) and therefore decomposable into direct sum of M subspaces $\mathcal{H}_{\ell k}(\Sigma_a)$. Having set $n_\ell = \dim\mathcal{H}_{\ell k}(\Sigma_a)$, we have $n = \sum_{\ell=1}^{q} d_\ell n_\ell$.

Definition 4.1 *We say that a is a generic point of Γ (with respect to the group $\mathcal{G}$) if $\dim\mathcal{H}(\Sigma_a) = t$ or, equivalently, if all the elements of Σ_a are distinct.*

The following results hold.

Theorem 4.2 ([1]) *Having fixed any point $a \in \Gamma$, if $\{\omega^{(\ell)}(\gamma_i)\}$ is a representation of order 1, then $\mathcal{H}_{\ell 1}(\Sigma_a) = \text{span}\{P_{\ell 1}\chi\}$ and $n_\ell \leq 1$. If $\{\omega^{(\ell)}(\gamma_i)\}$ is a representation of order $d_\ell \geq 2$, one has*

$$\mathcal{H}_{\ell k}(\Sigma_a) = \text{span}\{A^{(\ell)}_{k1}\chi, \ldots, A^{(\ell)}_{kd_\ell}\chi\}, \qquad k = 1, \ldots, d_\ell\,, \tag{4.3}$$

and therefore $n_\ell \leq d_\ell$. If a is a generic point, then $n_\ell = d_\ell$ for any ℓ.

Let now V^t be the column vector $(\chi(x), \chi(\gamma_2 x), \ldots, \chi(\gamma_t x))^*$, whose order is related to that one fixed for the elements of $\mathcal{G}$. Corresponding to the representations of order 1 of $\mathcal{G}$, for the elementary restriction matrices defined in (3.1), (3.2) we have $R_{\ell 1}V^t = \sqrt{t}P_{\ell 1}\chi$. From Theorem 4.2, it follows that

$$\mathcal{H}_{\ell 1}(\Sigma_a) = \text{span}\{R_{\ell 1}V^t\}\,. \tag{4.4}$$

Corresponding to the representations of order $d_\ell \geq 2$, for the elementary restriction matrices defined in (3.3) we have $R_{\ell k}V^t = \sqrt{t/d_\ell}\left(A^{(\ell)}_{k1}\chi, A^{(\ell)}_{k2}\chi, \ldots, A^{(\ell)}_{kd_\ell}\chi\right)^*$. From (4.3), it follows that

$$\mathcal{H}_{\ell k}(\Sigma_a) = \text{span}\{R_{\ell k}V^t\}\,. \tag{4.5}$$

In both cases, if *a is a generic point*, the components of the vector $R_{\ell k}V^t$ constitute a basis in $\mathcal{H}_{\ell k}(\Sigma_a)$. Therefore, for any *generic point* a, the elementary restriction matrix $R_{\ell k}$ represents the projection operator $P_{\ell k}$ from $\mathcal{H}(\Sigma_a)$ onto $\mathcal{H}_{\ell k}(\Sigma_a)$, if we choose V^t as a basis in $\mathcal{H}(\Sigma_a)$.

Now, we want to construct elementary restriction matrices $R_{\ell k}$ which represent the projection operators $P_{\ell k}$ from $\mathcal{H}(\Sigma_a)$ onto $\mathcal{H}_{\ell k}(\Sigma_a)$ for nongeneric points. Therefore let us suppose a to be a *nongeneric point*, i.e., such that the functions

$$\chi(x), \chi(\gamma_2 x), \ldots, \chi(\gamma_t x) \tag{4.6}$$

are *linearly dependent*. Let n be the maximum number of linearly independent functions among (4.6) and let the following functions be linearly independent,

$$\chi(\gamma_{i_1} x), \ldots, \chi(\gamma_{i_n} x). \tag{4.7}$$

It is convenient to order the functions (4.7) with increasing index i_α; therefore let us suppose $i_1 < i_2 < \cdots < i_n$. In this case elementary restriction matrices $R_{\ell k}$ will have n columns. The number n_ℓ of rows ($n_\ell \leq d_\ell$) of each $R_{\ell k}$ is not determined by $i_1, i_2, \ldots, i_n$.

In general, we only can say that matrices $R_{\ell 1}, \ldots, R_{\ell d_\ell}$ have the same number n_ℓ of rows, where $n_\ell = \dim \mathcal{H}_{\ell k}(\Sigma_a)$.

Then we now consider a significant class of nongeneric points. Having fixed ℓ ($\ell = 2, \ldots, t$), let $I_\ell(\Gamma)$ be the set of all points $a \in \Gamma$ such that

$$a = \gamma_\ell^{-1} a \,. \tag{4.8}$$

From (4.8) it follows, for any i : $\chi(\gamma_i x) = \chi(\gamma_\ell \gamma_i x)$. This implies that the functions (4.6) are naturally subdivided into subsets and any subset contains coincident functions. Then we can obtain elementary restriction matrices for the space $\mathcal{H}(\Sigma_a)$ with $a \in I_\ell(\Gamma)$ starting from elementary restriction matrices built in Section 3, with the following procedure,

- Let us sum to each column of index i_α ($\alpha = 1, \ldots, n$) all the columns of index j, with j such that $\gamma_j^{-1} a = \gamma_{i_\alpha}^{-1} a$. We indicate with $\tilde{R}_{\ell k}$ the obtained matrices, all with d_ℓ rows and n columns, but not all full-rank matrices; some of these may be zero matrices.
- Let us extract from *nonzero* matrices $\tilde{R}_{\ell k}$ submatrices $\bar{R}_{\ell k}$ made up of n_ℓ linearly independent rows.
- Finally, let us construct from $\bar{R}_{\ell k}$ matrices $R_{\ell k}$ with a row-orthonormalization procedure.

The (nonzero) matrices $R_{\ell k}$ verify the properties expressed by Theorem 3.1. Furthermore, matrices $R_{\ell k}$, applied to the vector $V^n = (\chi(\gamma_{i_1} x), \ldots, \chi(\gamma_{i_n} x))^*$ corresponding to a point $a \in I_\ell(\Gamma)$, give vectors whose components constitute a basis for $\mathcal{H}_{\ell k}(\Sigma_a)$. For this reason they represent the projection operators from $\mathcal{H}(\Sigma_a)$ onto $\mathcal{H}_{\ell k}(\Sigma_a)$, for any $a \in I_\ell(\Gamma)$. Then we will say that the matrices $R_{\ell k}$, with n_ℓ rows and n columns, are *elementary restriction matrices* for the space $\mathcal{H}(\Sigma_a)$ relative to points $a \in I_\ell(\Gamma)$. Furthermore $n = \sum_{\ell=1}^{q} d_\ell n_\ell$.

5 $\mathcal{H}(\Sigma)$ spaces and restriction matrices

Let Γ be the piecewise smooth boundary of Ω, Σ a set formed by N points of Γ constituting a not necessarily uniform mesh defined on Γ. Let us suppose Γ and Σ invariant with respect to $\mathcal{G}$. Let $\mathcal{H}(\Sigma)$ be the vector space of real functions defined in Σ. $\mathcal{H}(\Sigma)$ is a N-dimensional vector space, invariant with respect to $\mathcal{G}$; this is due to the fact that Σ is invariant with respect to $\mathcal{G}$. A natural basis B in $\mathcal{H}(\Sigma)$, invariant with respect to $\mathcal{G}$, is formed by functions having value 1 in a point of Σ and 0 in the remaining points. In order to more easily construct restriction matrices for the space $\mathcal{H}(\Sigma)$, or equivalently for the mesh Σ, it is suitable to introduce in the set Σ the following equivalence relation.

Definition 5.1 *We say that a point a' is equivalent to a'' if there exists an element $\gamma_i \in \mathcal{G}$ such that $a'' \equiv \gamma_i^{-1} a'$ (and therefore $a' \equiv \gamma_i a''$).*

The points of the set Σ are then subdivided into r equivalence classes. If $r = 1$ one has $\mathcal{H}(\Sigma) = \mathcal{H}(\Sigma_a)$, with $a \in \Sigma$. Then let us suppose $r \geq 2$. We order the points of the set Σ as follows; having indicated with $a_1, \ldots, a_r$ r pairwise inequivalent points of Σ, we consider the following ordered points

$$a_1, \gamma_2^{-1} a_1, \ldots, \gamma_t^{-1} a_1, a_2, \gamma_2^{-1} a_2, \ldots, \gamma_t^{-1} a_2, \ldots, a_r, \gamma_2^{-1} a_r, \ldots, \gamma_t^{-1} a_r. \tag{5.1}$$

If points (5.1) are distinct, we have $N = rt$. If some points among (5.1) coincide, we will erase from the sequence (5.1) a point if it is equal to a previous one. Then a sequence of N points, with $N < rt$, will remain, with $n^{(1)}$ points equivalent to a_1, $n^{(2)}$ equivalent to $a_2, \ldots$, $n^{(r)}$ equivalent to a_r. In both cases $\mathcal{H}(\Sigma) = \mathcal{H}(\Sigma_{a_1}) \oplus \mathcal{H}(\Sigma_{a_2}) \oplus \cdots \oplus \mathcal{H}(\Sigma_{a_r})$, with $\dim \mathcal{H}(\Sigma_{a_j}) = n^{(j)} \leq t$, $j = 1, \ldots, r$ and $N = n^{(1)} + n^{(2)} + \cdots + n^{(r)}$. We indicate by $C_{\ell k}^{(j)}$ the *elementary restriction matrices* relative to the space $\mathcal{H}(\Sigma_{a_j})$, constructed as indicated in Section 4. Let $n_\ell^{(j)}$ be the number of rows of the matrix $C_{\ell k}^{(j)}$; having fixed j, the number of columns of matrices $C_{\ell k}^{(j)}$, for any ℓ and k, is $n^{(j)}$. We consider therefore the following M block matrices

$$\tilde{R}_{\ell k} = \begin{pmatrix} C_{\ell k}^{(1)} & O & O & \cdots & O \\ O & C_{\ell k}^{(2)} & O & \cdots & O \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & O & \cdots & C_{\ell k}^{(r)} \end{pmatrix}, \tag{5.2}$$

with $\tilde{N}_\ell = n_\ell^{(1)} + n_\ell^{(2)} + \cdots + n_\ell^{(r)}$ rows and N columns, from which we have to eliminate the possible zero rows. Matrices $R_{\ell k}$ determined by this procedure, which we call *restriction matrices* for the space $\mathcal{H}(\Sigma)$ of dimension N, have rank equal to the number N_ℓ of the remained rows and for these matrices properties expressed in Theorem 3.1 still hold. In both cases, we have the following theorem.

Theorem 5.2 *Considering the basis B in $\mathcal{H}(\Sigma)$ as a column vector V^N with the order deduced from* (5.1)*, the components of the vector $R_{\ell k} V^N$ form a basis in $\mathcal{H}_{\ell k}(\Sigma)$. Therefore the M matrices $R_{\ell k}$, having fixed in $\mathcal{H}(\Sigma)$ the ordered basis V^N, determine a decomposition of $\mathcal{H}(\Sigma)$ in M subspaces, which coincides with the one obtained with the projection operators $P_{\ell k}$.*

Preliminary numerical results appear promising; algorithms for potential and linear elasticity problems are being implemented on parallel processors to analyse the efficiency of the proposed approach.

Bibliography

1. A. Aimi, L. Bassotti, and M. Diligenti, Groups of Congruences and Restriction Matrices, submitted to *BIT*.
2. A. Aimi and M. Diligenti, Hypersingular kernel integration in 3D Galerkin boundary element method, *J. Comp. Appl. Math.*, **138**, 1, (2002), 51–72.
3. L. Bassotti Rizza, Operatori lineari T-invarianti rispetto ad un gruppo di congruenze, *Ann. Mat. Pura ed Appl.*, **148**, (1987), 173–205.
4. J. L. Lions and E. Magenes: *Non-Homogeneous Boundary Value Problems and Applications I*, Springer-Verlag, Berlin, Heidelberg, New York, 1972.
5. V. I. Smirnov, *Linear Algebra and Group Theory*, McGraw Hill, New York, 1961.

The numerical simulation of the qualitative behaviour of Volterra integro-differential equations

John T. Edwards, Neville J. Ford and Jason A. Roberts
j.edwards@chester.ac.uk, njford@chester.ac.uk, j.roberts@chester.ac.uk

Chester College, Parkgate Road, Chester, CH1 4BJ, UK.

Abstract

We consider the qualitative behaviour of exact and approximate solutions of integral and integro-differential equations with fading memory kernels. Over long time intervals the errors in numerical schemes may become so large that they mask some important properties of the solution. One frequently appeals to stability theory to address this weakness, but it turns out that, in some of the model equations we have considered, there remains a gap in the analysis.
We consider a linear problem of the form

$$y'(t) = -\int_0^t e^{-\lambda(t-s)} y(s) ds, \qquad y(0) = 1,$$

and we solve the equation using simple numerical schemes. We outline the known stability behaviour of the problem and derive the values of λ at which the true solution bifurcates. We give the corresponding analysis for the discrete schemes and highlight that, for particular stepsizes, the methods give unexpected behaviour and we show that, as the step size of the numerical scheme decreases, the bifurcation points tend towards those of the continuous scheme. We illustrate our results with some numerical examples.

1 Introduction

The qualitative behaviour of numerical approximations to solutions of functional differential equations is an important area for analysis. We aim to investigate whether the behaviour of the numerical solution reflects accurately that of the true solution. We are particularly concerned with the behaviour of the solution over long time periods when (in particular) the convergence *order* of the method gives us limited insight, since the error depends on a constant that grows with the time interval. Many authors are concerned with stability of solutions and of their numerical approximations. We have considered elsewhere (see [7]) the stability of numerical solutions of equations of this type (and of non-linear extensions). This analysis raised a number of questions, which we consider here, about just how well the full range of qualitative behaviour of even quite a simple equation is understood.

Bifurcations (by which we shall mean any change in the qualitative behaviour of solutions) frequently arise only for systems or for higher order problems and therefore

one is particularly interested in finding suitable simple equations as the basis for analysis. In this paper, we consider the solution by numerical techniques of the integro-differential equation

$$y'(t) = -\int_0^t e^{-\lambda(t-s)} y(s)ds, \; y(0) = 1. \tag{1.1}$$

The equation is a linear convolution equation with separable fading memory convolution kernel and therefore is a simple example from an important class of problems familiar in applications. It is also possible to analyse the equation in the form of a second order ordinary differential equation.

The equation has several key properties that make it an ideal basis for our analysis:

1. it depends on the value of the *single* parameter λ,
2. when λ varies through real values, four distinctive qualitative behaviours in the solution can be detected, and
3. equations with exponential convolution kernels frequently arise in applications and elsewhere in the literature.

For λ real and positive, the kernel is of fading memory type. For λ real and negative, the kernel has a growing memory effect. This linear equation displays surprisingly rich dynamical behaviour for real values of the parameter λ and it is this behaviour that we want to consider for the numerical scheme. We note that the classical test equation

$$y'(t) = g(t) + \xi y(t) + \eta \int_0^t y(s)ds, \quad \eta \neq 0 \tag{1.2}$$

([1, 2]) displays the same range of qualitative behaviour possibilities as (1.1) for varying values of the *two* real parameters ξ, η.

This motivates us to consider equation (1.1) as a prototype problem that is interesting in its own right and that will also provide insight into the behaviour of more complicated equations. We propose to give a further analysis, where we consider the boundaries along which bifurcations occur for equation (1.2) in a sequel [3].

We consider the following questions.

1. Does the numerical scheme display the same four qualitatively different types of long term behaviour as are found in the true solution?
2. Are the interval ranges for the parameter λ that give rise to the changes in behaviour of the solution the same as in the original problem?

2 Behaviour of the exact solution

We consider the equation (1.1) which can be shown to have a unique continuous solution (see, for example, [10]). One can easily establish (by considering, for example, an equivalent ordinary differential equation) the general solution

$$y(t) = Ae^{\frac{-\lambda+\sqrt{\lambda^2-4}}{2}t} + Be^{\frac{-\lambda-\sqrt{\lambda^2-4}}{2}t} \tag{2.1}$$

where A, B are constants. For real values of λ the solution to (1.1) bifurcates (or changes qualitative behaviour) at $\lambda = 0, \pm 2$. We have the following qualitative behaviour.

A1. When $\lambda \geq 2$, $y \to 0$ as $t \to \infty$, with no oscillations.
A2. When $0 < \lambda < 2$, $y \to 0$ as $t \to \infty$, with infinitely many oscillations.
A3. When $\lambda = 0$, $y(t) = \cos(t)$ (persistent oscillations).
A4. When $-2 < \lambda < 0$, the solutions contain infinitely many oscillations of increasing amplitude.
A5. When $\lambda \leq -2$, the solution grows (in magnitude) without any oscillations.

3 Numerical analysis

To apply a numerical method to an integro-differential equation of the type

$$y'(t) = f\left(t, y(t), \int_0^t k(t, s, y(s))ds\right), \quad y(0) = y_0, \tag{3.1}$$

we write the problem in the form

$$y'(t) = f(t, y(t), z(t)) \tag{3.2}$$

$$z(t) = \int_0^t k(t, s, y(s))ds. \tag{3.3}$$

We solve (3.2), (3.3) numerically using a linear multistep method for solving equation (3.2) combined with a suitable quadrature rule for deriving approximate values of z from equation (3.3) (see [2]). Such a method is sometimes known as a *DQ-method*. For linear k-step methods, one also needs to provide a special starting procedure to generate the additional $k-1$ initial approximations to the solution that are not given in the equation but are needed by the multistep method on its first application. It turns out that one needs to choose the quadrature, multi-step method and starting schemes carefully to ensure that the resulting method is of an appropriate order of accuracy for the work involved. One should try to choose schemes of the same orders as one another since the order of the overall method is equal to the lowest of the orders of the three separate methods (the multistep formula, the starting value scheme and the quadrature) used to construct it. In this paper we have chosen to focus on one-step methods. There are two reasons for this: we have thereby avoided the need to construct special starting procedures which would make our analysis more complicated; as Wolkenfelt showed in [11], methods with a *repetition factor* of 1 (such as the ones we consider) are always stable and we also draw attention (see [9] for example), to the fact that the trapezoidal rule is an A-stable 1-step method.

For a well-behaved numerical scheme for (3.2), (3.3), we would anticipate four intervals (as with the continuous problem) of λ-values where the solutions to the discrete scheme behave qualitatively differently. However we know from investigation of bifurcation points for numerical solution of delay differential equations (see [12]) and indeed from stability analysis of integro-differential equations, that the points at which the qualitative behaviour of the solution changes may arise at the *wrong* values of the parameter. Based on previous experience (see [6]) we would expect this difference to be dependent upon the stepsize h of the numerical method and on the choice of method itself. Furthermore (see, for example [8], [12]), one might expect the bifurcation points of the discrete

scheme to approach the bifurcation points of the continuous problem as $h \to 0$ and one could anticipate that, for a method of overall order p, the approximation of the true bifurcation point by the bifurcation point of the numerical scheme would also be to $\mathcal{O}(h^p)$. We will show in this paper that (for $h \to 0$) the approximation of the bifurcation points in the methods we have chosen is at least to the order of the method.

To keep the analysis reasonably simple, we consider the following discrete form of (3.2). We use a linear θ-method in each case so that we solve the system

$$\begin{aligned} y_{n+1} &= y_n + h(\theta_1 F_n + (1-\theta_1)F_{n+1}),\ n = 0, 1, \ldots, && (3.4)\\ F_n &= f(nh, y_n, z_n), && (3.5)\\ z_n &= h\left(\theta_2 k(nh, 0, y_0) + \sum_{j=1}^{n-1} k(nh, jh, y_j) + (1-\theta_2)k(nh, nh, y_n)\right). && (3.6) \end{aligned}$$

One could choose any combination of $\theta_i, 0 \leq \theta_i \leq 1$ and a natural choice could be $\theta_1 = \theta_2$. However, in order to start with a simple method where the algebraic problem is tractable we have considered first the cases where $\theta_1 = 0$ and we consider a range of values of θ_2.

One solves equations of the form

$$y_{n+1} - y_n = -h^2\left(\theta_2 e^{-\lambda(n+1)h} y_0 + \sum_{j=1}^{n} e^{-\lambda h(n+1-j)} y_j + (1-\theta_2) y_{n+1}\right),\ y_0 = y_1 = 1. \tag{3.7}$$

Note that we have used a simple procedure to find the additional starting value $y_1 = 1$. We have observed from the integro-differential equation that $y'(0) = 0$ and have deduced that $y(h) = y(0)$ will provide a reasonable order 1 starting approximation. This choice of formula implies that we are combining a backward Euler scheme to discretise the differential equation, with, respectively, (for $\theta_2 = 1$) the forward rectangular (Euler) rule, (for $\theta_2 = \frac{1}{2}$) the trapezoidal rule and (for $\theta_2 = 0$) the backward rectangular rule for the quadrature. We will return to consider other combinations of θ_1, θ_2 later.

The equation (3.7) is equivalent to

$$\left(1 + h^2(1-\theta_2)\right) y_{n+2} + \left(h^2\theta_2 e^{-\lambda h} - 1 - e^{-\lambda h}\right) y_{n+1} + e^{-\lambda h} y_n = 0. \tag{3.8}$$

The behaviour of the solution as $t \to \infty$ depends on the roots of the characteristic equation

$$\left(1 + h^2(1-\theta_2)\right) k^2 + \left(h^2\theta_2 e^{-\lambda h} - 1 - e^{-\lambda h}\right) k + e^{-\lambda h} = 0. \tag{3.9}$$

Any solution of (3.8) will be asymptotically stable if both roots of (3.9) are of magnitude less than one and unstable if either root of (3.9) has magnitude greater than one. The solutions will contain (stable or unstable) oscillations when the roots of (3.9) are complex or, indeed, when at least one root is negative. It follows from this (see [4]) that the bifurcations occur as follows (for reasonably small $h > 0$).

B1. When $\lambda \geq -\frac{1}{h}\ln\left(\frac{1+2h^2-h^2\theta_2-2\sqrt{-h^2(h^2\theta_2-1-h^2)}}{h^4\theta_2^2-2h^2\theta_2+1}\right)$, $y_n \to 0$ as $n \to \infty$ with no oscillations. This condition can be written in the simpler form

$$\lambda \geq \frac{1}{h}\ln\left(1+2h^2-h^2\theta_2+2\sqrt{-h^2\left(h^2\theta_2-1-h^2\right)}\right)$$

and we thank the anonymous referee for pointing out this simplification.

B2. When $\frac{1}{h}\ln\left(\frac{1}{1+h^2(1-\theta_2)}\right) < \lambda < \frac{1}{h}\ln\left(1+2h^2-h^2\theta_2+2\sqrt{-h^2\left(h^2\theta_2-1-h^2\right)}\right)$, $y_n \to 0$ as $n \to \infty$, with infinitely many oscillations.

B3. When $\lambda = \frac{1}{h}\ln\left(\frac{1}{1+h^2(1-\theta_2)}\right)$ we obtain persistent oscillations.

B4. When $\frac{1}{h}\ln\left(1+2h^2-h^2\theta_2-2\sqrt{-h^2\left(h^2\theta_2-1-h^2\right)}\right) < \lambda < \frac{1}{h}\ln\left(\frac{1}{1+h^2(1-\theta_2)}\right)$, the solutions contain infinitely many oscillations of increasing amplitude.

B5. When $\lambda \leq \frac{1}{h}\ln\left(1+2h^2-h^2\theta_2-2\sqrt{-h^2\left(h^2\theta_2-1-h^2\right)}\right)$, the solution grows (in magnitude) without any oscillations.

4 Bifurcation points of the numerical scheme as approximations to true bifurcation points

We consider now the way in which the bifurcation points of the discrete scheme approximate those of the original problem. We are using a numerical scheme of order 1.

First we consider the value of $\lambda_1 = \frac{1}{h}\ln\left(1+2h^2-h^2\theta_2+2\sqrt{-h^2\left(h^2\theta_2-1-h^2\right)}\right)$ as θ_2 varies and $h \to 0$. It is easy to see that, as $h \to 0$, the value λ_1 satisfies $\lambda_1 \to 2$. In fact we can give greater precision to this. We can show that $\lambda_1 = 2-\theta_2 h+\mathcal{O}(\epsilon)$ as $h \to 0$. This means that, for θ methods in general, the approximation by our scheme approximates the true value (-2) to order 1 (the order of the method), as $h \to 0$. In the particular case $\theta_2 = 0$ the approximation is to order 2.

For $\lambda_2 = \frac{1}{h}\ln\left(\frac{1}{1+h^2(1-\theta_2)}\right)$ it is straightforward to show that stability is lost at a value of λ that approximates the true value (0) to order 1 in general. In fact, for $\theta_2 = 1$, the forward Euler scheme, the approximation is exact for all values of h.

The analysis of $\lambda_3 = \frac{1}{h}\ln\left(1+2h^2-h^2\theta_2-2\sqrt{-h^2\left(h^2\theta_2-1-h^2\right)}\right)$ follows in exactly the same way as for λ_1 and leads to an identical conclusion: the approximation of the bifurcation point $\lambda = -2$ is in general to order 1 as $h \to 0$ and to order 2 if $\theta_2 = 0$.

We illustrate our results graphically. Each of the plots shown in Figure 1 illustrate, for varying h, the ranges for the parameter λ where

1. the solutions are unstable due to at least one real root greater than unity in magnitude (the darkest region in the figures) (exponential growth if the root is positive, growing oscillations if the root is negative),
2. the solutions are unstable due to growing oscillations (the next darkest region in the figures),
3. the solutions are stable with asymptotically stable oscillations (the lightest region in the figure), and

4. the solutions are stable with exponentially stable decay.

We can compare with the right hand plot in Figure 2 which shows the true regions for the original problem and we can make the following observations.

1. As $h \to 0$ the values of λ at which changes in the behaviour occur approach the true values. This coincides with our previous experience in delay equations (see [8]).
2. There is some extremely surprising behaviour for some values of $h > 0$.
   (a) For the two values $\theta_2 = 0.5$ and $\theta_2 = 1$ we can see that the darkest region is in two parts: in the upper part there is a negative real root of magnitude greater than unity leading to exponentially growing oscillations in the solution; in the lower part there is a positive real root of modulus greater than unity leading to exponential growth in the solutions.
   (b) There can be a critical value of $h > 0$ ($h = \frac{1}{\sqrt{\theta_2}}$ when $\theta_2 > 0$) at which, for apparently arbitrarily large $\lambda < 0$ the numerical solution displays oscillatory behaviour.
   (c) There can be an additional thin region (visible only in larger scale versions of the plots) between the darkest and lightest regions in which there is a real negative root of magnitude less than unity leading to decaying oscillations.
   (d) For $\theta_2 = 0.5$ and $\theta_2 = 1$ the upper part of the darkest region indicates some really strange behaviour: spurious oscillations may arise for arbitrarily large negative values of λ and even (see figure 1) for some positive values of λ. Thus we can have the situation (for example for λ small and positive) where the true solution tends to zero while the approximate solution exhibits oscillations of growing magnitude. Alternatively, (for λ large and negative) the true solution could exhibit high index exponential growth while the approximate solution exhibits oscillations. We draw attention also to the fact that, for $\theta_2 = 0.5$ and $\theta_2 = 1$ the stability boundary of the method is made up of parts of the boundaries of two regions, making the prediction of behaviour for varying $h > 0$ particularly difficult.

We believe that these observations justify our view that more attention needs to be paid to changes in qualitative behaviour other than stability in reaching a good understanding of the behaviour of numerical methods for problems of this type.

We can consider next whether these observations are equally true for other choices of numerical method. We present in Figures 2 plots revealing the qualitative behaviour of solutions to equations (3.2), (3.3) with other choices of θ-method. It is easy to see that, even for combinations such as using the trapezium rule for both parts of the discretisation (a method characterised by $\theta_1 = \theta_2 = 0.5$ and known to do very well at preserving the stability boundary) there are problems in the preservation of other types of qualitative behaviour when h is not very small. Similarly, we can see that the choice $\theta_1 = \theta_2 = 1$ leads to a shrinking range (as h increases) for λ that lead to stable oscillatory solutions.

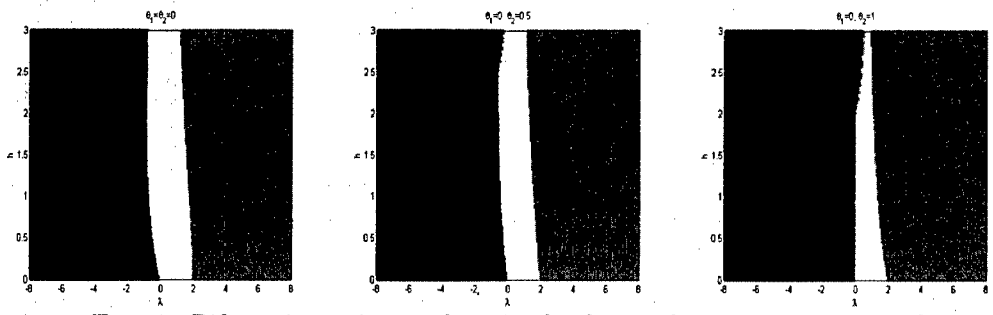


FIG. 1. Bifurcation points as h varies for $\theta_1 = 0, \theta_2 = 0, .5, 1$ respectively.

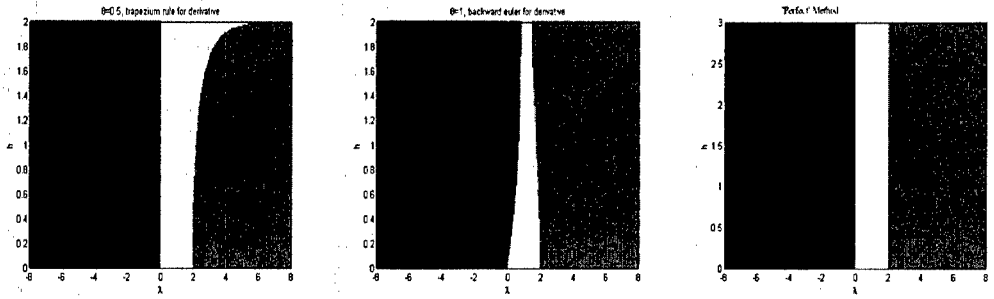


FIG. 2. Bifurcation points as h varies for, respectively, $\theta_1 = \theta_2 = 0.5, 1$ and for the analytical problem.

5 Alternative approaches

The particular equation we have considered can be formulated as an integro-differential equation, as an integral equation or as a second order differential equation. We have shown in [4] that the interesting and somewhat surprising observations about numerical behaviour that we made in the previous section also apply in these other formulations.

6 Closing remarks

The results presented in this paper show that the well-established stability theory based on the analysis of equation (1.2) gives only a very limited insight into the qualitative behaviour of solutions of the class of convolution equations with exponential memory kernel that we have considered here. We have observed elsewhere (see [5, 6, 7]) that the qualitative behaviour of numerical solutions to equations of this type may have surprising features and our consideration here of the prototype problem (1.1) illustrates how this unexpected behaviour may arise. We have seen in this paper how oscillations may arise in the numerical schemes when they should not, and how in other cases the numerical schemes may supress genuine oscillatory behaviour. When one seeks good methods based on a stability analysis, the desire is to focus on those methods where the step-length $h > 0$ is not subject to some upper bound to ensure the stability of the method. However our initial observations in this paper have shown that this may well prove an unreasonable

expectation when one is investigating these other changes in qualitative behaviour.

We believe that this paper introduces a range of worthwhile investigations in a field that is still quite open. Space restrictions have prevented us from considering the behaviour of more general methods in this paper and also from extending our analysis to consider other problems. The results we have presented here show that, for these simple methods at least, the bifurcation parameters are approximated in the numerical scheme to at least the order of the method, for sufficiently small $h > 0$. It is also very clear that, even for what appears to be a simple problem, the choice of numerical scheme and the form in which the problem is presented provide us with a rich source of example behaviour.

Bibliography

1. H. Brunner and J. Lambert, *Stability of numerical methods for Volterra integro-differential equations*, Computing, 12 (1974), pp. 75–89.
2. H. Brunner and P. J. van der Houwen, *The numerical solution of Volterra equations*, North-Holland, 1986.
3. J. T. Edwards, N. J. Ford, and J. A. Roberts, *Numerical approaches to bifurcations in solutions to integro-differential equations*, Proceedings of HERCMA, (2001).
4. ——, *The numerical simulation of an integro-differential equation with exponential memory kernel close to bifurcation points*, Tech. Rep. preprint, Manchester Centre for Computational Mathematics, (ISSN 1360 1725) 2001.
5. J. T. Edwards and J. A. Roberts, *On the existence of bounded solutions to a difference analogue for a nonlinear integro-differential equation*, International Journal of Applied Science and Computations, 6 (1999), pp. 55–60.
6. N. J. Ford, C. T. H. Baker, and J. A. Roberts, *Nonlinear Volterra integro-differential equations- stability and numerical stability of θ-methods*, Journal of Integral Equations and Applications, 10 (1998), pp. 397–416.
7. N. J. Ford, J. T. Edwards, J. A. Roberts, and L. E. Shaikhet, *Stability of a difference analogue for a nonlinear integro-differential equation of convolution type*, Tech. Rep. 312, Manchester Centre for Computational Mathematics, October (ISSN 1360 1725) 1997.
8. N. J. Ford and V. Wulf, *The use of boundary locus plots in the identification of bifurcation points in numerical approximation of delay differential equations*, Journal of Computational and Applied Mathematics, 111 (1999), pp. 153–162.
9. J. Lambert, *Numerical methods for ordinary differential systems*, Wiley, 1991.
10. P. Linz, *Analytical and Numerical Methods for Volterra Equations*, SIAM, 1985.
11. P. H. M. Wolkenfelt, *On the relation between the repetition factor and numerical stability of direct quadrature methods for second kind Volterra integral equations*, SIAM Journal on Numerical Analysis, 20 (1983), pp. 1049–1061.
12. V. Wulf, *Numerical analysis of delay differential equations undergoing a Hopf bifurcation*, PhD thesis, University of Liverpool, 1999.

Systems of delay equations with small solutions: a numerical approach

Neville J. Ford and Patricia M. Lumb

Chester College, Parkgate Road, Chester, CH1 4BJ, UK.
njford@chester.ac.uk, P.Lumb@chester.ac.uk

Abstract

We consider systems of delay differential equations of the form

$$y'(t) = A(t)y(t-1)$$

where $y \in \mathbb{R}^n$ and $A : \mathbb{R} \to \mathbb{R}^{n\times n}$. We investigate whether a numerical method can be used to determine whether or not the equation has so-called *small* solutions. Our work builds on recent analysis and experimental work completed in the scalar case and we are able to conclude that, at least when A is a suitable periodic matrix, one can predict small solutions by using a numerical approximation scheme of fixed step length.

1 Introduction and basic theory

The analysis of delay differential equations, both analytically and numerically, is well-established. One distinctive feature is that even a scalar delay differential equation is an infinite dimensional problem. For, if x satisfies

$$y'(t) = b(t)y(t-1) \tag{1.1}$$

the initial conditions that need to be specified take the form

$$y(t) = \varphi(t), \quad -1 \le t \le 0. \tag{1.2}$$

This infinite dimensionality has two significant implications for us:

(1) the dimension of a system of delay equations is *the same* as the dimension of a scalar delay equation, and

(2) the range of dynamical behaviour among solutions of delay equations is far wider than would be the case for ordinary differential equations.

In the present paper we are investigating an infinite dimensional property (that of possessing small solutions) where the analysis and results for systems needs to be presented quite separately from those for scalar equations because there are some interesting and distinctive features.

One way in which delay equations may be analysed is to view the solution operator as a dynamical system. The dimension of the dynamical system then inherits the infinite dimensionality of the delay equation itself. Small solutions (those that satisfy $x(t)e^{\alpha t} \to 0$

as $t \to \infty$ for all values of the parameter α) can arise in these infinite dimensional problems but would not be observed in finite dimensional equations. They are important because, when a delay equation has small solutions, the eigenfunctions and generalised eigenfunctions of the solution map do not form a complete set. This means that some standard analytical results do not hold and that particular care must be taken in solving and analysing the equation.

The easy detection of problems that have small solutions is still, in general, open, but we have seen [4, 5] that the use of a numerical approximation scheme can lead to good insights. Here we approximate the delay differential equation using a simple numerical scheme with fixed step length and then consider the spectrum of the resulting solution map.

In recent work (see, for example [3, 5]) the scalar case has been considered with some success. We have been able to see that, for the equation (1.1) with b periodic of period 1, we can detect the existence of small solutions by exploring the (finitely many) eigenvalues of the numerical scheme. We also found that it was not necessary to use a sophisticated numerical scheme for the investigation and this has justified us in focussing on the trapezium rule as the numerical method in this paper.

For the scalar case (1.1) it is known (see for example [4, 5]) that, when b satisfies the periodicity condition $b(t) = b(t-1)$, then non-trivial small solutions arise if and only if the function b changes sign. For the vector-valued case we can give a theorem, recently proved by Verduyn Lunel ([11]).

Theorem 1.1 *Consider the equation*

$$y'(t) = A(t)y(t-1), \text{ where } A(t) = A(t-1), \tag{1.3}$$

and where $y \in \mathbb{R}^n$. The equation has small solutions if and only if at least one of the eigenvalues λ_i satisfies, for some $\hat{t}$,

$$\Re\lambda_i(\hat{t}-) \times \Re\lambda_i(\hat{t}+) < 0, \lambda_i(\hat{t}) = 0. \tag{1.4}$$

Remark 1.2 *We shall describe the property (1.4) using the words an eigenvalue passes through the origin. We note that, even for real matrices A, the eigenvalues may be complex and it could be that a pair of complex conjugate eigenvalues will cross the $y-$axis away from the origin. In this case the equation has small solutions only if there is some other crossing of the $y-$axis by an eigenvalue where the crossing does take place at the origin.*

2 Numerical methods and systems of order two

All the important relevant features of systems of delay equations turn out to be exhibited in systems of two equations and so we shall focus on these for simplicity. We consider the equation

$$y'(t) = A(t)y(t-1) \quad \text{for} \quad A \in \mathbb{R}^{2\times 2} \quad \text{and} \quad y \in \mathbb{R}^2. \tag{2.1}$$

subject to $y(t) = \varphi(t)$ for $-1 \le t \le 0$ and we assume that $A(t) = A(t-1)$ for all t.

We introduce

$$y(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}, \quad A(t) = \begin{pmatrix} \alpha(t) & \beta(t) \\ \gamma(t) & \delta(t) \end{pmatrix}, \quad \varphi(t) = \begin{pmatrix} \varphi_1(t) \\ \varphi_2(t) \end{pmatrix}. \tag{2.2}$$

We apply the trapezium rule with step length $h = \frac{1}{N}$ and introduce the approximations $x_{1,j} \approx x_1(jh)$, and $x_{2,j} \approx x_2(jh), j > 0$; $x_{1,j} = \varphi_1(jh), x_{2,j} = \varphi_2(jh), -N \leq j \leq 0$. Set

$$y_n = \left(\, x_{1,n}, x_{1,n-1}, \ldots, x_{1,n-N}, x_{2,n}, x_{2,n-1}, \ldots, x_{2,n-N} \, \right)^T. \tag{2.3}$$

We note that, as in the one-dimensional case (see [3, 4, 5]), we can write the numerical scheme as $y_{n+1} = A(n)y_n$, where the matrix $A(n)$ now takes the form

$$A(n) = \begin{pmatrix}
1 & 0 & \ldots & 0 & \frac{h}{2}\alpha_{n+1} & \frac{h}{2}\alpha_n & 0 & \ldots & \ldots & 0 & \frac{h}{2}\beta_{n+1} & \frac{h}{2}\beta_n \\
1 & 0 & \ldots & \ldots & \ldots & 0 & 0 & \ldots & \ldots & \ldots & \ldots & 0 \\
0 & 1 & \ddots & & & \vdots & \vdots & & & & & \vdots \\
\vdots & \ddots & \ddots & \ddots & & \vdots & \vdots & & & & & \vdots \\
\vdots & & \ddots & \ddots & \ddots & \vdots & \vdots & & & & & \vdots \\
0 & \ldots & \ldots & 0 & 1 & 0 & 0 & \ldots & \ldots & \ldots & \ldots & 0 \\
0 & \ldots & \ldots & 0 & \frac{h}{2}\gamma_{n+1} & \frac{h}{2}\gamma_n & 1 & 0 & \ldots & 0 & \frac{h}{2}\delta_{n+1} & \frac{h}{2}\delta_n \\
0 & \ldots & \ldots & \ldots & \ldots & 0 & 1 & 0 & \ldots & \ldots & \ldots & 0 \\
\vdots & & & & & \vdots & 0 & 1 & \ddots & & & \vdots \\
\vdots & & & & & \vdots & \vdots & \ddots & \ddots & \ddots & & \vdots \\
\vdots & & & & & \vdots & \vdots & & \ddots & \ddots & \ddots & \vdots \\
0 & \ldots & \ldots & \ldots & \ldots & 0 & 0 & \ldots & \ldots & 0 & 1 & 0
\end{pmatrix}. \tag{2.4}$$

The sequence of matrices $\{A(n)\}$ is periodic, of period N (since the function A is periodic of period 1) and $y_2 = A(1)y_1, y_3 = A(2)A(1)y_1$ and so on. Therefore $y_{N+1} = Cy_1$ where $C = A(N)A(N-1).\cdots.A(2)A(1)$.

Remark 2.1 *The key to extending our discussion to larger systems, and indeed, to gaining a full understanding of the approach, is to note that in both the matrix $A(n)$ and the matrix C the original block structure is retained. Therefore although the matrices $A(n)$ and C are considerably larger than the original 2×2 matrix $A(t)$ in the problem, they are made up of 4 blocks in a 2×2 formation. Indeed the contents of each block is completely determined by our numerical method (the trapezium rule) and the values of the corresponding function, respectively $\alpha, \beta, \gamma, \delta$. There is no pollution of the blocks from the neighbouring functions.*

We consider three different cases:

(1) $\beta(t) = \gamma(t) = 0$ so that the matrix A is diagonal,
(2) either $\beta(t) = 0$ or $\gamma(t) = 0$ so that the matrix A is triangular, and

(3) the matrix A is neither diagonal nor triangular.

The first two cases can be dealt with quite quickly because of the fact that real diagonal and triangular matrices have only real eigenvalues and these eigenvalues lie on the diagonal. Therefore in these two cases we need consider only the question of whether the eigenvalues pass through zero; we do not need to concern ourselves with possible complex eigenvalues whose real parts change sign away from the origin.

We can go further: a diagonal matrix A leads to a block diagonal matrix $A(n)$ (with non-zero blocks top left and bottom right). Now by simple matrix theory we know that the eigenvalues of such a matrix are simply the union of the eigenvalues of the two blocks. A similar argument applies when there is a triangular matrix A because the matrices $A(n)$ are then block triangular. It follows that, for both of cases 1 and 2, the 2−dimensional eigenvalue problem simply reduces to two 1−dimensional problems. Therefore, when we consider the eigenspectra of the numerical schemes in cases 1 and 2, we expect the result to be the superposition of the eigenspectra from the two block matrices on the diagonal of C.

Case 3 is more complicated and we shall return to it after we give brief examples of Cases 1 and 2.

3 How to recognise small solutions: our previous work

Space restrictions here prevent us from giving a great many details of our previous work, but we provide a summary to show how the current investigation builds on the scalar case. In [3] we considered the eigenspectra of the matrix C. We showed that there were three characteristic patterns for the eigenspectra, represented by Figure 1. We take the presence of the closed loops that cross the x-axis to be characteristic of the cases where small solutions arise.

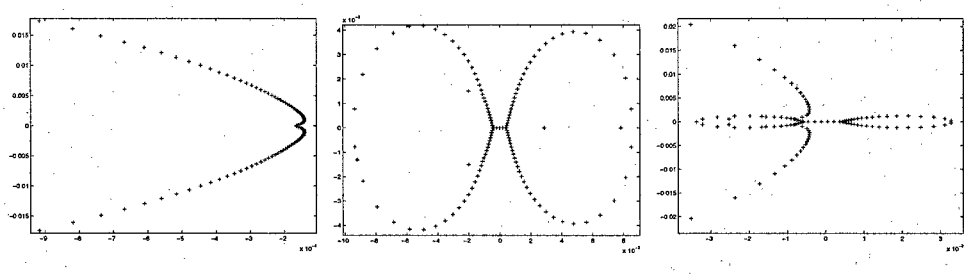


FIG. 1. Eigenspectra where $b(t)$ has no change of sign on $[0,1]$ (left), where $b(t)$ has a change of sign on $[0,1]$ and $\int_0^1 b(s)ds = 0$ (centre), and where $b(t)$ has a change of sign on $[0,1]$ and $\int_0^1 b(s)ds \neq 0$ (right).

4 The cases when $\beta(t) = 0$ and/or $\gamma(t) = 0$

As we have remarked already, the eigenspectrum when A is diagonal or triangular is just the same as the eigenspectra of the block matrices from the diagonal of C. We expect to

find the eigenspectra superimposed, which is indeed what we see in the examples given. Here we assume that at least one of $\gamma(t)$ or $\beta(t)$ is zero; the plots are then independent of the values taken by the other.

Example 4.1 We solve (2.1) with the choice $\alpha(t) = \sin 2\pi t + 1.4$ and $\delta(t) = \sin 2\pi t + 0.5$. Here α does not change sign but δ does change sign. We expect small solutions and Figure 2 provides confirmation.

Example 4.2 Now we solve (2.1) with $\alpha(t) = \sin 2\pi t$ and $\delta(t) = \begin{cases} -0.3 & \text{for } t \in (0, \frac{1}{2}], \\ 0.7 & \text{for } t \in (\frac{1}{2}, 1]. \end{cases}$ This time both α and δ change sign and we expect small solutions (see Figure 2).

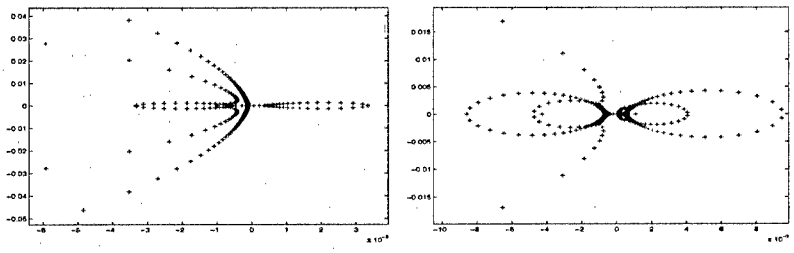


FIG. 2. Eigenspectra for Example 4.1 (left) and Example 4.2 (right).

4.1 The general two dimensional case

We now move on to consider the case when neither of $\beta(t), \gamma(t)$ is identically zero. In this situation the eigenvalues of $A(t)$ can be complex and so may cross the y-axis away from the origin.

First, we recall that $\det(A)$ is the product of the eigenvalues of A so that, by Theorem 1.1, it follows that $\det(A) = 0$ is a necessary condition for small solutions. However this condition cannot be used to characterise equations where small solutions arise; if the eigenvalues of A are real and one passes through the origin, then $\det(A)$ will change sign. If the eigenvalues of A are a complex conjugate pair and cross the y-axis at the origin then $\det(A)$ will instantaneously take the value zero but will otherwise remain positive (the same behaviour as when a real eigenvalue becomes zero but does not change sign). Therefore one cannot expect a change of sign in $\det(A)$ whenever there are small solutions. The fact that the trace of A is the sum of the eigenvalues of A can be used to characterise this case.

We summarise. For a real matrix A:

(1) if $\det(A)$ changes sign then there are small solutions,
(2) if $\det(A)$ becomes zero instantaneously and $\text{trace}(A)$ simultaneously changes sign then there are small solutions,
(3) if $\det(A)$ becomes zero instantaneously and $\text{trace}(A)$ does not simultaneously change sign then there are no small solutions indicated.

Example 4.3 We first consider the case when the matrix A takes the form

$$A(t) = \begin{pmatrix} \sin 2\pi t + a & \sin 2\pi t + b \\ \sin 2\pi t + c & \sin 2\pi t + d \end{pmatrix}.$$

By judicious choice of the constants a, b, c, d one can produce different types of behaviour. One can see that $|A(t)| = (a + d - b - c)\sin 2\pi t + (ad - bc)$. We will illustrate with the following choices of the constants

Case 1: $a = 1.5, b = 0.7, c = 0.5, d = 0.5$ where the determinant changes sign,
Case 2: $a = -2, b = 0.8, c = 1.8, d = 0.7$ where, again, the determinant changes sign,
Case 3: $a = 1.6, b = 0.8, c = 1.8, d = 0.7$ where the determinant never becomes zero.

From the plots for cases 1 and 2, we can easily see the presence of small solutions in the eigenspectra shown in Figure 3. In the Case 3, the eigenspectra in Figure 3 indicate that, as expected, no small solutions are present.

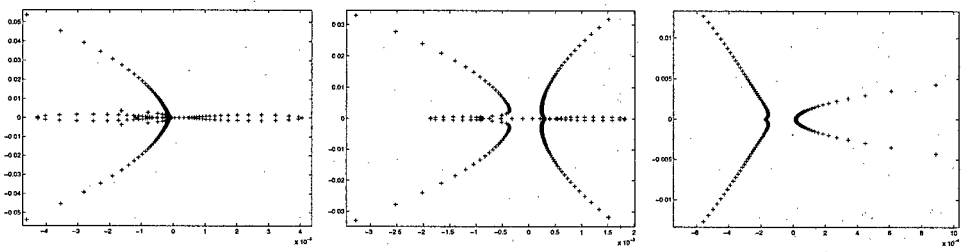


FIG. 3. Case 1. Case 2. Case 3

Example 4.4 Next, we consider the case when the matrix A takes the form

$$A(t) = \begin{pmatrix} \sin 2\pi t & -(\sin 2\pi t + b) \\ \sin 2\pi t + b & \sin 2\pi t \end{pmatrix}.$$

We choose the constant b in the following ways

Case 4: $b = 0$ so that $\det(A)$ becomes instantaneously zero at the same value that $\text{trace}(A)$ changes sign and the complex eigenvalues of A cross the y-axis at the origin,
Case 5: $b = 0.05$ so that the complex eigenvalues of A cross the y-axis away from the origin.

Here we can see that the characteristic shapes we familiar from our earlier work are not reproduced and further investigation is called for. We remark that (in the zoomed versions) the eigenspectrum where small solutions arise passes through the origin. This property is reproduced also for all other examples that we have tried.

Example 4.5 Now we consider the case when the matrix A takes the form

$$A(t) = \begin{pmatrix} t & t + b \\ -t - b & t \end{pmatrix}$$

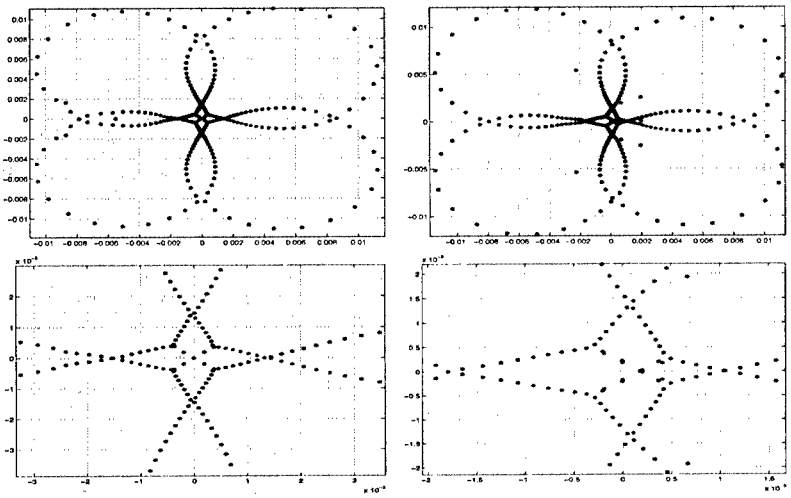


FIG. 4. Left: Case 4. Right: Case 5 and (below) zoomed versions.

for $t \in [-0.5, 0.5), A(t) = A(t-1)$ for $t \geq 0.5$ then it follows that A has complex eigenvalues that cross the $y-$axis at $y = b$ when $t = 0$. We plot the eigenspectra for

Case 6: $b = 0$ so the eigenvalues of A cross the y-axis at the origin,

Case 7: $b = 0.01$ so the eigenvalues of A cross the y-axis away from the origin.

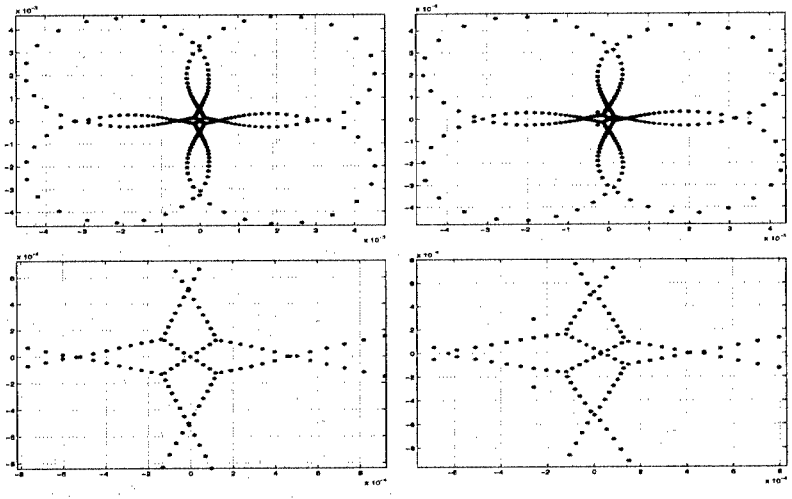


FIG. 5. Left: Case 6. Right: Case 7 and (below) zoomed versions

5 Conclusions

We have seen that it is easy to extend the detection of small solutions by numerical methods from one-dimensional to two-dimensional problems where the eigenvalues are real. Initial experiments indicate that the method works also for problems possessing complex eigenvalues, but here the patterns that arise in the eigenspecra plots are unfamiliar and require further investigation. However, based on our experimental evidence, it seems that small solutions arise in the latter case if and only if the eigenspectra plots pass through the origin.

Bibliography

1. O. Diekmann, S.A. van Gils, S. M. Verduyn Lunel, H.-O. Walther, *Delay Equations*, Springer Verlag, New York, 1995.
2. Y. A. Fiagbedzi, Characterization of Small Solutions in Functional Differential Equations, *Appl. Math. Lett.* **10** (1997), 97–102.
3. N. J. Ford, P. M. Lumb, Numerical approaches to delay equations with small solutions, *Proceedings of HERCMA 2001*, to appear.
4. N. J. Ford, S. M. Verduyn Lunel, Numerical approximation of delay differential equations with small solutions, *Proceedings of 16th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation*, Lausanne 2000, paper 173-3, New Brunswick, 2000 (ISBN 3-9522075-1-9).
5. N. J. Ford, S. M. Verduyn Lunel, Characterising small solutions in delay differential equations through numerical approximations, *Applied Mathematics and Computation*, to appear.
6. N. J. Ford, Numerical approximation of the characteristic values for a delay differential equation, *MCCM Numerical Analysis Report* No 350, Manchester University 1999 (ISSN 1360 1725).
7. J. K. Hale and S. M. Verduyn Lunel, *Introduction to Functional Differential Equations*, Springer Verlag, New York, 1993.
8. D. Henry, Small Solutions of Linear Autonomous Functional Differential Equations, *J. Differential Equations.* **8** (1970), 494–501.
9. S. M. Verduyn Lunel, A sharp version of Henry's theorem on small solutions, *J. Differential Equations.* **62** (1986), 266–274.
10. S. M. Verduyn Lunel, Series Expansions and Small Solutions for Volterra Equations of Convolution Type, *J. Differential Equations.* **85** (1990), 17–53.
11. S. M. Verduyn Lunel, private communication.

On an adaptive mesh algorithm with minimal distance control

Kamal Shanazari and Ke Chen

Department of Mathematical Sciences, The University of Liverpool, Liverpool L69 7ZL, UK.
{kamals, k.chen} @liv.ac.uk.

Abstract

In this paper, we present a new technique for generating error equidistributing meshes that satisfy both local quasi-uniformity and a preset minimal mesh spacing. This is firstly done in the one-dimensional case by extending the Kautsky and Nichols method [6] and then in the two-dimensional case by generalizing the tensor product methods to alternating curved line equidistributions. With the new meshing approach, we have achieved better accuracy in approximation using interpolatory radial basis functions (RBFs). Furthermore improved accuracy in numerical results have been obtained for a class of linear and non-homogeneous PDEs solved by the dual reciprocity method (DRM).

1 Introduction

The adaptive mesh algorithms have been widely used in the numerical solution of partial differential equations (PDEs) for boundary value problems [1, 13]. One undesirable feature of an error equidistributing mesh is that there is no guarantee of it being sufficiently smooth. For our applications of interpolation (using RBFs), the distance between points becoming too small can imply that the underlying interpolation matrix becomes ill-conditioned.

In this paper, we propose a method to deal with this problem in Section 2. Essentially our method consists of modifying the error monitor function in a suitable way and then equidistributing the new function so that the minimal mesh size constraint can be satisfied. We deal with the extension of adaptive mesh to two dimensions in Section 3. Finally, some numerical results will be given in Section 4.

2 An adaptive mesh with minimal mesh size control

In the 1D case, a typical adaptive mesh problem can be stated as follows: given a mesh (uniform or non-uniform) $t_0, t_1, \ldots, t_m$, and its corresponding error values (usually estimated from the numerical solution using a monitor function [5]) $f_0, f_1, \ldots, f_m$, we wish

Support of a studentship by the Ministry of Education (Iran) is gratefully acknowledged.

to find a new mesh

$$\Pi : x_0,\ x_1,\ \ldots,\ x_n, \tag{2.1}$$

that is locally bounded with respect to a positive constant $k \geq 1$ such that $1/k \leq h_j/h_{j-1} \leq k, j = 1, 2, \ldots, n-1$, $h_j = x_{j+1} - x_j$, while the errors are equidistributed on mesh Π. One solution to this problem was given in [6] by replacing f_j by $\hat{f}_j$ followed by a standard equidistribution algorithm. $\hat{f}_j$ is referred to as the padded function and the main idea of replacing f_j is increasing the values of the function f, where too small, to prevent considerably large mesh sizes. We now propose a method of further modifying $\hat{f}_j$ in such a way that the resulting equidistribution mesh satisfies the preset minimal mesh size h_{min}. Before proceeding, we consider replacing the piecewise linear function $\hat{f}(x)$ (with endpoint values $\hat{f}_j = \hat{f}(t_j)$) by another piecewise linear function $Z(x)$ (with endpoint values $Z_j \equiv \hat{f}(x_j)$). This is a technical approximation to simplify the presentation; actually the proposed method may work without this step. Note that if we were to equidistribute $Z(x)$, the resulting mesh would not differ from x_j much; define the average value of the monitor function as

$$d' = d'(Z) = \frac{1}{n}\sum_{j=0}^{n-1}(Z_j + Z_{j+1})\frac{h_j}{2}. \tag{2.2}$$

Our aim now is to modify some Z_j values so that the modified average value is the same as d' while the modified values ensure a preset minimal mesh size h_{min} is satisfied. To present our method, we note that insisting on $h_j \geq h_{min}$ implies $Z_j \leq \bar{Z}$ where

$$\bar{Z}h_{min} = d' \tag{2.3}$$

and $\bar{Z}$ is the critical constant to realize h_{min}. This points a way of modifying those large values of Z_j. However it is not obvious how to ensure the new and modified average values are the same, i.e. equidistribution is maintained for the same error constant. Suppose that among the current Z_j values, there are $M+1$ of them that are larger than $\bar{Z}$ (i.e. whose corresponding mesh size is less than h_{min}); denote these values by Z_{k_j} for $j = 0, 1, \ldots, M$. This means that $Z_{k_j} \leq \bar{Z}$ for $j = M+1, M+2, \ldots, n$. Here the sequence $k_0, k_1, \ldots, k_n$ represents a permutation of $0, 1, 2, \ldots, n$.

It turns out that a suitable modification (from Z_j to $\hat{Z}_j$) is the following:

$$\begin{cases} \text{(i)} \quad \hat{Z}_{k_j} = \bar{Z} \quad \text{when} \quad Z_{k_j} > \bar{Z}, \qquad \text{i.e. for } j = 0, 1, \ldots, M, \\ \text{(ii)} \quad \hat{Z}_{k_j} = Z_{k_j} + \dfrac{Z_{k_j}}{\displaystyle\sum_{l=M+1}^{n} Z_{k_l}} \left[\sum_{i=0}^{M}(Z_{k_i} - \bar{Z})\bar{h}_{k_i}\right] \Big/ \bar{h}_{k_j} \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{for} \quad j = M+1, M+2, \ldots, n, \end{cases} \tag{2.4}$$

where

$$\bar{h}_{k_i} = \begin{cases} (h_{k_i} + h_{k_i-1})/2 & \text{when} \quad k_i \neq 0, n, \\ h_0/2 & \text{when} \quad k_i = 0, \\ h_{n-1}/2 & \text{when} \quad k_i = n. \end{cases} \tag{2.5}$$

For a simple illustration, see the plot of Fig 3b. To prove that the above modification is suitable, we first present the following result for a simple case.

Theorem 2.1 *Let $x_0, x_1, \ldots, x_n$ be a non-uniform mesh with the mesh sizes $h_j = x_{j+1} - x_j$ and $Z_0, Z_1, \ldots, Z_n$ are the corresponding error values. If the critical constant value $\bar{Z}$ as in (2.3), and only one value $Z_1 > \bar{Z}$ (i.e. $M = 1$ and all others Z_j are less than or equal to $\bar{Z}$), the modification (2.4) takes the following form,*

$$\begin{cases} (i) \quad \hat{Z}_0 = Z_0, \qquad \hat{Z}_1 = \bar{Z}, \\ (ii) \quad \hat{Z}_j = Z_j + \dfrac{Z_j}{\sum_{i=2}^{n} Z_i} \left[(Z_1 - \bar{Z})(h_0 + h_1)/2\right] / (h_j + h_{j-1})/2 \text{ for } j = 2, 3, \ldots, n. \end{cases}$$

Then the average value $d = d(\hat{Z})$ of the modified values $\hat{Z}_j$ is the same as $d' = d'(Z)$ in (2.2).

Note $M = 1$ here; in fact the results holds for any one value $Z_j > \bar{Z}$. Now we are ready to present the main result on equation (2.4) with regard to minimal mesh size control.

Theorem 2.2 *With the error function modified as in (2.4), the new mesh $\hat{h}_j$ resulting from equidistribution satisfies (i) the average error value remains as d'; (ii) $\hat{h}_j \geq h_{min}$.*

Here h_{min} cannot be specified to be larger than $h = 1/n$ (the uniform mesh size); practically we found $h_{min} \in [h^2, h/2]$ is adequate. Full proofs to these results will be given in the full version of this paper [10].

In the method in (2.4), the values of Z_{k_j} which are less than but close to $\bar{Z}$ may become unnecessarily larger (e.g. larger than $\bar{Z}$) and therefore we can propose a further refinement. We can keep some of the Z_{k_j} values which are between $\bar{Z}/2$ and $\bar{Z}$. In other words, we only modify the very large and very small values of Z_{k_j} (see plot of Fig 3b). Then our theorems are still valid but the proofs may need minor changes. Finally we summarise our adaptive method with minimal mesh size control as follows (see the plot of Fig 3b for an illustration).

Algorithm 2.3. (Numerical algorithm) *For given non-uniform mesh $a = t_0, t_1, \ldots, t_m = b$, the error values $f_0, f_1, \ldots, f_m$, values c and h_{min}:*

(1) *Does the locally bounded mesh algorithm converge to the new mesh $a = x_0 < x_1 < \cdots < x_n = b$ which is sub-equidistributing with respect to c and f, that is, for a sufficiently large value of the integer n such that $\int_a^b f \leq nc$, and the inequalities*

$$\int_{x_j}^{x_{j+1}} f \leq c, \quad j = 0, 1, \ldots, n-1$$

are satisfied.

(2) *Check the minimal mesh size and compare it with the h_{min}. If it is less than h_{min}, go to the Step 3 otherwise stop.*

(3) *Approximate the padding values $Z_j = \hat{f}(x_j)$ corresponding to the new mesh by using piecewise linear interpolation of $\hat{f}_i$ values and calculate the average value*

$$d = \frac{1}{n} \sum_{j=0}^{n-1} (Z_j + Z_{j+1}) \frac{h_j}{2}, \quad \text{where} \quad h_j = x_{j+1} - x_j,$$

and $\bar{Z}$ *according to* $\bar{Z}h_{min} = d$.

(4) *Obtain the decreasing arrangement of* Z_j, Z_{k_j} *by ordering them.*

(5) *Modify the* Z_{k_j} *values as follows,*

$$\begin{cases} \textit{(i)} \quad \hat{Z}_{k_j} = \bar{Z} \quad \textit{when} \quad Z_{k_j} > \bar{Z}, \\ \\ \qquad \textit{assuming, that for} \quad j = 0, 1, \ldots, M \quad Z_{k_j} > \bar{Z}, \\ \textit{(ii)} \quad \hat{Z}_{k_j} = Z_{k_j} \quad \textit{when} \quad \bar{Z}/2 \le Z_{k_j} \le \bar{Z}, \\ \\ \qquad \textit{assuming, that for} \quad j = M+1, M+2, \ldots, N, \qquad \bar{Z}/2 \le Z_{k_j} \le \bar{Z}, \\ \textit{(iii)} \quad \hat{Z}_{k_j} = Z_{k_j} + \frac{Z_{k_j}}{\sum_{l=N+1}^{n} Z_{k_l}} \left[\sum_{i=0}^{M} (Z_{k_i} - \bar{Z}) \bar{h}_{k_i} \right] / \bar{h}_{k_j}, \\ \qquad\quad \textit{for} \quad j = N+1, N+2, \ldots, n, \end{cases}$$

where $\bar{h}_{k_i}$ *was introduced in (2.5).*

(6) *Check the modified values* $\hat{Z}_{k_j}$ *in the stage (iii) of the Step 5. If* $\hat{Z}_{k_j} \le \bar{Z}/2$ *for all* j, *go to Step 7 otherwise repeat Step 5.*

(7) *Perform the equidistribution procedure for the modified values* $\hat{Z}_{k_j}$ *and obtain the new adapting mesh.*

3 Extension to two dimensions

The concept of adapting mesh in one dimension is well known (see e.g. [5, 3]). Extension of this idea to two dimensions is not straightforward. For a given function $f(x, y)$ and 2D domain Ω, an obvious extension is dividing the domain Ω into some subdomains Ω_i in such a way that

$$\int\int_{\Omega_i} f(x, y) = \text{constant}. \tag{3.1}$$

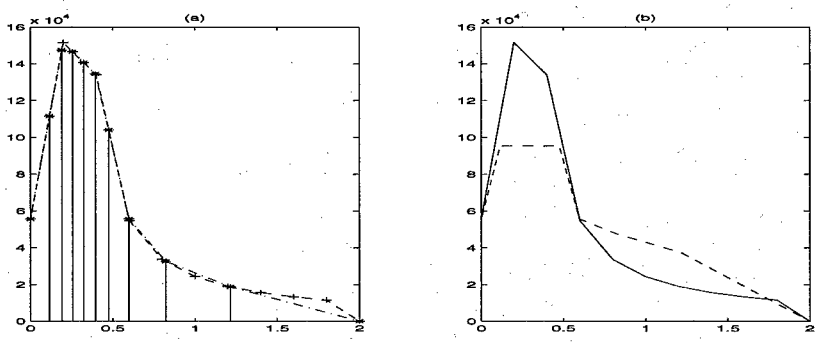


FIG. 1. In Fig (a) the monitor values corresponding to the new mesh are represented by '*', the linear interpolation for these values is shown by '-.' and in Fig (b) the modified values of the padded function, represented by dash line, are compared with the original values.

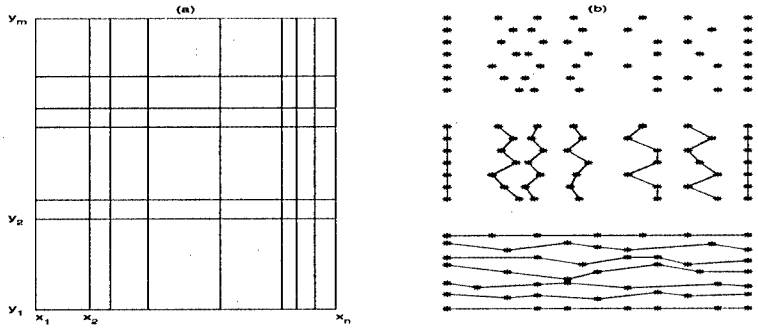


FIG. 2. In Fig (a) equidistribution of slabs in the two coordinate direction and in Fig (b) three stages of the new method are shown.

But, such a partition is not unique and furthermore satisfying condition (3.1) properly is not simple. Consequently, this condition has to be replaced. Among the methods given to satisfy the condition (3.1) as much as possible, two well known methods are transformation and dimension reduction. Transformation methods are based on mapping the physical domain into a simple domain with a uniform mesh and ultimately applying the equidistribution condition to obtain an adapting mesh in the physical domain [4, 12]. These methods are generally costly and complicated in theory. In this work we first consider the latter method which is easier and cheaper than the former method. We then present a new technique to generate a 2D mesh.

3.1 Dimensions reduction

We assume that Ω is a rectangle in the form $\Omega = \{(x,y),\ a \le x \le b,\ c \le y \le d\}$. A simple idea is to produce the mesh,

$$a = x_0 < x_1 < \ldots < x_{n-1} < x_n = b,$$

$$c = y_0 < y_1 < \ldots < y_{m-1} < y_m = d,$$

such that

$$\int_{x_i}^{x_{i+1}} \int_{y_0}^{y_m} f_x(x,y)\, dy\, dx = \text{constant}, \tag{3.2}$$

and

$$\int_{y_j}^{y_{j+1}} \int_{x_0}^{x_n} f_y(x,y)\, dx\, dy = \text{constant}, \tag{3.3}$$

where $f_x(x,y)$ and $f_y(x,y)$ are the monitors in the x and y directions respectively (see Fig 3.1a). Obviously the generated mesh by this method is much different from an equi-distributing mesh that one expects from (3.1). Another method which leads to a non-rectangular grid is dimensional splitting [11]. We now describe a new method of type dimension reduction.

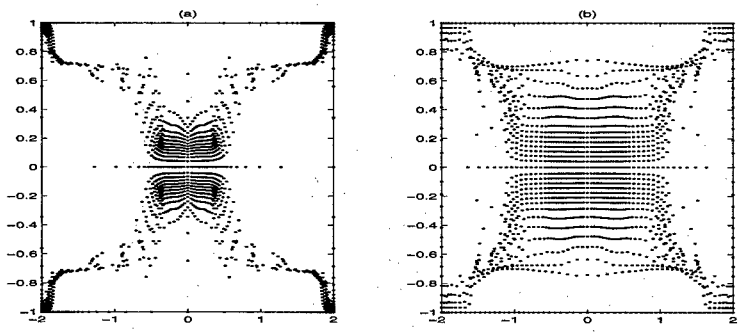


FIG. 3. In Fig (a) the mesh generated by the new method for function in (3.6) and in Fig (b) the resulting mesh when restricting the minimal mesh size as $h_{min} = h/2$ for the same function are shown.

3.2 A new approach for a 2D mesh

The idea is based on the tensor product method and therefore a non-rectangular grid. We start with a uniform mesh in a rectangular region Ω and perform the method in three stages. In the first stage, the error equidistributing is performed for each line in the horizontal direction (see the first part of Fig 3.1b), that is,

$$\int_{x_j}^{x_{j+1}} f_x(x, y_i)\, dx = \text{constant for } i = 0, 1, \ldots, m. \tag{3.4}$$

In the next stage, the mesh is redistributed in the vertical direction along the new grid lines (see the second part of Fig 3.1b), that is,

$$\int_{s_i}^{s_{i+1}} f_y(x_j, y)\, dy = \text{constant for } j = 0, 1, \ldots, n, \tag{3.5}$$

where $s_{i+1} - s_i$ is the distance between two consecutive points (x_j, y_i) and (x_j, y_{i+1}) along the new lines. In the final stage, equidistributing is repeated in the horizontal direction along the grid lines (the last part of Fig 3.1b). One can observe that repeating this procedure usually leads to a convergent mesh. According to our experiments, the number of iterations to achieve convergence is at most five. The resulting mesh by this procedure for function

$$u(x, y) = e^{\left(4 - x^2 - 4y^2\right)^2} \tag{3.6}$$

when applying the arc-length monitor is shown in Figure 3a. The idea of controlling the mesh size can also be applied in this technique. The generated mesh for the same function when the mesh sizes are restricted to $h_{min} = h/2$, where h is the mesh size in the case of uniform mesh, is given in Figure 3b.

4 Numerical examples

In this part the affect of adapting the mesh on the accuracy of interpolation and the DRM is considered. In the following examples, the infinity norm has been used to measure the

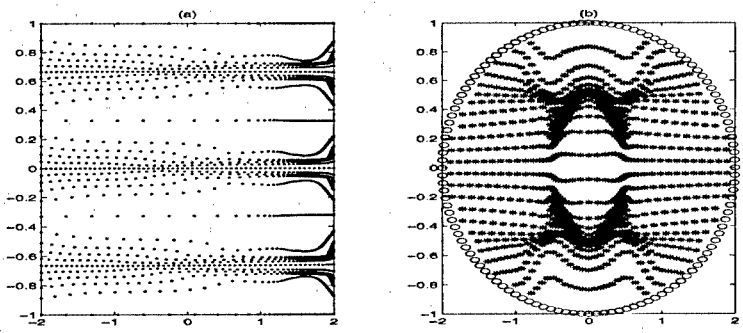


FIG. 4. The resulting mesh when using the new method for function in Examples 1 and 2 are shown in Figures (a) and (b) respectively.

Method	stage	Function (E1)	Derivative	Function (E2)	Derivative
uniform mesh	—	5.1E-2	9.5E-1	1.3E-2	2.2E-1
Adaptive mesh	first	5.4E-3	1.6E-1	2.5E-3	1.3E-2
with control	second	5.4E-3	3.0E-1	2.1E-3	1.0E-1
	third	3.8E-3	3.0E-1	3.7E-3	1.0E-1
Adaptive mesh	first	1.4E-2	9.9E-2	2.5E-3	1.5-2
without control	second	2.2E-2	7.5E-1	2.1E-3	1.0E-1
	third	1.8E-2	6.0E-1	4.5E-3	1.2E-1

TAB. 1. The interpolation error for Examples $1-2$ using adaptive mesh with and without control the mesh sizes.

accuracy, that is, if u and $\bar{u}$ are the exact and approximate values respectively then the error is calculated as

$$e_u = ||u(x) - \bar{u}(x)||_\infty = \max_{x \in D} |u(x) - \bar{u}(x)|.$$

A polynomial RBF, $1 + r^3$, has been employed in this work.

Example 4.1 We check the interpolation in terms of the RBFs for the function,

$$u(x, y) = \left(1 - e^{3x-3}\right) \sin(1.5\,\pi\, y), \tag{4.1}$$

in a rectangular domain. The generated mesh for this function is shown in Figure 4a.

Table 4 shows the affect of adapting mesh on the interpolation accuracy with and without controlling the mesh sizes. As one can observe, using the adapting mesh considerably improves the accuracy in comparison with the case of uniform mesh. Moreover, the result in the case of controlling the minimal mesh size is better.

Example 4.2 In this example we first check the function $f_2(x,y) = 0.5 - 0.5\tanh(-4 + 16x^2 + 16y^2)$ and then solve the linear PDE: $\nabla^2 u + y\frac{\partial u}{\partial x} + x\frac{\partial u}{\partial y} + xyu = d$, with the Dirichlet boundary condition over the elliptic domain $x^2 + 4y^2 = 4$, where d is a known function such that the exact solution is $u(x,y) = f_2(x,y)$.

Again from Table 4, we see improved approximation. We apply the DRM method [7] for solution, where the domain integrals are approximated by using RBF interpolation. The adaptive mesh for this function is given in Fig. 4b and has been observed to give rise to improved DRM solution.

5 Conclusions

We considered a new algorithm for producing a locally bounded mesh with a preset minimal mesh size. Such a mesh is used to overcome the ill-conditioning problems associated with radial basis function interpolation. Extension of the idea to the 2D case is also considered. Some preliminary and improved numerical results are given.

Bibliography

1. Ainsworth, M. and Oden, T. J., *A Posterior Error Estimation in Finite Element Analysis*, John Wiley, 2000.
2. Beckett, G., Mackenzie, J. A., Ramage, A. and Sloan, D. M., On the numerical solution of one-dimensional PDEs using adaptive methods based on equidistribution, *Journal of Computational Physics*, **167** (2), 372–392, 2001.
3. Carey, G. F. and Dinh, H. T., Grading Functions and Mesh Redistribution, *SIAM J. Numer. Anal.*, **22** (5), 1028–1040, 1985.
4. Chen, K., Two-Dimensional Adaptive Quadrilateral Mesh Generation, *Communications in Numerical Methods In Engineering*, **10**, 815–825, 1994.
5. Chen, K., Error Equidistribution And Mesh Adaptation, *SIAM J. Sci. Comput.*, **15**, No 4, 798–818, 1994.
6. Kautsky, J. and Nichols, N. K., Equidistributing Meshes With Constraints, *SIAM J. Sci. Statist. Comput.*, **1**, No 4, 449–511, 1980.
7. Partridge, P. W., Brebbia, C.A, and Wrobel, L. C., *The Dual Reciprocity Boundary Element Method*, Computational Mechanics Publications, 1992.
8. Pereyra V., and Sewell, E. G., Mesh selection for discrete solution of boundary problems in ordinary differential equations, *Numer. Math.*, **23**, 261–268, 1975.
9. Profit, A., Chen, K. and Amini, S., Application of the DRBEM with Adaptive Internal Points to Nonlinear Dopant Diffusion, *Proc. 2nd UK BIE conf.*, Brunel University Press, 1999.
10. Shanazari, K. and Chen, K., *On an adaptive mesh algorithm with minimal distance control for the dual reciprocity method.* In preparation.
11. Sweby, P. K., Data-Dependent Grids, *Numerical Analysis Report 7/87*, University of Reading, UK, 1987.
12. Thompson, J. F., Warsi, Z. U. A. and Mastin, C. W., *Numerical Grid Generation Foundations and Applications*, North-Holland, 1985.
13. White, A. B., On selection of equidistribution meshes for two-point boundary value problems, *SIAM J. Numer. Anal.*, **19**, 472–502, 1979.

An alternative approach for solving Maxwell equations

Wolfgang Sproessig

Freiberg University of Mining and Technology, Germany
sproessig@math.tu-freiberg.de

Ezio Venturino

Politecnico di Torino, Italia
egvv@calvino.polito.it

Abstract

At present the use of hypercomplex methods is pursued by a growing number of mathematicians, physicists and engineers. Quaternionic and Clifford calculus will be applied on wide classes of problems in very different fields of science. We explain Maxwell equations within the geometric algebras of real and complex quaternions. The connection between Maxwell equations and the Dirac equation will be elaborated. Using the Teodorescu transform we will deduce an iteration procedure for solving weak time-dependent Maxwell equations in isotropic homogeneous media. Assuming the so-called Drude-Born-Feodorov constitutive laws Maxwell equations in chiral media were deduced. Full time-dependent problems will be reduced to the consideration of Weyl operators.

1 Historical oriented introduction

Classical Maxwell equations were discovered in the second half of the nineteenth century as result of the stormy development of electromagnetic research in that time. The study of these equations has attracted generations of physicists and mathematicians but some of their secrets are still hidden.

At about the same time, also new algebraic structures were invented. W.R. Hamilton discovered in 1843 the algebra of real quaternions as a generalization of the field of complex numbers. Under the influence of H. Grassman's extension theory and Hamilton's quaternions, W.K. Clifford created in 1978 a geometric algebra, which is nowadays called Clifford algebra. Its construction starts with a basis in the signed $R^n = R^{p,q}$ with units $e_1, ..., e_n$. Assume that $e_i^2 = -1$, for $i = 1, ..., q$, and $e_j^2 = 1$, for $j = 1, ..., p$, as well as the anticommutator relation

$$e_i e_j + e_j e_i = 0$$

for $i \neq j$. Together with $e_0 = 1$ one can construct a basis in the 2^n-dimensional standard Clifford algebra $Cl_{p,q}$. Incidentally, in 1954 C. Chevalley [5] showed that each Clifford number, i.e. each element of $Cl_{p,q}$, can be identified with an antisymmetric tensor.

Let us go back to the electromagnetic field equations. Already J. C. Maxwell [15] himself and W. R. Hamilton [10] used these new algebraic techniques to try to simplify

Maxwell's equations. The aim was to obtain an equation of the type

$$Du + au = F$$

with suitable operators D and a. For this reason Hamilton introduced his "N'abla operator" as well as the notion "vector". The tendency of algebraisation of physics continued in the first half of the last century. A long list of important publications were devoted to this topic. We only stress here some of the milestones, beginning with the "Theory of Relativity" by L. Silberstein (1914)[18] , and H. Weyl's book "Raum-Zeit-Materie" of 1921. Important results of Einstein/Mayer, Lanczos and Proca followed. In 1935 this development highlighted with the thesis of M. Mercier (Genèva) [16]. After the reinvention of the concept of "spinors", firstly appeared in 1911 in a paper by E. Cartan, D. Hestenes [11, 12, 13], F. Bolinder [3] and M. Riesz [17] wrote fundamental algebra papers with applications in electromagnetic theory, using the framework of Clifford numbers and spinor spaces.

Meanwhile, in the late thirties the famous Swiss mathematician R. Fueter and his coworkers and followers used a function-theoretic approach for the same problems. These ideas were refreshed and fruitful extented by R. Delanghe and his group and A. Sudbury in the seventies and early eighties (cf. [4, 20]). Influenced by the success of complex analysis and Vekua theory a generalized operator theory with corresponding singular integral operators [19] and a corresponding hypercomplex theory for boundary value problems of elliptic partial differential equations were developed [8],[9].

Making use of a transformation of Maxwell's equations into a system of homogeneous coordinates we will propose an alternative solution method.

2 Maxwell equations

Let G be a bounded domain with sufficient smooth boundary Γ that is filled out with an isotropic homogeneous material.

Using Gauss units Maxwell equations read as follows:

$$\begin{aligned} c\,\mathrm{rot}\,H &= 4\pi J + \partial_t D && \text{(Biot-Savart-Ampere's law)} \\ c\,\mathrm{rot}\,E &= -\partial_t B && \text{(Faraday's law)} \\ \mathrm{div}\,D &= 4\pi\rho && \text{(Coulomb's law)} \\ \mathrm{div}\,B &= 0 && \text{(no free magnetic charge)} \end{aligned}$$

Furthermore, the continuity condition has to be fulfilled:

$$\mathrm{div}\,J = -\partial_t \rho,$$

where $E = E(t,x)$ is the electric field, $H = H(t,x)$ the magnetic field, $J = J(t,x)$ the electric current density, $D = D(t,x)$ the electric flux density, $B = B(t,x)$ the magnetic flux density, $\rho = \rho(t,x)$ the charge density, and c is the speed of light in a vacuum.

The relations between flux densities and the electric and magnetic fields depend on the material. It is well-known that for instance all organic materials contain carbon and

realize in this way some kind of optical activity. Therefore, Lord Kelvin introduced the notion of the chirality measure of a medium. This coefficient expresses the optical activity of the underlying material. The correspondent constitutive laws are the following:

$$D = \varepsilon\, E + \varepsilon\, \beta \,\mathrm{rot}\, E \qquad \text{(Drude-Born-Feodorov laws)},$$
$$B = \mu\, H + \mu\, \beta \,\mathrm{rot}\, H,$$

where $\varepsilon = \varepsilon(t,x)$ is the **electric permittivity**, $\mu = \mu(t,x)$ is the **magnetic permeability** and the coefficient β describes the **chirality measure** of the material. In isotropic cases one has the possibility to use the so-called **Tellegen representation**

$$D = \varepsilon\, E + \alpha\, H,$$
$$B = \mu\, H + \alpha^* E.$$

The connection between the electric field E and current density J is given by

$$J = \sigma\, E + \sigma\, g$$

where σ is the electric conductivity and g a given electric source.

Starting with $\beta = 0$ and replacing D and B by $D = \varepsilon\, E$ and $B = \mu\, H$ we get in the case of

$$\varepsilon = \varepsilon(x)\ ,\ \mu = \mu(x)$$

$$-\varepsilon \partial_t E + c\,\mathrm{rot}\, H = 4\pi\, J\ , \tag{2.1}$$
$$\mu \partial_t H + c\,\mathrm{rot}\, E = 0\ , \tag{2.2}$$
$$\varepsilon\,\mathrm{div}\, E = 4\pi\rho - (\nabla\varepsilon \cdot E)\ , \tag{2.3}$$
$$\mu\,\mathrm{div}\, H = -(\nabla\mu \cdot H)\ . \tag{2.4}$$

After summing (2.1) and (2.4) as well as (2.2) and (2.3) we obtain

$$-\varepsilon \partial_t E + c\,\mathrm{rot}\, H + \mu\,\mathrm{div}\, H = -(\nabla\mu \cdot H) + 4\pi\, J, \tag{2.5}$$
$$\mu \partial_t H + c\,\mathrm{rot}\, E + \varepsilon\,\mathrm{div}\, E = -(\nabla\varepsilon \cdot H) + 4\pi\,\rho. \tag{2.6}$$

In the case of ε, μ being constants we can introduce the new functions $\tilde{E}$, $\tilde{H}$ which are defined on a homogeneous space with a first coordinate x_0 and the other coordinates $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$. We obtain:

$$E(t,x) =: \tilde{E}\left(-\frac{1}{\varepsilon}\, t, \frac{1}{c}\, c\right),$$
$$H(t,x) =: \tilde{H}\left(\frac{1}{\mu}\, t, \frac{1}{c}\, c\right).$$

The equations (2.5)–(2.6) transform into

$$\partial_1 \tilde{E} + \mathrm{rot}\, \tilde{H} + \mu\, c\,\mathrm{div}\, \tilde{H} = 4\pi\, J,$$
$$\partial_1 \tilde{H} + \mathrm{rot}\, \tilde{E} + \varepsilon\, c\,\mathrm{div}\, \tilde{E} = 4\pi\,\rho.$$

3 Quaternionic representations

Let e_1, e_2, e_3 be the generating units of the algebra of real quaternions $\mathbb{H}$, which fulfil the conditions

$$e_i e_j + e_j e_i = -2\delta_{ij} \qquad (i,j = 1,2,3).$$

This leads to the following multiplication rule for two quaternions $u = u_0 + \underline{u}$, $v = v_0 + \underline{v}$:

$$uv = u_0 v_0 - \underline{u} \cdot \underline{v} + u_0 \underline{v} + v_0 \underline{u} + \underline{u} \times \underline{v} \quad (v_i \in \mathbb{R}),$$

where $\underline{u} = u_1 e_1 + u_2 e_2 + u_3 e_3$, $v = v_1 e_1 + v_2 e_2 + v_3 e_3$. Further, let $u = u_0 + \underline{u}$ be a quaternion. Then $\overline{u} = u_0 - \underline{u}$ is called to be its **conjugate quaternion**. The operator defined by

$$D = \partial_1 e_1 + \partial_2 e_2 + \partial_3 e_3$$

is called **Dirac operator**. It acts on a quaternionic valued function as follows:

$$Du = -\operatorname{div} \underline{u} + \operatorname{rot} \underline{u} + \operatorname{grad} u_0 .$$

With the multiplication operator m_θ

$$m_\theta u = \theta u_0 + \underline{u} \qquad (\theta \in \mathbb{R}^+),$$

with $u = u_0 + \underline{u}$, $\underline{u} = u_1 e_1 + u_2 e_2 + u_3 e_3$, we obtain

$$m_{\mu c}(\partial_1 \tilde{E} + D\tilde{H}) = 4\pi J ,$$
$$m_{\varepsilon c}(\partial_1 \tilde{H} + D\tilde{E}) = 4\pi \rho ,$$

and so

$$\begin{aligned} \partial_1 \tilde{E} + D\tilde{H} &= m_{\mu c}^{-1} 4\pi J , \\ \partial_1 \tilde{H} + D\tilde{E} &= m_{\varepsilon c}^{-1} 4\pi \rho . \end{aligned}$$

Finally, we get

$$\partial(\tilde{E} + \tilde{H}) = \partial_1(\tilde{E} + \tilde{H}) + D(\tilde{E} + \tilde{H}) = 4\pi(m_{\mu c}^{-1} J + m_{\varepsilon c}^{-1}\rho) =: F_1 ,$$
$$\overline{\partial}(\tilde{E} - \tilde{H}) = \partial_1(\tilde{E} - \tilde{H}) - D(\tilde{E} - \tilde{H}) = 4\pi(m_{\mu c}^{-1} J - m_{\varepsilon c}^{-1}\rho) =: F_2 ,$$

where ∂ is also called **Weyl operator** and $\overline{\partial}$ is the conjugate to ∂. By the way, a function u is called **quaternionic regular** if $\partial\, u = 0$ and **quaternionic anti-regular** if $\overline{\partial}\, u = 0$.

For simplifying we set: $\tilde{E} + \tilde{H} =: v$ and $\tilde{E} - \tilde{H} =: w$. Then it follows

$$\partial\, w = F_1(v,w), \tag{3.1}$$
$$\overline{\partial}\, w = F_2(v,w). \tag{3.2}$$

Let us have a closer look at the functions F_1, F_2. The electric current density J is given by

$$J = \sigma\, E + \sigma\, g,$$

where E and g are vector functions. This leads to the following simplification

$$F_1 = 4\pi\left[\sigma(\tilde{E}+g)+\frac{\rho}{\varepsilon c}\right] = 2\pi\left[\sigma(v+w)+\sigma g+\frac{\rho}{\varepsilon c}\right],$$
$$F_2 = 4\pi\left[\sigma(\tilde{E}+\sigma g)-\frac{\rho}{\varepsilon c}\right] = 2\pi\left[\sigma(v+w)+g-\frac{\rho}{\varepsilon c}\right].$$

Hence

$$F_2 = -\overline{F}_1\,.$$

Thus

$$\partial\, w = F_1(v,w), \tag{3.3}$$
$$\overline{\partial}\, w = -\overline{F}_1(v,w). \tag{3.4}$$

4 Integral representation

Let G be a bounded domain in $\mathbb{R}^3$ and a a positive constant. We consider in $\mathbb{R}^4$ the cylinder $Z = G\times[-a,a]$. A right inverse to the Weyl operator is the following **Teodorescu transform**:

$$(T_Z u)(x) = \frac{-1}{\sigma_3}\int_Z e(x-y)u(y)dy\,, \quad Z = G\times[-a,a]$$

with $e(x) = \overline{x}/|x|^4$, $\sigma_3 = 2\pi^{3/2}/\Gamma(3/2)$. We obtain in a straightforward manner

$$\partial\, T_Z\, u = \begin{cases} u & \text{in} \quad Z, \\ 0 & \text{in} \quad \mathbb{R}^4\setminus\overline{Z}, \end{cases}$$

and

$$T_Z\,\partial\, u + \phi_Z = \begin{cases} u & \text{in} \quad Z, \\ 0 & \text{in} \quad \mathbb{R}^4\setminus\overline{Z}, \end{cases}$$

with $\phi_Z \in \ker\partial$. In complete analogy a conjugate Teodorescu transform T_Z^* is introduced. We just have to replace $e(x)$ by its conjugate. Now it follows from (7)–(8) that

$$v = T_Z F_1(v,w) + \phi_Z \qquad (\partial\phi_Z = 0),$$
$$w = T_Z^* F_2(v,w) + \phi_Z^* \qquad (\overline{\partial}\phi_Z^* = 0).$$

Furthermore we have to introduce Cauchy-Bizadse-type operators, which are defined by the boundary data. These operators read as follows:

$$(F_{\partial Z}u)(x) = \frac{1}{\sigma_3}\int_{\partial Z} e(x-y)n(y)u(y)d(\partial Z)_y \quad (x\notin\partial Z)$$

and

$$(F_{\partial Z}^* u)(x) := \frac{1}{\sigma_3}\int_{\partial Z} \overline{e}(x-y)n(y)u(y)d(\partial Z)_y, \quad (x\notin\partial Z)$$

where $n(y) = (n_0+\underline{n})(y)$ denotes the unit vector of the outer normal on ∂Z at the point y.

It can be proved that

$$\phi_Z^* = F_{\partial Z}^* w \qquad \text{and} \quad \phi_Z = F_{\partial Z} v \qquad \text{in } Z\,.$$

It should be noted that we do not need the whole trace of the functions w and v on the boundary. We just have to consider these parts of $tr_Z v$ ($tr_Z w$) which are lying in the corresponding Hardy space of functions, which permit a quaternionic regular (quaternionic anti-regular) extension into Z, accordingly. We get the integral equations

$$v = 4\pi\sigma\, T_Z(v+w) + 4\pi\, T_Z(\sigma g + \frac{\rho}{\varepsilon c}) + h, \tag{4.1}$$

$$w = 4\pi\sigma\, T_Z^*(v+w) + 4\pi\, T_Z^*(\sigma g - \frac{\rho}{\varepsilon c}) + h^*, \tag{4.2}$$

where

$$h = F_{\partial Z} tr_{\partial Z} v \qquad \text{and} \qquad h^* = F_{\partial Z}^* tr_{\partial Z} w.$$

If $h, h*$ are known then under smallness conditions the iteration procedure:

$$v_n = 4\pi\sigma T_Z(v_{n-1} + w_{n-1}) + 4\pi T_Z(\sigma g + \frac{\rho}{\varepsilon c}) + h,$$

$$w_n = 4\pi\sigma T_Z^*(v_{n-1} + w_{n-1}) + 4\pi T_Z^*(\sigma g - \frac{\rho}{\varepsilon c}) + h^*,$$

with $(v_0 = w_0 = 0)$ will converge in suitable Banach spaces.

Remark 4.1 *In [1] is proved the following estimation:*

$$\|T_Z\|_{L(L_\infty, C)} \leq \frac{2\sigma_3^2}{3}\, a\, |G|.$$

5 Weak time dependent Maxwell-equations

Assume now $\varepsilon = \varepsilon(x), \mu = \mu(x), \kappa = \kappa(x)$ $(g = 0)$ and

$$E(t,x) = E_0(t)E(x) \qquad \text{and} \qquad H(t,x) = H_0(t)H_1(x),$$

where the scalar functions E_0 and H_0 are known. Maxwell equations then transform to

$$c\, E_0\, \text{rot}\, E_1 = -\partial_t(\mu H_0) H_1\,, \tag{5.1}$$

$$c\, H_0\, \text{rot}\, H_1 = (\partial_t(\varepsilon E_0) + 4\pi\kappa E_0) E_1\,, \tag{5.2}$$

$$E_0(\nabla\varepsilon \cdot E_1) + \varepsilon\, \text{div}\, E_1 = 4\pi\rho\,, \tag{5.3}$$

$$(\nabla\mu \cdot H_1) + \mu\, \text{div}\, H_1 = 0\,. \tag{5.4}$$

It follows

$$\text{rot}\, E_1 = -\frac{\mu}{c}\frac{\partial_t H_0}{E_0} H_1 =: \alpha_0 H_1\,,$$

$$\text{rot}\, H_1 = \left(\frac{\varepsilon}{c}\frac{\partial_t E_0}{H_0} + \frac{4\pi\kappa}{c}\frac{E_0}{H_0}\right) E_1 =: \beta_0 E_1\,,$$

$$-\text{div}\, E_1 = -\frac{4\pi\rho}{\varepsilon E_0} + \frac{\nabla\varepsilon}{\varepsilon} \cdot E_1 = \rho' - \underline{\alpha} \cdot E_1\,,$$

$$-\text{div } H_1 = \frac{\nabla\mu}{\mu}\cdot H_1 = -\underline{\beta}\cdot H_1 .$$

Here $\alpha = \alpha_0 + \underline{\alpha}$, $\beta = \beta_0 + \underline{\beta}$, $\alpha := -\frac{\nabla\varepsilon}{\varepsilon}$, $\beta := -\frac{\nabla\mu}{\mu}$. Using the fact that in $\mathbb{H}$

$$D\underline{u} = -\text{ div } \underline{u} + \text{ rot } \underline{u},$$

we get

$$\begin{aligned} D\,E_1 &= \alpha_0 H_1 + \rho' - \underline{\alpha}\cdot E_1, \\ D\,H_1 &= \beta_0 E_1 - \underline{\beta}\cdot H_1. \end{aligned}$$

The right inverse of D is the corresponding Teodorescu transform T_G over $G \subset \mathbb{R}^3$. A short calculation leads to

$$\begin{aligned} E_1 &= T_G\alpha_0 H_1 - T_G\underline{\alpha}\cdot E_1 + T_G\rho' + \phi_1, \\ H_1 &= T_G\beta_0 E_1 - T_G\underline{\beta}\cdot H_1 + \phi_2 , \end{aligned}$$

where $\phi_i \in \ker D$ $(i = 1, 2)$. The iteration method

$$\begin{aligned} E_1^{(n)} &= -T_G\underline{\alpha}\cdot E_1^{(n-1)} + T_G\alpha_0 H_1^{(n-1)} + T_G\rho' + \phi_1, \\ H_1^{(n)} &= T_G\underline{\beta}_0\cdot E_1^{(n)} - T_G\underline{\beta}\cdot H_1^{(n-1)} + \phi_2, \end{aligned}$$

with $H_1^{(0)} = E_1^{(0)} = 0$ converges in suitable Banach spaces (L_2, W_2^1, C) under smallness conditions.

In the time-harmonic case i.e. $H_0 = E_0 \equiv 1$ and ε, μ, are constants and $\kappa = \kappa(x)$ we have

$$DE_1 = \rho' \qquad \text{and} \qquad DH_1 = \beta_0\, E_1.$$

Setting $\beta_0 = \delta^{-1}$ we obtain

$$D\,\delta D\,H_1 = \frac{4\pi\rho}{\varepsilon} = \rho',$$

i.e.

$$\triangle\, H_1 = -f.$$

If boundary values of H_1 $(tr_\Gamma H_1)$ are known i.e. $tr_\Gamma H_1 = g$ the complete solution is given by

$$H_1 = F_\Gamma g + T_G\mathcal{P}_\delta Dh + T_G\mathcal{Q}_\delta\delta T_G f\ . \tag{5.5}$$

Here $\mathcal{P}_\delta$ and $\mathcal{Q}_\delta$ are orthoprojections on subspaces in the quaternionic Hilbert space $L_2(G)$, namely

$$L_2(G) = \delta \ker D \cap L_2(G) \bigoplus_\delta D\,\overset{\circ}{W}{}_2^1\,(G).$$

The scalar product is defined by

$$(u,v)_\delta := \int_G \overline{u}\delta v dG \in \mathbb{H}.$$

The operator $\mathcal{P}_\delta$ can be seen as a generalized **Bergman projection.**

In the representation formula from above is F_Γ the Cauchy-Bizadse operator on Γ and h a smooth continuation of g into G. Note that $\mathcal{P}_\delta$ and $\mathcal{Q}_\delta$ can be explicitly defined (cf. [9])! Then

$$E_1 = \frac{c}{4\pi\kappa}\,\mathcal{P}_\delta Dh + \mathcal{Q}_\delta \delta T_G f.$$

Let us prove that the boundary condition is fulfilled! Indeed,

$$\mathcal{Q}_\delta T_G f = D\tilde{f} \qquad \text{with} \quad \tilde{f} \in \overset{\circ}{W}{}_2^1 \quad \text{i.e.} \quad tr_\Gamma \tilde{f} = 0.$$

$$T_G D\tilde{f} = \tilde{f} - F_\Gamma \tilde{f} = 0 \qquad \text{(Borel-Pompeiu's formula).}$$

On the other hand, Plemelj-Sokhotzkij's formulae yield:

$$\begin{aligned} tr_\Gamma H_1 &= P_\Gamma g + tr_\Gamma \mathcal{P}_\delta Dh = P_\Gamma g + tr_\Gamma T Dh - tr_\Gamma T \mathcal{Q}_\delta Dh \\ &= P_\Gamma g + g - P_\Gamma g + 0 = g. \end{aligned}$$

P_Γ is the so-called Plemelj-projection onto that Hardy space of $\mathbb{H}$-regular extendible functions into G.

Bibliography

1. H. Bahmann, K. Guerlebeck , M. Shapiro and W. Sproessig, On a modified Teodorescu transform, *Integral Transforms and Special Functions* **12** (2001), 213–226.
2. A. W. Bitsadze, On two-dimensional integrals of Cauchy-type, *Akademii Nauk Grus. SSR* **16** (1955), 177–184 (Russian).
3. E. F. Bolinder, The classical electromagnetic equations expressed as complex four-dimensional quantities. *J. Franklin Inst.* **263** (1957), 213–223.
4. F. Brackx, R. Delanghe and F. Sommen, Clifford analysis, *Pitman Research Notes in Math.*, Boston, London, Melbourne, 1982.
5. C. Chevalley, The algebraic theory of spinors, Columbia University Press, New York, 1954.
6. W. K. Clifford, Applications of Grassmann's extensive algebra. *Americ. J. of Math. Pure and Appl.* **1** (1878), 350–358.
7. R. Fueter, Analytische Theorie einer Quaternionenvariablen. *Comment. Math. Helv.* **4** (1932), 9–20.
8. K. Guerlebeck and W. Sproessig, Quaternionic Analysis and Boundary Value Problems, Birkhuser Verlag, Basel, 1990.
9. K. Guerlebeck and W. Sproessig , Quaternionic and Clifford calculus for physicists and engineers, *Mathematical Methods in Practice* Vol. 1, John Wiley & Sons, 1997.

10. W. R. Hamilton, Elements of Quaternions (2 Vols), Chelsea, (reprint 1969) 1866.
11. D. Hestenes, Space-Time Algebra, Gordon and Breach, New York, 1966.
12. D. Hestenes, New foundations for classical mechanics, Reidel, Dordrecht, Boston, 1985.
13. D. Hestenes and G. Sobzyk, Clifford algebras for mathematics and physics, Reidel, Dordrecht, 1985.
14. V. V. Kravchenko and M. Shapiro, Integral representations for spatial models of mathematical physics, *Pitman Research Notes in Math.* Series 351, 1996.
15. J. C. Maxwell, The Scientific Papers (2 Vols), Dover, 1969.
16. M. Mercier, Expression des Equations de lectromagnetisme au moyen des nombres au Clifford, *Thesis Nr. 953, University of Geneva*, 1935.
17. M. Riesz, Clifford Numbers and Spinors, *Lecture Series 38*, Maryland, 1958.
18. L. Silberstein, The theory of relativity, Macmillan, London, 1914.
19. W. Sproessig and E. Venturino, The treatment of window problems by transform methods, *Zeitschrift für Analysis und Anwendungen* **12** (1996), 643–654.
20. A. Sudbery, Quaternionic analysis, *Math. Proc. Cambr. Phil. Soc.* **85** (1979), 199–225.

Chapter 3

Metrology

Orthogonal distance fitting of parametric curves and surfaces

Sung Joon Ahn, Engelbert Westkämper, and Wolfgang Rauh

Fraunhofer Institute for Manufacturing Engineering and Automation (IPA)
Nobelstr. 12, 70569 Stuttgart, Germany
{sja; wke; wor}@ipa.fhg.de

Abstract

Fitting of parametric curves and surfaces to a set of given data points is a relevant subject in various fields of science and engineering. In this paper, we review the current orthogonal distance fitting algorithms for parametric models in a well organized and easily understandable manner, and present a new algorithm. Each of these algorithms estimates the model parameters minimizing the square sum of the error distances between the model feature and the given data points. The model parameters are grouped and simultaneously estimated in terms of form, position, and rotation parameters. The form parameters determine the shape of the model feature, and the position/rotation parameters describe the rigid body motion of the model feature. The new algorithm is applicable to any kind of parametric curve and surface. We give fitting examples for circle, cylinder, and helix in space.

1 Introduction

The use of parametric curves and surfaces is very common and model fitting to a set of given data points is a relevant subject in various fields of science and engineering. For fitting of curves and surfaces, orthogonal distance fitting is of primary concern because of the applied error definition, namely the shortest distance from the given point to the model feature [5, 9]. While there are orthogonal distance fitting algorithms for explicit [3], and implicit models [2, 7] in the literature, we are considering in this paper fitting algorithms for parametric models [4, 6, 8, 10, 11] (Fig. 1).

The goal of the orthogonal distance fitting is the estimation of the model parameters minimizing the performance index

$$\sigma_0^2 = (\mathbf{X} - \mathbf{X}')^{\mathrm{T}} \mathbf{P}^{\mathrm{T}} \mathbf{P} (\mathbf{X} - \mathbf{X}') \tag{1.1}$$

or

$$\sigma_0^2 = \mathbf{d}^{\mathrm{T}} \mathbf{P}^{\mathrm{T}} \mathbf{P} \mathbf{d}\,, \tag{1.2}$$

where $\mathbf{X}^{\mathrm{T}} = (\mathbf{X}_1^{\mathrm{T}}, \ldots, \mathbf{X}_m^{\mathrm{T}})$ and $\mathbf{X}'^{\mathrm{T}} = (\mathbf{X}_1'^{\mathrm{T}}, \ldots, \mathbf{X}_m'^{\mathrm{T}})$ are the coordinates vectors of the m given points and of the m corresponding points on the model feature, respectively. Moreover, $\mathbf{d}^{\mathrm{T}} = (d_1, \ldots, d_m)$ is the distances vector with $d_i = \|\mathbf{X}_i - \mathbf{X}_i'\|$, $\mathbf{P}^{\mathrm{T}}\mathbf{P}$ is the weighting matrix. We are calling the fitting algorithms based on the performance indexes (1.1) and (1.2) *coordinate-based algorithm* and *distance-based algorithm*, respectively.

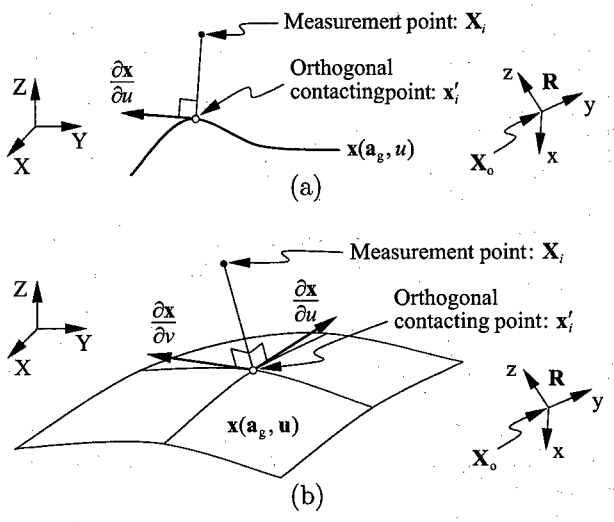


FIG. 1. Parametric features, and the orthogonal contacting point $\mathbf{x}'_i$ in frame xyz from the given point $\mathbf{X}_i$ in frame XYZ: (a) Curve; (b) Surface.

In this paper, the model parameters $\mathbf{a}$ are grouped and simultaneously estimated in three categories. First, the *form parameters* $\mathbf{a}_g$ (e.g. three axis lengths a, b, c of an ellipsoid) describe the shape of the standard model feature defined in model coordinate system xyz (Fig. 1)

$$\mathbf{x} = \mathbf{x}(\mathbf{a}_g, \mathbf{u}) \quad \text{with} \quad \mathbf{a}_g = (a_1, \ldots, a_l)^T. \tag{1.3}$$

The form parameters are invariant to the rigid body motion of the model feature. The second and the third parameters groups, respectively the *position parameters* $\mathbf{a}_p$ and the *rotation parameters* $\mathbf{a}_r$, describe the rigid body motion of the model feature in machine coordinate system XYZ:

$$\begin{aligned} & \mathbf{X} = \mathbf{R}^{-1}\mathbf{x} + \mathbf{X}_o \quad \text{or} \quad \mathbf{x} = \mathbf{R}(\mathbf{X} - \mathbf{X}_o), \\ \text{where} \quad & \mathbf{R} = \mathbf{R}_\kappa \mathbf{R}_\varphi \mathbf{R}_\omega = (\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3)^T, \quad \mathbf{R}^{-1} = \mathbf{R}^T, \\ & \mathbf{a}_p = \mathbf{X}_o = (X_o, Y_o, Z_o)^T, \quad \text{and} \quad \mathbf{a}_r = (\omega, \varphi, \kappa)^T. \end{aligned} \tag{1.4}$$

A subproblem of the orthogonal distance fitting of a parametric model is the finding of the location parameters $\{\mathbf{u}_i\}_{i=1}^m$, which represent the nearest points $\{\mathbf{X}'_i\}_{i=1}^m$ on the model feature from each given point $\{\mathbf{X}_i\}_{i=1}^m$. The model parameters $\mathbf{a}$ and the location parameters $\{\mathbf{u}_i\}_{i=1}^m$ will generally be estimated through iteration. By the *total method* [6, 10], $\mathbf{a}$ and $\{\mathbf{u}_i\}_{i=1}^m$ will be simultaneously determined, while they are to be separately estimated by the *variable-separation method* [4, 8, 11] in a nested iteration scheme. There could be four combinations for algorithmic approaches as shown in Table 1. One of the algorithmic approaches in Table 1 results in an obviously underdetermined linear system for iteration, thus, it has no practical application. We describe and compare the realistic three algorithmic approaches in the following sections.

Algorithmic approaches	Distance-based algor.	Coordinate-based algor.
Total method	Underdetermined system	I (ETH [6, 10])
Variable-separation method	II (NPL [4, 11])	III (FhG, this paper)

TAB. 1. Orthogonal distance fitting algorithms for parametric models.

2 Orthogonal distance fitting algorithm I (ETH)

The ETH algorithm [6, 10] is based on the performance index (1.1), and simultaneously estimates the model parameters $\mathbf{a}$ and the location parameters $\{\mathbf{u}_i\}_{i=1}^m$ for the nearest points on the model feature. We introduce the new estimation parameters vector $\mathbf{b}$ containing $\mathbf{a}$ and $\{\mathbf{u}_i\}_{i=1}^m$ as follows,

$$\mathbf{b}^{\mathrm{T}} = (\mathbf{a}^{\mathrm{T}}, \mathbf{u}_1^{\mathrm{T}}, \ldots, \mathbf{u}_m^{\mathrm{T}}) = (\mathbf{a}_{\mathrm{g}}^{\mathrm{T}}, \mathbf{a}_{\mathrm{p}}^{\mathrm{T}}, \mathbf{a}_{\mathrm{r}}^{\mathrm{T}}, \mathbf{u}_1^{\mathrm{T}}, \ldots, \mathbf{u}_m^{\mathrm{T}}) .$$

The parameters vector $\mathbf{b}$ minimizing the performance index (1.1) can be determined by the Gauss-Newton method

$$\mathbf{P}\frac{\partial \mathbf{X}'}{\partial \mathbf{b}}\bigg|_k \Delta\mathbf{b} = \mathbf{P}(\mathbf{X} - \mathbf{X}')|_k , \qquad \mathbf{b}_{k+1} = \mathbf{b}_k + \alpha\Delta\mathbf{b} , \tag{2.1}$$

with the Jacobian matrices of each point $\mathbf{X}_i'$ on the model feature, from (1.3) and (1.4)

$$\mathbf{J}_{\mathbf{X}_i',\mathbf{b}} = \frac{\partial \mathbf{X}}{\partial \mathbf{b}}\bigg|_{\mathbf{X}=\mathbf{X}_i'} = \left(\mathbf{R}^{-1}\frac{\partial \mathbf{x}}{\partial \mathbf{b}} + \frac{\partial \mathbf{R}^{-1}}{\partial \mathbf{b}}\mathbf{x} + \frac{\partial \mathbf{X}_{\mathrm{o}}}{\partial \mathbf{b}}\right)\bigg|_{\mathbf{u}=\mathbf{u}_i}$$

$$= \left(\mathbf{R}^{-1}\frac{\partial \mathbf{x}}{\partial \mathbf{a}_{\mathrm{g}}} \;\middle|\; \mathbf{I} \;\middle|\; \frac{\partial \mathbf{R}^{-1}}{\partial \mathbf{a}_{\mathrm{r}}}\mathbf{x} \;\middle|\; \mathbf{0}_1, \cdots, \mathbf{0}_{i-1}, \mathbf{R}^{-1}\frac{\partial \mathbf{x}}{\partial \mathbf{u}}, \mathbf{0}_{i+1}, \cdots, \mathbf{0}_m\right)\bigg|_{\mathbf{u}=\mathbf{u}_i} .$$

A disadvantage of the ETH algorithm is that the storage space and the computing time cost increase very rapidly with the number of the data points, unless the sparse linear system (2.1) is handled beforehand by a sparse matrix algorithm.

3 Orthogonal distance fitting algorithm II (NPL)

The NPL algorithm [4, 11] is based on the performance index (1.2), and separately estimates the model parameters $\mathbf{a}$ and the location parameters $\{\mathbf{u}_i\}_{i=1}^m$ in a *nested iteration* scheme

$$\min_{\mathbf{a}} \min_{\{\mathbf{u}_i\}_{i=1}^m} \sigma_0^2 \left(\{\mathbf{X}_i'(\mathbf{a}, \mathbf{u})\}_{i=1}^m\right) .$$

The inner iteration determines the location parameters $\{\mathbf{u}_i'\}_{i=1}^m$ for the minimum distance points $\{\mathbf{X}_i'\}_{i=1}^m$ on the current model feature from each given point $\{\mathbf{X}_i\}_{i=1}^m$, and, the outer iteration updates the model parameters. In this paper, in order to implement the parameters grouping of $\mathbf{a}^{\mathrm{T}} = (\mathbf{a}_{\mathrm{g}}^{\mathrm{T}}, \mathbf{a}_{\mathrm{p}}^{\mathrm{T}}, \mathbf{a}_{\mathrm{r}}^{\mathrm{T}})$, we have modified the initial NPL algorithm.

3.1 Orthogonal contacting point

For each given point $\mathbf{x}_i = \mathbf{R}(\mathbf{X}_i - \mathbf{X}_{\mathrm{o}})$ in frame xyz, we determine the orthogonal contacting point $\mathbf{x}_i'$ on the standard model feature (1.3). Then, the orthogonal contacting point $\mathbf{X}_i'$ in frame XYZ to the given point $\mathbf{X}_i$ will be obtained through a backward transformation of $\mathbf{x}_i'$ into XYZ. We are searching the location parameters $\mathbf{u}$ which minimizes the error distance between the given point $\mathbf{x}_i$ and the corresponding point $\mathbf{x}$ on the model

feature (1.3)

$$D = (\mathbf{x}_i - \mathbf{x}(\mathbf{a}_g, \mathbf{u}))^T (\mathbf{x}_i - \mathbf{x}(\mathbf{a}_g, \mathbf{u})) . \tag{3.1}$$

The first order necessary condition for a minimum of (3.1) as a function of $\mathbf{u}$ is

$$\mathbf{f}(\mathbf{x}_i, \mathbf{x}(\mathbf{a}_g, \mathbf{u})) = \frac{1}{2}\begin{pmatrix} D_u \\ D_v \end{pmatrix} = -\begin{pmatrix} (\mathbf{x}_i - \mathbf{x}(\mathbf{a}_g, \mathbf{u}))^T \mathbf{x}_u \\ (\mathbf{x}_i - \mathbf{x}(\mathbf{a}_g, \mathbf{u}))^T \mathbf{x}_v \end{pmatrix} = \mathbf{0} . \tag{3.2}$$

The condition (3.2) means that the error vector $(\mathbf{x}_i - \mathbf{x})$ and the surface tangent vectors $\partial\mathbf{x}/\partial\mathbf{u}$ at $\mathbf{x}$ should be orthogonal. We solve (3.2) for $\mathbf{u}$ by using the Newton method (how to derive the Jacobian matrix $\partial\mathbf{f}/\partial\mathbf{u}$ is shown in Section 4).

$$\left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_k \Delta\mathbf{u} = -\mathbf{f}(\mathbf{u})|_k , \qquad \mathbf{u}_{k+1} = \mathbf{u}_k + \alpha\Delta\mathbf{u} . \tag{3.3}$$

3.2 Orthogonal distance fitting

We update the model parameters $\mathbf{a}$ minimizing the performance index (1.2) by using the Gauss-Newton method (outer iteration)

$$\mathbf{P}\left.\frac{\partial \mathbf{d}}{\partial \mathbf{a}}\right|_k \Delta\mathbf{a} = -\mathbf{P}\mathbf{d}|_k , \qquad \mathbf{a}_{k+1} = \mathbf{a}_k + \alpha\Delta\mathbf{a} .$$

From $d_i = \|\mathbf{X}_i - \mathbf{X}_i'\|$, and equations (1.3) and (1.4), we derive the Jacobian matrices of each *orthogonal distance* d_i

$$\mathbf{J}_{d_i,\mathbf{a}} = \frac{\partial d_i}{\partial \mathbf{a}} = -\frac{(\mathbf{X}_i - \mathbf{X}_i')^T}{\|\mathbf{X}_i - \mathbf{X}_i'\|}\left.\frac{\partial \mathbf{X}}{\partial \mathbf{a}}\right|_{\mathbf{u}=\mathbf{u}_i'}$$

$$= -\frac{(\mathbf{X}_i - \mathbf{X}_i')^T}{\|\mathbf{X}_i - \mathbf{X}_i'\|}\left(\mathbf{R}^{-1}\left(\frac{\partial \mathbf{x}}{\partial \mathbf{a}} + \frac{\partial \mathbf{x}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{a}}\right) + \frac{\partial \mathbf{R}^{-1}}{\partial \mathbf{a}}\mathbf{x} + \frac{\partial \mathbf{X}_o}{\partial \mathbf{a}}\right)\Bigg|_{\mathbf{u}=\mathbf{u}_i'} .$$

With (1.4) and (3.2) at $\mathbf{u}=\mathbf{u}_i'$,

$$(\mathbf{X}_i - \mathbf{X}_i')^T \mathbf{R}^{-1}\left.\frac{\partial \mathbf{x}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}_i'} = (\mathbf{x}_i - \mathbf{x}_i')^T\left.\frac{\partial \mathbf{x}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}_i'} = \mathbf{0}^T ,$$

$$\text{and} \quad \mathbf{J}_{d_i,\mathbf{a}} = -\frac{(\mathbf{X}_i - \mathbf{X}_i')^T}{\|\mathbf{X}_i - \mathbf{X}_i'\|}\left(\mathbf{R}^{-1}\left.\frac{\partial \mathbf{x}}{\partial \mathbf{a}_g}\right|_{\mathbf{u}=\mathbf{u}_i'} \;\middle|\; \mathbf{I} \;\middle|\; \frac{\partial \mathbf{R}^{-1}}{\partial \mathbf{a}_r}\mathbf{x}_i'\right)$$

is the resultant Jacobian matrix for d_i. A drawback of the NPL algorithm is that the convergence and the accuracy of 3D-curve fitting (e.g. fitting of a circle in space) are relatively poor. 2D-curve fitting or surface fitting with the NPL algorithm do not suffer from such problems.

4 Orthogonal distance fitting algorithm III (FhG)

At the Fraunhofer Institute IPA (FhG-IPA), a new orthogonal distance fitting algorithm for parametric models is developed, which minimizes the performance index (1.1) in a nested iteration scheme (variable-separation method). The new algorithm is a generalized extension of an orthogonal distance fitting algorithm for implicit plane curves [1]. Interested readers are referred to [2] for the orthogonal distance fitting of implicit surfaces and plane curves. The location parameter values $\{\mathbf{u}_i'\}_{i=1}^m$ for the minimum distance

points $\{\mathbf{X}'_i\}_{i=1}^m$ on the current model feature from each given point $\{\mathbf{X}_i\}_{i=1}^m$ are to be found by the algorithm described in Section 3.1 (inner iteration). In this section, we intend to describe the outer iteration which updates the model parameters $\mathbf{a}$ minimizing the performance index (1.1) by using the Gauss-Newton method

$$\mathbf{P}\frac{\partial \mathbf{X}'}{\partial \mathbf{a}}\bigg|_k \Delta\mathbf{a} = \mathbf{P}(\mathbf{X}-\mathbf{X}')|_k\,, \qquad \mathbf{a}_{k+1} = \mathbf{a}_k + \alpha\Delta\mathbf{a}\,, \tag{4.1}$$

with the Jacobian matrices of each *orthogonal distance point* $\mathbf{X}'_i$, from (1.3) and (1.4)

$$\begin{aligned}\mathbf{J}_{\mathbf{X}'_i,\mathbf{a}} &= \frac{\partial \mathbf{X}}{\partial \mathbf{a}}\bigg|_{\mathbf{X}=\mathbf{X}'_i} = \left(\mathbf{R}^{-1}\left(\frac{\partial \mathbf{x}}{\partial \mathbf{a}} + \frac{\partial \mathbf{x}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{a}}\right) + \frac{\partial \mathbf{R}^{-1}}{\partial \mathbf{a}}\mathbf{x} + \frac{\partial \mathbf{X}_o}{\partial \mathbf{a}}\right)\bigg|_{\mathbf{u}=\mathbf{u}'_i} \\ &= \mathbf{R}^{-1}\frac{\partial \mathbf{x}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{a}}\bigg|_{\mathbf{u}=\mathbf{u}'_i} + \left(\mathbf{R}^{-1}\frac{\partial \mathbf{x}}{\partial \mathbf{a}_g}\bigg|_{\mathbf{u}=\mathbf{u}'_i} \;\Big|\; \mathbf{I} \;\Big|\; \frac{\partial \mathbf{R}^{-1}}{\partial \mathbf{a}_r}\mathbf{x}'_i\right).\end{aligned} \tag{4.2}$$

The derivative matrix $\partial\mathbf{u}/\partial\mathbf{a}$ at $\mathbf{u}=\mathbf{u}'_i$ in (4.2) describes the variational behavior of the location parameters $\mathbf{u}'_i$ for the orthogonal contacting point $\mathbf{x}'_i$ in frame xyz relative to the differential changes of the parameters vector $\mathbf{a}$. Purposefully, we derive $\partial\mathbf{u}/\partial\mathbf{a}$ from the condition (3.2). Because (3.2) has an implicit form, its derivatives lead to

$$\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{a}} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}_i}\frac{\partial \mathbf{x}_i}{\partial \mathbf{a}} + \frac{\partial \mathbf{f}}{\partial \mathbf{a}} = \mathbf{0} \quad \text{or} \quad \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{a}} = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}_i}\frac{\partial \mathbf{x}_i}{\partial \mathbf{a}} + \frac{\partial \mathbf{f}}{\partial \mathbf{a}}\right), \tag{4.3}$$

where $\partial\mathbf{x}_i/\partial\mathbf{a}$ is, from $\mathbf{x}_i = \mathbf{R}(\mathbf{X}_i - \mathbf{X}_o)$,

$$\frac{\partial \mathbf{x}_i}{\partial \mathbf{a}} = \frac{\partial \mathbf{R}}{\partial \mathbf{a}}(\mathbf{X}_i - \mathbf{X}_o) - \mathbf{R}\frac{\partial \mathbf{X}_o}{\partial \mathbf{a}} = \left(\begin{array}{c|c|c} \mathbf{0} & -\mathbf{R} & \dfrac{\partial \mathbf{R}}{\partial \mathbf{a}_r}(\mathbf{X}_i - \mathbf{X}_o) \end{array}\right).$$

The other three matrices $\partial\mathbf{f}/\partial\mathbf{u}$, $\partial\mathbf{f}/\partial\mathbf{x}_i$, and $\partial\mathbf{f}/\partial\mathbf{a}$ in (3.3) and (4.3) are to be directly derived from (3.2). The elements of these three matrices are composed of simple linear combinations of components of the error vector $(\mathbf{x}_i - \mathbf{x})$ with elements of the following three vector/matrices $\partial\mathbf{x}/\partial\mathbf{u}$, $\mathbf{H}$, and $\mathbf{G}$ (XHG matrix):

$$\frac{\partial \mathbf{x}}{\partial \mathbf{u}} = (\mathbf{x}_u \;\; \mathbf{x}_v)\,, \qquad \mathbf{H} = \begin{pmatrix} \mathbf{x}_{uu} & \mathbf{x}_{uv} \\ \mathbf{x}_{vu} & \mathbf{x}_{vv} \end{pmatrix}, \qquad \mathbf{G} = \begin{pmatrix} \mathbf{G}_0 \\ \mathbf{G}_1 \\ \mathbf{G}_2 \end{pmatrix} = \frac{\partial}{\partial \mathbf{a}_g}\begin{pmatrix} \mathbf{x} \\ \mathbf{x}_u \\ \mathbf{x}_v \end{pmatrix}, \tag{4.4}$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{u}} = (\mathbf{x}_u \;\; \mathbf{x}_v)^{\mathrm{T}}(\mathbf{x}_u \;\; \mathbf{x}_v) - \begin{pmatrix} (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}}\mathbf{x}_{uu} & (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}}\mathbf{x}_{uv} \\ (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}}\mathbf{x}_{vu} & (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}}\mathbf{x}_{vv} \end{pmatrix},$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}_i} = -(\mathbf{x}_u \;\; \mathbf{x}_v)^{\mathrm{T}}, \qquad \frac{\partial \mathbf{f}}{\partial \mathbf{a}} = \left(\begin{array}{c|c|c} \mathbf{x}_u^{\mathrm{T}}\mathbf{G}_0 - (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}}\mathbf{G}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{x}_v^{\mathrm{T}}\mathbf{G}_0 - (\mathbf{x}_i - \mathbf{x})^{\mathrm{T}}\mathbf{G}_2 & & \end{array}\right).$$

Now (4.3) can be solved for $\partial\mathbf{u}/\partial\mathbf{a}$ at $\mathbf{u} = \mathbf{u}'_i$, and the Jacobian matrix (4.2) and the linear system (4.1) can be completed and solved for the parameter update $\Delta\mathbf{a}$.

We would like to stress that only the standard model equation (1.3), without involvement of the position/rotation parameters, is required in (4.4). The overall structure of the FhG algorithm remains unchanged for all dimensional fitting problems of parametric models. All that is necessary for a new parametric model is to derive the XHG matrix of (4.4) from (1.3) of the new model feature, and to supply a proper set of initial para-

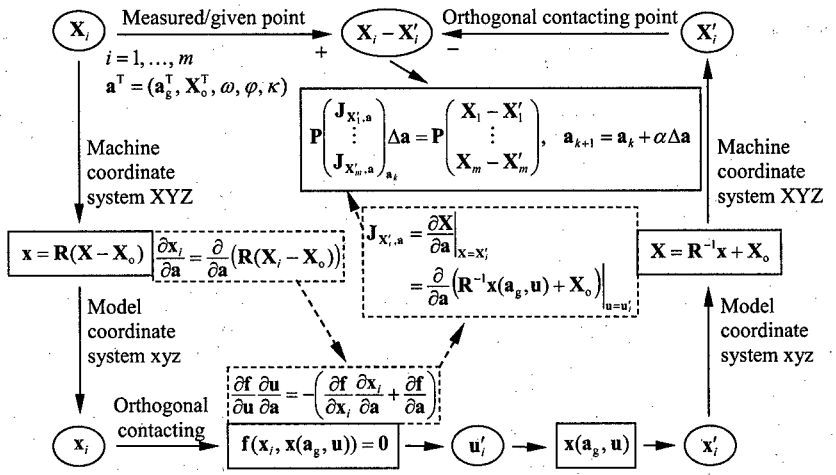


FIG. 2. Information flow with the FhG algorithm.

X	5	6	5	5	3	2	0	−1	−1	0	3	4	7	9
Y	1	3	4	6	5	4	2	0	−2	−5	−7	−8	−10	−9
Z	−3	−1	1	3	5	7	9	11	11	11	11	11	11	10

TAB. 2. Fourteen coordinate triples representing a helix.

meter values $\mathbf{a}_0$ for iteration (4.1). An overall schematic information flow with the FhG algorithm is shown in Fig. 2. The FhG algorithm shows robust and fast convergence with 2D/3D-curve and surface fitting. The storage space and computing time cost are proportional to the number of data points. A disadvantage of the FhG algorithm is that it additionally requires the second derivatives $\partial^2\mathbf{x}/\partial\mathbf{a}_{\mathrm{g}}\partial\mathbf{u}$ as shown in (4.4).

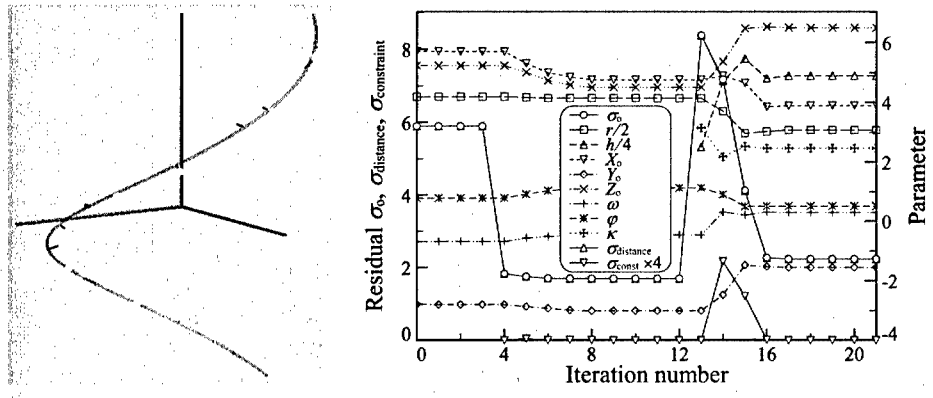
As a fitting example, we show the orthogonal distance fitting of a helix. The standard model feature (1.3) of a helix in frame xyz can be described as follows. $\mathbf{x}(\mathbf{a}_{\mathrm{g}}, u) = \mathbf{x}(r, h, u) = (r\cos u, r\sin u, hu/2\pi)^{\mathrm{T}}$, with a constraint on the position and rotation parameters

$$f_{\mathrm{c}}(\mathbf{a}_{\mathrm{p}}, \mathbf{a}_{\mathrm{r}}) = (\mathbf{X}_{\mathrm{o}} - \overline{\mathbf{X}})^{\mathrm{T}}\mathbf{r}_3(\omega, \varphi) = 0\,,$$

where r and h are respectively the radius and elevation of a helix. $\overline{\mathbf{X}}$ is the gravitational center of the given points set and $\mathbf{r}_3$ (see (1.4)) is the vector of direction cosines of the z-axis. We have obtained the initial parameter values from a 3D-circle fitting, and a cylinder fitting, successively. The helix fitting to the points set in Table 2 with the initial values of $h = 10$ and $\kappa = \pi$ terminated after 0.22s, 8 iteration cycles for $\|\Delta\mathbf{a}\| = 3.2\times10^{-7}$ with a Pentium 133 MHz PC (Table 3, Fig. 3). They were 0.33s, 10 iteration cycles for $\|\Delta\mathbf{a}\| = 3.6\times10^{-7}$ with the ETH algorithm, and, 1.05s, 61 iteration cycles for $\|\Delta\mathbf{a}\| = 8.8\times10^{-7}$ with the NPL algorithm. The computing cost with the ETH algorithm increases rapidly with the number of the data points. The NPL algorithm showed slow convergences with the 3D-circle and the helix fitting (3D-curve fitting).

Parameters $\hat{\mathbf{a}}$	σ_0	r	h	X_o
3D-Circle	5.8913	8.3850	--	5.6999
$\sigma(\hat{\mathbf{a}})$	--	0.7355	--	0.9939
Cylinder	1.6925	8.2835	--	4.7596
$\sigma(\hat{\mathbf{a}})$	--	0.2738	--	0.7465
Helix	2.2301	6.1368	19.5811	3.8909
$\sigma(\hat{\mathbf{a}})$	--	0.4238	1.3214	0.5488
Y_o	Z_o	ω	φ	κ
−2.7923	5.2333	−0.6833	0.7882	--
0.8421	0.8821	0.1177	0.1375	--
−3.0042	4.5081	−0.4576	1.1327	--
0.4525	0.6513	0.3049	0.2116	--
−1.5560	6.4871	0.3003	0.5114	2.4602
0.3934	0.7500	0.0880	0.0663	0.2881

TAB. 3. Results of the orthogonal distance fitting to the points set in Table 2.



(a) (b)

FIG. 3. Orthogonal distance fitting to the points set in Table 2: (a) Helix fit; (b) Convergence of the fit. Iteration number 0–3: 3D-circle, 4–12: circular cylinder, and 13–: helix fit with the initial value of $h=10$ and $\kappa=\pi$.

5 Summary

In this paper, we have reviewed the current orthogonal distance fitting algorithms for parametric curves and surfaces in an easily understandable manner, and presented a new algorithm. By each of the algorithms the model parameters are grouped and simultaneously estimated in terms of form/position/rotation parameters. The ETH algorithm demands a large amount of storage space and high computing cost, and the NPL algorithm shows relatively poor performance with 3D-curve fitting. The new algorithm, the FhG algorithm, has no such drawbacks of the ETH algorithm or of the NPL algorithm. A

disadvantage of the FhG algorithm is that it requires the second derivatives $\partial^2\mathbf{x}/\partial\mathbf{a}_g\partial\mathbf{u}$. The FhG algorithm does not require a necessarily good set of initial parameter values, which could also be internally supplied as demonstrated with the fitting examples. From the viewpoint of implementation and application to a new model feature, the FhG algorithm is universal and very efficient. Merely the standard model equation (1.3) of the new model feature is eventually required, which has only few form parameters. The functional interpretation and treatment of the position/rotation parameters are basically identical for all parametric models. The storage space and the computing time cost are proportional to the number of given data points. Together with other orthogonal distance fitting algorithms for implicit models [2], the FhG algorithm is certified by the German federal authority PTB [5, 9], with a certification grade that the parameter estimation accuracy is higher than 0.1μm for length unit, and 0.1μrad for angle unit for all parameters of all tested model features.

Bibliography

1. S. J. Ahn, W. Rauh, and H.-J. Warnecke, Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola, *Pattern Recognition* **34** (2001), 2283–2303.
2. S. J. Ahn, W. Rauh, and H.-J. Warnecke, Best-Fit of Implicit Surfaces and Plane Curves, in *Mathematical Methods for Curves and Surfaces: Oslo 2000*, T. Lyche and L. L. Schumaker (Eds.), Vanderbilt University Press, TN, 2001, 1–14.
3. P. T. Boggs, R. H. Byrd, and R. B. Schnabel, A stable and efficient algorithm for nonlinear orthogonal distance regression, *SIAM J. Sci. Stat. Comput.* **8** (1987), 1052–1078.
4. B. P. Butler, A. B. Forbes, and P. M. Harris, Algorithms for Geometric Tolerance Assessment, Report no. DITC 228/94, NPL, 1994.
5. R. Drieschner, B. Bittner, R. Elligsen, and F. Wäldele. Testing Coordinate Measuring Machine Algorithms: Phase II, BCR Report, EUR 13417 EN, Commission of the European Communities, Luxemburg, 1991.
6. W. Gander, G. H. Golub, and R. Strebel, Least-squares fitting of circles and ellipses, *BIT* **34** (1994), 558–578.
7. H.-P. Helfrich and D. Zwick, A trust region method for implicit orthogonal distance regression, *Numerical Algorithms* **5** (1993), 535–545.
8. H.-P. Helfrich and D. Zwick. A trust region algorithm for parametric curve and surface fitting, *J. Comput. Appl. Math.* **73** (1996), 119–134.
9. ISO/DIS 10360-6, Geometrical Product Specifications (GPS) - Acceptance test and reverification test for coordinate measuring machines (CMM) - Part 6: Estimation of errors in computing Gaussian associated features, ISO, Geneva, 1999.
10. D. Sourlier, *Three Dimensional Feature Independent Bestfit in Coordinate Metrology*, Ph.D. Thesis, ETH Zurich, 1995.
11. D. A. Turner, *The approximation of Cartesian coordinate data by parametric orthogonal distance regression*, Ph.D. Thesis, University of Huddersfield, 1999.

Template matching in the ℓ_1 norm

Iain J. Anderson and Colin Ross

School of Computing and Mathematics, University of Huddersfield, UK.
i.j.anderson@hud.ac.uk, c.ross@hud.ac.uk

Abstract

We present a method for matching a surface in three dimensions to a set of data sampled from the surface by means of minimising the distances from the data points to the closest point on the surface. This method of association is affine transformation invariant and as such is very useful in situations where the coordinate axes are essentially arbitrary. Traditionally, this problem has been solved by minimising the ℓ_2 norm of the distances from the data points to the corresponding points in the surface, while the use of other ℓ_p norms is less well known. We present a method for template matching in the ℓ_1 norm based upon a method of directional constraints developed by Watson for the related problem of orthogonal distance regression. An algorithm for this method is given and numerical results show its effectiveness.

1 Introduction

Template matching is used in a variety of applications such as the quality assurance of manufactured artifacts [1] and dental metrology [2]. Given a fixed template, i.e., curve or surface, and a set of data in a different frame of reference, template matching involves finding the frame transformation which maps the data onto the template.

A typical strategy for finding the optimal transformation parameters in the template matching problem is to minimize, in some norm, the orthogonal distances between the transformed data and the template. In this case, the template matching problem can be viewed as a form of orthogonal distance regression (ODR) [3], which is a technique commonly used for fitting curves and surfaces to measured data. Therefore, most algorithms for solving the template matching problem are extensions of algorithms for ODR. Template matching in the ℓ_2 norm is addressed by Turner [3] and in the ℓ_∞ norm by Butler et al. [1] as well as by Zwick [7] for the two dimensional case.

In this paper, we are specifically concerned with the following problem.

> Given a fixed differentiable parametric surface $\mathbf{f}(u,v)$ and a set of m data $\{\mathbf{x}_i\}_{i=1}^m \in \Re^3$, find points $\{\mathbf{f}(u_i,v_i)\}_{i=1}^m$, a rotation matrix R_Θ, and a translation vector $\mathbf{t}_0$ such that the ℓ_1 norm of the residual distances $\{\|R_\Theta(\mathbf{x}_i-\mathbf{t}_0)-\mathbf{f}(u_i,v_i)\|_2\}_{i=1}^m$ is minimal.

This is the template matching problem in the ℓ_1 norm, and although not as widely used as the ℓ_2 and ℓ_∞ counterparts, it does nonetheless have an important role to play. The importance of the ℓ_1 norm is that, generally speaking, any outlying data are effectively ignored with the result that an approximation is obtained which is largely independent

of any unreliable data. This has particular importance when our data arises as a result of some measurement process, perhaps involving many complicated and finely-tuned instruments. For such a measurement scenario, any change in the assumed measurement conditions can result in a datum which has gross error relative to other data. Thus, if we choose a measure which is susceptible to outlying data, we are in danger of obtaining an unrepresentative approximation. This situation is avoided by use of the ℓ_1 norm and we therefore advocate its use both here and in any situation involving measurement data where a representative approximation is required.

A feature of optimal ℓ_1 solutions is the likelihood of a small number of the data having a residual of zero, and it is therefore unclear whether the elements of the Jacobian matrix of partial derivatives are well-defined for these points. As a result, use of the usual Gauss-Newton method would appear to be handicapped due to its dependence upon the Jacobian matrix to calculate an updated transformation estimate. This difficulty also arises in the conventional ODR fitting problem and has recently been considered by Watson [6]. His solution is to adopt a method of fitting subject to directional constraints. By setting these directional constraints to be orthogonal to the approximant, Watson shows not only that the Jacobian is defined but also how to compute its elements without incurring a build-up of rounding error.

In this paper, we extend Watson's constrained direction fitting routine to the template matching problem. We show that Watson's results are equally valid for ℓ_1 template matching. Finally, we exploit these results to give a reliable algorithm for the ℓ_1 template matching problem.

The structure of this paper is as follows. Section 2 provides the results necessary to justify the new technique. Section 3 describes the algorithm adopted to implement the theory. Section 4 gives some numerical results for both a simple case and a more challenging case. Finally, Section 5 concludes this paper and presents possibilities for future work.

2 Theory

We are concerned with the minimisation of the quantity

$$E = \|(d_1, \ldots, d_m)\|, \tag{2.1}$$

where

$$d_i = \min_{u_i, v_i} \|\widehat{\mathbf{x}}_i - \mathbf{f}(u_i, v_i)\|_2, \quad i = 1, 2, \ldots, m, \tag{2.2}$$

and

$$\widehat{\mathbf{x}} = R_\Theta(\mathbf{x} - \mathbf{t}_0), \tag{2.3}$$

with respect to the rotation parameters

$$\Theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix},$$

the translation parameters

$$\mathbf{t}_0 = \begin{pmatrix} a \\ b \\ c \end{pmatrix},$$

and the location parameters

$$U = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}.$$

This is a constrained problem and can be solved using a separation-of-variables approach as described by Turner [3] among others. In this approach, the problem of obtaining the transformation parameters

$$\mathbf{t} = \begin{pmatrix} \Theta \\ \mathbf{t}_0 \end{pmatrix},$$

is separated from the subproblem of obtaining the location parameters U. At each iteration, the subproblem is solved to obtain an optimal U for the current transformation parameters $\mathbf{t}$ which is then used to obtain an update of the transformation parameters themselves.

2.1 Considerations specific to the ℓ_1 problem

Up to this point, we have not specified which norm we are using to measure the disparity between the transformed data and the template. Since we will be particularly interested in the ℓ_1 case, this section discusses problems inherent in the solution of such a problem.

The major problem with solving non-linear ℓ_1 problems is that in order to use a technique such as the Gauss-Newton method, derivative information is required. Unfortunately, derivatives of the distances $\mathbf{d}$ are not defined when a distance has a value of zero. Such is the nature of ℓ_1 approximation that zeros are to be expected at an optimal solution [5]. Thus, it is unclear whether the Jacobian matrix is defined at these data points. Recent work by Watson [6] has considered how the related problem of orthogonal distance regression might be solved by considering distances to be measured along fixed *direction vectors* $\mathbf{w}_i$. Orthogonal distance regression involves the fitting of a curve or surface to a set of data where the residuals are taken to be the shortest distance from the data to the approximant [3]. Template matching can be seen as a variant of this since the residuals are measured in the same way, but we are only altering the position and orientation of the approximant, rather than the actual shape itself. Thus, techniques for orthogonal distance regression can be used successfully in template matching.

By means of these directional constraints, it is possible to show that if we choose the directions $\mathbf{w}_i$ to be the orthogonal directions,

$$\mathbf{w}_i = \frac{\mathbf{f}(u_i, v_i) - \widehat{\mathbf{x}}_i}{\|\mathbf{f}(u_i, v_i) - \widehat{\mathbf{x}}_i\|_2}$$

then the derivatives are well defined in the limit as $\|\mathbf{f}(u_i, v_i) - \widehat{\mathbf{x}}_i\|_2 \to 0$.

This result may be summarised in the following Theorem (taken from Watson [6]).

Theorem 2.1 *For parametric fitting, let the (usual) Gauss-Newton method produce a sequence* $\{\mathbf{t}\}$ *such there is a unique unit normal vector to the template at* $\mathbf{f}(u_i, v_i)$, *and*

$\widehat{\mathbf{x}}_i$ remains on one side of the template. Then $\nabla_{\mathbf{t}} d_i$ is well defined on this sequence.

If $\mathbf{f}(u_i, v_i) \to \widehat{\mathbf{x}}_i$, then this formulation will lead to similar problems to which we are attempting to resolve as a result of the quotient becoming undefined. As a result, Watson [6] suggests leaving $\mathbf{w}_i$ unchanged once d_i becomes small. By this method, numerical problems arising as a result of a distance tending to zero may be avoided. However, the algorithm will still tend to the correct solution provided that the small residual corresponds to an interpolation point of the ℓ_1 solution. If this is not the case, then the solution will not be optimal, but will still be close to the optimal solution.

2.2 Possible problems

The most immediate problem that arises is how to ensure that there exists a point on the template which is situated along the direction vector given from each datum. Clearly in certain situations, there will not exist such a point — corresponding to the case where the direction vector lies within the tangent plane of the template in the region of the datum. In such a situation there would seem to be two possible recourses available.

(1) Ignore these data.
(2) Choose the point on the template that is closest to the line though the datum defined by the direction vector.

It has been found through empirical results that provided the problem only occurs on certain iterations rather than as a result of poor choice of the direction vectors associated with the template, ignoring the problem data is the better option. Use of the second option has been found to prevent convergence of the algorithm.

3 Algorithm

The algorithm to implement this technique consists of two sub-algorithms, each related to a specific section of the main algorithm. These sub-algorithms are

(1) the constrained closest point problem,
(2) the calculation of a new transformation estimate.

3.1 Constrained closest point problem

For each data point $\mathbf{x}_i$, this problem is that of finding u_i and v_i such that the constraint

$$\widehat{\mathbf{x}} - \mathbf{f}(u, v) = d\mathbf{w}, \tag{3.1}$$

is satisfied (subscripts dropped for clarity). Expanding this equation, we obtain

$$\begin{pmatrix} \widehat{x} \\ \widehat{y} \\ \widehat{z} \end{pmatrix} - \begin{pmatrix} f \\ g \\ h \end{pmatrix} - d \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = 0.$$

If we pre-multiply this equation by $\mathbf{a}^{\mathrm{T}}$, we obtain

$$\mathbf{a}^{\mathrm{T}}\widehat{\mathbf{x}} - \mathbf{a}^{\mathrm{T}}\mathbf{f}(u, v) - d\mathbf{a}^{\mathrm{T}}\mathbf{w} = 0. \tag{3.2}$$

Thus, by choosing $\mathbf{a}$ to be orthogonal to $\mathbf{w}$, we are able to eliminate d from equation (3.2). Similarly, if we multiply equation (3.1) by $\mathbf{b}$ we obtain the equation

$$\mathbf{b}^{\mathrm{T}}\widehat{\mathbf{x}} - \mathbf{b}^{\mathrm{T}}\mathbf{f}(u,v) - d\mathbf{b}^{\mathrm{T}}\mathbf{w} = 0.$$

We may thereby reduce the system (3.1) to that of two (nonlinear) equations in two unknowns (u and v). This system can then be solved by adopting a Newton-type method. Our problem has been reduced to that of solving

$$F(u,v) = [\mathbf{a} : \mathbf{b}]^{\mathrm{T}}(\widehat{\mathbf{x}} - \mathbf{f}(u,v)) = 0,$$

which has derivative

$$\nabla_{u,v}F = -[\mathbf{a} : \mathbf{b}]^{\mathrm{T}}(\nabla_u \mathbf{f} : \nabla_v \mathbf{f}),$$

by means of Newton's method which involves adopting an iterative approach and solving

$$\nabla_{u,v}F \begin{pmatrix} \delta u \\ \delta v \end{pmatrix} = -F(u,v), \tag{3.3}$$

at each stage to obtain better estimates $u + \delta u$ and $v + \delta v$. The quantities $F(u,v)$ and $\nabla_{u,v}F$ are straightforward to calculate as they arise directly from the explicit parameterisation of the template.

All that remains is the choice of $\mathbf{a}$ and $\mathbf{b}$. We obtain these vectors by taking the cross product of $\mathbf{w}$ with two arbitrary vectors — resulting in two vectors which are orthogonal to $\mathbf{v}$. More generally, the vectors $\mathbf{a}$ and $\mathbf{b}$ should be chosen to ensure that the system (3.3) is well-conditioned.

3.2 Updating the transformation estimate

The method we adopt to obtain an update of the transformation parameters is the Gauss-Newton method. This involves solving, at each iteration, the problem

$$J\delta\mathbf{t} = -\mathbf{d}, \tag{3.4}$$

in the ℓ_1 sense, where J is the Jacobian matrix of partial derivatives with entries $J_{ij} = \nabla_{t_j} d_i$. The estimate of the optimal transformation parameters is then updated according to

$$\mathbf{t} = \mathbf{t} + \delta\mathbf{t}.$$

Thus, since the distances $\mathbf{d}$ are obtained from the constrained closest point subproblem, we are left with the task of calculating the Jacobian matrix. For each datum, from equation (3.1), we have that

$$\widehat{\mathbf{x}}(\mathbf{t}) - \mathbf{f}(u(\mathbf{t}), v(\mathbf{t})) = \mathbf{w}d(u(\mathbf{t}), v(\mathbf{t})),$$

where we have explicitly included the dependency of the distance d on the location parameters U. Differentiating and rearranging, we obtain

$$\nabla_{\mathbf{t}}\widehat{\mathbf{x}} = \mathbf{w}\nabla_{\mathbf{t}}d + \nabla_U \mathbf{f}\nabla_{\mathbf{t}}U.$$

This is equivalent to the form

$$\nabla_{\mathbf{t}}\widehat{\mathbf{x}} = [\mathbf{w} : \nabla_U \mathbf{f}] \begin{pmatrix} \nabla_{\mathbf{t}}d \\ \nabla_{\mathbf{t}}U \end{pmatrix}.$$

Therefore,

$$J \equiv \nabla_{\mathbf{t}} d = \mathbf{e}_1^{\mathrm{T}} [\mathbf{w} : \nabla_{\mathbf{u}} \mathbf{f}]^{-1} \nabla_{\mathbf{t}} \widehat{\mathbf{x}},$$

where $\mathbf{e}_1$ is the first component vector. Having obtained the Jacobian matrix J and the distance vector $\mathbf{d}$, we are now in a position to solve the system (3.4) in order to update our estimate of the optimal transformation parameters $\mathbf{t}$.

We note that using the traditional orthogonal distances can lead to problems since calculation of the Jacobian matrix involves division of each row by the corresponding orthogonal distance — leading to exacerbation of rounding errors and possible division by zero especially in the ℓ_1 case.

4 Numerical results

In this section, we present two example to illustrate the techniques presented in this paper. In the first, we have a small number of data which we wish to match to a given plane. In the second, we have a larger number of data and we wish to match them to a cylinder. In both cases, although analytical expressions are available to obtain the constrained closest points on the templates, we nonetheless utilise the method presented above in order to test its effectiveness.

4.1 Simple problem

Here we describe the problem of matching a representative set of 8 data onto the plane defined as

$$\mathbf{f}(u, v) = u \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + v \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Since this problem is rank deficient if we use all six possible transformation parameters, we restrict ourselves to using a translation in the z-direction and rotations about the x and y axes.

Having three degrees of freedom, we might expect to obtain an optimal ℓ_1 solution which interpolates 3 of the data. However, as we shall see, this is unattainable in general and we can, in fact, only expect interpolation at two points. As Watson states [6], in such a situation, the rate of convergence can be unacceptably slow. This is found to be the case. It can be seen that not only is the convergence slow, but an optimal solution

Iteration	norm(residuals)	norm(update)
1	0.6662	4.9901e-02
5	0.3008	3.5716e-04
10	0.3007	8.8545e-06
50	0.3006	9.1533e-06
100	0.3008	3.8514e-04

TAB. 1 Progress of the Gauss-Newton method for planar data.

is never obtained, with the objective function $\|\mathbf{d}\|_1$ increasing occasionally.

To ensure convergence, a simple line-search algorithm was adopted which searches along the direction obtained from the Gauss-Newton step for the maximum reduction in the objective function. This modification affects convergence in 3 iterations.

4.2 A more challenging problem

As a more challenging problem, we consider the matching of a set of 128 data which supposedly represent a cylinder but which contain 8 wild points. The cylinder is parametrised by u and v as

$$\mathbf{f}(u,v) = \begin{pmatrix} \cos u \\ \sin u \\ v \end{pmatrix},$$

resulting in a cylinder with unit radius oriented along the z-axis. Again, the problem of matching the data onto this model is rank deficient. The rank deficiencies occur due to rotations about the z-axis and translations along the z-axis. As such, we omit these possible transformations.

Although we might initially expect to interpolate 4 data points at an optimal ℓ_1 solution, we find that in fact only two are guaranteed, although if a third point lies within two radii of one of these two points, then three points can be guaranteed. Typically, this will occur when the data is representative. For the data set we are considering, we expect three interpolation points due to the data representing the cylinder and in fact at the optimal solution, three interpolation points are obtained. In fact, the "missing" interpolation has the effect of slowing convergence of the Gauss-Newton method considerably so that in 100 iterations, the algorithm had not been deemed to converge. However, by the introduction of a simple line-search method, the algorithm converged in five iterations as displayed in Table 2.

Iteration	norm(residuals)	norm(update)
1	0.9654	5.6796e-03
2	0.9559	6.2932e-04
3	0.9557	1.0141e-04
4	0.9557	2.5812e-07
5	0.9557	4.4006e-14

TAB. 2 Progress of the Gauss-Newton method for cylindrical data using a line-search.

5 Conclusions

This paper has shown how perceived problems in ℓ_1 template matching can be avoided by use of the so-called "method of directional constraints". In this method, the closest point on the template along a given direction vector is calculated in order to obtain the residuals between data and template. By then altering this direction vector to be the normal to the surface at that projected point, the algorithm progresses to the expected ℓ_1 solution. Problems regarding undefined quotients are avoided by no longer updating

the direction vectors corresponding to a datum when the residual associated with that point is below a certain tolerance.

This work forms part of a larger project to consider novel approaches to ill-conditioned problems in metrology. It is hoped that the work presented in this paper will aid in the resolution of rank-deficient systems and ill-conditioned systems by altering the usual orthogonal distances to be these directional constraints, which should remove some of the rank deficiency.

As an example, consider the template matching problem where the template to be matched is an infinite cylinder with axis along the z-axis. Using typical template matching algorithms, this problem is rank deficient by two at the solution due to the possible translation in the z-axis and the possible rotation about the z-axis. By introducing these directional constraints, the rotational rank deficiency is almost completely resolved (there are now two possible rotations to obtain the optimal matching rather than the infinite number previously).

The use of the ℓ_1 norm is also being used to attempt and resolve any rank deficiencies and ill-conditioning present in the problem. This is achieved by ensuring that any local deviations from the template (caused by, for example, wear) are "ignored" so that regions of local deviations might be compared. This will then result in a resolution of the uncertainty in the transformation parameters.

Bibliography

1. B. P. Butler, A. B. Forbes, and P. M. Harris. Algorithms for geometric tolerance assessment. Technical Report DITC 228/94, National Physical Laboratory, Teddington, UK, 1994.
2. V. Jovanovski. Three-dimensional Imaging and Analysis of the Morpohology of Oral Structures from Co-ordinate Data. Ph.D. Thesis, Department of Conservative Dentistry, St Bartholomew's and the Royal London, School of Computing and Dentistry, Queen Mary and Westfield College, London, UK, 1999.
3. D. A. Turner. The approximation of Cartesian coordinate data by parametric orthogonal distance regression. Ph.D. Thesis, School of Computing and Mathematics, University of Huddersfield, UK, 1999.
4. D. A. Turner. Least squares profile matching using directional constraints. Preprint, 2001.
5. G. A. Watson. *Approximation Theory and Numerical Methods.* Wiley, New York, US, 1980.
6. G. A. Watson. On curve and surface fitting by minimizing the ℓ_1 norm of orthogonal distances. Preprint.
7. D. S. Zwick. A planar minimax algorithm for analysis of coordinate measurements Advances in Computational Mathematics, **2**:4, 1994, 375–391.

A bootstrap algorithm for mixture models and interval data in inter-comparisons

P. Ciarlini and G. Regoliosi

Istituto per le Applicazioni del Calcolo "M.Picone", CNR, Roma, Italy

F. Pavese

Istituto di Metrologia "G.Colonnetti", Torino, Italy

Abstract

To combine the information from several laboratories to output a representative value x_r and its probability distribution function is the main aim of an inter-comparison in Metrology. Here, the proposed procedure identifies a simple model for this probability function, by taking into account only the probability interval estimates as a measure of the uncertainty in each laboratory. A mixture density model is chosen to characterize the stochastic variability of the inter-comparison population considered as a whole. The bootstrap method is applied to approximate the distribution function of the comparison output in an automatic way.

1 Introduction

The "mise en pratique" of the Mutual Recognition Arrangement (MRA), issued by national metrological Institutions in 1999, prompted new studies and projects in Metrology mainly concerning the inter-laboratory comparisons area.

Recently, considerable effort has been devoted to finalise the problem of the choice of a suitable statistical procedure to summarise inter-comparison data. The problem solution is influenced by both metrological and statistical considerations, but it can also depend on the physical quantity under comparison.

Some of the critical issues now emerging are related to several different reasons. For instance, the statistical information supplied by each laboratory is synthetic, since it comes from a data reduction process performed on several experimental datasets. In each laboratory, assumptions and statistical reduction procedures may be different and sometimes not fully documented or the *a priori* information on the original data may be insufficient to define a "credible" probability distribution function (pdf) for output quantities of the inter-comparison.

The use of the whole sets of original data from each laboratory might be an unfeasible approach in the inter-comparison case, due to the unavailability of all needed data or to practical reasons. At present, the practice is to supply synthetic information x_i by each participant to the inter-comparison and to use a location estimator to output the representative value.

Efforts should be given to improving the reliability of inter-comparison results by asking for the use of any *a priori* information and of its "credibility" to go ahead, towards the direct estimation of the output of the comparison, x_r.

This paper proposes the identification of a solution without resorting to the synthetic values and its point estimates of the standard uncertainty, but only to the probability interval estimates as the measure of the uncertainty. This approach consists of two parts: a modelling procedure to identify a simple mixture model able to approximate the stochastic variability of the inter-comparison population as a whole; a parametric Monte Carlo algorithm to automatically estimate the probability distribution of the output x_r and any accuracy measures at a prescribed precision.

The concept of a mixture of distribution functions occurs when a population made up of distinct subgroups is sampled, for example, in biostatistics, when it is required to measure certain characteristics in natural populations of a particular species. In an inter-comparison each participant constitutes a subgroup.

The Monte Carlo method, based on the principle of mimicking sampling behaviour, can always compute a numerical solution in an automatic way, also when the required analytic calculations may not be simple. If the Monte Carlo method is applied with the principle of substitution (of the unknown probability function with a probability model estimated from the given sample), the approach is known as *the bootstrap* approach [4] and is already used in Metrology [2]. In [1] the case of a multivariate normal mixture model is considered and the standard errors are estimated by means of the parametric bootstrap. The present algorithm will be applied to a thermometric inter-comparison, where data cannot be assumed to be normally distributed.

2 Data structure of an inter-comparison with interval data

The number, N, of laboratories involved in an inter-comparison is typically small. In the i-th laboratory, the $(\xi_1^{(i)}, \ldots, \xi_k^{(i)})$ measurements are supposed to pertain to a single probability distribution function, say $F_i(\Lambda)$, where Λ is the parameter vector, that may be partially unknown. The measurements are statistically analysed and reduced to provide to the comparison the synthetic value x_i and its uncertainty u_i at 95% confidence level, or a 95% uncertainty interval (95%CI): $((x_1, u_1) \ldots, (x_N, u_N))$.

In this work the uncertainty is considered as "a 95%CI rather than as a multiple of the standard deviation" (see 4.3.4 in [6]). Then an aim of an inter-comparison is to combine the input data in the labs to characterise a representative value of the inter-comparison, i.e., the random variable θ and its pdf F. Hence a good estimate of the 95%CI for θ can be obtained if the output pdf F is a simple known function, describing the stochastic variability of the inter-comparison data. In other cases a suitable approximation of the expected value $E_F[X] = \int x dF(x)$ could be accepted to output the reference value x_r. The inter-comparison data structure is summarised here in terms of interval estimates:

INPUT Sample — Each one of the N participants originates a 95%CI that is one element of the inter-comparison sample:

$$\{[u_{il}, u_{iu}], i = 1, \ldots, N\}. \tag{2.1}$$

Here no value x_i in the interval $[u_{il}, u_{iu}]$ is chosen as representative; possible information on F_i (such as limited or unlimited support, symmetric or not) should be added. If a laboratory does not supply any information on the pdf, the uniform distribution is assumed.

Comparison OUTPUT — It includes the representative value and its 95%CI

$$(\hat{\theta}, [\epsilon_l, \epsilon_u]). \tag{2.2}$$

In many inter-comparisons, the differences to θ are also defined: $(y_i, [w_{il}, w_{iu}])$, where $y_i = x_i - \hat{\theta}, i = 1, \ldots, N$.

3 A classical approach to inter-comparisons

Let us recall the solution to the inter-comparison problem through the traditional estimator, the weighted mean. It is a location statistic that combines several measures and their standard uncertainties $(x_i, u_i)_{i=1}^N$. It provides the following estimate for θ,

$$\theta_w = u_w^2 \sum_{j=1}^{N} \frac{x_j}{u_j^2}, \quad u_w^2 = \left(\sum_{j=1}^{N} \frac{1}{u_j^2} \right)^{-1}, \tag{3.1}$$

and the following symmetric 95%CI,

$$\theta_w \pm k u_w, \tag{3.2}$$

where the coverage factor k is taken as the value $t_{N-1,0.95}$ of the Student distribution, N being small. In this approach, each x_i is viewed as an unbiased estimate of the laboratory mean value and the random variable θ_w is defined to be a linear combination of N independent random variables $X_1, \ldots, X_N$, where $\{x_1, \ldots, x_N\}$ is an observed sample. θ_w is supposed to be asymptotically normally distributed [6]. This estimator can be correctly adopted to solve an inter-comparison problem if the assumption of the homogeneity of the data is valid. This is equivalent to saying that, after considering the extent of the real effect and bias in each laboratory, the laboratories yield on the average the same value, so that the differences between the estimates are entirely due to random error. In this case, the selected estimator θ_w appropriately estimates θ and (3.2) accurately estimates its 95%CI.

Obstacles to applying this approach to a key-comparison have been discussed in [3]. The "credibility" of the representative values x_i, and of their uncertainty can critically affect the accuracy of the estimate of the representative value x_r. Moreover, the peculiar characteristics of a typical inter-comparison sample ((1) its very limited size, from a statistical point of view, (2) different experimental methods, used in each laboratory) often imply that the statistical assumptions are not satisfied, as for example in several thermometric cases. Indeed, the first characteristic implies that the Central Limit Theorem and the asymptotic theory do not hold. Then the normal distribution cannot be properly used to infer the estimates in (3.2).

Another example of the inadequacy of the weighted mean approach is when some laboratories provide data affected by bias, resulting from skewed distributions underlying their measurements. The symmetric confidence interval of (3.2) cannot be considered an

accurate approximation[1] of the true one, since it does not adjust for the skewness. Finally, it is necessary to point out that the homogeneity condition among the laboratories must be assured in some sense, otherwise it would be impossible to attempt to the computation of any summary estimate and its associated uncertainty.

4 The approach based on interval data

4.1 The mixture density function

This paper proposes to construct a simple model for the output pdf, and to estimate its expected value θ without requiring strong assumptions such as N large or each F_i normal. This approach enables us to compute the probability interval of the output value in terms of the identified density in each laboratory. The stochastic variability of the population of inter-comparison data is directly considered in the modelling approach as a whole, by means of a so-called mixture distribution model [5]. This model, being a linear superposition of several (say N) component densities, appears to be suitable from a computational point of view and can be embedded in a bootstrap algorithm to simulate several data needed to predict the output quantities.

In an inter-comparison, let us suppose that a density function $f_i(x;\Lambda^{(i)})$ is assumed for the i-th laboratory, then the following density mixture is identified to model the output pdf, where the parameter vector is $\Lambda = (\Lambda^{(1)},\dots,\Lambda^{(N)})$ and given weights $\pi_i \geq 0, i = 1,\dots,N$, have summation normalised to one:

$$g(x;\Lambda) = \sum_{i=1}^{N} \pi_i f_i(x;\Lambda^{(i)}). \tag{4.1}$$

To compute the output as estimate of the expected value of the mixture, $\theta = E_{G(\Lambda)}[X]$, the probability function $G(\Lambda)$, corresponding to the density in (4.1), must be known. When some laboratory provides only partial information on a pdf, we propose to identify its experimental variability by one of the following simple probabilistic models: uniform, normal or triangular pdf (right or left or symmetric triangular). Indeed, in thermometric experiments these three probabilistic models can represent several common stochastic variabilities for measurements, such as a limited or unlimited support, symmetric or not.

We want the mixture parameters to be estimated by means of the *INPUT Sample*, (2.1), as required in a bootstrap approach. Let us call I_i the *probability interval* to which the 100% measurements of the laboratory are supposed to pertain. For the uniform and the triangular types, $\Lambda^{(i)}$ parameters are defined to be the extremes of $I_i = [\lambda_{il}, \lambda_{iu}]$. For the normal model the parameters are the mean x_i and the variance u_i, while I_i becomes $(-\infty, +\infty)$.

A right triangular pdf (RT), a left triangular pdf (LT) or symmetric triangular pdf (ST) is chosen according to the position where the maximum of the probability density occurs, i.e., one extreme or the middle point of I.

[1] A 95% CI $[\epsilon_l, \epsilon_u]$ for θ is defined to be accurate if the following holds for every possible value for θ: $\text{Prob}_G\{\theta \geq \epsilon_u\} = 0.025$ and $\text{Prob}_G\{\theta \leq \epsilon_l\} = 0.025$

To compute the two components of the vector $\Lambda^{(i)} = (\lambda_{il}, \lambda_{iu})^T$ given the i-th input interval, a 0.025% portion of probability mass is added outside of each extreme, according to the supplied density shape. For example, if the ST density is chosen, the parameters are computed by:

$$\lambda_{il} = (0.89u_{il} - 0.11u_{iu})/0.78 \quad \lambda_{iu} = (0.89u_{iu} - 0.11u_{il})/0.78.$$

The mixture weights could be used to associate a degree of "credibility" to each laboratory. Then the choice $\pi_i = 1/N, i = 1, \ldots, N$, implies that every laboratory equally contributes to the inter-comparison.

When the mixture $G(\hat{\Lambda})$ is completely identified, it can be used to simulate data and to approximate the output value in the Monte Carlo algorithm.

4.2 The bootstrap algorithm

To avoid integral computations to estimate θ and its variance, the Monte Carlo method is commonly used to approximate them within a given precision. Since the *parametric* bootstrap approach does resampling from a parametric distribution model, in this case the mixture model $G(\hat{\Lambda})$, is adopted to approximate the following distribution,

$$H(x) = \text{Prob}_{\hat{G}}\{\theta^* \leq x\}. \tag{4.2}$$

The Monte Carlo method simulates a sufficiently high number B of data θ^* from $\hat{G} = G(\hat{\Lambda})$, to compute,

$$H(x)^{(B)} = \frac{1}{B}\sum_{b=1}^{B} \Pi\{\theta_b^* \leq x\}, \tag{4.3}$$

where the function $\Pi\{A\}$ is the indicator function of the set A. With probability one, it is known that the Monte Carlo approximation converges to the true value as $B \to \infty$. The Monte Carlo algorithm has been developed for a mixture density to estimate the comparison output. A hierarchical resampling strategy is used to reproduce the hierarchical variability in the inter-comparison population, throughout the following steps:

(1) (a) Choose at random an index, say k, of k-th laboratory by randomly resampling with replacement from the set $\{1, \ldots, N\}$

$$K \sim Prob\{K = k\} = \pi_i.$$

(b) Given k, generate, at random from the selected F_k of the distribution, a bootstrap value θ^* in $[\lambda_{kl}, \lambda_{ku}]$.

Repeat Step 1 B times to simulate the full bootstrap sample $\theta_1^*, \ldots, \theta_B^*$.

(2) Approximate the bootstrap mixture distribution as in (4.3) to compute:

— the bootstrap estimate of the expected mean

$$\hat{\theta_B^*} = \frac{1}{B}\sum_{b=1}^{B} \theta_b^*, \tag{4.4}$$

Lab1 (-0.05; 0.15) [-0.347, 0.247]	Lab2 (0.03; 0.30) [-0.564, 0.624]
Lab3 (0.18; 0.15) [-0.117, 0.477]	Lab4 (0.04; 0.15) [-0.257, 0.337]
Lab5 (0.71; 0.15) [0.413, 1.007]	Lab6 (-0.01; 0.15) [-0.307, 0.287]
Lab7 (-0.03; 0.15) [-0.327, 0.267]	

TAB. 1. Inter-comparison of 7 laboratories [7]: point estimates and simulated interval data.

— the bootstrap standard deviation: $Sd_B^* = \left(\frac{1}{B-1} \sum_{b=1}^{B} (\theta_b^* - \hat{\theta}_B^*)^2 \right)^{1/2}$,

— the 95%CI $[\epsilon_l^*, \epsilon_u^*]$, where the two extremes are computed as the α-th quantile [2] ($\alpha = 0.025$) of the bootstrap distribution $H_{Boot}^B(\alpha))^{-1} = q_B^{*\alpha}$, hence $\varepsilon_l^* = q_B^{*\alpha}$ and $\varepsilon_u^* = q_B^{*(1-\alpha)}$.

In Step 1b) the inverse transformation method has been used for simulating a random variable X having a continuous distribution F_k. For example, $X = F_k^{-1}(U)$, for a $U(\lambda_{kl}, \lambda_{ku})$ random variable. In Step 2 the bootstrap CI has been computed by means of the percentile method (see footnote). However, when the normal distribution is involved in the mixture, the *t-bootstrap* method gives more appropriate results [4]. To determine B in approximating the bootstrap confidence interval the coefficient of variation [4] can be used. The value of B is increased until the coefficient of variation cv of the sample quantile approaches the given precision δ_0. Indeed, from a metrological point of view, it appears easier to choose δ_0 instead of B as stopping rule in Step 1.

We would like to have also an automatic tool to investigate how well every laboratory contributes to the comparison, or to detect the possible presence of heterogeneous data. Here the concept of jackknife-after-bootstrap has been adopted to compute the mean and the bootstrap 95%CI. It is simply obtained by the following algorithm:

— for $i = 1, ..., N$, leave out the i-th lab and compute $\hat{\theta}_B^*(-i)$ and $q_B^*(-i)$,

— compare the N jackknife estimates to detect outlier values.

5 An application in thermometry

The proposed method is shown applied to an inter-comparison of Temperature Fixed Points, involving N =7 laboratories [7]. Each lab provided data x_i with the 95% standard uncertainty (Table 1: first item).

The second item (square brackets in the same table) represent the interval data generated with (3.2), that used to perform this simulated example. Since no specific pdf was supplied, the mixture distribution density has been constructed assuming the uniform type for each participant and equal weights. The parameters of every uniform density was computed using interval data, and the obtained mixture density was used in the resampling step of the algorithm to compute the representative value and its

[2]The percentile method of a statistics θ, based on B bootstrap samples, simply gives for a α-percentile $q_B^{*\alpha}$ = {(αB)th largest for θ_b^*}

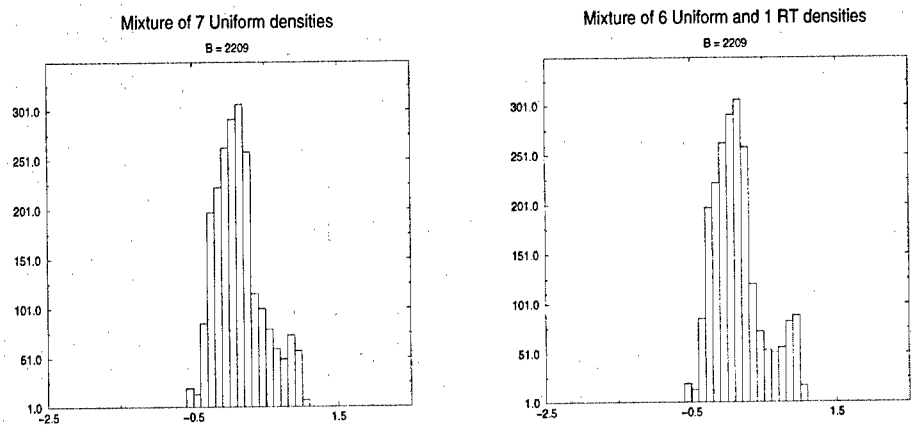


FIG. 1. Bootstrap histograms B =2209: left–mixture of 7 uniform distributions; right–mixture of 6 ST plus one RT density for Lab$i$.

probability interval with $\delta_0 = 0.05$. In Figure 1 (left) the bootstrap histogram, that approximates the mixture density, shows a bimodal behaviour. The computations are obtained for $\delta_0 = 0.05$ or $B = 2209$: $\hat{\theta}^* = 0.14$, bootstrap standard deviation Sd^*=0.33, 95%CI [-0.35, 0.92].

The proposed algorithm was also applied with a mixture of seven normal densities, and the results are $\hat{\theta}^* = 0.13$, $Sd^* = 0.43$, bootstrap 95%CI [-0.61, 1.1] for B =4752. The effect of assuming unlimited symmetric distributions to model the output pdf results in a wider 95%CI for a mixture of normal densities.

By comparing the jackknife results in Table 2, Lab5 appears to supply unusual values. To directly consider this behaviour in the inter-comparison, a mixture of six uniform densities plus a RT density, identifying Lab5, has been constructed. The approximated bootstrap distribution is displayed in Fig.1 (left), with bootstrap estimates, $\hat{\theta}^* = 0.15$, standard deviation $Sd^* = 0.35$ and [-0.35, 0.96] for the Bootstrap 95%CI, obtained for $B = 2209$.

6 Conclusions

The problem of the inter-comparison data has been described, and a new approach has been proposed. It is based on the uncertainty estimates, that should be provided by each Laboratory as interval estimate at 95% confidence level together with information, also partial, on the probability function. The constructive procedure directly characterises the stochastic variability of the reference value of the inter-comparison, by means of a mixture density model. The result of an inter-comparison is then viewed as a random variable, not directly measured, being the output of a complex process, that involves measures, statistical information and metrological considerations. These considerations suggest us constructing a mixture, with weights π_i to take into account each participating laboratory according to its credibility.

Lab1	0.34	[-0.45, 0.92]	Lab2	0.32	[-0.31, 0.94]
Lab3	0.34	[-0.40, 0.91]	Lab4	0.34	[-0.35, 0.92]
Lab5	0.23	[-0.42, 0.48]	Lab6	0.34	[-0.36, 0.95]
Lab7	0.34	[-0.42, 0.92]			

TAB. 2. Jackknife-after-bootstrap estimates. Standard deviation and 95%CI for mixture of 6 uniform densities ($B = 1000$): in the ith item, Lab$i$ is left out.

The parametric bootstrap approach has been adopted to estimate in a simple and automatic way the inter-comparison output, where information, even partial, on the probability hierarchical data of the participating laboratories, have been taken into account.

Also with a limited number of laboratories, the method can be applied, as it is shown in the thermal example, where ($N = 7$) and the experimental conditions implied to adopt skewed distributions. The automatic jackknife method of detecting the heterogeneous data succeeded in revealing an unusual value. To take into account this condition, a mixture of six uniform densities plus an RT density to identify Lab5 could be better used. The choice of equal weights emphasises that all the standards have equally contributed to the inter-comparison.

The bootstrap procedure, completely developed for a class of five simple distribution functions often used in thermal metrology, could be adapted to consider other distributions, when the synthetic data information provided by the laboratories, as summarised in Section 2, allow to compute the mixture parameters.

Bibliography

1. K. E. Basford, D. R. Greenway, G. J. McLachlan and D. Peel, *Standard errors of fitted component means of normal mixtures.* Computational Statistics **12**, 1–17, 1997.
2. P. Ciarlini et al., *Non-parametric bootstrap with application to metrological data.* In: Advanced Mathematical Tools in Metrology, Series on Advances in mathematics for applied sciences, **16**, Singapore, Ciarlini, Cox, Monaco, Pavese eds., World Scientific, 219–230, 1994.
3. M. Cox, *A discussion of approaches for determining a reference value in the analysis of key-comparison data.* In Advanced Mathematical and Computational Tools in Metrology IV, Series on Advances in mathematics for applied sciences, **53**, Singapore, Ciarlini, Cox, Pavese, Richter Eds , World Scientific, 45–65, 2000.
4. B. Efron and R. Tibshirani, An Introduction to the Bootstrap, Chapman and Hall, London, 1993.
5. B. S. Everitt, Finite Mixture Distributions, Chapman and Hall, London, 1981.
6. ISO, Guide to the Expression of Uncertainty in Measurement, Geneva, Switzerland, 1995.
7. F. Pavese, Monograph 84/4 of Bureau International des Poids et Mesures, BIPM Sevres, 1984.

Efficient algorithms for structured self-calibration problems

Alistair B. Forbes

National Physical Laboratory, Teddington, Middlesex TW11 0LW, UK.
alistair.forbes@npl.co.uk

Abstract

Self-calibration techniques have been used extensively in co-ordinate metrology. At their most developed, they are able to extract all systematic error behaviour associated with the measuring instrument as well as determining the geometry of the artefact being measured. However, this is generally at the expense of introducing extra parameters leading to moderately large observation matrices. Fortunately, these matrices tend to have sparse, block structure in which the nonzero elements are confined to much smaller submatrices. This structure can be exploited either in direct approaches in which QR factorisations are performed or in iterative algorithms which depend on matrix-vector multiplications. In this paper, we describe self-calibration approaches associated with high accuracy, dimensional assessment by co-ordinate measuring systems, highlighting how the associated optimisation problems can be presented compactly and solved efficiently. The self-calibration techniques lead to uncertainties significantly smaller than can be expected from standard methods.

1 Introduction

An important activity in metrology is the calibration of instruments and artefacts. Calibration defines a rule which converts the values output by the instrument's sensor(s) to values that can be related to the appropriate standard (SI or derived) units. Importantly, to these calibrated values it is required to assign uncertainties that reliably take into account the uncertainties of all quantities that have an influence. As a consequence, the size and complexity of the computational tasks associated with the data analysis can be significant, even for instruments that appear to be of simple design and operation. It is thus beneficial to design and implement algorithms that are efficient with respect to computation and memory. Fortunately, many of the calibration problems give rise to systems of equations with a well defined sparsity structure.

The rest of this paper is organised as follows. In Section 2 we review least squares approaches to calibration problems and go on to describe self-calibration problems in co-ordinate metrology in Section 3. Sections 4 and 5 describe solution methods for two types of sparsity structure. Our concluding remarks are given in Section 6.

2 Least squares solution to calibration problems

In many calibration problems, the observation equations involving measurements y_i

can be expressed as $y_i = \phi_i(\mathbf{a}) + \epsilon_i$, where ϕ_i is a function depending on parameters $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}}$ specifying the behaviour of the instrument, and ϵ_i represents random measurement error. For a set of measurement data $\{y_i\}_1^m$, best estimates $\mathbf{a}^*$ of the calibration parameters $\mathbf{a}$ are determined by solving

$$\min_{\mathbf{a}} \sum_{i=1}^{m} f_i^2(\mathbf{a}) = \mathbf{f}^{\mathrm{T}}\mathbf{f}, \tag{2.1}$$

where $f_i(\mathbf{a}) = y_i - \phi_i(\mathbf{a})$. The most common approach to solving this problem is derived from the Gauss-Newton algorithm; see, for example, [5]. If $\mathbf{a}$ is an estimate of the solution and J is the *Jacobian matrix* defined at $\mathbf{a}$ by $J_{ij} = \partial f_i/\partial a_j$, then an updated estimate of the solution is $\mathbf{a} + \mathbf{p}$, where $\mathbf{p}$ solves the Jacobian system

$$J\mathbf{p} = -\mathbf{f},$$

in the least squares sense. Starting with an appropriate initial estimate of $\mathbf{a}$, these steps are repeated until convergence criteria are met.

A numerically stable method of solving the Jacobian system is to find a factorisation $J = QR$, where Q is an $m \times n$ orthogonal matrix and R is an upper-triangular matrix of order n (see, e.g., [1, 6]). The solution $\mathbf{p}$ is determined efficiently by solving the upper-triangular system

$$R\mathbf{p} = -Q^{\mathrm{T}}\mathbf{f},$$

using back substitution. The matrix Q can be constructed using either Householder reflections, which process the Jacobian matrix a column at a time, or Givens plane rotations, which process the matrix row-wise. For either approach the orthogonal factorisation requires $O(mn^2)$ operations.

An alternative to the direct approaches to solve matrix equations is to use iterative procedures based on conjugate gradients. The advantage of these approaches is that they involve only matrix-vector multiplications and for sparse matrices these multiplications can be made efficient. In particular, the LSQR algorithm of Paigė and Saunders [7] implements an iterative approach to solving linear least squares problems.

Often, linear equality constraints on the parameters of the form $C^{\mathrm{T}}\mathbf{a} = \mathbf{c}$, where C is an $n \times p$ matrix, $p < n$, are required to eliminate degrees of freedom in the problem. However, we can use orthogonal projections to eliminate these constraints. Suppose C is of full column rank and has QR factorisation

$$C = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} S \\ 0 \end{bmatrix},$$

where V_1 and V_2, respectively, are the first p and last $n - p$ columns of the orthogonal factor V. If $\mathbf{a}_0$ is a solution of $C^{\mathrm{T}}\mathbf{a} = \mathbf{c}$, then for any $(n-p)$-vector $\tilde{\mathbf{a}}$, $\mathbf{a} = \mathbf{a}_0 + V_2\tilde{\mathbf{a}}$ automatically satisfies the constraints and the optimisation problem can be reformulated as the unconstrained non-linear least squares problem

$$\min_{\tilde{\mathbf{a}}} \sum_{i=1}^{m} f_i^2(\mathbf{a}_0 + V_2\tilde{\mathbf{a}}),$$

involving the reduced set of parameters $\tilde{\mathbf{a}}$. We note that the associated Jacobian matrix is simply $\tilde{J} = JV_2$, where $J_{ij} = \partial f_i / \partial a_j$, as before.

Unfortunately, even if J has structure $\tilde{J} = JV_2$ could be full. For indirect approaches, this is of little consequence since the matrix-vector multiplications can be formed in two stages (e.g., $\mathbf{y} = V_2\mathbf{x}$, $\mathbf{z} = J\mathbf{y}$) each of which can be implemented efficiently. For a direct approach, it may be possible to implement the constraints in such a way as to minimise the amount of fill-in during the orthogonal factorisation stage.

3 Self-calibration problems in co-ordinate metrology

Co-ordinate metrology is concerned with defining the geometry of two and three dimensional artefacts from measurements of the co-ordinates of points related to the surface of the artefacts. It is a key discipline in quality and process control in manufacturing industy. In a (conventional) co-ordinate measuring machine (CMM) with three mutually orthogonal linear axes, the position of the probe tip centre is inferred from scale readings on each of the three machine axes. In practice, CMMs have imperfect geometry with respect to the straightness of the axes, the squareness of pairs of axes and rotations describing roll, pitch and yaw, and these systematic errors have to be taken into account if the accuracy potential of the CMM is to be more fully realised. Two approaches can be adopted to nullify the effect of these systematic errors. The first – *error mapping* – involves performing a set of experiments to characterise as completely as possible the error behaviour of the instrument and then use error correction software to produce more accurate co-ordinate estimates. The disadvantages of this approach are, firstly, the set of experiments is expensive to perform and, secondly and more importantly, the error behaviour of the CMM is likely to drift so that, for example, an error correction valid on Monday will only be partially valid on Friday and may be of limited value a month later. The second approach – *self-calibration* – attempts to use any approximate symmetries, rotational or translational, of the artefact so that systematic errors associated with the measuring system are identified as part of the measurement process [4]. The advantage of this method is that the effect of systematic error behaviour of the instrument is cancelled out and the accuracy of the measurements are limited only by the smaller, random component.

3.1 Calibration of reference artefacts in 2-dimensions

As an example, we consider the accurate calibration of 2-dimensional artefacts by a two dimensional CMM. The artefacts define the location of targets nominally aligned in a grid pattern. Let $\mathbf{y}_j$, $j = 1, \ldots, n_Y$, be the locations of the targets in a fixed frame of reference, and let

$$\mathbf{y}_{j,k} = T(\mathbf{y}_j, \mathbf{t}_k)$$

be the location of the jth target in the kth measuring position. Here, the roto-translation T is specified by three parameters $\mathbf{t}$ defining the translation vector and angle of rotation.

We suppose the systematic error of the two dimensional CMM can be expressed as

$$\mathbf{x}^* = \mathbf{x}^*(\mathbf{x}, \mathbf{b}) = \mathbf{x} + \mathbf{e}(\mathbf{x}, \mathbf{b}),$$

where $\mathbf{x}^*$ are the true point co-ordinates, $\mathbf{x}$ are the indicated point co-ordinates output by the machine and $\mathbf{e}(\mathbf{x}, \mathbf{b})$ is the error correction term depending on $\mathbf{x}$ and error parameters $\mathbf{b}$. For instance, suppose the model describes scale and orthogonality errors so that

$$x^* = x(1 + b_1) + y(1 + b_2) \sin b_3, \quad y^* = y(1 + b_2) \cos b_3.$$

If $\mathbf{x}_i$ is the measurement of the jth target with the artefact in the kth position then the associated observation equation is

$$\mathbf{x}_i + \mathbf{e}(\mathbf{x}_i, \mathbf{b}) = \mathbf{y}_{j,k} + \boldsymbol{\epsilon}_i. \tag{3.1}$$

Given a set of such measurements $\{\mathbf{x}_i\}_1^{m_X}$ and associated index functions $(j(i), k(i))$ specifying the targets and artefact positions, estimates of the model parameters can be determined by solving a non-linear least squares problem

$$\min_{\{\mathbf{y}_j\},\{\mathbf{t}_k\},\mathbf{b}} \sum_{i=1}^{m_X} \mathbf{f}_i^{\mathrm{T}} \mathbf{f}_i,$$

where $\mathbf{f}_i(\mathbf{y}_{j(i)}, \mathbf{t}_{k(i)}, \mathbf{b}) = \mathbf{x}_i + \mathbf{e}(\mathbf{x}_i, \mathbf{b}) - \mathbf{y}_{j,k}$.

The model involves three sets of the parameters: the target locations $\{\mathbf{y}_j\}$, transformation parameters $\{\mathbf{t}_k\}$ and the error parameters $\mathbf{b}$. Each observation equation depends on only one target and one transformation, so that the Jacobian matrix J of partial derivatives can be ordered to have a block-angular structure [2]

$$J = \begin{bmatrix} K_1 & & & J_1 \\ & K_2 & & J_2 \\ & & \ddots & \vdots \\ & & K_{m_X} & J_{m_X} \end{bmatrix},$$

where K_j corresponds to the parameters $\mathbf{y}_j$ and the border blocks $\{J_j\}$ correspond to the border parameters $\mathbf{a} = \{\{\mathbf{t}_k\}, \mathbf{b}\}$. The frame of reference for the targets $\{\mathbf{y}_j\}$ can be specified by applying three appropriate linear equality constraints on the transformation parameters $\{\mathbf{t}_k\}$.

While scale and orthogonality errors are often major contributors to the systematic error behaviour of a CMM, there is no guarantee nor does experience show that they explain the full extent of the behaviour. For this reason, more comprehensive models have been developed [3, 9]. However, they all depend on the approximation of actual behaviour by empirical functions such as polynomials and the adequacy of the approximation is often difficult and expensive to evaluate. However, if we always rotate and translate the artefact according to the symmetries of the reference artefact so that the targets are always located (nominally) at a subset of a fixed grid of points in the CMM's working volume, then measurements are made at a finite number of machine locations. To the lth location we associate a machine error $\mathbf{e}_l$. If the ith measurement is made at the lth location then the observation equation corresponding to (3.1) is

$$\mathbf{x}_i + \mathbf{e}_l = \mathbf{y}_{j,k} + \boldsymbol{\epsilon}_i.$$

The advantage of this error model is that it entails no significant approximation: the

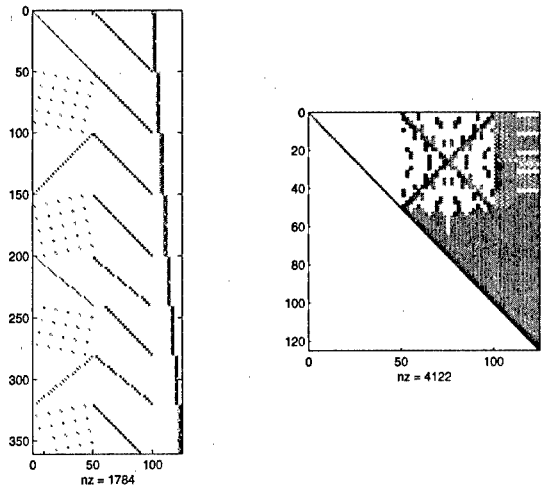


FIG. 1. Sparsity structure of the transpose of the Jacobian matrix associated with the measurement of a 5×5 hole plate in eight positions.

systematic errors are modelled exactly. An apparent disadvantage is that there are likely to be as many error parameters as target parameters giving rise to a sparsity structure in the Jacobian matrix for which direct, structure-exploiting methods provide relatively minor efficiency gains. Figure 1 shows on the left the sparsity structure of the Jacobian matrix J associated with the measurement of a 5×5 hole plate in eight positions, the first four corresponding to rotations by 0, 90, 180 and 270 degrees, the second four incorporating a translation as well as a rotation. In each position the location of the targets $\mathbf{y}_j$ are measured in order. The nonzero elements of the matrix are represented by a dot. The first (second) 50 columns correspond to the derivatives with respect to the machine error parameters $\mathbf{e}_l$ (target parameters $\mathbf{y}_j$) and the last 24 correspond to the eight sets of transformation parameters $\mathbf{t}_k$. On the right the sparsity structure of the triangular factor of J is illustrated and shows the substantial fill-in that occurs.

In the next two sections, we describe approaches for dealing efficiently with block-angular and more general sparse-block structure.

4 Algorithms for block-angular systems

We consider non-linear least squares problems where the optimisation parameters can be partitioned into two sets $\boldsymbol{\eta} = \{\mathbf{y}_j\}_1^{n_Y}$ and $\mathbf{a}$, and such that each observation equation involves $\mathbf{a}$ and at most one set of parameters $\mathbf{y}_j$. Corresponding to (2.1), we have instead an objective function of the form

$$F(\boldsymbol{\eta}, \mathbf{a}) = \mathbf{f}_0^{\mathrm{T}}(\mathbf{a})\mathbf{f}_0(\mathbf{a}) \quad + \quad \sum_j \mathbf{f}_j^{\mathrm{T}}(\mathbf{y}_j, \mathbf{a})\mathbf{f}_j(\mathbf{y}_j, \mathbf{a}).$$

The associated Jacobian matrix J and its triangular factor R can be arranged to have the form

$$J = \begin{bmatrix} K_1 & & & J_1 \\ & K_2 & & J_2 \\ & & \ddots & \vdots \\ & & K_{n_Y} & J_{n_Y} \\ & & & J_0 \end{bmatrix}, \quad R = \begin{bmatrix} R_1 & & & B_1 \\ & R_2 & & B_2 \\ & & \ddots & \vdots \\ & & R_{n_Y} & B_{n_Y} \\ & & & B_0 \end{bmatrix}.$$

The nonzero blocks of the matrix R can be stored compactly in a vector $\mathbf{r}$, row by row.

Efficient updating strategies for such triangular factors have been incorporated into a non-linear least-squares solver to deal with block-angular problems. It is assumed that the Jacobian matrix is composed of n_B blocks of rows, with the ith block depending on at most one set of parameters $\mathbf{y}_j$, $j = j(i)$. The user is required to supply a function and gradient evaluation module that given $\boldsymbol{\eta}$, $\mathbf{a}$ and $1 \leq i \leq n_B$, returns $j = j(i)$ and

$$\begin{aligned} &\mathbf{f}_i(\mathbf{a}),\ J_i, && j = 0, \\ &\mathbf{f}_i(\mathbf{y}_j, \mathbf{a}),\ J_i,\ K_i, && j > 0. \end{aligned}$$

For each i, the triangular factor and righthand side vector is updated by the ith block of rows:

$$\begin{bmatrix} R_{j(i)} & B_{j(i)} \\ K_i & J_i \end{bmatrix} \mapsto \begin{bmatrix} R_{j(i)} & B_{j(i)} \\ \mathbf{0} & \check{J}_i \end{bmatrix}, \quad \begin{bmatrix} R_0 \\ \check{J}_i \end{bmatrix} \mapsto \begin{bmatrix} R_0 \\ \mathbf{0} \end{bmatrix}.$$

Linear equality constraints on the border parameters $\mathbf{a}$ implemented using the orthogonal projection approach can be incorporated by setting $J_i := J_i V_2$ at the appropriate stage.

5 Algorithms for sparse-block matrices

Let $m \times n$ matrix S be composed of n_B submatrices S_k of dimension $m_k \times n_k$. We assume that S_k is stored (column-wise or row-wise) as a column vector $\mathbf{s}_k$. The information in S can be encoded in a column vector $\mathbf{s}_I$ and an indexing set I_S such that $I_S(1:5,k) = (i_k, j_k, m_k, n_k, l_k)$ where (i_k, j_k) specifies the location of $S_k(1,1)$ in S and l_k indicates that $\mathbf{s}_k = \mathbf{s}_I(l_k : l_k + m_k n_k - 1)$. Blocks of such matrices can be easily represented by concatenating the $\mathbf{s}$-vectors and index matrices I_S and performing some trivial index modifications. Matrix-vector multiplications of the form $\mathbf{y} := \alpha S\mathbf{x} + \beta\mathbf{y}$ are easily implemented through a sequence of full matrix multiplications: $\mathbf{y} := \beta\mathbf{y}$, followed by

$$\mathbf{y}(i_k : i_k + m_k - 1) := \mathbf{y}(i_k : i_k + m_k - 1) + \alpha S_k \mathbf{x}(j_k : j_k + n_k - 1),$$

$k = 1, \ldots, n_B$. A similar scheme calculates $\mathbf{x} := \alpha S^{\mathrm{T}}\mathbf{y} + \beta\mathbf{x}$. The storage and multiplication scheme can be modified to take into account the type or structure of the submatrices S_k.

To implement linear equality constraints, it is required to perform matrix multiplication by a submatrix V_2 of the orthogonal factor of the constraint matrix C. A simple scheme can be implemented using the LAPACK routines DGEQRF (orthogonal factorisation) and DORMQR (matrix multiplication by an orthogonal matrix stored as a product of Householder matrices) [8].

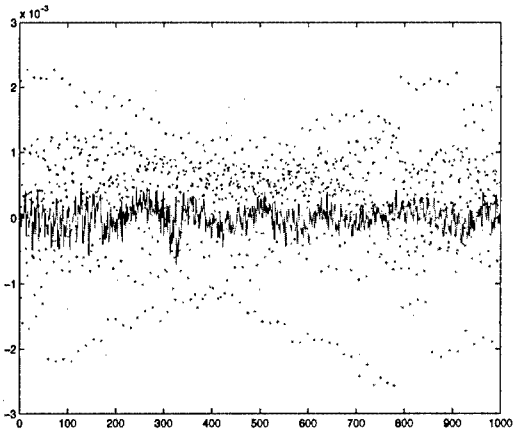


FIG. 2. Residual errors associated with the first 1000 observations for models a) with no error separation (dots) and b) with error separation.

We have implemented a non-linear least squares solver for sparse-block systems. The user is required to supply a module that takes as input the current estimate $\mathbf{a}$ of the optimisation parameters and outputs the function values $\mathbf{f}(\mathbf{a})$ and the Jacobian matrix stored in sparse-block form $\langle \mathbf{s}_I, I_S \rangle$. The solver implements a Gauss-Newton approach using the LSQR solver to find the Gauss-Newton step and caters in a straightforward way for linear equality constraints. The solver has been successfully tested in a number of self-calibration problems. For example, it was used recently in the calibration of a 13×13 grid of targets on a glass plate by a CMM with an optical probing system. The problem involved approximately 15,000 observation equations in over 800 optimisation parameters and was solved in a few tens of seconds using a standard laboratory PC (450 MHz). The advantage of the error separation model is illustrated in Figure 2 which shows the residual errors associated with the first 1000 observations for models a) with no error separation (dots) and b) with error separation. The fit for the error separation model is much superior. The practical metrological consequence of adopting the enhanced model is that uncertainties associated with the target locations can be reduced by a factor of five. Importantly, because the model is a realistic approximation of the measuring system, we can have confidence in the uncertainty estimates derived from the model.

6 Concluding remarks

The move to more accurate measurement systems has led to more comprehensive models of the measuring instrument and its interaction with the physical quantity being measured. These models include parameters that describe properties of the instrument and those of the measurand. The aim of self-calibration experiments is to determine as much as possible about both sets of parameters from a set of measurement experiments. For models with a small to modest set of parameters, a full matrix approach may be acceptable. For larger systems, exploitation of sparsity structure in the defining equations is

highly desirable and often a stark necessity if the computations are to be made in an acceptable time using the computing resources to hand. The exploitation of block-angular structure has been well-known and well-used in some areas of metrology. The supporting numerical technology based on structured orthogonal factorisations is mature, compact and easily implemented using standard numerical linear algebra. However, these techniques could be applied more widely in metrology, making feasible approaches that have to be rejected if full matrix methods only are to be used.

The use of sparse matrix techniques is relatively rare within metrology. We have attempted to show here that in self-calibration problems in dimensional metrology, they allow us to develop improved models that provide vastly superior fits to the data, with corresponding improvements in the evaluated uncertainties in the fitted parameters. The supporting numerical technology is maturing and accessible.

Acknowledgements: This work has been supported by the Department of Trade and Industry's National Measurement System Software Support for Metrology Programme and undertaken by a project team at the Centre for Mathematics and Scientific Software, National Physical Laboratory. The author is particularly thankful to Maurice Cox, Peter Harris and Ian Smith for their contributions.

Bibliography

1. A. Björck. *Numerical Methods for Least Squares Problems.* SIAM, Philadelphia, 1996.
2. M. G. Cox. The least-squares solution of linear equations with block-angular observation matrix. In M. G. Cox and S. Hammarling, editors, *Advances in Reliable Numerical Computation*, pages 227–240. Oxford University Press, 1989.
3. M. G. Cox, A. B. Forbes, P. M. Harris, and G. N. Peggs. Experimental design in determining the parametric errors of CMMs. In V. Chiles and D. Jenkinson, editors, *Laser Metrology and Machine Performance IV*, pages 13–22, Southampton, 1999. WIT Press.
4. A. B. Forbes and I. M. Smith. Self-calibration and error separation techniques in metrology. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology V*, pages 149–163, Singapore, 2001. World Scientific.
5. P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Academic Press, London, 1981.
6. G. H. Golub and C. F. Van Loan. *Matrix Computations.* John Hopkins University Press, Baltimore, third edition, 1996.
7. C. C. Paige and M. A. Saunders. LSQR: and algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1), 1982.
8. SIAM, Philadelphia. *The LAPACK Users' Guide*, third edition, 1999.
9. G. Zhang, R. Ouyang, B. Lu, R. Hocken, R. Veale, and A. Donmez. A displacement method for machine geometry calibration. *Annals of the CIRP*, 37:515–518, 1998.

On measurement uncertainties derived from "Metrological Statistics"

Michael Grabe

Am Hasselteich 5, 38104 Braunschweig, Germany.
michael.grabe@ptb.de

Abstract

As measurement uncertainties are closely tied up with error models, it might be of interest to review a model, which the author assigns to "Metrological Statistics". Given that the random errors are normally distributed, the experimentalist could either refer to B.L. Welch's concept of "effective degrees of freedom" or to the multidimensional Fisher-Wishart distribution density. In the first case, different numbers of repeated measurements are admissible, in the latter it is strictly required to have equal numbers of repeated measurements. In error propagation, however, only the latter mode of action opens up the possibility of designing confidence intervals according to Student and confidence ellipsoids according to Hotelling. Another point of view, closely linked to the choice of the numbers of repeated measurements, refers to the customary practice of attributing equal rights to statistical expectations and empirical estimators. However, the Fisher-Wishart distribution density suggests using only the information which is realistically accessible to experimentalists, namely empirical estimators. For the handling of unknown systematic errors, either the existence of a (rectangular) distribution density may be assumed or, and this is proposed here, they may be classified as time-constant quantities, biasing expectations and suspending a lot of tools and procedures of error calculus well-established otherwise.

1 Introduction

The joint propagation of random errors and unknown systematic errors currently places the experimentalist in the following dilemma.

In regard to the propagation of *random errors*, there are, at least in principle, two different choices. If one is willing to accept *unequal numbers* of repeated measurements of the physical quantities to be combined within a given function, one has, in order to express the influence of random errors, to resort to B. L. Welch's sophisticated concept of so-called *numbers of effective degrees of freedom* [8]. However, this procedure is tied up with difficulties: it is restricted to independent variables.

Though B. L. Welch's concept completely exhausts the information implied in measured data, unfortunately, from a metrological point of view, it is cumbersome to handle and obstructs the view to existing simpler procedures. On the other hand, if the experimentalist preferred *equal numbers* of repeated measurements, he would — if need be — have to give away part of his information, namely that which is carried by the

excessive numbers of repeated measurements of the variables involved. Up to now, the disregarding of excessive numbers is regarded as unfavourable. In spite of this view, just this precaution opens up a toolbox of applied statistics hitherto closed to metrologists, as only with equal numbers of repeated measurements, is the experimentalist in a position to call upon the standard model of statistics for jointly normally distributed random variables, i.e. the Fisher-Wishart density [3]. The advantages gained in that way outweigh by far the "lost information", as relatively few repeated measurements of experimental set-ups, operating in a stationary mode, are able to locate accurately the respective physical quantities. After all, in error propagation the experimentalist may define confidence intervals according to Student (Gosset) including *any number* of variables. In least squares, he may even establish multidimensional confidence intervals, and last but not least, certain problems of classical error calculus, such as the Fisher–Behrens problems no longer arise.

In regard to the interpretation and propagation of *unknown systematic errors*, the situation is not simpler. Let us assume that an unknown systematic error f, constant in time, is confined to an interval of the kind[1]

$$-f_s \leq f \leq f_s, \quad f_s \geq 0. \tag{1.1}$$

Now, the experimentalist may either assign a postulated probabilty density to f, usually a rectangular density [7],

$$p(f) = \frac{1}{2f_s}, \tag{1.2}$$

or he may set without exception

$$f = \text{constant}, \tag{1.3}$$

where f lies anywhere within (1.1). The latter interpretation introduces biased estimators, leading to a break-down of many procedures of error calculus otherwise well-established.

Seen mathematically, both interpretations should be justified. In the case of (1.2), the combination of random and systematic errors should be carried out geometrically, in the case of (1.3), arithmetically. Regarding (1.3), the author suggests adding linearly Student's confidence intervals to appropriately designed worst-case estimates of the propagated systematic errors, and no probability statements should be associated with so-defined overall uncertainties.

2 Error propagation

The fundamental error equations of *Metrological Statistics* are given as follows [4]. Let x_0 designate *the true value* of the physical quantity x to be measured. Furthermore, let ε_l be the random error and f_x = constant the unknown systematic error corresponding

[1]Should the interval be unsymmetrical to zero, it could be symmetrized by subtracting the halved sum of the upper and lower boundary — the same quantity would have to be subtracted from the data.

to (1.1). We then have

$$x_l = x_0 + \varepsilon_l + f_x, \quad l = 1, \ldots, n. \tag{2.1}$$

Let $\mu_x = x_0 + f_x$ be the expectation of the random variable $X = \{x_1, x_2, \ldots, x_n\}$, so that the x_l are some of its realizations. We then find

$$x_l = \mu_x + \varepsilon_l, \quad l = 1, \ldots, n. \tag{2.2}$$

Furthermore, let $\bar{x} = 1/n \sum_{l=1}^{n} x_l$ denote the arithmetic mean. We then have the useful identities

$$x_l = x_0 + (x_l - \mu_x) + f_x, \qquad \bar{x} = x_0 + (\bar{x} - \mu_x) + f_x. \tag{2.3}$$

While the arithmetic mean is biased, the empirical variance

$$s_x^2 = \frac{1}{n-1} \sum_{l=1}^{n} (x_l - \bar{x})^2 \tag{2.4}$$

is not. For the time being, let us consider just two quantities to be measured, x and y. As robust and simple uncertainty assessments are *a matter of linearization*, the overall uncertainty u_ϕ of a given function $\phi(x, y)$ is proposed to be [5],

$$u_\phi = \frac{t_{S,P}(n-1)}{\sqrt{n}} \sqrt{\left(\frac{\partial \phi}{\partial \bar{x}}\right)^2 s_x^2 + 2\left(\frac{\partial \phi}{\partial \bar{x}}\right)\left(\frac{\partial \phi}{\partial \bar{y}}\right) s_{xy} + \left(\frac{\partial \phi}{\partial \bar{y}}\right)^2 s_y^2} + \left|\frac{\partial \phi}{\partial \bar{x}}\right| f_{s,x} + \left|\frac{\partial \phi}{\partial \bar{y}}\right| f_{s,y} \tag{2.5}$$

where $t_{S,P}(n-1)$ is the Student-factor corresponding to a confidence level P. We distinctly see how the empirical covariance

$$s_{xy} = \frac{1}{n-1} \sum_{l=1}^{n} (x_l - \bar{x})(y_l - \bar{y})$$

enters the empirical variance of the $\phi(x_l, y_l)$; $l = 1, \ldots, n$, given by

$$s_\phi^2 = \frac{1}{n-1} \sum_{l=1}^{n} [\phi(x_l, y_l) - \phi(\bar{x}, \bar{y})]^2 = \left(\frac{\partial \phi}{\partial \bar{x}}\right)^2 s_x^2 + 2\left(\frac{\partial \phi}{\partial \bar{x}}\right)\left(\frac{\partial \phi}{\partial \bar{y}}\right) s_{xy} + \left(\frac{\partial \phi}{\partial \bar{y}}\right)^2 s_y^2.$$

The final result

$$\phi(\bar{x}, \bar{y}) \pm u_\phi \tag{2.6}$$

is expected to localize the true value $\phi(x_0, y_0)$ with "reasonable certainty" — but no proper confidence statement should be added, as u_ϕ is a mixture of a statistical and a non statistical component. The last term in (2.5) may overestimate the uncertainty, on the other hand linearization errors have been negleted. After all, this uncertainty statement should fulfill the prerequiste to be safe, robust and simple.

If there are m quantities to be measured, we replace the notation $\bar{x}, \bar{y}$ by $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m$. Then the overall uncertainty u_ϕ of the final result

$$\phi(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m) \pm u_\phi$$

is given by

$$u_\phi = \frac{t_{S,P}\,(n-1)}{\sqrt{n}} \sqrt{\sum_{i,j}^{m} \frac{\partial\phi}{\partial\bar{x}_i}\frac{\partial\phi}{\partial\bar{x}_j} s_{ij}} + \sum_{i=1}^{m} \left|\frac{\partial\phi}{\partial\bar{x}_i}\right| f_{s,i}. \tag{2.7}$$

When (2.5) and (2.7) are compared, it becomes obvious that the proposed formalism of error propagation works like a building kit, perspicuous and easy to handle. There are arguments against (2.7), in particular that an experimentalist who wishes to design his uncertainties in this way, would have to know *the complete set of repeated measurements*, in other words, the *complete* empirical variance-covariance matrix

$$s = (s_{ij}), \qquad i,j = 1,2,\ldots,m, \tag{2.8}$$

of the input data. Arguably, this is true, but in the days of computers and the internet such a challenge should no longer be apt to provoke difficulties worth mentioning. Another argument, that (2.7) might overestimate overall uncertainties, should be judged in view of the unique role of metrology in science. Standing "between" theory and experiment, metrology pursues the idea to localize reliably the value of the physical quantity in question.

3 Least squares

Let

$$A\beta \approx x \tag{3.1}$$

be a linear system of equations to be adjusted. Here, A designates the $m \times r$ design matrix of rank r, β the $r \times 1$ vector of unknowns and, finally, x the $m \times 1$ vector of the observations or input data. We assume $m > r$. The idea of least squares is of purely geometrical origin.

In what follows, A^T denotes the transpose of A. The idea is to project the vector x by means of a projection operator

$$P = A\left(A^T A\right)^{-1} A^T \tag{3.2}$$

orthogonally onto the column space of the matrix A, and the result is

$$\bar{\beta} = \left(A^T A\right)^{-1} A^T x. \tag{3.3}$$

As the solution vector $\bar{\beta}$ is linear in the input data, the transfer of (2.7) to its components $\bar{\beta}_k$, $k = 1,\ldots,r$, is straightforward.

Clearly, the orthogonal projection is in no way dependent on the error model implied. In contrast to this, the latter turns out to be crucial in regard to uncertainty assessments. Let us consider a set of *single observations*

$$x_i = x_{0,i} + \varepsilon_i + f_i = x_{0,i} + (x_i - \mu_i) + f_i, \quad i = 1,\ldots,m, \tag{3.4}$$

being the input data, where $E\{X_i\} = \mu_i$. Writing (3.4) in vector form, we have

$$x = x_0 + (x - \mu) + f \tag{3.5}$$

where

$$x = (x_1, x_2, \dots, x_m)^T, \qquad x_0 = (x_{0,1}, x_{0,2}, \dots, x_{0,m})^T,$$
$$\mu = (\mu_1, \mu_2, \dots, \mu_m)^T, \qquad f = (f_1, f_2, \dots, f_m)^T, \quad -f_{s,i} \le f_i \le f_{s,i}.$$

Given equal variances $\sigma^2 = E\left\{(X_i - \mu_i)^2\right\}$, the minimized sum $Q_{\min}$ of squared residuals of the adjusted system (3.1) should yield, according to quite familiar procedures, an estimator $s^2 \approx \sigma^2$. However, from

$$Q_{\min} = (x - Px)^T (x - Px),$$

we obtain something different, namely

$$E\{Q_{\min}\} = \sigma^2 (m - r) + f^T f - f^T P f. \tag{3.6}$$

As we see, even the simplest of all associated least squares procedures breaks down, should the model of time-constant unknown systematic errors be accepted. At the same time the related basic tool linked to $Q_{\min}$ and frequently used, namely *the test of consistency* of the input data based on the criterion

$$Q_{\min}/s^2 \approx m - r$$

breaks down as well. Indeed, during many decades, time and again, the observation

$$Q_{\min}/s^2 \gg m - r$$

has stunned experimentalists [2], so that, in the adjustments of the fundamental physical constants, even the abolition of least squares has been considered [1]. However, in view of (3.6), these observations are understandable.

After all, a least squares adjustment of *biased* input data requires arithmetic means

$$\bar{x}_i = x_{0,i} + (\bar{x}_i - \mu_i) + f_i, \quad i = 1, \dots, m, \tag{3.7}$$

so that the empirical variances and covariances

$$s_{ij} = \frac{1}{n-1} \sum_{l=1}^{n} (x_{il} - \bar{x}_i)(x_{jl} - \bar{x}_j), \qquad s_{ii} = s_i^2, \tag{3.8}$$

are known *a priori*. Replacing (3.5) by

$$\bar{x} = x_0 + (\bar{x} - \mu) + f \tag{3.9}$$

instead of (3.3), we find

$$\bar{\beta} = \left(A^T A\right)^{-1} A^T \bar{x}. \tag{3.10}$$

A matter of similar concern refers to the break-down of the Gauss-Markoff theorem. In view of (3.9), the solution vector $\bar{\beta}$ is biased, so that the experimentalist is no longer in a position to obtain a weight-matrix from the variance-covariance matrix of the input vector $\bar{x}$. Consequently, simple, *optimized adjustments*, to which we are customarily used, must be ruled out. Nevertheless, we may multiply (3.1) from the left with any

non-singular weighting matrix, e.g. with a diagonal one,

$$G = \{g_1, g_2, \dots, g_m\}, \quad g_i = \frac{1}{u_{\bar{x}_i}}, \tag{3.11}$$

and adjust the weights g_i by trial and error in order to find the shortest possible uncertainty intervals. As has been shown, this method is also able to detect inconsistencies among the input data, [6]. Indeed, as a non-singular weight-matrix cannot shift the true solution vector β_0, we are allowed to proceed this way.

To assign uncertainties to the components $\bar{\beta}_k; k = 1, \dots, r$ of the solution vector $\bar{\beta}$, we refer to (2.7). To abbreviate the notation, we set in (3.10)

$$B = A\left(A^T A\right)^{-1} \tag{3.12}$$

where the elements of the matrix B will be designated by b_{ik}. Upon insertion of (3.9) into (3.10), we arrive at

$$\bar{\beta} = B^T x_0 + B^T(\bar{x} - \mu) + B^T f. \tag{3.13}$$

Evidently, $\beta_0 = B^T x_0$ is the true value of the estimator $\bar{\beta}$. Setting $\mu_{\bar{\beta}} = E\{\bar{\beta}\} = \beta_0 + B^T f$, we may define the *theoretical* variance-covariance matrix

$$E\left\{\left(\bar{\beta} - \mu_{\bar{\beta}}\right)\left(\bar{\beta} - \mu_{\bar{\beta}}\right)^T\right\},$$

which, however, remains numerically inaccessible. Consequently, the only thing we can do is to resort to the *empirical* variance-covariance matrix

$$s_{\bar{\beta}} = \left(s_{\bar{\beta}_k \bar{\beta}_{k'}}\right) = B^T s B, \quad k = 1, 2, \dots, r, \tag{3.14}$$

whose elements are given by

$$s_{\bar{\beta}_k \bar{\beta}_{k'}} = \sum_{i,j}^{m} b_{ik} b_{jk'} s_{ij}, \qquad s_{\bar{\beta}_k \bar{\beta}_k} = s^2_{\bar{\beta}_k}. \tag{3.15}$$

Clearly, the s_{ij} are the elements of the empirical variance-covariance matrix s of the input data, as has been stated in (2.8) and (3.8).

These procedures presuppose, as has been pointed out, *equal numbers* of repeated measurements within each of the m means (3.7). The components $\bar{\beta}_k$ of the solution vector may be written as

$$\bar{\beta}_k = \frac{1}{n}\sum_{l=1}^{n} \bar{\beta}_{kl} \quad \text{with} \quad \bar{\beta}_{kl} = \sum_{i=1}^{m} b_{ik} x_{il}; \quad k = 1, \dots, r. \tag{3.16}$$

Evidently, the $\bar{\beta}_{kl}$ are independent and normally distributed. Let $\mu_{\bar{\beta}_k}$ denote the expectations

$$\mu_{\bar{\beta}_k} = E\{\bar{\beta}_k\}, \quad k = 1, \dots, r \tag{3.17}$$

of the $\bar{\beta}_k$. Looking for just *any one* of the $\bar{\beta}_k$,

$$\bar{\beta}_k - \frac{t_{S,P}(n-1)}{\sqrt{n}} s_{\bar{\beta}_k} \le \mu_{\bar{\beta}_k} \le \bar{\beta}_k + \frac{t_{S,P}(n-1)}{\sqrt{n}} s_{\bar{\beta}_k} \tag{3.18}$$

is a confidence interval according to Student, where $t_{S,P}(n-1)$ is the Student-factor. This interval localizes $\mu_{\bar{\beta}_k}$ with confidence P.

The components of the third term on the right-hand side of (3.13) are given by

$$f_{\bar{\beta}_k} = \sum_{i=1}^{m} b_{ik} f_i, \quad k = 1, \ldots, r. \tag{3.19}$$

Worst-case estimates are

$$f_{s,\bar{\beta}_k} = \sum_{i=1}^{m} |b_{ik}| f_{s,i} l, \quad k = 1, \ldots, r. \tag{3.20}$$

After all, the overall uncertainties $u_{\bar{\beta}_k}$ of the components of the solution vector $\bar{\beta}$, considered and employed individually, are proposed to be

$$u_{\bar{\beta}_k} = \frac{t_{S,P}(n-1)}{\sqrt{n}} s_{\bar{\beta}_k} + f_{s,\bar{\beta}_k}, \quad k = 1, \ldots, r. \tag{3.21}$$

4 Uncertainty spaces

The component representation of (3.13),

$$\bar{\beta}_k = \beta_{0,k} + \sum_{i=1}^{m} b_{ik} (\bar{x}_i - \mu_i) + \sum_{i=1}^{m} b_{ik} f_i \tag{4.1}$$

reveals the couplings between the least squares estimators. Those due to random errors may be expressed by means of Hotelling's density [3]. The last term on the right-hand side of (4.1),

$$f_{\bar{\beta}_k} = \sum_{i=1}^{m} b_{ik} f_i, \quad k = 1, \ldots, r, \tag{4.2}$$

expresses the couplings due to systematic errors. The r components $f_{\bar{\beta}_k}$ map the m-dimensional hypercuboid

$$-f_{s,i} \leq f_i \leq f_{s,i}, \quad i = 1, \ldots, m, \tag{4.3}$$

onto the r-dimensional space, yielding a convex polytope. Both solids may be combined to an overall uncertainty space, resembling a "convex potato". Figures 1–3 show the confidence ellipsoid, the "security polytope" and the combination of both to an overall uncertainty space for the example of a least squares adjustment of a circle.

5 Conclusion

As computer simulations reveal, the approach presented here leads to measurement uncertainties safeguarding *physical objectivity* in the sense that uncertainty intervals reliably locate the values of the physical quantities in question. With such a distinct statement, the traceability of units and standards will certainly be maintained.

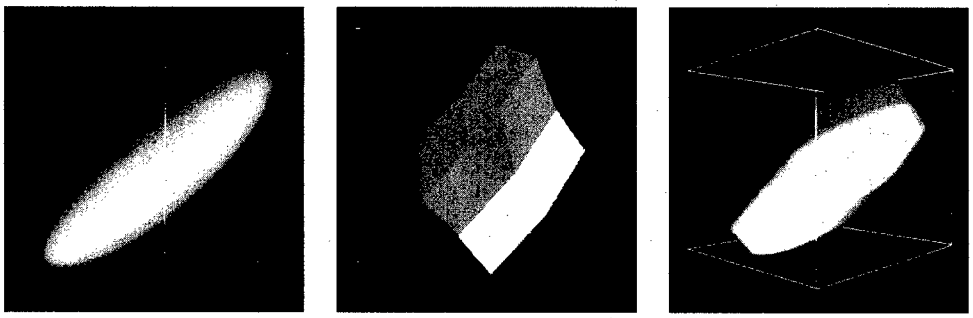


FIG. 1. Confidence ellipsoid, security polytope, overall uncertainty space resembling a "convex potato".

References

1. Bender, P.L., B. N. Taylor, E.R. Cohen, J.S. Thomas, P. Franken, and C. Eisenhart, *Should least squares adjustment of the fundamental constants be abolished?*, NBS Special Publication **343**, United States Department of Commerce, Washington D.C., 1971.
2. Cohen, E.R. and B.R. Taylor, *The 1986 adjustment of the fundamental physical constants*, CODATA BULLETIN Nr. **63** (1986).
3. Cramér, H., *Mathematical Methods of Statistics*, Princeton University Press, Princeton 1961.
4. Grabe, M., *Principles of "metrological statistics"*, metrologia **23** (1986/87) 213–219.
5. Grabe, M. *Estimation of measurement uncertainties, an alternative to the ISO-Guide*, metrologia **38** (2001) 97–106.
6. Grabe, M., An alternative algorithm for adjusting the fundamental physical constants, Physics Letters A **213** (1996) 125–137.
7. ISO, *Guide to the expression of uncertainty in measurement*, 1993. 1, Rue de Varambé, Boîte postale 56, CH-1211 Geneva 20, Switzerland.
8. Welch, B.L., *The generalization of Student's problem when several different population variances are involved*, Biometrika **34** (1947) 28–35.

l_1 and l_∞ ODR fitting of geometric elements

Hans-Peter Helfrich

Mathematisches Seminar der Landwirtschaftlichen Fakultät der Universität Bonn
helfrich@uni-bonn.de

Daniel S. Zwick

Wilcox Associates, Inc.
dzwick@wilcoxassoc.com

Abstract

We consider the fitting of geometric elements, such as lines, planes, circles, cones, and cylinders, in such a way that the sum of distances or the maximal distance from the element to the data points is minimized. We refer to this kind of distance based fitting as orthogonal distance regression or ODR. We present a *separation of variables* algorithm for l_1 and l_∞ ODR fitting of geometric elements. The algorithm is iterative and allows the element to be given in either implicit form $f(x,\beta)=0$ or in parametric form $x=g(t,\beta)$, where β is the vector of shape parameters, x is a 2- or 3-vector, and s is a vector of location parameters. The algorithm may even be applied in cases, such as with ellipses, in which a closed form expression for the distance is either not available or is difficult to compute. For l_1 and l_∞ fitting, the norm of the gradient is not available as a stopping criterion, as it is not continuous. We present a stopping criterion that handles both the l_1 and the l_∞ case, and is based on a suitable characterization of the stationary points.

1 Introduction

Let us be given N points $\{z_i\}_{i=1}^N \in \mathbb{R}^d$ and a geometric object S in

- *implicit form* $\{x : f(x,\beta)=0\}$ with a scalar function f, or
- *parametric form* $x=g(t,\beta)$ with a vector function g,

where the shape parameter vector $\beta \in C$ lies within a closed, convex subset C of $\mathbb{R}^m$. Denote by

$$\phi_i(\beta) = \inf\{\|z_i - x_i\|_2 : x_i \text{ on } S\}$$

the distance of the point z_i to the geometric object S. Let

$$\phi(\beta) = (\phi_1(\beta), \ldots, \phi_N(\beta))^T$$

be the distance vector with norm

$$\Phi(\beta) = \|\phi(\beta)\|,$$

where $\|\phi(\beta)\|$ denotes either the l_∞-norm

$$\Phi(\beta) = \max(\phi_1(\beta), \ldots, \phi_N(\beta))$$

or the l_1-norm

$$\Phi(\beta) = \sum_{i=1}^{N} \phi_i(\beta).$$

We consider the problem:

Find $\beta \in C$ and points $\{x_i\}_{i=1}^N$ on S such that $\Phi(\beta) = \|\phi(\beta)\|$ is minimal.

If the minimum is attained, each function $\phi_i(\beta) = \|z_i - x_i\|_2$ is minimal for the point $x_i \in S$. Then $z_i - x_i$ is orthogonal to S for interior points of S, hence the term "orthogonal distance regression" or "ODR".

Nonlinear l_1 ODR problems are treated in WATSON [10, 12]. A survey for linear problems is given in ZWICK [13].

As stated, the problem has dimension $Nd+m$. In typical metrology applications, the data set is very large so that a direct approach to the problem becomes computationally expensive. We use a *separation of variables* algorithm that was used in [2, 4] and TURNER [9] for the l_2 ODR problem. Each iteration of our algorithm consists of two steps. In the first step, the *foot points* $\{x_i\}_{i=1}^N$ on S, i.e., the location parameters, are calculated for a fixed parameter vector β. These d-dimensional subproblems can be efficiently handled by trust region methods [3].

In the second step, a first order approximation of $\phi_i(\beta)$ is employed, that can be given without explicit knowledge of the dependence of the optimal points $x_i(\beta)$ on β. At this stage, the norm of the correction to the parameter vector β is limited by a trust region strategy. The correction can be computed by solving a linear programming problem. For general nonlinear minimax problems such methods were proposed in MADSEN AND SCHJÆR–JACOBSEN [6], HALD AND MADSEN [1] and JÓNASSON AND K. MADSEN [5].

Our convergence analysis follows the general approach given in POWELL [8] and MORÉ [7]. But in order to handle the l_1 and l_∞ case we cannot use the norm of the gradient as a stopping or convergence criterion, since the gradient is not continuous. Moreover, a neccessary condition for a minimum is that the subgradient contains the zero functional, see, e.g., WATSON [11]. In order to overcome this difficulty, we introduce a replacement for the norm of the gradient that serves both as a stopping criterion and as an essential tool in the convergence proof.

2 The trust region algorithm

At each iteration of our algorithm we solve the low-dimensional subproblems (P_i) for $\beta = \beta_k$ for each fixed i, $i = 1, \ldots, N$:

Minimize $\|z_i - x_i\|_2$ subject to $f(x_i, \beta) = 0$ or $x_i = g(t_i, \beta)$.

In order to apply the trust region method to l_1 and l_∞ ODR we need a first order approximation $\psi_i(\beta, \alpha)$ to $\phi_i(\beta)$. With appropriate regularity assumptions, this can be computed without knowledge of the dependence of the optimal points $x_i(\beta)$ on β ([2], [4]). This means that the iterative improvement in β is *uncoupled* from the calculations of $x_i(\beta)$, whereby a true first order approximation of the objective function is attained.

In the case of the implicit form $f(x,\beta)=0$, the first order approximation $\phi_i(\beta+\alpha) = \psi_i(\beta,\alpha)+o(\alpha)$ is given by

$$\psi_i(\beta,\alpha) = \frac{\nabla_x f(x_i,\beta)^T(z_i-x_i)+\nabla_\beta f(x_i,\beta)^T\alpha}{\|\nabla_x f(x_i,\beta)\|_2}, \tag{2.1}$$

as a first order approximation to the signed distance $\pm\phi_i(\beta+\alpha)$. For the parametric form $x=g(t,\beta)$, we have

$$\psi_i(\beta,\alpha) = \|z_i-x_i\|_2 - \frac{(z_i-x_i)^T}{\|z_i-x_i\|_2} D_\beta g(x_i,\beta)\alpha. \tag{2.2}$$

Note that (2.1) makes sense even for points on the surface. For an orientable hypersurface in parametric form, the expression $\frac{(z_i-x_i)^T}{\|z_i-x_i\|_2}$ in (2.2) should be replaced by the unit normal for points on the surface.

Denote by

$$\psi(\beta) = (\psi_1(\beta),\ldots,\psi_N(\beta))^T$$

the vector of the linearized distances and let

$$\Psi(\beta,\alpha) = \|\psi(\beta,\alpha)\| - \|\phi(\beta)\|\,.$$

The main algorithm:

- Step 0: An initial $\beta_0\in\mathbb{R}^m$, a trust region radius $\Delta_0>0$, and constants $0<\mu<1$ and $0<\gamma<1<\overline{\gamma}$, $\overline{\Delta}$ are given. Set $k=0$.
- Step 1: Minimize $\Psi(\beta_k,\alpha)$ subject to $\|\alpha\|_2\le\Delta_k$ and $\beta_k+\alpha\in C$. Let α_k denote the solution with minimal norm.
- Step 2: If $\alpha_k=0$, stop.
- Step 3: Compute
$$\rho_k = \frac{\Phi(\beta_k+\alpha_k)-\Phi(\beta_k)}{\Psi(\beta_k,\alpha_k)}.$$
- Step 4:

  (1) *Successful step.* If $\rho_k\ge\mu$ set
  $$\beta_{k+1}=\beta_k+\alpha_k$$
  and choose Δ_{k+1} such that
  $$\Delta_k\le\Delta_{k+1}\le\min(\overline{\gamma}\Delta_k,\overline{\Delta}). \tag{2.3}$$

  (2) *Unsuccessful step.* Otherwise, set
  $$\beta_{k+1}=\beta_k \text{ and } 0<\Delta_{k+1}\le\gamma\Delta_k.$$
- Step 5: Increment k by one and go to Step 1.

3 Global convergence

In an abstract setting our problem may be formulated as

Minimize $\Phi(\beta)=\|\phi(\beta)\|$ *on a closed, convex set* C.

To solve this problem, at each stage of the iteration we solve the following constrained, linearized problem:

Minimize $\Psi(\beta, \alpha)$ *subject to* $\beta + \alpha \in C$ *and* $\|\alpha\| \leq \Delta$.

In order to get the linearization in our case, we solve the least distance subproblems (P_i), $i = 1, \ldots, N$, with a shape parameter β, and use (2.2), or (2.1).

For the purpose of characterizing stationary points, we introduce the quantity

$$\nabla_1(\beta) = -\inf\{\Psi(\beta, \alpha) \mid \|\alpha\| \leq 1,\ \beta + \alpha\ \in C\}.$$

Note that $\nabla_1(\beta) \geq 0$, since $\Psi(\beta, 0) = 0$.

By convexity, $\nabla_1(\beta) = 0$ implies that $\alpha = 0$ is a solution of the linearized minimization problem. MADSEN AND SCHJÆR-JACOBSEN [6] have shown that the latter condition is equivalent to a condition given therein for the functional to have a stationary point. In order to prove Theorem 3.3 we prove a lemma that was given in a similar form for the l_∞ case in MADSEN AND SCHJÆR-JACOBSEN [6] and JÓNASSON AND MADSEN [5]). We give a different proof that is applicable to both the l_1 and l_∞ cases.

Lemma 3.1 *Let* $\nabla_1(\beta) \geq \epsilon$ *and* $\Delta \leq \overline{\Delta}$. *For the solution of the linearized problem the estimate*

$$\Psi(\beta, \alpha) \leq -C\epsilon\Delta \tag{3.1}$$

holds, with a constant that depends only on ϵ *and* $\overline{\Delta}$.

Proof: According to the definition of $\nabla_1(\beta)$ and the continuity of Ψ there exists a feasible α_1 with $\|\alpha_1\| \leq 1$ such that

$$\Psi(\beta, \alpha_1) = -\epsilon.$$

Let $\alpha = t\alpha_1$, where $t = \min(1, \Delta)$. Since $\Psi(\beta, \alpha)$ is a convex function, we get

$$\Psi(\beta, \alpha) \leq (1 - t)\Psi(\beta, 0) + t\Psi(\beta, \alpha_1) = -t\epsilon.$$

Since

$$t \geq \Delta \min(1, 1/\overline{\Delta})$$

we get the conclusion with $C = \min(1, 1/\overline{\Delta})$. □

Proposition 3.2 *For a minimum point,*

$$\nabla_1(\beta) = 0$$

holds.

Proof: Assume the contrary, then $\nabla_1(\beta) = \epsilon > 0$ holds. According to the definition of $\Psi(\beta, \alpha)$ we have

$$\Phi(\beta + \alpha) = \Phi(\beta) + \Psi(\beta, \alpha) + o(\alpha).$$

By Lemma 3.1, we can find an α with $\|\alpha\| \leq \Delta$ such that (3.1) holds. As in the proof of the Lemma, we may conclude that

$$\Phi(\beta + t\alpha) \leq \Phi(\beta) - C\epsilon t\Delta + o(t\alpha)$$

for $0 < t \leq 1$. If we let $t \to 0$ we get a contradiction to the minimum property. □

Theorem 3.3 *Either the algorithm ends in a finite number of steps, or a sequence β_k is generated for which* $\liminf_{k\to\infty} \nabla_1(\beta_k) = 0$.

Proof: Assume the contrary. Then there exists $\epsilon > 0$ such that $\nabla_1(\beta_k) \geq \epsilon$ holds for all k. By the definition of ρ_k and the lemma, it follows that for a successful step

$$\phi(\beta_{k+1}) \leq \phi(\beta_k) - \mu C \epsilon \Delta_k$$

and by the updating rule for Δ_{k+1} we get

$$\Delta_{k+1} \leq c(\phi(\beta_{k+1}) - \phi(\beta_k)),$$

with $c = 1/(\mu C \epsilon)$. Combining this inequality with the updating rule for an unsuccessful step yields

$$\Delta_{k+1} \leq \gamma \Delta_k + c(\phi(\beta_{k+1}) - \phi(\beta_k)).$$

By summation and the monotonicity of $\phi(\beta_k)$ it follows that for all N

$$\sum_{k=0}^{N} \Delta_k \leq \frac{\Delta_0}{1-\gamma} + \frac{c}{1-\gamma}\phi(\beta_1).$$

Since this implies the convergence of $\sum \Delta_k$, we get $\lim \Delta_k = 0$. From $\|\beta_k\| \leq \Delta_k$ we obtain the convergence of β_k. From the definition of ρ_k it then follows that $\lim \rho_k = 1$. But then the updating rule (2.3) implies that eventually $\Delta_{k+1} \geq \Delta_k$, which gives a contradiction. □

Theorem 3.4 *(Global Convergence, cf.* MORÉ *[7],* POWELL *[8]) Assume that $\nabla_1(\beta)$ is uniformly continuous. Then either the algorithm ends in a finite number of steps, or a sequence β_k is generated for which*

$$\lim_{k\to\infty} \nabla_1(\beta_k) = 0.$$

Proof: Assume the contrary. Then there exists an ϵ_1 such that for each k_0 there exists a $k \geq k_0$ with

$$\nabla_1(\beta_k) \geq \epsilon_1.$$

By Theorem 3.3 we can find an index $l > k$ such that

$$\nabla_1(\beta_l) \leq \epsilon_1/2$$

(k_0 will be determined later). We choose the smallest such l. As in the proof of Theorem 3.3, it follows that for that a successful step with $k \leq i < l$,

$$\|\beta_{i+1} - \beta_i\| \leq \Delta_k \leq 2c_1(\phi(\beta_i) - \phi(\beta_{i+1})).$$

Clearly, this also holds for an unsuccessful step. This yields

$$\|\beta_l - \beta_k\| \leq 2c_1(\phi(\beta_k) - \phi(\beta_l)).$$

Since $\phi(\beta_i)$ converges by monotonicity, we can make $\|\beta_l - \beta_k\|$ arbitrarily small for large enough k_0 . By the uniform continuity of $\nabla_1(\beta)$ we infer

$$|\nabla_1(\beta_k) - \nabla_1(\beta_l)| < \epsilon_1/2,$$

which is a contradiction. □

4 A numerical example

As an illustrative example, we fit an ellipse to data, given as coordinate pairs in $\mathbb{R}^2$. There are 24 data points and five components to the shape parameter vector (i.e., $n = 2, d = 2, m = 5, N = 24$). We used a standard parameterization involving a center (x_0, y_0), the axes (a, b), and a rotation angle θ.

The output is shown below. The initial values for the parameters and the obtained parameters in three different norms are given in Table 1. In the l_2 case, we give as the error the root mean square error, in the l_1 case the *mean absolute deviation*, and in the l_∞ case the *maximum deviation*.

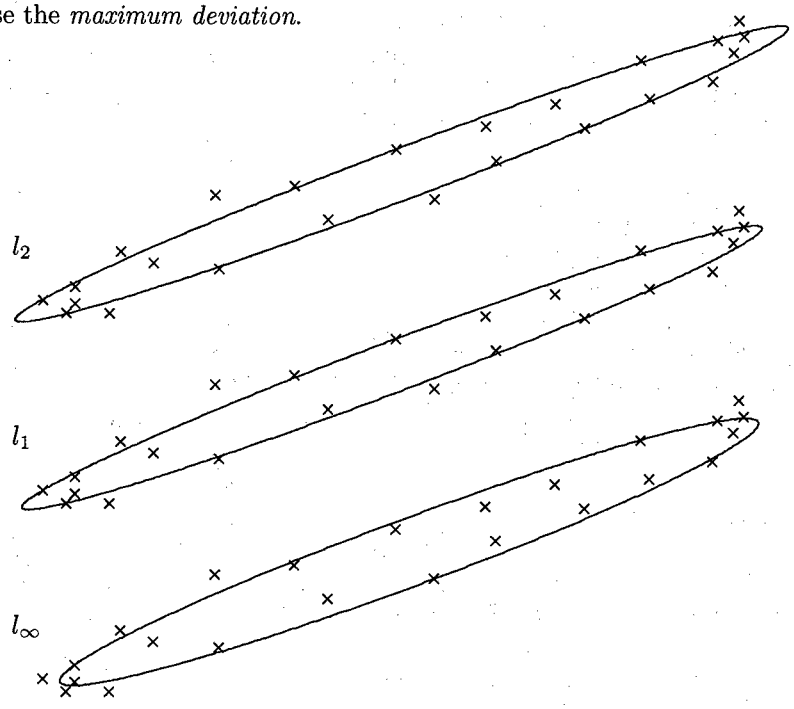


FIG. 1. l_2, l_1, and l_∞-Approximation.

	x_0	x_1	a	b	θ (degrees)	Error
Initial values	0.4989881	-1.4262126	4.6719913	0.4364267	20.75913	
l_2	0.6637511	-1.3987826	5.5124671	0.3376480	20.90124	0.11520
l_1	0.5368646	-1.4465520	5.2778061	0.3358224	20.88869	0.09047
l_∞	0.7694412	-1.3829474	4.9731226	0.4491259	20.66893	0.23489

TAB. 1. Parameters for different norms.

The number of iterations in each case was five or six. We note that the deviations for the best fit l_1 and l_∞ ellipses exhibit behavior typical to these norms: five of the data points lie on the best fit l_1 ellipse and there are six deviations of largest magnitude in the l_∞ case.

Bibliography

1. J. Hald and K. Madsen. Combined LP and Quasi-Newton methods for minimax optimization. *Mathematical Programming*, 20:49–62, 1981.
2. H.-P. Helfrich and D. Zwick. A trust region method for implicit orthogonal distance regression. *Numerical Algorithms*, 5:535–545, 1993.
3. H.-P. Helfrich and D. Zwick. Trust region algorithms for the nonlinear least distance problem. *Numerical Algorithms*, 9:171–179, 1995.
4. H.-P. Helfrich and D. Zwick. A trust region algorithm for parametric curve and surface fitting. *J. Comput. Appl. Math.*, 73:119–134, 1996.
5. K. Jónasson and K. Madsen. Corrected sequential linear programming for sparse minimax optimization. *BIT*, 34:372–387, 1994.
6. K. Madsen and H. Schjær–Jacobson. Linearly constrained minimax optimization. *Mathematical Programming*, 14:208–223, 1978.
7. J. J. Moré. Recent developments in algorithms and software for trust region methods. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical Programming Bonn 1982–The State of the Art*, pages 259–287. Springer, 1983.
8. M. J. D. Powell. Convergence properties of a class of minimization algorithms. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, editors, *Nonlinear Programming 2*, pages 1–27. Academic Press, 1975.
9. D. A. Turner, I. J. Anderson, J. C. Mason, M. G. Cox, and A. B. Forbes. An efficient separation-of-variables approach to parametric orthogonal distance regression. In P. Ciarlini, M. G. Cox, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology IV*, pages 246–255, Singapore, 2000. World Scientific.
10. G. A. Watson. The use of the l_1 norm in nonlinear errors-in-variables problems. In S. Van Huffel, editor, *Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling*, pages 183–192, Philadelphia, 1997. SIAM.
11. G. A. Watson. Choice of norms for data fitting and function approximation. *Acta Numerica*, pages 337–377, 1998.
12. G. A. Watson. Some robust methods for fitting parametrically defined curves and surfaces to measured data. In F. Pavese P. Ciarlini, A. B. Forbes and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology IV*, volume 53 of *Series on Advances in Mathematics for Applied Sciences*, pages 256–272. World Scientific, 2000.
13. D. Zwick. Algorithms for orthogonal fitting of lines and planes: a survey. In P. Ciarlini, M. G. Cox, F. Pavese, D. Richter, editors, *Advanced Mathematical Tools in Metrology II*, pages 272–283. World Scientific, 1996.

Evaluation of measurements by the method of least squares

Lars Nielsen

Danish Institute of Fundamental Metrology (DFM), Lyngby, DK.[1]
LN@dfm.dtu.dk

Abstract

In this paper, a general technique for evaluation of measurements by the method of Least Squares is presented. The input to the method consist of estimates and associated uncertainties of the values of measured quantities together with specified constraints between the measured quantities and any additional quantities for which no information about their values are known a priori. The output of the method consist of estimates of both groups of quantities that satisfy the imposed constraints and the uncertainties of these estimates. Techniques for testing the consistency between the estimates obtained by measurement and the imposed constraints are presented. It is shown that linear regression is just a special case of the method. It is also demonstrated that the procedure for evaluation of measurement uncertainty that is currently agreed within the metrology community can be considered as another special case in which no redundant information is available. The practical applicability of the method is demonstrated by two examples.

1 Introduction

In 1787, the French mathematician and physicist Laplace (1749–1827) used the method of Least Squares to estimate 8 unknown orbital parameters from 75 discrepant observations of the position of Jupiter and Saturn taken over the period 1582–1745. Since then, the method of Least Squares has been used extensively in data analysis. Like Laplace, most people use a special case of the method, known as unweighted linear regression. The calculation of the average and the standard deviation of a repeated set of observations is the most simple example of that. The unweighted regression analysis is based on the assumptions that the observations are independent and have the same (unknown) variance. In addition, the linear regression is based on the assumption that the observations can be modelled by a function that is linear in the unknown quantities to be determined by the regression analysis. For most measurements carried out in practice, none of these assumptions can be justified. In order to evaluate the result of a general measurement, in which some redundant information has been obtained, one therefore has to apply the method of Least Squares in its general form.

This paper describes how measurements can be evaluated by the method of Least Squares in general. The paper is based on an earlier work of the author [2] but includes

[1]Address: Building 307, Matematiktorvet, DK-2800 Lyngby, Denmark

several new features not published before as well as practical examples from the daily work at DFM. An alternative approach is described in [6].

2 Measurement model

In a general measurement, a number $m > 0$ of quantities is either measured directly using measuring instruments or known a priori, for example from tables of physical constants etc. The (exact) values of these m quantities are denoted $\boldsymbol{\zeta}$

$$\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_m)^T .$$

Due to measurement uncertainty, the values $\mathbf{z}$ obtained by the measurement (or from tables etc.)

$$\mathbf{z} = (z_1, ..., z_m)^T$$

are only estimates of the values $\boldsymbol{\zeta}$. The standard uncertainties of the estimates z_i,

$$u(z_i) \quad , \quad i = 1, \ldots, m,$$

are determined in accordance with the GUM [1] and depend on the accuracy of the instruments and the reliability of any tabulated value used. In general, some of the estimates z_i may be correlated. If $r(z_i, z_j)$ is the correlation coefficient between the estimates z_i and z_j then the covariance $u(z_i, z_j)$ between these two estimates is given by

$$u(z_i, z_j) = u(z_i) r(z_i, z_j) u(z_j).$$

Because of the uncertainty, the estimates $\mathbf{z}$ can be considered as an outcome of a m-dimensional random variable $\mathbf{Z}$ with expectation $\boldsymbol{\zeta}$ (the exact values of the quantities) and covariance matrix $\boldsymbol{\Sigma}$

$$\boldsymbol{\Sigma} = u(\mathbf{z}, \mathbf{z}^T) = \begin{pmatrix} u^2(z_1) & u(z_1, z_2) & \cdots & u(z_1, z_m) \\ u(z_2, z_1) & u^2(z_2) & \cdots & u(z_2, z_m) \\ \vdots & \vdots & \ddots & \vdots \\ u(z_m, z_1) & u(z_m, z_2) & \cdots & u^2(z_m) \end{pmatrix} .$$

In addition to the m quantities for which prior information is available either from direct measurement or from other sources, a general measurement may involve a number $k \geq 0$ of quantities for which no prior information is available. The values of these quantities are denoted by

$$\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)^T .$$

In general, the values $\boldsymbol{\beta}$ and $\boldsymbol{\zeta}$ are constrained by a number n of physical or empirical laws. These constraints may be written in terms of an n-dimensional function

$$\mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta}) = \begin{pmatrix} f_1(\boldsymbol{\beta}, \boldsymbol{\zeta}) \\ f_2(\boldsymbol{\beta}, \boldsymbol{\zeta}) \\ \vdots \\ f_n(\boldsymbol{\beta}, \boldsymbol{\zeta}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad , \quad k \leq n < m + k. \tag{2.1}$$

It is assumed that $f_i : \Omega \to R$, $i = 1 \ldots n$, are differentiable functions (with con-

tinuous derivatives) defined in a region $\Omega \subset R^{k+m}$ around $(\boldsymbol{\beta}, \boldsymbol{\zeta})$. As indicated in (2.1), the number n has to be larger than or equal to the number k of quantities for which no prior information is available; otherwise some of the values $\boldsymbol{\beta}$ cannot be determined. In addition, the number n of constraints has to be smaller than the total number $k+m$ of quantities involved; otherwise the values of $\boldsymbol{\beta}$ and $\boldsymbol{\zeta}$ would be uniquely determined by the constraints and no measurements would be needed.

The estimates $\mathbf{z}$, the covariance matrix $\boldsymbol{\Sigma}$ and the n-dimensional function $\mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta})$ are the input to the general Least Squares method. It should be stressed that no probability distribution has to be assigned to the input estimates $\mathbf{z}$. On the contrary, if a probability distribution has been assigned to an estimate, it should be used to calculate the mean value and the variance of the estimate which should then serve as input to the Least Squares method.

Like any other covariance matrix, the covariance matrix $u(\mathbf{z}, \mathbf{z}^T) = \boldsymbol{\Sigma}$ is positive semi-definite. Otherwise, at least one linear combination $\mathbf{x}^T\mathbf{z}$ of the estimates $\mathbf{z}$ would have *negative* variance $u(\mathbf{x}^T\mathbf{z}, \mathbf{z}^T\mathbf{x}) = \mathbf{x}^T\boldsymbol{\Sigma}\mathbf{x}$. In the following it is assumed that $\boldsymbol{\Sigma}$ is positive definite and therefore non-singular.

3 Normal equations

Least Squares estimates $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\zeta}}$ of the values $\boldsymbol{\beta}$ and $\boldsymbol{\zeta}$ are found by minimizing the chi-square function

$$\chi^2(\boldsymbol{\zeta}; \mathbf{z}) = (\mathbf{z} - \boldsymbol{\zeta})^T \boldsymbol{\Sigma}^{-1} (\mathbf{z} - \boldsymbol{\zeta})$$

subject to the constraints

$$\mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta}) = 0.$$

It is convenient to solve this minimization problem by using Lagrange multipliers [5]: If a solution $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})$ to the minimization problem exists, the solution satisfies the equation

$$\nabla_{(\beta,\zeta,\lambda)} \Phi(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}}, \boldsymbol{\lambda}; \mathbf{z}) = \mathbf{0}$$

where

$$\Phi(\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{\lambda}; \mathbf{z}) = (\mathbf{z} - \boldsymbol{\zeta})^T \boldsymbol{\Sigma}^{-1} (\mathbf{z} - \boldsymbol{\zeta}) + 2\boldsymbol{\lambda}^T \mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta})$$

for a particular set of Lagrange multipliers $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)^T$. By taking the gradient of the function Φ, the following $n + m + k$ equations in $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}}, \boldsymbol{\lambda})$ evolve:

$$\begin{aligned} \nabla_\beta \mathbf{f}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})^T \boldsymbol{\lambda} &= \mathbf{0}, \\ -\boldsymbol{\Sigma}^{-1}(\mathbf{z} - \hat{\boldsymbol{\zeta}}) + \nabla_\zeta \mathbf{f}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})^T \boldsymbol{\lambda} &= \mathbf{0}, \\ \mathbf{f}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}}) &= \mathbf{0}, \end{aligned} \tag{3.1}$$

where

$$\nabla_\beta \mathbf{f} = \begin{pmatrix} \frac{\partial f_1}{\partial \beta_1} & \cdots & \frac{\partial f_1}{\partial \beta_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \beta_1} & \cdots & \frac{\partial f_n}{\partial \beta_k} \end{pmatrix} \quad \text{and} \quad \nabla_\zeta \mathbf{f} = \begin{pmatrix} \frac{\partial f_1}{\partial \zeta_1} & \cdots & \frac{\partial f_1}{\partial \zeta_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \zeta_1} & \cdots & \frac{\partial f_n}{\partial \zeta_m} \end{pmatrix}.$$

The equations (3.1) are called the *normal equations* of the Least Squares problem.

4 Solving the normal equations

If $(\boldsymbol{\beta}_l, \boldsymbol{\zeta}_l, \boldsymbol{\lambda}_l)$ is an approximate solution to the normal equations, a refined solution $(\boldsymbol{\beta}_{l+1}, \boldsymbol{\zeta}_{l+1}, \boldsymbol{\lambda}_{l+1})$ can be found by the iteration

$$\begin{pmatrix} \boldsymbol{\beta}_{l+1} \\ \boldsymbol{\zeta}_{l+1} \\ \boldsymbol{\lambda}_{l+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\beta}_l \\ \boldsymbol{\zeta}_l \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \Delta\boldsymbol{\beta}_l \\ \Delta\boldsymbol{\zeta}_l \\ \boldsymbol{\lambda}_{l+1} \end{pmatrix} \quad , \quad l = 1, \ldots, \infty.$$

The step $(\Delta\boldsymbol{\beta}_l, \Delta\boldsymbol{\zeta}_l, \boldsymbol{\lambda}_{l+1})$ is given by

$$\mathbf{D}(\boldsymbol{\beta}_l, \boldsymbol{\zeta}_l) \begin{pmatrix} \Delta\boldsymbol{\beta}_l \\ \Delta\boldsymbol{\zeta}_l \\ \boldsymbol{\lambda}_{l+1} \end{pmatrix} = \begin{pmatrix} \mathbf{0}^{(k,1)} \\ \boldsymbol{\Sigma}^{-1}(\mathbf{z} - \boldsymbol{\zeta}_l) \\ -\mathbf{f}(\boldsymbol{\beta}_l, \boldsymbol{\zeta}_l) \end{pmatrix}, \tag{4.1}$$

where

$$\mathbf{D}(\boldsymbol{\beta}_l, \boldsymbol{\zeta}_l) = \begin{pmatrix} \mathbf{0}^{(k,k)} & \mathbf{0}^{(k,m)} & \nabla_{\boldsymbol{\beta}}\mathbf{f}(\boldsymbol{\beta}_l, \boldsymbol{\zeta}_l)^T \\ \mathbf{0}^{(m,k)} & \boldsymbol{\Sigma}^{-1} & \nabla_{\boldsymbol{\zeta}}\mathbf{f}(\boldsymbol{\beta}_l, \boldsymbol{\zeta}_l)^T \\ \nabla_{\boldsymbol{\beta}}\mathbf{f}(\boldsymbol{\beta}_l, \boldsymbol{\zeta}_l) & \nabla_{\boldsymbol{\zeta}}\mathbf{f}(\boldsymbol{\beta}_l, \boldsymbol{\zeta}_l) & \mathbf{0}^{(n,n)} \end{pmatrix} \tag{4.2}$$

is a symmetric matrix. This iteration procedure is similar to Newton iteration except that the second order partial derivatives of the functions f_i have been neglected as it is practice to do in non-linear Least Squares estimation [4].

In order to reduce the effects of numerical rounding errors, it is recommended to calculate the step $(\Delta\boldsymbol{\beta}_l, \Delta\boldsymbol{\zeta}_l, \boldsymbol{\lambda}_{l+1})$ by solving the linear equations (4.1) by Gauss-Jordan elimination with full pivoting [4]. This algorithm also provides the inverse matrix $\mathbf{D}(\boldsymbol{\beta}_l, \boldsymbol{\zeta}_l)^{-1}$ which is needed at the final stage for estimating the covariance matrix of the solution as shown in Section 5.

If proper starting values $(\boldsymbol{\beta}_1, \boldsymbol{\zeta}_1)$ are selected, the iteration is expected to converge towards the solution $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})$

$$\begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{\zeta}} \\ \boldsymbol{\lambda} \end{pmatrix} = \lim_{l \to \infty} \begin{pmatrix} \boldsymbol{\beta}_l \\ \boldsymbol{\zeta}_l \\ \boldsymbol{\lambda}_l \end{pmatrix}.$$

Since the solutions $\hat{\boldsymbol{\zeta}}$ are expected to be close to the estimates $\mathbf{z}$ of $\boldsymbol{\zeta}$ available a priori, the estimates $\mathbf{z}$ are obviously the proper starting values $\boldsymbol{\zeta}_1$ to be selected for the iteration. The selection of proper starting values $\boldsymbol{\beta}_1$ is more difficult in general. If, however, $\mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta})$ are linear functions in the variables $\boldsymbol{\beta}$, the iteration process will converge after a few iterations, independent of the choice of $\boldsymbol{\beta}_1$.

Most differentiable functions $\mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta})$ can be handled by the described method. In order to get reliable standard uncertainties, it is required, however, that the function can be approximated by a first order Taylor expansion, i.e.

$$\mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta}) \cong \mathbf{f}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}}) + \nabla_{\boldsymbol{\beta}}\mathbf{f}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) + \nabla_{\boldsymbol{\zeta}}\mathbf{f}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})(\boldsymbol{\zeta} - \hat{\boldsymbol{\zeta}})$$

when the values $\boldsymbol{\beta}$ and $\boldsymbol{\zeta}$ are varied around the solution $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\zeta}}$ on a scale comparable to the standard uncertainties of the solution. If this vaguely formulated criterion is met, the function $\mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta})$ is said to be *linearizable*. Note that almost any differentiable function

will be linearizable if the standard uncertainties are sufficiently small. On the other hand, if the uncertainties are sufficiently high, all non-linear functions will no longer be linearizable. The requirement that $\mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta})$ is linearizable is considered to be the only major limitation of the method of Least Squares!

It should be mentioned that the minimization using Lagrange multipliers will fail in case the gradients $\nabla_\beta f_i$ and $\nabla_\zeta f_i$ of one of the constraint functions f_i are both equal to zero at the point of the solution $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})$. This gives some restrictions on how a constraint may be formulated. A function f_i defining a constraint may, for example, be replaced by the square of that function, f_i^2. But since $f_i(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}}) = 0$, the gradient of f_i^2 will be zero at the point of the solution $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})$ although the gradient of f_i is not.

5 Properties of the solution

Since the solution $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}}, \boldsymbol{\lambda})$ depends on the estimates $\mathbf{z}$, which are considered as the realization of the multivariate random variable $\mathbf{Z}$, the solution $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}}, \boldsymbol{\lambda})$ can also be regarded as a multivariate random variable. If the functions $f_i(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})$ are linearizable, the estimates $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})$ are linear in $\mathbf{Z}$

$$\begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{\zeta}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\zeta} \\ \mathbf{0} \end{pmatrix} + \mathbf{D}(\boldsymbol{\beta}, \boldsymbol{\zeta})^{-1} \begin{pmatrix} \mathbf{0}^{(k,1)} \\ \boldsymbol{\Sigma}^{-1}(\mathbf{Z} - \boldsymbol{\zeta}) \\ \mathbf{0}^{(n,1)} \end{pmatrix}. \tag{5.1}$$

In that case, the expectation of the solution is

$$E \begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{\zeta}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\zeta} \\ \mathbf{0} \end{pmatrix}$$

which means that $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})$ are central estimators of the values $(\boldsymbol{\beta}, \boldsymbol{\zeta})$. Under the same assumption, the covariances of the solution are given by the symmetric matrix $\mathbf{D}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})^{-1}$ provided by the Gauss-Jordan elimination algorithm[2]

$$\begin{pmatrix} u(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\beta}}^T) & u(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}}^T) & ()^{(k,n)} \\ u(\hat{\boldsymbol{\zeta}}, \hat{\boldsymbol{\beta}}^T) & u(\hat{\boldsymbol{\zeta}}, \hat{\boldsymbol{\zeta}}^T) & ()^{(m,n)} \\ ()^{(n,k)} & ()^{(n,m)} & -u(\boldsymbol{\lambda}, \boldsymbol{\lambda}^T) \end{pmatrix} = \mathbf{D}(\boldsymbol{\beta}, \boldsymbol{\zeta})^{-1} \cong \mathbf{D}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})^{-1}. \tag{5.2}$$

This relation can be derived as follows. Partition the symmetric matrix $\mathbf{D}^{-1}$ into nine sub-matrices similar to the left hand side of (5.2) or similar to the partitioning of $\mathbf{D}$ according to the definition (4.2). Express the covariance matrix of the solution (5.1) in terms of the covariance matrix $\boldsymbol{\Sigma}$ of the random variable $\mathbf{Z}$ and the matrix $\mathbf{D}^{-1}$. Insert the partitioned $\mathbf{D}^{-1}$ into the resulting matrix double product and express the covariances of the solution in terms of $\boldsymbol{\Sigma}^{-1}$ and the sub-matrices of $\mathbf{D}^{-1}$. Reduce these expressions to the final result by multiple use of the nine relations between the sub-matrices of $\mathbf{D}^{-1}$ and $\mathbf{D}$ derived from the identity $\mathbf{D}\mathbf{D}^{-1} = \mathbf{I}$.

[2]The empty brackets in the left hand matrix indicates the parts of $\mathbf{D}^{-1}$ that do not contain information about covariances.

From equation (5.1) and (5.2) the covariance matrices between $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})$ and the estimates $\mathbf{z}$ are found to be

$$u(\mathbf{z}, \hat{\boldsymbol{\beta}}^T) = u(\hat{\boldsymbol{\zeta}}, \hat{\boldsymbol{\beta}}^T)$$
$$u(\mathbf{z}, \hat{\boldsymbol{\zeta}}^T) = u(\hat{\boldsymbol{\zeta}}, \hat{\boldsymbol{\zeta}}^T).$$

From the last of these two relations, a relation of particular interest is derived,

$$u(\mathbf{z} - \hat{\boldsymbol{\zeta}}, \mathbf{z}^T - \hat{\boldsymbol{\zeta}}^T) = u(\mathbf{z}, \mathbf{z}^T) - u(\hat{\boldsymbol{\zeta}}, \hat{\boldsymbol{\zeta}}^T).$$

For the diagonal elements, this relation reads

$$u^2(z_i - \hat{\zeta}_i) = u^2(z_i) - u^2(\hat{\zeta}_i) \quad , \quad i = 1, \ldots, m.$$

That is, the variance of the difference between the initial estimate z_i of ζ_i and the refined estimate $\hat{\zeta}_i$ is equal to the variance of z_i minus the variance of $\hat{\zeta}_i$. This relation is useful when testing if the difference $z_i - \zeta_i$ is significantly different from its zero expectation.

6 χ^2 test for consistency

When the estimates $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})$ have been found, the minimum χ^2 value

$$\chi^2(\hat{\boldsymbol{\zeta}}; \mathbf{z}) = (\mathbf{z} - \hat{\boldsymbol{\zeta}})^T \boldsymbol{\Sigma}^{-1} (\mathbf{z} - \hat{\boldsymbol{\zeta}})$$

can be used to test if the measured values $\mathbf{z}$ are consistent with the measurement model (2.1) within the uncertainties defined by the covariance matrix $\boldsymbol{\Sigma}$. If the model is linearizable, the expectation of the random variable $\chi^2(\hat{\boldsymbol{\zeta}}; \mathbf{Z})$ is equal to the number m of measured quantities, minus the number $m + k$ of adjusted quantities, plus the number n of constraints, that is

$$E\left[\chi^2(\hat{\boldsymbol{\zeta}}; \mathbf{Z})\right] = m - (m + k) + n = n - k = \nu.$$

If, in addition, the random variables $\mathbf{Z}$ are assumed to follow a multivariate normal distribution with mean values $\boldsymbol{\zeta}$ and covariance matrix $\boldsymbol{\Sigma}$, the random variable $\chi^2(\hat{\boldsymbol{\zeta}}; \mathbf{Z})$ will follow a $\chi^2(\nu)$ distribution with $\nu = n - k$ degrees of freedom. In that case, the probability p of finding a χ^2 value larger than the value $\chi^2(\hat{\boldsymbol{\zeta}}; \mathbf{z})$ actually observed can be calculated from the $\chi^2(\nu)$ distribution

$$p = P\left\{\chi^2(\nu) > \chi^2(\hat{\boldsymbol{\zeta}}, \mathbf{z})\right\} = 1 - P\{\chi^2(\nu) \leq \chi^2(\hat{\boldsymbol{\zeta}}, \mathbf{z})\}.$$

If this probability p is smaller than a certain value α, the hypothesis that the measured values are consistent with the measurement model has to be rejected at a level of significance equal to α. As the result of measurements are normally quoted at a 95% level of confidence, an $\alpha = 5\%$ level of significance is a reasonable choice for the consistency test.

Although the assumption of a normal distribution of $\mathbf{Z}$ may not be fulfilled, it is suggested to carry out the test of consistency as described above anyway. This is justified by the fact that a value of $\chi^2(\hat{\boldsymbol{\zeta}}; \mathbf{z})$ significantly higher than the expectation ν indicates inconsistency no matter what the distribution of $\mathbf{Z}$ might be. The calculated probability

p simply describes *how* unlikely the observed χ^2 value is *if* a normal distribution is assigned to Z.

7 Normalized deviations

If the test described in the previous section leads to a rejection of the measurements, a tool for identifying the outlying measurements is desirable. A measured value z_i is defined as an outlier if the difference $z_i - \hat{\zeta}_i$ is significantly different from zero taking into account the standard uncertainty $u(z_i - \hat{\zeta}_i)$ of that difference. This leads to the introduction of the normalized deviation d_i defined by[3]

$$d_i = \frac{z_i - \hat{\zeta}_i}{u(z_i - \hat{\zeta}_i)} = \frac{z_i - \hat{\zeta}_i}{\sqrt{u^2(z_i) - u^2(\hat{\zeta}_i)}} \quad , \quad i = 1, \ldots, m.$$

The normalized deviation d_i has zero expectation and variance 1. A normalized deviation with $|d_i|$ larger than 2 or 3 is therefore rather unlikely no matter what the distribution of the random variable d_i might be.

If a multivariate normal distribution is assigned to $\mathbf{Z}$ and the model function $\mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta})$ is linearizable, the normalized deviation d_i is normally distributed,

$$d_i \in N(0,1) \quad , \quad i = 1, \ldots, m.$$

In that case

$$P\{|d_i| > 2\} = 5\%,$$

and a measurement with $|d_i| > 2$ is therefore identified as an outlier at a 5% level of significance. It is suggested to use the criteria $|d_i| > 2$ to identify potential outliers even if the distribution assigned to $\mathbf{Z}$ is not normal.

8 Adjustment of a variance σ^2

If some values z_i have a common but unknown variance $u^2(z_i) = \sigma^2$, this variance can be estimated by adjusting σ^2 by an iterative procedure until the "observed" χ^2 value becomes equal to its expectation value ν

$$\chi^2(\hat{\boldsymbol{\zeta}}; \mathbf{z}) = (\mathbf{z} - \hat{\boldsymbol{\zeta}})^T \boldsymbol{\Sigma}^{-1} (\mathbf{z} - \hat{\boldsymbol{\zeta}}) = \nu,$$

where the covariance matrix $\boldsymbol{\Sigma}$ is a function of the unknown variance σ^2. As the estimates $\hat{\boldsymbol{\zeta}}$ depends on the value assigned to σ^2, these estimates have to be updated together with the estimates $\hat{\boldsymbol{\zeta}}$ each time the value of σ^2 is changed during the iteration.

This way of estimating the unknown variance σ^2 leads to the well-known expression for the standard deviation in the case of a repeated measurement of a single quantity as shown in Section 13.

[3] If $u(z_i - \hat{\zeta}_i) = 0$, the difference $z_i - \hat{\zeta}_i$ will be zero as well and d_i may be set equal to zero. This situation occurs whenever there is no redundant information available regarding the value of the quantity ζ_i.

9 Example: Calibration of an analytical balance

An analytical balance with capacity Max=220 g, resolution d=0.1 mg, and built-in adjustment weight was calibrated by DFM in October 1999 during an inter-laboratory measurement comparison piloted by DFM. Two mass standards were used as reference standards. One of them was a traditional 200 g weight (named R200g) of known conventional mass value[4] m_R and density ρ_R. The other reference standard was a specially designed 200 g stack of weights consisting of four discs (named 100g, 50g, 25g and 25g*) machined from the same metal bar of known density ρ. The conventional mass values m_1, m_2, m_3, m_4 respectively of these four discs were not known a priori; only the conventional mass value $m_S = m_1 + m_2 + m_3 + m_4$ of the stack was known.

The calibration was performed by placing a weight combination at the weighing pan of the balance and by recording the corresponding average indication I in the display. A total of 18 weight combinations were used. Each weight combination was weighed 3 times from which the average indication was calculated. The calibration was repeated 4 times during a period of 10 days in which the inter-laboratory comparison took place. From these four calibrations, a grand average indication $I_i, i = 1, \ldots, 18$ was calculated for each of the 18 weight combinations specified in Table 1. The standard uncertainty of the grand average was estimated from the observed variation in indication over the four calibrations.

I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9
100g	100g	100g	100g	100g	100g	100g	100g	100g
50g	50g	50g	50g	50g	25g	25g*	25g	
25g	25g	25g	25g*		25g*			
25g*	25g*							
I_{10}	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	I_{16}	I_{17}	I_{18}
50g	50g	50g	50g	25g	25g	25g*	R200g	R200g
25g	25g	25g*		25g*				
25g*								

TAB. 1. The weight combinations corresponding to the 18 balance indications I_i.

Due to the effect of air buoyancy, the balance indication depends not only on the mass of the weighed body, but also on the density of the body as well as the density of the air. When calibrated in air of known density a, the reference indication I_R of the balance corresponding to a load generated by a weight with conventional mass value m and density ρ is given by

$$I_R = m\left(1 - (a - a_0)\left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)\right)$$

where a_0=1.2 kg/m^3 and ρ_0=8000 kg/m^3 are the reference densities of the air and the weight respectively to which the conventional value of mass refers. As a model for the

[4]The conventional mass value of a body is defined as the mass of a hypothetical weight of density 8000 kg/m^3 that balances the body when weighed in air of density 1.2 kg/m^3 and temperature 20 ^{0}C.

	m_S [g]	m_R [g]	ρ_R [kg/m^3]	ρ [kg/m^3]	a [kg/m^3]	I_1 [div]
z	199.988816	199.999043	7833.01	7965.76	1.1950	199.988617
$u(z)$	0.000010	0.000008	0.29	0.71	0.0035	0.000023
$\hat{\zeta}$	199.988814	199.999043	7833.01	7965.76	1.1946	199.988620
$u(\hat{\zeta})$	0.000010	0.000008	0.29	0.71	0.0035	0.000011
d	1.66	-1.66	1.66	-1.66	1.66	-0.16

	I_2 [div]	I_3 [div]	I_4 [div]	I_5 [div]	I_6 [div]	I_7 [div]
z	199.988608	174.992133	175.009992	150.013558	149.980675	125.002083
$u(z)$	0.000023	0.000023	0.000023	0.000023	0.000023	0.000023
$\hat{\zeta}$	199.988620	174.992149	175.010024	150.013558	149.980672	125.002087
$u(\hat{\zeta})$	0.000011	0.000012	0.000012	0.000013	0.000012	0.000014
d	-0.56	-0.77	-1.61	0.03	0.14	-0.20

	I_8 [div]	I_9 [div]	I_{10} [div]	I_{11} [div]	I_{12} [div]	I_{13} [div]
z	124.984217	100.005650	99.982925	74.986433	75.004325	50.007892
$u(z)$	0.000023	0.000023	0.000023	0.000023	0.000023	0.000023
$\hat{\zeta}$	124.984212	100.005632	99.982899	74.986450	75.004325	50.007881
$u(\hat{\zeta})$	0.000014	0.000013	0.000013	0.000014	0.000014	0.000012
d	0.25	0.93	1.38	-0.87	0.03	0.54

	I_{14} [div]	I_{15} [div]	I_{16} [div]	I_{17} [div]	I_{18} [div]
z	49.974992	24.978533	24.996417	199.998867	199.998875
$u(z)$	0.000023	0.000023	0.000023	0.000023	0.000023
$\hat{\zeta}$	49.974995	24.978557	24.996432	199.998851	199.998851
$u(\hat{\zeta})$	0.000013	0.000011	0.000011	0.000011	0.000011
d	-0.19	-1.17	-0.77	0.78	1.19

	f [g/div]	A [1/div]	m_1 [g]	m_2 [g]	m_3 [g]	m_4 [g]
$\hat{\beta}$	1.00000186	-4.4E-09	100.005774	50.007963	24.978601	24.996476
$u(\hat{\beta})$	0.00000019	1.0E-09	0.000011	0.000010	0.000010	0.000010

TAB. 2. Measured and estimated values and associated standard uncertainties.

calibration curve of the balance, a second order polynomial through zero is assumed

$$I_R = f\left(I + AI^2\right)$$

where f and A are unknown quantities to be determined from the calibration data.

In this example, there are $m = 23$ quantities for which prior information is available from the measurements performed:

$$\boldsymbol{\zeta} = (m_S, m_R, \rho_R, \rho, a, I_1, \ldots, I_{18})^T$$

whereas there are $k = 6$ quantities for which no prior information is available:

$$\boldsymbol{\beta} = (f, A, m_1, m_2, m_3, m_4)^T.$$

	f	A	m_1	m_2	m_3	m_4
f	1	-0.945	0.021	0.071	0.096	0.096
A	-0.945	1	0.124	-0.016	-0.094	-0.094
m_1	0.021	0.124	1	-0.194	-0.269	-0.268
m_2	0.071	-0.016	-0.194	1	-0.287	-0.287
m_3	0.096	-0.094	-0.269	-0.287	1	-0.287
m_4	0.096	-0.094	-0.268	-0.287	-0.287	1

TAB. 3. Correlation coefficients of the estimated $\hat{\beta}$ values.

Between these quantities, there are $n = 19$ constraints:

$$\mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta}) = \begin{pmatrix} (m_1 + m_2 + m_3 + m_4)\left(1 - (a - a_0)\left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)\right) - f\left(I_1 + AI_1^2\right) \\ \vdots \\ m_R\left(1 - (a - a_0)\left(\frac{1}{\rho_R} - \frac{1}{\rho_0}\right)\right) - f\left(I_{18} + AI_{18}^2\right) \\ m_S - (m_1 + m_3 + m_3 + m_4) \end{pmatrix} = \mathbf{0} \ .$$

The measured values $\mathbf{z}$ and associated standard uncertainties are given in Table 2 under the row headings z and $u(z)$. All measured values are assumed to be uncorrelated.

By solving the normal equations, the estimates $\hat{\boldsymbol{\zeta}}$ and $\hat{\boldsymbol{\beta}}$ and associated standard uncertainties given in Table 2 under the row headings $\hat{\zeta}$, $u(\hat{\zeta})$, $\hat{\beta}$ and $u(\hat{\beta})$ are obtained. Selected correlation coefficients derived from $\mathbf{D}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})^{-1}$ are given in Table 3. The observed minimum χ^2 value is $\chi(\hat{\boldsymbol{\zeta}}, \mathbf{z}) = 8.6$ which should be compared to the expectation value $\nu = n - k = 19 - 6 = 13$. Since $P\left\{\chi^2(13) > 8.6\right\} = 80.3\%$, it is concluded that the measured values are consistent with the specified constraints taking into account the measurement uncertainties. This conclusion is confirmed by the calculated normalized deviations given in Table 2 under the row heading d; all normalized deviations satisfy the criterion $|d| < 2$.

From the estimates of the quantities f and A and the associated covariance matrix, the error of indication E, defined as

$$E = I - I_R = I - f\left(I + AI^2\right),$$

and the associated standard uncertainty $u(E)$ can be calculated as a function of the indication I. The result is shown in Figure 1 as the full lines representing $E - u(E)$, E, and $E + u(E)$. The measured points $E_i, i = 1, \ldots, 18$ shown in the figure are the observed average balance indications I_i minus the corresponding reference values I_R. The error bars of the measured points indicate the standard uncertainties $u(E_i)$ that have been calculated taking into account the covariance between I_i and I_R.

10 Example: Evaluation of calibration history

A weight (named R1mg) of nominal mass 1 mg has been calibrated 39 times in the period 1992-2001. For calibration number i, the mass m_i of the weight at the time t_i and the associated standard uncertainties $u(m_i)$ and $u(t_i)$ are given. The calibration history of the weight is shown in Figure 2 as dots with error bars indicating the standard

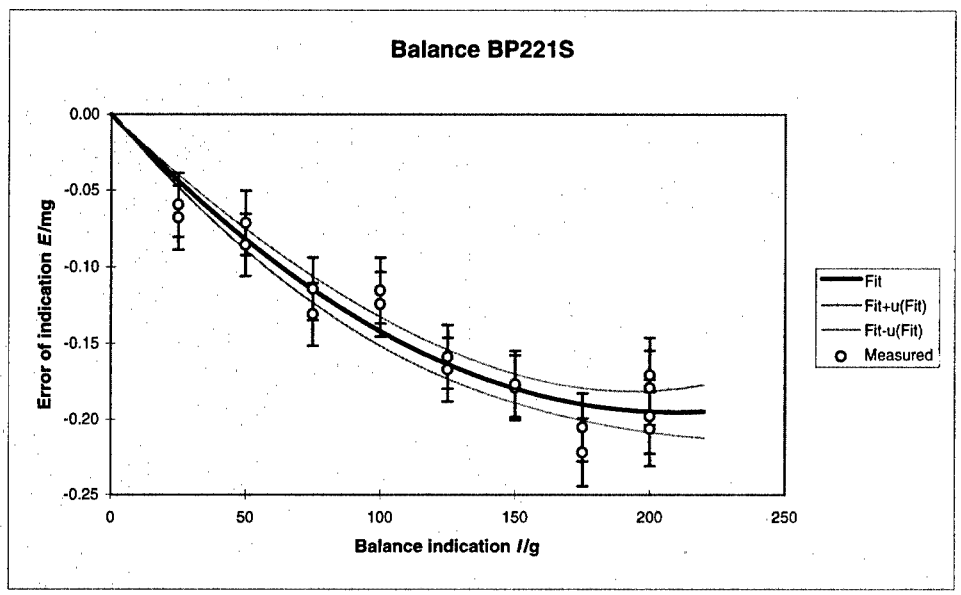


FIG. 1. Error of indication of the calibrated balance.

uncertainties; the scale mark 1992-01 on the time axis indicates the position of the date 1 January 1992 etc.

Due to wear and changes in the amount of dirt adsorbed to the surface, the mass of the weight is expected to change in time. A reasonable model of the change in mass as a function of time is a superposition of a deterministic linear drift and a random variation

$$m_i = a_1 + a_2 t_i + \delta m_i \quad , \quad i = 1, \ldots, 39,$$

where δm_i is a random variable with zero expectation and variance σ^2. The drift parameters a_1, a_2 and the associated covariance matrix as well as the variance σ^2 are unknown a priori and are to be estimated from the calibration history available. Once the estimates $\hat{a}_1$ and $\hat{a}_2$ have been found, it is possible to predict a value $\hat{m}$ of the mass of the weight as a function of time t

$$\hat{m} = \hat{a}_1 + \hat{a}_2 t + \hat{\delta m},$$

where $\hat{\delta m} = 0$ with standard uncertainty $u(\hat{\delta m}) = \sigma$. The standard uncertainty of the predicted mass value is given by

$$u^2(\hat{m}) = u^2(\hat{a}_1) + t^2 u^2(\hat{a}_2) + 2tu(\hat{a}_1, \hat{a}_2) + \sigma^2.$$

The measurement model used for evaluating the calibration history is

$$\boldsymbol{\zeta} = (m_1, \ldots, m_{39}, t_1, \ldots, t_{39}, \delta m_1, \ldots, \delta m_{39})^T \quad , \quad \boldsymbol{\beta} = (a_1, a_2)^T,$$

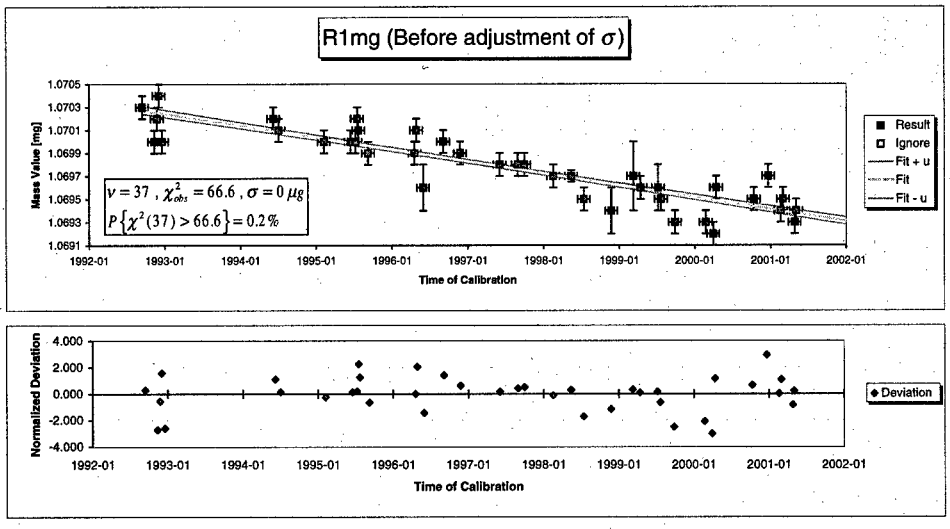


FIG. 2. Evaluation of the calibration history of a 1 mg weight assuming that $\sigma = 0$.

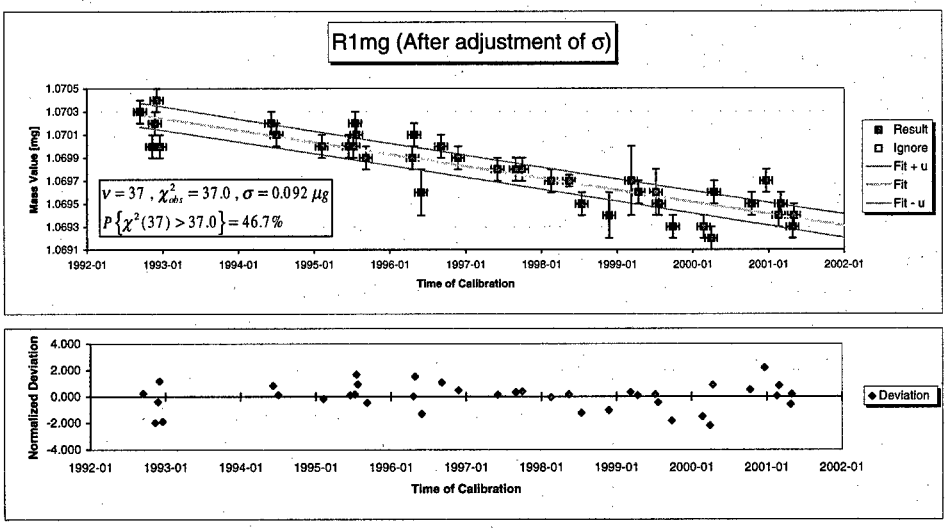


FIG. 3. Evaluation of the calibration history of the 1 mg weight with σ adjusted to 0.092 μg.

$$\mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta}) = \begin{pmatrix} m_1 - (a_1 + a_2 t_1 + \delta m_1) \\ \vdots \\ m_{39} - (a_1 + a_2 t_{39} + \delta m_{39}) \end{pmatrix} = \mathbf{0}.$$

The measured values $\mathbf{z}$ are given by the calibration history, except for the values of

$\delta m_i, i = 1, \ldots, 39$ which are set equal to the expectation value zero. The associated covariance matrix $u(\mathbf{z}, \mathbf{z}^T) = \mathbf{\Sigma}$ is built up from the uncertainties $u(m_i)$ and $u(t_i)$ available from the calibration history and a negligible but finite[5] initial value of the unknown variance σ^2. Since the standard uncertainties $u(m_i)$ are of the order 0.1 μg, the value σ=1E-07 μg is considered negligible and is selected as a starting point.

By solving the normal equations, estimates $\hat{a}_1$ and $\hat{a}_2$ of the drift parameters and the associated covariance matrix are found after a few iterations. The predicted value $\hat{m}$ of the mass of the weight and the associated standard uncertainty $u(\hat{m})$ as a function of time are shown in Figure 2 as solid lines. The normalized deviations d associated with the mass values m_i are shown in Figure 2 as well[6]. The observed minimum chi-square value is $\chi^2 = 66.6$ which is large compared to the expectation value $\nu = 39 - 2 = 37$. Since $P\{\chi^2(37) > 66.6\} = 0.2\%$, the hypothesis $\sigma = 0$, or no random variation in the mass, is rejected at a 0.2% level of significance.

The value of σ is therefore increased as described in Section 8 until the calculated minimum χ^2 value becomes equal to its expectation value $\nu = 37$. In this way the standard uncertainty reflecting the random variation of the mass of the weight is found to be σ=0.092 μg. The result of the evaluation of the calibration history after adjustment of σ is shown in Figure 3. Note the significant increase in the standard uncertainty of the predicted value of the mass of the weight and the decrease in the absolute value of the normalized deviations d.

The calibration history can also be evaluated by an iterative technique based on linear regression [3]. The results obtained are identical to the results presented in this section.

11 Case I: Univariate output quantity, $Y = h(X_1, \ldots, X_N)$

In this section it is shown that the evaluation of measurements by the method of least squares is consistent with the generally accepted principles for evaluating measurement uncertainty as described in the GUM [1].

Using the nomenclature of the GUM, a univariate output quantity Y is assumed to be related to N input quantities $X_1, \ldots, X_N$ through a specified function h,

$$Y = h(X_1, \ldots, X_N).$$

The values assigned to the input and output quantities are denoted $x_1, \ldots, x_N$ and by y respectively. In the nomenclature of this paper, the measurement model is

$$\boldsymbol{\zeta} = (X_1, \ldots, X_N)^T \quad , \quad \boldsymbol{\beta} = (Y),$$

$$\mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta}) = (Y - h(X_1, \ldots, X_N)) = 0.$$

The measured values are

$$\mathbf{z} = (x_1, \ldots, x_N)^T$$

[5] If the variance σ^2 is assumed to be exactly zero, the quantities δm_i have to be removed from the model. Otherwise the covariance matrix $\mathbf{\Sigma}$ will be singular.

[6] The absolute value of normalized deviations of t_i and δm_i is equal to the absolute value of the normalized deviation of m_i.

with the known covariance matrix

$$\mathbf{\Sigma} = u(\mathbf{z}, \mathbf{z}^T) = \begin{pmatrix} u^2(x_1) & \cdots & u(x_1, x_N) \\ \vdots & \ddots & \vdots \\ u(x_N, x_1) & \cdots & u^2(x_N) \end{pmatrix}.$$

The coefficient matrix $\mathbf{D}$ of the normal equations is

$$\mathbf{D}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}}) = \begin{pmatrix} 0 & \mathbf{0}^{(1,N)} & 1 \\ \mathbf{0}^{(N,1)} & \mathbf{\Sigma}^{-1} & -\nabla_{\mathbf{x}} h(\mathbf{x})^T \\ 1 & -\nabla_{\mathbf{x}} h(\mathbf{x}) & 0 \end{pmatrix},$$

where

$$\nabla_{\mathbf{x}} h = \left(\frac{\partial h}{\partial X_1}, \ldots, \frac{\partial h}{\partial X_N} \right).$$

In the present case, the solution to the normal equations is found after one iteration,

$$y = \hat{\boldsymbol{\beta}} = h(x_1, \ldots, x_N) \quad , \quad \hat{\boldsymbol{\zeta}} = (x_1, \ldots, x_N)^T, \quad \lambda = 0.$$

The associated covariances are given by

$$\begin{pmatrix} u^2(y) & u(y, \hat{\boldsymbol{\zeta}}^T) & ()^{(1,1)} \\ u(\hat{\boldsymbol{\zeta}}, y) & u(\hat{\boldsymbol{\zeta}}, \hat{\boldsymbol{\zeta}}^T) & ()^{(N,1)} \\ ()^{(1,1)} & ()^{(1,N)} & -u^2(\lambda) \end{pmatrix} = \mathbf{D}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})^{-1}$$

$$= \begin{pmatrix} \nabla_{\mathbf{x}} h(\mathbf{x}) \, \mathbf{\Sigma} \, \nabla_{\mathbf{x}} h(\mathbf{x})^T & \nabla_{\mathbf{x}} h(\mathbf{x}) \, \mathbf{\Sigma} & 1 \\ \mathbf{\Sigma} \, \nabla_{\mathbf{x}} h(\mathbf{x})^T & \mathbf{\Sigma} & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

In other words,

$$u^2(y) = \nabla_{\mathbf{x}} h(\mathbf{x}) \, \mathbf{\Sigma} \, \nabla_{\mathbf{x}} h(\mathbf{x})^T = \sum_{i=1}^{N} \sum_{j=1}^{N} c_i u(x_i, x_j) c_j, \quad c_i \equiv \frac{\partial h}{\partial X_i}(x_i)$$

which is identical to the linear variance propagation formula given in the GUM.

12 Case II: Linear regression, $\mathbf{Y} = \mathbf{Xa}$

Linear regression is applied when there is a linear relationship $\mathbf{Y} = \mathbf{Xa}$ between some observed quantities $\mathbf{Y}$ and some unknown quantities $\mathbf{a}$. The *design matrix* $\mathbf{X}$ is made up of known elements that may be given as specified functions of one or several independent variables. In the notation of this paper, the measurement model for the linear regression problem is

$$\boldsymbol{\zeta} = \mathbf{Y} = (Y_1, \ldots, Y_n)^T \quad , \quad \boldsymbol{\beta} = \mathbf{a} = (a_1, \ldots, a_k)^T,$$

$$\mathbf{f}(\boldsymbol{\zeta}, \boldsymbol{\beta}) = \mathbf{Y} - \mathbf{Xa} = \mathbf{0},$$

where $\mathbf{X}^{(n,k)}$ is the known design matrix. The measured values are

$$\mathbf{z} = \mathbf{y} = (y_1, \ldots, y_n)^T$$

with known covariance matrix

$$\mathbf{\Sigma} = u(\mathbf{z}, \mathbf{z}^T) = \begin{pmatrix} u^2(y_1) & \cdots & u(y_1, y_n) \\ \vdots & \ddots & \vdots \\ u(y_n, y_1) & \cdots & u^2(y_n) \end{pmatrix}.$$

The coefficient matrix $\mathbf{D}$ of the normal equations is

$$\mathbf{D}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}}) = \begin{pmatrix} \mathbf{0}^{(k,k)} & \mathbf{0}^{(k,n)} & -\mathbf{X}^T \\ \mathbf{0}^{(n,k)} & \mathbf{\Sigma}^{-1} & \mathbf{I}^{(n,n)} \\ -\mathbf{X} & \mathbf{I}^{(n,n)} & \mathbf{0}^{(n,n)} \end{pmatrix}.$$

Again, the solution to the normal equations is found after one iteration,

$$\hat{\mathbf{a}} = \hat{\boldsymbol{\beta}} = \mathbf{C}\mathbf{X}^T\mathbf{\Sigma}^{-1}\mathbf{y} \quad , \quad \hat{\mathbf{Y}} = \hat{\boldsymbol{\zeta}} = \mathbf{X}\hat{\mathbf{a}} \quad , \quad \boldsymbol{\lambda} = -\mathbf{\Sigma}^{-1}(\hat{\mathbf{Y}} - \mathbf{y})$$

where $\mathbf{C} \equiv (\mathbf{X}^T\mathbf{\Sigma}^{-1}\mathbf{X})^{-1}$. The associated covariances are given by

$$\begin{pmatrix} u(\hat{\mathbf{a}}, \hat{\mathbf{a}}^T) & u(\hat{\mathbf{a}}, \hat{\mathbf{Y}}^T) & ()^{(k,n)} \\ u(\hat{\mathbf{Y}}, \hat{\mathbf{a}}^T) & u(\hat{\mathbf{Y}}, \hat{\mathbf{Y}}^T) & ()^{(n,n)} \\ ()^{(n,k)} & ()^{(n,n)} & -u(\boldsymbol{\lambda}, \boldsymbol{\lambda}^T) \end{pmatrix} = \mathbf{D}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})^{-1}$$

$$= \begin{pmatrix} \mathbf{C} & \mathbf{C}\mathbf{X}^T & -\mathbf{C}\mathbf{X}^T\mathbf{\Sigma}^{-1} \\ \mathbf{X}\mathbf{C} & \mathbf{X}\mathbf{C}\mathbf{X}^T & \mathbf{I} - \mathbf{X}\mathbf{C}\mathbf{X}^T\mathbf{\Sigma}^{-1} \\ -\mathbf{\Sigma}^{-1}\mathbf{X}\mathbf{C} & \mathbf{I} - \mathbf{\Sigma}^{-1}\mathbf{X}\mathbf{C}\mathbf{X}^T & \mathbf{\Sigma}^{-1}\mathbf{X}\mathbf{C}\mathbf{X}^T\mathbf{\Sigma}^{-1} - \mathbf{\Sigma}^{-1} \end{pmatrix},$$

that is,

$$\hat{\mathbf{a}} = \mathbf{C}\mathbf{X}^T\mathbf{\Sigma}^{-1}\mathbf{y} \quad , \quad u(\hat{\mathbf{a}}, \hat{\mathbf{a}}^T) = \mathbf{C} = (\mathbf{X}^T\mathbf{\Sigma}^{-1}\mathbf{X})^{-1}$$

as is known from the theory of linear regression.

13 Case III: Repeated observations of a single quantity

Assume that a quantity X is measured n times with the same uncertainty σ. Such a measurement can be modelled by n quantities $X_1, \ldots, X_n$ having a common value μ

$$\boldsymbol{\zeta} = \mathbf{X} = (X_1, \ldots, X_n)^T \quad , \quad \boldsymbol{\beta} = (\mu),$$

$$\mathbf{f}(\boldsymbol{\beta}, \boldsymbol{\zeta}) = \begin{pmatrix} X_1 - \mu \\ \vdots \\ X_n - \mu \end{pmatrix} = \mathbf{0}.$$

The measured values are

$$\mathbf{z} = \mathbf{x} = (x_1, \ldots, x_n)^T,$$

and under the assumption that the measurement results are mutually independent, the associated covariance matrix is given by

$$\mathbf{\Sigma} = u(\mathbf{z}, \mathbf{z}^T) = \begin{pmatrix} \sigma^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma^2 \end{pmatrix}.$$

The coefficient matrix $\mathbf{D}$ of the normal equations is

$$\mathbf{D}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}}) = \begin{pmatrix} \mathbf{0}^{(1,1)} & \mathbf{0}^{(1,n)} & -\mathbf{1}^{(1,n)} \\ \mathbf{0}^{(n,1)} & \sigma^{-2}\mathbf{I}^{(n,n)} & \mathbf{I}^{(n,n)} \\ -\mathbf{1}^{(n,1)} & \mathbf{I}^{(n,n)} & \mathbf{0}^{(n,n)} \end{pmatrix},$$

where $\mathbf{1}$ denotes a matrix with all elements equal to 1. The solution of the normal equations is found after one iteration,

$$\hat{\mu} = \hat{\boldsymbol{\beta}} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad , \quad \hat{\mathbf{X}} = \hat{\boldsymbol{\zeta}} = \hat{\mu}\mathbf{1}^{(n,1)} \quad , \quad \boldsymbol{\lambda} = -\sigma^{-2}(\hat{\mathbf{X}} - \mathbf{x}).$$

The associated covariances are given by

$$\begin{pmatrix} u^2(\hat{\mu}) & u(\hat{\mu}, \hat{\mathbf{X}}^T) & ()^{(1,n)} \\ u(\hat{\mathbf{X}}, \hat{\mu}) & u(\hat{\mathbf{X}}, \hat{\mathbf{X}}^T) & ()^{(n,n)} \\ ()^{(n,1)} & ()^{(n,n)} & -u(\boldsymbol{\lambda}, \boldsymbol{\lambda}^T) \end{pmatrix} = \mathbf{D}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\zeta}})^{-1}$$

$$= \begin{pmatrix} \sigma^2 n^{-1} & \sigma^2 n^{-1}\mathbf{1}^{(1,n)} & n^{-1}\mathbf{1}^{(1,n)} \\ \sigma^2 n^{-1}\mathbf{1}^{(n,1)} & \sigma^2 n^{-1}\mathbf{1}^{(n,n)} & \mathbf{I}^{(n,n)} - n^{-1}\mathbf{1}^{(n,n)} \\ n^{-1}\mathbf{1}^{(n,1)} & \mathbf{I}^{(n,n)} - n^{-1}\mathbf{1}^{(n,n)} & \sigma^{-2}(n^{-1}\mathbf{1}^{(n,n)} - \mathbf{I}^{(n,n)}) \end{pmatrix}.$$

As expected,

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad , \quad u^2(\hat{\mu}) = \sigma^2/n.$$

If σ^2 is not known a priori, it can be estimated by solving the equation

$$\chi^2(\hat{\boldsymbol{\zeta}}; \mathbf{z}) = \sum_{i=1}^{n} \frac{(x_i - \hat{\mu})^2}{\sigma^2} = n - 1,$$

i.e.,

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \hat{\mu})^2$$

which is the well known expression for the experimental standard deviation s.

14 Conclusion

A general technique for evaluation of measurements by the method of Least Squares has been presented. The applicability of the method has been demonstrated by two examples. It has been shown that the method is fully compatible with the generally accepted principles for evaluation of measurement uncertainty laid down in the GUM and that ordinary linear regression is just a special case of the method.

The **input** to the method consists of

- An estimate of the value of each measured quantity, including any relevant influence quantity.
- The covariance matrix of these estimates formed by the standard uncertainties of the estimates and the correlation coefficients between the estimates.

- A measurement model describing all the known relations between the measured quantities and some additional quantities (if needed) for which no prior information is available.

The **output** of the method consists of

- An adjusted estimate of the value of each measured quantity and an estimate of each additional quantity introduced in the measurement model.
- The covariance matrix of all these estimates from which the standard uncertainties and correlation coefficients can be calculated.
- A chi-square value which is a measure of the degree of consistency between the measurement model, the input estimates, and the covariances of the input quantities.

The adjusted estimate of the value of a measured quantity differs from the input estimate only if the measurement model imposes additional information regarding the value of that particular quantity. In that case the standard uncertainty of the adjusted estimate will be smaller than the standard uncertainty of the input estimate. For a good measurement, the difference between the adjusted estimate and the input estimate of a measured quantity should not be large compared to the standard uncertainty of that difference. It has therefore been suggested that the ratio d of the difference to its standard uncertainty is calculated and assessed against a selected criterion, e.g. $|d| < 2$. By plotting the d values of the adjusted estimates it is possible to assess whether a too high chi-square value is caused by a few poor input estimates or is due to a poor model.

Bibliography

1. BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, OIML, *Guide to the expression of uncertainty in measurement*, ISO, 1995.
2. L. Nielsen, Least-squares estimation using Lagrange multipliers, *Metrologia* **35** (1998), 115–118. Erratum, *Metrologia* **37** (2000), 183.
3. L. Nielsen, Evaluation of the calibration history of a measurement standard, DFM report DFM-01-R25, 2001, 1–6.
4. W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipies in C*, 2nd ed., Cambridge, Cambridge University Press, 1992, 36–40 and 681–688.
5. T. L. Saaty and J. Bram, *Nonlinear Mathematics*, Dover Publications, New York, 1981, 93–95.
6. K. Weise and W. Wöger, *Uncertainty and Measurement Data Evaluation*, Wiley-VCH, 1999, 183–224 [in German].

An overview of the relationship between approximation theory and filtration

Paul J. Scott

Taylor Hobson Limited, Leicester, UK.
PScott@taylor-hobson.com

Xiang Q. Jiang

University of Huddersfield, Huddersfield, UK.
x.jiang@hud.ac.uk

Liam A. Blunt

University of Huddersfield, Huddersfield, UK.
l.a.blunt@hud.ac.uk

Abstract

This paper gives an overview of the similarities and differences between the requirements and techniques used in mathematical approximation theory and filtration in surface metrology. Although the two fields tend to use the same or similar mathematical objects to produce functions that simplify a function in a controlled manner, it is the way that this simplification is achieved which is the main difference between the two. Approximation theory uses norms to judge the closeness of the approximation while filtration uses the concept of wavelength to control the "smoothness" of the result of filtration. The new ISO definition of a filter is stated, together with a generalisation of the concept of wavelength through "brickwall" filters. This new ISO definition of a filter illustrates the closeness of approximation theory and filtration. The paper then proceeds to survey some recent developments in filtration in the hope that there can be some cross-fertilisation between approximation theory and filtration. These include wavelets, robust filters and non-linear filters such as the family of morphological filters, which includes envelope filters and alternating sequence filters (non-linear multiresolution). Examples from surface texture are used throughout the paper.

1 Introduction

This paper gives an overview of the similarities and differences between the requirements and techniques used in mathematical approximation theory and filtration in surface metrology. It is not the intention of this paper to give full mathematical detail but to survey recent developments in filtration in the hope that there can be some cross-fertilisation between approximation theory and filtration.

Although the two fields tend to use the same or similar mathematical objects to produce functions that simplify the original function in a controlled manner, it is the

way that this simplification is achieved which is the main difference between the two.

Mathematical approximation theory is concerned with best and good approximation of a large family of functions from a smaller set (usually finitely generated, linear or non-linear) in certain normed spaces (such as L_p), the construction of good approximants (if possible) and the determination of approximation order. Classical tools to achieve this include polynomial tools and splines. More recent tools include wavelets and multiresolution that decompose the normed spaces.

Filtration uses the concept of "wavelength" to control the "smoothness" of the result of filtration. In surface metrology, filtration is concerned with the extraction of features within a prescribed "wavelength" band defined by "wavelength cut-offs". Classical tools to achieve this include Gaussian filters [1], polynomials and splines [4]. Recently there has been a resurgence of activity, both fundamentally and practically, in filtration for surface metrology.

The International Standards Organisation Technical Committee 213 (ISO TC/213), whose remit includes surface metrology, has recently set up an Advisory Group (AG9) to explore filtration for surface metrology. They are producing a series of technical specifications (ISO/TS 16610 series [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]) to standardise filter terminology and to introduce to industry other filtration tools, which include spline wavelets [5], morphological filters [9] and scale-space techniques [10].

Other groups are also producing filtration for surface metrology. The University of Huddersfield has used second generation wavelets to produce an improved spline wavelet [12]. The University of Hanover is exploring robust Gaussian filtration [6]. PTB has developed a Robust Spline filter [7]. The rest of the paper surveys some of the results of this recent activity.

2 Basic concepts of filtration

This section is a summary of the basic concepts of filtration as given in ISO/TS 16610 part 1 [2].

Let $\mathcal{P}$ be the space of real surfaces.

Let $\mathcal{V}_\lambda$ be a set of nested subspaces indexed by $\lambda \in \mathcal{R}^+$ (here $\mathcal{R}^+$ is the set of positive reals which includes zero) such that

$$\forall \lambda > \mu \geq 0; \mathcal{V}_\lambda \subseteq \mathcal{V}_\mu \subseteq \mathcal{P} \text{ and } \mathcal{V}_0 \text{ is dense on } \mathcal{P}.$$

The nesting index λ a number indicating the relative level of nesting for a particular subspace in such a way that given a particular nesting index, subspaces with lower indices contain more surface information and subspaces with higher nesting indices contain less surface information. By convention, as the nesting index approaches zero there exists a surface in that indexed subspace that approximates the real surface to within any given measure of closeness as defined by a suitable norm. Thus approximation theory is used to define Filtration. The usual norm used in filtration is L_2 but others are used such as the one-sided Chebychev for morphological filters.

Let $\Phi_\lambda : \mathcal{P} \rightarrow \mathcal{V}_\lambda$ be a projection from the space of real surfaces onto the subspace indexed by $\lambda \geq 0$ which satisfies the following two properties.

- The sieve criterion: $\forall \lambda, \mu \geq 0$ and $\forall a \in \mathcal{P}; \Phi_\lambda(\Phi_\mu(a)) = \Phi_{\sup(\lambda,\mu)}(a)$.
- The projection criterion: $\forall \lambda \geq 0$ and $\forall a \in \mathcal{V}_\lambda; \Phi_\lambda(a) = a$.

Φ_λ is called the brickwall filter (or primary mapping) and is a method of choosing a particular surface belonging to a subspace with a specified nesting index, to represent the real surface, which satisfies the projection and sieve criteria [16].

The sieve criterion allows brickwall filters to have the property that once the surface has been brickwall filtered at a particular nesting index, subsequent brickwall filtering with a higher nesting index will produce the same surface as brickwall filtering the original surface with the brickwall filter with the higher nesting index.

The projection criterion is required in order that the nesting index is a scale or size. For define the set operator $\Psi_\lambda : \mathcal{P} \rightarrow \mathcal{P}$ as

$$\forall \lambda \geq 0 \text{ and } \forall P \subseteq \mathcal{P}; \Psi_\lambda(P) := \{p : p \in P \text{ and } \Phi_\lambda(p) = p\}.$$

That is to say $p \in \Psi_\lambda(P)$ if and only if $p \in P$ and $\Phi_\lambda(p) = p$. Then it is easily demonstrated that the set operator Ψ_λ is a granulometry [16] on $\mathcal{P}$ and λ is the scale/size of the granulometry.

Since the nesting index of brickwall filters is a scale/size and it satisfies the sieve criterion, it can be used to define the generalised concept of wavelength. An example of a brickwall filter is a morphological closing filter using a sphere as the structuring element. Here the nesting index is the radius of the sphere.

Other filters can be constructed using brickwall filters (e.g. weighted mean of brickwall filters, supremum of brickwall filters, etc.).

3 Wavelet filters

An important example of the concepts discussed in the previous section is wavelet filtration. The multiresolution form of the wavelet transform consists of constructing a ladder of smooth approximations to the profile. The first rung is the original profile. Each rung in the ladder consists of a filter bank where the profile A_i is split into two components giving, a smoother version A_{i+1} of the profile which becomes the next rung and a component D_{i+1} that is the "difference" between the two rungs.

The multiresolution ladder structure lends itself naturally to a set of nested mathematical models of the profile, with the ith model m_i, reconstructed from (D_1, D_2, D_3, ..., D_i, A_i). The nesting index is the order of the model, the higher the model the smoother the representation with less detail. Thus m_{i+1} is a smoother version of the profile than m_i .

As part of a research programme at the University of Huddersfield, the use of biorthogonal wavelets for surface analysis has been investigated because of their significant merits [12]. A very fast, second-generation, in-place algorithm, which uses the lifting scheme, has been developed at Bell Laboratories for biorthogonal wavelets [13]. One important property of biorthogonal wavelets is that they allow the construction of symmetric wavelets and thus linear phase filters that preserves the location of surface features with far less distortion than phase shift filters.

Surface texture analysis usually breaks down a surface into defined wavelength components of the surface called roughness, waviness and form. There are many well-known problems with the current standardised filter [14], i.e. Gaussian filter [1], including lost data at the edges, distortion due to form, retention of unwanted wavelengths, etc.. Huddersfield has investigated the possibility of using a 'lifting wavelet' model to overcome some of these problems and enhance the extraction accuracy for roughness, waviness and form. This is achieved by using the wavelet transform to break down the surface into subsets at different scales and recombining only those subsets of the scales of interest (i.e. setting all the other subsets to zero and applying the inverse wavelet transform). Figure 1 shows the application of the wavelet filtering technique a femoral head from an artificial hip joint. Full details of the particular biorthogonal wavelet and its associated lifting scheme together with some engineering applications are given in reference [12].

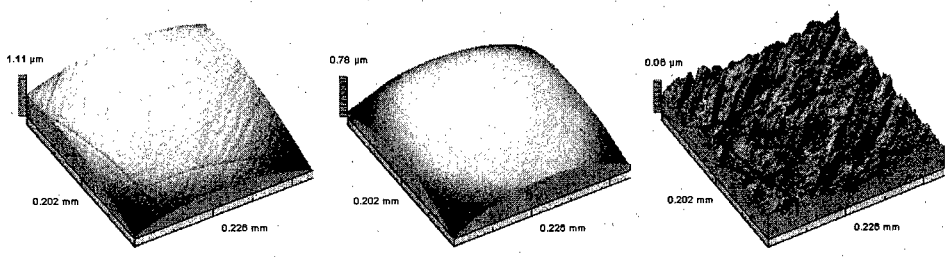


FIG. 1. Metallic femoral head showing original, reference and roughness surfaces.

4 Envelope filters

Traditional linear filters, such as the Gaussian filter [1], produce a smoothed mean surface through a measured surface. Many engineering applications of functional surfaces involve mechanical contact where the envelope of the surface is of interest rather than the mean surface. But what exactly is the envelope of a surface?

The following are defining properties of the envelope of a surface used by ISO TC/213 AG9 [8]:

- the envelope filter must be Extensive, i.e., $\forall A, F(A) \geq A$,
- the envelope filter must be Increasing, i.e., $A \leq B$ implies $F(A) \leq F(B)$,
- the envelope filter must be Idempotent, i.e., $F(F(A)) = F(A)$,

where A, B are surfaces and F(A) is the filtered surface of surface A.

But these are also the defining properties of a morphological closing filter [15]; hence all envelope filters are morphological closing filters. A morphological closing filter using a disk as the structuring element is illustrated in Figure 2.

Unfortunately, envelope filters, by definition, are not very robust to outliers, consisting of large spikes, in the surface. Scale-space is an attempt to overcome this problem with the morphological closing filter.

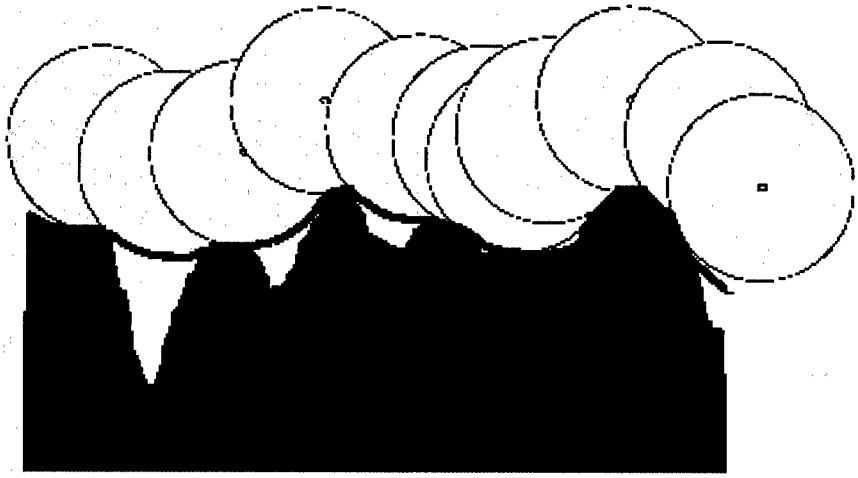


FIG. 2. An envelope filter using a closing filter with a disk as a structural element.

5 Scale-space

Scale-space is a way of breaking down a signal or image into objects of different scales. To define scale-space we need to define the size of objects in a signal or image. This is achieved using Alternating Sequence Filters [10].

Alternating Sequence Filters (ASFs) are defined in terms of matched pairs of closing and opening filters. A closing followed by an opening both at a given scale (radius of the circle, length of the horizontal segment, etc.) will eliminate features of the surface whose "scales" are smaller than the given scale.

ASFs begin by eliminating very small features, then eliminating slightly larger features, and then eliminating slightly larger features still etc., in a systematic way up to a given scale. Usually there is a constant ratio between successive scales. This process produces a ladder structure similar to wavelet analysis. At each rung in the ladder the profile is filtered by a matched pair of closing and opening filters at a given scale to obtain the next rung profile and a component that is the "difference" between the two rungs. The ladder structure leads to a multiresolution analysis, similar to wavelet analysis, with all of the associated analysis techniques. An example of scale space of a profile from a ceramic surface is given in Figure 3. The top part of this figure shows the original non-smoothed profile with the final smoothed profile.

6 Robustness

Robustness of filtration is an increasingly important area of interest in surface metrology. Robustness is not in general an absolute property of a filter but a relative one. One can only say that a particular filter is more robust than an alternative filter against a particular phenomenon if there is less distortion in that filter's response to that phenomenon than in the alternative filter's response.

To make robustness an absolute property of filters we need to define a reference class

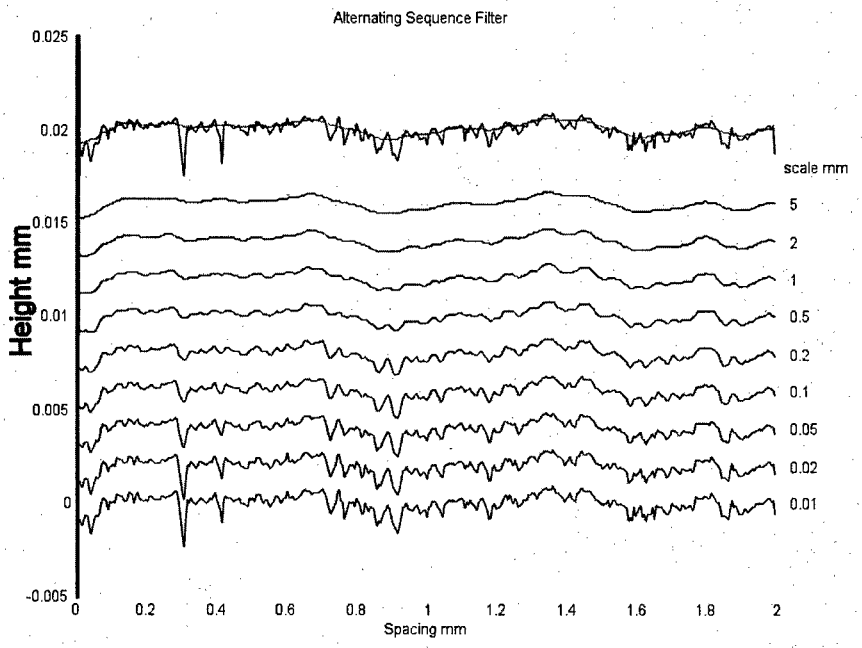


FIG. 3. Successively smooth profiles of a ceramic profile using an ASF with a disk.

of profile filters with which to compare. The reference class of filters defined in ISO TC/213 AG9 is the class of linear filters [3]. Hence by this definition all robust filters must be non-linear. There are several well-known techniques (all non-linear) which can produce robust filters for a particular phenomenon. These are indicated in the next sections.

6.1 Metric based

Here the metric used to fit the filter to the surface is altered to a more "robust" metric.

For example the metric based on the L_1 norm is more robust against spike discontinuities than the metric based on the least square norm (L_2 norm), which in turn is more robust then the metric based on the Chebychev norm (L_∞ norm).

The Robust Spline Filter given in ISO/TS 16610 part 32 uses an L_1 metric rather than the usual L_2 norm to make it more robust [7].

6.2 Robust statistics

Here each point on the surface is weighted according to its relative height position to the filter's smooth response, with points further away being given less influence on the filter response than points nearer in height. This is an attempt to make the filter more robust against spike discontinuities. There are several standard functions used to allocate the weights to points (Huber, Beaton functions, etc.) which can be found in any standard book on robust statistics [17].

The Robust Gaussian regression filter given in ISO/TS 16610 part 31 uses a Beaton function to alter the influence of outliers [6].

6.3 Pre-filtering

Pre-filtering is a technique where a phenomenon (such as spikes, form, etc.) in the surface are removed or greatly reduced, by other means, before filtration, thus removing or greatly reducing any effect the phenomenon can have on the filter's response. This approach has the advantage that once a method has been found to remove unwanted phenomenon then this method will work with any filter.

Form pre-filtering, involving removing the form of the surface before filtration, is a very common technique used in surface metrology. Less common is using scale space pre-filtering which involves removing singularities and other features of a certain size before filtration.

7 Conclusions

The paper has given an overview of the similarities and differences between the requirements and techniques used in mathematical approximation theory and filtration in surface metrology. Some recent work on filtration has been reported. It is hoped that this paper can generate some cross-fertilisation between the two areas of approximation theory and filtration.

Bibliography

1. ISO 11562 1996. Geometrical product specifications (GPS)- Surface texture: Profile method -Metrological characteristics of phase correct filters.
2. ISO/TS 16610–1. Geometrical product specifications (GPS) — Filtration Part 1: Overview and basic terminology.
3. ISO/TS 16610–20. Geometrical product specifications (GPS) — Filtration Part 20: Linear profile filters; Basic concept.
4. ISO/TS 16610–22. Geometrical product specifications (GPS) — Filtration Part 22: Linear profile filters; Spline filters.
5. ISO/TS 16610–29. Geometrical product specifications (GPS) — Filtration Part 29: Linear profile filters; Spline wavelets.
6. ISO/TS 16610–31. Geometrical product specifications (GPS) — Filtration Part 31: Robust profile filters; Gaussian regression filters.
7. ISO/TS 16610–32. Geometrical product specifications (GPS) — Filtration Part 32: Robust profile filters; Spline filters.
8. ISO/TS 16610–40. Geometrical product specifications (GPS) — Filtration Part 40: Morphological profile filters; Basic concepts.
9. ISO/TS 16610–41. Geometrical product specifications (GPS) — Filtration Part 41: Morphological profile filters; Disk and horizontal line segment filters.
10. ISO/TS 16610–49. Geometrical product specifications (GPS) — Filtration Part 49: Morphological profile filters; Scale Space Techniques.

11. ISO/TS 16610–60. Geometrical product specifications (GPS) — Filtration Part 60: Linear areal filters; Basic concepts.
12. X. Q. Jiang, L. A. Blunt and K. J. Stout. Development of a lifting wavelet representation for surface characterization, Proc. R. Soc. Lond. A **456** (2000), 2283–2313.
13. W. Sweldens. The lifting scheme: A construction of second generation wavelets, SIAM J. Math. Anal, **29** (1997), No. 2, 511–546.
14. X. Q. Xiang, L. A. Blunt and K. J. Stout. Application of the lifting wavelet to rough surfaces. Precision Engineering **25** (2001), 83–89.
15. J. Serra. *Image Analysis and Mathematical Morphology Vol. 1*, Academic Press, New York, 1982.
16. G. Mathron. *Random Sets and Integral Geometry*, John Wiley & Sons, New York, 1976.
17. P. J. Huber. *Robust Statistics*, John Wiley & Sons, New York, 1981.

Chapter 4

Radial Basis Functions

Preceding Page Blank

Applications of radial basis functions: Sobolev-orthogonal functions, radial basis functions and spectral methods

M.D. Buhmann

Mathematisches Institut, Justus-Liebig University, 35392 Giessen, Germany
buhmann@uni-giessen.de

A. Iserles

DAMTP, University of Cambridge, Silver Street, Cambridge, CB3 9EW, UK
ai@amtp.cam.ac.uk

S.P. Nørsett

Department of Mathematics, Norwegian University of Science and Technology, Trondheim, Norway
norsett@math.ntnu.no

Abstract

In this paper we consider an application of Sobolev-orthogonal functions and radial basis function to the numerical solution of partial differential equations. We develop the fundamentals of a spectral method, present examples via reaction–diffusion partial differential equations and discuss briefly some links with theory of wavelets.

1 Introduction

Radial basis functions are a well-known and useful tool for functional approximation in one or more dimensions. The general form of approximations is always a linear combination (finite or infinite) number of shifts of a single function, the *radial basis function.* In more than one dimension, this function is made rotationally invariant by composing a univariate function, usually called ϕ, with the Euclidean norm. In one dimension such approximation usually simplifies to univariate polynomial splines. For a recent review of radial basis function approximations, see [5].

This note is about applications for radial basis functions and other approximation schemes such as Sobolev-orthogonal polynomials and more general Sobolev-orthogonal functions to the numerical solution of partial differential equations. The basic ideas stem from the theory of Sobolev-orthogonal polynomials ([13]), and in this paper there is a remarkable connection developed between applications of Sobolev-orthogonality with radial basis functions (e.g. [5]), and wavelets are mentioned as well (e.g. [8, 9]). Sobolev-

orthogonal polynomials are a device to extend the standard theory of orthogonal polynomials (see, for instance, [12]) by requiring orthogonality with respect to non-selfadjoint inner products of the form

$$(f,g)_\lambda = \int_a^b f(x)g(x)\,\mathrm{d}x + \lambda \int_a^b f'(x)g'(x)\,\mathrm{d}x$$

for a positive parameter λ and a suitable interval (a,b), $a,b \in \mathbb{R} \cup \{\pm\infty\}$. The $\mathrm{d}x$ in the two integrals is often replaced by more general Borel measures, $\mathrm{d}\psi$, say. The scheme which we want to discuss in this short article is one of spectral type: in lieu of e.g. finite element spaces as underlying piecewise polynomial approximation spaces for the solution, we take purpose-build approximations which make the linear systems which we need to solve particularly simple, sometimes even diagonal.

Therefore, in the first instance, we develop a theory of applying Sobolev-orthogonal polynomial basis functions for the numerical solution of partial differential equations via a spectral method. Then we extend this idea to general classes of radial basis function-type methods, where shift-invariant approximation spaces are generated with Sobolev-orthogonal basis functions. Due to the introductory character of this paper, our discussion is restricted to relatively simple cases. Our presentation is illustrated with the one-dimensional reaction–diffusion partial differential equation.

This is the place to note that radial basis functions have found a number of other applications in the discretisation of PDEs. Thus, for example, Driscoll and Fornberg [10] have used fast-converging 'flat' multiquadrics in pseudospectral methods, while Frank and Reich [11] applied radial basis functions with particle methods in order to conserve enstrophy in the solution of certain shallow-water equations. Our application is of an altogether different nature.

1.1 Examples of PDEs and Sobolev-orthogonality

Consider the partial differential equation

$$\frac{\partial u}{\partial t} = \nabla\,(a\nabla u) + bu + c, \tag{1.1}$$

where $u = u(\mathbf{x},t)$ is of sufficient smoothness with respect to $\mathbf{x}$ and t, $\mathbf{x}$ is given in a cube $\mathcal{V} \subset \mathbb{R}^d$ (more generally, in a finite domain), $t \geq 0$, $a = a(\mathbf{x}) > 0$, $b = b(\mathbf{x})$ and $c = c(\mathbf{x})$. We impose zero Dirichlet boundary conditions. The stipulation of cube as a domain and zero Dirichlet conditions is unduly restrictive, but it will suffice for the short presentation in this paper and adequately illustrate the main novel concepts in our presentation. In the next section, we shall also introduce a nonlinearity into the underlying PDE.

We wish to approximate the solution $u(\mathbf{x},t)$ as a finite linear combination of the generic form

$$u(\mathbf{x},t) = \sum_{l=1}^{m} \alpha_l(\mathbf{x})w_l(t),$$

where t is nonnegative and $\mathbf{x}$ resides in the domain. In the sequel we shall also use expansions into infinite series with $l \in \mathbb{Z}$. Thus, a Galerkin ansatz (in the usual L_2 inner product on $\mathbb{R}^d$ which we denote by $(\cdot,\cdot)$ in contrast to the specialised Sobolev-inner

product $(\cdot,\cdot)_\lambda$ above) gives

$$\sum_{l=1}^{m}(\alpha_l,\alpha_k)w_l' = \sum_{l=1}^{m}(\nabla(a\nabla\alpha_l),\alpha_k)\,w_l + \sum_{l=1}^{m}(b\alpha_l,\alpha_k)w_l + (c,\alpha_k), \qquad k=1,2,\ldots,m.$$

Integration by parts in the second term above and substitution of the requisite zero boundary conditions yield the alternative formulation

$$\sum_{l=1}^{m}(\alpha_l,\alpha_k)w_l' = -\sum_{l=1}^{m}(a\nabla\alpha_l,\nabla\alpha_k)\,w_l + \sum_{l=1}^{m}(b\alpha_l,\alpha_k)w_l + (c,\alpha_k), \qquad k=1,2,\ldots,m. \tag{1.2}$$

We solve the ODE system (1.2) with respect to t, for example with the backward Euler scheme (we use backward Euler for the sake of simplicity, but it should be noted that the same analysis applies to any implicit multistep method, because our use of Sobolev-orthogonality is only linked to the implicitness of the solution method)

$$w_l^{n+1} = w_l^n + \Delta t F_l(\mathbf{w}^{n+1}), \qquad n\in\mathbb{Z}_+, \quad l=1,2,\ldots,m, \tag{1.3}$$

where the function F_l is given implicitly by the equations (1.2) and where $\mathbf{w}^{n+1}$ in the expression above is the vector with components w_l^{n+1}, $l=1,2,\ldots,m$. Let us now multiply expression (1.3) by (α_l,α_k) and sum up for $l=1,2,\ldots,m$. Then, exploiting (1.2), a little algebra yields

$$\begin{aligned}&\sum_{l=1}^{m}\left\{\int_{\mathcal{V}}[1-\Delta t b(\mathbf{x})]\,\alpha_l(\mathbf{x})\alpha_k(\mathbf{x})\mathrm{d}\mathbf{x} + \Delta t\int_{\mathcal{V}} a(\mathbf{x})\nabla^T\alpha_l(\mathbf{x})\nabla\alpha_k(\mathbf{x})\mathrm{d}\mathbf{x}\right\} w_l^{n+1}\\ =\;&\sum_{l=1}^{m}\int_{\mathcal{V}}\alpha_l(\mathbf{x})\alpha_k(\mathbf{x})\mathrm{d}\mathbf{x}w_l^n + \int_{\mathcal{V}} c(\mathbf{x})\alpha_k(\mathbf{x})\mathrm{d}\mathbf{x}.\end{aligned} \tag{1.4}$$

The connection with Sobolev-inner products is clear. Indeed, let us now choose the set $\mathbf{W}_{m,n} := \{w_1, w_2, \ldots, w_m\}$ as a set of functions that are orthogonal with respect to the homogeneous Sobolev $\overset{\circ}{H}_{d,2}$ inner product (see, e.g., [13])

$$\langle f,g\rangle_{\Delta t} := \int_{\mathcal{V}}[1-\Delta t b(\mathbf{x})]f(\mathbf{x})g(\mathbf{x})\mathrm{d}\mathbf{x} + \Delta t\int_{\mathcal{V}} a(\mathbf{x})\nabla^T f(\mathbf{x})\nabla g(\mathbf{x})\mathrm{d}\mathbf{x} \tag{1.5}$$

(this of course requires that $\Delta t b(\mathbf{x}) \le 1$, hence may restrict in a minor way the choice of the time step Δt). Further below we shall also use infinite sets $\mathbf{W}$ instead of the finite set $\mathbf{W}_{m,n}$. It is important to note that in general the Sobolev inner-product depends upon the step size. Subject to this formulation, the linear system (1.4) diagonalises and its numerical solution becomes trivial. We turn now to a more elaborate example in the next subsection, namely the reaction-diffusion equation.

1.2 Reaction-diffusion as a paradigm for nonlinear PDEs

Let us consider the nonlinear partial differential equation

$$\frac{\partial u}{\partial t} = \nabla(a\nabla u) + f(u), \tag{1.6}$$

where otherwise all the quantities are as in (1.1), including the boundary conditions. Suppose that an approximation u^n to $u(\mathbf{x}, n\Delta t)$ is available at all the spatial grid points. We commence by interpolating u^n to requisite precision by some function v. Thus, v is defined throughout the cube $\mathcal{V}$ and coincides with u^n at the grid points. This allows us to linearise the source function f about u^n, the outcome being

$$\frac{\partial u}{\partial t} = \nabla(a\nabla u) + c + bu + g(u), \tag{1.7}$$

where

$$\begin{aligned} b(\mathbf{x}) &= f'(v(\mathbf{x})), \\ c(\mathbf{x}) &= f(v(\mathbf{x})) - f'(v(\mathbf{x}))v(\mathbf{x}), \\ g(\mathbf{x}, u) &= f(u) - f(v(\mathbf{x})) - f'(v(\mathbf{x}))[u - v(\mathbf{x})]. \end{aligned}$$

Note that

$$g(\mathbf{x}, u) = O(|u - v|^2).$$

We can now solve the nonlinear system (1.7) by functional iteration, i.e. by letting as a start

$$w_l^{n+1,0} = w_l^n, \qquad l = 1, 2, \ldots, m,$$

and recurring, employing the inner product (1.5),

$$\begin{aligned} &\sum_{l=1}^{m} \langle \alpha_l, \alpha_k \rangle_{\Delta t} w_l^{n+1,j+1} \\ = &\sum_{l=1}^{m} (\alpha_l, \alpha_k) w_l^n + \left(g\left(\cdot, \sum_{l=1}^{m} \alpha_l w_l^{n+1,j} \right), \alpha_k \right), \quad k = 1, 2, \ldots, m, \end{aligned} \tag{1.8}$$

for $j \in \mathbb{Z}_+$.

If, as in the previous subsection, we choose $\mathbf{W}_m$ so as to diagonalise the linear system, each step of (1.8) becomes relatively cheap. Hence this approach might offer a realistic means to derive spectral approximation to nonlinear PDEs. Indeed, a special one-dimensional case can be treated straightforwardly and it is presented in the sequel.

1.3 The one-dimensional case using polynomial splines

Let (1.1) be given in one space dimension and without source terms, whence it becomes the familiar diffusion equation with variable diffusion coefficient,

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(a \frac{\partial u}{\partial x} \right).$$

Thus, provided that $0 \leq x \leq 1$ and t nonnegative, we require the 'usual' Sobolev orthogonality [13] with respect to the inner product

$$\langle f, g \rangle_{\Delta t} = (f, g)_1 = \int_0^1 f(x) g(x) \mathrm{d}\varphi(x) + \int_0^1 f'(x) g'(x) \mathrm{d}\psi(x),$$

where

$$\frac{\mathrm{d}\varphi(x)}{\mathrm{d}x} = 1 - \Delta t b, \qquad \frac{\mathrm{d}\psi(x)}{\mathrm{d}x} = \Delta t a.$$

We emphasise again the dependence of the Sobolev-inner product on the step size. Taking the approach of the previous subsection as our point of departure, an obvious option is to use Sobolev-orthogonal polynomials. An alternative approach which can be worked out explicitly and which we wish to demonstrate in this subsection, is to use univariate polynomial spline approximations. It has the advantage of being more amenable to a generalisation to several space dimensions.

We suppose that the unit-interval $[0,1]$ is divided into N intervals of length $h := \frac{1}{N}$ and consider a piecewise-quadratic basis of continuous functions $s_1, s_2, \ldots, s_N$ such that

$$s_l(x) := \begin{cases} \frac{1}{h}[x-(l-1)h] + \alpha_l(x-lh)[x-(l-1)h], & (l-1)h \le x \le lh, \\ \frac{1}{h}[(l+1)h-x] + \beta_l(x-lh)[x-(l+1)h], & lh \le x \le (l+1)h, \\ 0, & |x-lh| \ge h. \end{cases}$$

Clearly, s_l is a continuous, $C[0,1]$ cardinal function of Lagrange interpolation at the knots (hence, a quadratic spline with double knots, cf., Powell [16], the added degree of freedom taken up by the requirement of Sobolev-orthogonality). Next, we need just to impose Sobolev orthogonality, and solve for the coefficients α_l and β_l. This is equivalent to the requirement that

$$\langle s_l, s_{l+1} \rangle_{\Delta t} = 0, \qquad l = 1, 2, \ldots, N-1.$$

In the special case $a(x) \equiv 1$, $b(x), c(x) \equiv 0$, we have $\varphi(x) = x$, $\psi(x) = \Delta t x$ and

$$\begin{aligned} \langle s_l, s_{l+1} \rangle_{\Delta t} &= \int_0^h \left[\frac{x}{h} + \alpha_{l+1}(x-h)x\right] \cdot \left[\frac{h-x}{h} + \beta_l x(x-h)\right] \mathrm{d}x \\ &\quad + \Delta t \int_0^h \left(\frac{1}{h} + 2\alpha_{l+1}x - \alpha_{l+1}h\right)\left(-\frac{1}{h} + 2\beta_l x - \beta_l h\right) \mathrm{d}x \\ &= h\int_0^1 \left[\xi + \alpha_{l+1}h^2(\xi-1)\xi\right] \cdot \left[1 - \xi - \beta_l h^2 \xi(1-\xi)\right] \mathrm{d}\xi \\ &\quad - \frac{\Delta t}{h}\int_0^1 (1 + 2\alpha_{l+1}\xi - \alpha_{l+1})(1 + \beta_l - 2\beta_l\xi)\mathrm{d}\xi \\ &= h\left[\left(\frac{1}{6} - \frac{h^2}{12}(\alpha_{l+1} + \beta_l) + \frac{h^4}{30}\alpha_{l+1}\beta_l\right) + \frac{\Delta t}{h^2}\left(-1 + \frac{1}{3}\alpha_{l+1}\beta_l\right)\right]. \end{aligned}$$

Let $\mu = \Delta t / h^2$ be the Courant number. Since we have two degrees of freedom for each l and because each equation is otherweise independent of l, we may fix $\alpha \equiv \alpha_l \equiv \beta_l$. Then, letting $\hat{\alpha} := h^2\alpha$, requiring $\langle s_l, s_{l+1} \rangle_{\Delta t} = 0$ is equivalent to

$$5 - 5\hat{\alpha} + \hat{\alpha}^2 + 10\mu\alpha^2 - 30\mu = 0 \tag{1.9}$$

or

$$(10\mu + h^4)\alpha^2 - 5h^2\alpha + 5 - 30\mu = 0.$$

We wish to solve this quadratic equation for α for a suitable range of Courant numbers. Indeed, the equation (1.9) has two real solutions α for every $\mu > \frac{1}{6}$ if h is small enough, since its discriminant is

$$(120\mu + 5)h^4 + 1200\mu^2 - 200\mu.$$

In the case $\mu = \frac{1}{6}$ each s_l reduces, upon the choice of $\hat{\alpha} = 0$, to a *chapeau* function. Otherwise we obtain $\alpha = O(1)$. We may give up a small support, characteristic of spline functions (which, anyway, is of marginal importance, since we do not solve linear systems!). This is a case discussed in the next section. Another obvious alternative is to construct an orthogonal basis from *chapeau* functions. This, however, is easily seen to be identical to the LU factorization of the standard FEM matrix

$$\begin{bmatrix} \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 & \cdots & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \\ 0 & \cdots & 0 & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ 0 & \cdots & 0 & 0 & 0 & \frac{1}{6} & \frac{2}{3} \end{bmatrix}.$$

2 Applications of radial basis functions and wavelets

2.1 Sobolev-orthogonal translates of a radial basis function

In this section, we wish to develop a more general approach employing the concepts of wavelets and radial basis functions and employ shift-invariant spaces of approximations for our spectral methods. We begin by giving up the compactness of the domain $\mathcal{V}$ and work on the entire real line instead. For this, we shall demonstrate the use of Sobolev-inner products and shift-invariant spaces and concentrate solely on this part of the analysis in the present article. So, in particular, the set $\mathbf{W}$ above is of the form $\{\phi(\cdot - nh) \mid n \in \mathbb{Z}\}$. In the sequel we shall add several remarks about how to find compactly-supported ϕ that allow the treatment of partial differential equations on compact domains. We remark that n is no longer used for the time-steps in the differential equation solver but for the shifts of the radial functions.

To start with, we wish to find a function $\phi \in \mathbf{H}^2(\mathbb{R})$, where $\mathbf{H}^2(\mathbb{R})$ is a non-homogeneous Sobolev space, such that for a positive constant λ and positive spacing h it is true that

$$\int_{-\infty}^{\infty} \phi(x)\phi(x - hn)\,\mathrm{d}x + \lambda \int_{-\infty}^{\infty} \phi'(x)\phi'(x - hn)\,\mathrm{d}x = \delta_{0n}, \qquad n \in \mathbb{Z}. \tag{2.1}$$

We multiply both left- and right-hand-side of the general pattern (2.1) by $\exp(\mathrm{i}\theta n)$ and sum over $n \in \mathbb{Z}$,

$$\sum_{n=-\infty}^{\infty} \exp(\mathrm{i}\theta n) \left\{ \int_{-\infty}^{\infty} \phi(x)\phi(x - hn)\,\mathrm{d}x + \lambda \int_{-\infty}^{\infty} \phi'(x)\phi'(x - hn)\,\mathrm{d}x \right\} = 1,\ \theta \in [-\pi, \pi]. \tag{2.2}$$

In order to be able to exchange summation and integration and apply the Poisson summation formula (Stein and Weiss [17], p. 252) we make a number of assumptions. The version of the Poisson summation formula that we wish to use states that for a univariate function f with

$$|f(x)| = O\Big((1+|x|)^{-1-\epsilon}\Big)$$

and

$$|\hat{f}(x)| = O\Big((1+|x|)^{-1-\epsilon}\Big)$$

and positive ϵ, the following equality holds (note that the first bound in the above implies existence and continuity of the one-dimensional Fourier transform)

$$\sum_{j=-\infty}^{\infty} \hat{f}(\theta + 2\pi j) = \sum_{j=-\infty}^{\infty} \exp(\mathrm{i}\theta j) f(j).$$

Specifically, we assume that the following three decay estimates hold:

$$|\phi(x)| \leq c(1+|x|)^{-1-\epsilon},$$
$$|\phi'(x)| \leq c(1+|x|)^{-1-\epsilon},$$

and

$$|\hat{\phi}(\xi)| \leq c(1+|\xi|)^{-3-\epsilon},$$

where c is a generic positive constant, $\epsilon > 0$, $\hat{\phi}$ denotes the Fourier transform and we demand the faster rate of decay in the last display because we shall later require summability of translates of the Fourier transform multiplied by the square of its argument. Note in particular that the first decay condition renders the Fourier transform $\hat{\phi}$ continuous and well defined.

An example for a function ϕ that satisfies the three decay conditions above is the second divided difference of the multiquadric radial basis function [4] $\sqrt{r^2+C^2}$ that is

$$\phi(x) = \frac{1}{2}\sqrt{(x-1)^2+C^2} - \sqrt{x^2+C^2} + \frac{1}{2}\sqrt{(x+1)^2+C^2}.$$

Here, C is a positive constant parameter. The above function decays cubically [4] and its Fourier transform even decays exponentially due to the exponential decay of the modified Bessel function K_1 [1] that features in the generalised Fourier transform of the multiquadric, here stated only in the one-dimensional case,

$$-2C\frac{K_1(C|\xi|)}{|\xi|}$$

(cf. Jones [14]).

Once summation and integration are interchanged, (2.2) becomes

$$\int_{-\infty}^{\infty} \phi(x) \sum_{n=-\infty}^{\infty} \exp(\mathrm{i}\theta n)\phi(x-hn)\,\mathrm{d}x$$

$$+\lambda\int_{-\infty}^{\infty}\phi'(x)\sum_{n=-\infty}^{\infty}\exp(\mathrm{i}\theta n)\phi'(x-hn)\,\mathrm{d}x=1,\qquad \theta\in[-\pi,\pi], \tag{2.3}$$

or, applying the Poisson Summation Formula (Stein and Weiss, [17], p. 252)

$$\int_{-\infty}^{\infty}\phi(x)\sum_{n=-\infty}^{\infty}\exp\Big(\mathrm{i}h^{-1}x(\theta+2\pi n)\Big)\hat{\phi}\Big(h^{-1}(\theta+2\pi n)\Big)\,\mathrm{d}x+\mathrm{i}\lambda h^{-1} \tag{2.4}$$

$$\times\int_{-\infty}^{\infty}\phi'(x)\sum_{n=-\infty}^{\infty}\exp\Big(\mathrm{i}h^{-1}x(\theta+2\pi n)\Big)(\theta+2\pi n)\hat{\phi}\Big(h^{-1}(\theta+2\pi n)\Big)\,\mathrm{d}x=h,$$

where $\theta\in[-\pi,\pi]$. Because ϕ vanishes at infinity, integration by parts of the second term of (2.4) gives

$$\int_{-\infty}^{\infty}\phi(x)\sum_{n=-\infty}^{\infty}\exp\Big(\mathrm{i}h^{-1}x(\theta+2\pi n)\Big)\hat{\phi}\Big(h^{-1}(\theta+2\pi n)\Big)\,\mathrm{d}x$$

$$+\frac{\lambda}{h^2}\int_{-\infty}^{\infty}\phi(x)\sum_{n=-\infty}^{\infty}\exp(\mathrm{i}h^{-1}x(\theta+2\pi n))(\theta+2\pi n)^2\hat{\phi}(h^{-1}(\theta+2\pi n))\mathrm{d}x$$

$$=\sum_{n=-\infty}^{\infty}\hat{\phi}(h^{-1}(\theta+2\pi n))\hat{\phi}\Big(-h^{-1}(\theta+2\pi n)\Big)\Big[1+\lambda h^{-2}(\theta+2\pi n)^2\Big]=h.$$

Since ϕ is real, $\hat{\phi}(-\xi)=\overline{\hat{\phi}}(\xi)$, and this implies

$$\sum_{n=-\infty}^{\infty}|\hat{\phi}(h^{-1}(\theta+2\pi n))|^2(1+\lambda h^{-2}(\theta+2\pi n)^2)=h,\qquad \theta\in[-\pi,\pi]. \tag{2.5}$$

This is our condition that leads to the required Sobolev-orthogonality. In summary, we have established the following theorem.

Theorem 2.1 *If the decay conditions on ϕ, as stated above, hold in tandem with the expression* (2.5)*, then the required orthogonality condition* (2.1) *is satisfied.*

We note that, if we are given a ψ such that

$$\sum_{n=-\infty}^{\infty}\Big|\hat{\psi}\Big(h^{-1}(\theta+2\pi n)\Big)\Big|^2=h,\qquad \theta\in[-\pi,\pi], \tag{2.6}$$

then

$$\hat{\phi}(\xi):=\frac{\hat{\psi}(\xi)}{\sqrt{1+\lambda\xi^2}} \tag{2.7}$$

satisfies (2.5). This expression can be used to derive an explicit transformation which takes a ψ that satisfies (2.6), into a ϕ satisfying (2.5), although its practical computation may be nontrivial. Indeed, by the Parseval–Plancherel theorem [17], we get the useful identity

$$\phi(x)=\frac{1}{\pi\sqrt{\lambda}}\int_{-\infty}^{\infty}\psi(x-y)K_0\left(\frac{|y|}{\sqrt{\lambda}}\right)\,\mathrm{d}y, \tag{2.8}$$

which is a convolution and whose Fourier transform is therefore (2.7) (cf., for instance, Jones [14]). In (2.8), K_0 is the 0th modified Bessel function (Abramowitz and Stegun [1]) which is positive on positive reals and satisfies $K_0(t) \sim -\log t$ near zero and $K_0(t) \sim \sqrt{\pi/(2t)}\mathrm{e}^{-t}$ for large t, similar to the asymptotics we have used before for the K_1 modified Bessel function. Hence, by a lemma in [7], see also (Light and Cheney [15]) ϕ decays algebraically of a certain order if ψ does. Moreover, because $1/\sqrt{1+\lambda x^2}$ is positive, integer translates of ϕ are dense in L^2, say, provided that this is the case with integer translates of ψ [18].

In some trivial cases we may evaluate the integral (2.8) explicitly, for instance for $\psi(x) = \cos x$, where the integral is again a constant multiple of the cosine function (Abramowitz and Stegun [1]). Otherwise, the smoothness and fast exponential decay of the modified Bessel function can be used together with a quadrature formula.

We may now use the translates of such Sobolev-orthogonal functions in the spectral approximation of a PDE as above, letting $\mathbf{W} := \{\phi(\cdot - nh) \mid n \in \mathbb{Z}\}$.

An example of a function $\hat{\psi}$ that satisfies (2.5) is simply the characteristic function scaled by h of the interval $[-h\pi, h\pi]$. In that case, $|\psi(x)|$ decays like $1/|x|$. In fact, any ψ that satisfies $|\hat{\psi}(\xi)| \leq c(1+|\xi|)^{-1/2-\varepsilon}$ for positive ε can be made to satisfy (2.6) by subjecting it to the transformation

$$\hat{\psi}(\xi) \mapsto \tilde{\hat{\psi}}(\xi) := \frac{\sqrt{h}\hat{\psi}(\xi)}{\sqrt{\sum_{n=-\infty}^{\infty} |\hat{\psi}(\xi + h^{-1}2\pi n)|^2}}, \tag{2.9}$$

see for instance (Battle [2]). If ψ is compactly supported then the transformed $\tilde{\psi}$ will not necessarily be compact supported but decay exponentially [6].

In order to find a class of examples of *compactly supported* ψ that satisfy (2.6), see Daubechies [8] for her compactly supported scaling functions ψ which are fundamental for the construction of Daubechies wavelets. For example, the following conditions are sufficient for ψ which shall be defined by its Fourier transform to satisfy (2.6) for $h = 1$ (other h can be used by scaling):

$$\hat{\psi}(\xi) = \prod_{j=1}^{\infty} \tilde{h}\Big(\frac{\xi}{2^j}\Big),$$

where, for some suitable coefficients $\tilde{h}_k$,

$$\tilde{h}(\xi) = \sum_{k=0}^{2N-1} \tilde{h}_k \mathrm{e}^{-\mathrm{i}k\xi}$$

has to satisfy $\tilde{h}(0) = 1$, $\tilde{h}(\pi) = 0$, and

$$|\tilde{h}(\xi)|^2 + |\tilde{h}(\xi+\pi)|^2 = 1, \qquad \xi \in [-\pi, \pi].$$

For the construction of such $\tilde{h}$, see [8]. Compactly supported basis functions are important to approximate the numerical solution of a PDE as in the above example defined on

a compact $\mathcal{V}$. Moreover, any ψ with the aforementioned decay property can be made to satisfy (2.5) by the transformation

$$\hat{\psi}(\xi) \mapsto \frac{\sqrt{h}\hat{\psi}(\xi)}{\sqrt{\sum_{n=-\infty}^{\infty} |\hat{\psi}(\xi + h^{-1}2\pi n)|^2(1 + \lambda(\xi + h^{-1}2\pi n)^2}}. \tag{2.10}$$

They can also be found by applying the transformation (2.10) and using the transformation (2.9) as well.

We note finally, that for instance, when ψ is a B-spline then its translates are dense in L^2 if we allow h to become arbitrarily small (see, for instance, Powell [16]) and the last section of this paper).

2.2 Sobolev-orthogonal translates of a function in higher dimensions

Applying the approach of the previous subsection to the Sobolev inner product

$$\int_{\mathbf{R}^d} f(\mathbf{x})g(\mathbf{x})\,\mathrm{d}\mathbf{x} + \lambda \int_{\mathbf{R}^d} \nabla^T f(\mathbf{x})\nabla g(\mathbf{x})\,\mathrm{d}\mathbf{x},$$

the outcome is the orthogonality condition

$$\sum_{\mathbf{n}\in\mathbb{Z}^d} |\hat{\phi}(h^{-1}(\theta + 2\pi\mathbf{n}))|^2(1 + \lambda h^{-2}\|\theta + 2\pi\mathbf{n}\|^2) = h^d, \qquad \theta \in [-\pi, \pi]^d, \tag{2.11}$$

which replaces (2.5). We are now also interested in the more general case of Sobolev-type inner products

$$\int_{\mathbf{R}^d} f(\mathbf{x})g(\mathbf{x})\mu(\mathbf{x})\,\mathrm{d}\mathbf{x} + \lambda \int_{\mathbf{R}^d} \nabla^T f(\mathbf{x})\nabla g(\mathbf{x})\nu(\mathbf{x})\,\mathrm{d}\mathbf{x},$$

where the weights μ and ν are positive. Here the orthogonality condition becomes more complicated. Specifically, it is

$$\sum_{\mathbf{n}\in\mathbb{Z}^d} \hat{\phi}_\mu\Big(h^{-1}(\theta + 2\pi\mathbf{n})\Big)\overline{\hat{\phi}_\mu}\Big(h^{-1}(\theta + 2\pi\mathbf{n})\Big) + \lambda h^{-2}\hat{\phi}_\nu\Big(h^{-1}(\theta + 2\pi\mathbf{n})\Big)\overline{\hat{\phi}_\nu}\Big(h^{-1}(\theta + 2\pi\mathbf{n})\Big) = h^d, \qquad \theta \in [-\pi, \pi]^d,$$

where

$$\begin{aligned} \hat{\phi}_\mu &:= \hat{\phi} * \widehat{\sqrt{\mu}}, \\ \hat{\phi}_\nu &:= (\|\cdot\| \times \hat{\phi}) * \widehat{\sqrt{\nu}}, \end{aligned}$$

and $*$ denotes continuous convolution, used as in (2.8), where ψ is convolved with a modified Bessel function.

2.3 Error estimates

We can offer error estimates for the Sobolev-orthogonal bases, firstly, in the case when ϕ is a univariate spline of fixed degree m, say, with knots on $h\mathbb{Z}$, and, secondly, in the

case when ϕ is a linear combination of translates of the radial Gauss kernel

$$e^{-\alpha^2 x^2/2}, \qquad x \in \mathbb{R},$$

along $h\mathbb{Z}$. In the former case it is known that the *uniform* approximation error to a sufficiently smooth function from the linear space spanned by $\phi(\cdot - nh)$, $n \in \mathbb{Z}$, is at most a constant multiple of h^{m+1} ([16]). We have already mentioned that we require $\lambda = O(h^2)$, therefore it can be deduced by twofold integration by parts that the Sobolev error is indeed $O(h^{m+1})$. This can be generalized in a straightforward way to higher dimensions by tensor-product B-splines.

Our $L^2(\mathbb{R})$ error estimates can be carried out as follows: Let f be a band-limited function, that is, one with a compactly-supported Fourier transform, which satisfies such assumptions that imply that the best least-squares approximation using a Sobolev inner product

$$s_h(x) = \sum_{n=-\infty}^{\infty} \langle f, \phi(\cdot - hn) \rangle_{\lambda,h} \phi(x - nh), \qquad x \in \mathbb{R}, \tag{2.12}$$

is well defined. For instance, we may require that $\langle f, f \rangle_{\lambda,h} < \infty$, as well as sufficient decay of the radial basis function ϕ, i.e.

$$\begin{aligned} |\phi(r)| &\leq c(1+|r|)^{-1-\varepsilon}, \\ |\phi'(r)| &\leq c(1+|r|)^{-1-\varepsilon}, \\ |\hat{\phi}(r)| &\leq c(1+|r|)^{-1-\varepsilon} \end{aligned}$$

for a positive ε. Here $\langle \cdot, \cdot \rangle_{\lambda,h}$ is the Sobolev inner product which we study in this note and it is helpful to emphasise its dependence on h in the subscript. We begin with the piecewise polynomial, i.e. spline, case. Hence, let ϕ be from the space of splines of degree m with knots on $h\mathbb{Z}$ such that its translates are Sobolev orthogonal.

Theorem 2.2 *Subject to the assumptions of the last paragraph, we have the error estimate*

$$\|s_h - f\|_2 = O(h^{m+1}), \qquad h \to 0. \tag{2.13}$$

Proof: We shall establish in the course of this proof an error estimate for the first derivative of the error function in (2.13), so that an order of convergence can also be concluded for the norm associated with our Sobolev inner product. Indeed, because the Fourier transform is an $L^2(\mathbb{R})$ isometry, we may prove (2.13) by considering

$$\|\hat{s}_h - \hat{f}\|_2 \tag{2.14}$$

instead of the left-hand side of (2.13). The Fourier transform of (2.12) is

$$\hat{s}_h(\theta) = \sum_{n=-\infty}^{\infty} \langle f, \phi(\cdot - nh) \rangle_{\lambda,h} e^{-i\theta nh} \hat{\phi}(\theta), \qquad \theta \in \mathbb{R}.$$

The absolute convergence of the above is guaranteed by the decay conditions on ϕ. Hence the square of (2.14) is, by the Parseval–Plancherel Formula and periodisation of

the integrand with respect to θ,

$$\begin{aligned}
&\int_{-\infty}^{\infty}\left|\hat{f}(\theta)-\sum_{n=-\infty}^{\infty}\langle f,\phi(\cdot-nh)\rangle_{\lambda,h}\mathrm{e}^{-\mathrm{i}\theta hn}\hat{\phi}(\theta)\right|^2\mathrm{d}\theta\\
&=\int_{-\infty}^{\infty}\left|\hat{f}(\theta)-\sum_{n=-\infty}^{\infty}\int_{-\infty}^{\infty}\hat{f}(\xi)\hat{\phi}(\xi)\mathrm{e}^{\mathrm{i}\xi hn}(1+\lambda\xi^2)\,\mathrm{d}\xi\mathrm{e}^{-\mathrm{i}\theta hn}\hat{\phi}(\theta)\right|^2\mathrm{d}\theta\\
&=\int_{-\pi/h}^{\pi/h}\sum_{k=-\infty}^{\infty}\left|\hat{f}(\theta+2\pi k/h)-\hat{\phi}(\theta+2\pi k/h)\right.\\
&\quad\left.\times\sum_{n=-\infty}^{\infty}\int_{-\infty}^{\infty}\hat{f}(\xi)\hat{\phi}(\xi)\mathrm{e}^{\mathrm{i}\xi nh}(1+\lambda\xi^2)\,\mathrm{d}\xi\mathrm{e}^{-\mathrm{i}\theta nh}\right|^2\mathrm{d}\theta. \qquad (2.15)
\end{aligned}$$

The $(1+\lambda\xi^2)$ term in the above comes from the derivative in the Sobolev inner product and Fourier transform. Because f is band-limited, for small enough h (2.15) assumes the form

$$\int_{-\pi/h}^{\pi/h}\sum_{k=-\infty}^{\infty}\left|\hat{f}(\theta)\delta_{0k}-\hat{\phi}(\theta+2\pi k/h)\sum_{n=-\infty}^{\infty}\int_{-\infty}^{\infty}\hat{f}(\xi)\hat{\phi}(\xi)\mathrm{e}^{\mathrm{i}\xi nh}(1+\lambda\xi^2)\,\mathrm{d}\xi\mathrm{e}^{-\mathrm{i}\theta nh}\right|^2\mathrm{d}\theta. \qquad (2.16)$$

Using again the band limitedness of f, together with the Poisson Summation Formula, (2.16) can be brought into the form

$$\begin{aligned}
&\int_{-\pi/h}^{\pi/h}\sum_{k=-\infty}^{\infty}\left|\hat{f}(\theta)\delta_{0k}-\hat{\phi}(\theta+2\pi k/h)\right.\\
&\quad\left.\times\frac{1}{h}\sum_{n=-\infty}^{\infty}\hat{f}(\theta+2\pi n/h)\hat{\phi}(\theta+2\pi n/h)\Big(1+\lambda(\theta+2\pi n/h)^2\Big)\right|^2\mathrm{d}\theta\\
&=\int_{-\pi/h}^{\pi/h}\sum_{k=-\infty}^{\infty}\left|\hat{f}(\theta)\delta_{0k}-h^{-1}\hat{\phi}(\theta+2\pi k/h)\hat{f}(\theta)\hat{\phi}(\theta)(1+\lambda\theta^2)\right|^2\mathrm{d}\theta. \qquad (2.17)
\end{aligned}$$

In the case when ϕ is in the aforementioned spline space, it can be expressed as the inverse Fourier transform of

$$\hat{\phi}(\xi)=\frac{\sqrt{h}\hat{r}(\xi)}{\sqrt{\sum_{n=-\infty}^{\infty}|\hat{r}(\xi+h^{-1}2\pi n)|^2\big(1+\lambda(\xi+h^{-1}2\pi n)^2\big)}},\qquad \xi\in\mathbb{R}, \qquad (2.18)$$

where $\hat{r}(\xi)=\xi^{-m-1}$. This follows from (2.5) and from the fact that all splines from our space are linear combinations of integer translates of $r(x):=|x|^m$, whose generalised Fourier transform is a multiple of ξ^{-m-1} [14]. Since any constant factors in front of the function ξ^{-m-1} in $\hat{r}$ cancel in the expression for $\hat{\phi}$ above, we have ignored them

straightaway. Substituting (2.18) into (2.17), we get the integral over $[-\pi/h, \pi/h]$ of

$$\sum_{k=-\infty}^{\infty} \left| \hat{f}(\theta)\delta_{0k} - \frac{\hat{r}(\theta + h^{-1}2\pi k)\hat{r}(\theta)}{\sum_{n=-\infty}^{\infty} |\hat{r}(\theta + h^{-1}2\pi n)|^2(1 + \lambda(\theta + h^{-1}2\pi n)^2)} \hat{f}(\theta)(1 + \lambda\theta^2) \right|^2 . \quad (2.19)$$

Considering (2.19) for each m separately, it follows from (2.19) and from $\hat{r}(\xi) = \xi^{-m-1}$ that our claim is true. Indeed for the sum over all terms with $k \neq 0$, it is evident that we obtain a factor of h^{2m+2} from the numerator, because the denominator is periodic, containing one term independent of h, and the nonvanishing expression $h^{-1}2\pi k$ in the argument of $\hat{r}(\theta + h^{-1}2\pi k)$ guarantees $\hat{r}(\theta + h^{-1}2\pi k) \sim h^{m+1}$ due to $\hat{r}(\xi) = \xi^{-m-1}$. Of course, the squares then taken provide the h^{2m+2} instead of h^{m+1}.

On the other hand, for $k = 0$, we have for small enough h

$$\begin{aligned} & \left| \hat{f}(\theta) - \frac{|\hat{r}(\theta)|^2(1 + \lambda\theta^2)\hat{f}(\theta)}{\sum_{n=-\infty}^{\infty} |\hat{r}(\theta + h^{-1}2\pi n)|^2(1 + \lambda(\theta + h^{-1}2\pi n)^2)} \right|^2 \\ = \; & |\hat{f}(\theta)|^2 \left| \frac{\sum_{n\neq 0} |\hat{r}(\theta + h^{-1}2\pi n)|^2(1 + \lambda(\theta + h^{-1}2\pi n)^2)}{1 + \sum_{n\neq 0} |\hat{r}(\theta + h^{-1}2\pi n)|^2(1 + \lambda(\theta + h^{-1}2\pi n)^2)} \right|^2 \end{aligned}$$

which is also $O(h^{2m+2})$, as required, because the numerator provides an $O(h^{2m})$, according to the rate of the decay of $\hat{r}$ and the power of h in its argument. This is then squared to provide $O(h^{4m}) = O(h^{2m+2})$.

As for the derivatives, one only has to multiply the Fourier transform of the error function in (2.14) with θ, and we get the same error estimate by multiplying the integrands in all the following integrals with $|\theta|^2$. □

The same analysis remains valid when considering integer translates of the Gauss kernel $e^{-\gamma^2 x^2/2}$ in order to form ϕ. In this case we make use of the fact that the Gauss kernel has a Fourier transform which is a multiple of $e^{-x^2/(2\gamma^2)}$. We put this instead of $\hat{r}$ into (2.19), and we then get arbitrarily-high orders of convergence from (2.14) as long as we take $\gamma = O(h)$, see also [3]. For this choice ϕ is exponentially decaying, whereas for splines of degree m we merely get algebraic decay at infinity of order $-m - 1$.

Bibliography

1. Abramowitz, M. and I.A. Stegun (1970) Handbook of Mathematical Functions, Dover Publications.
2. Battle, G. (1987)"A block-spin construction of ondelettes, Part I: Lemarié functions", Comm. Math. Phys.
3. Beatson, R.K. and W.A. Light (1992) "Quasi-interpolation in the absence of polynomial reproduction", in Numerical Methods of Approximation Theory, D. Braess and L.L. Schumaker (eds.), Birkhäuser-Verlag, Basel, 21–39.

4. Buhmann, M.D. (1988) "Convergence of univariate quasi-interpolation using multiquadrics", IMA Journal of Numerical Analysis 8, 365–384.
5. Buhmann, M.D. (2000) "Radial basis functions", Acta Numerica 9, 1–38.
6. Chui, C.K. (1992) An Introduction to Wavelets, Academic Press, New York.
7. Buhmann, M.D. and N. Dyn (1993) "Spectral convergence of multiquadric interpolation", Proc. Edinburgh Math. Soc. 36, 319–333.
8. Daubechies, I. (1988) "Orthogonal bases of compactly supported wavelets", Comm. Pure Appl. Maths 16, 909–996.
9. DeVore, R.A. and B. Lucier (1992) "Wavelets", Acta Numerica 1, 1–55.
10. Driscoll, T.A. and Fornberg, B. (2001), "Interpolation in the limit of increasingly flat radial basis functions", to appear in Computers and Maths & Applics.
11. Frank, J. and Reich, S. (2001) "A particle-mesh method for the shallow water equations near geostrophic balance", Tech. Rep., Imperial College, London.
12. Gautschi, W. (1996) "Orthogonal polynomials: applications and computation", Acta Numerica 5, 45–119.
13. Iserles, A., P.E. Koch, S.P. Nørsett and J.M. Sanz-Serna (1991) "On polynomials orthogonal with respect to certain Sobolev inner products", J. Approx. Th. 65, 151–175.
14. Jones, D.S. (1982) The Theory of Generalised Functions, Cambridge University Press, Cambridge.
15. Light, W.A. and E.W. Cheney (1992) "Quasi-interpolation with translates of a function having non-compact support", Constr. Approx. 8, 35–48.
16. Powell, M.J.D. (1981) Approximation Theory and Methods, Cambridge University Press, Cambridge.
17. Stein, E.M. and G.Weiss (1971) Introduction to Fourier Analysis on Euclidean Spaces, Princeton.
18. Wiener, N. (1933) The Fourier Integral and Certain of its Applications, Cambridge University Press, Cambridge.

Approximation with the radial basis functions of Lewitt

J. J. Green

Dept. Applied Mathematics, University of Sheffield, UK.
j.j.green@sheffield.ac.uk

Abstract

R. M. Lewitt has introduced a family of compactly supported radial basis functions which are particularly useful in discretising for inversion ill-posed problems involving line integrals. We consider some practical considerations in their use and implementation, compare square and triangular grids of the functions in two dimensions, and describe some particularly favourable choices of the defining parameters.

1 Introduction

In the article [5], R. M. Lewitt introduced a family of window functions

$$\psi(r) = \begin{cases} (1-(r/a)^2)^{m/2} I_m(\alpha(1-(r/a)^2)^{1/2})/I_m(\alpha), & 0 \le r \le a, \\ 0, & r > a, \end{cases} \tag{1.1}$$

where I_m is the modified Bessel function of order m (see Ch. III, 3.7 [13]). The implicit dependence of ψ on the parameters $\alpha > 0$, $a > 0$ and $m \in \mathbf{N}$ is discussed below. Lewitt's motivation for studying these functions is the use of translates of the radially symmetric function

$$\Psi(x) = \psi(\|x\|) \qquad (x \in \mathbf{R}^d)$$

(see Figure 1) as a basis for the discretisation of tomographic problems [8, 9]. Such a basis overcomes a number of difficulties associated with the usual, pixel-based, representation in problems involving the recovery of function from a set of line, curve or strip integrals across its domain, while retaining the advantage of a *sparse* discretisation. The author's interest in these functions arises in their application to a Radon-like problem in the remote sensing of ocean waves [15], a detailed exposition of which may be found in [3].

2 Discretising x-ray problems

The discretisation of an x-ray transform inversion problem with Lewitt's basis is straightforward. Given a set of *centres* $x_i \in \mathbf{R}^d$, one represents the (unknown) function f as a linear combination of the translates of Ψ,

$$f(x) = \sum_i \xi_i \Psi(x - x_i) \qquad (x \in \mathbf{R}^d). \tag{2.1}$$

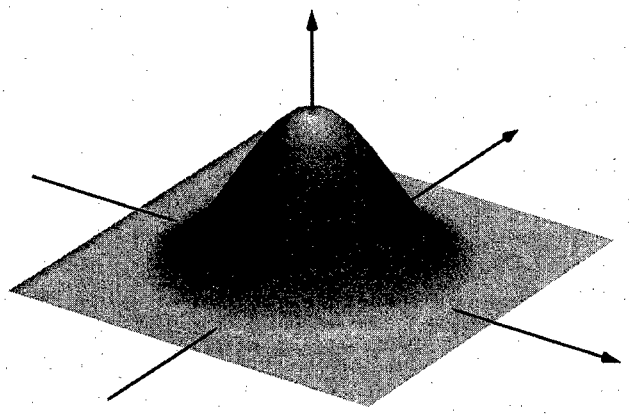


FIG. 1. Lewitt's radial basis function in dimension 2 with $m = 2$, $\alpha = 3$.

The given data in such problems are the values I_j of integrals of f over lines (or more generally, submanifolds) L_j

$$I_j = \int_{L_j} f(x) = \sum_i \xi_i \int_{L_j} \Psi(x - x_i). \tag{2.2}$$

The latter integral in (2.2) is the *projection* or *Abel transform* of f, which can be calculated explicitly in the linear case. For a line L_j whose closest point to x_i is at a distance s from it, and with the dependence of ψ on m here made explicit,

$$2\int_0^\infty \psi_m(\sqrt{s^2 + t^2})dt = a\frac{I_{m+1/2}(\alpha)}{I_m(\alpha)}\left(\frac{2\pi}{\alpha}\right)^{1/2}\psi_{m+1/2}(s)$$

(see A7, [5]). Thus (2.2) reduces to a linear system which may be solved for the coefficients ξ_i. If the support of the basis functions is small (i. e., if a is small) then this linear system has an unstructured sparsity which can be exploited by, for example, an iterative row-action solution method [2].

The computational cost of such a discretisation lies mainly in the evaluation of the Abel transform which requires the calculation of a Bessel function. Fortunately, Bessel functions of half-integer order can be calculated efficiently from their recurrence relations (see the Atlas, [12], for details).

The discretisation techniques describe here can also be applied to problems in which the integrals are over curves of sufficient smoothness to allow a local linear approximation.

3 Fourier transform and invertibility

The Fourier transform of the d-dimensional basis function Ψ_m is radially symmetric and given in (A3) of [5] as

$$\hat{\Psi}_m(x) = \frac{a^d \alpha^m (2\pi)^{d/2}}{I_m(\alpha)} \frac{J_{m+d/2}(z)}{z^{m+d/2}}, \quad z = \sqrt{(2\pi a\|x\|)^2 - \alpha^2}. \tag{3.1}$$

The presence of the Bessel function $J_{m+d/2}(z)$ in this expression clearly implies that it is not non-negative, and so by Bochner's characterisation of positive definite translation-invariant functions, Ψ is *not* positive definite for any choices of the parameters.

This fact denies us the attractive approximation theory of the compactly supported radial functions of Wu, Wendland and Buhmann (Section 3, [1]). In particular, there is no guarantee, *per se*, on the invertibility of the interpolation matrix $[\Psi(x_i - x_j)]$, needed to ensure that (2.1) can represent an arbitrary function at its centres. However, this interpolation matrix is invertible if it is strictly diagonally dominant (Corollary 5.6.17, [4]) which, for a set of centres on a uniform grid Γ, holds if

$$\Psi(0) > \sum_{x \in \Gamma \setminus \{0\}} \Psi(x). \tag{3.2}$$

Values of the parameters for which (3.2) is satisfied for the square planar grid $\Delta\mathbf{Z}^2$ are shown in Figure 2.

As is noted in [5], there are several reasons why a rapid decay of the Fourier transform of the basis function is advantageous in functional representation for the inversion of x-ray and related transforms.

- Such inversions may be complicated by functions in the nullspace of the transform, so-called ghosts. For some transforms [7] it can be shown that such functions have a Fourier transform which is small close to zero in the frequency domain, and so representation by a basis with Fourier transform localised around zero will suppress these ghosts.
- These inversions are often *ill-posed* and the given data noisy. Representation of the sought function by a basis with localised Fourier transform imposes smoothness, and so acts to regularise the problem in the sense of Tikhonov.
- It is often convenient to sample the inverted function on a grid which differs from the set of centres x_i of the basis. With a localised Fourier transform, such a sampling can be performed without significant aliasing.

The asymptotic estimate $\hat{\Psi}_m(x) = O(1/\|x\|^{m+(d+1)/2})$ may be derived from (3.1) and estimates I_m with large argument (see Eq. A4, [5]), a fact which should inform our choice of m.

4 Choice of parameters

One agreeable feature of Lewitt's radial functions is that the choice of parameters of the functions correspond in a natural way to the balance between representation quality and efficiency of computation. For example, the asymptotic rate of decay of the Fourier transform increases with m (see above), but so does the cost of the calculation of I_m.

A similar choice arises when the centres lie on a uniform (square or triangular) grid Γ. Let Δ denote the *grid spacing* of such a grid, i. e., the minimum distance between distinct centres in Γ. It is desirable that the *grid ratio* a/Δ be small, as this results in sparsity of the discretisation. As a guide to fixing the values of α and the grid ratio, Lewitt suggests the error in *quasi interpolation to a constant*, the error with which the function

$$g(x) := \sum_{i \in \Gamma} \Psi(x)$$

approximates the function whose constant value is that of g at the centres (edge effects are ignored here). In Figure 2 the root mean square of this representation error (estimated numerically) is shown for the square planar grid, $m = 2$ and a range of values of α and a/Δ. The distinctive "trenches" in the error can be explained with Poisson summation formula (see [11]),

$$\sum_{n \in \mathbf{Z}^2} \Psi(x + \Delta n) = \frac{1}{\Delta^2} \sum_{n \in \mathbf{Z}^2} \exp(2\pi i\, n.x/\Delta) \hat{\Psi}(n/\Delta). \tag{4.1}$$

The summand for $n = 0$ in the second sum is $\hat{\Psi}(0)$, so the representation error depends only on the values of $\hat{\Psi}$ on the *dual grid*, $\mathbf{Z}^2/\Delta$. Provided that $\hat{\Psi}$ decays rapidly, we would expect a small error when $\hat{\Psi}$ is zero, or close to zero, for the dual grid-nodes close to the origin.

By (3.1), $\hat{\Psi}(x)$ is zero exactly when

$$J_{m+d/2}(\sqrt{(2\pi a \|x\|)^2 - \alpha^2}) = 0,$$

i. e., for radial values $\|x\| = R_k$,

$$R_k = \frac{1}{2\pi a} \sqrt{\alpha^2 + \eta_k^2} \qquad (k = 1, 2, \ldots),$$

where η_k is the k-th zero of $J_{m+d/2}$. Thus, the requirement that that the k-th zero of $\hat{\Psi}(x)$ occurs at the radius of the closest non-zero dual grid node (i. e., $R_k = 1/\Delta$) is a constraint on the values of α and a/Δ

$$\alpha = \sqrt{(2\pi a/\Delta)^2 - \eta_k^2}. \tag{4.2}$$

The contours (4.2) agree well with the trenches evident in Figure 2. With the same intent we can require that the l-th zero of $\hat{\Psi}(x)$ occur at the radius of the *second* closest dual grid node ($R_l = \sqrt{2}/\Delta$). Points satisfying *both* of these constraints can be expected to have a particularly small representation error. In Figure 2 these favourable choices are labelled `k:1`.

The above argument can be also be applied the triangular grid. Establishing the Poisson summation formula for such is straightforward (either generalised from VII Section 2 of [11] or specialised from the formula for topological groups in [6]), and one finds that dual grid is the triangular grid with node spacing $2/(\Delta\sqrt{3})$. The representation error is, qualitatively, similar to that shown in Figure 2. To make a quantitative comparison we plot, in Figure 3, the representation error on the principal trench (i. e., along the contour

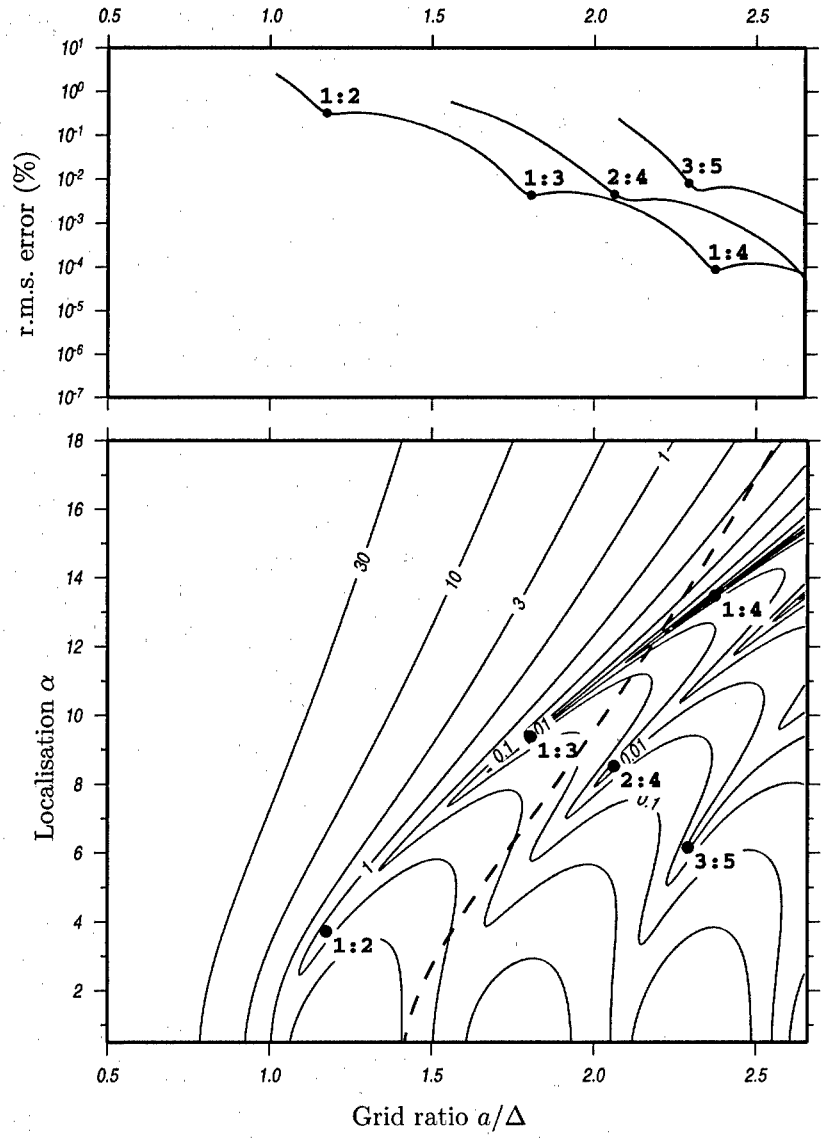


FIG. 2. The representation error of the square planar grid for $m = 2$. The lower contour map shows the root mean square error in representation for different values of the grid ratio a/Δ and localisation α. The upper figure shows the error along the trenches evident in the lower. Favourable choices of the parameters are marked `1:2`, `1:3`,..., and are also shown in the lower figure. Values of the parameters to the left of the dashed line give rise to a diagonally dominant interpolation matrix.

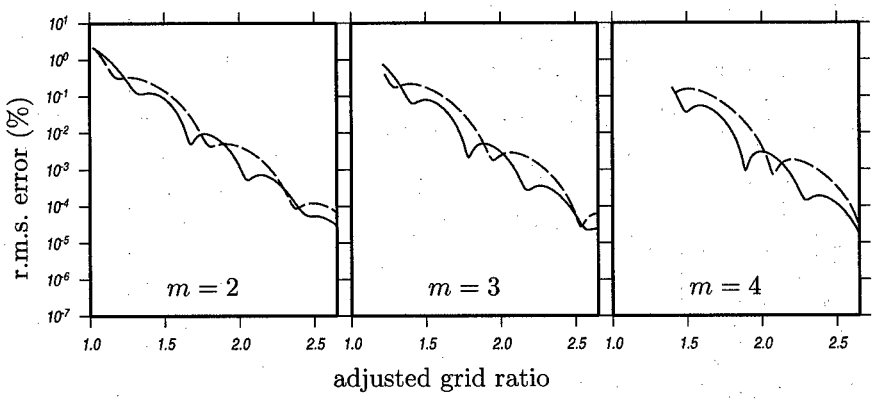


FIG. 3. Error in the principal trench for square (dashed) and triangular (solid) grids.

(4.2) for $k = 1$ in the case of the square grid) for each grid type and a number of values of m.

To ensure a fair comparison, the horizontal scale in Figure 3 is adjusted for each grid-type to give equal node densities. As is seen, the two grid-types have similar error performance, suggesting that the square grid (with attendant ease of implementation) is to be preferred in practice.

5 The functions of Wendland

It is interesting to compare Lewitt's functions with the radial basis functions of Wendland [1, 14], positive definite functions whose window functions are piecewise polynomial. The positive definiteness of Wendland's functions indicate their usefulness in approximation, for which extensive results exist, and a number of recent papers have explored their use in the discretisation of partial differential equations.

The use of Wendland's functions in x-ray problems does not appear to have been investigated, although their Abel transforms can be obtained analytically. We do not address this question here, but indicate why Lewitt's functions *may* offer some advantages for such problems. The Fourier transform of Wendland's function $\Phi_{2,0}$, whose window is $\phi_{2,0}(r) = (1-r)_+^2$, is proportional to

$$r^{-4} \int_0^{2\pi r} (2\pi r - t)^2 t J_0(t)\, dt = O(r^{-3}) \qquad (r = \|x\|)$$

(see Section 3, [14]). In Figure 4, $\hat{\Phi}_{2,0}$ is plotted along with the Fourier transform $\hat{\Psi}_2$, of Lewitt's function with $a = 1$ and the parameter choice 1:2 of Figure 2. Although both have the same *asymptotic* decay of the Fourier transform, Lewitt's is more localised about zero and thus *may* offer better suppression of ghosts in x-ray problems.

Finally we mention that Buhmann has shown, in [1], that Wendland's window func-

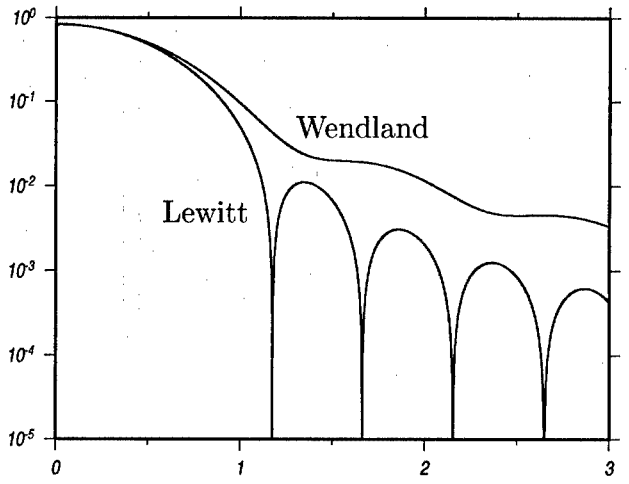


FIG. 4. Fourier transforms of basis functions.

tion admits a convolution representation of the form

$$\rho(r) := \int_0^\infty (1 - r^2/t)_+^n t^n g(t)\, dt \tag{5.1}$$

for the weight $g(t) = (1-t)_+^k$ with suitable k and n. We note that (5.1) may be solved for g, since substituting $x = r^2$ in (5.1) allows it to be reduced to a standard integral equation whose solution,

$$g(x) = \frac{(-1)^n}{n!} f^{(n)}(x), \qquad f(x) = \rho(r),$$

can be found in Article 1.1–4.32 of [10]. In the case that ρ is Lewitt's window ψ_m, one may use the differentiation formula, A11 of [5], to find the corresponding weight g. For $n = 1$ we find that

$$g(x) = -\frac{1}{2\alpha^{m-2}} \frac{I_{m-1}(\alpha)}{I_m(\alpha)} \psi_{m-1}(r),$$

a weight qualitatively different from that of Wendland's function.

Acknowledgements: The author wishes to thank L. R. Wyatt and the referees for a number of helpful comments, and acknowledges the financial support provided by the EC with the grant MAS3-CT98-0168.

Bibliography

1. M. D. Buhmann. Radial basis functions. In *Acta Numerica*, volume 9, pages 1–38. Cambridge University Press, 2000.
2. Y. Censor. Row-action methods for huge and sparse systems and their applications. *SIAM Review*, 23(4):444–466, October 1981.
3. J. J. Green. Discretizing Barrick's equations. Submitted.

4. R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 1990.
5. R. M. Lewitt. Multidimensional digital image representations using generalized Kaiser-Bessel window functions. *J. Opt. Soc. Am. A*, 7(10):1834–1846, October 1990.
6. L. H. Loomis. *An Introduction to Abstract Harmonic Analysis.* D. Van Nostrand, 1953.
7. A. K. Louis. Orthogonal function series expansion and the null space of the Radon transform. *SIAM J. Math. Anal.*, 15(3):621–633, May 1984.
8. S. Matej, G. T. Herman, T. K. Narayan, S. S. Furuie, R. M. Lewitt, and P. E. Kinahan. Evaluation of task-oriented performance of several fully 3d PET reconstruction algorithms. *Phys. Med. Biol*, 39:355–367, 1994.
9. S. Matej and R. M. Lewitt. Practical considerations for 3-d image reconstructions using spherically symmetric volume elements. *IEEE Transactions on Medical Imaging*, 15(1):68–78, 1996.
10. A. D. Polianin and V. Manzhirov. *Handbook of Integral Equations.* CRC Press, 1998.
11. E. M. Stein and G. Weiss. *Fourier Analysis on Euclidean Spaces.* Princetion University Press, 1971.
12. William J. Thompson. *Atlas for computing mathematical functions.* John Wiley & Sons Inc., New York, 1997.
13. G. N. Watson. *A Treatise on the Theory of Bessel Functions.* Cambridge, second edition, 1944.
14. H. Wendland. Error estimates for interpolation by compactly supported radial basis functions of minimal degree. *Journal of Approximation Theory*, 93:258–272, 1998.
15. L. R. Wyatt. A relaxation method for integral inversion applied to HF radar measurement of the ocean wave directional spectrum. *International J. Remote Sensing*, 11:1481–1494, 1990.

Computing with radial basic functions the Beatson-Light way!

Will Light

Department of Mathematics and Computer Science, University of Leicester, UK.
pwl@mcs.le.ac.uk

Abstract

In this paper we discuss a number of recent developments in the practice of how to compute with radial basic functions. The two main problems addressed are how to develop fast evaluation schemes for radial basic functions, and how to efficiently carry out the solution of the interpolation problem. The approach is to mainly describe work which has involved the author and Professor Rick Beatson as contributors, and to include an idiosyncratic selection of works by other researchers which have attracted the attention of the author.

1 Introduction

Research into radial basic functions has been active now for about 30 years. The basic setup is as follows. A function $\psi : \mathbb{R}^n \to \mathbb{R}$, which we refer to as the *basic function*, is specified. A subspace is then constructed by reference to points $x_1, \ldots, x_m$ in $\mathbb{R}^n$. The members of this subspace all have the form

$$s(x) = \sum_{i=1}^{m} a_i \psi(x - x_i), \qquad x \in \mathbb{R}^n,$$

where the $a_1, \ldots, a_m$ are real numbers. It is important to appreciate at the outset that throughout this paper, and indeed in most of the papers appearing in this area, the underlying assumption is that the points $x_1, \ldots, x_m$ are distinct. One of the most common tasks for which these functions are used is interpolation. A small amount of research has been carried out where the points at which an interpolant is developed are arbitrary distinct points in $\mathbb{R}^n$, but by far the majority of the work relates to interpolation which is carried out at the same points as those used to effect the translation. Accordingly, data $d_1, \ldots, d_m$ are given at $x_1, \ldots, x_m$, and we require that

$$d_j = s(x_j) = \sum_{i=1}^{m} a_i \psi(x_j - x_i), \qquad j = 1, \ldots, m. \tag{1.1}$$

Two immediate observations present themselves. Firstly, at the present level of generality there is absolutely no guarantee that the Equations (1.1) will have a unique solution. Secondly, one knows from the work of Mairhuber [14] that there are no Haar subspaces

of significant dimension in any space $\mathbb{R}^n$ for $n \geq 2$. What this means is that if we are to construct interpolation problems which have a unique solution for each location of the data points $x_1, \ldots, x_m$ and for each choice of the data $d_1, \ldots, d_m$, then the subspace used must vary as the interpolation points vary. If we pause for a moment and consider how we might in some sensible and orderly way vary the subspace as the points $x_1, \ldots, x_m$ vary, then using simple shifts of a single basic function ψ is one of the most natural choices. It is very common to work with a function ψ which is a radial function. Thus we take a function $\phi : \mathbb{R}^+ \to \mathbb{R}$ and determine ψ by the rule $\psi(x) = \phi(|x|)$ for all $x \in \mathbb{R}^n$. Note that throughout this account, the symbol $|\cdot|$ will stand for the Euclidean norm in $\mathbb{R}^n$. At this point a common inaccuracy arises. The function ψ can be correctly referred to as a *radial basic function.* However, many authors give this appellation to the function ϕ, whose radiality is of no consequence whatsoever, since it would imply that ϕ was simply an even function on $\mathbb{R}$. Since ϕ only acts on $\mathbb{R}^+$ the idea that ϕ can be radial is vacuous. Let us continue in this spirit of criticism a little while longer. As far as the author is aware, only two people in the world would refer to ψ as a basic function, or a radial basic function. All other authors would use the word *basis* in place of *basic.* There are very obvious problems with this terminology. We are seeking to generate subspaces which are suitable for interpolation. Such subspaces will naturally have the same dimension as the number of data, and the functions $\{\psi(\cdot - x_i) : i = 1, \ldots, m\}$ should form a basis for the subspace. The use of the word basis in two completely different senses seems to the author to be misleading and unhelpful, whereas use of the word basic — a difference of one character — eliminates any possibility of confusion, and avoids the use of the word *basis*, which has a very specific mathematical meaning, in a context where its meaning is not the usual mathematical one.

The problem about whether interpolation is possible has a highly satisfactory answer in the work of Micchelli [15]. We direct the reader to the book of Cheney and Light [10] for a full account of these matters. A couple of examples will be helpful. If one chooses

$$s(x) = \sum_{i=1}^{m} a_i \phi(|x - x_i|) = \sum_{i=1}^{m} a_i \exp(-|x - x_i|^2), \qquad x \in \mathbb{R}^n,$$

or

$$s(x) = \sum_{i=1}^{m} a_i \phi(|x - x_i|) = \sum_{i=1}^{m} a_i |x - x_i|, \qquad x \in \mathbb{R}^n,$$

then the resulting interpolation problem is uniquely solvable for any choice of $x_1, \ldots, x_m$ and for any data $d_1, \ldots, d_m$. This result contrasts very strongly with the case for polynomial interpolation, where the data points $x_1, \ldots, x_m$ have to be constrained not to lie on an algebraic surface of appropriate degree. Indeed, the alternative formulation of the above result for the second example is quite often surprising to mathematicians who are uninitiated in the theory of radial basic functions.

Theorem 1.1 *Let $x_1, \ldots, x_m$ be distinct points in $\mathbb{R}^n$. Then the matrix $(|x_j - x_i|)$ is invertible.*

Having drawn a clear distinction between polynomial approximation and approximation by (radial) basic functions it is at this point that we must consider having some

polynomial ingredients in our interpolant. This is done in a very standard way by a process we call augmentation by polynomials. We consider interpolants of the form

$$s(x) = \sum_{i=1}^{m} a_i \phi(|x - x_i|) + p(x), \qquad (x \in \mathbb{R}^n).$$

Here p is a polynomial of total degree at most $k-1$. We still wish to interpolate to m pieces of information, but now have more than m parameters to determine with this information. The remaining parameters are determined via the 'natural' boundary conditions. The full set of equations is

$$\begin{aligned} d_j &= s(x_j) = \sum_{i=1}^{m} a_i \phi(|x_j - x_i|), \qquad j = 1, \ldots, m \\ 0 &= \sum_{i=1}^{m} a_i q(x_i), \qquad \text{for all } q \in \pi_{k-1}(\mathbb{R}^n). \end{aligned}$$

Here $\pi_{k-1}(\mathbb{R}^n)$ represents the space of polynomials of total degree $k-1$ in $\mathbb{R}^n$. Two questions present themselves pretty quickly from this additional hypothesis. Why should polynomials be added to the interpolant, and why are the boundary conditions chosen in this particular way? In some sense it is essential that we allow ourselves the possibility of adding polynomial terms to some of the interpolants, as we shall soon see. The most important example of a radial basic function interpolant which has a polynomial part will be the thin-plate spline. We will make considerable reference to this interpolant in $\mathbb{R}^2$, where it has the form

$$s(x) = \sum_{i=1}^{m} a_i |x - x_i|^2 \ln |x - x_i| + ax + b, \qquad (x \in \mathbb{R}^2).$$

Note here that the parameter a is a vector with two entries, as is x. Thus ax stands for the dot product between a and x. The parameter b is a real number. The natural boundary conditions take the form

$$\sum_{i=1}^{m} a_i = \sum_{i=1}^{m} a_i s_i = \sum_{i=1}^{m} a_i t_i = 0,$$

where $x_i = (s_i, t_i)$, $i = 1, \ldots, m$. This particular interpolant exhibits a feature common to all the cases where augmentation by polynomials is either necessary or desirable: the degree of the polynomial added is very low. The usual choices are $k = 0$ (when no polynomial term is added), $k = 1$ (when the term is a constant polynomial) and $k = 2$ (when the added polynomial is linear). It is now no longer possible to carry out interpolation for all choices of the points $x_1, \ldots, x_m$. One must avoid distributions of these points which lie on a zero surface of the corresponding polynomial subspace. In the explicit case we considered above (thin-plate splines), the very mild restriction needed is that $x_1, \ldots, x_m$ should not all lie on a single straight line. The theory developed by Micchelli [15] includes the case of augmentation by polynomials.

We now propose to take a look at a very simple example which we hope will give the

reader a feel for some of the ideas and concepts we have introduced so far. We consider

$$s(x) = \sum_{i=1}^{m} a_i |x - x_i| + b \qquad (x \in \mathbb{R}).$$

Here the parameter b is a real number, and the natural boundary condition gives us $\sum_{i=1}^{m} a_i = 0$. A unique feature of the univariate case is that we can order the interpolation points $x_1 < x_2 < \cdots < x_m$. Now consider the function s in one of the intervals $[x_i, x_{i+1}]$, $i = 1, \ldots, m-1$. It is clear that in such an interval s is simply a linear function. The demand that s interpolates the data at $x_1, \ldots, x_m$ means that s must be the piecewise linear interpolant to the data in the interval $[x_1, x_m]$. What is the effect of the 'natural' boundary conditions? In the interval $[x_m, \infty)$ we can write

$$s(x) = \sum_{i=1}^{m} a_i (x - x_i) + b = -\sum_{i=1}^{m} a_i x_i + b.$$

Thus s is constant in $[x_m, \infty)$. A similar calculation reveals that s is constant in $(-\infty, x_1]$. Combining all these observations shows that s is the natural linear spline interpolant to the data at $x_1, \ldots, x_m$. This goes some way to explaining why the word 'natural' is appended the boundary or extra conditions. But we can go a little further. It is well known that the natural splines satisfy a variational principle. For the linear spline, if we examine

$$\mathcal{X} = \{f \in \mathcal{S}' : f' \in L^2(\mathbb{R})\},$$

then

$$\int_{-\infty}^{\infty} (s')^2 \leq \int_{-\infty}^{\infty} (f')^2$$

for all $f \in \mathcal{X}$ which also interpolate the data. This variational principle is very useful in developing error estimates, and we shall return to this general thread of ideas later in this account. However, we ought to observe that $\mathcal{S}'$ is the space of tempered distributions, and that the first derivative is to be taken in the distributional sense. There are ways of getting round this distributional approach (see Cheney and Light [10] for an example which corresponds closely to the discussion here), but it does give the most succinct description, and creates the technical background which will underpin all the theory which has been developed in this area. Notice also that the quantity being minimised can be used to specify a seminorm on $\mathcal{X}$ simply by taking the square root of the integral. This seminorm has as kernel $\pi_0(\mathbb{R})$, which is precisely the polynomial subspace we use to augment the original radial basic function. Something very fundamental is happening here. Most mathematicians would regard this seminorm as being a measure of smoothness of the corresponding function. The natural linear spline therefore interpolates the data, and is the smoothest interpolant to the data from $\mathcal{X}$ in the sense that it possesses the smallest derivative in the L^2-norm. If we are to pursue this very natural idea of making higher derivatives of s small, then we will naturally develop seminorms with polynomial kernels. This goes a long way towards explaining the need for augmentation.

Finally in this introduction, we want to discuss briefly the uses to which radial basic

function interpolation is put. There are two significant feelings about interpolation by these functions. Firstly, it is thought that radial basic function interpolation is very good for treating scattered data. Loosely speaking, data is scattered when there is no possibility of determining either a natural choice of coordinate axes, or an origin. It is at the opposite end of the spectrum to gridded data. In the presence of a cartesian product for the data sites, it is much more efficient to use univariate methods together with tensor product constructions to do the interpolation. Secondly, radial basic function interpolation is thought to be very good for dealing with high dimensional data. There is some evidence from the realm of neural networks that this is indeed the case, but we will not venture into the area of high dimensional data interpolation in this paper. Finally, many of the data sets we want to treat have very large numbers of data sites and so our aim is to develop methods which will handle 10,000 to 1,000,000 data sites or more.

2 Computational difficulties and fast evaluation

In this Section, we want to discuss the difficulties that arise when a large radial basic function interpolation problem is posed. We shall also deal with one of the essential tools for overcoming some of the difficulties. The system we want to solve has the form

$$d_j = s(x_j) = \sum_{i=1}^{m} a_i \phi(|x_j - x_i|) + p(x_j) \qquad (j = 1, \ldots, m) \tag{2.1}$$

$$0 = \sum_{i=1}^{m} a_i q(x_i), \qquad \text{for all } q \in \pi_{k-1}(\mathbb{R}^n). \tag{2.2}$$

If we declare a basis for $\pi_{k-1}(\mathbb{R}^n)$ then we can write these equations in matrix form as

$$\begin{pmatrix} A & Q \\ Q^T & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} d \\ 0 \end{pmatrix}.$$

Here the matrix A has entries $\phi(|x_j - x_i|)$ and is $m \times m$. The matrix Q has entries $p_\ell(x_j)$, where $p_1, \ldots, p_\nu$ is a basis for $\pi_{k-1}(\mathbb{R}^n)$, and is of size $m \times \nu$. Recall from our assumptions that only low degree polynomials are used, and so Q is a long thin matrix. In the case of thin-plate splines in $\mathbb{R}^2$ it would have size $m \times 3$. However, A is a very large matrix, with absolutely no sparsity. In fact, for thin-plate splines, the matrix A is zero on the diagonal, and has large off-diagonal entries. In solving a large system of linear equations, the only effective strategy is to use an iterative solver. Such a solver will involve many multiplications of the matrix A with a vector a, and the full nature of A makes this a very costly process. One of the key discoveries in this area was the Beatson and Newsam [8] result which showed how fast multipole algorithms could be applied to this area. If we consider the expression

$$s(x_j) = \sum_{i=1}^{m} a_i |x_j - x_i|^2 \ln(|x_j - x_i|) + p(x_j)$$

for some $x_j \in \mathbb{R}^n$, then this can be considered as an evaluation of the function s at the point x_j, or the formation of an element in the matrix vector product Aa. Because of

this, most authors tend only to consider how to *evaluate* the function s in an efficient way — generating what are known as fast evaluation algorithms. It is impossible to estimate properly the importance of this discovery. Anyone involved in programming iterative solutions to the thin-plate spline equations with tens of thousands of points would find that any such algorithm would just grind itself into the dust without this technology. The technology really has two aspects: a mathematical tool, and a programming structure. Here we intend to give only the flavour of the argument. The reader who really wants to know the details is advised to look either at the original paper [8], or the later paper of Beatson and Light [5] which deals with polyharmonic splines. She can also look at two papers which give clear explanations of simple cases. The first is found in a survey paper by Beatson and Greengard [3]. The second is a technical report by Beatson, Levesley and Light [7]. This last paper discusses fast evaluation methods on the circle and higher dimensional spheres, and the reader will find a very careful and full account of the one-dimensional circle case. The first trick with problems in $\mathbb{R}^2$ is to consider complex variables, rather than points in $\mathbb{R}^2$. Let z be a point at which we wish to evaluate s, and u a data point, or *centre*. Then

$$|z-u|^2 \ln|z-u| = \mathcal{RE}(|z-u|^2 \ln(z-u)) = \mathcal{RE}(|z-u|^2 \ln z) + \mathcal{RE}\Big(|z-u|^2 \ln\Big(1-\frac{u}{z}\Big)\Big).$$

Look at the last two expressions here. The first of them has the centre u in the square of the modulus term, and this expression is quite cheap to evaluate, even if there are many centres u. The effect of many centres on the second term is however quite profound, and it is with this term that we must work. The idea is to set a tolerance, and only aim to evaluate s to within this tolerance, rather than exactly. The appropriate series expansion can then be used:

$$|z-u|^2 \ln\Big(1-\frac{u}{z}\Big) = \sum_{p=1}^{\infty} e_p \Big(\frac{u}{z}\Big)^p \approx \sum_{p=1}^{N} e_p \Big(\frac{u}{z}\Big)^p = \sum_{p=1}^{N} f_p(u) z^{-p}.$$

The value of N depends on the tolerance demanded of the evaluation and the relative sizes of u and z. For this reason, we think of z as far away from the origin in $\mathbb{R}^2$, and u close to the origin. If there are now many centres $u_1, \ldots, u_m$ near the origin, and z is far away from the origin, then we can *summarise* the effects of linear combinations of all these centres as follows:

$$\begin{aligned} \sum_{i=1}^{m} a_i |z-u_i|^2 \ln\Big(1-\frac{u_i}{z}\Big) &\approx \sum_{i=1}^{m} a_i \sum_{p=1}^{N} f_p(u_i) z^{-p} \\ &= \sum_{p=1}^{N} \sum_{i=1}^{m} a_i f_p(u_i) z^{-p} = \sum_{p=1}^{N} g_p(u_1, \ldots, u_m) z^{-p}. \end{aligned}$$

The principle now is to use the last expression above to make an approximate evaluation of s. Of course, the assumption that z was far from the origin and $u_1, \ldots, u_m$ were close to the origin is not important. It is simply important that z be far away from the cluster of centres $u_1, \ldots, u_m$. The summarising expression is referred to as a Laurent type expansion, because it summarises the contribution of the centres $u_1, \ldots, u_m$ in terms of

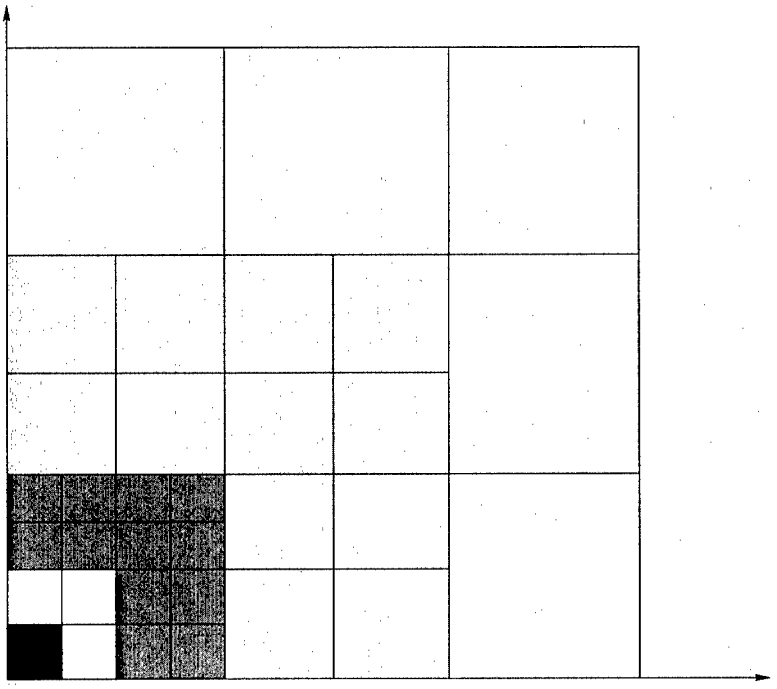


FIG. 1. Fast evaluation panelling.

series involving negative powers of z. There is now a lot of preprocessing to go on before the fast evaluation algorithm is ready to roll. Figure 1 shows how the algorithm proceeds. The shaded square at the bottom left of the domain is the point which contains z, the evaluation point. All the squares around this one which are the same size are deemed to be 'close' to the evaluation square. All other squares are 'far away'. Of course, as the squares get further away from z it becomes possible to use our summarising technique to total up the contributions of larger and larger numbers of points. This is done in a very explicit manner, which is represented by the shading in Figure 1. As we get further away, we double the size of the squares over which we summarise, and there is a band of same-size squares (or a ring, if the evaluation square was in the middle of the domain) two squares wide surrounding the evaluation square. Once all the preprocessing is done, and we shall discuss this a little more in a moment, all the needed coefficients g_p are available, and evaluation can be carried out in about $\mathcal{O}(\log m)$ FLOPS instead of $\mathcal{O}(m)$.

The above account does not quite reveal the whole story. The coefficients g_p are calculated in an orderly manner which greatly improves the efficiency of the algorithm. Suppose our problem is located in $[0,1]^2$. An initial decision is made to divide the original domain into squares of size 2^{-n}. There is then a parent child relationship derived through a quad-tree data structure. The parent $[0,1]^2$ has four children: $[0,0.5]^2$, $[0.5,1]^2$, $[0,0.5] \times [0.5,1]$ and $[0.5,1] \times [0,0.5]$. This parent-child relationship helps in setting up the coefficients $g_p(u_1,\ldots,u_m)$ in an efficient way. There is also a further idea involving

Taylor series, which gives more efficiency. We omit any description of this technique.

3 Inverting the interpolation matrix

Recall as at the beginning of the previous section that the equations specifiying the interpolation problem are as follows:

$$d_j = s(x_j) = \sum_{i=1}^{m} a_i \phi(|x_j - x_i|) + p(x_j), \qquad (j = 1, \ldots, m) \tag{3.1}$$

$$0 = \sum_{i=1}^{m} a_i q(x_i), \qquad \text{for all } q \in \pi_{k-1}(\mathbb{R}^n). \tag{3.2}$$

In matrix terms we have

$$\begin{pmatrix} A & Q \\ Q^T & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} d \\ 0 \end{pmatrix},$$

where A is a full matrix which tends to exhibit poor conditioning. The poor conditioning of A is similar to problems experienced by researchers in the theory of finite elements — as the interpolation points become very dense in a given region, the conditioning gets worse. In fact, there are formal statements relating some impression of the condition number of A (usually the smallest eigenvalue of A) to the minimum interpoint distance. The following table represents the condition number of A when the interpolation points are given on a uniform 5×5 grid in $[0, \alpha]^2$. Of course, on a philosophical level, it does not

Scale parameter α	Condition Number
1.0	3.6458×10^2
0.1	2.5179×10^4
0.01	2.4364×10^6
0.001	2.4349×10^8

TAB. 1 Two norm condition numbers of A.

make any sense whatsoever to describe an interpolation problem as being ill-conditioned. Let's discuss this point in a little more depth. Suppose $x_1, \ldots, x_m$ are points in $\mathbb{R}^n$, and $G_1, \ldots, G_m$ are a set of functions from $\mathbb{R}^n$ to $\mathbb{R}$ which are linearly independent over $\{x_1, \ldots, x_m\}$. That is, interpolation to arbitrary data at $x_1, \ldots, x_m$ by linear combinations of $G_1, \ldots, G_m$ is always uniquely possible. Then there is a basis $F_1, \ldots, F_m$ for the linear span of $G_1, \ldots, G_m$ such that $F_i(x_j)$ is 1 if $i = j$ and is zero for all other values of i, j between 1 and m. If the given data is $d_1, \ldots, d_m$, then the interpolant can be written down immediately as

$$\sum_{i=1}^{m} d_i F_i(x) \qquad (x \in \mathbb{R}^n).$$

If one has in one's hands the basis $\{F_1, \ldots, F_m\}$ and wants to know the coefficients which must be used then one need only invert the identity matrix to obtain the solution, and there are not many matrices which are better conditioned than the identity matrix! Of course, getting one's hands on the basis $F_1, \ldots, F_m$ is usually rather difficult — as hard as solving the original problem in fact. It has become traditional to refer to the basis $F_1, \ldots, F_m$ as the Lagrange basis (in sympathy with the fact that Lagrange was a person who wrote down this basis for polynomial interpolation in one dimension) or the cardinal basis. This last term seems to the author to be quite appropriate, indicating that the basis is special. However, it does not find favour with spline theorists, since they think of the word cardinal in a very technical sense (the interpolation points are $\mathbb{Z}^n$). Terminology aside, the point is still made that the conditioning of any interpolation problem is a function of the available basis. A more practical case of this phenomenon is the problem of natural cubic splines in $\mathbb{R}$. They fit into the radial basic function interpolation scenario, because a natural cubic spline with knots at $x_1, \ldots, x_m$ can be written as

$$s(x) = \sum_{i=1}^{m} a_i |x - x_i|^3 + ax + b \qquad (x \in \mathbb{R}).$$

If we require this spline to interpolate data $d_1, \ldots, d_m$ at $x_1, \ldots, x_m$ then we have to require that $s(x_j) = d_j$ for $j = 1, \ldots, m$. The natural property comes, as expected, from the natural boundary conditions:

$$\sum_{i=1}^{m} a_i = \sum_{i=1}^{m} a_i x_i = 0.$$

The ill-conditioning illustrated in Table 1 would be equally present in this example, and the remark that the conditioning increases as the interpoint spacing decreases would also hold good. Of course, to suggest the use of this basis to a spline practitioner would not be a good idea! We are well used to the idea that B-splines are the correct basis to use in this situation.

I suppose the two principles to emerge from the above discussion are that the basis we have used thus far to describe the interpolation problem is not satisfactory from a computational point of view, and that in at least some of the cases under discussion (all of them one-dimensional) there are other bases which are superior. There are other ways to conceptualise the difficulties we experience with the radial basic functions. Most of them tend to grow at infinity, and have small value at zero. As a general principle, we would like a basis to mimic the B-spline basis. That is, we would like the basis to be local if possible — each basis function having a fairly small support around one of the interpolation points. The first people to make progress in this area were Dyn and Levin [11] in 1983. There is a later paper with Rippa [12] in 1986 which is also worth looking at. Their technique was based on the observation that if $F(x) = |x|^2 \ln |x|$, and $x \in \mathbb{R}^2$, then $\nabla^4 F = 8\pi\delta$. Here, ∇^4 represents the bilaplacian, and δ is the Dirac delta distribution whose action on each rapidly decreasing function in $\mathcal{S}$ is to evaluate it at zero. This description alone should alert us to the fact that $\nabla^4 F = 8\pi\delta$ is a distributional equation, and as such must be handled with care. However, numerical analysts dash in

where others fear to tread, and we can approximate the Laplacian as follows:

$$(\nabla^2 F)(x) \approx h^{-2}\{F(x-he_1)+F(x+he_1)+F(x-he_2)+F(x+he_2)-4F(x)\} \qquad (x \in \mathbb{R}^2).$$

Here h is a real parameter, and e_1 and e_2 are the usual unit vectors in $\mathbb{R}^2$. Pictorially, we can represent this approximation by the stencil shown in Figure 2. The bilaplacian stencil

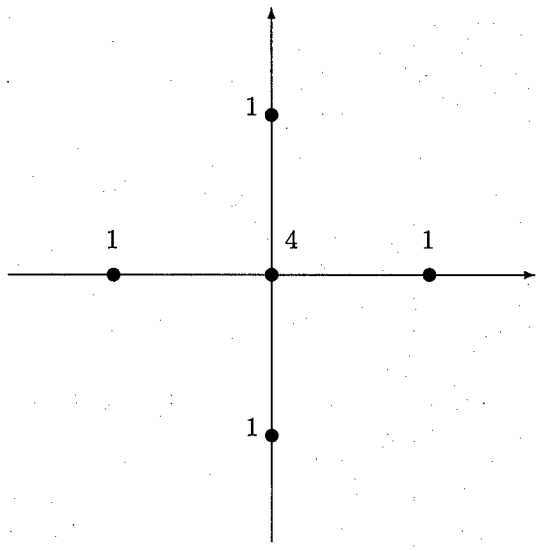


FIG. 2. The stencil for the Laplacian.

is shown in Figure 3. This observation is used in a straightforward way if the interpolation points lie on a grid. Instead of using the thin-plate spline radial basic function to generate a basis, one uses the appropriate linear combinations which represent the bilaplacian of this function. Because one has a distributional equation relating this quantity to the δ function, one does not expect to get the δ function exactly, but one certainly does expect to get a function which decays rapidly at ∞, and this is exactly what happens. Dyn and Levin provide some encouraging numerical results. Of course, there remains the problem of what to do when the data is not gridded. Here one must develop first the appropriate stencil for the Laplacian on a point by point basis. This may seem laborious, but in fact the next few methods we will describe all compute better basis elements on a point by point basis.

Perhaps the most successful class of schemes of this nature — computing a new basis on a point by point approach — comes from Beatson, Goodsell and Powell [2] and Beatson, Cherie and Mouat [1]. Their approach is perhaps simpler to appreciate and implement than that of Dyn and Levin. They begin with the observations I made earlier — what we are really after is the cardinal basis $F_1, \ldots, F_m$ with the property that $F_i(x_j)$ is 1 if $i = j$ and is zero for all other values of i, j between 1 and m. However, because this problem is as difficult to solve as the original one, we proceed as follows. Consider

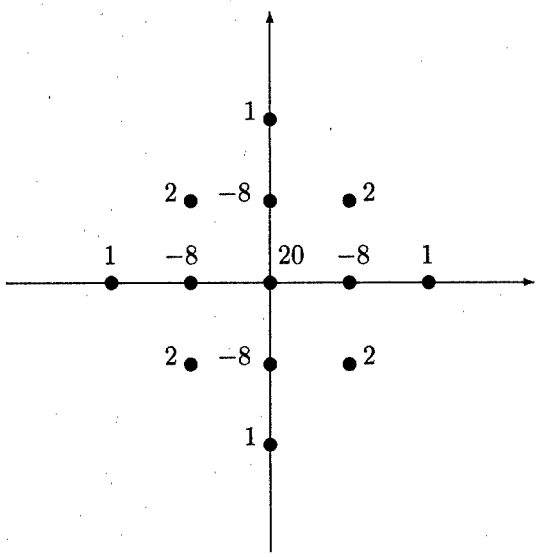


FIG. 3. The stencil for the bilaplacian.

the job of trying to construct F_i. This function is supposed to be 1 at x_i and zero at all other points. Choose about 50 near neighbours of x_i, say $y_j \in \{x_1, \ldots, x_m\}$. This choice must include x_i. Then take

$$F_i(x) = \sum_{j=1}^{50} a_j |x - y_j|^2 \ln |x - y_j| + bx + c, \qquad (x \in \mathbb{R}^2).$$

We demand that

$$F_i(y_j) = \begin{cases} 1 & \text{if } y_j = x_i, \\ 0 & \text{otherwise,} \end{cases}$$

and that the natural boundary conditions are also satisfied. Thus we are producing *approximate cardinal functions* which have the value 1 at the required point, but are only zero on about 50 neighbouring points. This suggestion is based on the fact, observed by many workers, that such functions are often small elsewhere in the domain. We produce some pictures to illustrate this. In the first (Figure 4), 289 points are spaced on a regular grid in $[0, 1]^2$. The approximate cardinal function is based on the 13 points shown in bold in Figure 4. Figure 5 illustrates the same situation, but now as shown the points used to develop the cardinal function are all clustered in one corner of the domain. The effect is to produce significant values at the opposite corner of the domain. One can infer from this that whenever the data is pretty much uniformly distributed, the cardinal functions using points well inside the domain will have good properties, while those at the edge will be poor. Similarly, in a non-uniform distribution, those interior to a cloud of points will behave well, while those at cloud boundaries might not.

There are two methods for dealing with the difficulties which have shown up above.

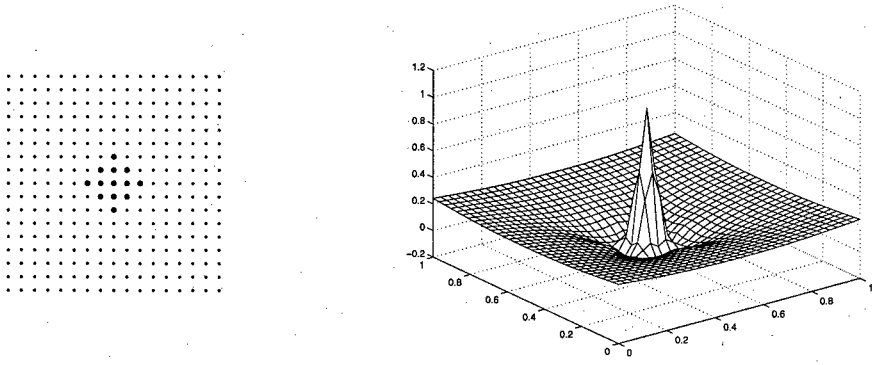


FIG. 4. Approximate cardinal function with points central to the domain.

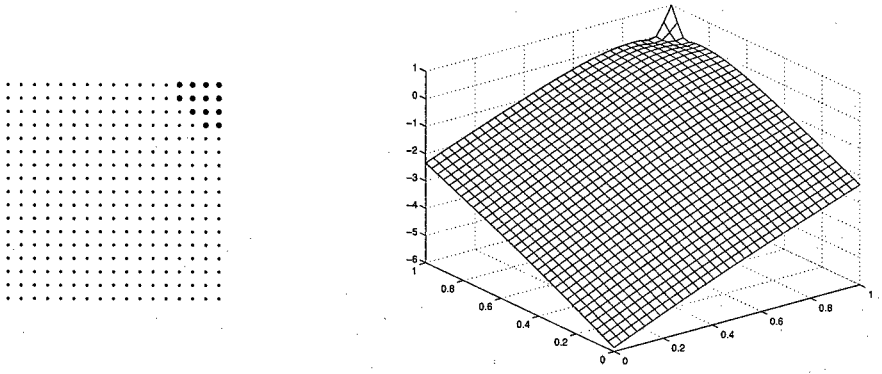


FIG. 5. Approximate cardinal function with points at one corner of the domain.

Firstly, one can *pin* all the cardinal functions at a fixed set of judiciously chosen points — so that every cardinal function must have the value zero at these points. This is very effective in the case of regularly spaced data as Figures 6 and 7 show. One can imagine however, that a data set with a number of clouds might benefit from a judicious choice of points at which to carry out the pinning. What one would really like is a method which does not rely on any user intelligence in the choice of points. As mentioned before, a desirable feature of a good basis function is one which decays at infinity. This decay should be at some rate if possible. The Beatson, Cherie and Mouat prescription for thin-plate splines in $\mathbb{R}^2$ is that the elements should decay like $|x|^{-3}$ as $|x| \to \infty$. There is a problem here, in that if we opt for decay elements everywhere, then we will not obtain a basis for our space. To get around this problem, we accept an element F_i as a decay element if it satisfies

$$\sum_{i=1}^{50} |F_i(y_j) - \delta_{ij}| < \mu$$

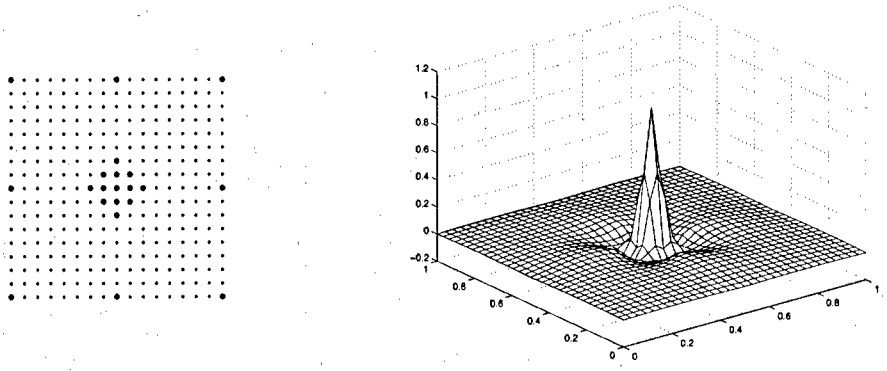


FIG. 6. Approximate pinned cardinal function with points central to the domain.

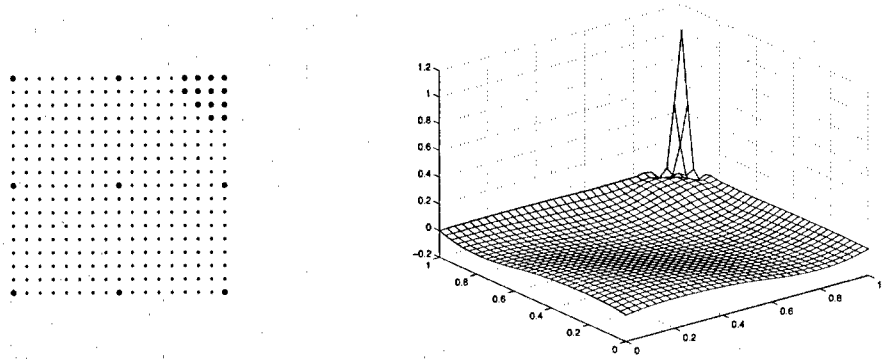


FIG. 7. Approximate pinned cardinal function with points at one corner of the domain.

and

$$|F_i(x)| = \mathcal{O}(|x|^{-3}) \qquad \text{as } |x| \to \infty.$$

Otherwise, we use the F_i which is defined by the previous conditions of cardinality. Again there are a few bells and whistles needed to make this method operate efficiently, but we hope that sufficient detail is present for the reader to be able to see the general idea. All the above methods are providing ways of constructing a better conditioned basis with which to solve the problem. A method still has to be selected to invert the matrix associated with the new basis, which is now much better conditioned than the original matrix corresponding to the conventional basis. The method of choice for most authors is some version of GMRES.

Beatson called the points at which decay could be obtained 'good' points, and points at which decay could not be obtained 'bad' points. This idea has been built on in a recent technical report by Beatson and Levesley [4]. The general spirit is to define good and bad points in the same way as Beatson, and then to develop an iterative solver, solving first on the good, then the bad, then returning to the good and so on.

Finally, a very successful method has recently emerged from the researches of Beatson, Light and Billings [6]. This method has the advantage that it is a fast iterative solver which may be regarded as a preconditioner in its own right (thus it may be combined with a solver such as GMRES). We will describe it here as a solver. It is essentially the domain decomposition method, although as with previous solvers, our description will be very much at a 'bare bones' level, and the interested reader is referred to [6] for the fine details, which include some error estimates, some interesting comments on an alternative basis, and a good deal of theory. We shall describe the method as applied to data on the unit square $[0,1]^2$ in $\mathbb{R}^2$, and we will not make any attempt to make the method adaptive in character. The reader will be able to see these improvements for herself. We will test our method on randomly chosen data in $[0,1]^2$.

We begin with a set of nodes $X = \{x_1, \ldots, x_m\}$ at which interpolation is to be carried out. We will describe the algorithm as it is implemented for solving the thin-plate spline interpolation problem on the node set X. We divide up the square $[0,1]^2$ into a fairly large number of sub-domains $X_1, \ldots, X_\ell$. There are two constraints on these subdomains. It is important that they are constructed so that about equal numbers of points lie in each subdomain — about 50 points per subdomain is ideal. Secondly, it is essential that each subdomain overlap all surrounding subdomains. In our terminology, two subdomains overlap if they have a (small) number of points in common. In each subdomain there are some points in X which lie only in that subdomain and not in any other. We call these points the *inner* points of the subdomain. A *coarse* set Y of inner points in the node set X is also chosen. We will say more about this coarse set in a moment, but at this stage it simply consists of a small number of inner points from each subdomain. The algorithm will then construct the interpolant s and proceeds as follows. We initialise the interpolant s as $s = 0$. We want to solve the equations

$$d_j = s(x_j) = \sum_{i=1}^{m} a_i |x_j - x_i|^2 \ln |x_j - x_i| + \alpha x_j + \beta, \qquad (j = 1, \ldots, m) \tag{3.3}$$

subject to the boundary conditions

$$\sum_{i=1}^{m} a_i = \sum_{i=1}^{m} a_i s_i = \sum_{i=1}^{m} a_i t_i = 0, \tag{3.4}$$

where $x_i = (s_i, t_i)$. In matrix form these equations are

$$\begin{pmatrix} A & Q \\ Q^T & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} d \\ 0 \end{pmatrix},$$

as we have already seen. Our method will operate by residual correction, so we begin by setting

$$r = \begin{pmatrix} d \\ 0 \end{pmatrix}.$$

It is important to recall that α is a vector of length 2, which we write as $\alpha = (\alpha_1, \alpha_2)$. Suppose now we have begun our iterative procedure and generated an approximation s with a residual r. The next few steps describe how to update the approximation and the

residual.

Step 1. We construct $s_1, \ldots, s_\ell$ such that each s_j is an interpolant based only on all points of the subdomain X_j, using as data the residual vector r restricted to X_j.

Step 2. For each inner point x we now have a single real number a_x which is the coefficient of $|\cdot - x|^2 \ln|\cdot - x|$. If we look at the collection of coefficients belonging to all the inner points of all domains, then this collection is not in general orthogonal to π_1. That is, they fail to satisfy boundary conditions of the type given in Equation (3.4). We now correct so that the collection of coefficients corresponding to all inner points of all domains is orthogonal to π_1.

Step 3. We set

$$\mathcal{S}_1 = \sum \{a_x |\cdot - x|^2 \ln|\cdot - x| : x \text{ is an inner point}\}. \tag{3.5}$$

Step 4. We evaluate the residual $\mathcal{R} = r - \mathcal{S}_1$ at the coarse grid points, and then construct the interpolant $\mathcal{S}_2$ to this residual on the coarse grid points Y.

Step 5. We update s by $s \leftarrow s + \mathcal{S}_1 + \mathcal{S}_2$. The new residual is then given by

$$r = \begin{pmatrix} z \\ 0 \end{pmatrix},$$

where

$$z_i = d_i - s(x_i), \qquad i = 1, \ldots, m.$$

This iterative process can either be continued to convergence, or used as a preconditioner followed by GMRES. Table 2 shows some run times taken to obtain an error of less than 1×10^{-6} for the Franke 1 function (see [13] for the definition of this function). Random nodes were generated in $[0,1]^2$ and an Intel Celeron PC was used. Recently,

Number of nodes	Number of iterations	Time (seconds)
10,000	8	7.0
20,000	8	17.5
40,000	6	35.5
80,000	6	105.7
160,000	7	407.8

TAB. 2 Run times for domain decomposition.

the group at Leicester, using a twin processor Compaq PC, has obtained solutions to a problem with 1,000,000 random points in less than 9 minutes, and we can safely say that the combination of domain decomposition methods and multipole fast evaluation has produced a robust and effective method. Most practitioners will be aware of other ways to run a domain decomposition algorithm. In particular, one can use a nesting approach where one starts with only four subdomains each containing large numbers of points. To solve each subdomain problem, one subdivides again and does domain decomposition in the subdomain.

Bibliography

1. Beatson, R.K., J.B. Cherie and C.T. Mouat, *Fast fitting of radial basis functions: methods based on preconditioned GMRES iteration.* Advances in Computational Mathematics, **11** (1999), 253–270.
2. Beatson, R.K., G. Goodsell and M.J.D. Powell, *On multigrid techniques for thin plate spline interpolation in two dimensions*, Lectures in Applied Mathematics **32** (1996), 77–97.
3. Beatson, R.K. and L. Greengard, *A short course on fast multipole methods*, in *Wavelets, multilevel methods and elliptic PDEs*, Ainsworth, M., J. Levesley, W.A. Light and M. Marletta (eds), Oxford University Press, Oxford (1997), 1–38.
4. Beatson, R.K. and J. Levesley, *Good point/bad point iterations for solving the thin-plate spline interpolation equations*, University of Leicester Technical Report, 2001/34 (2001).
5. Beatson, R.K. and W.A. Light, *Fast evaluation of radial basis functions: Methods for two-dimensional polyharmonic splines*, IMA Journal of Numerical Analysis **17** (1997), 343–372.
6. Beatson, R.K., W.A. Light and S. Billings, *Domain decomposition methods for solution of the radial basis function interpolation problem*, SIAM Journal Scient. Stat. Comp. **22**(5) (2001), 1717–1740.
7. Beatson, R.K., J. Levesley and W.A. Light, *Fast evaluation of radial basic functions on spheres*, preprint.
8. Beatson, R.K. and G.N. Newsam, *Fast evaluation of radial basis functions I*, Computers and Mathematics with Applications, **24** (12) (1992), 7–19.
9. Beatson, R.K. and M.J.D. Powell, *An iterative method for thin plate spline interpolation that employs approximations to the Lagrange functions*, in *Numerical Analysis 1993*, D.F. Griffiths and G.A. Watson (eds), Longmans, Harlow, 1994.
10. Cheney, E.W. and W.A. Light, *A course in approximation theory*, Brooks Cole, Pacific Grove Ca, 1999.
11. Dyn, N. and D. Levin, *Iterative solution of systems originating from integral equations and surface interpolation*, SIAM J. Numer. Anal. **20** (1983), 377–390.
12. Dyn, N., D. Levin and S. Rippa, *Numerical procedures for surface fitting of scattered data by radial functions*, SIAM Journal Scient. Comp. **7**
13. Franke, R., *Scattered data interpolation: Tests of some methods*, Mathematics of Computation **38** (1982), 181–200.
14. Mairhuber, J.C., *On Haar's theorem concerning Chebychev approximation problems having unique solution*, Proc. Amer. Math. Soc. **7** (1956), 609–615.
15. Micchelli, C.M., *Interpolation of scattered data: distance matrices and conditionally positive definite functions,* Constr. Approx. **2** (1986), 11–22.
16. Sibson, R. and G. Stone, *Computation of thin-plate splines*, SIAM Journal on Scient. Stat. Comp. **12** (1991), 1304–1313.

Application of orthogonalisation procedures for Gaussian radial basis functions and Chebyshev polynomials

John C Mason and Andrew Crampton

School of Computing and Mathematics, University of Huddersfield, Huddersfield, UK.
j.c.mason@hud.ac.uk, a.crampton@hud.ac.uk

Abstract

Procedures for orthogonalisation of Gaussians and B-splines are recalled and it is shown that, provided Gaussians are negligible in appropriate regions, the same recurrence formulae may be adopted in both and render the computation relatively efficient. Chebyshev polynomial collocation is well known to be rapidly defined by discrete orthogonalisation, and similar ideas are commonly applicable to partial differentieal equations (PDEs) and integral equations (IEs). However, it is shown that the most elementary mixed methods (both boundary conditions and PDEs being satisfied) for the Dirichlet problem in rectangular types of domain can lead to a singular linear system, which may be rendered non-singular, for example, by a small modification of interpolation nodes.

1 Introduction

Gaussian radial basis functions (RBFs) are negligible outside a certain range, which depends on the accuracy required and the exponent used. For example, if four decimal place accuracy is sufficient, then outside $[-2, 2]$ the function $e^{-\lambda x^2}$ is negligible for $\lambda \geq 2.5$. Indeed the translated RBFs

$$\phi_i(x) = e^{-\lambda(x-i)^2} \quad i = -1, 0, \ldots, n+1, \tag{1.1}$$

resemble, at least superficially, a set of translated cubic B-splines, each having a support of four sub-intervals of length one, contained in $[i-2, i+2]$.

Following work of Mason *et al* [4] and Goodman *et al* [1], we show that these RBFs, rounded to the required accuracy, may be conveniently and efficiently orthogonalised so that

(i) a 4 term recurrence may be adopted identical to the one in [4] for cubic B-splines,

(ii) inner products may be determined very simply in terms of 4 parts of a normal distribution,

(iii) a well conditioned calculation results and best l_2 approximations may be obtained immediately with an orthogonalised basis,

(iv) a continuous or discrete inner product (and best approximation) may be adopted.

In a second application of orthogonalisation, this time to polynomials, it is shown that a two-dimensional $(n+1)\times(n+1)$ polynomial collocation problem, which includes amongst its nodes n Chebyshev polynomial zeros on each of 4 sides of a square, leads to a singular (rank one deficient) system. For all n, one superfluous equation is readily identified and a suitable replacement equation is readily found. Discrete orthogonalisation is used to combine and greatly simplify the equations and prove singularity.

2 Orthogonalised Gaussians

An orthogonal system $\{P_i\}$ is developed from the Gaussians ϕ_i in (1.1) using

$$P_k = \phi_k - a_{k1}P_{k-1} - a_{k2}P_{k-2} - a_{k3}P_{k-3}, \quad k = -1,\ldots,n+1, \tag{2.1}$$

where $a_{13} = a_{03} = a_{02} = a_{-1,3} = a_{-1,2} = a_{-1,1} = 0$.

Now we define coefficients b_{kr}, for $r = 0,\ldots,k+1$ and $k = -1,\ldots,n+1$, as the inner products

$$b_{kr} = \langle \phi_k, \phi_{k-r}\rangle = \int_{I_{k,r}} \phi_k(x)\phi_{k-r}(x)dx, \tag{2.2}$$

where $I_{k,r}$ is the common support of ϕ_k and ϕ_{k-r} and normalising constants n_k are the squared norms

$$n_k = \|P_k\|^2 = \langle P_k, P_k\rangle, \tag{2.3}$$

where $\langle \bullet, \bullet\rangle$ is the inner product (2.2) and $\| \bullet \|$ is the corresponding norm.

Then, setting $\langle P_k, P_{k-r}\rangle = 0$ for $r = 1,2,3$ gives

$$\langle \phi_k, P_{k-r}\rangle = a_{kr}n_{k-r}. \tag{2.4}$$

Taking the inner product of (2.1) with itself gives

$$n_k = b_{k0} + \sum_{r=1}^{3} \left[-2a_{kr} \langle \phi_k, P_{k-r} \rangle + a_{kr}^2 n_{k-r} \right], \tag{2.5}$$

which, by using (2.4), gives

$$n_k = b_{k0} - \sum_{r=1}^{3} a_{kr}^2 n_{k-r}. \tag{2.6}$$

This is the first basic equation for writing $\{n_k\}$ in terms of $\{a_{kr}\}$ and $\{b_{kr}\}$.

Now, using (2.1), with k replaced by $k-1, k-2, k-3$ we obtain

$$\begin{aligned} \langle \phi_k, P_{k-3} \rangle &= b_{k3} = a_{k3} n_{k-3} \\ \langle \phi_k, P_{k-2} \rangle &= b_{k2} - a_{k-2,1} \langle \phi_k, P_{k-3} \rangle . \end{aligned} \tag{2.7}$$

Hence

$$a_{k2} n_{k-2} = b_{k2} - a_{k-2,1} b_{k3}. \tag{2.8}$$

Finally

$$\langle \phi_k, P_{k-1} \rangle = b_{k1} - a_{k-1,1} \langle \phi_k, P_{k-2} \rangle - a_{k-1,2} \langle \phi_k, P_{k-3} \rangle ,$$

so that

$$a_{k1} n_{k-1} = b_{k1} - a_{k-1,1} (a_{k2} n_{k-2}) - a_{k-1,2} b_{k3}. \tag{2.9}$$

Equations (2.6), (2.7), (2.8) and (2.9) may be solved to determine all the required coefficients $\{a_{kr}\}$ and $\{n_k\}$ explicitly by substitution, starting from $n_{-1} = \|\phi_{-1}\|^2$. This involves $\mathcal{O}(n)$ operations for $n+3$ basis functions. The best approximation to a function f (either continuous $f = f(x)$ or discrete $f = (f_1, \ldots, f_m)^T$) by orthogonalised Gaussians may be determined explicitly as

$$f \approx \sum_{j=-1}^{n+1} c_j P_j,$$

where $c_j = \langle P_j, P_j \rangle^{-1} \langle f, P_j \rangle = (n_j)^{-1} \langle f, P_j \rangle .$

2.1 Numerical example

Here we use the procedure for constructing orthogonalised Gaussians to produce an interpolant to data obtained from a fast response oscilloscope[1]. To the left of Figure 1 we see the first three orthogonalised Gaussian functions, with centres specified at the integers -1, 0 and 1, with support growing from left to right. The figure on the right shows the oscilloscope data ** and the fitted o-Gaussian interpolant —.

[1]Oscilloscope data supplied by Centre for Electromagnetic and Time Metrology, National Physical Laboratory, London, UK.

In this example we use 512 centres and choose $\lambda = 2.5$ in (1.1). Since our choice for λ requires only four decimal place accuracy, the normal equations produce the usual identity matrix and the coefficient vector $\{c_{-1}, \ldots, c_{n+1}\}$ can then be determined by the equations $c = A^T f$ where $f = \{f_1, \ldots, f_m\}$ and $A_{i,j} = P_j(x_i)$. The fit is extremely good and vindicates the neglecting of the Gaussians outside the interval considered.

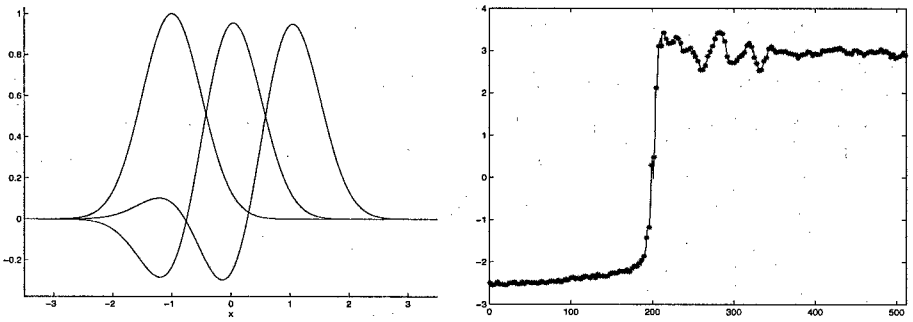


FIG. 1. First three orthogonalised basis functions and o-Gaussian fit to oscilloscope data.

2.2 Extensions to orthogonalised Gaussians

The following extensions are clearly possible.

(i) Use of generally placed centres (knots) and/or a discrete inner product.

(ii) Use of higher dimensions – as in Anderson *et al* [2].

(iii) Replacement of interval $(-\infty, \infty)$ in a continuous norm by $[0, n]$ and $[0, n]$ by $[0, 1]$ using scaling.

(iv) Consideration of a function with wider (approximate) support, such as $[-3, 3]$ or more generally $[-r, r]$ for $r > 2$.

3 Chebyshev polynomials in two-dimensional collocation

The (first kind) Chebyshev polynomial $T_i(x)$ of degree i is defined by

$$T_i(x) = \cos i\theta \qquad i = 0, \ldots, m, \quad -1 \leq x \leq 1, \tag{3.1}$$

where $x = \cos\theta$ and $0 \leq \theta \leq \pi$.

Among its many properties is the discrete orthogonality property

$$\sum_{k=1}^{m} T_i(x_k) T_j(x_k) = \begin{cases} 0 & \text{for} \quad i \neq j; \quad i, j \leq m-1 \\ m & \text{for} \quad i = j = 0 \\ \frac{1}{2}m & \text{for} \quad i = j \neq 0, \end{cases} \tag{3.2}$$

where x_k are the m zeros of $T_m(x)$, namely

$$x_k = \cos\left(\frac{(2k-1)\pi}{2m}\right), \quad k = 1, \ldots, m. \tag{3.3}$$

The orthogonality property of (3.2) is not a unique one amongst the Chebyshev polynomials of four kinds. Indeed, Mason and Venturino [5] showed that there are at least fourteen such formulae, depending on alternative weights, choices of Chebyshev-related abscissae and kinds of Chebyshev polynomial.

3.1 The elliptic problem — mixed methods

Let us now exploit this property (3.2) in a pseudo-spectral method for a linear elliptic PDE problem on a square. The PDE

$$Lu = f(x, y), \quad |x|, |y| \leq 1, \tag{3.4}$$

subject to

$$u = g(x, y), \tag{3.5}$$

where $g(x, y)$ is a function known explicitly only on $x = \pm 1$ and $y = \pm 1$, can be solved approximately in the form

$$u = u_{mn} = \sum_{i=0}^{m}{}' \sum_{j=0}^{n}{}' a_{ij} T_i(x) T_j(y), \tag{3.6}$$

where a dashed summation denotes that the first term in a sum is halved.

To obtain equations for a_{ij}, we solve

$$Lu_{mn} = f, \quad \text{at the } (m-1) \times (n-1) \text{ zeros of } T_{m-1}(x) T_{n-1}(y), \tag{3.7}$$

$$u_{mn} = g, \quad \text{on } x = \pm 1 \text{ at zeros of } T_n(y) \quad (2n \text{ equations}), \tag{3.8}$$

$$u_{mn} = g, \quad \text{on } y = \pm 1 \text{ at zeros of } T_m(x) \quad (2m \text{ equations}). \tag{3.9}$$

Together (3.7)–(3.9) form $(m+1) \times (n+1)$ equations for $\{a_{ij}\}$. However, we claim that the included equations (3.8), (3.9) are singular of joint rank $2m + 2n - 1$. If this is so, then the system is singular without consideration of the PDE collocation equations (3.7). The equations (3.8), (3.9) become

$$g_{k,\pm 1} = \sum_{i=0}^{m}{}' \sum_{j=0}^{n}{}' a_{ij} T_i(x_k) T_j(\pm 1), \quad g_{\pm 1,\ell} = \sum_{i=0}^{m}{}' \sum_{j=0}^{n}{}' a_{ij} T_i(\pm 1) T_j(y_\ell), \tag{3.10}$$

where x_k, y_ℓ are zeros of $T_m(x), T_n(y)$ respectively and where

$$g_{1,\ell} = g(1, y_\ell), \qquad g_{-1,\ell} = g(-1, y_\ell),$$
$$g_{k,1} = g(x_k, 1), \qquad g_{k,-1} = g(x_k, -1).$$

If we add/subtract the first pair and also the second pair of equations in (3.10), noting that

$$T_j(1) = 1, \quad T_j(-1) = (-1)^j,$$

we deduce that

$$d_k^{(0)} = \sum_{\substack{i=0 \\ (j \text{ even})}}^{m}{}' \sum_{j=0}^{n}{}' a_{ij} T_i(x_k), \quad d_k^{(1)} = \sum_{\substack{i=0 \\ (j \text{ odd})}}^{m}{}' \sum_{j=0}^{n}{}' a_{ij} T_i(x_k), \quad k = 1, \ldots, m, \tag{3.11}$$

$$e_k^{(0)} = \sum_{\substack{i=0 \\ (i \text{ even})}}^{m}{}' \sum_{j=0}^{n}{}' a_{ij} T_j(y_\ell), \quad e_k^{(1)} = \sum_{\substack{i=0 \\ (i \text{ odd})}}^{m}{}' \sum_{j=0}^{n}{}' a_{ij} T_j(y_\ell), \quad \ell = 1, \ldots, n, \tag{3.12}$$

where,

$$\begin{aligned} d_k^{(0)} &= \tfrac{1}{2}(g_{k,1} + g_{k,-1}), & d_k^{(1)} &= \tfrac{1}{2}(g_{k,1} - g_{k,-1}), \\ e_k^{(0)} &= \tfrac{1}{2}(g_{1,\ell} + g_{-1,\ell}), & e_k^{(1)} &= \tfrac{1}{2}(g_{1,\ell} - g_{-1,\ell}). \end{aligned}$$

Multiplying (3.11) by $2T_r(x_k)/(m+1)$ and summing over k, and multiplying (3.12) by $2T_s(y_\ell)/(n+1)$ and summing over ℓ, discrete orthogonality (3.2) gives

$$R_{r+1}^{(0)} \equiv \sum_{\substack{j=0 \\ (j \text{ even})}}^{n}{}' a_{rj} = b_{r+1}^{(0)}, \quad R_{r+1}^{(1)} \equiv \sum_{\substack{j=r \\ (j \text{ odd})}}^{n} a_{rj} = b_{r+1}^{(1)}, \quad r = 0, \ldots, m-1, \tag{3.13}$$

$$C_{s+1}^{(0)} \equiv \sum_{\substack{i=0 \\ (i \text{ even})}}^{m}{}' a_{is} = c_{s+1}^{(0)}, \quad C_{s+1}^{(1)} \equiv \sum_{\substack{i=0 \\ (i \text{ odd})}}^{m} a_{is} = c_{s+1}^{(1)}, \quad s = 0, \ldots, m-1, \tag{3.14}$$

where

$$\begin{aligned} b_{r+1}^{(0)} &= \frac{2}{m+1} \sum_{k=1}^{m} d_k^{(0)} T_r(x_k) & b_{r+1}^{(1)} &= \frac{2}{m+1} \sum_{k=1}^{m} d_k^{(1)} T_r(x_k), \\ c_{s+1}^{(0)} &= \frac{2}{n+1} \sum_{\ell=1}^{n} e_k^{(0)} T_s(y_\ell) & c_{s+1}^{(1)} &= \frac{2}{n+1} \sum_{\ell=1}^{n} e_k^{(1)} T_s(y_\ell). \end{aligned}$$

This constitutes a greatly simplified system to replace (3.10). Indeed we may verify that, for $m = n$,

$$\sum_{\substack{i=0 \\ (m-i \text{ odd})}}^{m-1}{}' R_{i+1}^{(t)} = \sum_{\substack{i=0 \\ (m-i \text{ odd})}}^{m-1}{}' C_{i+1}^{(t)}, \tag{3.15}$$

where $t = 0, 1$ for $m =$ odd, even, respectively, and hence that the equations (3.13) and (3.14) are singular. For example, for $m\ (= n) = 2$, we seek equations in $a_{00}, \ldots, a_{22}$, and (3.13) gives

$$\begin{aligned} R_1^{(0)} &\equiv \tfrac{1}{2}a_{00} + a_{02}, & R_1^{(1)} &\equiv a_{01}, \\ R_2^{(0)} &\equiv \tfrac{1}{2}a_{10} + a_{12}, & R_2^{(1)} &\equiv a_{11}, \end{aligned} \tag{3.16}$$

meanwhile (3.14) gives

$$\begin{aligned} C_1^{(0)} &\equiv \tfrac{1}{2}a_{00} + a_{20}, & C_1^{(1)} &\equiv a_{10}, \\ C_2^{(0)} &\equiv \tfrac{1}{2}a_{01} + a_{21}, & C_2^{(1)} &\equiv a_{11}. \end{aligned} \tag{3.17}$$

Clearly $R_2^{(1)} = C_2^{(1)}$, consistent with (3.15) for $m = 2$. Which equation do we eliminate? For simplicity, in the case of m even, we delete the equation for $C_2^{(1)}$ and replace it by the equation for $R_{m+1}^{(0)}$. It is easy to verify that, within the system (3.13) and (3.14), this leads to full rank, and $R_{m+1}^{(0)}$ is equivalent to boundary specifications of either of

$$\begin{gathered} u(0,1) + u(0,-1), \\ u(1,1) + u(-1,1) + u(1,-1) + u(-1,-1). \end{gathered} \tag{3.18}$$

For $m = n = 2$, this is equivalent to

$$R_3^{(0)} \equiv \frac{1}{2}a_{20} + a_{22}. \tag{3.19}$$

In the case when m is odd, we delete the equation for $C_1^{(0)}$ and replace it by the equation for $C_{m+1}^{(1)}$, the latter being equivalent to adding four boundary point conditions anti-symmetrically, i.e.,

$$u(1,1) - u(-1,1) + u(-1,-1) - u(1,-1). \tag{3.20}$$

If $g(x, y)$ is known everywhere in the square, then we could of course consider replacing a mixed collocation problem by an interior collocation problem by including the boundary conditions automatically in the form of approximations. For example, we could replace the form (3.6) by

$$u_{mn} = (x^2 - 1)(y^2 - 1) \sum_{i=0}^{m-2}{}' \sum_{j=0}^{n-2}{}' a_{ij} T_i(x) T_j(y) + g(x, y), \tag{3.21}$$

or by an alternative form such as

$$u_{mn} = \sum_{i=0}^{m}{}' \sum_{j=0}^{n}{}' a_{ij} \left(T_i(x) - T_{\underline{i}}(x)\right) \left(T_j(y) - T_{\underline{j}}(y)\right) + g(x, y), \tag{3.22}$$

where $T_{\underline{i}} = T_0(x)$ or $T_1(x)$ according as i is even or odd. These forms have the disadvantage of being difficult to generalise to other kinds of (non-rectangular) boundaries, although (3.21) is adaptable to the case where an equation of the boundary is known (see Mason [3]).

The best Chebyshev method available for the Poisson problem on a rectangle is probably a "differentiation matrix" method, such as is described in Trefethen [6], which represents the solution by nodal values rather than Chebyshev coefficients.

Acknowledgement: We thank the referees for their perceptive remarks.

Bibliography

1. T. N. T. Goodman, C. A. Micchelli, G. Rodriguez and S. Seatzu, On the Cholesky factorization of the Gram matrix of locally supported functions, BIT **35**(2), 1995, 233–257.
2. I. J. Anderson, J. C. Mason, G. Rodriguez and S. Seatzu, Training radial basis function networks using separable and orthogonalised Gaussians, in *Mathematics of Neural Networks*, S. W. Ellacot, J. C. Mason and I. J. Anderson (eds), Kluwer, 1997, 265–269.
3. J. C. Mason, Chebyshev polynomial approximations for the L-membrane eigenvalue problem, in *SIAM J. of Appl. Math* **15** (1967), 172–186.
4. J. C. Mason, G. Rodriguez and S. Seatzu, Orthogonal splines based on B-splines with applications to least squares, smoothing and regularisation problems, in *Numerical Algorithms* **5** (1993), 25–40.
5. J. C. Mason and E. Venturino, Integration methods of Clenshaw- Curtis type based on four kinds of Chebyshev polynomials, in *Multivariate Approximation and Splines*, G. Nuernberger, J. W. Schmidt and G. Walz (eds), Birkhauser, Basel, 1997, 158–165.
6. L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM, 2000.

Geometric knot selection for radial scattered data approximation

Rossana Morandi and Alessandra Sestini

Dipartimento di Energetica, Università di Firenze, IT.
morandi@de.unifi.it, sestini@de.unifi.it

Abstract

Scattered exact and non-exact data are approximated by means of radial basis functions with compact support and the related knot selection is based on the information given by the discrete Gaussian curvature defined on a data triangulation. In case of non-exact data, a strategy to obtain a sign-reliable estimate of its distribution is given extending an approach already studied by the authors for non-exact 2D data.

1 Introduction

It is well known that, for any interpolation/approximation scheme, data shape preservation is often a desirable quality and, as a consequence, the determination of some criteria to establish the data shape is a very important topic. For this purpose, the use of the discrete curvature in case of exact 2D data is a standard approach. On the other hand, in case of non-exact data, the proposal in [6] allows the determination of a reasonable and sign-reliable discrete curvature estimate if the maximum data error is *a priori* given. In recent literature, interesting formulas have been introduced [3, 4] for defining the discrete Gaussian curvature when scattered 3D exact data are given and a related triangulation is assigned. Starting from these formulas, the approach considered in [6] is extended to the case of 3D scattered non-exact data in order to define a reasonable and sign-reliable estimate of the Gaussian curvature at the data points thereby obtaining important shape information. Thus we get some suggestions for determining the supports of the local radial basis functions [8] used in the approximation scheme together with the number, the position and the multiplicity of the related knots. The result is a good approximating surface (in particular with respect to its shape) with a high data reduction [2, 7].

The outline of the paper is as follows. In Section 2 the discrete Gaussian curvature is defined and an inequality is given to check its sign-reliability in case of non-exact data. In Section 3 the approximation scheme is presented and the knot selection strategy is given. Finally, in Section 4 some numerical results are presented to illustrate the features of the proposed approach.

2 Information about the shape

In this section, following the approach presented in [3, 4], we define the discrete Gaussian curvature (dGc) to obtain information about the shape suggested by the data. For this

purpose, we need the following notation

- $\mathcal{P}_{xy} := \{\mathbf{X}_j = (x_j, y_j), j = 1, \ldots, N\} \subset \mathbb{R}^2$ is the set of the assigned distinct vertices on the xy-plane;
- $\mathcal{P} := \{\mathbf{P}_j = (\mathbf{X}_j, z_j), j = 1, \ldots, N\} \subset \mathbb{R}^3$ is the data set, with $z_j = f(\mathbf{X}_j)$;
- $\mathcal{T} := \{\mathbf{l}_j \in \mathbb{N}^3,\ 1 \leq l_{kj}, \leq N, k = 1, 2, 3, j = 1, \ldots, T\}$ is a given triangulation of $\mathcal{P}_{xy}$.

Thus, for any $\mathbf{X}_j \in \mathcal{P}_{xy}$ not belonging to the boundary of the convex hull of $\mathcal{P}_{xy}$ we can define the integral Gaussian curvature with respect to a related area S_j, [3]

$$\bar{K}_j := 2\pi - \sum_{k=1}^{n_j} \alpha_k^{(j)},$$

where the angles $\alpha_k^{(j)}, k = 1, \ldots, n_j$ are as follows

$$\alpha_k^{(j)} := \angle(\mathbf{e}_k^{(j)}, \mathbf{e}_{k+1}^{(j)}),\ \ \mathbf{e}_k^{(j)} := \mathbf{V}_k^{(j)} - \mathbf{P}_j,\ k = 1, \ldots, n_j,\ \ \mathbf{e}_{n_j+1}^{(j)} := \mathbf{e}_1^{(j)}$$

and $\{\mathbf{V}_1^{(j)}, \ldots, \mathbf{V}_{n_j}^{(j)}\} \subset \mathcal{P}$ is the set of ordered neighboring points of $\mathbf{P}_j$ given by the assigned triangulation. To derive the curvature at the vertex $\mathbf{P}_j$ from the above integral value, we normalize by the Voronoi area S_j [4]

$$K_j := \frac{\bar{K}_j}{S_j}. \tag{2.1}$$

If $\mathbf{X}_j$ is on the boundary of the convex hull of $\mathcal{P}_{xy}$, some auxiliary suitable "phantom" points should be defined in order to obtain a reliable estimate of the Gaussian curvature from (2.1).

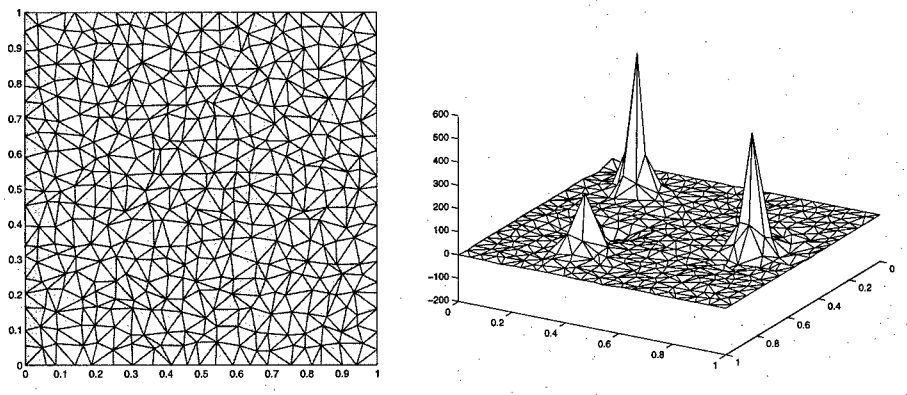


FIG. 1. The triangulation (left) and the discrete Gaussian curvature (right).

Shown on the left of Figure 1 is the Delaunay triangulation related to a set $\mathcal{P}_{xy}$ of 441 scattered vertices in the unit square and shown on the right is the discrete Gaussian curvature distribution related to the Franke function sampled on $\mathcal{P}_{xy}$.

In case of non-exact data, we need to check the sign-reliability of $\bar{K}_j$ for deriving some useful information about the shape suggested by the data. For this purpose, we use the theorem below, where

$$\begin{aligned} c_k^{(j)} &:= \frac{\mathbf{e}_k^{(j)} \cdot \mathbf{e}_{k+1}^{(j)}}{|\mathbf{e}_k^{(j)}|\,|\mathbf{e}_{k+1}^{(j)}|},\ k = 1, \ldots, n_j, \\ \mathcal{K}_j &:= \frac{\pi}{2}\left(4 - n_j + \sum_{k=1}^{n_j} c_k^{(j)}\right), \end{aligned} \tag{2.2}$$

Remark 2.1 *$\mathcal{K}_j$ is an approximation of $\bar{K}_j$ obtained by replacing the angle $\alpha_k^{(j)}$ with $\frac{\pi}{2}(1 - c_k^{(j)}), k = 1, \ldots, n_j$.*

Theorem 2.2 *Let $\mathbf{P}_j \in \mathbb{R}^3, j = 1, \ldots, N$ be assigned distinct non-exact data points and let ϵ be a positive quantity such that $|\mathbf{P}_j - \mathbf{P}_j^e| \le \epsilon,\ j = 1, \ldots, N$, where $\mathbf{P}_j^e$ is the (unknown) exact data point corresponding to $\mathbf{P}_j$. If ϵ is sufficiently small and*

$$|\mathcal{K}_j| > 5\pi\epsilon \sum_{k=1}^{n_j} \frac{1}{|\mathbf{e}_k^{(j)}|}, \tag{2.3}$$

then

$$\mathcal{K}_j \mathcal{K}_j^e > 0,$$

where $\mathcal{K}_j^e$ is defined as $\mathcal{K}_j$ using the exact data points.

Proof: Let us consider a point $\mathbf{P}_j$ and its neighboring points $\{\mathbf{V}_1^{(j)}, \ldots, \mathbf{V}_{n_j}^{(j)}\} \subset \mathcal{P}$ and let us write the corresponding (unknown) exact points as follows

$$\begin{aligned} \mathbf{P}_j^e &:= \mathbf{P}_j - \epsilon_0 \mathbf{w}_0, \\ \mathbf{V}_k^{(j)e} &:= \mathbf{V}_k^{(j)} - \epsilon_k \mathbf{w}_k, k = 1, \ldots, n_j \end{aligned}$$

with $0 \le \epsilon_0, \epsilon_1, \ldots, \epsilon_{n_j} \le \epsilon$ and $|\mathbf{w}_0| = |\mathbf{w}_1| = \cdots = |\mathbf{w}_{n_j}| = 1$.

So, if ϵ is sufficiently small, we can define the non-zero vectors

$$\mathbf{e}_k^{(j)e} := \mathbf{V}_k^{(j)e} - \mathbf{P}_j^e$$

and we have

$$\mathbf{e}_k^{(j)e} = \mathbf{e}_k^{(j)} - \epsilon_k \mathbf{w}_k + \epsilon_0 \mathbf{w}_0.$$

Thus, if

$$\mathbf{c}_k^{(j)e} := \frac{\mathbf{e}_k^{(j)e} \cdot \mathbf{e}_{k+1}^{(j)e}}{|\mathbf{e}_k^{(j)e}|\,|\mathbf{e}_{k+1}^{(j)e}|},$$

using a first order Taylor approximation, we obtain

$$\mathbf{c}_k^{(j)e} = \mathbf{c}_k^{(j)}(1 + A_k) + \frac{1}{|\mathbf{e}_k^{(j)}|\,|\mathbf{e}_{k+1}^{(j)}|} B_k + \mathcal{O}(\epsilon^2)$$

where

$$\begin{aligned} A_k &= \epsilon_k \frac{\mathbf{e}_k^{(j)} \cdot \mathbf{w}_k}{|\mathbf{e}_k^{(j)}|^2} + \epsilon_{k+1} \frac{\mathbf{e}_{k+1}^{(j)} \cdot \mathbf{w}_{k+1}}{|\mathbf{e}_{k+1}^{(j)}|^2} - \epsilon_0 \left(\frac{\mathbf{e}_k^{(j)}}{|\mathbf{e}_k^{(j)}|^2} + \frac{\mathbf{e}_{k+1}^{(j)}}{|\mathbf{e}_{k+1}^{(j)}|^2} \right) \cdot \mathbf{w}_0, \\ B_k &= -\epsilon_k \mathbf{e}_{k+1}^{(j)} \cdot \mathbf{w}_k - \epsilon_{k+1} \mathbf{e}_k^{(j)} \cdot \mathbf{w}_{k+1} + \epsilon_0 (\mathbf{e}_k^{(j)} + \mathbf{e}_{k+1}^{(j)}) \cdot \mathbf{w}_0. \end{aligned} \tag{2.4}$$

Thus, we can write

$$\mathcal{K}_j^e = \mathcal{K}_j \left(1 + \frac{\pi}{2\mathcal{K}_j} \sum_{k=1}^{n_j} \left(A_k c_k^{(j)} + \frac{B_k}{|\mathbf{e}_k^{(j)}||\mathbf{e}_{k+1}^{(j)}|} \right) \right) + \mathcal{O}(\epsilon^2).$$

So, if ϵ is sufficiently small, $\mathcal{K}_j \mathcal{K}_j^e > 0$ if

$$\frac{\pi}{2\mathcal{K}_j} \sum_{k=1}^{n_j} \left(A_k c_k^{(j)} + \frac{B_k}{|\mathbf{e}_k^{(j)}||\mathbf{e}_{k+1}^{(j)}|} \right) > -1$$

and this is true if

$$-\frac{\pi}{2|\mathcal{K}_j|} \sum_{k=1}^{n_j} \left(|A_k||c_k^{(j)}| + \frac{|B_k|}{|\mathbf{e}_k^{(j)}||\mathbf{e}_{k+1}^{(j)}|} \right) > -4/5. \tag{2.5}$$

Now, from (2.4) it is easy to verify that $|A_k| \leq 2\epsilon(|\mathbf{e}_k^{(j)}|^{-1} + |\mathbf{e}_{k+1}^{(j)}|^{-1})$ and $|B_k| \leq 2\epsilon(\mathbf{e}_k^{(j)}|^{-1} + |\mathbf{e}_{k+1}^{(j)}|^{-1})|\mathbf{e}_k^{(j)}||\mathbf{e}_{k+1}^{(j)}|$. Using these inequalities, after a little algebra, we obtain that, if ϵ is sufficiently small, (2.3) implies (2.5). □

If ϵ is an assigned small positive quantity such that $|\mathbf{P}_j - \mathbf{P}_j^e| \leq \epsilon$, $j = 1, \ldots, N$, if (2.3) holds we use (2.1) to define K_j because we consider it sign-reliable. Otherwise, we try to get information about the sign of the Gaussian curvature at the point $\mathbf{P}_j$, repeating the check after substituting the neighboring points of $\mathbf{P}_j$ with other new suitable n_j points. In particular, these are chosen among the neighbors of all the $\mathbf{V}_k^{(j)}, k = 1, \ldots, n_j$ and they are uniformly spaced as much as possible with respect to the azimuth (defined relating to $\mathbf{P}_j$). If after this substitution (2.3) holds the new neighboring points are used to define K_j through (2.1), otherwise this strategy is repeated until we consider that the new neighbors are too far from $\mathbf{P}_j$. In the last case, we put the curvature value equal to 0.

3 Knot selection in radial approximation

Let $\phi : \mathbb{R}_{\geq 0} \to \mathbb{R}$, be a compactly supported radial basis function. We approximate the given data by the surface

$$z(\mathbf{X}) := a_0 + \sum_{l=1}^{M} a_l \phi \left(\frac{\| \mathbf{X} - \mathbf{X}_l^* \|_2}{\delta_l} \right),$$

where the set of knots $\{\mathbf{X}_l^*, l = 1, \ldots, M\} \subset \mathcal{P}_{xy}$ and the set of positive δ–parameters $\{\delta_l, l = 1, \ldots, M\}$ are previously chosen. The coefficients $a_0, \ldots, a_M$ are determined minimizing $\sum_{j=1}^{N} (z_j - z(\mathbf{X}_j))^2$. The knot number and their positions are selected considering

the information given by the discrete Gaussian curvature distribution as defined in the previous section.

Inspired by the algorithm proposed in [6], the strategy for the $\mathbf{X}_l^*$ and $\delta_l, l = 1, \ldots, M$ choice can be summarized as follows:

- an input tolerance tol_G is given;
- a first set of distinct knots $\{\mathbf{X}_l^*, l = 1, \ldots, M_0\} \subset \mathcal{P}_{xy}$ with $M_0 \leq M$ is chosen. This is done selecting the areas where the absolute value of the discrete Gaussian curvature is greater than tol_G. A knot is located in the middle of an area if the sign of the related curvature is positive. In case of negative curvature, four knots are located near the boundary of the area also taking into consideration the suggestions given by the data distribution;
- initial values for the δ–parameters $\delta_l, l = 1, \ldots, M_0$ are determined considering the knot separation distance;
- the final set of knots is defined by possibly increasing the multiplicity of the previously selected knots. In this case, the δ–parameters associated to the same knot must be different.

Remark 3.1 *We observe that, to be sure that the least squares problem has a unique solution, it should be proved that the related collocation matrix is of full rank and this is clearly equivalent to the uniqueness of the corresponding interpolation problem (the only result we know about uniqueness of the radial interpolant defined with different scales is given in a submitted paper [1] where interesting sufficient conditions are given). However, we believe that the least squares problem is much more robust than the corresponding interpolation problem and in all the numerical experiments we have never had problems related to the rank of the collocation matrix (see also [5, 7]).*

4 Numerical results

In this section we use the compactly supported radial basis function [8]

$$\phi(r) := (1 - r)_+^3 (1 + 3r)$$

for checking the features of the proposed approach on two test functions. The first is the well known Franke function and the second is the function $z(\mathbf{X}) = 0.35(\sin(2\pi x) + \sin(2\pi y))$, $\mathbf{X} \in [0, 1]^2$. For both tests, $N = 441$ data points are considered. The exact data are obtained by evaluating the functions at the vertices represented on the left of Figure 1. The corresponding non-exact data are defined adding a random noise to the exact values. In particular, in the first test we have used $\epsilon = 0.07$ and in the second we have used $\epsilon = 0.08$, in $[0, 0.5]^2 \cup [0.5, 1]^2$ and $\epsilon = 0.008$, otherwise. The related discrete Gaussian curvature (dGc) distributions computed with the strategy sketched at the end of Section 2 are reported in Figure 2.

Figures 3 and 4 relate to the first test with exact and non-exact data, respectively. The distinct knots are $\mathbf{X}_1^* = (0.207, 0.205), \mathbf{X}_2^* = (0.449, 0.797), \mathbf{X}_3^* = (0.756, 0.349)$ and each of them is repeated three times with three different δ-parameter values, $0.6, 0.4, 0.3$. The mean error $\sqrt{\sum_{j=1}^{N}(z_j - z(\mathbf{X}_j))^2/N}$ is about 0.016 in Figure 3 and 0.025 in Figure

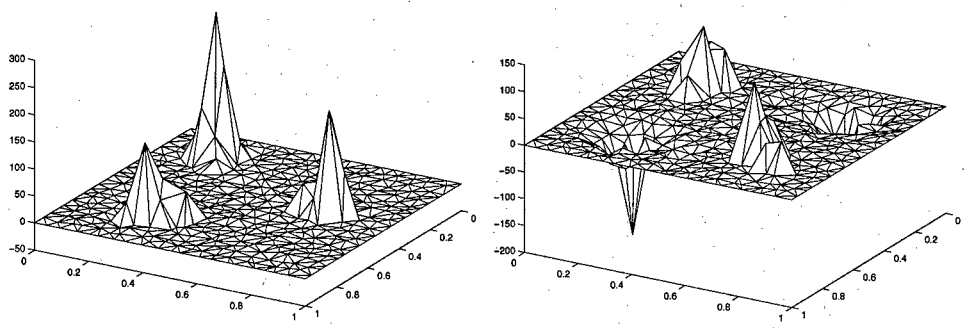


FIG. 2. dGc for the first (left) and second (right) set of non-exact data.

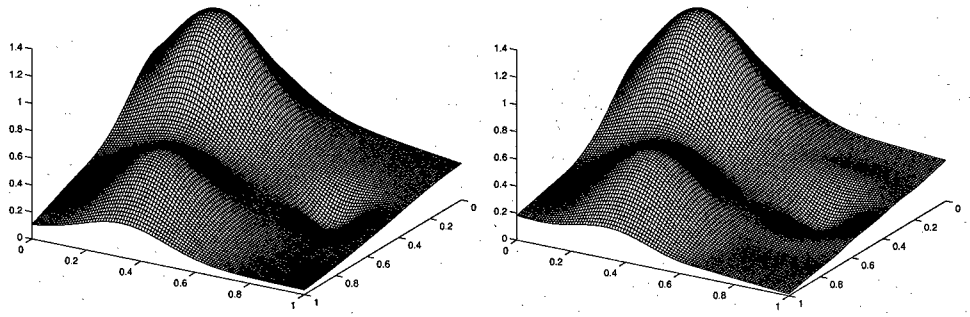


FIG. 3. The parent Franke surface (left) and its approximation (right).

4 (it was about 1/3 using only 3 distinct knots with all the δ–parameters equal to 0.6). Figures 5 and 6 relate to the second test. The distinct knots are $(0.258, 0.238)$, $(0.749, 0.737)$, $(0.950, 0.264)$, $(0.700, 0.264)$, $(0.756, 0.050)$, $(0.756, 0.300)$, $(0.050, 0.751)$, $(0.300, 0.751)$, $(0.264, 0.700)$, $(0.264, 0.950)$. The related δ–parameters are $0.8, 0.8, 0.6, 0.4, 0.6, 0.4, 0.6, 0.4, 0.4, 0.6$. The mean error is about 0.020 in Figure 5 and 0.026 in Figure 6.

Acknowledgments: The authors would like to thank the referees for their useful comments.

Bibliography

1. M. Bozzini, L. Lenarduzzi, M. Rossini and R. Schaback, Interpolation by basis functions of different scales and shapes, submitted to *Adv. Comp. Math.*, available at http://www.num.math.uni-goettingen.de/schaback/.

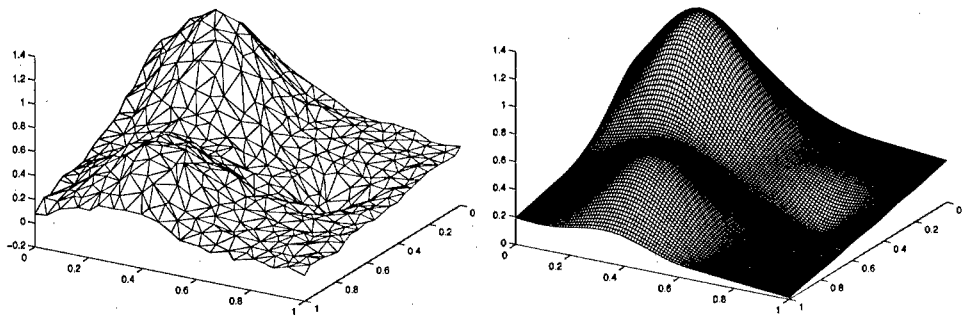


FIG. 4. The non-exact set of data (left) and its approximation (right).

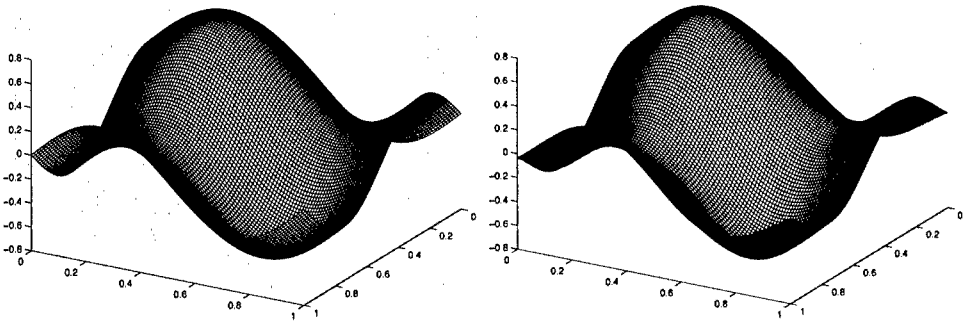


FIG. 5. The parent surface (left) and its approximation (right).

2. C. Conti, R. Morandi, C. Rabut and A. Sestini, Cubic spline data reduction: choosing the knots from a third derivative criterium, to appear in *Numerical Algorithms.*
3. R. van Damme and L. Alboul, Tight triangulations, *Mathematical Methods for Curves and Surfaces*, M. Dæhlen, T. Lyche and L.L.Schumaker (eds), Vanderbilt University Press, 1995, 517–526.
4. N. Dyn, K. Hormann, S. J. Kim and D. Levin, Optimizing 3D triangulations using discrete curvature analysis, *Mathematical Methods for Curves and Surfaces: Oslo 2000*, T. Lyche and L. L. Schumaker (eds), Vanderbilt University Press, 2001, 135–146.
5. R. Franke, H. Hagen and G. Nielson, Least squares surface approximation to scattered data using multiquadric functions, *Adv. Comp. Math.* **2** (1994), 81–99.
6. R. Morandi, D. Scaramelli and A. Sestini, A geometric approach for knot selection in convexity–preserving spline approximation, *Curve and Surface Design*, P.J. Laurent, P. Sablonniere and L.L. Schumaker (eds), Vanderbilt University Press, 2000, 287–296.

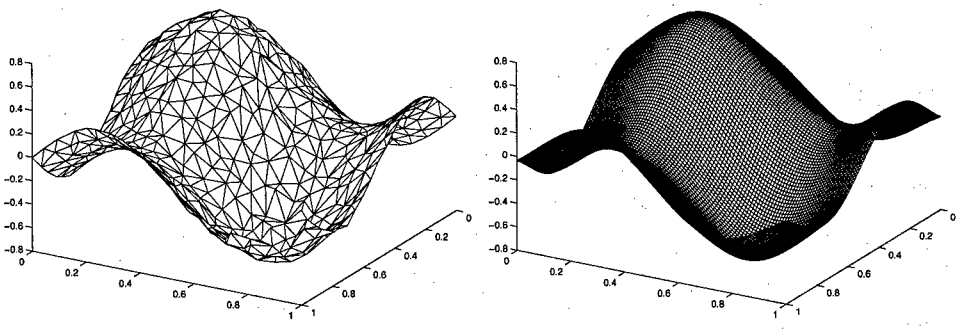


FIG. 6. The non-exact set of data (left) and its approximation (right).

7. R. Morandi and A. Sestini, Data reduction in surface approximation, *Mathematical Methods for Curves and Surfaces: Oslo 2000*, T. Lyche and L. L. Schumaker (eds), Vanderbilt University Press, 2001, 315–324.
8. H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Adv. Comp. Math.* **4** (1995), 389–396.

On the boundary over distance preconditioner for radial basis function interpolation

C. T. Mouat and R. K. Beatson

Dept. of Mathematics and Statistics, Univ. of Canterbury, Christchurch, New Zealand.
cam@mouat.net, R.Beatson@math.canterbury.ac.nz

Abstract

In this paper we consider the boundary over distance preconditioner for radial basis function interpolation problems. We give both theoretical and numerical results indicating that it performs extremely well.

1 Introduction

Let $\Phi : \mathcal{R}^d \to \mathcal{R}$, $X = \{x_1, \ldots, x_N\}$ be a set of N distinct points in $\mathcal{R}^d$ and f be a real valued function which we can evaluate at least at the x_i's. Define

$$S_{\Phi,X} = \left\{ \begin{array}{l} g : g = \sum_{i=1}^{N} \lambda_i \Phi(\cdot - x_i) \\ \quad \text{where } \sum_{j=1}^{N} \lambda_j q(x_j) = 0, \quad \text{for all } q \in \pi_1^d \end{array} \right\}. \tag{1.1}$$

We consider the problem of finding an element s of $S_{\Phi,X} + \pi_1^d$ satisfying the interpolation conditions

$$s(x_i) = f(x_i), \quad \text{for all} \quad x_i \in X. \tag{1.2}$$

Assume Φ is strictly conditionally positive definite of order 2 (SCPD2) and X is unisolvent for π_1^d. Then there is a unique element of $S_{\Phi,X} + \pi_1^d$ satisfying the interpolation conditions (1.2). This setting includes popular choices of the basic function such as the thin-plate spline, $\Phi(\cdot) = |\cdot|^2 \log|\cdot|$, and minus the ordinary multiquadric, $\Phi(\cdot) = -\sqrt{|\cdot|^2 + c^2}$. In this paper we consider various ways of formulating the interpolation problem, showing in particular that a certain inexpensive change of basis can dramatically improve its conditioning.

The usual way to formulate this problem is in terms of the functions $\{\Phi(\cdot - x_i)\}$ and some basis $\{p_0, p_1, \ldots, p_d\}$ for π_1^d. Then the interpolation conditions together with the side conditions taking away the extra degrees of freedom introduced by the polynomial part can be written as

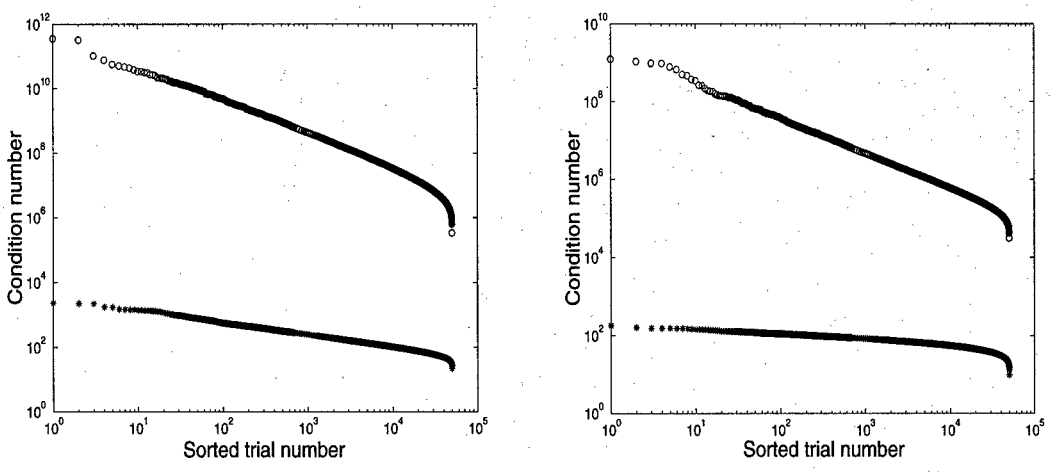
$$A\lambda + Pc = f \qquad \text{and} \qquad P^T\lambda = 0, \tag{1.3}$$

where $A_{ij} = \Phi(x_i - x_j)$, $P_{ij} = p_j(x_i)$, and $f = [f(x_1), \ldots, f(x_N)]^T$. It is well known [3, 4, 5] that the matrix

$$A_\Phi = \begin{bmatrix} A & P \\ P^T & O \end{bmatrix}, \tag{1.4}$$

of this usual formulation is frequently badly conditioned, even when the number of nodes is small. Indeed many authors have commented on the numerical difficulties that solving this system presents [3, 4, 5]. Results of Narcowich and Ward show that conditioning of the system (1.4) depends very heavily on the geometry of the nodes. However, frequently in numerical analysis a change of basis, or other reformulation, can make a highly intractable problem tractable. Hence, our goal is to find an inexpensive but highly effective preconditioner for RBF interpolation systems.

In this paper we establish properties of a preconditioning method for the RBF interpolation equations which was first presented in Sibson and Stone [5]. In the following section we give a detailed account of the preconditioning method. In Section 3 we prove that the construction produces an $N \times (N-3)$ matrix Q whose columns are orthogonal to P, and which is of full rank whenever the nodes X are unisolvent for π_1^2. Finally, Section 4 contains numerical results for different SCPD2 basic functions over a range of data sets and scales. These numerical results show that using this inexpensive $\mathcal{O}(N \log N)$ flop preconditioner and variants of it, dramatically improves the conditioning of RBF interpolation problems. See Figure 1 below.



(a) Multiquadric basic function.

(b) Thin-plate spline basic function.

FIG. 1. Sorted 2-norm condition numbers of the unpreconditioned matrices, A_Φ, (top) and of the preconditioned matrices, S, (bottom) for fifty thousand random data sets of size one hundred.

2 A preconditioning method

A general approach to preconditioning interpolation problems with SCPD2 basic functions in $\mathcal{R}^2$ [1, 5] is to choose Q as any $N \times (N-3)$ matrix whose columns are orthogonal

to P and has rank $N-3$. Letting $\lambda = Q\mu$ and premultiplying (1.3) by Q^T gives the new system to be solved for μ, or equivalently λ,

$$B\mu = Q^T f \qquad \text{where} \qquad B = Q^T A Q. \tag{2.1}$$

The three polynomial coefficients can then be found by a small subsidiary calculation.

In this section we present the boundary over distance method of Sibson and Stone [5] for constructing the matrix Q. We will prove in the subsequent section that Q has full rank and is orthogonal to P for any set of distinct nodes $X = \{x_1, \ldots, x_N\} \subset \mathcal{R}^2$, which are unisolvent for π_1^2. These properties of Q are well known (see e.g. [1, 5]) to imply that the matrix of the preconditioned system $B = Q^T A Q$ is positive definite. The construction of Q is appealing in that for "interior" points x_j of X it is local. That is, for such points the entries in the j-th column of Q depend only on the geometry of the nodes near x_j and not on any properties of nodes far away.

Choose a closed bounded convex polygonal region W of $\mathcal{R}^2$ such that $X \subset W$. Suppose without loss of generality that $\{x_{N-2}, x_{N-1}, x_N\}$ is unisolvent for π_1^2. We will refer to these points as special points. They are generally chosen so that they are well spread throughout W. In our experience, and that of Sibson and Stone, for typical data sets the choice of special points is not at all critical, as long as the triangle they define has largish area. However, for contrived data sets, such as all but a very few points on a straight line, the choice of special points becomes important. In these cases we have observed that bad choices of special points can lead to large condition numbers. However, the strategy of choosing the three special points to maximise the area of the corresponding triangle has always led to small condition numbers.

The region W is divided into panels by intersecting a Voronoi diagram of the points of X with the region W. We denote this panelling of W by

$$V_W(X) = \bigcup_{i=1}^{N} V_i$$

where V_i is the Voronoi panel about the ith centre and is defined by

$$V_i = \{x \in W : |x - x_i| < |x - x_j|, \quad \text{for all } 1 \le j \le N \text{ with } j \ne i\}.$$

Recall that the locus of points equidistant from two fixed points is the perpendicular bisector of the segment connecting the points. It follows that each Voronoi region is polygonal. Associated with a panel V_i are its edges. These are a finite number of distinct closed line segments of non-zero length. They are the boundaries between different Voronoi panels, or between a Voronoi panel and W^C. The collection of all edges of all the Voronoi panels will be denoted by $\mathcal{E}$.

Definition 2.1 *Two polygonal regions of $\mathcal{R}^2$ will be said to be strongly contiguous if they have a common boundary of non-zero length.*

Definition 2.2 *Two Voronoi regions V_i and V_j will be said to be C-related if there is a sequence*

$$\{V_i, V_{\ell_1}, V_{\ell_2}, \ldots, V_{\ell_m}, V_j\}, \qquad 1 \le i, j, \ell_1, \ldots, \ell_m \le N-3,$$

in which all adjacent pairs are strongly contiguous.

Loosely speaking V_i and V_j are C-related if they are connected by a chain of strongly contiguous pairs. C-related is an equivalence relation on the set $\{V_i\}_{i=1}^{N-3}$ of Voronoi regions of non-special points. Therefore it breaks this set into a finite number of nonempty equivalence classes $\{\mathcal{G}_l : 1 \leq l \leq k\}$.

Lemma 2.3 *Let $\mathcal{G}_\ell$ be any of the equivalence classes above. Then there is at least one Voronoi region V_i in $\mathcal{G}_\ell$ which is strongly contiguous to either W^C or one of $\{V_{N-2}, V_{N-1}, V_N\}$.*

Proof: Consider
$$T = \bigcup_{i:V_i \in \mathcal{G}_\ell} \overline{V_i}\,.$$
This union is a closed bounded connected polygonal set whose boundary can be written as the union of some of the line segments from $\mathcal{E}$. Recall in particular that all these line segments have non-zero length. Pick one line segment $< a, b >$ from the boundary of T. Since it forms part of the boundary of T on one side of it lies a Voronoi region V_i from $\mathcal{G}_\ell$. On the other side lies either W^C or another Voronoi region V_j. In the first case the Lemma is proven. Consider the second case. If $1 \leq j \leq N-3$ then V_i is strongly contiguous to V_j. Consequently, $V_j \in \mathcal{G}_\ell$. This contradicts $< a, b >$ being on the boundary of T. Hence, $N-2 \leq j \leq N$ and the Lemma follows. □

We now detail the construction of the $N \times (N-3)$ matrix Q using boundary over distance weights. Note that because most elements of Q are zero sparse storage of Q requires only $\mathcal{O}(N)$ memory. A non-special point from $\{x_i : 1 \leq i \leq N-3\}$ which has Voronoi tile that is strongly contiguous to W^C will be called a *Voronoi external point.* Define $V_E(X)$ as the set of indices of all Voronoi external points. All other points are referred to as *Voronoi internal points.* The corresponding indices are $V_I(X) = \{1, \ldots, N-3\} - V_E(X)$.

We first consider forming a column of Q for an index, j, such that $j \in V_I(X)$. In this case the panel V_j shares non-trivial edges only with other Voronoi panels and not with W^C. The column is formed using boundary over distance weights, found from the Voronoi diagram. For $j \in V_I(X)$ the boundary over distance weight r_{ij} is

$$r_{ij} = \frac{b(x_i, x_j)}{|x_i - x_j|}, \quad \text{for all } V_i \text{ strongly contiguous to } V_j, \tag{2.2}$$

where $b(x_i, x_j)$ is the length of the boundary between V_i and V_j. For other values of $i \neq j$, r_{ij} is set to zero. In order that column j of Q is orthogonal to constants the diagonal element r_{jj} is specified as

$$r_{jj} = -\sum_{i \neq j} r_{ij}. \tag{2.3}$$

Finally, the jth column of R is scaled by dividing by the area of V_j to obtain the jth column of Q. Note that the column is by construction diagonally dominant, but not strictly so.

If $j \in V_E(X)$ then V_j is stongly contiguous to the complement of W, W^C. The boundary segment corresponds to a Voronoi edge between x_j and an artificial point, the reflection of x_j in the boundary (see Figure 3 in [7]). The reflected point, $\hat{x}_j$, can be

written as a linear combination of the special points, i.e.,

$$\hat{x}_j = \lambda_N x_N + \lambda_{N-1} x_{N-1} + \lambda_{N-2} x_{N-2}, \tag{2.4}$$

where $\lambda_N + \lambda_{N-1} + \lambda_{N-2} = 1$. If V_j has k edges with W^C then k reflected points $\{\hat{x}_j^1, \ldots, \hat{x}_j^k\}$ are required. Associated with each reflected point, $\hat{x}_j^a$, are the coefficients $\{\lambda_N^a, \lambda_{N-1}^a, \lambda_{N-2}^a\}$. The boundary over distance weights for $\hat{x}_j^a$ are partitioned amongst the special points to obtain for all $j \in V_E(X)$ and $i \neq j$

$$r_{ij} = \begin{cases} \frac{b(x_i,x_j)}{|x_i - x_j|}, & V_i \text{ strongly contiguous to } V_j, \\ \sum_{l=1}^{k} \lambda_i^l \frac{b(\hat{x}_j^l, x_j)}{|\hat{x}_j^l - x_j|}, & i \in \{N, N-1, N-2\}. \end{cases} \tag{2.5}$$

Of course, V_j could be strongly contiguous with a Voronoi panel associated with a special point. If this is the case $r_{ij} = \frac{b(x_i,x_j)}{|x_i - x_j|} + \sum_{l=1}^{k} \lambda_i^l \frac{b(\hat{x}_j^l, x_j)}{|\hat{x}_j^l - x_j|}$. Again, for other values of $i \neq j$, r_{ij} is set to zero. Finally r_{jj} is specified as in (2.3) and column j of Q is defined as column j of R scaled by dividing by the area of V_j.

Partition Q as

$$Q = \begin{bmatrix} E \\ F \end{bmatrix}, \tag{2.6}$$

where E is $(N-3) \times (N-3)$. Thus E results from interactions between non-special points, and F those between special and non-special points. Note in the construction above that for $1 \leq i, j \leq N-3$, e_{ij} is non-zero if and only if V_i is strongly contiguous to V_j. Furthermore, note that E is necessarily column diagonally dominant, with strict dominance in column j whenever V_j is strongly contiguous to the Voronoi region of a special point, or to W^C.

Relabelling if necessary we can assume the indices of the Voronoi regions in each of the equivalence classes $\mathcal{G}_i$ form a contiguous subset of $\{1, \ldots, N-3\}$. Similarly, we can also assume that the indices corresponding to any $\mathcal{G}_i$ precede those corresponding to $\mathcal{G}_{i+1}$. Furthermore, by construction if $i \neq j$ none of the regions in $\mathcal{G}_i$ is strongly contiguous with a region in $\mathcal{G}_j$. Thus, corresponding entries in the matrix E constructed using boundary over distance weights and artificial points are zero. That is E is block diagonal with the square matrix E_{ii} on the main diagonal corresponding to the equivalence class of Voronoi regions $\mathcal{G}_i$. More precisely, Q will have form

$$Q = \begin{bmatrix} E_{11} & O & \cdots & O \\ O & E_{22} & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & E_{kk} \\ F_1 & F_2 & \cdots & F_k \end{bmatrix}. \tag{2.7}$$

3 Properties of the matrix Q

In this section we establish the fundamental properties of the matrix Q of (2.7). Namely that it is of full rank and that its columns are orthogonal to those of P.

Definition 3.1 *For $m \geq 2$, an $m \times m$ matrix K is irreducible if there does not exist an $m \times m$ permutation matrix P such that*

$$PKP^T = \begin{bmatrix} M_{11} & M_{12} \\ 0 & M_{22} \end{bmatrix},$$

where M_{11} is $r \times r$, M_{22} is $(m-r) \times (m-r)$, and $1 \leq r < m$.

The following result is well known, see for example Varga [6].

Theorem 3.2 *Suppose the square matrix K is irreducible and row (column) diagonally dominant with strict row (column) diagonal dominance in at least one row (column). Then K is invertible.*

Lemma 3.3 *Let X be a finite set of distinct points unisolvent for π_1^2. Let E_{ii} be one of the square blocks from the diagonal of Q constructed in the previous section. Then E_{ii} is invertible.*

Proof: From the construction E_{ii} is column diagonally dominant. Furthermore, by Lemma 2.3 the diagonal dominance is strict for at least one column of E_{ii}. From the definition of the equivalence relation C-related there is a chain of strongly contiguous pairs of Voronoi regions, connecting any two Voronoi regions in $\mathcal{G}_i$. This implies the corresponding entries in E_{ii} are non-zero and hence from [6] Theorem 1.6 E_{ii} is irreducible. It follows from Theorem 3.2 that E_{ii} is invertible. □

Theorem 3.4 *The matrix Q described in Section 2 is orthogonal to P i.e. $Q^T P = O$.*

Proof: Omitted, see [2] and [7]. □

Theorem 3.5 *Let X be a set of distinct points unisolvent for π_1^2. Let Q be formed by the construction in Section 2 and $A_{ij} = \Phi(x_i - x_j)$ where Φ is strictly conditionally positive definite of order 2. Then $B = Q^T A Q$ is positive definite.*

Proof: From Lemma 3.3 each of the matrices E_{ii} occurring in the block partitioning of Q given in Equation (2.7) is invertible. Hence Q has full rank. Also from Theorem 3.4 the columns of Q are orthogonal to the columns of P. Let μ be any non-zero vector in $\mathcal{R}^{N-3}$, and define $\lambda = Q\mu$. Then $\lambda \neq 0$, $P^T\lambda = P^T Q\mu = 0$, and $\mu^T B\mu = \mu^T Q^T A Q\mu = \lambda^T A\lambda$. Hence, by the definition of strictly conditionally positive definite, $\mu^T B\mu > 0$ whenever $\mu \neq 0$ and B is symmetric positive definite. □

Theorem 3.6 *Let Φ be strictly conditionally positive definite of order 2 and such that $\Phi(hx, hy) = h^\gamma \Phi(x, y) + p_h(x - y)$, $h > 0$ with $p_h \in \pi_3^2$. The preconditioned matrix B_h, which corresponds to preconditioning on the point set hX, is a homogeneous function of scale. Thus its condition number and the relative clustering of its eigenvalues are the same over all scales.*

Proof: Omitted, see [7]. □

Theorem 3.6 applies in particular to the usual thin-plate spline, $\Phi(\cdot) = |\cdot|^2 \log|\cdot|$, in $\mathcal{R}^2$.

The extended version of this paper [7] contains a proof that the elements B_{ij} decay like $|x_i - x_j|^{-\kappa}$ when $|x_i - x_j|$ is large. For the multiquadric κ is three and for the thin-plate spline κ is two.

Definition 3.7 *The preconditioned matrix S is obtained from B by pre-multiplying and post-multiplying B by the diagonal matrix D with ii entry $1/\sqrt{b_{ii}}$.*

4 Numerical results

In this section we present numerical results for the thin-plate spline and multiquadric basic functions. In the following tables the matrix A_Φ is defined in (1.4), B in (2.1), S in Definition 3.7 and the homogeneous matrix, C, is presented in [1]. In Table 1 we show 2–norm condition numbers of matrices for the various preconditioning techniques over seven different scales. It is clear that the algorithm in Section 2 gives a matrix which dramatically improves the conditioning of the interpolation problem. In one case by a factor of 10^{14}! Tables 2 and 3 contain condition numbers of the matrices resulting from applying the preconditioning techniques of this paper for the thin-plate spline and multiquadric basic functions. For $N < 3200$, the entries in the tables are the maximum over one hundred random point sets of size N. For $N = 3200$, the tables contain the maximum over twenty random point sets of size 3200. In all cases the preconditioning results in a smaller condition number. For these basic functions the maximum observed condition number of the scaled preconditioned matrix, S, grows very slowly with N. Certainly there is no numerical evidence of power growth with N.

Scale parameter α	Conventional matrix A_ϕ	Homogeneous matrix C	Preconditioned matrix B	Scaled matrix S
0.001	1.531(11)	1.534(5)	4.905(1)	2.405(1)
0.01	1.544(9)	1.534(5)	4.905(1)	2.405(1)
0.1	1.597(7)	1.534(5)	4.905(1)	2.405(1)
1	3.107(5)	1.534(5)	4.905(1)	2.405(1)
10	1.915(6)	1.534(5)	4.905(1)	2.405(1)
100	1.271(11)	1.534(5)	4.905(1)	2.405(1)
1000	4.006(15)	1.534(5)	4.905(1)	2.405(1)

TAB. 1. Condition numbers for one hundred points in $[0,\ \alpha]^2$ and the thin-plate spline. The point set for scale α is $X_\alpha = \alpha X_1$.

In an attempt to rule out the possibility that our numerical results were flukes due to the small number of 100 experiments we also conducted 50,000 trials with random data sets of size 100. The results of these trials are shown in Figure 1. The maximum condition number, over all trials with the thin-plate spline, for the matrix A_ϕ was 1.2465(9), for matrix C, 1.5750(9) and for matrix S, 1.8066(2). In our experiments the matrix S is always well conditioned. This held even for geometries of centres for which the matrix A_ϕ is very badly conditioned.

To test further the behaviour of S for "bad" configurations of points a similar experiment was run with one thousand trials of one hundred points almost on a circle. The maximum condition numbers of the A matrix, C matrix and S matrix were 1.2885(9), 7.2692(8) and 6.6005(2) respectively over 1000 trials. Even though the Voronoi regions

Number of data points	Conventional matrix A_ϕ	Homogeneous matrix C	Preconditioned matrix B	Scaled matrix S
200	6.555(7)	3.068(7)	1.617(3)	6.028(1)
400	5.675(8)	3.397(8)	1.945(3)	8.946(1)
800	1.960(10)	1.348(10)	2.034(3)	9.775(1)
1600	1.092(10)	8.413(9)	8.099(3)	1.258(2)
3200	4.997(10)	3.783(10)	1.261(4)	1.569(2)

TAB. 2. Maximum condition numbers encountered over a sample of 100 random point sets of size N in $[0,\ 1]^2$ with the thin-plate spline.

Number of data points	Conventional matrix A_ϕ	Preconditioned matrix B	Scaled matrix S
200	2.014(8)	1.532(2)	4.224(1)
400	2.045(10)	5.932(2)	7.669(1)
800	6.641(10)	4.559(2)	5.826(1)
1600	1.554(10)	7.025(2)	5.601(1)
3200	2.477(11)	9.362(2)	6.280(1)

TAB. 3. Maximum condition numbers encountered over a sample of 100 random point sets of size N in $[0,\ 1]^2$ with the multiquadric function, parameter $c = 1/\sqrt{N}$.

are long and thin the matrix S is still well conditioned!

Bibliography

1. R. K. Beatson, W. A. Light and S. Billings, Fast solution of the radial basis function interpolation equations: Domain decomposition methods, *SIAM Journal on Scientific Computing*, **22** (2000), 1717-1740.
2. N. H. Christ, R. Friedberg and T. D. Lee, Weights of links and plaquettes in a random lattice, *Nuclear Physics B* **210** (1982), 337-346.
3. N. Dyn, D. Levin and S. Rippa, Numerical procedures for surface fitting of scattered data by radial functions, *SIAM Journal of Scientific and Statistical Computing*, **7** (1986), 639-659.
4. F. J. Narcowich and J. D. Ward, Norm estimates for the inverse of a general class of scattered-data radial-function interpolation matrices, *Journal of Approximation Theory*, **69** (1992), 84-109.
5. R. Sibson and G. Stone, Computation of thin-plate slines, *SIAM Journal on Scientific and Statistical Computing*, **12** (1991), 1304-1313.
6. R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, New Jersey (1962).
7. C. T. Mouat and R. K. Beatson, Some properties of the boundary over distance preconditioner for radial basis function interpolation, Research report UCDMS 2001/**6**, *Department of Mathematics and Statistics, University of Canterbury*, (2001).

What are 'good' points for local interpolation by radial basis functions?

Robert P. Tong

The Numerical Algorithms Group Ltd, Jordan Hill, Oxford, OX2 8DR, UK.
robert.tong@nag.co.uk

Andrew Crampton

School of Computing and Mathematics, University of Huddersfield, Huddersfield, UK.
a.crampton@hud.ac.uk

Anne E. Trefethen

The Numerical Algorithms Group Ltd, Jordan Hill, Oxford, OX2 8DR, UK.
anne.trefethen@nag.co.uk

Abstract

Radial basis function interpolation has an advantage over other methods in that the interpolation matrix is nonsingular under very weak conditions on the location of the interpolation points. However, we show that point location can have a significant effect on the performance of an approximation in certain cases. Specifically, we consider multiquadric and thin plate spline interpolation to small data sets where derivative estimates are required. Approximations of this type are important in the motion of unsteady interfaces in fluid dynamics. For data points in the plane, it is shown that interpolation to data on a circle can be related to the polynomial case. For scattered data on the sphere, a comparison is made with the results of Sloan and Womersley.

1 Introduction

Radial basis functions (RBFs) such as multiquadrics or thin plate splines have been successfuly used for scattered data approximation in many applications. They have been shown to perform well for data fitting, although problems of ill-conditioning and the computational cost of processing large data sets must be handled carefully. In general, when considering the accuracy of a RBF interpolant, a balance must be achieved between the reduction in fill distance necessary for convergence of the approximation to an assumed underlying function and the need to maximise the separation distance between data points to avoid problems of ill-conditioning [4].

In the present study, we focus on the use of RBF approximation as one stage of a larger algorithm to compute the evolution of an unsteady interface in fluid dynamics. The accuracy of the approximations made in the algorithm and the interaction between its different stages determine whether the output is close to the true solution of the

governing equations or whether spurious effects are produced. In the three-dimensional setting, a typical example is described by Zinchenko *et al.* [8] where the deformation of liquid drops in a viscous medium is studied. A critical feature of the algorithm is the approximation of the normal directions and curvatures of the droplet surface defined at a number of discrete points.

The focus here is algorithmic rather than theoretical and we investigate the performance of multiquadric and thin plate spline local interpolants applied to the determination of normal directions and curvatures of a smooth, closed surface. Certain configurations of data points, such as points located on a circle, impose constraints on the interpolant. A framework for understanding the behaviour of the RBF interpolants is provided by a comparison with the multivariate polynomial interpolant of de Boor and Ron [1] and by considering the free parameter in the multiquadric as a tensioning parameter [2].

2 Approximation method

A common approach to solving fluid dynamics problems that include moving interfaces, combines a computational grid with meshless approximation methods. The governing partial differential equations, or corresponding integral equation formulation, are solved on the grid, while quantities characterising the interface are computed as meshless scattered data approximations.

Here we examine the behaviour of local RBF approximations in the general context described by Zinchenko *et al.* [8]. For a given data set, a particular point is selected together with its nearest neighbours giving a set of typically 6 or 7 points. The initial locations of these points may be determined by a regular mesh, but the surface is allowed to deform so that the approximation is essentially to a small set of scattered data. The constructed RBF interpolant, S, can be expressed as

$$S(x) = \sum_{j=1}^{N} a_j \phi(||x - x_j||) + \sum_{i=1}^{K} b_i p_i(x),$$

with the constraint

$$\sum_{j=1}^{N} a_j p_i(x_j) = 0, \qquad \text{for } 1 \le i \le K,$$

where $x \in \Re^2$ and $\{p_i(x)\}_{i=1:K}$ is a basis for the space of bivariate polynomials of degree $\le m-1$ with $K = m(m+1)/2$. The chosen forms for ϕ are the thin plate spline

$$\phi(||x - x_j||) = ||x - x_j||^2 \log ||x - x_j||, \qquad \text{(TPS)}$$

and the multiquadric

$$\phi(||x - x_j||) = (||x - x_j||^2 + c^2)^{\frac{1}{2}}, \qquad \text{(MQ)}$$

with $||\cdot||$ taken to be the Euclidean norm.

A framework for interpreting the computed results in the context being considered can be derived from [2] where the arbitrary parameter, c, of the MQ function is viewed as a tensioning parameter. As $c \to \infty$ the MQ interpolant approaches the correspond-

ing polynomial interpolant to the given data, while as $c \to 0$, the MQ surface is tensioned. Multivariate polynomial interpolation can fail on particular point sets and this has provided a motivation for using RBF methods. However, the algorithm of de Boor and Ron [1] provides a reliable means of computing the 'least' polynomial interpolant. This algorithm is used to compute a polynomial fit as one reference point for the interpretation of the MQ interpolants. A second reference point is provided by the TPS interpolant which gives a minimum energy surface in a certain norm. This is shown to correspond closely to the MQ fit for a 'small', but nonzero value of c. The MQ interpolant can thus be shown to connect the minimum energy, tensioned, TPS surface with the polynomial fit to given data as c increases. In a fluid dynamics context a fluid–fluid interface is often assumed to be represented by a C^∞ function (although cusps may occur requiring a change in the representation). This would suggest that a high degree polynomial would be preferred to a TPS surface.

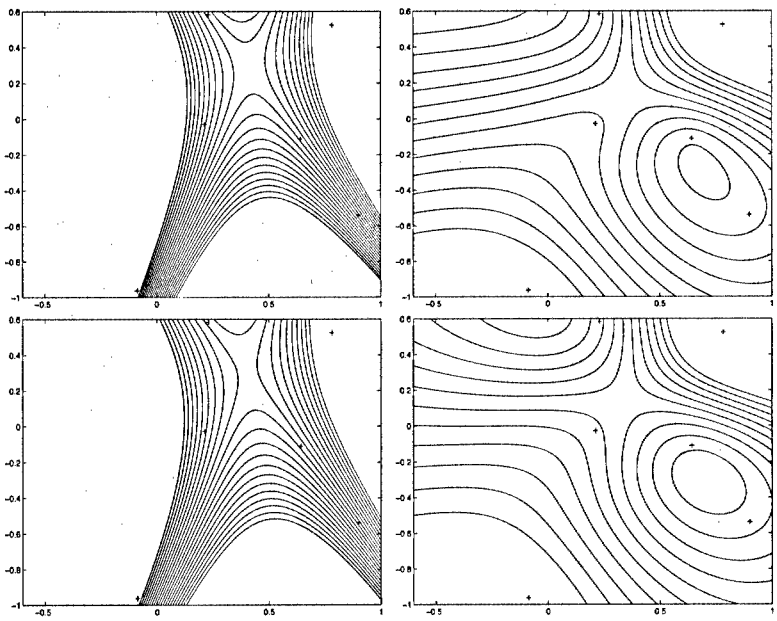


FIG. 1. Interpolants to random data at 6 points (+) in the plane: (left) polynomial (upper) and multiquadric (c = 10) (lower), contours [0:0.1:2]; (right) thin plate spline (upper) and multiquadric (c = 0.4) (lower), contours [0:0.1:1.1].

3 Scattered data in the plane

To illustrate the behaviour of local interpolation by MQ and TPS methods, random points in the xy-plane (with $-1 < x_i, y_i < 1$, for $i = 1 : 6$) are associated with random data values, f_i $(-1 < f_i < 1)$. Figure 1 shows, in the upper frames, the two reference

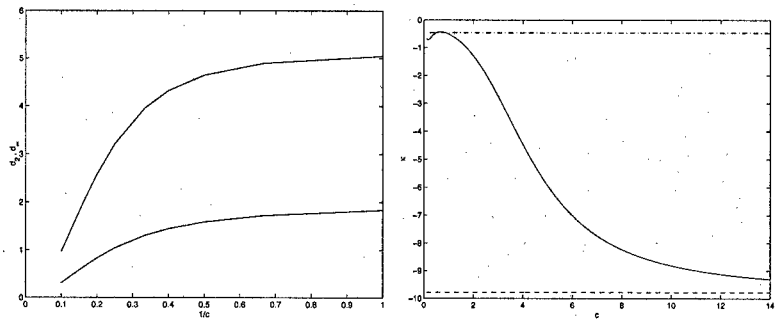


FIG. 2. Effect of varying the parameter c on multiquadric interpolants to random data in the plane: (left) norms of the difference between multiquadric and polynomial interpolants (upper curve $d_\infty = ||.||_\infty$, lower curve $d_2 = ||.||_2/\sqrt{N}$); (right) curvature ($\kappa = 2H$) computed at the centroid: — multiquadric; $-\cdot-$ thin plate spline; $--$ polynomial.

interpolating surfaces: (left) the polynomial surface computed by the algorithm of [1] and (right) the TPS surface. The lower frames give the contours of the MQ interpolants for $c = 10.0$ (left) and $c = 0.4$ (right). There is a close correspondence between the upper and lower frames on each side, but a large difference between the polynomial and TPS surfaces.

Figure 2 (left) shows the difference between the MQ surface and the polynomial reference interpolant computed on a regular grid on the interior of the circle with centre at the centroid of the data points $(0.44, -0.09)$ and radius the maximum distance from the centroid to a data point. There is convergence of the MQ surface to the polynomial as $1/c \to 0$, but the condition number of the interpolation matrix increases until the calculation cannot be continued. For $c = 10.0$ the condition number is 3×10^7.

As an indication of the behaviour of first and second partial derivatives of the interpolating surfaces we calculate the curvature at the centroid of the data points for the polynomial and TPS, together with MQ as c varies, using $\kappa = 2H$ where H is the mean curvature. Figure 2 (right) shows that κ_{MQ} for the MQ interpolant coincides with the value $\kappa_{TPS} = -0.46$ for the TPS when $c \approx 0.4$. When $c < 0.4$, $\kappa_{MQ} < \kappa_{TPS}$, while $\kappa_{MQ} \to \kappa_P = -9.78$, the polynomial curvature, as c increases.

An interesting example is presented in [1] of polynomial interpolation for points located at the vertices of a regular hexagon

$$(x_i, y_i) = \left(\cos\left(\frac{2\pi i}{6}\right), \sin\left(\frac{2\pi i}{6}\right)\right), \qquad i = 1, \ldots, 6 \tag{3.1}$$

with data values $f_i = (-1)^i$. This gives the interpolant

$$p(x, y) = x^3 - 3xy^2. \tag{3.2}$$

Since the points lie on the unit circle, the quadratic polynomial

$$p_2(x, y) = 1 - x^2 - y^2$$

vanishes at the data points and this causes difficulties for general polynomial methods. MQ interpolants do not suffer from these difficulties. When $c = 10.0$, the MQ surface is very close to (3.2). As c becomes smaller, the MQ surface approaches that of the TPS with the data values becoming local maxima or minima as the surface is tensioned. In addition, the restriction of the data points to a circle implies that the interpolating polynomial is harmonic, but the convergence of the approximation is only first order [1]. The MQ surface for large c inherits the properties of the polynomial fit. Thus, points on a circle are 'good' if the data being interpolated correspond to a harmonic function, but 'bad' if the data describe a function which has a maximum or minimum within the circle or a singularity. These constraints on the interpolant are discussed further in Section 5.

4 Scattered data on the sphere

In this section we examine the accuracy obtained from three separate methods for interpolating scattered data on the unit sphere $S^2 \subset \Re^3$. In particular we compare the results obtained using the MQ basis function in $\Re^3$ with those obtained using the spherical harmonics of Sloan and Womersley[1] [6] and the C^1 Hermite interpolant of Renka [5]. For the multiquadric function, we list the uniform norm interpolation errors calculated using a range of values for the shape parameter c.

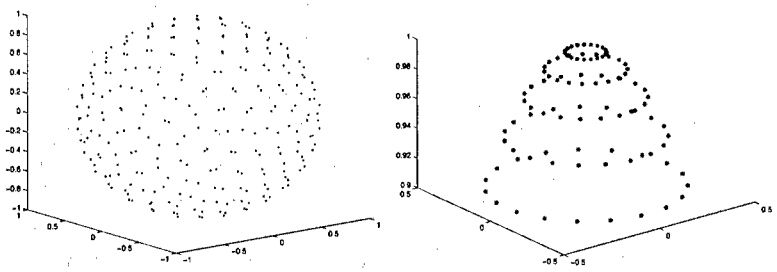


FIG. 3. Minimum energy points and spherical cap.

The point distribution used is the 256 'minimum energy' points of Fliege and Maier [3] and the uniform norm interpolation errors are calculated at points distributed on a spherical cap (see [5]).

The following functions are used for the comparisons in Table 4, where the results presented in [7] are labelled 'W&S'.

$$\begin{aligned} \text{F1} &= \tfrac{1}{10}e^{x+y+z}, & \text{F2} &= -5\sin(1+10z), \\ \text{F3} &= \|\mathbf{x}\|_1/10, & \text{F4} &= \sin^2(1+\|\mathbf{x}\|_1)/10. \end{aligned}$$

We note from Table 4 that the multiquadric function provides consistently better interpolants to the four test functions compared with the spherical harmonics. Here, the

[1]Uniform norm errors used for comparison are approximate only and were taken from graphical representations presented in Womersley and Sloan [7].

Method	F1	F2	F3	F4
W & S	2.0000e-10	0.5000	0.1100	0.0500
Renka	0.0013	0.1951	0.0054	0.0055
MQ c = 0.01	6.0128e-04	0.3276	0.0051	0.0051
MQ c = 1	4.5807e-10	0.0175	0.0076	0.0062
MQ c = 2	2.2615e-13	0.0227	0.0079	0.0065

TAB. 1. Comparison of uniform norm errors.

points have been chosen to minimise the interpolation errors for the harmonic functions, yet we see from results given in [7] that increasing the number of points in the distribution (which also increases the degree of the interpolating function) does not necessarily produce better accuracy. However, these point distributions when used for the multiquadric function provide consistently better accuracy. Further evidence suggests that points considered optimal for the spherical harmonics are also 'good' for the multiquadric function when compared to an equal number of generally scattered points. However, this is due to the uniformity of the point distributions and similar results can be obtained on a refined icosahedral mesh.

Method	12 pts	92 pts	362 pts
Renka	0.1730	0.0103	0.8230e-03
MQ c = 0.01	0.2596	0.0170	0.0020
MQ c = 1	0.0715	7.7662e-05	1.9678e-10
MQ c = 2	0.0442	3.8206e-05	3.4113e-11

TAB. 2. Multiquadric vs Renka for $f(x,y,z) = \sin(x+y) + \sin(xz)$.

The Renka algorithm produced similar results to those obtained using the multiquadric (for small c) for the F3 and F4 functions, although the results for the functions F1 and F2 were poor. Further comparisons with the Renka algorithm have been made using 12, 92 and 362 icosahedral points to interpolate the function $f(x,y,z) = \sin(x+y)+\sin(xz)$. The uniform norm interpolation errors have been calculated on the previously mentioned spherical cap. Again we see that the multiquadric function produces better accuracy than the Renka method when the number of interpolation points is increased.

5 Evolution of a smooth closed surface

In this section we return to the local interpolation scheme of §3 and apply it to scattered data on a smooth closed surface. This is the setting described in [8], where initially the interface is spherical with the point locations determined by subdivision of an icosahedral mesh. Each set of points consists of a central point together its nearest neighbours, giving sets of 6 points associated with the 12 vertices of the icosahedron and sets of 7 points otherwise. The local method of Renka [5] is followed and, for a chosen point, a local coordinate system is defined with this point on the z-axis. The local point set

is projected onto the xy-plane and the surface heights provide the data values. This typically gives a configuration very close to the hexagon points (3.1) with an additional point at the centre, except for those points associated with the icosahedron vertices where the arrangement is a pentagon. As the iscosahedral mesh is refined these configurations become less regular.

The addition of the central point to the hexagon points increases the order of the approximation. When the surface is spherical, the symmetry of the data ensures that the computed unit normal at the centre point for polynomial, MQ or TPS is exact except for rounding error (*e.g.* for MQ the error is $||n - n_{MQ}||_2 = 3 \times 10^{-14}$). However, taking MQ with $c = 10$ and a sphere of radius 9, if the central point is displaced from the origin to (0.01,0.01) the error in the normal is 3×10^{-3}. To illustrate convergence for an irregular point set, the hexagon points are perturbed by the addition of a factor $(i-1)\epsilon h[1,1]^T$ for points $i = 1, 2 \ldots, 6$ with h the radius of the circumcircle and taking $\epsilon = 0.05$. For MQ with $c = 10$, the error in the surface normal is $O(h^3)$ whereas, for $c = 0.4$, the error is larger and the rate of convergence varies (see Table 3).

h	$\|\|n - n_{MQ}\|\|_2$, $c = 10$	$\|\|n - n_{MQ}\|\|_2$, $c = 0.4$
1.0	3.15×10^{-5}	6.14×10^{-3}
0.5	3.85×10^{-6}	1.26×10^{-3}
0.1	3.06×10^{-8}	6.35×10^{-6}
0.05	4.99×10^{-9}	8.47×10^{-7}

TAB. 3. Error in MQ approximation to surface normal of sphere, irregular point set.

Accurate curvature values are essential for an interface which is driven by surface tension. The exact value of $\kappa = -2/9$ for a sphere of radius 9, together with the computed values, are shown in Table 5. The polynomial and MQ with $c = 10$ are close to the exact value.

Method	κ
exact	-0.222...
polynomial	-0.222912
MQ c = 0.1	-1.638002
MQ c = 10.0	-0.225387

TAB. 4. Curvature, $\kappa = 2H$, evaluated at the central point of a regular hexagon.

It is found that, for the icosahedral mesh with $N = 362$, the local point sets are sufficiently regular to give good accuracy for surface normals and curvature using MQ interpolants when c is chosen to be 'large' in relation to the point spacing. This mesh also gives a corresponding accuracy for the discretised integral equation. These points can thus be considered 'good' for the MQ approximation. However, if the mesh is further refined or the surface deforms during its evolution, then the approximation becomes

'less good' as the regularity of the point locations is lost. Numerical experiments suggest second order convergence with point separation for irregular local point sets.

6 Conclusions

The behaviour of MQ and TPS interpolants can be interpreted by reference to the corresponding 'least' polynomial interpolant, with the MQ connecting the polynomial C^∞ surface to the tensioned surface of the TPS as the parameter c decreases. The MQ interpolant with 'large' c (relative to the point separation) exhibits the properties of the polynomial case and is similarly affected by the location of data points. Thus, points on a circle in the plane can be 'good' if the function to be represented is harmonic, but in general give only first order convergence on the interior. For data on the sphere, 'good' points for polynomial interpolation are also good for the MQ with 'large' c, but other near equispaced point distributions appear to give similar accuracy with MQ. The tensioning effect of smaller values of c can improve the results if the underlying function is not C^∞. When applied to an evolving interface, starting from an initially spherical shape and a refined icosahedral point distribution, it is found that local MQ approximations to the surface derivatives are affected by the point locations. This can be understood by reference to the polynomial interpolant to data located on a circle and causes an irregularity in the convergence as N increases.

Bibliography

1. C. de Boor and A. Ron, Computational aspects of polynomial interpolation in several variables, *Math. Comp.* **58** (1992), 705–727.
2. M. Eck, MQ-curves are curves in tension, in *Mathematical Methods in Computer Aided Geometric Design II*, T. Lyche and L. L. Schumaker (eds), Academic Press, 1992, 217–228.
3. J. Fliege and U. Maier, The distribution of points on the sphere and corresponding cubature formulae, *IMA J. Num. Anal* **19** (1999), 317–334.
4. A. Iske, Perfect centre placement for radial basis function methods, preprint (1999).
5. R. J. Renka, Interpolation of data on the surface of a sphere, *ACM Trans. Math. Softw.* **10** (1984), 417–436.
6. I. H. Sloan and R. S. Womersley, The search for good polynomial interpolation points on the sphere, in *Numerical Analysis 1999*, D. F. Griffiths and G. A. Watson (eds), Chapman and Hall, 2000, 211–229.
7. R. S. Womersley and I. H. Sloan, How good can polynomial interpolation on the sphere be? preprint (1999).
8. A. Z. Zinchenko, M. A. Rother and R. H. Davis, A novel boundary-integral algorithm for viscous interaction of deformable drops, *Phys. Fluids* **9** (1997), 1493–1511.

Chapter 5

Regression

Preceding Page Blank

Generalised Gauss-Markov regression

Alistair B Forbes, Peter M Harris and Ian M Smith

National Physical Laboratory, Teddington, Middlesex, TW11 0LW, UK.
alistair.forbes@npl.co.uk, peter.harris@npl.co.uk, ian.smith@npl.co.uk

Abstract

Experimental data analysis is an key activity in metrology, the science of measurement. It involves developing a mathematical model of the physical system in terms of mathematical equations involving parameters that describe all the relevant aspects of the system. The model specifies how the system is expected to respond to input data and the nature of the uncertainties in the inputs. Given measurement data, estimates of the model parameters are determined by solving the mathematical equations constructed as part of the model, and this requires developing an algorithm (or estimator) to determine values for the parameters that best explain the data. In many cases, the parameter estimates are given by the solution of a least-squares problem. This paper discusses how various uncertainty structures associated with the measurement data can be taken into consideration and describes the algorithms used to solve the resulting regression problems. Two applications from NPL are described which require the solution of generalised distance regression problems: the use of measurements of primary standard natural gas mixtures to estimate the composition of a new natural gas mixture, and the analysis of calibration data to estimate the effective area of a pressure balance.

1 Introduction

Many metrology experiments involve determining the behaviour of a response variable y as a function of a set of independent variables $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$. Model building involves establishing the functional relationship between these quantities, usually involving a set of model parameters $\mathbf{a}$, i.e.,

$$y^* = \phi(\mathbf{x}^*, \mathbf{a}),$$

where y^* and $\mathbf{x}^*$ represent exact values of the variables. The terms $\mathbf{a}$ parametrize the range of possible response behaviour and the actual behaviour is specified by determining values for these parameters from measurement data. In practice, measurements are subject to error, and the error structure must be taken into account firstly in order to determine effective methods for obtaining parameter estimates and secondly in determining the uncertainty in the fitted model parameters. For a set of measurement data $\{\mathbf{x}_i, y_i\}_{i=1}^m$, the data analysis problem involves the accurate estimation of the parameters $\mathbf{a}$, taking into account knowledge of the uncertainties in $\{\mathbf{x}_i\}$ and/or $\{y_i\}$, and typically leads to a least-squares problem [4].

This paper describes the various uncertainty structures that arise and corresponding regressions problems for determining estimates of the model parameters. If the covari-

ance information associated with the measurements is structured so that only the ith set of measurement errors are correlated with each other, a generalised distance regression approach is appropriate. However, some applications have quite general correlation structure and a full Gauss-Markov estimation approach is required to make efficient use of the statistical model [7]. This leads to a *generalised Gauss-Markov regression* problem to take into account the errors in the variables and the general correlation structure. While the covariance structure may dictate which solution algorithms are to be employed, the information required of the model function ϕ is limited to the evaluation of the function and its derivatives with respect to $\mathbf{a}$ and $\mathbf{x}$. This means that solution algorithms can be based on a compact set of model-dependent modules and a generic set of harnessing routines that link the models to general purpose least-squares optimisation software.

The layout of the paper is as follows. In Section 2 we consider the various error structures and corresponding regression problems. Section 3 introduces two measurement problems encountered at NPL: the use of measurements of primary standard natural gas mixtures to estimate the composition of a new natural gas mixture; and the analysis of calibration data to estimate the effective area of a pressure balance. Although the functional models for these measurement systems are simple, taking the form of low-order polynomials, the statistical models need to account for (a) uncertainties in both the dependent and independent variables, and (b) possible correlations between measurements. These requirements lead us to solve generalised regression problems. An overview of solution algorithms for the various problems is given in Section 4. Concluding remarks are made in Section 5.

2 Error structures and regression problems

Within metrology, various error structures arise all of which can be taken into account. We now consider the main types.

2.1 Error in one variable only

2.1.1 *Ordinary (weighted) least squares*

The simplest type of error structure occurs when only one of the system variables is subject to error and there is no correlation between errors. The model is summarised by

$$y_i^* = \phi(\mathbf{x}_i^*, \mathbf{a}), \quad y_i = y_i^* + \epsilon_i, \quad \mathbf{x}_i = \mathbf{x}_i^*,$$

where it is assumed that

$$E(\epsilon_i) = 0, \quad \mathrm{var}(\epsilon_i) = \sigma_i^2, \quad \mathrm{cov}(\epsilon_i, \epsilon_j) = 0, \quad i \neq j. \tag{2.1}$$

Good estimates of $\mathbf{a}$ can be found by solving

$$\min_{\mathbf{a}} \sum_{i=1}^{m} w_i^2 [y_i - \phi(\mathbf{x}_i, \mathbf{a})]^2,$$

where $w_i = 1/\sigma_i$, $i = 1, \ldots, m$.

2.1.2 *Gauss-Markov regression*

If instead of (2.1), the measurement errors are correlated so that

$$E(\boldsymbol{\epsilon}) = \mathbf{0}, \quad \text{var}(\boldsymbol{\epsilon}) = V,$$

with V full rank, then an estimate of $\mathbf{a}$ can be found by solving

$$\min_{\mathbf{a}} [\mathbf{y} - \boldsymbol{\phi}(\mathbf{a})]^{\mathrm{T}} V^{-1} [\mathbf{y} - \boldsymbol{\phi}(\mathbf{a})], \tag{2.2}$$

where the ith element of $\boldsymbol{\phi}(\mathbf{a})$ is $\phi(\mathbf{x}_i, \mathbf{a})$.

2.2 Errors in more than one variable

In many metrological applications more than one of the measured variables is subject to error, and this must be taken into account in order to determine estimates of the model parameters which are statistically efficient and free from major bias.

2.2.1 *Orthogonal distance regression*

The simplest case arises when the covariance matrix associated with the ith set of measurements is a multiple of the identity matrix and there is no correlation between any of the errors, summarised by the model

$$y_i^* = \phi(\mathbf{x}_i^*, \mathbf{a}), \quad y_i = y_i^* + \epsilon_i, \quad \mathbf{x}_i = \mathbf{x}_i^* + \boldsymbol{\delta}_i,$$

with

$$E(\boldsymbol{\eta}_i) = \mathbf{0}, \quad \text{var}(\boldsymbol{\eta}_i) = \rho_i^2 I, \tag{2.3}$$

where $\boldsymbol{\eta}_i = (\epsilon_i, \boldsymbol{\delta}_i^{\mathrm{T}})^{\mathrm{T}}$. In this case, appropriate estimates of the parameters are determined by the solution of

$$\min_{\{\mathbf{x}_i^*\}, \mathbf{a}} \sum_{i=1}^{m} v_i^2 \{(\mathbf{x}_i - \mathbf{x}_i^*)^{\mathrm{T}} (\mathbf{x}_i - \mathbf{x}_i^*) + (y_i - \phi(\mathbf{x}_i^*, \mathbf{a}))^2\},$$

where $v_i = 1/\rho_i$, $i = 1, \ldots, m$.

Note that this optimisation problem involves m sets of parameters $\mathbf{x}_i^*$ as well as the parameters $\mathbf{a}$ specifying the model $y = \phi(\mathbf{x}, \mathbf{a})$.

2.2.2 *Generalised distance regression*

If we assume that the errors $\boldsymbol{\eta}_i$ are correlated with $\text{var}(\boldsymbol{\eta}_i) = V_i$ with V_i full rank, but that $\text{cov}(\boldsymbol{\eta}_i, \boldsymbol{\eta}_j) = \mathbf{0}$, $i \neq j$, then the appropriate regression problem is

$$\min_{\{\mathbf{x}_i^*\}, \mathbf{a}} \sum_{i=1}^{m} \begin{bmatrix} y_i - \phi(\mathbf{x}_i^*, \mathbf{a}) \\ \mathbf{x}_i - \mathbf{x}_i^* \end{bmatrix}^{\mathrm{T}} V_i^{-1} \begin{bmatrix} y_i - \phi(\mathbf{x}_i^*, \mathbf{a}) \\ \mathbf{x}_i - \mathbf{x}_i^* \end{bmatrix}. \tag{2.4}$$

2.2.3 *Generalised Gauss-Markov regression*

The most complicated error structure arises when all variables are subject to measurement error and there is general correlation between the errors. If $\boldsymbol{\xi}$ ($\boldsymbol{\xi}^*$) is the vector of measurements $\{\mathbf{x}_i\}$ (variables $\{\mathbf{x}_i^*\}$), then the corresponding regression problem is

$$\min_{\boldsymbol{\xi}, \mathbf{a}} \begin{bmatrix} \mathbf{y} - \boldsymbol{\phi}(\boldsymbol{\xi}, \mathbf{a}) \\ \boldsymbol{\xi} - \boldsymbol{\xi}^* \end{bmatrix}^{\mathrm{T}} V^{-1} \begin{bmatrix} \mathbf{y} - \boldsymbol{\phi}(\boldsymbol{\xi}, \mathbf{a}) \\ \boldsymbol{\xi} - \boldsymbol{\xi}^* \end{bmatrix}, \tag{2.5}$$

where the ith element of $\boldsymbol{\phi}(\boldsymbol{\xi}, \mathbf{a})$ is $\phi(\mathbf{x}_i^*, \mathbf{a})$.

3 Examples from metrology

3.1 Preparation of primary standard natural gas mixtures

Within the Centre for Optical and Analytical Measurement at NPL, one part of the work of the Environmental Standards Group is to prepare primary standard natural gas mixtures. These are cylinders containing natural gas prepared gravimetrically to contain known compositions of each of the 11 constituent components (methane, ethane, propane, l-butane, n-butane, l-pentane, n-pentane, neo-pentane, hexane, nitrogen and carbon dioxide). Mixtures are prepared to cover various concentration ranges, e.g., methane: $64\% - 98\%$. These primary standard mixtures are used as the basis for determining the composition of a new mixture and hence its calorific value.

Given a number of primary standard natural gas mixtures containing known concentrations of one of the constituent components (e.g., CO_2), the detector response for each mixture and the detector response for the new mixture, we wish to determine the concentration of CO_2 in the new mixture.

An approach to solving this problem is firstly to use the calibration data (relating to the primary gas mixtures) to calibrate the detector and, secondly, to use the calibration curve so constructed with the new measurement to predict the concentration in the new mixture.

Errors to be accounted for are:

- the calibration data is known inexactly. The process of preparing the primary standards means that they are known inexactly, and indeed the errors in the standards may be correlated (this is a consequence of the gravimetric process used to prepare the standard mixtures which involves comparing on a balance each standard mixture at each stage of preparation against calibrated masses selected from a common set of masses),
- the data returned by the detector (which is based on the analytical technique of chromatography) is subject to measurement error.

Consequently, we wish our data analysis to account for the inexactness of the measurement data and to quantify the resulting uncertainty associated with the final measurement result.

Figure 1 shows a sample set of measurement data, with the ellipses around the calibration points illustrating the errors in the concentrations and detector responses. (The error ellipses have been magnified greatly for illustrative purposes.) The figure also shows a calibration curve which is used to estimate the concentration of the component for which the detector response (and its uncertainty) is known.

3.2 Calibration of pressure balances

The principal role of the Pressure and Vacuum Section in the Centre for Mechanical and Acoustical Metrology at NPL is the development and maintenance of primary measurement standards for pressure and vacuum and their dissemination to industry. Pressure balances are pressure generators and consist essentially of finely-machined pistons moun-

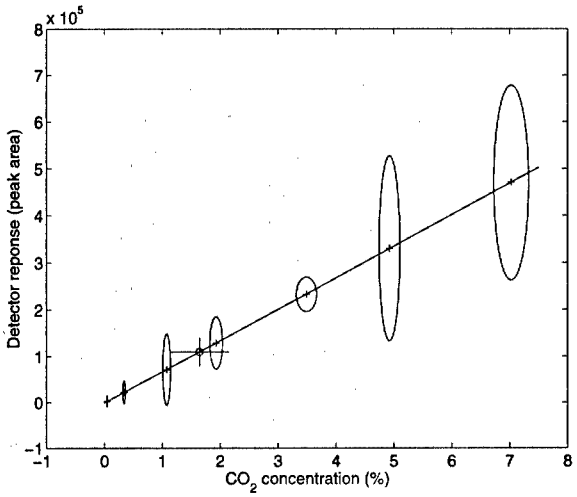


FIG. 1. Sample data (+), fitted calibration curve and predicted measurement (o).

ted vertically in close-fitting cylinders. The pressure required to support a piston and associated ring-weights depends on the mass of the piston and ring-weights and the cross-sectional area of the piston [5]. Due to various fluid dymamic effects, the effective area $A(p, \mathbf{a})$ of the piston-cylinder assembly is a function of pressure, usually taken to be a linear function $A(p, \mathbf{a}) = a_1 + a_2 p$. Many other factors such as temperature and air buoyancy have to be taken into account but for our purposes here, the pressure generated satisfies

$$a_1 p + a_2 p^2 = y(m),$$

where $\mathbf{a}$ are the instrument parameters and $y(m)$ is a simple function of the applied load m. This equation determines p implicitly as a function of m and $\mathbf{a}$. Suppose a reference pressure balance has been calibrated so that estimates of the instrument parameters $\mathbf{a}$ and their uncertainties are known. The reference balance can be used to calibrate a test balance in a cross-floating experiment in the following way. A load m_i is applied to the reference balance to generate pressure $p_i = p(m_i, \mathbf{a})$. A load n_i is applied to the test balance so that the pressures generated are matched. The test calibration curve is determined from a best fit to the data (n_i, p_i)

$$b_1 p_i^* + b_2 (p_i^*)^2 = y(n_i^*), \quad p_i = p_i^* + \epsilon_i, \quad n_i = n_i^* + \epsilon_i,$$

where δ_i and ϵ_i represent measurement error associated with the pressures and masses, respectively. However, the following must be taken into account. Firstly, the pressures p_i all depend on the common estimates $\mathbf{a}$ of the instrument parameters of the reference balance, leading to correlation of the measurement errors δ_i. Secondly, the masses n_i and m_i are made up from the same ensemble of masses $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_M)^{\mathrm{T}}$ so that

$$n_i = \mathbf{n}_i^{\mathrm{T}} \boldsymbol{\mu}, \quad m_i = \mathbf{m}_i^{\mathrm{T}} \boldsymbol{\mu},$$

where $\mathbf{n}_i$ and $\mathbf{m}_i$ are binary coefficient vectors. This means that measurement errors associated with the masses μ_k give rise to (further) correlation between δ_i and ϵ_i. Taking this general correlation into account, estimates of the the instrument parameters $\mathbf{b}$, are found from solving

$$\min_{\mathbf{b},\mathbf{p}^*} \begin{bmatrix} \mathbf{y}-\boldsymbol{\phi} \\ \mathbf{p}-\mathbf{p}^* \end{bmatrix}^{\mathrm{T}} V^{-1} \begin{bmatrix} \mathbf{y}-\boldsymbol{\phi} \\ \mathbf{p}-\mathbf{p}^* \end{bmatrix}, \tag{3.1}$$

where the ith elements of $\boldsymbol{\phi}$ and $\mathbf{y}$ are $b_1 p_i^* + b_2 (p_i^*)^2$ and $y(n_i)$, respectively, and V is the appropriate covariance matrix determined from the dependence of $\mathbf{y}$ and $\boldsymbol{\phi}$ on $\mathbf{a}$ and $\boldsymbol{\mu}$. This is a generalised Gauss-Markov regression problem.

4 Algorithms for generalised regression

Algorithms for ordinary least squares problems of the form $\min_{\mathbf{a}} \sum_i f_i^2(\mathbf{a})$ are well known and include QR factorisation methods for linear models or the Gauss-Newton algorithm for non-linear models; see, e.g., [2, 6]. The latter algorithm requires the user to supply a software module to evaluate the vector of function values $\mathbf{f}(\mathbf{a})$ and the Jacobian matrix J of partial derivatives

$$J_{ij} = \frac{\partial f_i}{\partial a_j}.$$

If $f_i(\mathbf{a}) = y_i - \phi(\mathbf{x}_i, \mathbf{a})$ as considered above, the user has to supply a module to calculate $\phi(\mathbf{x}, \mathbf{a})$ and $\partial\phi/\partial a_j$.

If V is symmetric and strictly positive definite, the Gauss-Markov regression problem (2.2) can be formulated as an ordinary least squares problem. If $V = LL^{\mathrm{T}}$ where L is lower-triangular, then the problem becomes

$$\min_{\mathbf{a}} \tilde{f}_i^2(\mathbf{a}),$$

where $\tilde{\mathbf{f}} = L^{-1}\mathbf{f}$. The associated Jacobian matrix is $\tilde{J} = L^{-1}J$. If the matrix V is well-conditioned, matrix operations with V or L^{-1} should not lead to unnecessary loss of precision. However, explicit calculations involving V can be avoided by using the generalised QR factorisation [2, 8, 9], leading to solution algorithms with good numerical properties.

The generalised distance regression problem (2.4) can be solved efficiently by making use of the fact that the parameters $\mathbf{x}_i^*$ appear only in the ith summand. The associated Jacobian matrix has a block-angular structure that can be exploited effectively in the QR factorisation stage [2, 3]. Alternatively, a separation-of-variables approach can be adopted in which the parameters $\mathbf{x}_i^*(\mathbf{a})$ are first determined as functions of $\mathbf{a}$ specified by the solution of the corresponding footpoint problem

$$\min_{\mathbf{x}_i^*} \begin{bmatrix} y_i - \phi(\mathbf{x}_i^*, \mathbf{a}) \\ \mathbf{x}_i - \mathbf{x}_i^* \end{bmatrix}^{\mathrm{T}} V_i^{-1} \begin{bmatrix} y_i - \phi(\mathbf{x}_i^*, \mathbf{a}) \\ \mathbf{x}_i - \mathbf{x}_i^* \end{bmatrix}$$

and the problem formulated as a non-linear least squares problem in $\mathbf{a}$ [1, 4]. Either approach yields an algorithm requiring $O(mn^2)$ flops while a full matrix approach requires $O(m^3)$ flops.

The generalised Gauss Markov problem (2.5) can be solved as a Gauss-Markov problem problem in the variables $\{\mathbf{x}_i^*\}$ and $\mathbf{a}$, but ideally, we would like to develop algorithms that exploit problem structure as in generalised distance regression algorithms. In particular, while the covariance matrix V may well be full, in many situations it is constructed from smaller matrices and for which more efficient algorithms could be developed.

From the user's point of view, all the regression algorithms discussed here require only the calculation of the model function ϕ and its derivatives $\frac{\partial \phi}{\partial x_k}$ and $\frac{\partial \phi}{\partial a_j}$. Thus, a wide range of regression problems can be solved using standard optimisation modules along with generic harness modules that perform the conversion without input from the user over and above the calculation of ϕ and its derivatives. For example, we have implemented a generalised Gauss-Markov solver to solve problems such as (3.1) for any explicit model $y = \phi(x, \mathbf{a})$. However, issues of efficiency and numerical stability need to be taken into account. As part of the UK Department of Trade and Industry's Software Support for Metrology programme, NPL is developing and making available to metrologists a suite of routines for the generalised regression problems discussed above. By combining structure exploiting linear algebra and numerically stable components such as the orthogonal factorisation, it is hoped that metrologists will be able to use these routines with the same confidence and effectiveness that they currently experience with standard, well-engineered regression modules available in numerical libraries.

5 Concluding remarks

In metrology, we are interested in the determination of accurate estimates of the parameters that describe a physical process. It is imperative that knowledge of the measurement system should be used to describe the error structure as accurately as possible. We have described the five types of regression problems that can occur in metrology depending on the error structures that are assumed. In all cases it is important that we employ efficient, numerically stable algorithms and exploit any structure in both the Jacobian and covariance matrices.

Acknowledgements. This work has been supported by the Department of Trade and Industry's National Measurement System Software Support for Metrology Programme and undertaken by a project team at the Centre for Mathematics and Scientific Software, National Physical Laboratory. The authors are particularly grateful to Paul Holland (Centre for Optical and Analytical Measurement) and the Pressure and Vacuum Section for their contributions.

Bibliography

1. M. Bartholomew-Biggs, B. P. Butler, and A. B. Forbes, Optimisation algorithms for generalised distance regression in metrology, in *Advanced Mathematical and Computational Tools in Metrology IV*, P. Ciarlini, A. B. Forbes, F. Pavese and D. Richter (eds), 21–31, World Scientific, Singapore, 2000.
2. A. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

3. M. G. Cox, The least-squares solution of linear equations with block-angular observation matrix, in *Advances in Reliable Numerical Computation*, M. G. Cox and S. Hammerling (eds), 227–240, Oxford University Press, 1989.
4. M. G. Cox, A. B. Forbes, and P. M. Harris, *Software Support for Metrology Best Practice Guide 4: Modelling Discrete Data*, National Physical Laboratory, Teddington, 2000.
5. A. B. Forbes, and P. M. Harris, Estimation algorithms in the calculation of the effective area of pressure balances, *Metrologia*, 36(6): 689–692, 1999.
6. G. H. Golub and C. F. Van Loan, Matrix Computations, John Hopkins University Press, Baltimore, third edition, 1996.
7. K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis*, Academic Press, London, 1979.
8. C. C. Paige, Fast numerically stable computations for generalized least squares problems, *SIAM J. Numer. Anal.*, 16:165–171, 1979.
9. SIAM, Philadelphia, *The LAPACK Users' Guide*, third edition, 1999.

Nonparametric regression subject to a given number of local extreme value

Ali Majidi and Laurie Davies

Department of Mathematics and Computer Science, University of Essen, Germany.

{ali.majidi,laurie.davies}@stat-math.uni-essen.de

Abstract

We consider the problem of nonparametric regression. The aim is to get a smooth function which represents the dataset and has a reasonable number of extreme values. An iterative method, the QSOR method is introduced. Problems with the slow convergence of the method are reduced using multigrid techniques.

1 Introduction

Given a dataset $\{y(t_i), i = 1, \ldots, n\}$ which we denote by y, we look for a decomposition

$$y(t_i) = f(t_i) + r(t_i), (t_i = i/n, i = 1, \ldots, n)$$

where f is a simple function and the $\{r(t_i), (i = 1, \ldots, n)\}$ are the resulting residuals which approximate white noise. We use two different concepts of simplicity. The first is the number of local extreme values. The second is the smoothness of the function as measured by the standard smoothness functional

$$S(f) := \int_0^1 (f^{(2)}(t))^2 dt,$$

where $f^{(2)}$ is the second derivative of f. The number of local extremes is taken to have priority over smoothness. The number of local extremes and their locations are determined by the taut string method developed in [3]. This is described briefly in the next section. The residuals are required to look like white noise in the sense that the means over certain dyadic intervals are required to lie within given bounds [3]. The multiresolution coefficients for $(n = 2^\nu)$ are defined by: $w_{ij} := 2^{(-i/2)} \sum_{k=j2^i+1}^{(j+1)2^i} r(t_k), (i = 0, \ldots, \nu), (j = 0, \ldots, 2^{(\nu-i)} - 1)$. The multiresolution condition now requires that $-c_n \leq w_{ij} \leq c_n$, where c_n represents some form of thresholding. The default value of c_n which we use is $c_n = \sigma_n \sqrt{2.5 \log(n)}$ where $\sigma_n = 1.482 \cdot \text{median}(|y_2 - y_1|, \ldots, |y_n - y_{n-1}|)/\sqrt{2}$.

Supported by SFB 475, University of Dortmund

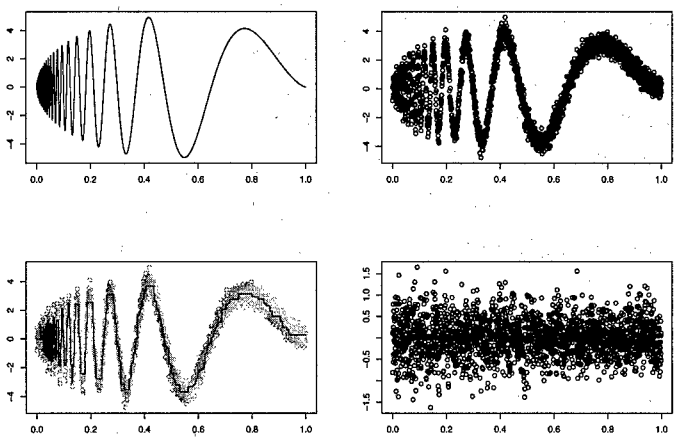


FIG. 1. The top-left caption shows the original doppler function and the top-right caption shows the noisy version. The bottom-left caption shows the result of the taut string algorithm with the resulting residuals being shown in the bottom-right caption.

2 Taut string

A short description of the taut string method is as follows. We write $f = (f_1, \ldots, f_n)^T := (f(t_1), \ldots, f(t_n))^T \in \mathbf{R}^n$ and denote the cumulative sums of y and f by Y and F respectively, $Y_i = \sum_{j=1}^{i} y_j$, $F_i = \sum_{j=1}^{i} f_j, (i = 0, \ldots, n)$, with $Y_0 = F_0 = 0$. We specify bounds defined by $\lambda = (\lambda_1, \ldots, \lambda_n)^T \in \mathbf{R}_+^n$ and consider the tube

$$\{G : |Y - G| \leq \lambda\}. \tag{2.1}$$

The taut string $V(\lambda)$ is now the function defined by a taut string attatched to the points $(0, Y_0)$ and (n, Y_n) and constrained to lie within the tube (2.1). It can be shown that the taut string minimizes the number of extreme values of the functions g whose cumulative sums G lie within the tube. The taut string is continuous and piecewise linear. Its derivative $v(\lambda)$ is taken as a candidate regression functions. The vector λ is determined in a data dependent manner by the requirement that the residuals associated with $v(\lambda)$ $\{r(\lambda)_i = y_i - v(\lambda)_i, i = 1, \ldots, n\}$ satisfy the multiresolution condition. If such a condition fails on an interval then the λ-values associated with that interval are reduced in size. An application of the taut string method to the doppler data of Donoho and Johnstone (see e.g. [4]) is shown in Figure 1. The function is defined by $f(t) = 21\sqrt{(v(1-t))}\sin\left(2\pi\frac{1+0.05}{t+0.05}\right)$. The derivative $v(\lambda)$ is piecewise constant as may be seen from Figure 1. The function $v(\lambda)$ determines the number of local extremes. We take the midpoints of the intervals associated with a local extremes as the locations of the local extremes for the smoothing algorithm.

3 The smoothing problem

We make the smoothing problem precise as follows. The number, locations and type of extreme values are taken from the taut string as explained in the last section. We further require the function f to lie in the tube determined by the taut string. This is to prevent the smoothing procedure from moving too far from the data. These restrictions may be described in the form

$$Af \geq b \tag{3.1}$$

for an appropriate matrix A and vector b. This leads to the following problem:

$$\text{minimize } \sum_{i=1}^{n}(f_{i+1} - 2f_i + f_{i-1})^2 \text{ subject to (3.1)},$$

or equivalently

$$\text{minimize } F^T Q_3 F \text{ subject to (3.1)},$$

for some quadratic form Q_3. We denote this latter quadratic programming problem by QP3. Clearly the matrix associated with the quadratic form $\sum_{i=1}^{n}(f_{i+1} - 2f_i + f_{i-1})^2$ is singular. Nevertheless the solution of QP3 may be unique. We have the following theorem.

Theorem 3.1 *Let $V(\lambda)$ be the result of the taut string method. Assume that $V(\lambda)$ has one extreme value. We define the bounds L, U by*

$$L := Y - \lambda, U := Y + \lambda.$$

Let $\hat{F}_1, \hat{F}_2$ be two solutions of the corresponding quadratic program. Additionally let $\hat{F}_1$ touch three bounds alternately

$$\textit{(i.e. } U_{i_1}, L_{i_2}, U_{i_3} \textit{ or } L_{i_1}, U_{i_2}, L_{i_3}, (i_1 < i_2 < i_3) \textit{ are active).}$$

Then

$$\hat{F}_1 = \hat{F}_2.$$

We call a problem with a unique solution a *nondegenerate* problem. From now on we assume that our problem is nondegenerate.

3.1 Quadratic programming

There are many algorithms which solve quadratic programming problems directly. Unfortunately most of them are expensive in terms of memory requirements and are not feasible for data sets of the order say $n = 8196$. To overcome this we look for iterative methods which converge to the solution. Gradient projection methods (e.g. as defined in [8], [2] or [9]) are not appropriate for this purpose as the monotonicity constraints make the projection into the feasible set too expensive. Instead we use a modified version of the QSOR (quasi successive over relaxation) method developed by Metzner in [7]. QSOR is a very cheap iteration and converges to the solution of QP3. Unfortunately the convergence is very slow on sections where the solution is smooth. To overcome this we use multigrid methods which have to be adapted to our requirements.

4 QSOR

The QSOR algorithm is an iterative method which produces a feasible sequence $\{F^k\}_{k=0}^{\infty}$ converging towards the solution of QP3. For simplicity, we describe the iteration only for a convexity interval. Let $F^0 \in \mathbf{R}^n$ be an arbitrary feasible vector. The obvious candidate is the derivative of the taut string. Let $Q = Q_3$ and $\omega \in (0,2)$. The following defines a QSOR iteration.

- While convergence not achieved
- $F = F^k$
  $i = 1$
  $\tilde{F}_i = F_i - \frac{\omega}{Q_{ii}}(Qf)_i, \tilde{L}_i = \max\{2F_{i+1} - F_{i+2}, L_i\}, \tilde{U}_i = U_i, \hat{F}_i = \mathrm{med}\{\tilde{L}_i, \tilde{U}_i, \tilde{F}_i\}$
  $i = 2$
  $\tilde{F}_i = F_i - \frac{\omega}{Q_{ii}}(Qz)_{ii}, \tilde{L}_i = \max\{2F_{i+1} - F_{i+2}, L_i\}$
  $\tilde{U}_i = \min\{(F_{i+1} + F_{i-1})/2, U_i\}, \hat{F}_i = \mathrm{med}\{\tilde{L}_i, \tilde{U}_i, \tilde{F}_i\}$
- for (i in 3:(n-2)){
  $\tilde{F}_i = F_i - \frac{\omega}{Q_{ii}}(Qz)_i, \tilde{L}_i = \max\{2F_{i+1} - F_{i+2}, 2F_{i-1} - F_{i-2}, L_i\}$
  $\tilde{U}_i = \min\{(F_{i+1} + F_{i-1})/2, U_i\}, \hat{F}_i = \mathrm{med}\{\tilde{L}_i, \tilde{U}_i, \tilde{F}_i\}$
  if(i active) mark i
  }
  }
  $i = n$
  $\tilde{F}_i = F_2 - \frac{\omega}{Q_{ii}}(Qz)_{ii}, \tilde{L}_i = \max\{2F_{i-1} - F_{i-2}, L_i\}, \tilde{U}_i = U_i, \hat{F}_i = \mathrm{med}\{\tilde{L}_i, \tilde{U}_i, \tilde{F}_i\}$
  $i = 1$
  $\tilde{F}_i = F_i - \frac{\omega}{Q_{ii}}(Qz)_i$
  $\tilde{L}_i = \max\{2F_{i+1} - F_{i+2}, L_i\}$
  $\tilde{U}_i = U_i \hat{F}_i = \mathrm{med}\{\tilde{L}_i, \tilde{U}_i, \tilde{F}_i\}$
- correct the active intervals:
  * Let $[F_\nu, F_{\nu+k}]$ be an active Interval: $F_i = F_\nu + \frac{t_j - t_\nu}{t_{\nu+k} - t_\nu}(F_{\nu+k} - F_\nu)$. Denoting the i-th unit vector in $\mathbb{R}^n$ by e_i and a, b defined by

$$b = \frac{\sum_{i=\nu}^{\nu+k}(Qz)_i}{\sum_{i=\nu}^{\nu+k}\sum_{j=\nu}^{\nu+k} Q_{ij}}, \quad a = \frac{\sum_{i=\nu}^{\nu+k} t_i (Qz)_i}{\sum_{i=\nu}^{\nu+k} t_i \sum_{j=\nu}^{\nu+k} t_j Q_{ij}}$$

  set $F_j^k := \hat{F}_j - \hat{\alpha}(at_j + b)$ with
- $F_i^k = \hat{F}_i$ for all i in other intervals

Theorem 4.1 *(convergence) Let $(F^k)_{k=0}^{\infty}$ be the sequence in $\mathbb{R}^n$ produced by the QSOR algorithm and let the problem QP3 be nondegenerate. Then*

- *$(F^k)_{k=0}^{\infty}$ converges in $\mathbb{R}^n$.*
- *$F^* := \lim_{k\to\infty} F^k$ is the solution of QP3.*

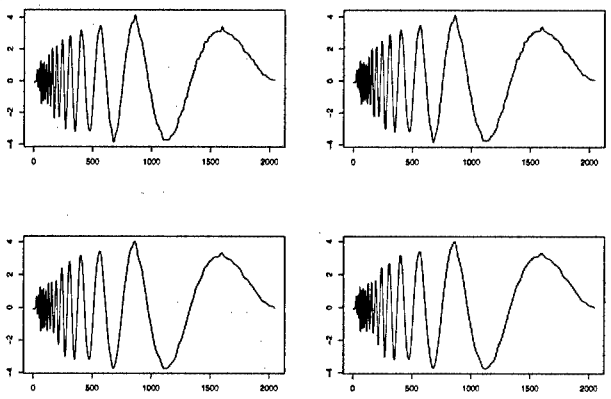


FIG. 2. The captions top-left, top-right, bottom-left, bottom-right show the result of the QSOR iteration for the doppler data ($n = 2048$) after $1000, 5000, 10000, 20000$ steps respectively.

The slowness of the convergence can be seen by the fact that the doppler data of Figure 1 required two million iterations before a satisfactory solution was obtained. This is shown in Figure 2. After a small number of iterations the solution does not change any more on the "left side" where the function oscilates rapidly. After 1000 iterations of QSOR (which is fast because one QSOR step is very cheap!) the solution looks very smooth except for a few "buckles" on the "right side" of the solution. The method needs many iterations (up to two million) to reach an adequate smoothness. The slowness of the convergence is known from the original SOR method for solving linear equations. In the standard case of solving linear equations multigrid methods can be used to speed up the rate of convergence. We now apply this idea to solving the problem QP3.

5 Multigrid QSOR

Multigrid techniques are general techniques to speed up iterative methods which indeed have other good properties. The ideas are given for example in [1] or [5]. We will give here a short description of the multgrid idea in our case. First some notation. Given a *grid* $G = G_f = (t_1, \ldots, t_n)$ we define the *coarse grid* $G_c = (t_1, t_{i_2}, \ldots, t_{i_{m-1}}, t_n), i_1 = 1, i_m = n$ with $i_j \in \{1, \ldots, n\}$. We define the projection *down* $I_c x = (F_1, F_{i_2}, \ldots, F_{i_{m-1}}, F_n)^T$ and the projection *up* $I^c x = y$ where $y_l = F_l$ $(l \in \{i_j | j = 1, \ldots, m\})$, and by linear interpolation elsewhere, i.e.,

$$y_l = \frac{F_{i_j} - F_{i_{j-1}}}{t_{i_j} - t_{i_{j-1}}} \quad (i_{j-1} < l < i_j).$$

We define now the *multigrid QSOR* with only two grids, i.e., of level two. The general case of level $\nu \in \mathbb{N}$ is defined similarly. Let $QSOR(G, A, b, \mu, x)$ denote the result of the

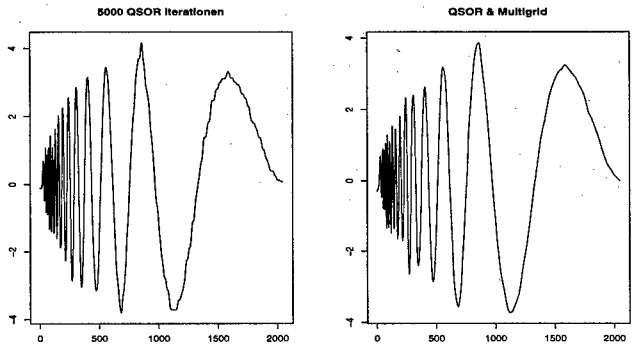


FIG. 3. The left figure shows the result of QSOR after 5000 iterations. The right figure shows the result of (1000) multigrid QSOR with one coarsing step (i.e. the right figure is "cheaper" than 2000 QSOR streps).

QSOR method applied to the problem on the grid G after μ iterations on the Grid G with starting vector x and constraints defined by A, b. Additionally let F^k be given.

- **Multigrid QSOR**
  - while precision not achieved
    - $\hat{F} = QSOR(G, A, b, \mu, F^k)$
    - $\tilde{F} = I^c QSOR(G_c, A_c, b_c, \mu, I_c \hat{F})$
    - $F^{k+1} = QSOR(G, A, b, \mu, \tilde{F})$
    - $k \leftarrow k + 1$

where A_c, b_c are the corresponding constraints for the coarser grid. The question is now how to define the *projection* of the constraints. One can think of an example where the canonical projection of bounds like G_c can fail. This happens for example if *strong* constraints (e.g. tight bounds) are not on the coarse grid. To overcome this problem one has to think of a method of defining the problem QP3 on the coarser grid in a reasonable way. One way to handle this problem is to define $L_{i_j} := \max\{L_k | i_{j-1} < k < i_{j+1}\}$ and "min" for the upper bounds. Special cases have to be treated but we do not go into details here. A coarser grid means that the QSOR step on this grid converges much faster. On the other hand the approximation of the solution gets worse by coarsening the grid. In our case (see Figure 4) we have $n = 2048$. The coarsest grid was made by taking every eighth gridpoint. We iterated until there was no recognizable improvement.

6 Proofs

Proof of Theorem 3.1: We set $D = \hat{F}_2 - \hat{F}_1$. One simply verifies that D has to be a line, i.e., there are $a, b \in \mathbf{R}$ such that $D_i = at_i + b$. Touching three bounds alternately means that D changes its sign at least two times which leads to $D = 0$. □

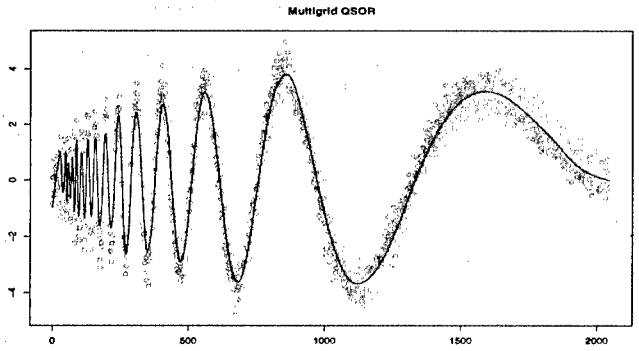


FIG. 4. Multigrid QSOR applied to the doppler data with $n = 2048$. The figure took less than 6 seconds comparing to three hours without multigrid on the same computer.

Proof of Theorem 4.1: We set $Q = Q_3$. We have to show:

1) $(S_3(F^k))_{k\in\mathbb{N}_0}$ decreases;
2) $(F^k)_{k\in\mathbb{N}}$ is feasible;
3) If F^s is a stationary point of QSOR, then F^s minimizes S_3 in the feasible set.

- our feasible set is compact, so the sequence has a convergent subsequence,
- a limit of a subsequence of $(F^k)_{k=1}^{\infty}$ is a stationary point of QSOR,
- the problem has only one solution.

To the first point, we only remark that a, b as defined in the algorithm, are the minimizers of the term:

$$\left(z - \left(x\sum_{i=\nu}^{\nu+k} t_i e_i + y\sum_{i=\nu}^{\nu+k} t_i e_i\right)\right)^T Q \left(z - \left(x\sum_{i=\nu}^{\nu+k} t_i e_i + y\sum_{i=\nu}^{\nu+k} t_i e_i\right)\right).$$

The others are treated as in [6]. The second point is clear, because by the definition we start with a feasible vector and we retain the feasibility in every single step. It remains to show the third point. Let F^s be a stationary point of the algorithm. It is sufficient to show that $\langle QF^s, Z - F^s\rangle \geq 0$ for all feasible vectors z (see [6]), where $\langle,\rangle$ denotes the standard inner product in $\mathbb{R}^n$. To show this we first note that $Q = D^T Q_2 D$, where

$$D = \begin{pmatrix} 1 & & & & & \\ -1 & 1 & & & & \\ & \ddots & \ddots & & & \\ & & & -1 & 1 & \\ & & & & -1 & 1 \end{pmatrix}$$

and Q_2 is the matrix according to QP3, i.e., to the direct problem. So we can deduce that $\langle QF^s, Z - F^s\rangle = (Z - F^s)^T QF^s = (Z - F^s)^T D^T Q_M DF^s = \langle Q_M f^s, z - f^s\rangle (f^s := DF^s, z := DZ)$. Now we only have to look at the "active points" because $(QF^s)_i$ is

zero everywhere else. Let Z be an arbitrary feasible vector and j be an index with $Z_j^s = L_j$ and $(Qz)_j \neq 0$, so $-\omega(QF^s) < 0$. With the feasibility of Z, it follows that $(QF^s)_j(Z_j - Z_j^s) = (QF^s)_j(Z_j - L_j) \geq 0$. With the same argument we can derive $(QF^s)_j(Z_j - Z_j^s) \geq 0$ if F^s touches the upper bound. It remains to show the inequality for the linearity intervals. Let $[t_\nu, t_{\nu+k}]$ be a linearity interval of F^s. Then obviously $[t_{\nu+1}, t_{\nu+k}]$ is a constancy interval for F^s. Furthermore it follows from the stationarity of F^s that a, b as defined in the algorithm are zero. This is equivalent to

$$\sum_{i=\nu}^{\nu+k}(QF^s) = 0, \qquad \sum_{i=\nu}^{\nu+k} t_i QF^s = \frac{1}{n}\sum_{i=\nu}^{\nu+k} iQF^s = 0 \quad (t_i = i/n),$$

which implies that

$$\sum_{i=1}^{l}(QX)_i = (Q_M x)_l, \qquad \sum_{i=\nu}^{l} i(QX)_i = \sum_{i=1}^{l}(Q_M x)_i$$

for arbitrary $X \in \mathbb{R}^n$ and $x = DX$. So our conditions are

$$(Q_M F^s)_\nu = 0, \qquad \sum_{i=\nu}^{\nu+k}(Q_M F^s)_i = 0 \Rightarrow \sum_{i=\nu+1}^{\nu+k}(Q_M F^s)_i = 0.$$

This case was proved by Löwendick [6]. □

Bibliography

1. William L. Briggs. *A Multigrid Tutorial.* SIAM, New York, 1994.
2. Paul H. Calamai and Jorge J. Moré. Projected gradient methods for linearly constrained problems. *Mathematical Programming*, 39:93–116, 1987.
3. P. L. Davies and A. Kovac. Modality, Runs, Strings and Multiresolution. *To appear in Annals of Statistics*, 2001.
4. D. L. Donoho and I.M. Johnstone. Ideal Spatial Adaption by Wavelet Shrinkage. *Biometrika*, 81:425–455, 1994.
5. W. Hackbush. *Multi-Grid Methods and their Applications.* Springer, Berlin, 1985.
6. M. Löwendick. On Smoothing under Bounds and Geometric Constraints. *Dissertation, Universität Essen*, 2000.
7. L. Metzner. Facettierte Nichtparametrische Regression. *Dissertation, Universität Essen*, 1997.
8. Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer, Berlin, 1999.
9. Gerardo Toraldo and Jorge J. Moré. On the solution of large quadratic programming problems with bound constraints. *SIAM J. Optimization*, 1:93–113, 1991.

Model fitting using the least volume criterion

Chris Tofallis

University of Hertfordshire Business School
Dept. of Statistics, Economics, Accounting and Management Systems
Mangrove Rd, Hertford, SG13 8QF, UK
c.tofallis@herts.ac.uk

Abstract

Given data on multiple variables we present a method for fitting a function to the data which, unlike conventional regression, treats all the variables on the same basis i.e. there is no distinction between dependent and independent variables. Moreover, all variables are permitted to have error and we do not assume any information is available regarding the errors. The aim is to generate law-like relationships between variables where the data represent quantities arising in the natural and social sciences. Such relationships are referred to as structural or functional models. The method requires that a (monotonic) relationship exists; thus, in the two variable case we do not allow cases where there is zero correlation. Our fitting criterion is simply the sum of the products of the deviations in each dimension and so corresponds to a volume, or more generally a hyper-volume. One important advantage of this criterion is that the fitted models will always be units (i.e. scale) invariant. We formulate the estimation problem as a fractional programming problem. We demonstrate the method with a numerical example in which we try and uncover the coefficients from a known data-generating model. The data used suffers from multicollinearity and there is preliminary evidence that the least volume method is much more stable against this problem than least squares.

1 On the undeserved ubiquity of least squares regression

In fitting a function to data, conventional regression requires one variable to be 'special' — this is the dependent variable. In the sciences however, one often wishes to re-arrange the model equation by changing the subject of the formula. Statisticians tell us that in that case we should carry out a second regression. Yet scientists are uncomfortable with having separate models for each variable, which are not equivalent to each other and yet are meant to represent the same relationship. Calibration is another case where one would like mutual equivalence: e.g. in psychology one can have two tests that are intended to measure the same ability: a formula or conversion table is required to relate the score on one test to that on the other.

Another case where regression is inappropriate is where one wants to deduce a parameter such as the rate of change (slope). If both variables are subject to error then ordinary least squares will under-estimate the slope, and regressing x on y will over-estimate it. A simple example involves plotting galaxy speed (or redshift) against distance from the observer. The slope of the fitted line gives what is called the Hubble constant, whose

value crucially determines the future of the universe: will it continue expanding or will it eventually begin to collapse in on itself? Conventional regression gives different values for the Hubble constant depending on which variable is treated as being dependent, but there is no apparent reason for choosing one variable as against the other.

An oft-cited reason for using least squares fitting is that under certain assumptions on the errors, it will provide the best linear unbiased estimate ('BLUE') of the slope. This is the Gauss-Markov theorem, where 'best' is taken to mean minimum variance. What is not widely appreciated is that 'linear' here refers not to the form of the fitted model, but rather that the expression for the estimated coefficient be linear in y. One can find estimators with even lower variance by removing this non-essential condition e.g. other L_p-norm estimators are not linear in y.

In multiple regression it is widely, and mistakenly, believed that that the fitted coefficients tell us the contribution that a particular variable makes to the dependent variable. In fact, not even the sign of the coefficient can be relied upon to tell us the direction of the relationship i.e. a particular x-variable may be positively correlated with the y-variable, and yet have a negative coefficient in the regression model. This is the problem of multicollinearity: if there are near-linear relations among the explanatory variables then the coefficients produced by regression will not only be highly uncertain (large standard error) but also not be open to sensible interpretation.

We shall present a technique for model-fitting which treats all variables on the same basis. The method has the important property of being units-invariant; this is an advantage not shared by the total least squares approach (also known as orthogonal regression), and arises from the fact that we use the product of the deviations in each direction rather the sum (or sum of squares) when calculating the fitting criterion.

2 The least areas criterion

Consider a set of data points in two dimensions as in Figure 1. By drawing the vertical and horizontal deviations from the line we create a right-angled triangle for each data point. Our fitting criterion is simply to minimise the sum of these areas. A key advantage of this approach is that changing the units of measurement will not affect the resulting line. In other words it is a scale invariant method. Furthermore we can add a constant to either variable and the geometry is such that the line merely gets shifted vertically or horizontally. Combining the scale and translation invariance implies that the least areas line is invariant to linear transformations of the data. It is also apparent that switching the axes has no effect: the variables are treated symmetrically. (A textbook discussion of this method appears in Draper and Smith [5].)

We note that it is essential that there be a non-zero correlation in the data otherwise the method fails. For those seeking to quantify relationships between data variables in the experimental sciences this would hardly seem to be a restrictive requirement. However for those working in the area of design and who are concerned with geometrical shapes, it does rule out the fitting to data scattered around a vertical or horizontal line, or circle, or rectangle with sides parallel to the co-ordinate axes etc.. We shall not discuss fitting curves in this paper but we note that this method is not suitable for fitting a relationship

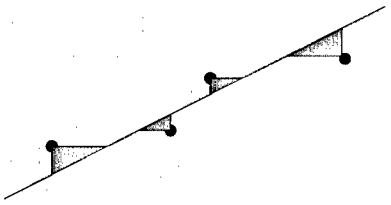


FIG. 1. Sum of areas to be minimised in least area calculation.

that is not monotone over the range of the data i.e. there cannot be maxima or minima over the data range otherwise the area deviation associated with a given point may not be uniquely specified. Such problems may be avoided by breaking up the data set into subsets at the optima and fitting a monotone function to each subset, thus producing a piecewise monotone function.

The least areas method has an interesting history, it has surfaced under different guises in diverse research literatures throughout the twentieth century. In astronomy it is known as Stromberg's impartial line. In biology it is the line of organic correlation. In economics it is the method of minimised areas or diagonal regression. In statistics it is sometimes referred to as the 'standard or reduced major axis'. This derives from the fact that if the data are standardised by dividing by their standard deviation, then the fitted line corresponds to the major (i.e. principal) axis of the ellipse of constant probability for the bivariate normal distribution. Yet another name for this technique is the geometric mean functional relationship. This is because the slope has a magnitude equal to the geometric mean of the two slopes arising from ordinary least squares (OLS) (proved in Barker, Soh and Evans [2], and Teissier [20]) i.e. if we regress y on x and get a slope b_1 and then regress x on y (so as to minimise the sum of squared deviations in the x- direction) and obtain a regression line $y = a + b_2 x$, then the geometric mean slope is $b = (b_1 b_2)^{1/2}$. It is interesting to note that the two OLS slopes are connected via the correlation between the variables

$$r^2 = \frac{b_1}{b_2}.$$

This implies that as the correlation falls the disagreement between the two OLS slopes increases; for example, even with a correlation as high as 0.71 one of these slopes will be twice as large as the other! It also follows that the magnitude of the slope of the least areas line lies between those of the two OLS lines. This is intuitively satisfying in a technique that aims to treat x and y deviations symmetrically. Specifically, for the case of positive but imperfect correlation, we have $b_2 > b > b_1$ because $b/r > b > rb$.

From the geometric mean property and the expressions for OLS slopes one can deduce that the magnitude of the slope of the least areas line takes on a particularly simple closed form: it is the standard deviation of y divided by the standard deviation of x. The sign of the slope is provided by the sign of the correlation between y and x.

Numerical experiments have been carried out to compare this fitting technique against five others (Babu and Feigelson [1]). A specified underlying model was used to generate data (mostly bivariate normal samples) and the aim was to see which method could

best recover the slope of the model. The simulations involved varying the sample size, correlation and variances. Orthogonal regression gave the poorest accuracy. There were two methods that came out with highest accuracy: the least areas method and the least squares bisector. The latter bisects the smaller angle formed between the two OLS lines. Unfortunately the OLS bisector is not units invariant and so does not suit our purposes (Ricker [17]).

Turning now to applications, the method seems to have appeared most often in the field of biometrics (the application of statistics to biological data). For example, in relating the size of one body part to another (or to the total weight or height) in humans and other animals, one may collect data from an individual at successive stages in their growth, or from many individuals at different points in their development. It is not generally possible to distinguish between dependent and independent variables in such a context. Isometric growth is the special case where the two body parts grow such that their size ratio remains constant. Miller and Kahn [13] argue in favour of our method thus: 'there is usually no clear justification for saying, e.g. that increase in skull length is dependent upon increase of body length; it is more realistic to consider changes in skull length and body length as due to a set of common factors'. Ricker [16] discusses the value of the method in fishery research. Applications include modelling relationships between weight and length, between weight and fecundity (the number of eggs), and estimating the 'catchability' of fish (the fraction of the stock taken by one unit of fishing effort). Rayner [15] gives an application to the flight speed of birds as related to the windspeed.

We have already noted the scope for application in astronomy. Babu and Feigelson [1] point out that 'differences in regression methods on similar data may be responsible for a portion of the long-standing controversy over the value of Hubble's constant, which quantifies the recession rate of the galaxies'. The earliest appearance of our method in the astronomical literature seems to be that of Stromberg [19].

The method has also been proposed in the context of educational and psychological testing. A very early reference being that of Otis [14] who named it the 'relation line'. If two tests are meant to measure the same aptitude or attainment one may need to match pairs of equivalent scores on the pair of tests for creating a conversion table. The direction of the conversion should obviously not affect which values are paired off, hence the need for a symmetric approach. Greenall [7] proposes the 'equivalence line' for this purpose:

$$\frac{y-\mu_y}{\sigma_y} = \frac{x-\mu_x}{\sigma_x}.$$

This turns out to be yet another name for our least areas line. For standardised scores the line equation reduces to $y = x$. He also proves a very interesting uniqueness result: 'When we seek a relation that will deem a pair of scores mutually equivalent if and only if the proportion of x-scores less than X equals the proportion of y-scores less than Y, we aim at pairing off scores that give rise to equal percentile ranks. In the case of continuous bivariate distributions which satisfy a simple condition $[F(x,y) = F(y/c, cx)]$, only the equivalence relation will provide this relation'. The normal distribution is one case which satisfies this condition. A relevant theoretical result due to Kruskal [12] is that if the two

variables are normally distributed and a line is needed to predict x from y as often as y from x, then the least areas line maximises the probability of correct prediction (i.e. the probability of being within z standard deviations, for any given z-value). This provides another justification for the use of this line.

Hirsch and Gilroy [9] show how it can be useful in hydrology and geomorphology where one may be interested in relationships between e.g. stream slope versus elevation, or stream length versus basin area, etc.. 'In such cases there is no clear direction of causality but there is clearly an inter-relation of variables'. 'A major motivation for the use of the line lies in the equivalence of the cumulative function of y and y_{est}'.

In general terms when should the least areas method be used? Rayner [15] cites the result of Kendall and Stuart [10] that if no error information is available then this method gives the least-bias or maximum likelihood estimate of the functional relation. Rayner goes on to demonstrate that this line also has the property of being independent of the correlation between the errors of the two variables.

Ricker [17] deals with the question of usage by first distinguishing between random measurement error and mutual natural variability (as arises for example in biology). In the former case for each observation there is an associated true point which would arise if the errors in both variables were zero. If one can estimate the variances of the errors by replicating the measurements then measurement error models can be used to estimate the line. One monograph on such models is Cheng and Van Ness [4]. If one cannot estimate the error variances (or their ratio, λ) then Ricker recommends the use of the least areas line as being the best approximation: it gives y and x equal weight and will be exact if $\lambda = \text{var}(y)/\text{var}(x)$, i.e. when the ratio of error variances equals the ratio of data variances. For the case of mutual natural variability 'there is no basis for assigning separate vertical and horizontal components to the deviation', i.e. 'it is impossible to say whether it is y or x that is responsible for the deviations from the line'. In this case Ricker concludes that if the data are binormally distributed then the least areas line be used to describe the central trend, and least squares to estimate one variable from the other. For the mixed case i.e. having both measurement error and natural variability, 'the best that can be done is to treat them in terms of whichever source of variation makes the larger contribution to the total. In biological work this will usually be natural variability'.

Despite appearing in so many other fields, it is remarkable that this technique does not seem to have appeared in the numerical analysis/approximation literature. For example it is not listed in Grosse's Algorithms for Approximation catalogue. The present paper looks at an obvious way of extending the approach to any number of variables by minimising volumes.

3 Least volume fitting

We now intend to fit a linear function of the form $\sum_{j=1}^{p} a_j x_j = c$ to data $\{X_j\}$ in p dimensions, in other words we have data on p variables and we seek a linear relationship between them. Of course this is not uniquely specified because we can divide through by any non-zero constant. Thus we are free to impose a constraint on the coefficients, such

as $c = 1$. Note that we shall not permit any of the coefficients a_j to be zero because that would imply the associated variable x_j is unrelated to the other variables

One obvious way of generalising the least areas procedure to higher dimensions is to minimise the volumes (or hypervolumes). Each data point will have associated with it a 'volume deviation' which is simply the product of its deviations from the fitted plane in each dimension. We must take care to make all these non-negative by taking the absolute values. For the ith data point this volume deviation V_i is proportional to

$$\left| \frac{(\sum_{j=1}^{p} a_j X_{ij} - c)^p}{\prod_{j=1}^{p} a_j} \right|.$$

We now introduce non-negative variables u_i, v_i to deal with the absolute value of the numerator. The positive u_i represent points on one side of the fitted plane, and positive v_i refer to points on the opposite side. Setting $c = 1$ allows us to model the bracketed term thus:

$$u_i - v_i = \sum_j a_j X_{ij} - 1.$$

At least one of each of the pairs $\{u_i, v_i\}$ will be forced to be zero by their presence in the objective function which is being minimised. Consequently the numerator can be represented as $\sum(u_i^p + v_i^p)$. We shall suppose the denominator is positive; if it is not we can always make it so by multiplying one of the x-variables by -1 so that its coefficient, and hence that of the product of coefficients, also changes sign. We can now formulate our problem as the following fractional programme:

$$\begin{aligned} \text{Minimise} \quad & \sum_i (u_i^p + v_i^p) / \prod a_j \\ \text{such that} \quad & u_i - v_i = \sum_j a_j X_{ij} - 1 \\ \text{and} \quad & u_i, v_i \geq 0. \end{aligned}$$

The field of fractional programming is comprehensively covered by Stancu-Minasian [18]. We note that Draper and Yang [6] used a different route to generalising the technique to more than two dimensions. They minimised the pth root of the squared volumes and showed that the estimated coefficients were a convex combination of those from the p OLS estimates.

4 Numerical test

We shall now apply the least volume criterion to try and uncover the coefficients from data that have been generated from a known underlying model with some randomness thrown in. In order to make this a difficult test we shall choose data, which suffers from multicollinearity. This means that there is a near linear dependence within the data, i.e., one of the variables almost lies in the space spanned by the remaining variables, and so we are close to being rank-deficient. The data is taken from Belsley's [3] comprehensive monograph on collinearity. The generating model is

$$y = 1.2 - 0.4x_1 + 0.6x_2 + 0.9x_3 + \epsilon$$

with ϵ normally distributed with zero mean and variance 0.01. The absolute correlations between the variables ranged from 0.35 to 0.61 and so these in themselves would not have alerted the researcher to any difficulty associated with multicollinearity. Two very similar data sets (A,B) are tabulated in Belsley based on this model. For set A ordinary least squares (OLS) gives:

$$y = 1.26 + 0.97x_1 + 9.0x_2 - 38.4x_3.$$

The fit as measured by $\mathbf{R}^2$ is very good at 0.992 but the underlying model is far from being uncovered. In particular, the coefficient of x_2 is 15 times too high and two of the coefficients have the wrong sign! Getting the signs wrong is very serious if one is trying to understand how variables are related to each other. Turning to the least volume approach we find:

$$y = 1.20 - 0.43x_1 + 0.37x_2 + 1.97x_3.$$

We now have all the correct signs and the magnitudes are much closer to the true ones.

Repeating this for data set B:

$$\begin{aligned} \text{OLS}: &\quad y = 1.275 + 0.25x_1 + 4.5x_2 - 17.6x_3 \\ \text{Least volume}: &\quad y = 1.20 - 0.43x_1 + 0.37x_2 + 1.98x_3. \end{aligned}$$

Once again the least volume approach produces a superior model. Moreover it is also worth noting that the two OLS models are very different from each other whereas the least volume models seem to be more stable to small variations in the data. This is noteworthy because of how similar the two data sets were: the y-values were identical for sets A and B, and the x-values never varied by more than one in the third digit. Thus our method seems to be much more stable than OLS. Of course a comprehensive set of Monte Carlo simulations is required to fully explore this aspect.

5 Conclusion

We have presented a fitting method whose criterion combines the deviations in each dimension by multiplying them together. This simple device means that re-scaling of any of the variables e.g. by changing the units of measurement, will give rise to an equivalent model. This property of units-invariance is not shared by the total least squares approach (or orthogonal regression: where the sum of the perpendicular distances to the fitted plane are minimised). By taking the product of the deviations we ensure that all variables are treated on the same basis and this is useful if the purpose is to find an underlying relationship rather than to predict one of the variables.

When we applied the technique to data we were able to recover the underlying relationship much more closely than when least squares was used. Not only were the signs of the coefficient correctly reproduced (which is crucial for understanding directions of change) but also the magnitudes were much closer to the true values than least squares estimates. It appears that the least volume method may be superior when there is multicollinearity in the data. Much more simulation needs to be done to investigate this potentially very valuable feature.

Bibliography

1. G. J. Babu and E. D. Feigelson, Analytical and Monte Carlo comparisons of six different linear least squares fits, *Communications in Statistics: Simulation and Computation*, **21** (2) (1992), 533–549.
2. F. Barker, Y. C. Soh, and R. J. Evans, Properties of the geometric mean functional relationship, *Biometrics* **44**, (1988) 279–281.
3. D. A. Belsley, *Conditioning Diagnostics*, Wiley, New York, 1991.
4. C-L Cheng and J. W. Van Ness, *Statistical Regression with Measurement Error*, Arnold, London, 1999.
5. N. R. Draper and H. Smith, *Applied Regression Analysis* (3rd edition), Wiley, New York, 1998.
6. N. R. Draper and Y. Yang, Generalization of the geometric mean functional relationship, *Computational Statistics and Data Analysis* **23** (1997), 355–372.
7. P. D. Greenall, PD (1949). The concept of equivalent scores in similar tests. *British J. of Psychology: Statistical Section* **2** (1949), 30–40.
8. E. Grosse, (1989). A catalogue of algorithms for approximation, in *Algorithms for Approximation II*, eds. J. C. Mason and M. Cox.
9. R. M. Hirsch and E. J. Gilroy, Methods of fitting a straight line to data: examples in water resources, *Water Resources Bulletin* **20** (5) (1984), 705–711.
10. M. G. Kendall and A. Stuart, *The Advanced Theory of Statistics*, 4th edition, vol.2, 391–409, Griffin, London, 1979.
11. D. K. Kimura, Symmetry and scale dependence in functional relationship regression, *Systematic Biology* **41** (2) (1992), 233–241.
12. W. H. Kruskal, On the uniqueness of the line of organic correlation, *Biometrics* **9** (1953), 47–58.
13. R. L. Miller and J. S. Kahn, *Statistical Analysis in the Geologocal Sciences*, Wiley, NY, 1962.
14. A. S. Otis, The method for finding the correspondence between scores in two tests, *J. of Educational Psychology XIII* (1922), 524–545.
15. J. M. V. Rayner, Linear relations in biomechanics: the statistics of scaling functions, *J. Zool., Lond.(A)* **206** (1985), 415–439.
16. W. E. Ricker, Linear regressions in fishery research, *J. Fisheries Research Board of Canada* **30** (1073), 409–434.
17. W. E. Ricker, Computation and uses of central trend lines, *Canadian J. of Zoology* **62** (1984), 1897–1905.
18. I. M. Stancu-Minasian, *Fractional Programming: Theory, Methods and Applications*, Kluwer Academic, Dordrecht, 1997.
19. G. Stromberg, Accidental and systematic errors in spectroscopic absolute magnitudes for dwarf G0-K2 stars, *Astrophysical J.* **92** (1940), 156–169.
20. G. Teissier, (1948). La relation d'allometrie, *Biometrics* **4** (1) (1948), 14–48.

Some problems in orthogonal distance and non-orthogonal distance regression

G. A. Watson

Department of Mathematics, University of Dundee, Dundee DD1 4HN, Scotland.
gawatson@maths.dundee.ac.uk

Abstract

Of interest here is the problem of fitting a curve or surface to given data by minimizing some norm of the distances from the points to the surface. These distances may be measured orthogonally to the surface, giving orthogonal distance regression, and for this problem, the least squares norm has attracted most attention. Here we will look at two other important criteria, the l_1 norm and the Chebyshev norm. The former is of value when the data contain wild points, the latter in the context of accept/reject criteria. There are however circumstances when it is not appropriate to force the distances to be orthogonal, and two possibilities of this are also considered. The first arises when the distances are aligned with certain fixed directions, and the second when angular information is available about the measured data points. For the least squares norm, we will consider some algorithmic developments for these problems.

1 Introduction

Of interest here is the problem of fitting to given data a curve or surface which depends on a vector $\mathbf{a} \in R^n$ of parameters. The underlying approach is such that (1) a point on the surface is associated with each data point, (2) the fit of the surface is measured by a norm of the vector whose components are the distances between each pair of corresponding points, (3) the (correct) Gauss-Newton steps in $\mathbf{a}$ are used as a basis for minimizing this norm. The distances may be orthogonal to the surface, giving orthogonal distance regression (ODR), or may be forced to satisfy some other criterion which makes them non-orthogonal in general. We consider both situations.

For the ODR problem, most attention has been given to the least squares norm (eg [5], [8], [9], [16], [17], [22]). Here we will look at two other important criteria, the l_1 norm and the Chebyshev norm. The former is of value when the data contain wild points, the latter in the context of accept/reject criteria. For the non-orthogonal distance problem we will restrict attention to the least squares case.

In terms of a vector $\mathbf{a} \in R^n$ of parameters, the curve or surface may be defined in two ways, (a) **parametrically,** when a point $\mathbf{x}$ on the surface is given by

$$\mathbf{x} = \mathbf{x}(\mathbf{a}, \mathbf{t}),$$

with $\mathbf{t}$ the parameters whose values define the particular point, or (b) **implicitly**, when the surface is defined by the set of points $\mathbf{x}$ satisfying the scalar equation

$$f(\mathbf{a}, \mathbf{x}) = 0.$$

It is also assumed here that the expressions required in these representations are differentiable functions of their parameters.

2 l_1 and l_∞ ODR

Consider first the l_1 case. Then the problem is

$$\text{minimize} \sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{z}_i(\mathbf{a})\|,$$

where the points $\mathbf{z}_i(\mathbf{a})$ are the nearest points to $\mathbf{x}_i$ on the surface defined by $\mathbf{a}$, and where we will assume throughout that unadorned norms are Euclidean norms. Let

$$\delta_i = \|\mathbf{x}_i - \mathbf{z}_i(\mathbf{a})\|, \ i = 1, \ldots, m.$$

Then the problem is effectively now defined in terms of the vector $\mathbf{a}$ alone. It is easy to to calculate the correct Gauss-Newton step in $\mathbf{a}$, which minimizes

$$\|\delta + \nabla_{\mathbf{a}} \delta \mathbf{d}\|_1$$

with respect to $\mathbf{d}$. Now

$$\nabla_{\mathbf{a}} \delta_i = -\frac{(\mathbf{x}_i - \mathbf{z}_i(\mathbf{a}))^T}{\delta_i} \nabla_{\mathbf{a}} z_i(\mathbf{a}), \ \delta_i \neq 0,$$

so that there are potential problems if any $\delta_i \to 0$. Given the nature of the l_1 problem, we cannot exclude that possibility. In fact although δ is not a smooth function, because derivative discontinuities only occur at zero values it is a **strong semi-smooth function**, as defined in [12]. Ideas from smooth analysis and from strong semi-smooth analysis as developed in [11] can then be combined to give a local convergence analysis for the present problem. Fast local convergence for the usual smooth problem relies on strong uniqueness [4]; for the l_1 norm, this can be interpreted in terms of a requirement that the sequence of solutions $\mathbf{d}^k$ is "well-behaved" in a certain sense [1]. An analogous requirement can be stated here.

Let the current approximation be $\mathbf{a}^k$ and let J^k denote the Jacobian matrix $\nabla_{\mathbf{a}} \delta(\mathbf{a}^k)$, assuming this exists. Then the Gauss-Newton step $\mathbf{d}^k$ minimzes

$$\|\delta(\mathbf{a}^k) + J^k \mathbf{d}\|_1.$$

It is well known (see for example [18]) that if J^k has full rank then there always exists a solution $\mathbf{d}^k$ and an index set Z^k containing n indices such that

$$\delta_i(\mathbf{a}^k) + \mathbf{e}_i^T J^k \mathbf{d}^k = 0, \ i \in Z^k,$$

where $\mathbf{e}_i$ is the ith coordinate vector. Let $\mathbf{a}^*$ be a limit point of the iteration. Then for $\mathbf{a}^k$ close enough to $\mathbf{a}^*$, assume that J^k exists and

(i) $\delta(\mathbf{a}^k) + J^k \mathbf{d}^k$ has **exactly** n zeros, corresponding to an index set Z^k,

(ii) $Z^k = Z^*$, independent of k,

(iii) the $n \times n$ matrices whose rows are $\mathbf{e}_i^T J^k, i \in Z^*$, are bounded away from singularity.

In practice these conditions ensure that $\mathbf{d}^k$ is unique, and there is no redundancy in the zero components. An analysis is given in [21] for both parametric and implicit fitting. The main result is the following.

Theorem 2.1 *[21] Let the Gauss-Newton method produce a sequence $\mathbf{a}^k \to \mathbf{a}^*$, where $\delta(\mathbf{a}^k)$ has no zero components, and let (i)-(iii) above hold. In the parametric case, assume that for all $i \in Z^*$, there exists a unique unit normal vector $\mathbf{n}_i$ (up to change of sign) at the point $\mathbf{x}_i$ on the surface defined by $\mathbf{a}^*$. Then the (undamped) Gauss-Newton method converges to $\mathbf{a}^*$ at a second order rate.*

The significance of this result is that, for both parametric and implicit fitting, any δ_i tending to zero is not **by itself** necessarily an obstacle to good performance of the Gauss-Newton method in the l_1 case. What is more significant is the possibility of very slow convergence and this has more to do with the **number** of those zero components of δ at a limit point, rather than just their presence. A fundamental requirement for the condition (ii) is that the number of zero components of $\delta(\mathbf{a}^*)$ is n. Of course, this condition is a rather special one, and for many problems, will not be satisfied. There is slow (possibly very slow) convergence associated with this case.

Turning now to the l_∞ problem, this can be stated

$$\text{minimize } \max_i \|\mathbf{x}_i - \mathbf{z}_i(\mathbf{a})\|,$$

with $\mathbf{z}_i(\mathbf{a})$ defined as before. Again $\delta_i = \|\mathbf{x}_i - \mathbf{z}_i(\mathbf{a})\|$ is not a smooth function, but a solution normally occurs in a region where δ is smooth. Therefore the problem does not differ significantly from the usual nonlinear minimax problem: the main requirement for fast local convergence is that at a limit point the norm is attained at $n+1$ indices [4].

Two simple examples in 2 dimensions are given by way of illustration. A standard line search is incorporated to force global convergence, although trust region methods are a popular alternative. Indeed, local convergence is the main concern here, and we have not begun to address important issues to do with the development of robust general purpose algorithms.

Example 2.2 Consider the Späth data set [13] ($m = 7$), and consider fitting an ellipse defined implicitly, using the l_∞ and l_1 norms. The solutions are illustrated in Figure 1, where the dashed ellipse and dashed lines are the l_∞ solution and corresponding orthogonal directions, and the solid ellipse and solid lines are the l_1 solution and corresponding directions. Both ellipses were obtained using the Gauss-Newton method starting from the circle centre (5,5), radius 2, in 4 and 5 iterations respectively for 5 figure accuracy.

Example 2.3 Consider next the GGS data set [6], which has $m = 8$. Similar fits to those for Example 1 are shown in Figure 2. Again the Gauss-Newton method was used starting from the circle centre (5,5), radius 2, to give convergence in 6 iterations (l_∞) and 7 iterations (l_1).

For both these examples $n = 5$, and favourable conditions hold so that there is quadratic convergence both in the l_1 and l_∞ cases. Otherwise, the key to recovering fast

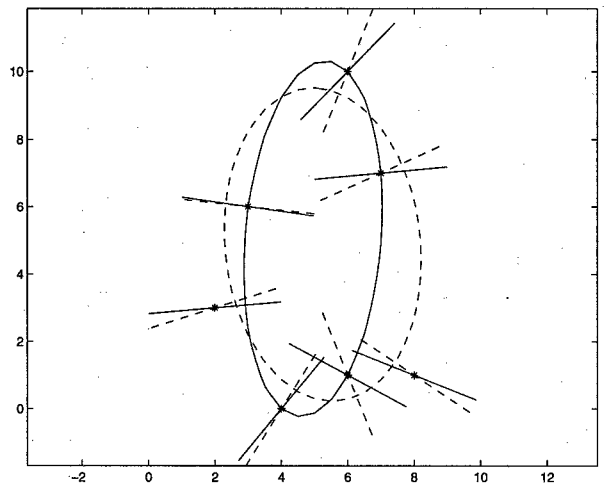


FIG. 1. l_1 and l_∞ fits to Späth data set.

local convergence in the l_1 case is to identify Z^* and to reformulate the problem locally as

$$\text{minimize} \sum_{i \notin Z^*} \|\mathbf{x}_i - \mathbf{z}_i(\mathbf{a})\| \text{ subject to } \mathbf{x}_i - \mathbf{z}_i(\mathbf{a}) = 0,\ i \in Z^*. \tag{2.1}$$

A similar remedy in the l_∞ case is as follows. For a limit point $\mathbf{a}^*$ of the iteration, let

$$I^* = \{i : \delta_i(\mathbf{a}^*) = \max_i \delta_i(\mathbf{a}^*)\}.$$

Then if we can identify I^*, $\mathbf{a}^*$ solves, for any $j \in I^*$:

$$\text{minimize } \delta_j(\mathbf{a}) \text{ subject to } \delta_i(\mathbf{a}) - \delta_j(\mathbf{a}) = 0,\ i \in I^* \backslash j.$$

Example 2.4 Fitting an l_∞ ODR line in R^3 to 100 random data points (equivalent to finding the circumscribing cylinder of smallest radius) gives slow convergence of the basic method, because $|I^*| = 3$ and $n = 4$. But once we identify $I^* = \{4, 42, 58\}$, only 5 iterations of the NAG Fortran subroutine E04UCF are required for 6 figure accuracy.

3 Non-orthogonal l_2 distance regression

3.1 Using fixed directions

Suppose that the data come from sampling the surface of a manufactured part, using a coordinate measuring machine with a touch probe. It has been argued by Hulting [10] that choosing the directions to be the known probe directions $\mathbf{v}_i$ (relative to a fixed frame of reference) not only makes explicit use of the measurement design, but

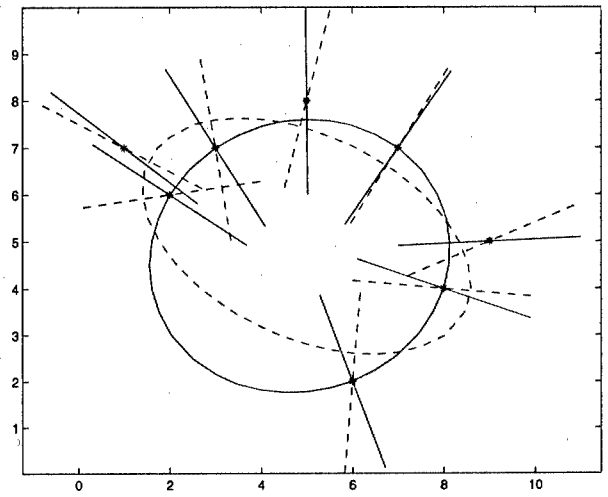


FIG. 2. l_1 and l_∞ fits to GGS data set.

also complies with traditional fixed-regressor assumptions (enabling standard inference theory to apply).

Let $\mathbf{x}_i$, $i = 1, \ldots, m$ as usual be the data points, and let $\mathbf{z}_i$ be the corresponding points on the surface reached by travelling along the lines from $\mathbf{x}_i$ in the direction $\mathbf{v}_i$. Then we require to minimize $\|\delta\|$ where

$$\delta_i = \|\mathbf{x}_i - \mathbf{z}_i(\mathbf{a})\|, \ i = 1, \ldots, m,$$

with $\mathbf{z}_i(\mathbf{a})$ defined by

$$\mathbf{z}_i(\mathbf{a}) - \mathbf{x}_i = \delta_i \mathbf{v}_i, \ i = 1, \ldots, m,$$

where $\mathbf{v}_i$ satisfying $\mathbf{v}_i^T \mathbf{v}_i = 1$ is given for each i. In case of ambiguity, the smallest value of δ_i is chosen. The basic idea in efficient algorithmic development is again to treat the problem as one in $\mathbf{a}$ alone, which can be solved as before by the Gauss-Newton method (or variants). Let $\mathbf{a}$ be given. Then for each point $\mathbf{x}_i$, the point where the line through $\mathbf{x}_i$ in the direction $\mathbf{v}_i$ first cuts the surface can be obtained (this calculation replaces the "footpoint problem" of calculating $\mathbf{z}_i(\mathbf{a})$ as the point on the surface in the orthogonal distance problem), giving δ_i as a function of $\mathbf{a}$. Methods based on Gauss-Newton steps are developed for the parametric case in [19], [20], and for the implicit case in [7].

By way of illustration, the 2 data sets previously considered in Examples 1 and 2 are used to fit ellipses defined implicitly with a particular choice of directions $\mathbf{v}_i$. The initial (circles) and final ellipses (together with the data points and the directions $\mathbf{v}_i$) are shown in Figures 3 and 4. The calculations needed respectively 19 and 17 iterations, reflecting the fact that, unlike the l_1 and l_∞ cases, the convergence rate is linear.

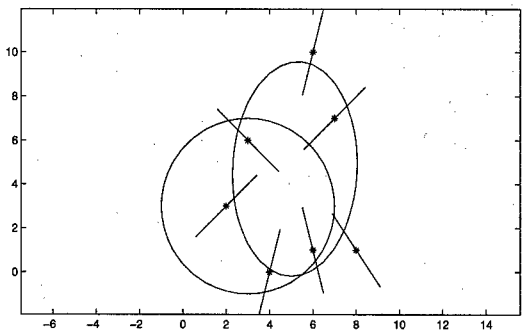


FIG. 3. l_2 fit to Späth data set: fixed $\mathbf{v}_i$.

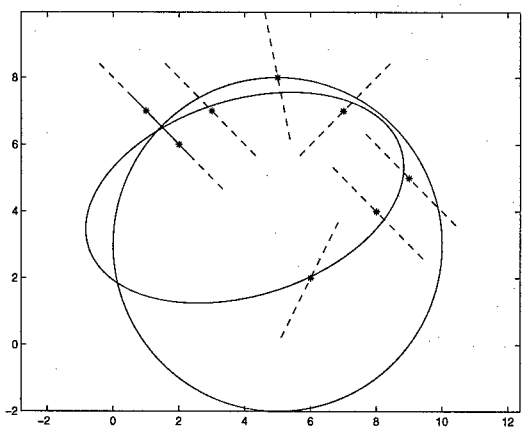


FIG. 4. l_2 fit to GGS data set: fixed $\mathbf{v}_i$.

3.2 Using angular information

Berman and Griffiths [2, 3] consider fitting a circle when angular differences between successively measured data points are known, with applications in physics and archaeology. This fitting problem has been extended to the case of ellipses and ellipsoids by Späth in [14, 15] and it is this kind of problem which is of interest here. The methods of [14] and [15] are based on the alternating algorithm, and while this can be perhaps surprisingly effective (particularly with a reparameterization of the problem), we consider here a correct separated Gauss-Newton method similar to that used before. In addition to (usually) better local convergence properties, standard step-length control can be incorporated.

To illustrate, consider fitting an ellipse in general position. It is convenient to do this by allowing the data to rotate, and fitting to those an ellipse in normal position, aligned with the axes. Let (x, y) denote the components of $\mathbf{x}$. Then we work with the data

$$x_i(\phi) = x_i \cos\phi + y_i \sin\phi, \;\; y_i(\phi) = -x_i \sin\phi + y_i \cos\phi,$$

for $i = 1, \ldots, m$, where ϕ is an unknown parameter. Therefore we require to minimize, with respect to the 6 parameters a, b, p, q, α, ϕ, the function

$$\sum_{i=1}^{m} \{(x_i(\phi) - a - p\cos(\alpha + t_i))^2 + (y_i(\phi) - b - q\sin(\alpha + t_i))^2\},$$

where the numbers t_i are given. Because $(\alpha + t_{i+1}) - (\alpha + t_i) = t_{i+1} - t_i$, for each i, we can interpret this as saying say that the angular differences are known, with a degree of freedom given by the parameter α. Note that at a solution to this problem, the directions between pairs of points $(x_i(\phi), y_i(\phi))$ and the corresponding points on the ellipse will not generally be orthogonal to the ellipse.

Differentiating the above expression with respect to a, p, b, q gives

$$A_1 \begin{bmatrix} a \\ p \end{bmatrix} = \mathbf{c}_1, \tag{3.1}$$

where

$$A_1 = \begin{bmatrix} m & \sum_{i=1}^m \cos(\alpha + t_i) \\ \sum_{i=1}^m \cos(\alpha + t_i) & \sum_{i=1}^m \cos^2(\alpha + t_i) \end{bmatrix}, \mathbf{c}_1 = \begin{bmatrix} \sum_{i=1}^m x_i(\phi) \\ \sum_{i=1}^m x_i(\phi)\cos(\alpha + t_i) \end{bmatrix};$$

$$A_2 \begin{bmatrix} b \\ q \end{bmatrix} = \mathbf{c}_2, \tag{3.2}$$

where

$$A_2 = \begin{bmatrix} m & \sum_{i=1}^m \sin(\alpha + t_i) \\ \sum_{i=1}^m \sin(\alpha + t_i) & \sum_{i=1}^m \sin^2(\alpha + t_i) \end{bmatrix}, \mathbf{c}_2 = \begin{bmatrix} \sum_{i=1}^m y_i(\phi) \\ \sum_{i=1}^m y_i(\phi)\sin(\alpha + t_i) \end{bmatrix}.$$

Then (3.1) and (3.2) give (a, b, p, q) as functions of α and ϕ, provided that A_1 and A_2 are nonsingular: this will be assumed. For given α and ϕ, we can therefore define the function to be minimized as

$$F(\alpha, \phi) = \|\delta(\alpha, \phi)\|,$$

where

$$\delta_i = \|\mathbf{w}_i\|, \;\; i = 1, \ldots, m, \tag{3.3}$$

with

$$\mathbf{w}_i = (x_i(\phi) - a - p\cos(\alpha + t_i), \;\; y_i(\phi) - b - q\sin(\alpha + t_i))^T,$$

and with a, b, p, q defined by (3.1) and (3.2). Then we can apply the Gauss-Newton method to the minimization of $F(\alpha, \phi)$. The basic step $\mathbf{d} = (\delta\alpha, \delta\phi)^T$ is given by finding

$$\min_{\mathbf{d} \in R^2} \|\delta + J\mathbf{d}\|, \tag{3.4}$$

where $J \in R^{m\times 2}$ has ith row given by

$$\mathbf{e}_i^T J = \nabla_{\alpha,\phi}\delta_i(\alpha,\phi),\ i = 1,\ldots,m.$$

Now

$$\nabla_{\alpha,\phi}\delta_i(\alpha,\phi) = \frac{\mathbf{w}_i^T}{\delta_i}(\nabla_{\alpha,\phi}\mathbf{w}_i + (\nabla_{a,p,b,q}\mathbf{w}_i)M),\ \delta_i \neq 0, \tag{3.5}$$

where

$$M = \nabla_{\alpha,\phi}\begin{pmatrix} a \\ p \\ b \\ q \end{pmatrix} \in R^{4\times 2}.$$

It is easy to compute M from (3.1) and (3.2) which can be interpreted as identities in α and ϕ. The details are omitted, but all the linear systems use just the matrices A_1 and A_2, and apart from the solution of (3.4) (a least squares problem in two variables), there remains only evaluation of expressions.

Example 3.1 Consider Example 1 from [14], which has $m = 11$. Starting from $\alpha = 0$, $\phi = 0$, 15 iterations are required to satisfy the stopping criterion $\|\mathbf{d}\|_\infty < 0.001$. The resulting value of $\|\delta\|^2$ is 7.7211, with $a = 2.1253$, $b = -0.1700$, $p = 4.1281$, $q = 3.0931$, $\alpha = 13.2348°$, $\phi = 34.7309°$.

Example 3.2 Next consider Example 2 from [14], which has $m = 8$. Again starting from $\alpha = 0$, $\phi = 0$, 9 iterations are required to satisfy the stopping criterion $\|\mathbf{d}\|_\infty < 0.001$. The resulting value of $\|\delta\|^2$ is 4.4946, with $a = 4.3608$, $b = 1.9537$, $p = 5.3717$, $q = 3.3704$, $\alpha = -0.6215°$, $\phi = 26.3889°$.

4 Conclusions

We have examined some aspects of fitting curves and surfaces to given data. The underlying criterion involves associating with each data point a point on the surface and minimizing some norm of the vector whose components are the distances between pairs of points. The distances can be orthogonal to the surface, or fixed in some other way. But the problems have in common that methods based on separated Gauss-Newton steps can readily be developed.

Bibliography

1. Anderson, D. H. and M. R. Osborne, Discrete, non-linear approximation problems in polyhedral norms, Num. Math. **28**, 143–156 (1977).
2. Berman, M., Estimating the parameters of a circle when angular differences are known, Appl. Statist. **32**, 1–6 (1983).
3. Berman, M. and D. Griffiths, Incorporating angular information into models for stone circle data, Appl. Statist. **34**, 237–245 (1985).
4. Cromme, L., Strong uniqueness: a far reaching criterion for the convergence of iterative processes, Numer. Math. **29**, 179–193 (1978)

5. Forbes, A. B., Least squares best fit geometric elements, in *Algorithms for Approximation II*, eds. J. C. Mason and M. G. Cox, Chapman and Hall, London, 311–319 (1990).
6. Gander, W., G. H. Golub and R. Strebel, Fitting of circles and ellipses: least square solution, BIT **34**, 556 –577 (1994).
7. Gulliksson, M., I. Söderkvist and G. A. Watson, Implicit surface fitting using directional constraints, BIT **41**, 331–344 (2001).
8. Helfrich, H.-P. and D. Zwick, A trust region method for implicit orthogonal distance regression, Numer. Alg. **5**, 535–545 (1993).
9. Helfrich, H.-P. and D. Zwick, A trust region algorithm for parametric curve and surface fitting, J. Comp. Appl. Math. **73**, 119–134 (1996).
10. Hulting, F. L., Discussion contribution to the paper by M. M. Dowling, P. M. Griffin, K.-L. Tsui and C. Zhou, Statistical issues in geometric feature inspection using coordinate measuring machines, Technometrics **39**, 18–20 (1997).
11. Qi, L., Convergence analysis of some algorithms for solving nonsmooth equations, Math. of Operations Research **18**, 227–244 (1993).
12. Qi, L. and G. Jiang, Semismooth Karush-Kuhn-Tucker equations and convergence analysis of Newton methods and quasi-Newton methods for solving these equations, Math. of Operations Research **22**, 301–325 (1997).
13. Späth, H., Least squares fitting by circles, Computing **57**, 179–185 (1996).
14. Späth, H., Estimating the parameters of an ellipse when angular differences are known, Comput. Stat. **14**, 491–500 (1999).
15. Späth, H. Least squares fitting of spheres and ellipsoids using not orthogonal distances, Math. Comm. **6**, 89–96 (2001).
16. Turner, D. A., *The Approximation of Cartesian Co-ordinate Data by Parametric Orthogonal Distance Regression*, PhD Thesis, University of Huddersfield (1999).
17. Turner, D. A., I. J. Anderson, J. C. Mason, M. G. Cox and A. B. Forbes, An efficient separation-of-variables approach to parametric orthogonal distance regression, in *Advanced Mathematical and Computational Tools in Metrology IV*, eds P. Ciarlini, A. B. Forbes, F. Pavese and D. Richter, Series on Advances in Mathematics for Applied Sciences, Volume 53, World Scientific, Singapore, 246–255 (2000).
18. Watson, G. A., *Approximation Theory and Numerical Methods*, John Wiley, Chichester (1980).
19. Watson, G. A., Least squares fitting of circles and ellipses to measured data, BIT **39**, 176–191 (1999).
20. Watson, G. A., Least squares fitting of parametric surfaces to measured data, ANZIAM J **42** (E), C68–C95 (2000).
21. Watson, G. A, On the Gauss-Newton method for l_1 orthogonal distance regression, IMAJ. Num. Anal. (to appear).
22. Zwick, D. S., Applications of orthogonal distance regression in metrology, in *Recent Advances in Total Least Squares and Errors-in-Variables Techniques*, ed S. Van Huffel, SIAM, Philadelphia, pp. 265–272 (1997).

Chapter 6

Splines and Wavelets

Preceding Page Blank

Nonlinear multiscale transformations: From synchronization to error control

F. Arandiga and R. Donat

Dept. Matematica Aplicada, University of Valencia, Spain.
arandiga@uv.es donat@uv.es

Abstract

Data-dependent interpolatory techniques can be used in the reconstruction step of a multiresolution "*à la Harten*". These interpolatory techniques lead to nonlinear multiresolution schemes. When dealing with nonlinear algorithms, the issue of the stability needs to be carefully considered. In this paper we analyze and compare several strategies for image compression and their ability to effectively control the global error due to compression.

1 Introduction

Multiscale transformations are being used in recent times in the first step of transform coding algorithms for image compression. Ideally, a multiscale transformation allows for an *efficient* representation of the image data, which is then processed using a (non-reversible) quantizer and passed on to the encoder which produces the final compressed set of data which is ready to be transmitted or stored. Compression is indeed achieved during the second and third steps: the quantization and the encoding of the transformed set of discrete data.

It is quite clear that the properties of the multiscale transformation are most important in the overall performance of the transform coding algorithm. Until recently, the multiscale transformations used for image compression were always based on linear filter banks, however, the nonlinear alternative has been explored lately by various authors from different points of view, and preliminary results show the alternative to be very promising [12, 8, 6, 2, 3]. The key question when using, or even designing, a nonlinear multiscale transformation is that of stability. In order for such transformations to be useful tools in image coding, it is absolutely necessary to keep a tight control on the effect of quantization errors in the decoding process.

In this paper we examine the question of stability for nonlinear multiscale transformations within Harten's framework for multiresolution [14, 15]. Harten's framework is broad enough to include all classical wavelet transformations as particular cases (just as it happens in the Lifting framework of W. Sweldens [17], developed slightly later in time but independently), however the design of the multiscale transformation is done directly on the spatial domain.

The building blocks of Harten's multiresolution framework are two operators that connect adjacent resolution levels. The *Decimation* (or also, *Restriction*) operator is a linear operator which acts as a low-pass filter, extracting low-resolution information from a discrete data set. The *Prediction* operator (also *Projection*) uses low-resolution data to predict discrete data at a higher resolution level. It is precisely the *design* of this operator what distinguishes Harten's framework from all other multiresolution frameworks. The prediction operator is based on a *consistent Reconstruction* technique, and this opens up a tremendous number of possibilities in the design of multiresolution schemes. The use of the reconstruction process as a design tool makes it, conceptually, a simple matter to introduce adaptivity into the multiscale transformation; we only need to make the reconstruction process data-dependent [5, 4, 14].

This paper is organized as follows. In Section 2 we recall the so-called cell-average framework, an appropriate setting for image compression, and describe a class of nonlinear prediction operators obtained by mean-average interpolation [10, 14, 15]. In Section 3 we examine the question of stability for nonlinear multiscale transformations and relate it to the *synchronization* of the data-dependent choices made in the encoder and the decoder. We also include a set of numerical experiments that illustrate he performance of several nonlinear multiscale transformations.

2 Multiscale transformations in the cell-average setting

Harten's general framework for multiresolution [15] relies on two operators, Decimation and Prediction, that define the basic interscale relations. These operators act on finite dimensional linear vector spaces, V^j, that represent the different resolution levels (j increasing implies more resolution)

$$\text{(a) } D^j : V^j \to V^{j-1}, \qquad \text{(b) } P_j : V^{j-1} \to V^j, \tag{2.1}$$

and must satisfy two requirements of algebraic nature; D^j needs to be a *linear* operator and $D^j P_j = I_{V^{j-1}}$, i.e., the identity operator on the lower resolution level represented by V^{j-1}. For all practical purposes, V^j can be considered as spaces of finite dimensional sequences.

Using these two operators, a vector (i.e., a discrete sequence) $v^j \in V^j$ can be decomposed and reassembled as follows

$$\text{(a)} \quad \begin{array}{ccccc} v^j & \to & v^{j-1} & = & D^j v^j \\ & \searrow & e^j & = & v^j - P_j v^{j-1} \end{array}, \qquad \text{(b)} \quad v^j = P_j v^{j-1} + e^j \tag{2.2}$$

where e^j represents the error in trying to predict the jth level data, v^j, from the low resolution data $v^{j-1} = D^j v^j$, using the prediction operator P_j .

In the cell-average setting, the discrete data are interpreted as the cell-averages of a function on an underlying grid, which determines the level of resolution of the given data. The one dimensional case, in which one considers a set of nested dyadic grids on the interval $[0,1]$, $\{X^j\}$, $j \geq 0$ of size $h_j = 2^{-j} h_0$,

$$X^j = \{x_i^j\} \qquad x_i^j = i \cdot h_j, \quad i = 0, \ldots, N_j \qquad N_j \cdot h_j = 1 \tag{2.3}$$

is the easiest one to describe, and it is also directly applicable to two-dimensional (2D) data via tensor product [2, 3] (the cell-average framework in several dimensions and non-tensor product (unstructured) grids is considered in e.g. [1]).

In this simple one-dimensional setting, the cell-average framework is characterized by the following decimation operator D^j

$$(D^j v^j)_i = \frac{1}{2}(v^j_{2i-1} + v^j_{2i}), \qquad 1 \le i \le N_{j-1}, \tag{2.4}$$

where N_j is the number of equally spaced intervals on X^j, the grid on $[0,1]$ that represents the jth resolution level. The *consistency* requirement for the prediction operator, i.e., $D^j P_j = I_{V^{j-1}}$ which is the only necessary requirement for the prediction in Harten's framework, becomes then

$$(P_j v^{j-1})_{2i-1} + (P_j v^{j-1})_{2i} = 2v^{j-1}_i. \tag{2.5}$$

Observe that (2.4) and (2.5) imply that

$$(P_j v^{j-1})_{2i} + (P_j v^{j-1})_{2i-1} = v^j_{2i} + v^j_{2i-1}.$$

Hence

$$e^j_{2i-1} = v^j_{2i-1} - (P_j v^{j-1})_{2i-1} = (P_j v^{j-1})_{2i} - v^j_{2i} = -e^j_{2i}.$$

Therefore the prediction errors at even and odd grid points on the jth level in (2.2) are not independent. By considering only the prediction errors at (for example) the odd points of the grid X^j, one immediately gets a one-to-one correspondence between the sets $\{v^j_i\}_{i=1}^{N_j} \leftrightarrow \{\{v^{j-1}_i\}_{i=1}^{N_{j-1}}, \{d^j_i\}_{i=1}^{N_{j-1}}\}$, with $d^j_i = e^j_{2i-1}$ and $v^{j-1} = D^j v^j$. The one-dimensional multiscale transformation and its inverse can be written as follows,

$$v^L \longrightarrow Mv^L = (v^0, d^1, \ldots, d^L) \quad \left\{ \begin{array}{l} \text{For } j = L, \ldots, 1 \\ \quad \text{For } i = 1, \ldots, N_{j-1} \\ \qquad v^{j-1}_i = (v^j_{2i} + v^j_{2i-1})/2 \\ \qquad d^j_i = v^j_{2i-1} - (P_j v^{j-1})_{2i-1} \end{array} \right\}, \tag{2.6}$$

$$v_d = (v^0, d^1, \ldots, d^L) \longrightarrow M^{-1} v_d \quad \left\{ \begin{array}{l} \text{For } j = 1, \ldots, L \\ \quad \text{For } i = 1, \ldots, N_{j-1} \\ \qquad v^j_{2i-1} = (P_j v^{j-1})_{2i-1} + d^j_i \\ \qquad v^j_{2i} = 2v^{j-1}_i - v^j_{2i-1} \end{array} \right\}. \tag{2.7}$$

Observe that since $d^j_i = e^j_{2i-1} = -e^j_{2i}$, the consistency relation (2.5) implies that the computation of v^j_{2i} in (2.7) is equivalent to

$$v^j_{2i} = 2v^{j-1}_i - v^j_{2i-1} = (P_j v^{j-1})_{2i} - d^j_i = (P_j v^{j-1})_{2i} + e^j_{2i}. \tag{2.8}$$

Therefore (2.6) and (2.7) are just the repeated application of the decomposition and reassembling specified in (2.2)(a) and (2.2)(b). Thus (2.6) defines a multiscale transformation and (2.7) is the inverse transformation, whether or not the prediction operator is linear.

Next, we follow [4, 14, 15] to describe a class of *linear* prediction operators that leads to the $(1, M)$ branch of the Cohen-Daubechies-Feauveau family [7], which is biorthogonal

to the box function [11, 15]. This class is also considered in [6] within the lifting framework, where it is described as a particular case of Donoho's average interpolation [9].

Given an integer $s \geq 1$, for each $1 \leq i \leq N_{j-1}$ we construct a polynomial, $p_i(x)$, of degree 2s such that

$$\frac{1}{h_{j-1}} \int_{x_{i+l-1}^{j-1}}^{x_{i+l}^{j-1}} p_i(x)dx = v_{i+l}^{j-1}, \qquad \text{for } l = -s, \ldots, s. \tag{2.9}$$

There are various ways to prove that $p_i(x)$ in (2.9) always exists and it is uniquely defined by the $2s+1$ conditions in (2.9) [1, 9, 14]. Then we define

$$(P_j v^{j-1})_{2i} = \frac{1}{h_j} \int_{x_{2i-1}^j}^{x_{2i}^j} p_i(x)dx, \qquad (P_j v^{j-1})_{2i-1} = \frac{1}{h_j} \int_{x_{2i-2}^j}^{x_{2i-1}^j} p_i(x)dx. \tag{2.10}$$

The prediction operator defined by (2.10) is data-independent, hence linear, and it clearly satisfies the consistency relation (2.5). It can be shown that the multiscale transformations (2.6) and (2.7) for this class of prediction operators turns out to be the $(1, M = 2s+1)$ branch of the Cohen-Daubechies-Feauveau family.

A nonlinear prediction operator is obtained if we construct $p_i(x)$ in a data-dependent way. An example of nonlinear multiresolution transformation constructed in this fashion is considered in [14, 4, 2], where a nonlinear ENO-type technique (Essentially Non Oscillatory, see [16]) is used to construct $p_i(x)$. The key idea, which is in essence common to the approach used in designing nonlinear filter banks, is to avoid using data across an edge for the prediction step.

The ENO nonlinear technique is better described if we associate to each polynomial piece $p_i(x)$ a *stencil*, $\mathcal{S}_i$, which is the set of indices of the values used to define $p_i(x)$. In the linear case $\mathcal{S}_i = \{i-s, \ldots, i+s\}$; the stencil is *independent* of the data set $\{v^{j-1}\}$ and, as a consequence, P_j is a linear operator. In the ENO technique described in [16], the selection of stencil is made in a data-dependent way using the divided differences of the data as a measure of its smoothness. Large divided differences occur when considering data across an edge, while divided (or undivided) differences of data on smoother regions tend to be smaller in size.

The information contained in the divided differences is then used to decide what is $\mathcal{S}_i$ for each i, with the only restriction that $i \in \mathcal{S}_i$ (to satisfy the consistency requirement (2.5)). We follow [4] and consider all polynomial pieces of the same degree. In our case $\#\mathcal{S}_i = 2s$, but in principle one could decide to lower the degree of $p_i(x)$, or that of some of its neighbours, whenever an edge-detection mechanism finds an edge at the ith interval. By lowering the degree of some polynomial pieces close to an edge, one can avoid crossing the edge in the prediction step, as much as possible. This option is closely related to the nonlinear multiscale transformation considered in [6] (within the Lifting framework), where the nonlinearity comes in from adaptively choosing from the $(1, M)$ family of linear filters.

Once $\mathcal{S}_i$ is determined ($i \in \mathcal{S}_i$), $p_i(x)$ can be uniquely determined when degree $p_i(x) =$

$\#\mathcal{S}_i$ [1] so that

$$\frac{1}{h_{j-1}} \int_{x_{m-1}^{j-1}}^{x_m^{j-1}} p_i(x)dx = v_m^{j-1} \qquad \text{for } m \in \mathcal{S}_i, \tag{2.11}$$

and the prediction operator is then defined by (2.10).

One can be slightly more 'sophisticated' in the design of the polynomial pieces. The *Subcell Resolution* technique [4, 13] allows to account for discontinuities within a cell as follows. If an edge is detected in the ith cell, the polynomial piece $p_i(x)$ is discarded and substituted by its left and right neighbours, $p_{i+1}(x)$ and $p_{i-1}(x)$, assuming that their respective stencils do not intersect, i.e. $\mathcal{S}_{i-1} \bigcap \mathcal{S}_{i+1} = \emptyset$. At a true one-dimensional edge (a jump) on the ith cell, the function

$$F(y) = \frac{1}{h_j} \int_{x_{2i-1}^j}^{y} p_{i-1}(x)dx + \frac{1}{h_j} \int_{y}^{x_{2i}^j} p_{i+1}(x)dx$$

will have a zero on the ith cell [13], say η, and the location of η is used to substitute the polynomial piece $p_i(x)$ by the discontinuous piecewise polynomial function

$$q_i(x) = \begin{cases} p_{i-1}(x) & x \leq \eta, \\ p_{i+1}(x) & x > \eta. \end{cases} \tag{2.12}$$

The prediction operator is again defined by (2.10) at *nonsingular* cells (cells in which no edge has been detected), while at the *singular* cell

$$(P_j v^{j-1})_{2i} = \frac{1}{h_j} \int_{x_{2i-1}^j}^{x_{2i}^j} q_i(x)dx, \qquad (P_j v^{j-1})_{2i-1} = \frac{1}{h_j} \int_{x_{2i-2}^j}^{x_{2i-1}^j} q_i(x)dx.$$

In practice it is unnecessary to compute explicitly the value of η; only its location with respect with x_{2i-1}^j is needed, which can be found by a sign check. We refer the reader to [4] (and references therein) for specific details on this technique, in particular on the detection mechanism, and on its performance.

3 The question of stability: Error control versus synchronization, with numerical examples

Lossy coding schemes introduce errors into the transform coefficients, and it becomes crucial that the nonlinearities do not unduly amplify these errors. In lossy compression the decoder only has the quantized detail coefficients. If we use a nonlinear prediction operator (whether it is constructed as described in the previous section or based on locally adapted filters, as in [6] within the Lifting framework), the quantization errors in coarse scales could cascade across the scale ladder and cause a series of incorrect choices (either on the filters or on the stencils) leading to serious reconstruction errors.

To avoid incorrect choices in the prediction step, whether within Harten's or the Lifting framework, one would need to send side information on which filter was used (Lifting) or what was the interpolatory stencil (Harten's). This is clearly inappropriate when trying to design a compression scheme. One way to avoid storing (and sending) side information is to somehow *synchronize* the nonlinear prediction operators in the encoder

and the decoder, so as to ensure that at a given spatial location on a given scale, the prediction operator will select the same stencil (filter bank), both in the encoding and the decoding steps.

Within the Lifting framework, synchronization is achieved in [6] by changing the typical Split-Predict-Update steps to Split-Update-Predict. In doing so, it is possible to base the *choice* of predictor directly on already 'quantized data', thus synchronizing the nonlinear decisions made by the encoder and the decoder.

Within Harten's framework, synchronization is just a consequence of a strategy that is designed to *fully control* the compression error. Because the main design tool in Harten's framework for multiresolution is a reconstruction technique, and because A. Harten had already worked with nonlinear reconstruction techniques in the context of the numerical simulation for hyperbolic conservation laws, so-called *Error-Control* (EC) strategies can be found already in the early papers of Harten on multiresolution [14].

Harten's mechanism to control the global accumulated error is based on a modification of the direct multiscale transformation, M, that ensures a prescribed tolerance on the global prediction errors (explicit error bounds can be found in [4, 13]). The modified transformation incorporates the quantizer to the direct multiscale transformation in such a way that the prediction operator in the encoder also acts on already 'quantized' data, hence synchronization is achieved because the nonlinear prediction operators both in M and M^{-1} work on the *same* set of discrete data at each resolution level.

To illustrate the effect of the different techniques, we take a particular nonlinear prediction operator, a third order ENO reconstruction technique with Subcell Resolution, as described in last section. We denote by M_{SR} the multiscale transformation (2.6), while M_{SR}^{M} denotes the EC modified transform as described in [2, 4], and M_{SR}^{S} a multiscale transformation in which only synchronization is enforced, as proposed in [6]. The quantization step is carried out as follows:

$$\mathbf{qu}(d^j) = 2\epsilon_j \text{round}\left[d^j/(2\epsilon_j)\right]$$

and it is incorporated to the direct transformation in M_{SR}^{M} and M_{SR}^{S} (see [2, 6] for specific details), while in M_{SR} it is applied to the scale coefficients obtained after the transformation. In the numerical tests we report, we take $\epsilon_L = 8$ with $L = 4$ and $\epsilon_j = \epsilon_{j+1}/2$.

We consider two different images: the familiar image of Lena as an example of a 'real' image, and a purely geometrical image, to which texture has been added, as in [6].

After the direct transformation (plus the quantization step) has taken place, a lossless Lempel-Ziv compression algorithm is applied to reduce the size of the transformed image, then a *compression ratio* is computed as the number of bits of the compressed representation over the number of bits of the original image. To recover the original image, we undo the lossless compression and transform back using (2.7) in all three cases. The full compression algorithm is identified in each case by an acronym, 'ST' for M_{SR}, 'EC' for M_{SR}^{M} and 'SYNC' for M_{SR}^{S}.

In Tables 1 and 2 we compile a number of quantities that measure the 'quality' of the reconstructed image, and therefore the robustness and reliability of each multiresolution-based compression algorithm, the magnitude of the global compression error, measured

Method	$\|\|\cdot\|\|_\infty$	$\|\|\cdot\|\|_1$	$\|\|\cdot\|\|_2$	r_c	entropy
ST	258	5.71	9.08	11.3:1	.6449
SYNC	195	6.45	9.82	7.9:1	.8875
EC	25.4	4.47	5.73	9.7:1	.6850

TAB. 1. Geometrical image.

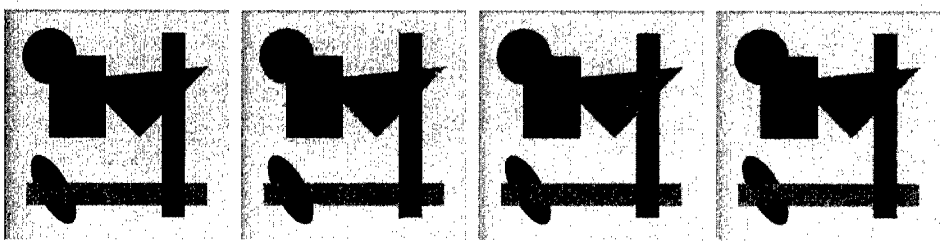


FIG. 1. Geometrical image: (a) original, (b) ST, (c) EC, (d) SYNC.

in various norms, the compression rate r_c and the entropy of the transformed image. The reconstructed images in both cases can be observed in Figures 1 and 2.

It can be clearly observed that the absence of any type of synchronization procedure can lead to a very poor reconstructed image. Synchronization *only* improves the quality, but is not as robust as the full EC mechanism, designed in this case to enforce a certain error bound in the 2-norm (as observed in Tables 1 and 2, the 2-norm of the global error is kept below $\epsilon_L = 8$). It is worth mentioning that the compression rate and the entropy of the compressed data are all very close, however the visual quality of the reconstructed image is significantly better for the EC compression algorithm.

Bibliography

1. R. Abgrall and A. Harten. Multiresolution representation in unstructured meshes. *SIAM J. Numer. Anal.* **35**, 2128–2146 (electronic), 1998.
2. S. Amat, F. Aràndiga, A. Cohen, and R. Donat. Tensor product multiresolution analysis with error control for compact image representation. Submitted to Signal Processing, 2000.
3. S. Amat, F. Aràndiga, A. Cohen, R. Donat, G. García, and M. Von Oehsen. Data compression with ENO schemes. *Applied and Computational Harmonic Analysis* **11**, 273–288, 2001.
4. F. Aràndiga and R. Donat. Nonlinear multi-scale decompositions: The approach of A. Harten. *Numer. Algorith.* **23**, 175–216, 2000.
5. F. Aràndiga, R. Donat, and A. Harten. Multiresolution based on weighted averages of the hat function II: Nonlinear reconstruction operators. *SIAM J. Sci. Comput.* **20**, 1053–1093, 1999.
6. R. L. Claypoole, G. Davis, W. Sweldens, and R. Baraniuk. Nonlinear wavelet transforms for image coding via lifting scheme. *submitted to IEEE Trans. on Image*

Method	$\|\cdot\|_\infty$	$\|\cdot\|_1$	$\|\cdot\|_2$	r_c	entropy
ST	318	5.66	10.59	8.8:1	.8261
SYNC	277	5.97	10.56	7.5:1	.9430
EC	26.4	3.59	4.84	8.2:1	.8704

TAB. 2. Lena.



FIG. 2. Lena: (a) original, (b) ST, (c) EC, (d) SYNC.

Processing., 1999.

7. A. Cohen, I. Daubechies, and J.C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Comm. Pure Applied Math.* **45**, 485–560, 1992.
8. R. L. de Quieroz, D. A. Florêncio, and R. W. Schafer. Non-expansive pyramid for image coding using a non-linear filter bank. *IEEE Trans. Image Processing* **7**, 246–252, 1998.
9. D. L. Donoho. Interpolating wavelet transforms. Technical report, Department od Statistics, Stanford University, 1992.
10. D. L. Donoho and Thomas P.Y. Yu. Nonlinear pyramid transforms based on median-interpolation. *SIAM Journal on Mathematical Analysis* **31**, 1030–1061, 2000.
11. M. Guichaoua. *Analyses Multirésolution Biorthogonales associées à la Résolution d'Equations aux Dérivées Partielles.* PhD thesis, Ecole Superieure de Mecanique de Marseille, Universite de la Mediterranee Aix-Marseille II, 1999.
12. F.J. Hampson and J.C. Pesquet. A nonlinear subband decomposition with perfect reconstruction. In *Proce. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, 1996.
13. A. Harten. ENO schemes with subcell resolution. *J. Comput. Phys.* **83**, 148–184, 1989.
14. A. Harten. Discrete multiresolution analysis and generalized wavelets. *J. of Applied Num. Math.* **12**, 153–193, 1993.
15. A. Harten. Multiresolution representation of data: A general framework. *SIAM J. Numer. Anal.* **33**, 1205–1256, 1996.
16. A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy. Uniformly high-order accurate essentially nonoscillatory schemes. III. *J. Comput. Phys.*, **7** 1231–303, 1987.
17. W. Sweldens. The lifting scheme: a custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.* **3**, 186–200, 1996.

Splines: a new contribution to wavelet analysis

Amir Z. Averbuch, and Valery A. Zheludev

School of Computer Science, Tel Aviv University, Israel.
amir@math.tau.ac.il, zhel@post.tau.ac.il

Abstract

We present a new approach to the construction of biorthogonal wavelet transforms using polynomial splines. The construction is performed in a "lifting" manner and we use interpolatory, as well as local quasi-interpolatory and smoothing splines as predicting aggregates in this scheme. The transforms contain some scalar control parameters which enable their flexible tuning in either time or frequency domains. The transforms are implemented in a fast way. They demonstrated efficiency in application to image compression.

1 Introduction

Until recently, two methods have been used for the construction of wavelet schemes using splines. One is to construct orthogonal and semi-orthogonal wavelets in the spline spaces (Battle-Lemarié [2, 7], Chui-Wang [6], Unser-Aldroubi-Eden [12]). Another way was introduced by Cohen, Daubechies and Feauveau [3] who constructed symmetric compactly supported spline wavelets whose duals, remaining compactly supported and symmetric, do not belong to a spline space. However, since the introduction of the lifting scheme for the design of wavelet transforms [11], a new way was opened to use splines as a tool for devising a full discrete scheme of wavelet transforms. Namely, various splines can be employed as predicting aggregates in lifting constructions.

2 Lifting scheme of biorthogonal wavelet transform

The sequences $\{a(k)\}_{k=-\infty}^{\infty}$, which belong to the space l_1, we call the discrete-time signals. The $z-$transform of a signal $\{a(k)\}$ is defined as follows: $a(z) = \sum_{k=-\infty}^{\infty} z^{-k}\, a(k)$. Throughout the paper we assume that $z = e^{i\omega}$. We introduce a family of biorthogonal wavelet-type transforms that operate on the signal $\mathbf{x} = \{x(k)\}_{k=-\infty}^{\infty}$, which we construct through lifting steps.

The lifting scheme for the wavelet transform of a signal can be implemented in primal or dual modes. For brevity we consider only the primal mode.

Decomposition Generally, the primal lifting scheme for decomposition of signals consists of three steps: 1. Split. 2. Predict. 3. Update or lifting.

SPLIT - We split the array $\mathbf{x}$ into even and odd sub-arrays:

$$\mathbf{e_1} = \{e_1(k) = x(2k)\}, \quad \mathbf{d_1} = \{d_1(k) = x(2k+1)\},\ k \in \mathbb{Z}.$$

PREDICT - We use the even array $\mathbf{e_1}$ to predict the odd array $\mathbf{d_1}$ and redefine the array $\mathbf{d_1}$ as the difference between the existing array and the predicted one. To be specific, we apply some filter with transfer function $zU(z)$ to the sequence $\mathbf{e_1}$ and predict the function $d_1(z^2)$ which is the z^2-transform of $\mathbf{d_1}$. The z^2-transform of the new $d-$array is defined as follows:

$$d_1^u(z^2) = d_1(z^2) - zU(z)e_1(z^2). \tag{2.1}$$

From now on the superscript u means an *update* operation of the array. Obviously, the prediction $zU(z)e_1(z^2)$ should approximate $d_1(z^2)$ well.

LIFTING - We update the even array using the new odd array:

$$e_1^u(z^2) = e_1(z^2) + \beta(z)z^{-1}\, d_1^u(z^2). \tag{2.2}$$

Generally, the goal of this step is to eliminate aliasing which appears while downsampling the original signal $\mathbf{x}$ into $\mathbf{e_1}$. Further on we will discuss how to achieve this effect by a proper choice of the filter β.

Reconstruction The reconstruction of the signal $\mathbf{x}$ from the arrays $\mathbf{e_1^u}$ and $\mathbf{d_1^u}$ is implemented in reverse order: 1. Undo Lifting. 2. Undo Predict. 3. Unsplit.

UNDO LIFTING - We restore the even array: $e_1(z^2) = e_1^u(z^2) - \beta(z)z^{-1}\, d_1^u(z^2)$.

UNDO PREDICT - We restore the odd array: $d_1(z^2) = d_1^u(z^2) + zU(z)e_1(z^2)$.

UNSPLIT - The last step represents the standard restoration of the signal from its even and odd components. In the $z-$domain this is $x(z) = e_1(z^2) + z^{-1}d_1(z^2)$.

The lifting scheme presented above, yields an efficient algorithm for the implementation of the forward and backward transform of $\mathbf{x} \longleftrightarrow \mathbf{e_1^u} \cup \mathbf{d_1^u}$. These operations can be interpreted as a transformation of the signal by a filter bank that possesses the perfect reconstruction properties and it is associated with the biorthogonal pairs of bases in the space of discrete-time signals. These basis signals are synthesis and analysis wavelets. Further steps of the transform are implemented in an iterative way by the same lifting operations.

3 Polynomial splines

We will construct polynomial splines of various kinds using the even subarray of a signal, calculate their values in the midpoints between nodes and use these values for prediction of the odd array. In this section we discuss some properties of such splines and derive the corresponding filters U.

3.1 $B-$splines

The central $B-$spline of first order on the grid $\{kh\}$ is defined as follows:

$$M_1^h(x) = \begin{cases} 1/h & \text{if } x \in [-h/2, h/2], \\ 0 & \text{elsewhere.} \end{cases}$$

The central $B-$spline of order p is the convolution $M_h^p(x) = M_h^{p-1}(x) * M_h^1(x) \quad p \geq 2$. Note that the $B-$spline of order p is supported at the interval $(-ph/2, ph/2)$. It is positive within its support and symmetric around zero. The nodes of $B-$splines of even orders are located at points $\{kh\}$ and of odd orders at points $\{h(k+1/2)\}$, $k \in \mathbb{Z}$. It is readily

verified that $hM_h^p(hx) = M^p(x)$, where $M^p(x) := M_1^p(x)$. Let

$$\mathbf{u}^p := \{hM_h^p(hk) = M^p(k)\}, \quad \text{and} \quad \mathbf{w}^p := \{hM_h^p\left(h(k+1/2)\right) = M^p\left(k+1/2\right)\}, \; k \in \mathbb{Z}. \tag{3.1}$$

Due to the compact support of $B-$splines, these sequences are finite. We will use for our constructions only splines of odd orders $p = 2r-1$. In Table 1 we present the sequences for initial values r which are of practical importance.

k	-3	-2	-1	0	1	2	3
$\mathbf{u}^3 \times 8$	0	0	1	6	1	0	0
$\mathbf{u}^5 \times 384$	0	1	76	230	76	1	0
$\mathbf{w}^3 \times 2$	0	0	1	1	0	0	0
$\mathbf{w}^5 \times 24$	0	1	11	11	1	0	0

TAB. 1. Values of the sequences $\mathbf{u}^p$ and $\mathbf{w}^p$.

We need the z^2-transforms of the sequences $\mathbf{u}^p$ and $\mathbf{w}^p$:

$$u^p(z^2) := \sum_{k=-\infty}^{\infty} z^{-2k} u^p(k), \quad w^p(z^2) := \sum_{k=-\infty}^{\infty} z^{-2k} w^p(k).$$

These functions are Laurent polynomials, and are called the Euler-Frobenius polynomials [10].

Proposition 3.1. ([9]) *On the circle $z = e^{i\omega}$ the Laurent polynomials $u^p(z^2)$ are strictly positive. Their roots are all simple and negative. Each root ζ can be paired with a dual root θ such that $\zeta\,\theta = 1$. Thus, if $p = 2r+1$ is odd, then $u^p(z^2)$ can be represented as follows:*

$$u^p(z^2) = \prod_{n=1}^{r} \frac{1}{\gamma_n}(1+\gamma_n z^2)(1+\gamma_n z^{-2}), \quad 0 < \gamma_n < 1. \tag{3.2}$$

We denote

$$U_i^p(z) := z^{-1}\frac{w^p(z^2)}{u^p(z^2)}. \tag{3.3}$$

Proposition 3.2 *The rational functions $U_i^p(z)$ are real-valued and $U_i^p(-z) = -U_i^p(z)$. If $p = 2r+1$ is odd then*

$$1 - U_i^p(z) = \frac{(\alpha-2)^{r+1}\xi_r(\alpha)}{u^p(z^2)}, \quad 1 + U_i^p(z) = \frac{(-\alpha-2)^{r+1}\xi_r(-\alpha)}{u^p(z^2)}, \tag{3.4}$$

where $\alpha := z + z^{-1}$ and $\xi_r(\alpha)$ is a polynomial of degree $r-1$.

3.2 Interpolatory splines

The shifts of $B-$splines form a basis in the space $\mathbf{S}_h^p$ of splines of order p on the grid kh. Namely, any spline $S_h^p \in \mathbf{S}_h^p$ has the following representation:

$$S_h^p(x) = h\sum_l q(l)\, M_h^p(x - lh). \tag{3.5}$$

Let $\mathbf{q} := \{q(l)\}$, and $q(z^2)$ be the z^2-transform of $\mathbf{q}$. We introduce also the sequences $\mathbf{s}^p := h\{S_h^p(hk) = S_1^p(k)\}$ and $\mathbf{m}^p := \{S_h^p(h(k+1/2)) = S_1^p(k+1/2)\}$ of values of the spline on the grid points and on the midpoints. Let $s^p(z^2)$ and $m^p(z^2)$ be the corresponding z^2-transforms. We have

$$S_1^p(k) = \sum_l q(l)\, M_h^p(k-l), \quad \text{and} \quad S_1^p\left(k+\frac{1}{2}\right) = \sum_l q(l)\, M_h^p\left(k-l+\frac{1}{2}\right). \tag{3.6}$$

Respectively, $s^p(z^2) = q(z^2)u(z^2)$, and $m^p(z^2) = q(z^2)w(z^2)$.

From these formulae we can derive expression for the coefficients of a spline which interpolates a given sequence $\mathbf{e} := \{e(k)\}$ at grid points:

$$hS_h^p(hk) = e(k),\ k \in \mathbb{Z}, \Longleftrightarrow q(z^2)u^p(z) = e(z^2) \Longleftrightarrow q(z^2) = \frac{e(z^2)}{u^p(z^2)}. \tag{3.7}$$

The z^2-transform of the sequence $\mathbf{m}^p$ is:

$$m^p(z^2) = q(z^2)w^p(z^2) = zU_i^p(z)e(z^2). \tag{3.8}$$

Our further construction exploits the super-convergence property of the interpolatory splines of odd orders (even degrees).

Theorem 3.3. ([13]) *Let a function $f \in L^1(-\infty,\infty)$ have $p+1$ continuous derivatives and let $S_h^p \in \mathbf{S}_h^p$ interpolate f on the grid $\{kh\}$. Denote $\tilde{f}_k = f((k+1/2)h)$. Then in the case of odd $p = 2r+1$, the following asymptotic relation holds.*

$$S_h^p(h(k+1/2)) = \tilde{f}_k - h^{2r+2} f^{(2r+2)}(h(k+1/2))(2r+1)\frac{b_{2r+2}(0) - b_{2r+2}(\frac{1}{2})}{(2r+2)!} + o(h^{2r+2} f^{(2r+2)}), \tag{3.9}$$

where $b_s(x)$ is the Bernoulli polynomial of degree s.

Recall, that in general the interpolatory spline of order $2r+1$ approximates the function f with accuracy of h^{2r+1}. Therefore, we may claim that $\{(k+1/2)h\}$ are points of super-convergence of the spline S_h^p. Note, that the spline of order $2r+1$, which interpolates the values of a polynomial of degree $2r$, coincides with this polynomial. However, the spline of order $2r+1$ which interpolates the values of a polynomial of degree $2r+1$ on the grid $\{kh\}$ restores the values of this polynomial at the mid-points $\{(k+1/2)h\}$. This property will result in the vanishing moments property of the wavelets to be constructed later.

3.3 Quasi-interpolatory splines

We can see from (3.7) and (3.8) that in order to find values at the midpoints of the spline interpolating the signal $\mathbf{e}$, the signal has to be filtered with the filter whose transfer function is $zU_i^p(z)$. This filter has infinite impulse response (IIR). However, the property of super-convergence at the midpoints is not an exclusive attribute of the interpolatory splines. It is also inherent to the so called local quasi-interpolatory splines of odd orders, which can be constructed using finite impulse response (FIR) filtering.

Definition 3.4 *Let the function f have p continuous derivatives and $\mathbf{f} := \{f_k = f(hk)\}$, $k \in \mathbb{Z}$. The spline $S_h^p \in \mathbf{S}_h^p$ of order p given by (3.5) is said to be the local*

quasi-interpolatory spline if the array $\mathbf{q}$ *of its coefficients is derived by FIR filtering the array of samples* $\mathbf{f}$

$$\mathbf{q}(z^2) = \Gamma(z^2)\mathbf{f}(z^2), \tag{3.10}$$

where $\Gamma(z^2)$ *is a Laurent polynomial, and the difference* $|f(x) - S_h^p(x)| = O(f^{(p)}h^p)$. *If* f *is a polynomial of degree* $p-1$, *then the spline* $S_h^p(x) \equiv f(x)$.

If $\mathbf{w}^p$ is the sequence defined in (3.1) then the midpoint values $\mathbf{m}^p$ are produced by the following FIR filtering of the array of samples $\mathbf{f}$: $\mathbf{m}^p(z^2) = zU^p(z)\mathbf{f}(z^2)$, $U^p(z) := z^{-1}\Gamma(z^2)w^p(z^2)$. Explicit formulas for the construction of quasi-interpolatory splines as well as the estimations of the differences were established in [13]. In the present work we are interested in splines of odd orders $p = 2r+1$. There are many FIR filters which generate quasi-interpolatory splines but only one filter of minimal length $2r+1$ for each order $p = 2r+1$. Let $\lambda(z) := z^{-2} - 2 + z^2$.

Theorem 3.5 *A quasi-interpolatory spline of order* $p = 2r+1$ *can be produced by filtering (3.10) with filters* Γ *of length no less than* $2r+1$. *There exists a unique filter* Γ_m^r *of length* $2r+1$ *which produces the minimal quasi-interpolatory spline* $\tilde{S}_h^{2r+1}(x)$. *Its transfer function is:*

$$\Gamma_m^r(z^2) = 1 + \sum_{k=1}^{r} \beta_k^r \lambda^k(z), \quad \left(\frac{2\arcsin t/2}{t}\right)^{2r+1} = \sum_{k=0}^{\infty} (-1)^k \beta_k^r t^{2k}. \tag{3.11}$$

If the function f *has* $2r+3$ *derivatives then the following asymptotic relations hold for the midpoint values of the minimal quasi-interpolatory spline of odd order:*

$$\tilde{S}_h^{2r+1}(h(k+1/2)) = f(h(k+1/2)) + h^{2r+2} f^{(2r+2)}(h(k+1/2))A^r + O(f^{(2r+3)}h^{2r+3}),$$
$$A^r := \frac{(2r+1)b_{2r+2}(0)}{(2r+2)!} - \beta_{r+1}^r, \tag{3.12}$$

where $b_s(x)$ *is the Bernoulli polynomial of degree* s.

This implies that the super-convergence property is similar to that of the interpolatory splines. The asymptotic representation (3.12) provides tools for custom design of predicting splines retaining or even enhancing the approximation accuracy of the minimal spline at the midpoints.

Proposition 3.6 *If the coefficients of the spline* $S_{h,\rho}^{2r+1} \in \mathbf{S}_h^{2r+1}$ *of order* $2r+1$ *are derived as in (3.10) using the filter* Γ_ρ^r *of length* $2r+3$, *with the transfer function* $\Gamma_\rho^r(z^2) = \Gamma_m^r(z^2) + \rho\lambda^{r+1}(z)$, *then the spline restores polynomials of degree* $2r+1$ *at the midpoints between nodes, for any real value* ρ. *However, if* $\rho = -A^r$ *then the spline restores polynomials of degree* $2r+3$.

If the parameter ρ is chosen such that $\rho = (-1)^r|\rho|$ then the spline $S_{h,\rho}^{2r+1}$ possesses the smoothing property [14].

3.4 Examples

3.4.1 *Quadratic splines*

Interpolatory spline Let $\alpha = z^{-1} + z$. Then

$$U_i^1(z) = \frac{4\alpha}{z^2 + 6 + z^{-2}}, \text{ and } 1 - U_i^1(z) = \frac{(\alpha - 2)^2}{z^{-2} + 6 + z^2},$$

Minimal spline The filters are

$$\begin{aligned} \Gamma_m^1(z^2) &= 1 - \frac{1}{8}\lambda(z),\ U_m^1(z) = \frac{-z^{-3} + 9z^{-1} + 9z - z^3}{16}, \\ \text{and } 1 - U_m^1(z) &= \frac{(\alpha - 2)^2(z^{-1} + 4 + z)}{16}. \end{aligned}$$

Extended spline

$$\Gamma_e^1(z) = \Gamma_m^1(z^2) + \frac{1}{64}\lambda^2(z),\ U_e^1(z) = \frac{3z^{-5} - 25z^{-3} + 150z^{-1} + 150z - 25z^3 + 3z^5}{256},$$
$$\text{and } 1 - U_e^1(z) = \frac{(\alpha - 2)^3(3z^{-2} + 18z^{-1} + 38 + 18z + 3z^2)}{256},$$

Remark 3.7 *In [5] Donoho presented a scheme where an odd sample is predicted by the value in the central point of the polynomial of odd degree which interpolates adjacent even samples. One can observe that our filter U_m^1 coincides with the filter derived by Donoho's scheme using the cubic interpolatory polynomial. The filter U_e^1 coincides with the filter derived using the interpolatory polynomial of fifth degree. On the other hand, the filter U_i^1 is closely related to the commonly used Butterworth filter [8]. Namely, in this case the filter transfer functions $\Phi_i^{1,l}(z) := (1+U_i^1(z))/2$, $\Phi_i^{1,h}(z) := (1-U_i^1(z))/2$ coincide with magnitude squared of the transfer functions of the discrete-time low-pass and high-pass half-band Butterworth filters of order 4, respectively.*

3.4.2 *Splines of fifth order (fourth degree)*

Interpolatory spline

$$U_i^2(z) = \frac{16(z^3 + 11z + 11z^{-1} + z^{-3})}{z^4 + 76z^2 + 230 + 76z^{-2} + z^{-4}},\ 1 - U_i^2(z) = \frac{(\alpha - 2)^3(\alpha - 10)}{z^4 + 76z^2 + 230 + 76z^{-2} + z^{-4}}.$$

Minimal spline The filter is

$$U_m^2(z) = \frac{47(z^{-7} + z^7) + 89(z^{-5} + z^5) - 2277(z^{-3} + z^3) + 15965\alpha}{27648}.$$

4 Wavelet transforms using spline filters

4.1 Choosing the filters for the *lifting* step

In the previous section we presented a family of filters U for the *predicting* step which were originated from splines of various types. But, as it is seen from (2.2), to accomplish the transform we have to define the filter β. There is a remarkable freedom in the choice of these filters. The only requirement needed to guarantee a perfect reconstruction property of the transform is that $\beta(-z) = \beta(z)$. In order to make synthesis and analysis filters

similar in their properties, we choose $\beta(z) = \breve{U}(z)/2$, where $\breve{U}$ means one of filters U presented above. In particular, $\breve{U}$ may coincide with the filter U which was used for the prediction.

We say that a wavelet ψ has m vanishing moments if the following relations hold: $\sum_{k\in\mathbb{Z}} k^s\psi(k) = 0, \quad s = 0, 1, \ldots, m-1$.

Proposition 4.1 *Suppose the filters $U(z)$ and $\beta(z) = \breve{U}(z)/2$ are used for the predicting and lifting steps, respectively. If $1 - U(z)$ contains the factor $(z - 2 + 1/z)^r$ then the high-frequency analysis wavelets $\tilde{\psi}^1$ have $2r$ vanishing moments. If, in addition $1 - \breve{U}(z)$ contains the factor $(z-2+1/z)^p$ then the synthesis wavelet ψ^1_β has $2q$ vanishing moments, where $q = \min\{p, r\}$.*

4.2 Implementation of the transforms

Suppose, we have chosen the filter $\beta = \breve{U}/2$. The functions $zU(z)$ and $z\breve{U}(z)$ depend on z^2 and we write $F(z^2) := zU(z)$ and $\breve{F}(z^2) := z\breve{U}(z)$. Then the decomposition procedure is (see (2.1), (2.2)):

$$d_1^u(z) = d_1(z) - F(z)e_1(z), \quad e_1^u(z) = e_1(z) + \frac{1}{2z}\breve{F}(z)\, d_1^u(z). \tag{4.1}$$

Equation (4.1) means that in order to obtain the detail array $\mathbf{d}_1^u$, we must process the even array $\mathbf{e}_1$ with the filter F, with transfer function $F(z)$, and extract the filtered array from the odd array $\mathbf{d}_1$. In order to obtain the smoothed array $\mathbf{e}_1^u$, we must process the detail array $\mathbf{d}_1^u$ with the filter Φ that has the transfer function $\Phi(z) = z^{-1}\breve{F}(z)/2$ and add the filtered array to the even array $\mathbf{e}_1$. But the filter Φ differs from $\breve{F}_r/2$ only by one-sample delay and it operates similarly. Thus, both operations of the decomposition are, in principle, identical. For the reconstruction the same operation is conducted in reverse order.

Therefore, it is sufficient to outline the implementation of the filtering with the function $F(z)$.

Implementation of FIR filters originating from local splines is straightforward and, therefore we only make a few remarks on IIR filters originating from interpolatory splines. A detailed description can be found in [1]. Equations (3.2) and (3.3) imply that, while the interpolatory spline of order $2r+1$ is used, the transfer function $F(z) = P(z)/\prod_{n=1}^{r}\frac{1}{\gamma_n}(1+\gamma_n z)(1+\gamma_n z^{-1})$, where $P(z)$ is the Laurent polynomial. It means that the IIR filter F can be split into a cascade consisting of a FIR filter with the transfer function $P(z)$, r elementary causal recursive filters denoted by $\overrightarrow{R(n)}$, and r elementary anti-causal recursive filters, denoted by $\overleftarrow{R(n)}$. The causal and anti-causal filters operate as follows:

$$\mathbf{y} = \overrightarrow{R(n)}\mathbf{x} \Longleftrightarrow y(l) = x(l) + \gamma_n y(l-1), \quad \mathbf{y} = \overleftarrow{R(n)}\mathbf{x} \Longleftrightarrow y(l) = x(l) + \gamma_n y(l+1).$$

Example 4.2 (Example of recursive filter) We present IIR filters derived from the interpolatory splines of third order.

Let $\gamma_1^1 = 3 - 2\sqrt{2} \approx 0.172$. Then

$$F_i^1(z) = 4\gamma_1^1 \frac{1+z}{(1+\gamma_1^1 z)(1+\gamma_1^1 z^{-1})}.$$

The filter can be implemented with the following cascade:

$$x_0(k) = 4\gamma_1^1(x(k)+x(k+1)), \quad x_1(k) = x_0(k)-\gamma_1^1 x_1(k-1), \quad y(k) = x_1(k)-\gamma_1^1 y(k+1).$$

Bibliography

1. A. Z. Averbuch, A. B. Pevnyi and V. A. Zheludev, Butterworth wavelets derived from discrete interpolatory splines: Recursive implementation, to appear in *Signal Processing*, www.math.tau.ac.il/∼amir (∼zhel).
2. G. Battle, A block spin construction of ondelettes. Part I. Lemarié functions, *Comm. Math. Phys.* **110** (1987), 601–615.
3. A. Cohen, I. Daubechies and J.-C. Feauveau, Biorthogonal bases of compactly supported wavelets, *Commun. on Pure and Appl. Math.* **45** (1992), 485–560.
4. I. Daubechies, *Ten lectures on wavelets*, SIAM, Philadelphia, PA, 1992.
5. D. L. Donoho, *Interpolating wavelet transform,* Preprint 408, Department of Statistics, Stanford University, 1992.
6. C. K. Chui and J. Z. Wang, On compactly supported spline wavelets and a duality principle, *Trans. Amer. Math. Soc.* **330** (1992), 903–915.
7. P. G. Lemarié, Ondelettes à localisation exponentielle, *J. de Math. Pures et Appl.* **67** (1988), 227–236.
8. A. V. Oppenheim, R. W. Shafer, *Discrete-time signal processing*, Englewood Cliffs, New York, Prentice Hall, 1989.
9. I. J. Schoenberg, Contribution to the problem of approximation of equidistant data by analytic functions, *Quart. Appl. Math.* **4** (1946), 112–141.
10. I. J. Schoenberg, Cardinal spline interpolation, CBMS **12**, SIAM, Philadelphia, 1973.
11. W. Sweldens, The lifting scheme: A custom design construction of biorthogonal wavelets, *Appl. Comput. Harm. Anal.* **3** (1996), 186–200.
12. M. Unser, A. Aldroubi and M. Eden, A family of polynomial spline wavelet transforms, *Signal Processing* **30** (1993), 141–162.
13. V. A. Zheludev, Local spline approximation on a uniform grid, *U.S.S.R. Comput. Math. & Math. Phys.* **27** (1987), 8–19.
14. V. A. Zheludev, Local smoothing splines with a regularizing parameter, *Comput. Math. & Math Phys.* **31** (1991), 193–211.

Knot removal for tensor product splines

T. Brenna

Dept. of Informatics, Univ. of Oslo, Oslo.
trondbre@ifi.uio.no

Abstract

Given a spline function as a B-spline expansion the object of knot removal is to remove as many knots as possible without perturbing the spline by more than a specified tolerance. In 1987 Lyche and Mørken proposed an efficient knot removal algorithm which determines both the number of remaining knots and their position automatically. In this paper we show how their method can be extended to knot removal techniques for multivariate tensor product splines. We propose a number of new strategies for removing as many knots as possible, and discuss some of the advantages and challenges posed by the special structure of tensor product splines.

1 Introduction

Given a spline function we are often interested in an approximate representation requiring less data. The object of knot removal is to remove as many knots as possible from a given spline without perturbing the spline by more than a given tolerance. An efficient knot removal strategy presented in [6] determines both the number of remaining knots and their location automatically. This strategy was later extended to parametric curves and surfaces in [5], and incorporated with various constraints such as monotonicity and convexity in [1]. An efficient implementation of knot removal for the special case of trilinear splines is given in [3]. In this paper we address some of the questions and problems arising when extending the knot removal technique to multivariate tensor product splines.

The outline of this paper is as follows. We start by fixing notation and presenting techniques for representing tensor product splines. We then proceed with generalizations of coefficient norms, approximation methods, methods for ranking the knots etc., as we review the central parts of the knot removal strategy. Two different ways of performing knot removal are given together with accompanying strategies for finding the desired approximations. We end the paper with two examples demonstrating various aspects of the knot removal techniques presented.

2 Notation

Let $\mathbf{d} = (d_k), \mathbf{m} = (m_k) \in \mathbb{Z}^s$ with $\mathbf{0} \leq \mathbf{d} < \mathbf{m}$ (component-wise) for some positive integer s. Also let $\mathbf{t}^k = \{t_i^k\}_{i=1}^{m_k+d_k+1}$ be a knot vector with $d_k + 1$ equal knots at both ends and with no knot value occurring more than $d_k + 1$ times, for $k = 1, \ldots, s$. In this paper we will treat the collection $\mathbf{t} = \{\mathbf{t}^k\}_{k=1}^s$ as a "single" knot vector with "length"

$\mathbf{m}+\mathbf{d}+\mathbf{1}$ defined to be the sum of the length of the knot vectors $\mathbf{t}^k$, $k = 1, \ldots, s$. Given such a knot vector we may form products of the basis functions associated with each individual knot vector $\mathbf{t}^k$. By letting

$$B_{\mathbf{i}}(\mathbf{x}) = B_{\mathbf{i},\mathbf{d},\mathbf{t}}(\mathbf{x}) = \prod_{k=1}^{s} B_{i_k,d_k,\mathbf{t}^k}(x_k) \quad \text{for} \quad \mathbf{1} \leq \mathbf{i} \leq \mathbf{m},$$

where $\mathbf{i} = (i_k) \in \mathbb{Z}^s$ and $\mathbf{x} = (x_k) \in \mathbb{R}^s$,we get a total of $\prod_{k=1}^{s} m_k$ new basis functions for the tensor product space $\mathbb{S}_{\mathbf{d},\mathbf{t}} = \overset{s}{\underset{k=1}{\otimes}} \mathbb{S}_{d_k,\mathbf{t}^k}$. In this paper we let $B_{i_k,d_k,\mathbf{t}^k}$ be the i_kth B-spline of degree d_k associated with $\mathbf{t}^k$, for $k = 1, \ldots, s$.

To represent an element of $\mathbb{S}_{\mathbf{d},\mathbf{t}}$ we use a variant of the classical Kronecker product of matrices. Recall that if $\mathbf{A} = (\mathbf{a_{i,j}})_{\mathbf{i=1,j=1}}^{\mathbf{m_1,n_1}} \in \mathbb{R}^{\mathbf{m_1,n_1}}$, $\mathbf{B} = (\mathbf{b_{i,j}})_{\mathbf{i=1,j=1}}^{\mathbf{m_2,n_2}} \in \mathbb{R}^{\mathbf{m_2,n_2}}$ then this product is given by $\mathbf{A} \otimes \mathbf{B} = (\mathbf{a_{i,j}B})_{\mathbf{i=1,j=1}}^{\mathbf{m_1,n_1}}$. In this paper we will use the "equivalent" product defined by $\mathbf{A} \otimes \mathbf{B} = (\mathbf{Ab_{i,j}})_{\mathbf{i=1,j=1}}^{\mathbf{m_2,n_2}}$, which gives a more convenient ordering of the matrix elements for our use. Also recall that for real matrices $\mathbf{A}$,$\mathbf{B}$,$\mathbf{C}$,$\mathbf{D}$ we have the following useful relations (assuming that the matrix products and inverses are defined) $(\mathbf{A}\otimes\mathbf{B})(\mathbf{C}\otimes\mathbf{D}) = (\mathbf{AC})\otimes(\mathbf{BD})$, $(\mathbf{A}\otimes\mathbf{B})^{-1} = \mathbf{A}^{-1}\otimes\mathbf{B}^{-1}$ and $\mathbf{A}\otimes\mathbf{B} = \mathbf{P_1}(\mathbf{B}\otimes\mathbf{A})\mathbf{P_2}$, for some permutation matrices $\mathbf{P_1}$ and $\mathbf{P_2}$. In addition we have that the product $\mathbf{A}\otimes\mathbf{B}$ will have linearly independent columns, provided the same holds for $\mathbf{A}$ and $\mathbf{B}$. For further properties of the Kronecker product we refer to [4].

An element

$$f(\mathbf{x}) = \sum_{i_1=1}^{m_1} \cdots \sum_{i_s=1}^{m_s} f_{i_1,..,i_s} \prod_{k=1}^{s} B_{i_k,d_k,\mathbf{t}^k}(x_k) = \sum_{\mathbf{i}\leq\mathbf{m}} f_{\mathbf{i}} B_{\mathbf{i},\mathbf{d},\mathbf{t}}(\mathbf{x}) \in \mathbb{S}_{\mathbf{d},\mathbf{t}}$$

can now be written

$$f(\mathbf{x}) = \mathbf{B_t^T f},$$

where $\mathbf{B_t} = \overset{\mathbf{s}}{\underset{\mathbf{k=1}}{\otimes}} \mathbf{B_{t^k}}$ with $\mathbf{B_{t^k}} = (\mathbf{B_{1,d_k,t^k}}, \ldots, \mathbf{B_{m_k,d_k,t^k}})^{\mathbf{T}}$ for $k = 1, \ldots, s$. Here $\mathbf{f}$ is a vector containing the B-spline coefficients $\mathbf{F} = (\mathbf{f_{i_1,..,i_s}})$ of f given by $\mathbf{f} = \mathbf{vec(F)} := \sum_{\mathbf{i}\leq\mathbf{m}} \mathbf{f_i e_i}$, where $\mathbf{e_i} = \overset{s}{\underset{k=1}{\otimes}} \mathbf{e}_{i_k}$ with $\mathbf{e}_{i_k} \in \mathbb{R}^{m_k}$. Finally we state that for a tensor of real coefficients $\mathbf{F} = (\mathbf{f_i})_{\mathbf{1}\leq\mathbf{i}\leq\mathbf{m}} \in \mathbb{R}^{\mathbf{m}}$ we let $\mathbf{F}^{(\boldsymbol{\sigma_k})}$ denote the tensor $\mathbf{F}$ with its elements rearranged according to the cyclic permutation of the s-tuple $\{1, 2, \ldots, s\}$ given by $\boldsymbol{\sigma}_k = \{k, k+1, \ldots, s, 1, \ldots, k-1\}$, for $k = 1, \ldots, s$.

Finally, for a spline $f = \sum_{\mathbf{i}\leq\mathbf{m}} f_{\mathbf{i}} B_{\mathbf{i},\mathbf{d},\mathbf{t}}(\mathbf{x})$ we define a class of weighted l^p-norms of its B-spline coefficients, given by

$$\| f \|_{l^p,\mathbf{t}} = \begin{cases} (\sum_{\mathbf{i}\leq\mathbf{m}} w_{\mathbf{i}} |f_{\mathbf{i}}|^p)^{1/p}, & \text{for} \quad 1 \leq p < \infty, \\ \max\limits_{\mathbf{1}\leq\mathbf{i}\leq\mathbf{m}} |f_{\mathbf{i}}|, & \text{for} \quad p = \infty, \end{cases}$$

where the weights are given by $w_{\mathbf{i}} = \prod_{k=1}^{s} \frac{t^k_{i_k+d_k+1} - t^k_{i_k}}{d_k+1}$, for $\mathbf{1} \leq \mathbf{i} \leq \mathbf{m}$. Using the notation introduced above we have that $\| f \|_{l^p,\mathbf{t}} = \| \mathbf{W_t^{1/p} f} \|_{\mathbf{l^p}}$, $(p \geq 1)$ where $\mathbf{W_t}$ is a

diagonal scaling matrix given by

$$\mathbf{W_t} = \bigotimes_{\mathbf{k=1}}^{\mathbf{s}} \mathbf{W_{t^k}}, \quad \text{with} \quad \mathbf{W_{t^k}} = \mathbf{diag}\left(\left(\frac{\mathbf{t^k_{d_k+2}} - \mathbf{t^k_1}}{\mathbf{d_k+1}}\right), \ldots, \left(\frac{\mathbf{t^k_{m_k+d_k+1}} - \mathbf{t^k_{m_k}}}{\mathbf{d_k+1}}\right)\right).$$

These coefficient norms are easy to compute and are known to approximate the ordinary L^p-norms well for splines of moderate degree [2,6]. In the algorithms we use $p = 2$ when computing approximations and $p = \infty$ to measure the error.

3 The knot removal algorithm

Given an element $f \in \mathbb{S}_{\mathbf{d},\mathbf{t}}$, a tolerance $\varepsilon > 0$ and some norm $\|\cdot\|$ the goal of the knot removal algorithm presented in [6] is to find a subspace $\mathbb{S}_{\mathbf{d},\boldsymbol{\tau}}$ of $\mathbb{S}_{\mathbf{d},\mathbf{t}}$ ($\boldsymbol{\tau} \subseteq \mathbf{t}$) and an element $g \in \mathbb{S}_{\mathbf{d},\boldsymbol{\tau}}$ with $\|f - g\| < \varepsilon$, and where we want $\boldsymbol{\tau}$ to be of minimal length. In this section we review the basic parts of this algorithm as we extend the theory to tensor product splines. Further details of the material in this section can be found in [2].

3.1 Finding approximations

To approximate $f \in \mathbb{S}_{\mathbf{d},\mathbf{t}}$ in a subspace $\mathbb{S}_{\mathbf{d},\boldsymbol{\tau}}$, where $\boldsymbol{\tau}$ is of "length" $\mathbf{n} + \mathbf{d} + \mathbf{1}$ with $\mathbf{n} \leq \mathbf{m}$, we use the spline g which is the best approximation to f in the $l^2, \mathbf{t}$-norm. In other words, the spline we seek will be the solution to the minimization problem $\min_{h \in \mathbb{S}_{\mathbf{d},\boldsymbol{\tau}}} \|f - h\|^2_{l^2,\mathbf{t}}$. Solving this problem is equivalent to solving the linear least squares problem given by

$$\min_{\mathbf{C} \in \mathbb{R}^{\mathbf{n}}} \|\mathbf{W_t^{1/2}}(\mathbf{Ac} - \mathbf{f})\|^2_{\mathbf{l^2}}, \tag{3.1}$$

where $\mathbf{A} = \bigotimes_{\mathbf{k=1}}^{\mathbf{s}} \mathbf{A_k}$ is the knot insertion matrix from $\boldsymbol{\tau}$ to $\mathbf{t}$ (i.e. $\mathbf{A_k}$ is the knot insertion matrix from $\boldsymbol{\tau}^k$ to $\mathbf{t}^k$, for $k = 1, \ldots, s$), $\mathbf{f} = \mathbf{vec}(\mathbf{F})$ are the given B-spline coefficients of f in $\mathbb{S}_{\mathbf{d},\mathbf{t}}$ and $\mathbf{c} = \mathbf{vec}(\mathbf{C})$ are the unknown B-spline coefficients of g in $\mathbb{S}_{\mathbf{d},\boldsymbol{\tau}}$. Since the knot insertion matrix $\mathbf{A}$ has full rank and $\mathbf{W_t}$ is non-singular, the normal equations $\mathbf{A^T W_t A c} = \mathbf{A^T W_t f}$ associated with the system (3.1) will have a unique solution which can be found ([2,3]) by solving a series of s tensor equation systems given by

$$(\mathbf{A_k^T W_{t^k} A_k})\mathbf{D_k^{(\sigma_k)}} = (\mathbf{A_k^T W_{t^k}})\mathbf{D_{k-1}^{(\sigma_k)}}, \tag{3.2}$$

for $k = 1, \ldots, s$. Here $\mathbf{D_k} \in \mathbb{R}^{\mathbf{n_k}}$ with $\mathbf{n}_k = (n_1, \ldots, n_k, m_{k+1}, \ldots, m_s)$, and we let $\mathbf{D_0} = \mathbf{F}$, and set the coefficients of the approximation g equal to the solution of the last tensor equation system, $\mathbf{C} = \mathbf{D_s}$. The tensor equations (3.2) can be efficiently solved by calculating the Cholesky factorization of the banded coefficient matrix $(\mathbf{A_k^T W_{t^k} A_k})$ and solving for each right hand side in the tensor $(\mathbf{A_k^T W_{t^k}})\mathbf{D_{k-1}^{(\sigma_k)}}$.

3.2 Ranking the knots

The final approximation to the initial spline is found by searching through a sequence of approximations, constructed by using the approximation method of the previous section, on subsets of the knots of the initial spline. These subsets are calculated by associating a weight with each interior knot, representing a rough measure of its importance. See [6] for

the details. For higher dimensional tensor product splines we set the weight for a given knot to the maximum of the weights corresponding to this knot when the calculation is iterated over the "remaining" parameter directions. We refer to [2] for further details.

4 Knot removal methods

When removing knots from a tensor product spline we are faced with more options than in the case of a spline curve. In this section we present two different ways of performing knot removal. The first one studied in [2] based on a symmetric approach, treats all the parameter directions of a tensor product spline simultaneously, while the second one will treat one parameter direction at a time.

4.1 Knot removal based on a symmetric approach

If we let $G_f(\boldsymbol{\tau})$ denote the approximation to $f \in \mathbb{S}_{\mathbf{d},\mathbf{t}}$ defined on the knot vector $\boldsymbol{\tau}$ we see that the approximations in the sequence mentioned above can be written $\{G_f(\boldsymbol{\tau}_j)\}_{j=0}^N$, where $\boldsymbol{\tau}_j$ is constructed from $\mathbf{t}$ by removing j of its interior knots, and $N = \sum_{k=1}^s [m_k - (d_k + 1)]$ is the total number of interior knots of $\mathbf{t}$. Given such a sequence of approximations we can perform a search on the index j to determine an approximation $g^* = G_f(\boldsymbol{\tau}^*)$ to the initial spline f with a preferably short knot vector $\boldsymbol{\tau}^*$, and with the property that $\| f - g^* \|_{l^\infty,\mathbf{t}} \leq \varepsilon$, where ε is the specified tolerance. If the knot vector $\boldsymbol{\tau}^*$ is not equal to any of the two knot vectors $\boldsymbol{\tau}_0$ or $\boldsymbol{\tau}_N$ we may repeat the process to find a new approximation based on g^* as proposed in [6]. Taking into account how the sequence $\{G_f(\boldsymbol{\tau}_j)\}_{j=0}^N$ was constructed we expect the error $\| f - G_f(\boldsymbol{\tau}_j) \|_{l^\infty,\mathbf{t}}$ to decrease, but not necessarily strictly, for decreasing values of the search parameter j. How the search among the possible approximations is done will generally depend on a number of factors, including some which will be discussed later through examples. Also note that we only have to compute approximations for indexes actually used in the search. By treating all the directions simultaneously we take into consideration the inherent symmetry of the problem. As we will see later this will in some cases enable us to remove more knots than by treating one parameter direction at a time, but it will also lead to more complicated and slower code in an implementation.

4.2 Knot removal for one parameter direction at a time

In the second knot removal method we start by thinking of a spline $f \in \mathbb{S}_{\mathbf{d},\mathbf{t}}$ as a series of parametric curves in corresponding high dimensional spaces. We can then perform a parametric knot removal for each parameter direction. The advantage of this approach is that it is easy to implement since we may use existing knot removal routines for spline curves with only minor modifications.

In the following discussion we let $\varepsilon = \sum_{i=1}^s \varepsilon_i$, with $\varepsilon_i \geq 0$ for all i, be a given tolerance. Also let $f(\mathbf{x}) = \sum_{\mathbf{i} \leq \mathbf{m}} f_{\mathbf{i}} B_{\mathbf{i},\mathbf{d},\mathbf{t}}(\mathbf{x}) = \mathbf{B}_{\mathbf{t}}^{\mathbf{T}} \mathbf{f}$ be a spline in $\mathbb{S}_{\mathbf{d},\mathbf{t}} = \bigotimes_{k=1}^{s} \mathbb{S}_{d_k,\mathbf{t}^k}$, with $\mathbf{B}_{\mathbf{t}}^{\mathbf{T}} = \bigotimes_{\mathbf{k=1}}^{\mathbf{s}} \mathbf{B}_{\mathbf{t}^k}^{\mathbf{T}}$ and $\mathbf{f} = \mathbf{vec}(\mathbf{F})$. We start by identifying a series of parametric curves which may be naturally associated with this tensor product spline. We say that the spline f consists of the curves $\vec{f}_k(x_k)$, for $k = 1, \ldots, s$, where $\vec{f}_k(x_k)$ is the parametric

curve in $\mathbb{R}^{M_k}$, for $M_k = (\prod_{p=1}^{k-1} m_k)(\prod_{p=k+1}^{s} m_k)$, given by

$$\vec{f}_k(x_k) = \left[\left(\bigotimes_{l=1}^{k-1} \mathbf{I_{m_l}}\right) \otimes \mathbf{B_{t^k}^T} \otimes \left(\bigotimes_{\mathbf{l=k+1}}^{\mathbf{s}} \mathbf{I_{m_l}}\right)\right]\mathbf{f}.$$

We now return to the problem of finding a preferably short knot vector $\boldsymbol{\tau} \subseteq \mathbf{t}$ and a spline $g(\mathbf{x}) = \sum_{\mathbf{j}\leq\mathbf{n}} c_{\mathbf{j}} B_{\mathbf{j},\mathbf{d},\boldsymbol{\tau}}(\mathbf{x}) \in \mathbb{S}_{\mathbf{d},\boldsymbol{\tau}} = \bigotimes_{k=1}^{s} \mathbb{S}_{d_k,\boldsymbol{\tau}^k}$ with the property that $\| f - g \|_{l^\infty,\mathbf{t}} \leq \varepsilon$. To apply knot removal to $f \in \mathbb{S}_{\mathbf{d},\mathbf{t}}$ we can now go through the following steps for $k = 1, \ldots, s$.

1. Apply parametric knot removal with the tolerance ε_k to the parametric curve
$$\vec{f}_k(x_k) = \left[\left(\bigotimes_{l=1}^{k-1} \mathbf{I_{n_l}}\right) \otimes \mathbf{B_{t^k}^T} \otimes \left(\bigotimes_{\mathbf{l=k+1}}^{\mathbf{s}} \mathbf{I_{m_l}}\right)\right]\mathbf{f_{k-1}},$$
defined on $\mathbf{t}^k$, starting with $\mathbf{f}_0 = \mathbf{f}$.
2. This will produce a new parametric curve defined on the knot vector $\boldsymbol{\tau}^k \subseteq \mathbf{t}^k$
$$\vec{f}'_k(x_k) = \left[\left(\bigotimes_{l=1}^{k-1} \mathbf{I_{n_l}}\right) \otimes \mathbf{B_{\tau^k}^T} \otimes \left(\bigotimes_{\mathbf{l=k+1}}^{\mathbf{s}} \mathbf{I_{m_l}}\right)\right]\mathbf{f_k},$$
where $\mathbf{f}_k = \mathbf{vec}(\mathbf{F_k})$ for $\mathbf{F_k} \in \mathbb{R}^{\mathbf{n_1},\ldots,\mathbf{n_k},\mathbf{m_{k+1}},\ldots,\mathbf{m_s}}$.
3. We also have that
$$\begin{aligned}\vec{f}'_k(x_k) &= \left[\left(\bigotimes_{l=1}^{k-1} \mathbf{I_{n_l}}\right) \otimes \mathbf{B_{\tau^k}^T} \otimes \left(\bigotimes_{\mathbf{l=k+1}}^{\mathbf{s}} \mathbf{I_{m_l}}\right)\right]\mathbf{f_k} \\ &= \left[\left(\bigotimes_{l=1}^{k-1} \mathbf{I_{n_l}}\right) \otimes \mathbf{B_{t^k}^T} \otimes \left(\bigotimes_{\mathbf{l=k+1}}^{\mathbf{s}} \mathbf{I_{m_l}}\right)\right]\left[\left(\bigotimes_{\mathbf{l=1}}^{\mathbf{k-1}} \mathbf{I_{n_l}}\right) \otimes \mathbf{A_k} \otimes \left(\bigotimes_{\mathbf{l=k+1}}^{\mathbf{s}} \mathbf{I_{m_l}}\right)\right]\mathbf{f_k},\end{aligned}$$
where $\mathbf{A_k}$ is the knot insertion matrix from $\boldsymbol{\tau}^k$ to $\mathbf{t}^k$.
4. And consequently
$$\| \vec{f}_k - \vec{f}'_k \|_{l^\infty,\mathbf{t}^k} = \left\| \mathbf{f}_{k-1} - \left[\left(\bigotimes_{l=1}^{k-1} \mathbf{I_{n_l}}\right) \otimes \mathbf{A_k} \otimes \left(\bigotimes_{\mathbf{l=k+1}}^{\mathbf{s}} \mathbf{I_{m_l}}\right)\right]\mathbf{f_k} \right\|_{l^\infty} \leq \varepsilon_k.$$

Finally we let the coefficients of the function $g(\mathbf{x}) = \mathbf{B_\tau^T c} \in \mathbb{S}_{\mathbf{d},\boldsymbol{\tau}}$ be $\mathbf{c} = \mathbf{vec}(\mathbf{F_s})$, and we have the following result.

Theorem 4.1 *If we let $f(\mathbf{x}) = \mathbf{B_t^T f} \in \mathbb{S}_{\mathbf{d},\mathbf{t}}$ and $g(\mathbf{x}) = \mathbf{B_\tau^T c} \in \mathbb{S}_{\mathbf{d},\boldsymbol{\tau}}$ be the tensor product splines from the discussion above, then we have $\| f - g \|_{l^\infty,\mathbf{t}} \leq \varepsilon$.*

Proof: Let $\mathbf{A} = \bigotimes_{\mathbf{k=1}}^{\mathbf{s}} \mathbf{A_k}$ be the knot insertion matrix from $\boldsymbol{\tau}$ to $\mathbf{t}$, and let $f_0(\mathbf{x}) = \mathbf{B_t^T f_0}$

be equal to f and $f_s(\mathbf{x}) = \mathbf{B}_\tau^{\mathbf{T}}\mathbf{f_s}$ be equal to g, i.e. $\mathbf{f}_0 = \mathbf{f}$ and $\mathbf{f}_s = \mathbf{c}$. Then

$$
\begin{aligned}
&\|f - g\|_{l^\infty,\mathbf{t}} = \|\mathbf{f}_0 - \mathbf{A}\mathbf{f_s}\|_{l^\infty} \\
&= \left\| \mathbf{f}_0 + \sum_{k=2}^{s} \left(\left[\left(\overset{k-1}{\underset{l=1}{\otimes}} \mathbf{A_l} \right) \otimes \left(\overset{s}{\underset{l=k}{\otimes}} \mathbf{I_{m_l}} \right) \right] \mathbf{f_{k-1}} - \left[\left(\overset{k-1}{\underset{l=1}{\otimes}} \mathbf{A_l} \right) \otimes \left(\overset{s}{\underset{l=k}{\otimes}} \mathbf{I_{m_l}} \right) \right] \mathbf{f_{k-1}} \right) - \overset{s}{\underset{k=1}{\otimes}} \mathbf{A_k} \mathbf{f_s} \right\|_{l^\infty} \\
&\leq \sum_{k=1}^{s} \left\| \left[\left(\overset{k-1}{\underset{l=1}{\otimes}} \mathbf{A_l} \right) \otimes \left(\overset{s}{\underset{l=k}{\otimes}} \mathbf{I_{m_l}} \right) \right] \left[\mathbf{f_{k-1}} - \left[\left(\overset{k-1}{\underset{l=1}{\otimes}} \mathbf{I_{n_l}} \right) \otimes \mathbf{A_k} \otimes \left(\overset{s}{\underset{l=k+1}{\otimes}} \mathbf{I_{m_l}} \right) \right] \mathbf{f_k} \right] \right\|_{l^\infty} \\
&\leq \sum_{k=1}^{s} \left\| \mathbf{f}_{k-1} - \left[\left(\overset{k-1}{\underset{l=1}{\otimes}} \mathbf{I_{n_l}} \right) \otimes \mathbf{A_k} \otimes \left(\overset{s}{\underset{l=k+1}{\otimes}} \mathbf{I_{m_l}} \right) \right] \mathbf{f_k} \right\|_{l^\infty} = \sum_{k=1}^{s} \left\| \vec{f}_k - \vec{f}_k' \right\|_{l^\infty,\mathbf{t}^k} \leq \varepsilon. \qquad \square
\end{aligned}
$$

5 Examples

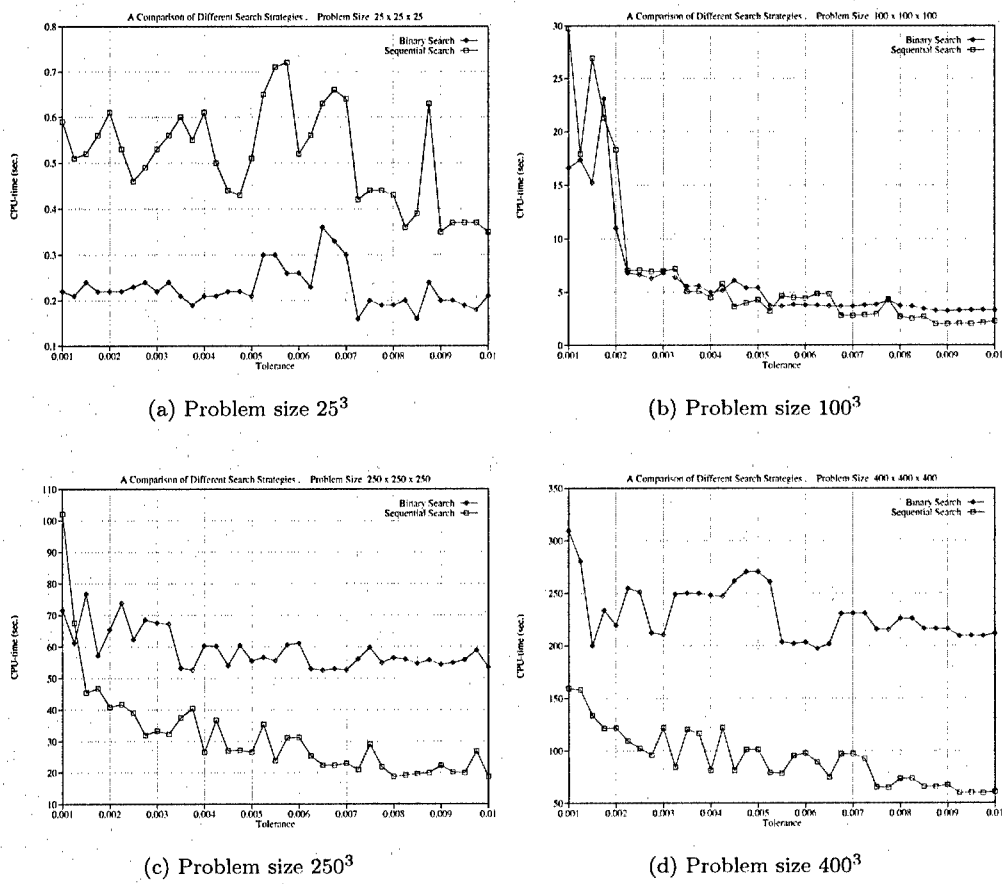
The knot removal methods presented above have been implemented and tested on a computer. In this section we present trivariate examples from this implementation and propose different knot removal strategies depending on the problem at hand. See [3] for a detailed description of this implementation.

Example 5.1 In this first example we will compare two different strategies for searching through a list of approximations $\{G_f(\tau_j)\}_{j=0}^N$ introduced above. We will consider the knot removal method treating one parameter direction at a time, which means that we end up solving a parametric knot removal problem with tolerance $\varepsilon_i = \varepsilon/3$, $i = 1, 2, 3$, for each of the three parameter directions.

To improve efficiency the parametric knot removal routine implemented is constructed in a way that lets it abort the computation if an approximation for any component of the parametric curve fails to lie within the specified tolerance. This fact suggests a search strategy where we compute successive approximations to the initial spline by adding one interior knot at a time, starting with zero interior knots, and where each intermediate approximation is given by the first of these approximation processes to be completed. Intuitively we would expect such a *sequential search strategy* to perform best for "large" tolerances and/or large problems, where it is more to gain by aborting an approximation process. In this example we have compared this search strategy with a strategy proposed in [6] using a binary search.

In all the tests we have used an initial trilinear spline constructed by sampling the function given by $f(x, y, z) = \frac{1}{3}[\sin(2\pi x) + \sin(2\pi y) + \sin(2\pi z)]$ in the points specified by a uniform 3-dimensional grid on the domain $\Omega = [0, 1]^3$, for four selected grid sizes. Each spline was reduced by using both of the search strategies mentioned above, for tolerances varying from $\varepsilon = 0.001$ to $\varepsilon = 0.01$. Both of the search strategies produced approximately the same end grid size in each test.

In Figure 1 the CPU-time of the two search strategies is plotted against the tolerance for the selected grid sizes. We observe that the reductions utilizing a binary search perform best on small problems, while the sequential search strategy turn out to be superior for large problems.



(a) Problem size 25^3

(b) Problem size 100^3

(c) Problem size 250^3

(d) Problem size 400^3

FIG. 1. A comparison of two different search strategies.

Example 5.2 In this example we compare the two different knot removal methods presented in this paper. Here we have used an initial trilinear spline constructed by sampling a function given by $f(x, y, z) = e^{\sin(2\pi x^2 yz)}$ in the points specified by a uniform 3-dimensional grid on the domain $\Omega = [0, 1]^3$, for varying grid sizes. Each spline was reduced by both the method based on the symmetric approach and the method treating one parameter direction at a time.

The results are presented in Table 1. We see that in our implementation the method using the symmetric approach is by far the slowest method. However, at least for the type of function considered in this example the method based on the symmetric approach will give a much better reduction than the other.

Knot Removal for Trilinear Splines, Tolerance $\varepsilon = 0.005$						
Start grid	Parametric, binary search			Symmetric, binary search		
	CPU	End grid	Error	CPU	End grid	Error
100^3	16.53	$72 \times 65 \times 65$	$4.93800 \cdot 10^{-3}$	63.23	$54 \times 53 \times 53$	$4.92080 \cdot 10^{-3}$
150^3	56.44	$81 \times 71 \times 71$	$4.80243 \cdot 10^{-3}$	122.2	$51 \times 49 \times 49$	$4.77236 \cdot 10^{-3}$
200^3	99.48	$68 \times 66 \times 66$	$4.91142 \cdot 10^{-3}$	300.9	$54 \times 50 \times 51$	$4.98275 \cdot 10^{-3}$
250^3	165.3	$74 \times 62 \times 62$	$4.74970 \cdot 10^{-3}$	584.8	$61 \times 56 \times 56$	$4.85916 \cdot 10^{-3}$
300^3	256.8	$72 \times 62 \times 62$	$4.85316 \cdot 10^{-3}$	1094	$60 \times 54 \times 53$	$4.81551 \cdot 10^{-3}$
350^3	391.4	$75 \times 65 \times 63$	$4.77028 \cdot 10^{-3}$	1312	$54 \times 50 \times 50$	$4.92422 \cdot 10^{-3}$
400^3	494.6	$71 \times 59 \times 63$	$4.79631 \cdot 10^{-3}$	1865	$54 \times 50 \times 50$	$4.81064 \cdot 10^{-3}$

TAB. 1 Knot removal for the trilinear splines of Example 2.

Bibliography

1. Arge, E., Dæhlen, M., Lyche, T. and Mørken, K. (1990). *Constrained spline approximation of functions and data based on constrained knot removal.* In: *Algorithms for Approximation II*, J. C. Mason and M. G. Cox (eds.), Chapman and Hall, London, 4–20
2. Brenna, T. (1998). *Knot removal for multivariate tensor product splines.* Master thesis, part I. Dept. of Informatics, Univ. of Oslo.
3. Brenna, T. (1998). *Knot removal for linear, bilinear and trilinear splines.* Master thesis, part II. Dept. of Informatics, Univ. of Oslo.
4. Graham, A. (1981). *Kronecker Products and Matrix Calculus With Applications.* Ellis Horwood Series. Mathematics and its applications.
5. Lyche, T. and Mørken, K. (1986). *Knot removal for parametric B-spline curves and surfaces.* Computer Aided Geometric Design, 4, 217–230
6. Lyche, T. and Mørken, K. (1987). *A data reduction strategy for splines with applications to the approximation of functions and data.* IMA Journal of Numerical Analysis, 8, 185–208.

Fixed- and free-knot univariate least-squares data approximation by polynomial splines

Maurice Cox, Peter Harris and Paul Kenward

National Physical Laboratory, Teddington, Middlesex, TW11 0LW, UK
maurice.cox@npl.co.uk, peter.harris@npl.co.uk, paul.kenward@npl.co.uk

Abstract

Fixed- and free-knot least-squares data approximation by polynomial splines is considered. Classes of knot-placement algorithms are discussed. A practical example of knot placement is presented, and future possibilities in free-knot spline approximation are addressed.

1 Introduction

The representation of univariate polynomial splines in terms of B-splines is reviewed (Section 2), leading to the problem of obtaining fixed- and free-knot ℓ_2 spline approximations (Section 3). The accepted approach to the fixed-knot case is recalled (Section 4) and the manner in which spline uncertainties can be evaluated given (Section 5). The importance of families of spline approximants is emphasised (Section 6). The free-knot problem is formulated (Section 7) and several of the established and some lesser-known knot-placement strategies reviewed (Section 8). Conclusions are drawn and future possibilities indicated (Section 9).

2 Univariate polynomial splines

Let $I := [x_{\min}, x_{\max}]$ be an interval of the x-axis, and $x_{\min} = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{N-1} \leq \lambda_N < \lambda_{N+1} = x_{\max}$ a partition of I. A spline $s(x)$ of order n (degree $n-1$) on I is a piecewise polynomial of order n on $(\lambda_j, \lambda_{j+1})$, $j = 0, \ldots, N$. The spline s is C^{n-k-1} at λ_j if $\text{card}(\lambda_\ell = \lambda_j, \ell \in \{1, \ldots, n\}) = k$. The partition points $\boldsymbol{\lambda} = \{\lambda_j\}_1^N$ are the (interior) *knots* of s. To specify the complete set of knots needed to define s on I in terms of B-splines, the knots $\{\lambda_j\}_1^N$ are augmented by knots $\{\lambda_j\}_{1-n}^{-1}$ and $\{\lambda_j\}_{N+2}^q$, $q = N + n$, satisfying

$$\lambda_{1-n} \leq \cdots \leq \lambda_0, \qquad \lambda_{N+1} \leq \cdots \leq \lambda_q.$$

For many purposes, a good choice [10] of additional knots is

$$\lambda_{1-n} = \cdots = \lambda_0, \qquad \lambda_{N+1} = \cdots = \lambda_q.$$

It readily permits derivative boundary conditions to be incorporated in spline approximants [7]. On I, $s(x)$ has the *B-spline representation* [5]

$$s(x) := s(\mathbf{c}, \boldsymbol{\lambda}; x) = \sum_{j=1}^{q} c_j N_{n,j}(\boldsymbol{\lambda}; x), \tag{2.1}$$

where $N_{n,j}(\boldsymbol{\lambda}; x)$ is the *B-spline* [5, 12] of order n with knots $\{\lambda_k\}_{j-n}^{j}$ and $\mathbf{c} = (c_1, \ldots, c_q)^{\mathrm{T}}$ are the *B-spline coefficients* of s. Each $N_{n,j}(\boldsymbol{\lambda}; x)$ is a spline with knots $\boldsymbol{\lambda}$, is non-negative and has compact support. Specifically,

$$N_{n,j}(\boldsymbol{\lambda}; x) > 0, \quad x \in (\lambda_{j-n}, \lambda_j), \qquad \mathrm{supp}(N_{n,j}(\boldsymbol{\lambda}; x)) = [\lambda_{j-n}, \lambda_j]. \tag{2.2}$$

The B-spline basis $\{N_{n,j}(\boldsymbol{\lambda}; x)\}_{j=1}^{q}$ for splines of order n with knots $\boldsymbol{\lambda}$ is generally very well-conditioned [10]. Moreover, the basis functions for any $x \in [x_{\min}, x_{\max}]$ can be formed in an unconditionally stable manner using a three-term recurrence relation [5, 12]. Specifically, the *relative errors* in the values $fl(N_{n,j}(\boldsymbol{\lambda}; x))$ of the basis function computed using IEEE floating-point arithmetic [18] satisfy

$$|fl(N_{n,j}(\boldsymbol{\lambda}; x)) - N_{n,j}(\boldsymbol{\lambda}; x)| \leq CnN_{n,j}(\boldsymbol{\lambda}; x)\eta,$$

where C is a constant that is a small multiple of unity and η is the unit roundoff of the floating point processor [5]. The B-spline basis for splines of order 3 with interior knots at $x = (1, 2, 5)^{\mathrm{T}}$ and coincident end knots at $x = 0$ and 10, is shown in Figure 1.

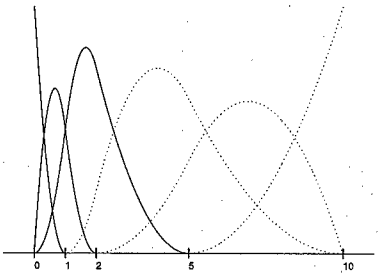


FIG. 1. The B-spline basis for splines of order 3 for some nonuniformly spaced knots. The first three B-spline basis functions are shown as solid lines and the remaining three as dotted lines.

Valuable properties of s can be deduced [12] from those of the B-splines. A useful property is that, for any $x \in I$, $s(x)$ is a convex combination of the coefficients of the B-splines whose support contains x. Thus, local bounds for s can readily be found:

$$\min_{j<k\leq j+n} c_k \leq s(x) \leq \max_{j<k\leq j+n} c_k, \qquad x \in [\lambda_j, \lambda_{j+1}].$$

These bounds imply a mimicking property for s, viz., that the elements of $\mathbf{c}$ tend to vary in much the same way that s varies. Figure 2 depicts a spline curve s of order 4 with "non-polynomial" shape having interior knots at $x = (1, 2, 5)^{\mathrm{T}}$, coincident end knots at $x = 0$ and 10, and B-spline coefficients $(0.00, 0.20, 0.60, 0.22, 0.18, 0.14, 0.12)^{\mathrm{T}}$.

To reproduce this shape to visual accuracy with a polynomial would require a high degree and hence many more defining coefficients. The mimicking property is evident: successive elements of $\mathbf{c}$ rise, fall sharply and then gently, behaving in a similar way to s.

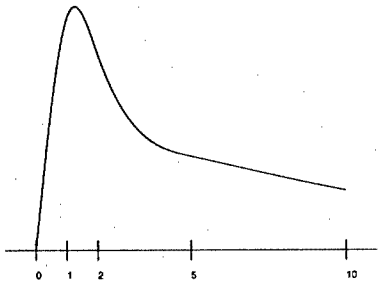


FIG. 2. A spline curve with "non-polynomial" shape illustrating the mimicking property.

3 Fixed- and free-knot approximation

Two types of data approximation (or data modelling) in the ℓ_2 norm by splines are regularly considered. One is the determination of the B-spline coefficients $\mathbf{c}$ for given data, a prescribed order n and prescribed knots $\boldsymbol{\lambda}$. The other is the determination of $\mathbf{c}$ *and* $\boldsymbol{\lambda}$ for given data and spline order n. The former problem is linear with respect to the parameters of the spline, just $\mathbf{c}$ being regarded as unknown. The latter is nonlinear, both $\mathbf{c}$ and $\boldsymbol{\lambda}$ being unknown.

The linear case is well understood, with highly satisfactory algorithms [10] and software implementations [1, 16] available. The nonlinear case remains a research problem, although useful algorithms (Section 8) have been proposed, implemented and used. Many of these algorithms "iterate" with respect to $\boldsymbol{\lambda}$, where for each choice of knots the resulting linear problem is solved for $\mathbf{c}$. Thus, the linear problem (Section 4) is important in its own right and as part of the solution strategy for knot-placement algorithms.

4 Least-squares data approximation by splines with fixed knots

The ℓ_2 data approximation problem for splines with fixed knots can be posed as follows. Given are data points $\{(x_i, y_i)\}_1^m$, with $x_1 \leq \cdots \leq x_m$, and corresponding *weights* $\{w_i\}_1^m$ or *standard uncertainties* $\{u_i\}_1^m$. The w_i reflect the *relative* quality of the y_i,[1] u_i is the standard uncertainty of y_i and corresponds to the standard deviation of possible "measurements" at $x = x_i$ of the function underlying the data, y_i being one realisation. Given also are the N knots $\boldsymbol{\lambda} = \{\lambda_j\}_1^N$ and the order n of the spline s.

When *weights* are specified, the problem is to determine the spline $s(x)$ of order n, with knots $\boldsymbol{\lambda}$, such that the two-norm of $\{w_i e_i\}_1^m$ is minimised with respect to $\mathbf{c}$.

[1]The x_i are taken as exact for the treatment here. A generalised treatment is possible, in which the x_i are also regarded as inexact. The problem becomes nonlinear (in $\mathbf{c}$).

When *standard uncertainties* are specified, the two-norm of $\{u_i^{-1}e_i\}_1^m$ is minimised with respect to $\mathbf{c}$. If $w_i = u_i^{-1}$, $i = 1, \ldots, m$, the two formulations are identical in terms of the spline produced. When weights are specified, s is referred to as a *spline approximant.* When uncertainties are prescribed, s is known as a *spline model.* There are differences (Section 5) in interpretation in terms of the statistical uncertainties associated with the solution and in terms of validating the spline model so obtained.

The use of a formulation in terms of standard uncertainties, together with the B-spline representation (2.1) of s, gives the linear algebraic formulation[2]

$$\min_{\mathbf{c}} \mathbf{e}^{\mathrm{T}} V_{\mathbf{y}}^{-1} \mathbf{e}, \qquad \mathbf{e} = \mathbf{y} - A\mathbf{c}, \tag{4.1}$$

where $\mathbf{y} = (y_1, \ldots, y_m)^{\mathrm{T}}$, A is an $m \times q$ matrix with $a_{i,j} = N_{n,j}(x_i)$, and $V_{\mathbf{y}} = \mathrm{diag}(u_1^2, \ldots, u_m^2)$. Matrix computational methods can be applied to this formulation. As a consequence of property (2.2) of the B-splines, A is a rectangular banded matrix of bandwidth n [8].

The linear algebraic solution can be effected using Givens rotations to triangularise the system, back-solution then yielding the coefficients $\mathbf{c}$ [6]. The number of floating-point operations (flops) required is to first order $O(mn^2)$, i.e., independent of the number of knots. Hence computing a spline model for many knots is hardly more expensive than one for a few knots. Moreover, since for many problems cubic splines ($n = 4$) yield a good balance between approximation properties and smoothness (continuity class C^2), regarding the order as *fixed* gives a flop count $O(m)$.

The vector $\mathbf{c}$ is unique [11] if there is a strictly ordered subset $\mathbf{t} = \{t_j\}_1^q$ of $\mathbf{x}$ such that the Schoenberg–Whitney conditions [21]

$$t_j \in \mathrm{supp}(N_{n,j}(\boldsymbol{\lambda}; x)), \qquad j = 1, \ldots, q, \tag{4.2}$$

hold. In a case where the conditions (4.2) do not hold[3], an appropriate member can be selected from the space of possible solutions. Such a selection is also advisable if the conditions are in a practical sense "close" to being violated. A particular solution can be determined by augmenting the least-squares formulation by a minimal number of *equality* constraints for $\mathbf{c}$ such that A has full column rank [10].

An instance of the type of data set to which the algorithms of this paper are addressed is shown in Figure 3. Such a data set (cf. Section 2) has the variety of behaviour that cannot readily be reproduced by some other classes of approximating functions.

5 Spline uncertainties

Once a valid spline model has been obtained, the uncertainties associated with the spline can be evaluated [9]. Uncertainty evaluations are essential in metrology, where all measurement results are to be accompanied by a quantification of their reliability [2], and important in other fields. The key entity is the covariance matrix $V_{\mathbf{c}}$ of the spline

[2] A further generalisation is possible in which mutual dependencies are permitted among the measurement errors. In this case, $V_{\mathbf{y}}$ is non-diagonal.

[3] A set of knots giving rise to this circumstance may be a consequence of an automatic knot-placement procedure.

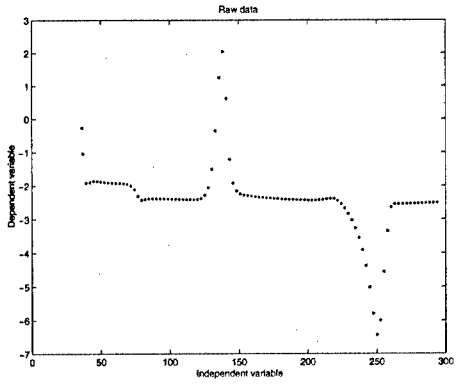


FIG. 3. A data set representing heat flow as a function of temperature. Such data forms the basis of the determination of thermophysical properties of materials under test. For clarity only every fifth data point is shown.

coefficients $\mathbf{c}$. Using recognised procedures of linear algebra,

$$V_{\mathbf{c}} = (A^{\mathrm{T}} V_{\mathbf{y}}^{-1} A)^{-1}. \tag{5.1}$$

From this result, the standard uncertainty of any quantity that depends on $\mathbf{c}$ can be evaluated. Specifically, for a given constant vector $\mathbf{p}$, the standard uncertainty $u(\mathbf{p}^{\mathrm{T}}\mathbf{c})$ of $\mathbf{p}^{\mathrm{T}}\mathbf{c}$ is given by

$$u^2(\mathbf{p}^{\mathrm{T}}\mathbf{c}) = \mathbf{p}^{\mathrm{T}} V_{\mathbf{c}} \mathbf{p}.$$

By setting $\mathbf{p}$ to contain the values of the B-spline basis at a point $x \in I$, the standard uncertainty of $s(x)$ can be formed. The standard uncertainty of a nonlinear function of $\mathbf{c}$ can be estimated by first linearising the expression about the solution value of $\mathbf{c}$.

If weights rather than uncertainties are specified for the data, (5.1) takes the form

$$V_{\mathbf{c}} = \hat{\sigma}^2 (A^{\mathrm{T}} W^2 A)^{-1},$$

where $\hat{\sigma}$ estimates the standard deviation of the weighted residuals $\{w_i e_i\}_1^m$, $W = \mathrm{diag}(w_1, \ldots, w_m)$, and

$$\hat{\sigma}^2 = \mathbf{e}^{\mathrm{T}} W^2 \mathbf{e} \,/ (m - q)$$

evaluated at the solution.

6 Families of approximants

When dealing with certain classes of approximating function it is natural and useful to consider *families of approximants*. A simple example is polynomial approximation, for polynomials $p_j(x)$ of order $j = 1, 2, \ldots, N$, for some maximum order N. Each member of the family "contains" the previous member. It is then meaningful to consider the approximation measure, e.g., the ℓ_2-norm here, with respect to indices denoting members. Thus, the value of the ℓ_2-norm for the polynomial approximant of order j can be inspected with respect to index j for $j = 1, 2, \ldots, N$. For data approximation, it is more

meaningful to use as the measure the *root-mean-square residual* given by dividing the ℓ_2-norm by $(m-j)^{1/2}$. For representative data, the expectation is that as j increases this quantity should stabilise to an essentially constant value. This property provides a useful validation procedure. If weights u_i^{-1} are used as in Section 4 this measure should settle to the value unity. Thus the approximant with index j (normally the smallest such) that achieves the value one is sought.

Within most of the strategies outlined in Section 8 it is possible to produce results for $N = 1, 2, \ldots$ knots, and thus to study the effect of the number of knots on the quality of the approximant. From such information it may be possible to select an acceptable solution. If for each number of knots, the knots contain those for the previous number, and an ℓ_2 approximant is determined, the sequence of approximants for $N = 1, 2, \ldots$ knots forms a *family*. A family has the property that the sequence of values of the ℓ_2-norm is monotonically decreasing.

7 Least-squares data approximation by splines with free knots

The problem of least-squares data approximation by splines with free knots can be formulated in the same way as that for fixed knots (Section 4), except that the knots are not specified *a priori*, either in location or number. The formulation (4.1) no longer yields a linear problem, since the matrix A of B-spline values is now a function of $\boldsymbol{\lambda}$. Instead, $\mathbf{e}(\boldsymbol{\lambda}) = \mathbf{y} - A(\boldsymbol{\lambda})\mathbf{c}$, and it is required to solve

$$\min_{\boldsymbol{\lambda};\mathbf{c}} \mathbf{e}^{\mathrm{T}}(\boldsymbol{\lambda}) V_{\mathbf{y}}^{-1} \mathbf{e}(\boldsymbol{\lambda}). \tag{7.1}$$

In order to reflect the fact that for any given knot set the B-spline coefficients are given by solving a relatively simple, linear problem, formulation (7.1) can be expressed as

$$\min_{\boldsymbol{\lambda}} \left(\min_{\mathbf{c}} \mathbf{e}^{\mathrm{T}}(\boldsymbol{\lambda}) V_{\mathbf{y}}^{-1} \mathbf{e}(\boldsymbol{\lambda}) \right). \tag{7.2}$$

Extensive use is made of this elementary result.

8 Knot-placement strategies

Many knot-placement strategies have been proposed and used. Some of these strategies are outlined and their properties indicated. Several of the strategies generate a *family* of candidate spline approximants, with advantages for model validation.

8.1 Manual methods

Manual methods can be classed as those methods for which the user examines the general "shape" of the function underpinning the data, selecting the number and location of the knots on this basis. With practice and visual aids, acceptable solutions can often be obtained [6]. Naturally, knots are chosen to be more concentrated where "things are happening" in contrast to regions where the underpinning behaviour is innocuous.

8.2 Strategies that depend only on abscissa values

Strategies based on the manner in which the values of the independent variable are distributed may be used to place the knots (at points that are not necessarily the data

abscissae themselves). A facility in DASL (the NPL Data Approximation Subroutine Library) [1] provides one such strategy, based on the Schoenberg–Whitney conditions (4.2) in the following way. Intuitively, these conditions imply that there is no region where there are "too many" knots compared with the number of data points. *Mathematically*, these conditions guarantee uniqueness. *Numerically*, their satisfaction does not ensure that the solution is well-defined. If the conditions are "close" to being violated, $\mathbf{c}$ will be sensitive to perturbations in the data. In particular, since the behaviour of $\mathbf{c}$ "controls" that of s (Section 2), the spline is likely to exhibit spurious behaviour such as large undesirable oscillations if $\|\mathbf{c}\|_2 \gg \|\mathbf{y}\|_2$.

It follows that a sensible choice of knots would be such that the Schoenberg–Whitney conditions are satisfied "as well as possible" for a data subset. Such a choice is made in DASL [1] for spline approximation of arbitrary order. It is also made in a cubic spline *interpolation* routine in the NAG Library [16], regarding spline interpolation as a special case of spline approximation in which $q = m$ and $N = m - n$. The choice made is seen most simply by first applying it to spline interpolation. Consider the choice

$$\lambda_j = \frac{1}{2}(x_{j+\lfloor n/2 \rfloor} + x_{j+\lfloor (n+1)/2 \rfloor}), \qquad j = 1, \ldots, m-n,$$

where $\lfloor v \rfloor$ is the largest integer no larger than v. For n even, $\lambda_j = x_{j+n/2}$. Thus, the choice $t_j = \lambda_{j-n/2}$ would be made. However (Section 2), $\mathrm{supp}(N_{n,j}) = [\lambda_{j-n}, \lambda_j]$. Thus, *index-wise*, the Schoenberg–Whitney conditions are satisfied as well as possible in the sense that the index of $\lambda_{j-n/2}$ falls halfway between the indices of the support endpoints λ_{j-n} and λ_j. Comparable considerations apply for n odd. Precisely this choice is recommended [14, 16] in the context of cubic spline interpolation. It is the "not a knot" criterion, as a practical alternative to the classical use of boundary derivatives. A knot is placed at each "interior" data value x_i apart from x_2 and x_{m-1}.

The above choice can be interpreted as follows. Consider the graph $x = F(\ell)$ given by the join of the points $\{(i, x_i)\}_1^m$. The jth interior knot, λ_j, for $j = 1, \ldots, m-n$, is given by $F(j + n/2)$. The successive spacings between the index arguments of F for $j = 0, \ldots, N+1$, using $F(0) = x_{\min}$ and $F(N+1) = x_{\max}$, are therefore

$$1 + n/2, \underbrace{1, \ldots, 1}_{N-1}, 1 + n/2.$$

For *approximation*, these successive spacings are proportionally increased to account for the fact that there are fewer knots. The resulting expression for the jth interior knot is

$$\lambda_j = F(1 + (m-1)(j + n/2 - 1)/(q-1)), \qquad j = 1, \ldots, N.$$

The choice can be interpreted as placing the interior knots such that there is an approximately equal number of data points in each knot interval (interval between adjacent knots), except that in the first and the last interval there are approximately $n/2$ times as many points. The strategy [1] has the property that when N is such that the data is interpolated, the choice of knots agrees with one of the recommended choices for spline interpolation.[4]

[4]The approach tends to give better knot locations if the data is gathered in a manner which ensures that the local

Figure 4 illustrates the above strategy for a spline interpolant and approximant of order 4 to data with abscissae $\mathbf{x} = (0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 3, 4, 5, 7.5, 10)^{\mathrm{T}}$. Each figure shows the graph $x = F(\ell)$. For the interpolant (left-hand graph), ten knots are chosen to coincide with the abscissa values $x_3, \ldots, x_{12}$. For the approximant (right-hand graph), four knots are chosen such that there are two points in each interval, excepting the first and last interval where there are four points, i.e., $n/2 = 2$ times as many. The distribution of the knots reflects that of the abscissa values.

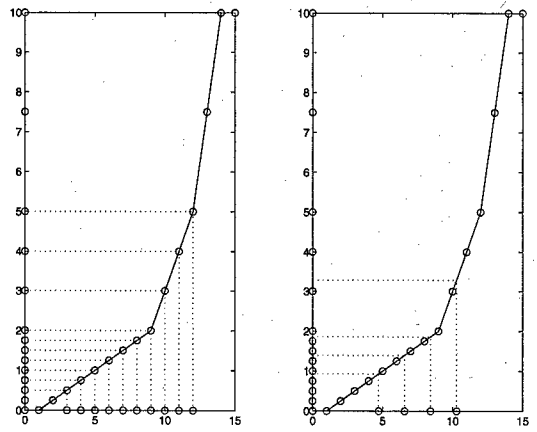


FIG. 4. A knot placement strategy depending only on the abscissa values.

A simpler strategy is to select uniformly spaced knots. The Schoenberg–Whitney conditions will not necessarily automatically be satisfied by such a choice, and the spline approximant would therefore not be unique, although the approach indicated at the end of Section 4 could be applied.

8.3 Sequential knot-insertion strategies

In a sequential knot-insertion strategy, a succession of approximants is obtained, in which for each approximant a knot is inserted in the knot interval that gives rise to the greatest contribution to the ℓ_2 error. A knot interval is an interval between adjacent knots, where the endpoints of I count as knots for this purpose. Previously inserted knots are retained undisturbed. Several variants are possible (also see Section 8.10), e.g.:

- Start the process with a number of knots already in place, perhaps obtained from information specific to the application.
- Candidate positions for a new knot are
  - * The continuum of points within the interval. The approach gives rise to the minimisation of a univariate function that may possess local minima.
  - * The subset within the interval of a discrete set of points chosen *a priori*, e.g., the data abscissa themselves or a uniformly spaced set of x-values. The approach

density of the data is greater in regions where the behaviour of y is more marked.

gives rise to a finite computation for the globally-best choice of knot, relative to the discretisation, with respect to previous knots.

- More than one knot can be inserted at a time. Doing so gives an approach that is intermediate between full optimisation (Section 8.6) and sequential (single) knot insertion. Computation times rise rapidly with the number of "simultaneous" knots so inserted, so in practice only a small number, say two or three, might be feasible.

The "upper set" of crosses in Figure 5 shows the root-mean-square residual as a function of the number of knots for the application of this strategy to the thermophysical data of Figure 3.

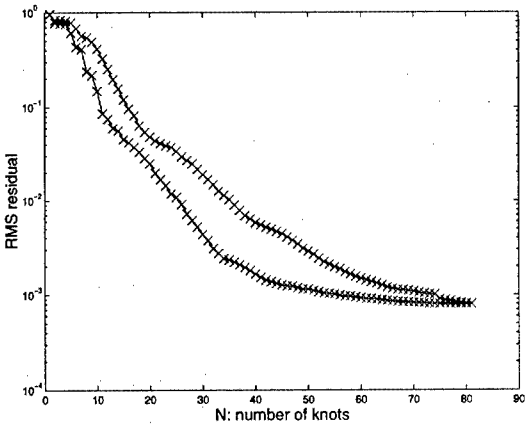


FIG. 5. The root-mean-square residual as a function of the number of knots for the application of knot-insertion and knot-removal strategies to the thermophysical data of Figure 3. The "upper set" of crosses indicate the values obtained for knot insertion and the lower for knot removal. The knot-removal strategy starts with the knot set provided by the knot-insertion strategy, which was terminated after 81 knots had been placed. The figure depicts the root-mean-square residual on a logarithmic scale, so its value varies by a factor of 1000 from 1 to 81 knots.

8.4 Sequential knot-removal strategies

In a sequential knot-removal strategy, the starting point is an initial spline approximant having a "large" number of knots that typically would be regarded as an acceptable approximant to the data and that contains (perhaps many) more knots than desired. Also see Section 8.10. Each successive approximant is obtained from the previous approximant by deleting one (or more) knots. The knot selected for removal is chosen as that having least effect in terms of the change in the ℓ_2 error. The process is continued until an acceptable approximant is no longer obtained.

The initially large number of knots (Section 8.10) provides an appreciable number of candidate knots for removal and thus greater flexibility. The rationale is that in

contrast to successive knot insertion a succession of acceptable approximants is obtained as opposed to a succession of unacceptable approximants, until the final "solution" is provided. There are variants, as with sequential knot insertion. For example, several knots can be removed at each stage.

A different class of knot removal algorithms [20] is based on a general class of ℓ_p norms. It is not concerned specifically with data approximation, but with replacing an initial spline approximant (that may correspond to an approximant) by one that is acceptably close according to the measure.

The "lower set" of crosses in Figure 5 shows the root-mean-square residual as a function of the number of knots for the application of this strategy to the thermophysical data part-depicted in Figure 3.

8.5 Theory-based approaches

The distance of a spline $s(x)$ with knots $\boldsymbol{\lambda}$ from a sufficiently differentiable function $f(x)$ is proportional to $h^n|f^{(n)}(\xi)|$, where h is the local knot spacing and ξ is a value of x [14]. Consider inverting this expression in order approximately to equalise the error with respect to x. The lengths of the knot intervals should consequently be chosen to be proportional to $|f^{(n)}(\xi)|^{-1/n}$, where ξ is a value in the neighbourhood of the respective knot interval. Consider the function

$$F(x) = \int_{x_{\min}}^{x} |f^{(n)}(t)|^{1/n} dt \Big/ \int_{x_{\min}}^{x_{\max}} |f^{(n)}(t)|^{1/n} dt \,. \tag{8.1}$$

Take knots given by

$$F(\lambda_j) = \frac{j}{N+1}, \qquad j = 1, \ldots, N. \tag{8.2}$$

This result corresponds to dividing the range of the monotonically increasing function $F(x)$, for $x \in I$, into $N+1$ contiguous subranges of equal length, taking the values of x corresponding to the subrange endpoints as the knots.

In practice f, let alone F, is unknown. Various efforts have been made to estimate f and hence F from the data points. For instance, if the data is approximated by a spline of order $n+1$, its nth derivative, a piecewise-constant function, can be used to estimate F [3]. It is then straightforward to form the required knots. The approach begs the question in the case of data. In order to estimate knots for a spline of order n, it is first necessary to construct a spline approximant of order $n+1$ for the data, the construction of which itself requires a choice of knots.

Alternatively [13], a spline approximant of order n for the data can be constructed for some *convenient* choice of knots. Its nth derivative is of course zero (except at the knots). However, its $(n-1)$th derivative is piecewise constant, a function that can be approximated by the join of the mean values at the knots of the constant pieces to the immediate right and left, with special consideration at the endpoints of I. The derivative of this piecewise-linear function then provides a piecewise-constant representation of the nth derivative, that can be used as before. Knots can then be deduced from this form as above. The advantage of this approach is that it can be iterated [13]. If the process "converges", the result can be used to provide the required knot set. The process can

work well, but is capable of producing disappointing results. Several variants of the basic concept are possible. The approach warrants careful re-visiting.

8.6 "Overall" optimisation approaches

For any given value of N, the problem is regarded as an optimisation problem with respect to the overall error measure. It is necessary to provide a sensible initial estimate of the knot positions. Local solutions which may be grossly inferior to the global solution are possible [4]. At an optimal solution, knots may coalesce, thus reducing the continuity of the spline at such points [19]; the same comment applies to the sequential-knot-insertion and optimisation approach (Section 8.7).

8.7 Sequential knot insertion and optimisation

Sequential knot insertion with optimisation is identical to the sequential knot-insertion strategy (Section 8.3) except that, after each knot is inserted, all previously-inserted knots are adjusted such that the complete set of knots at that stage are (locally) optimal with respect to the overall error measure. One such strategy [15] carries out the optimisation at each stage by adjusting in turn each knot in the current knot set in order to achieve satisfactory reduction in the ℓ_2 norm, and repeating the complete adjustment as necessary. This strategy is not as poor as the traditional one-variable-at-a-time strategy for nonlinear optimisation because knots far from the newly-inserted knot tend to have little effect on the error measure.

Buffering to prevent knots coalescing and reducing the continuity of the approximant can be used. Various features can be incorporated to improve computational efficiency, including the use of contemporary nonlinear least-squares optimisation. It is emphasised that for each choice of knots the problem is linear (cf. Section 7).

8.8 Optimal discontinuous piecewise-polynomial approximation

Consider the class S_N of splines having N interior knots of multiplicity n (i.e., nN interior knots in all, counting coincidences). An $s \in S_N$ will in general be discontinuous at these knots. It is possible to determine the globally optimal locations of such knots, using the principle of dynamic programming [4]. The approach is based on the fact that the best approximant $s_N \in S_N$ to the leading p ($\geq nN$) data points is given by the best over $q = nN - n + 1, nN - n + 2, \ldots, p - N$ of $s_{N-1} \in S_{N-1}$ for the leading $q \leq p - N$ points, together with a *polynomial piece* of order n over points $q+1$ to p. By this simple recursive means the globally best knots for splines of any order that are discontinuous at any number of knots can be computed.

Such a solution may not be suitable as the final result in an application. However, it can be useful as part of a knot placement strategy. For example, suppose good knots for a spline of order n are required. An approach would be to determine an optimal discontinuous spline of order $n+1$. Use this spline to estimate f in expression (8.1). The integral in the numerator of (8.1) will be continuous piecewise linear and estimates of the optimal knots for a $C^{(n-2)}$ spline readily obtained from (8.2). Mixed results have informally been obtained by the authors with an implementation of this approach. It is suggested that it be revisited.

8.9 Knot dispersion

A set of knots of multiplicity n is positioned using an appropriate strategy, such as that in Section (8.8) and a $C^{(-1)}(I)$ spline with these knots determined. Each of these multiple knots is "dispersed", viz., replaced by n nearby simple knots, and a replacement $C^{(n-2)}(I)$ spline computed. A careful strategy for knot dispersion is required. Again, informal experiments have been made by the authors and mixed results obtained.

8.10 Knot initialisation and candidate knot locations

Several of the above procedures require or can benefit from an initial placement of the knots. Some make use of "candidate knot locations".

The solution to the free-knot spline approximation problem returned by iterative algorithms typically depends on the starting set of knots. Although an algorithm may return a result that satisfies the necessary and sufficient conditions for a solution [17], this result may be locally rather than globally optimal. There is no known characterisation of a globally optimal solution. The careful interpretation of solutions is therefore important.

The use of candidate knot positions can be helpful. For instance, it may be decided that for splines of even order, only knots that coincide with data abscissae are in the candidate set, or, for splines of odd order, knots only at points mid-way between adjacent data abscissae may be so regarded. Such criteria are consistent with the choice for interpolating splines and the generalisation covered in Section 8.2. The Lyche-Mørken knot removal algorithms [20] use data abscissae as candidate knots. The use of a finite number of candidate knot locations helps to reduce the dimensionality of the problem: there can then only be a *finite* number of possible knot sets. For large N this number can be extremely large, making it prohibitive to examine all possibilities. However, for small N, e.g., 1, 2 and 3, it may indeed be possible, and can pay dividends. Knot insertion and knot removal algorithms can also implement the concept. For example, at each stage of a knot insertion strategy, two or three knots can be inserted "simultaneously". By the method of their introduction these new knots will be optimal relative to the knots previously used and the available candidate knot locations.

Another aspect of a candidate knot set is that if it is sufficiently dense it will contain, to a degree of approximation dictated by its "spacing", the optimal knots for the given data set [19]. For instance, consider a set of $m \gg 100$ data points specified over an interval I normalised to $[-1, 1]$. Take 100 uniformly spaced points spanning this interval. This set will contain, to approximately two figures, each globally optimal knot set having $N \leq 98$ knots[5] (assuming all knots are simple). If a spline based on these 98 candidate interior knots provided a valid model, a suitable knot removal algorithm might be expected to be able to identify reasonably closely the optimal knot sets. Work is required to determine the degree of success in this regard.

9 Conclusions, discussion and future possibilities

There are theoretical difficulties associated with existence, uniqueness and characterisation of best free-knot ℓ_2 spline approximants, which influence practical considerations.

[5]The two endpoints do not constitute interior knots.

A best spline in the class of splines required may not exist. Take as $\{x_i\}_1^m$, $m = 21$ uniformly spaced values in $[-1, 1]$ and $y_i = |x_i|^3$. To see that a best ℓ_2 spline s of order 4 with three interior knots for this data may not exist, consider the choice $\lambda_1 = -\epsilon$, $\lambda_2 = 0$ and $\lambda_3 = \epsilon$. The ℓ_2 error can be made smaller than any given $\delta > 0$ for some $\epsilon > 0$. However, if the ℓ_2 error is made zero by the choice $\epsilon = 0$, the resulting three coincident knots at $x = 0$ mean that s has lower continuity than the class of splines considered. In practice, allowing knots to come "too close" together can introduce undesirable "sharpness" into the approximant. Buffering of knots [15], to ensure a minimal separation helps in this regard. The use of a candidate knot set introduces a form of buffering. In some circumstances the coalescing of knots would be ideal in terms of the resulting closeness of s to the data. In some applications the loss of smoothness would be unacceptable. Therefore, whether buffering is appropriate depends on the use to be made of s.

The solution may not be unique. Figure 6 shows a set of 201 uniformly spaced points in $[-1, 1]$ taken from $f(x) = \text{sign}(x)\min(x, 1/2)$. Figure 7 shows the root-mean-square residual as a function of knot location for ℓ_2 splines of order 4 with one interior knot. There are two best approximants, one with its knot at $x = -0.63$ and the other at $x = +0.63$. One of the two approximants is shown in Figure 6. The other spline is its skew-symmetric counterpart.

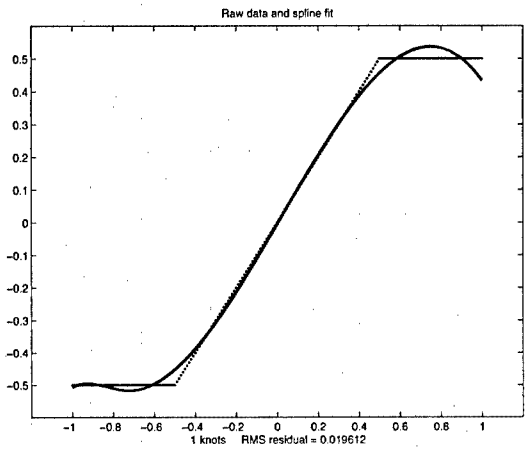


FIG. 6. 201 uniformly spaced points in $[-1, 1]$ taken from $f(x) = \text{sign}(x)\min(x, 1/2)$ and a best ℓ_2 spline approximant with one knot.

It is rarely required to determine an ℓ_2 spline approximant that is globally or even locally optimal with respect to its knots. An approximant that met some closeness requirement with the smallest possible number of knots is an academic rather than a pragmatic objective. Today, the more important consideration is to obtain an approximant that represents the data in that its smoothness is consistent with that of the function underlying the data and the uncertainties in the data. (This statement must be qualified for situations where the continuity class of splines is a consideration as discussed

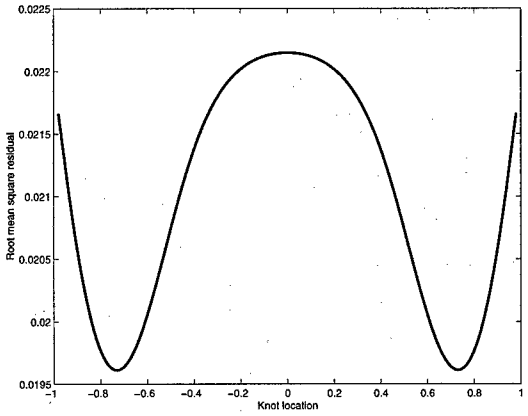


FIG. 7. The root-mean-square residual as a function of knot location for ℓ_2 spline approximants with one knot to the data of Figure 6.

above.) These ends *may* be achieved by seeking an approximant with a reasonable but not necessarily optimal number of knots.

The use of knot removal strategies is likely to attract research effort in the future. One reason for this statement is that the need to work with large initial knot sets is not as computationally prohibitive with today's powerful personal and other computers. Another reason is that the approach can be expected to produce better approximants, i.e., smaller ℓ_2 errors for the same number of knots.

The two sets of crosses in Figure 5 correspond to the values of the root-mean-square residual as a function of the number of knots for the application of the knot-insertion strategy followed by the knot-removal strategy for the thermophysical data of Figure 3. The two sets, where the "progress" takes place from left to right along the "top set", followed by right to left along "the bottom set", constitutes a form of hysteresis. The behaviour in the two directions is distinctly different. In particular, the figure indicates that once an acceptable approximation has been obtained by knot insertion, the use of knot removal can deliver an approximation of comparable quality with many fewer knots or alternatively for the same number of knots an appreciably better approximation can be obtained. In this case, with 30 knots, knot removal gives an ℓ_2 error that is one quarter of that for knot insertion. For an ℓ_2 error of 0.005, 30 knots are required using knot removal and 43 using knot insertion.

Large data sets, as are now frequently being produced in metrology from computer-controlled measuring systems, are ideal for the purpose of obtaining a sound initial approximant in the form of a valid model containing possibly many more knots than the minimum possible. Their size permits initial approximants to be obtained, even with large numbers of uniformly spaced knots, that provide valid but highly redundant models for the data. The fact that such sets do not contain "appreciable gaps", because of the manner in which they gathered, means that this fact together with the quantity of data far outweighing this initial number of knots goes a long way towards ensuring that this

initial approximant is valid. There is much scope for an appreciable number of knots to be removed. The initial large number of knots may also have been obtained by the use of a knot *insertion* strategy. It is the experience of the authors that knot insertion can introduce appreciably more knots than given by the optimal choice.

Because the early approximants may be far from optimal, an insertion algorithm can produce knots that are totally different from those in an optimal approximant. In contrast, a knot removal algorithm has a possibility to obtain good knots. (See Section 8.10.) For instance, because of the sequential manner in which knots are inserted, there may be two or more close or even coincident knots, although a good knot set might not have this property. It is also possible that such knots, although not part of an optimal set, are influential in their effect on a knot removal algorithm, with the result that they appear in the "final" approximant.

The problem of data containing wild points is not addressed satisfactorily by existing knot placement algorithms. Because such points are responsible for a large contribution to the ℓ_2 error, more knots would be placed in the neighbourhood of such a point than would otherwise had been done. The knot placement strategy can then be influenced more by the errors in the data than by the properties of the underlying function. Formulations and hence algorithms are needed that have greater resilience to such effects.

In solving the fixed-knot spline approximation problem as part of the free-knot problem, a knot set differs from a previous knot set only by the addition or removal of a small number of knots. In linear algebraic terms the "new" matrix $A(\boldsymbol{\lambda}')$, say, differs in only a few rows from the previous matrix $A(\boldsymbol{\lambda})$. Considerable gains in computational efficiency can be obtained by accounting for this fact. This paper has not addressed this issue, concentrating more on the *concepts* in the area. There is much scope, however, for the application of the recognised stable updating and downdating techniques of linear algebra [17]. Their application will not reduce the *computational complexity* of a procedure, but could reduce computation times for large problems by an appreciable factor.

The work described here was supported by the National Measurement System Policy Unit of the UK Department of Trade and Industry as part of its NMS Software Support for Metrology programme. The referee provided carefully considered comments that permitted the paper to be improved.

Bibliography

1. G. T. Anthony and M. G. Cox. The National Physical Laboratory's Data Approximation Subroutine Library. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation*, pages 669–687, Oxford, 1987. Clarendon Press.
2. BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, and OIML. Guide to the Expression of Uncertainty in Measurement, 1995. ISBN 92-67-10188-9, Second Edition.
3. H. G. Burchard. On the degree of convergence of piecewise polynomial approximations on optimal meshes. *Amer. Math. Soc.*, 234:531–559, 1977.
4. M. G. Cox. Curve fitting with piecewise polynomials. *J. Inst. Math. Appl.*, 8:36–52, 1971.

5. M. G. Cox. The numerical evaluation of B-splines. *J. Inst. Math. Appl.*, 10:134–149, 1972.
6. M. G. Cox. A survey of numerical methods for data and function approximation. In D. A. H. Jacobs, editor, *The State of the Art in Numerical Analysis*, pages 627–668, London, 1977. Academic Press.
7. M. G. Cox. The incorporation of boundary conditions in spline approximation problems. In G. A. Watson, editor, *Lecture Notes in Numerical Analysis 630: Numerical Analysis*, pages 51–63, Berlin, 1978. Springer-Verlag.
8. M. G. Cox. The least squares solution of overdetermined linear equations having band or augmented band structure. *IMA J. Numer. Anal.*, 1:3–22, 1981.
9. M. G. Cox. The NPL Data Approximation Subroutine Library: current and planned facilities. *NAG Newsletter*, 2/87:3–16, 1987.
10. M. G. Cox. Algorithms for spline curves and surfaces. In L. Piegl, editor, *Fundamental Developments of Computer-Aided Geometric Modelling*, pages 51–76, London, 1993. Academic Press.
11. M. G. Cox and J. G. Hayes. Curve fitting: a guide and suite of algorithms for the non-specialist user. Technical Report NAC 26, National Physical Laboratory, Teddington, UK, 1973.
12. C. de Boor. On calculating with B-splines. *J. Approx. Theory*, 6:50–62, 1972.
13. C. de Boor. Good approximation by splines with variable knots II. In G. A. Watson, editor, *Numerical Solution of Differential Equations, Lecture Notes in Mathematics No. 363*, pages 12–20. Springer-Verlag, Berlin, 1974.
14. C. de Boor. *A Practical Guide to Splines.* Springer-Verlag, New York, 1978.
15. C. de Boor and J. R. Rice. Least squares cubic spline approximation II - variable knots. Technical Report CSD TR 21, Purdue University, 1968.
16. B. Ford, J. Bentley, J. J. du Croz, and S. J. Hague. The NAG Library 'machine'. *Software - Practice and Experience*, 9:56–72, 1979.
17. P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Academic Press, London, 1981.
18. IEEE. IEEE standard for binary floating-point arithmetic. Technical Report ANSI/IEEE standard 754-1985, IEEE, IEEE Computer Society, New York, USA, 1985.
19. D. Jupp. Non-linear least square spline approximation. Technical report, Flinders University, Australia, 1971.
20. T. Lyche and K. Mørken. A discrete approach to knot removal and degree reduction for splines. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation*, pages 67–82, Oxford, 1987. Clarendon Press.
21. I. J. Schoenberg and Anne Whitney. On Pólya frequency functions III. *Trans. Am. Math.*, 74:246–259, 1953.

On the approximation power of local least squares polynomials

Oleg Davydov

Universität Giessen, Mathematisches Institut, D-35392 Giessen, Germany.
oleg.davydov@math.uni-giessen.de

Abstract

We discuss the relationship between the norm of the local discrete least squares polynomial approximation operator, the minimal singular value $\sigma_{\min}(P_\Xi)$ of the matrix P_Ξ of the evaluations of the basis polynomials, and the norming constant of the set of data points Ξ with respect to the space of polynomials. Since these three quantities are equivalent up to bounded constants, and since $\sigma_{\min}(P_\Xi)$ can be efficiently computed, it is feasible to use $\sigma_{\min}(P_\Xi)$ as a tool for distinguishing good local point constellations, which is useful for scattered data fitting. In addition, we give a simple new proof of a bound by Reimer for the norm of the interpolation operators on the sphere and extend it to discrete least squares operators.

1 Introduction

Let Ω be a bounded domain in $\mathbb{R}^d$, $d \geq 1$, and let $\Xi = \{\xi_1, \ldots, \xi_m\}$ be a set of scattered points in Ω. Given the values $f|_\Xi = (f(\xi_1), \ldots, f(\xi_m))^T$ of an otherwise unknown function $f : \Omega \to \mathbb{R}$, we want to reconstruct f from these data. The *least squares method* consists in choosing some linear independent functions $p_1, \ldots, p_n$ on Ω, $n \leq m$, and computing the coefficients $a_1, \ldots, a_n \in \mathbb{R}$ that minimize the ℓ_2 norm of the residual on Ξ,

$$\|f|_\Xi - p|_\Xi\|_2 = \Big(\sum_{i=1}^m |f(\xi_i) - p(\xi_i)|^2\Big)^{1/2},$$

with $p = a_1 p_1 + \cdots + a_n p_n \in \mathcal{P} := \operatorname{span}\{p_1, \ldots, p_n\}$. Let $\mathcal{P}|_\Xi := \operatorname{span}\{p_1|_\Xi, \ldots, p_n|_\Xi\}$. If $\dim \mathcal{P}|_\Xi = n$, then the least squares solution is unique, and we denote it by $L_{\mathcal{P},\Xi} f$. Note that the minimum norm solution available in the case of a rank deficient problem ($\dim \mathcal{P}|_\Xi < n$) seems less useful since in general it does not reproduce the elements of $\mathcal{P}$ exactly.

The computation of least squares approximation $L_{\mathcal{P},\Xi} f$ of f is expensive if m and n are large. To obtain a scattered data fitting algorithm with *linear complexity* with respect to the size of data, a *two-stage method* [8] can be employed which consists in 1) covering the original domain Ω with a number of subdomains Ω_k each containing only a small subset $\Xi_k = \Xi \cap \Omega_k$ of Ξ, computing *local* approximations to the data in Ξ_k, and 2) using the information obtained from these local approximations to build the final approximation of the (possibly huge) original data set. The least squares method

can be employed in the local approximation stage, especially to deal with "real world" data usually contaminated with errors or just containing undesirable "high frequency" components.

If $\mathcal{P}$ is chosen to be the space Π_q^d of algebraic polynomials in d variables of a suitable degree q, then $n = \binom{d+q}{d}$. To achieve high approximation order, it is desirable to choose q such that n is only a little smaller then m. However, this is not always possible due to the rank deficiency or ill-conditioning of the least squares problem, which is especially difficult to control if $\xi_1, \ldots, \xi_m \in \Xi_k$ are unevenly distributed in Ω_k. This difficulty can in principle be overcome by constructing, for each Ξ_k, a suitable subspace of higher degree polynomials (least interpolation space [2]). If, however, the polynomial degree is not allowed to exceed a fixed small value, then a common practical approach is to choose larger sets $\Xi_k \subset \Xi$, with m substantially greater than n, see *e.g.* [4] where it is suggested to use for local least squares approximation $m = 11$ points if $\mathcal{P} = \Pi_2^2$ with $n = 6$ and $m = 15$ points if $\mathcal{P} = \Pi_3^2$ with $n = 10$. However, even these higher m provide no guaranty that the matrix

$$P_{\Xi_k} := [p_j(\xi_i) \,:\, i = 1, \ldots, m, \quad j = 1, \ldots, n]$$

of the local least squares problem will always be well-conditioned. Moreover, for some data, this method may lead to the use of inappropriately distant points for the local approximation.

The purpose of this paper is to draw attention to the fact that the conditioning of the matrix P_{Ξ_k} is not only the issue of numerical stability of the computation of least squares. Indeed, the reciprocal of the minimal singular value $\sigma_{\min}(P_\Xi)$ of P_Ξ provides a bound for the norm of the least squares operator $L_{\mathcal{P},\Xi}$ if both m and n are small. Therefore, the approximation power of local least squares depends on $\sigma_{\min}(P_\Xi)$ and the best approximation from $\mathcal{P}$. Since $\sigma_{\min}(P_\Xi)$ can be efficiently computed for a small matrix P_Ξ by well known numerical algorithms, it is feasible to use it as a tool to decide whether a particular portion of data is suitable for building local least squares approximation from $\mathcal{P}$ with reasonable approximation power. If $\sigma_{\min}(P_\Xi)$ is too small, then either Ξ or $\mathcal{P}$ should be modified, *e.g.* by adding more points to Ξ or using an appropriate subspace of $\mathcal{P}$. A two-stage algorithm for fitting large irregularly distributed scattered data sets employing the conditioning of the local observation matrices P_{Ξ_k} is studied in [3, 5].

The paper is organized as follows. In Section 2 we discuss the relationship between the norm of the discrete least squares approximation operator, the minimal singular value $\sigma_{\min}(P_\Xi)$, and the *norming constant* $\nu(\mathcal{P}, \Xi)$. As a by-product, we obtain a new proof of a known bound for the norm of the interpolation operators on the sphere [7], and extend it to the discrete least squares operators. Section 3 illustrates the above concepts in the univariate case, when they are also related to the *separation distance* of Ξ, while Section 4 is devoted to a discussion of the least squares multivariate polynomial approximation.

2 Bounds for $\|L_{\mathcal{P},\Xi}\|$ and approximation error

Let $p_1,\dots,p_n$ be linearly independent continuous functions on $\Omega \subset \mathbb{R}^d$ spanning a linear space $\mathcal{P}$. Since all norms on a finite dimensional linear space are equivalent, there are positive constants K_1, K_2 such that

$$K_1\|a\|_2 \le \left\| \sum_{j=1}^n a_j p_j \right\|_{C(\Omega)} \le K_2\|a\|_2 \tag{2.1}$$

for any coefficient vector $a = (a_1,\dots,a_n)^T \in \mathbb{R}^n$.

Given $\Xi = \{\xi_1,\dots,\xi_m\} \subset \Omega$, we consider the matrix $P_\Xi \in \mathbb{R}^{m\times n}$ as defined in the introduction. Obviously, $\operatorname{rank} P_\Xi = \dim \mathcal{P}|_\Xi$. If P_Ξ has full rank, then $\dim \mathcal{P}|_\Xi = n$, and the least squares approximation $L_{\mathcal{P},\Xi}f$ is uniquely determined, giving rise to the operator $L_{\mathcal{P},\Xi} : C(\Omega) \to \mathcal{P} \subset C(\Omega)$.

It is easy to see that $L_{\mathcal{P},\Xi}$ exactly reproduces the elements of $\mathcal{P}$, *i.e.*,

$$L_{\mathcal{P},\Xi}p = p, \qquad \text{all } p \in \mathcal{P}. \tag{2.2}$$

Therefore, a standard argument shows that

$$\|f - L_{\mathcal{P},\Xi}f\|_{C(\Omega)} \le (1 + \|L_{\mathcal{P},\Xi}\|)E(f,\mathcal{P})_{C(\Omega)}, \tag{2.3}$$

where $E(f,\mathcal{P})_{C(\Omega)}$ denotes the error of the best approximation of f from $\mathcal{P}$ in Chebyshev norm,

$$E(f,\mathcal{P})_{C(\Omega)} := \inf_{p\in\mathcal{P}} \|f-p\|_{C(\Omega)}.$$

Thus, an estimate for $\|L_{\mathcal{P},\Xi}\|$ immediately gives an upper bound for $\|f - L_{\mathcal{P},\Xi}f\|_{C(\Omega)}$.

The *norming constant* $\nu(\mathcal{P},\Xi)$ of Ξ with respect to $\mathcal{P}$ [6] can be defined by

$$\nu(\mathcal{P},\Xi) = \min_{p\in\mathcal{P}} \|p|_\Xi\|_\infty / \|p\|_{C(\Omega)}. \tag{2.4}$$

Given any matrix A, we denote by $\sigma_{\min}(A)$ the *minimal singular value*

$$\sigma_{\min}(A) = \min_{\|x\|_2=1} \|Ax\|_2.$$

Recall that if A has full rank, then $\sigma_{\min}(A) = \|A^+\|_2^{-1}$, where A^+ is the pseudoinverse of A, see *e.g.* [1].

Theorem 2.1 *If* $\operatorname{rank} P_\Xi = n$, *then*

$$K_1/\sigma_{\min}(P_\Xi) \le \|L_{\mathcal{P},\Xi}\| \le K_2\sqrt{m}/\sigma_{\min}(P_\Xi), \tag{2.5}$$

$$1/\nu(\mathcal{P},\Xi) \le \|L_{\mathcal{P},\Xi}\| \le \sqrt{m}/\nu(\mathcal{P},\Xi), \tag{2.6}$$

$$K_1\nu(\mathcal{P},\Xi) \le \sigma_{\min}(P_\Xi) \le K_2\sqrt{m}\,\nu(\mathcal{P},\Xi). \tag{2.7}$$

Proof: We first prove (2.5). Let $L_{\mathcal{P},\Xi}f = \sum_{j=1}^n a_j p_j$. It follows by a well-known result in numerical linear algebra that the vector $a = (a_1,\dots,a_n)^T$ can be computed as the product of the pseudoinverse P_Ξ^+ of P_Ξ with the vector $f|_\Xi$. Therefore,

$$\|a\|_2 = \|P_\Xi^+ f|_\Xi\|_2 \le \|P_\Xi^+\|_2\|f|_\Xi\|_2 = \sigma_{\min}^{-1}(P_\Xi)\|f|_\Xi\|_2.$$

Since $\|L_{\mathcal{P},\Xi} f\|_{C(\Omega)} \le K_2 \|a\|_2$ and $\|f|_\Xi\|_2 \le \sqrt{m}\,\|f|_\Xi\|_\infty \le \sqrt{m}\,\|f\|_{C(\Omega)}$, the upper bound in (2.5) follows. To prove the lower bound in (2.5), we choose a function $\tilde{f} \in C(\Omega)$ such that

$$\|P_\Xi^+ \tilde{f}|_\Xi\|_2 = \|P_\Xi^+\|_2 \|\tilde{f}|_\Xi\|_2, \quad \|\tilde{f}|_\Xi\|_\infty = \|\tilde{f}\|_{C(\Omega)},$$

which is obviously possible. Then by (2.1) we have

$$\|L_{\mathcal{P},\Xi}\tilde{f}\|_{C(\Omega)} \ge K_1 \|P_\Xi^+ \tilde{f}|_\Xi\|_2 = K_1 \sigma_{\min}^{-1}(P_\Xi)\|\tilde{f}|_\Xi\|_2,$$

which implies the desired lower bound since $\|\tilde{f}|_\Xi\|_2 \ge \|\tilde{f}|_\Xi\|_\infty = \|\tilde{f}\|_{C(\Omega)}$.

Since $\|L_{\mathcal{P},\Xi} f\|_{C(\Omega)} \le \nu^{-1}(\mathcal{P},\Xi)\|(L_{\mathcal{P},\Xi} f)|_\Xi\|_\infty$, the upper bound in (2.6) follows by

$$\|(L_{\mathcal{P},\Xi} f)|_\Xi\|_\infty \le \|(L_{\mathcal{P},\Xi} f)|_\Xi\|_2 \le \|f|_\Xi\|_2 \le \sqrt{m}\|f\|_{C(\Omega)}.$$

To prove the lower bound, we denote by $\tilde{p}$ an element of $\mathcal{P}$ for which the minimum in (2.4) is attained and choose a function $\tilde{f} \in C(\Omega)$ such that $\tilde{f}|_\Xi = \tilde{p}|_\Xi$ and $\|\tilde{f}\|_{C(\Omega)} = \|\tilde{f}|_\Xi\|_\infty$. Then by (2.2),

$$\|L_{\mathcal{P},\Xi}\tilde{f}\|_{C(\Omega)} = \|\tilde{p}\|_{C(\Omega)} = \nu^{-1}(\mathcal{P},\Xi)\|\tilde{p}|_\Xi\|_\infty = \nu^{-1}(\mathcal{P},\Xi)\|\tilde{f}\|_{C(\Omega)},$$

which implies $\|L_{\mathcal{P},\Xi}\| \ge \nu^{-1}(\mathcal{P},\Xi)$.

We finally establish (2.7). For any $p \in \mathcal{P}$, let $p = \sum_{j=1}^n a_j p_j$ and $a = (a_1,\dots,a_n)^T$. Then $p|_\Xi = P_\Xi a$ and hence

$$\|p|_\Xi\|_\infty \le \|P_\Xi a\|_2 \le \sqrt{m}\,\|p|_\Xi\|_\infty.$$

Since

$$\sigma_{\min}(P_\Xi) = \min_{a\in\mathbf{R}^n} \|P_\Xi a\|_2/\|a\|_2,$$

(2.7) follows by (2.1). □

In view of (2.3), the upper bound in (2.5) implies

$$\|f - L_{\mathcal{P},\Xi} f\|_{C(\Omega)} \le (1 + K_2\sqrt{m}/\sigma_{\min}(P_\Xi))E(f,\mathcal{P})_{C(\Omega)}, \tag{2.8}$$

which shows that the approximation power of discrete least squares proportionally reduces if $\sigma_{\min}(P_\Xi)$ (or $\nu(\mathcal{P},\Xi)$) is small. We will discuss some practical consequences of this fact in the next two sections.

Although $\nu(\mathcal{P},\Xi)$ gives tighter bounds for $\|L_{\mathcal{P},\Xi}\|$, $\sigma_{\min}(P_\Xi)$ has a clear practical advantage that it is easily computable by using *e.g.* the singular value decomposition of the small "local" matrix P_Ξ. On the other hand, the norming constants were used in [6, 9] to derive estimates for the approximation error of radial basis function interpolation and moving least squares, respectively.

Remark 2.2 If $p_1,\dots,p_n$ is an *orthonormal basis* for $\mathcal{P}$, then $\|a\|_2 = \|p\|_{L_2(\Omega)}$, $p = \sum_{j=1}^n a_j p_j$, and the constants K_1, K_2 in (2.1) are closely related to *Nikolskii constants* of the space $\mathcal{P}$, namely,

$$K_1 = N_{2,\infty}^{-1}(\mathcal{P}), \qquad K_2 = N_{\infty,2}(\mathcal{P}),$$

where

$$N_{q_1,q_2}(\mathcal{P}) := \max_{p\in\mathcal{P}} \|p\|_{L_{q_1}(\Omega)}/\|p\|_{L_{q_2}(\Omega)}, \qquad 1 \le q_1, q_2 \le \infty.$$

In particular, if $\Omega = S^{d-1}$, the unit sphere in $\mathbb{R}^d$, and $\{p_1, \dots, p_n\}$ is the set of *spherical harmonics* forming an orthonormal basis for the space $\mathcal{P} = \mathcal{H}_q^d$ of spherical polynomials of degree q in d variables, then it is not difficult to prove that $K_2 = N_{\infty,2}(\mathcal{H}_q^d) = \sqrt{n/|S^{d-1}|}$, where $|S^{d-1}|$ denotes the surface area of S^{d-1}. Therefore, for any set $\Xi \subset S^{d-1}$ with $\#\Xi = m \geq n$, we have by (2.5),

$$\|L_{\mathcal{H}_q^d,\Xi}\| \leq \sqrt{nm/|S^{d-1}|}/\sigma_{\min}(P_\Xi), \tag{2.9}$$

which recovers in the case of interpolation ($m = n$) an error bound by Reimer [7] originally proved by using Lagrangian square sums (see also [10]).

3 Univariate polynomials

Let Ω be an interval $[-h, h]$ on the real line $\mathbb{R}$, and let

$$p_j(t) = (t/h)^{j-1}, \qquad j = 1, \dots, n.$$

Then $\mathcal{P}$ is the restriction to $[-h, h]$ of the space Π_{n-1}^1 of all univariate polynomials of degree at most $n-1$. By the well-known interpolation properties of the univariate polynomials, $\operatorname{rank} P_\Xi = n$ for any $\Xi = \{\xi_1, \dots, \xi_m\} \subset [-h, h]$, $m \geq n$, with distinct ξ_i's.

For any $\Xi' = \{\xi_{i_1}, \dots, \xi_{i_n}\} \subset \Xi$, let $q_{\Xi'}$ denote the *separation distance*,

$$q_{\Xi'} := \frac{1}{2} \min_{j \neq k} |\xi_{i_j} - \xi_{i_k}|.$$

The Lebesgue constant $\|L_{\mathcal{P},\Xi'}\|$ of the corresponding interpolation scheme can be easily estimated as

$$\|L_{\mathcal{P},\Xi'}\| \leq \frac{2^{n-1}}{(n-1)!} (h/q_{\Xi'})^{n-1}.$$

Since Ξ' may be any subset of Ξ of cardinality n and since $\nu(\mathcal{P}, \Xi) \geq \|L_{\mathcal{P},\Xi'}\|^{-1}$, we get

$$\nu^{-1}(\mathcal{P}, \Xi) \leq \frac{2^{n-1}}{(n-1)!} (h/q_{\Xi,n})^{n-1},$$

where

$$q_{\Xi,n} := \max_{\substack{\Xi' \subset \Xi \\ \#\Xi' = n}} q_{\Xi'}.$$

Hence, by (2.3) and (2.6),

$$\|f - L_{\Pi_{n-1}^1,\Xi} f\|_{C[-h,h]} \leq \left(1 + \frac{\sqrt{m}\, 2^{n-1}}{(n-1)!} (h/q_{\Xi,n})^{n-1}\right) E(f, \Pi_{n-1}^1)_{C[-h,h]}. \tag{3.1}$$

This last estimate shows that the univariate least squares polynomials have the approximation power of the best local polynomial approximation as $h \to 0$ provided $h/q_{\Xi,n}$ remains bounded. However, if the scattered points $\xi_1, \dots, \xi_m \in [-h, h]$ are clustered together in at most $n-1$ very tight groups, then $q_{\Xi,n}$ may be arbitrarily small, thus forcing the right hand side of (3.1) to blow up. To figure out what happens to $\|f - L_{\Pi_{n-1}^1,\Xi} f\|_{C[-h,h]}$ in these circumstances, we consider the following example.

Let $h = 1$, $n = 2$, $f(t) = t^2 - 1/2$, and $\Xi = \{-\xi, 0, \xi\}$ for some $0 < \xi \leq 1$. It is easy to see that $L_{\Pi_1^1,\Xi} f \equiv -1/2 + 2\xi^2/3$. Since $E(f, \Pi_1^1)_{C[-1,1]} = 1/2$, we have

$$\|f - L_{\Pi_1^1,\Xi} f\|_{C[-1,1]} = 1/2 + |1/2 - 2\xi^2/3| \leq 2E(f, \Pi_1^1)_{C[-1,1]}$$

even though, by a simple calculation,

$$\|L_{\Pi_1^1,\Xi}\| = 1/3 + 1/\xi,$$

$$\sqrt{2}/\sigma_{\min}(P_\Xi) = 1/\nu(\mathcal{P}, \Xi) = 1/q_{\Xi,2} = 1/\xi \to \infty \quad \text{as} \quad \xi \to 0.$$

This may contribute to the opinion that $\|L_{\Pi_1^1,\Xi}\|$, $\sigma_{\min}(P_\Xi)$, $\nu(\mathcal{P}, \Xi)$ and $q_{\Xi,n}$ are not the right quantities to describe the behaviour of the approximation. Indeed, as the three points $-\xi, 0, \xi$ coalesce, $L_{\Pi_1^1,\Xi} f$ converges to a Hermite interpolation polynomial provided the entries of P_Ξ as well as the values of $f|_\Xi$ are exact. However, if we simulate "real world" data by adding to $f(-\xi), f(0), f(\xi)$ normally distributed errors with standard deviation 10^{-4}, then the picture substantially changes. Table 1 shows that $\|f - L_{\Pi_1^1,\Xi} f\|_{C[-1,1]}$ does blow up in this case. For comparison we also include in the table the error of $\|f - L_{\Pi_0^1,\Xi} f\|_{C[-1,1]}$ for the same contaminated data.

TAB. 1 Average (d^1_{mean}) and maximum ($d^1_{\max}$) of $\|f - L_{\Pi_1^1,\Xi} f\|_{C[-1,1]}$ as well as maximum of $\|f - L_{\Pi_0^1,\Xi} f\|_{C[-1,1]}$ ($d^0_{\max}$) in 1000 tests with contaminated data

ξ	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
d^1_{mean}	1.06	1.56	6.63	57.3	564	5630
$d^1_{\max}$	1.24	3.39	24.9	240	2390	23900
$d^0_{\max}$	1.00018	1.00018	1.00018	1.00018	1.00018	1.00018

Thus, if $q_{\Xi,n}$ is too small, we cannot practically achieve with least squares the approximation order of $E(f, \Pi^1_{n-1})_{C[-h,h]}$ simply because the points lying too close to each other carry redundant information and we have at most $n - 1$ clusters of such points. Therefore, we should adjust the polynomial degree to the given data paying attention to the trade-off between higher approximation power of higher degree polynomials and the "pollution" caused by the factor $q^{-1}_{\Xi,n}$ that increases with n. In practice one may choose maximal n such that $h/q_{\Xi,n}$ is smaller than a prescribed tolerance value $0 < E < \infty$.

4 Multivariate polynomials

The situation becomes substantially more complicated when we turn to multivariate polynomials. Let Ω be a bounded domain in $\mathbb{R}^d$ and let $\{p_1, \ldots, p_n\}$, $n = \binom{d+q}{d}$, be a basis of the space $\mathcal{P} = \Pi^d_q$ of polynomials in d variables of total degree q satisfying (2.1) on Ω. (For example, we may consider a properly scaled standard power basis with the center at a point in Ω or the Bernstein-Bézier basis with respect to some simplex overlapping Ω or a significant part of it.) Let, furthermore, Ξ be an arbitrary finite set of points in Ω such that $m = \#\Xi \geq n$.

The first problem we face in the case $d \geq 2$ is that the matrix P_Ξ may be rank deficient. It is clear, however, that there is no practical difference between this situation and the

one when P_Ξ has full rank but is extremely ill-conditioned, *i.e.*, $\sigma_{\min}(P_\Xi)$ is very small. Moreover, (2.8) shows that even moderately small $\sigma_{\min}(P_\Xi)$ may significantly reduce the approximation power of $L_{\mathcal{P},\Xi}$. Clearly, the same can also happen in the univariate case if $q_{\Xi,n}$ is too small. The real difficulty of the multivariate case seems to be that simple characteristics of Ξ, like separation distance $q_{\Xi,n}$, do not give much information about the norm of $L_{\mathcal{P},\Xi}$. For example, six equidistant points on the unit circle in $\mathbb{R}^2$ are well separated and look reasonably distributed. However, they are not good for least squares approximation from the space Π_2^2 since the matrix P_Ξ is rank deficient. Suitably perturbed, these points will give rise to the least squares operator $L_{\Pi_2^2,\Xi}$ with a very large norm. More generally, the norm of $L_{\Pi_q^d,\Xi}$ will be large if the points in $\Xi \subset \mathbb{R}^d$ lie "too close" to an algebraic hypersurface of order q.

If the data are comparatively dense in Ω, namely the *fill distance*

$$h_{\Xi,\Omega} := \sup_{x\in\Omega} \min_{\xi\in\Xi} |x-\xi|$$

does not exceed some small positive constant depending on Ω and the polynomial degree, then the estimates of the norming constant $\nu(\Pi_q^d,\Xi)$ given in [9] provide a bound for $\|L_{\Pi_q^d|_\Omega,\Xi}\|$, in view of (2.6). For example, if Ω is a ball of radius r, then $\nu(\Pi_q^d|_\Omega,\Xi) \geq 1/2$ if $h_{\Xi,\Omega} < 0.11r/q^2$.

On the other hand, without any density assumptions we can always rely on (2.8), where $\sigma_{\min}(P_\Xi)$ can be efficiently computed by well known algorithms of numerical linear algebra. In some sense, small $\sigma_{\min}(P_\Xi)$ indicates that the local data has "hidden redundancies" (*e.g.* too many points lying very close to the same straight line or the same ellipse) that prevent it from carrying enough information for a "full power" approximation of the underlying function from Π_q^d. Similar to the univariate case, but using $\sigma_{\min}(P_\Xi)$ instead of $q_{\Xi,n}$, we can adaptively choose the polynomial degree according to the following algorithm that has proven to be useful for scattered data fitting [3, 5].

Let $\Omega \subset \mathbb{R}^d$, $\Xi \subset \Omega$, $\#\Xi = m$. Denote by P_Ξ^q the matrix of the evaluations of appropriate basis functions for Π_q^d, $q \geq 0$, at the points $\xi \in \Xi$.

Algorithm 4.1 *Starting with some $q = q_0 \geq 0$ such that $\binom{d+q}{d} \leq m$, compute $\sigma_{\min}(P_\Xi^q)$. If $1/\sigma_{\min}(P_\Xi^q)$ is smaller than a prescribed tolerance $E < \infty$, then compute the least squares Π_q^d-approximation to the data in Ξ and accept it as a reliable approximation on Ω. Otherwise, repeat the same with $q = q_0 - 1$ and successively reduce the degree q to $q_0 - 2, \ldots, 0$, while $1/\sigma_{\min}(P_\Xi^q) \geq E$. For $q = 0$ no comparison of $1/\sigma_{\min}(P_\Xi^0)$ with E is needed since $\|L_{\Pi_0^d|_\Omega,\Xi}\|$ is bounded for any Ω and Ξ.*

Note that, optionally, the *condition number* $\|P_\Xi^q\|_2/\sigma_{\min}(P_\Xi^q)$ of P_Ξ^q can be used in the above algorithm instead of $1/\sigma_{\min}(P_\Xi^q)$, as it has been formulated in [5].

Bibliography

1. Å. Björck, *Numerical Methods for Least Squares Problems,* SIAM, Philadelphia, 1996.

2. C. de Boor and A. Ron, On multivariate polynomial interpolation, *Constr. Approx.* **6** (1990), 287–302.
3. O. Davydov and F. Zeilfelder, Scattered data fitting by direct extension of local polynomials with bivariate splines, 2002, preprint.
4. T. A. Foley, Scattered data interpolation and approximation with error bounds, *Comput. Aided Geom. Design* **3** (1986), 163–177.
5. J. Haber, F. Zeilfelder, O. Davydov and H.-P. Seidel, Smooth approximation and rendering of large scattered data sets, in *Proceedings of IEEE Visualization 2001,* Th. Ertl, K. Joy and A. Varshney (eds), IEEE, 2001, 341–347, 571.
6. K. Jetter, J. Stöckler and J. Ward, Error estimates for scattered data interpolation on spheres, *Math. Comp.* **226** (1999), 733–747.
7. M. Reimer, Interpolation on the sphere and bounds for the Lagrangian square sums, *Results in Mathematics* **11** (1987), 144–164.
8. L. L. Schumaker, Two-stage methods for fitting surfaces to scattered data, in *Quantative Approximation,* R. Schaback and K. Scherer (eds), Lecture Notes 556, Springer, Berlin, 1976, 378–389.
9. H. Wendland, Local polynomial reproduction and moving least squares approximation, *IMA J. Numer. Anal.* **21** (2001), 285–300.
10. R. S. Womersley and I. H. Sloan, How good can polynomial interpolation on the sphere be?, *Advances in Comp. Math.* **14** (2001), 195–226.

A wavelet-based preconditioning method for dense matrices with block structure

Judith M. Ford* **and Ke Chen**

Department of Mathematical Sciences, University of Liverpool, Liverpool L69 7ZL, UK.
Judyford@liv.ac.uk, k.chen@liv.ac.uk

Abstract

In recent years application of a discrete wavelet transform (DWT) has become an established tool for the design of preconditioners for smooth, dense matrices, such as those that arise in the solution of certain integral equations. In this paper we consider the higher dimensional case, where the matrix A is not itself smooth, but has a smooth block structure. To precondition such matrices, we use repeated application of a level 1 *block-wise* DWT to exploit the fact that corresponding entries in adjacent blocks are close in value. We illustrate the effectiveness of our methods by means of numerical examples.

1 Introduction

We have previously ([9]) considered wavelet-based preconditioning methods for dense matrices having the property that the entries vary smoothly (that is to say, adjacent entries are close in value) apart from known areas of singularity, for example a non-smooth diagonal band. The main idea is to use wavelet compression (see, for example [14]) to convert "smoothness" in the original matrix into "smallness" in the transformed matrix, and then to approximate the transformed matrix by dropping small entries. Smooth matrices arise in a range of applications (see, for example, [6, 8, 10]) involving an essentially 1-dimensional discretization process. In higher dimension cases the corresponding matrices have a block structure: each block is smooth and corresponding entries in different blocks vary smoothly; but discontinuities at the edges of the blocks mean that standard application of DWT does not give good compression. In this paper we extend the ideas of [9] to enable preconditioners to be designed for such matrices. Throughout we use Daubechies wavelets, which are orthogonal and have compact support.

2 DWT-based preconditioners

We are interested in fast solution of linear systems

$$Ax = b, \quad x, b \in \mathbb{C}^n, \; A \in \mathbb{C}^{n \times n}, \tag{2.1}$$

where A is a large, dense matrix. Krylov subspace iterative methods, such as GMRES (described in [13]), can be used to solve (2.1), but in most cases preconditioning is

* The first author was supported by the Engineering and Physical Sciences Research Council, UK

required in order to obtain good convergence. One method of preconditioning is to seek a matrix $M \approx A$ such that $M^{-1}v$ can be calculated cheaply for any vector v. For smooth dense A the task is usually made easier by transforming (2.1) into a wavelet basis (see e.g. [4, 5, 6, 10, 11]). When a DWT is applied to such an A, the resulting matrix $\tilde{A}$ has many small elements. A sparse $\bar{A} \approx \tilde{A}$ can be obtained by setting to zero small elements. This is the main idea underlying most wavelet-based preconditioners.

2.1 Preconditioners for 1-D problems

Typically A is smooth apart from a narrow diagonal band. When a level k standard DWT is applied $\tilde{A}$ has a 'finger' pattern of large entries (caused by the non-smooth diagonal feature) and an $n/2^k \times n/2^k$ block of large entries at the top-left corner. Here n should be a power of 2. We can form a preconditioner $M \approx \tilde{A}$ by setting to zero entries that fall below some chosen threshold, but, because of the finger pattern, a large amount of fill-in occurs under LU factorization. To avoid this problem M can be obtained by setting to zero entries in $\tilde{A}$ that fall outside of a diagonal band. We describe this approach as a "band cut".

The finger pattern can be avoided by using DWTPer (DWT with permutations, first proposed in [6], see also [7]), which centres the fingers to form a sparse diagonal band whose width can be predicted accurately. M can then be formed by applying a band cut to $\tilde{A}$ and (optionally) imposing a threshold.

An alternative way of avoiding the creation of a finger pattern matrix is to use the Non-Standard-forms (NS-forms) of Beylkin, Coifman and Rokhlin (see [3]) to represent A in terms of the blocks of a larger matrix. In [9] we presented a new way of using the NS-form submatrices to precondition A based on the Schur complement and recursive application of a flexible GMRES iteration. We compared four alternative DWT-based preconditioning methods:

P1 standard DWT preconditioner with band cut ([5]),
P2 DWTPer preconditioner with band cut ([6, 10]),
P3 NS-form preconditioner with threshold ([3, 11]),
P4 Recursive Schur complement preconditioner ([9]),

and found that, for smooth matrices with a diagonal singularity, **P4** gave consistently good performance, **P1** performed well for moderate singularities and **P2** was best when the diagonal singularity was very pronounced. When we came to consider 2-D problems, the robustness of **P4** encouraged us to consider ways of extending it to higher dimensions.

2.2 Extension to matrices with block structure

In the 2-dimensional case we are concerned with matrices that have a smooth *block* structure. We can compress dense block matrices of this type using two different types

of DWT: The **block DWT** has a transform matrix of the form

$$W_B^{(m,n)} = I_m \otimes W^{(n)} = \begin{pmatrix} W^{(n)} & 0 & \cdots & 0 \\ 0 & W^{(n)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & W^{(n)} \end{pmatrix}, \tag{2.2}$$

where $W^{(n)}$ is a standard $n \times n$ DWT matrix and 0 is the $n \times n$ zero matrix. It exploits smoothness *within* blocks. The **Big Block DWT (BBDWT)** exploits smoothness *between* blocks. It has a transform matrix of the form

$$\begin{aligned} W_{BB}^{(m,n)} &= W^{(m)} \otimes I_n \\ &= \begin{pmatrix} h_0 I & h_1 I & \cdots & h_{D-1} I & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & h_0 I & h_1 I & \cdots & h_{D-1} I & 0 & \cdots & \cdots & 0 \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & & & & \ddots & \ddots & \ddots & \ddots & & \vdots \\ h_2 I & \cdots & h_{D-1} I & 0 & \cdots & \cdots & \cdots & 0 & h_0 I & h_1 I \\ g_0 I & g_1 I & \cdots & g_{D-1} I & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & g_0 I & g_1 I & \cdots & g_{D-1} I & 0 & \cdots & \cdots & 0 \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & & & & \ddots & \ddots & \ddots & \ddots & & \vdots \\ g_2 I & \cdots & g_{D-1} I & 0 & \cdots & \cdots & \cdots & 0 & g_0 I & g_1 I \end{pmatrix} \end{aligned} \tag{2.3}$$

where $h_0, \ldots, h_{D-1}$ and $g_0, \ldots, g_{D-1}$ are the low-pass and high-pass filter coefficients respectively (D being the order of the wavelet transform), I is the $n \times n$ identity matrix and 0 is the $n \times n$ zero matrix. The resulting transformed matrix has a 'finger' structure of blocks, each with a diagonal structure. We can avoid the finger pattern by permuting the rows and columns of the transformed matrix so as to centre the blocks containing large entries. We call this modified big block transform BBDWTPer, because it is a big block version of the DWTPer transform described in [10]. We anticipate that BBDWTPer may be useful for preconditioning block matrices with a very strong block diagonal singularity (see the comparison of DWTPer and other DWT-based preconditioners in [9]), but we have not yet found example matrices for which BBDWTPer provides a good preconditioner. Preconditioners based on BBDWT and BBDWTPer are tested in Section 4; we now present a more effective method.

3 Recursive BBDWT-based preconditioning

An alternative way of avoiding the 'finger' pattern is to use a 'Big Block' version of the NS-forms presented in [3]. We define the Big Block NS-form (BBNS-form) of a matrix as follows. To transform a matrix consisting of m^2 blocks, each of dimension n (where m and n are powers of 2) we define P_i, Q_i to be the $mn/2^i \times mn/2^{i-1}$ matrices such that

$$W_{BB}^{(m/2^{(i-1)},n)} = \begin{pmatrix} P_i \\ Q_i \end{pmatrix}. \tag{3.1}$$

Given an $mn \times mn$ matrix A, define $T_0 = A$,

$$T_i = P_i T_{i-1} P_i^T, \quad A_i = Q_i T_{i-1} Q_i^T, \quad B_i = Q_i T_{i-1} P_i^T, \quad C_i = P_i T_{i-1} Q_i^T, \tag{3.2}$$

$$\tilde{T}_i = \begin{pmatrix} A_{i+1} & B_{i+1} \\ C_{i+1} & T_{i+1} \end{pmatrix}. \tag{3.3}$$

The level k BBNS-form of A comprises T_k together with A_i, B_i, C_i, $i = 1, 2, \ldots, k$. (The blocks of $\tilde{T}_i$ are arranged differently from those of the standard level 1 DWT of T_i. We have used this ordering in order to be consistent with the notation of [3].)

We propose to use banded approximations to the submatrices of the BBNS-form as the basis for our preconditioner. If the blocks of A vary smoothly apart from a diagonal block band, then each of A_{i+1}, B_{i+1}, C_{i+1} will have small entries except for a wrap-round diagonal block band. So we can approximate them by $\bar{A}_{i+1}$, $\bar{B}_{i+1}$, $\bar{C}_{i+1}$, formed by cutting to a block band, giving an approximation $\bar{T}_i$ to $\tilde{T}_i$:

$$\bar{T}_i = \begin{pmatrix} \bar{A}_{i+1} & \bar{B}_{i+1} \\ \bar{C}_{i+1} & T_{i+1} \end{pmatrix}. \tag{3.4}$$

(In practice, it is unnecessary to compute A_{i+1}, B_{i+1}, C_{i+1} and then to set entries outside the block band to zero; instead we can compute only the non-zero entries of $\bar{A}_{i+1}$, $\bar{B}_{i+1}$, $\bar{C}_{i+1}$. This enables us to reduce the cost of forming $\bar{T}_i$.)

We now show how this can help us to solve (2.1). We use a flexible GMRES iteration (see [12]) preconditioned by approximate solution of an equation of the form $Ay = v$ at each step. To do this we first apply a level 1 BBDWT with a block band cut to give

$$\begin{pmatrix} \bar{A}_1 & \bar{B}_1 \\ \bar{C}_1 & T_1 \end{pmatrix} \begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \end{pmatrix} = \begin{pmatrix} \tilde{v}_1 \\ \tilde{v}_2 \end{pmatrix}, \tag{3.5}$$

where $\tilde{y}_1 = Q_1 y$, $\tilde{y}_2 = P_1 y$, $\tilde{v}_1 = Q_1 v$, $\tilde{v}_2 = P_1 v$. We solve this equation using the Schur complement $S_1 = T_1 - \bar{C}_1 \bar{A}_1^{-1} \bar{B}_1$. This requires us to solve an equation of the form

$$S_1 \tilde{y}_2 = \tilde{w}_2, \tag{3.6}$$

which we do by a further GMRES iteration. We expect that T_1 will be an effective preconditioner for S_1 (see [1]§9.3), so we now seek a cheap way of applying T_1^{-1} to a vector. To do this we repeat the process of applying a level 1 BBDWT and using the Schur complement. In summary, during the solution of (3.6) we solve a preconditioning equation of the form

$$T_1 y = v, \quad y, v \in \mathbb{C}^{mn/2}. \tag{3.7}$$

To do this cheaply we repeat the process of applying a level 1 BBDWT and using the Schur complement and obtain an equation of the form

$$S_2 z = w, \quad z, w \in \mathbb{C}^{mn/4}. \tag{3.8}$$

This in turn can be solved using flexible GMRES preconditioned by T_2. We continue

recursively, solving equations of the form

$$S_i z = w, \quad z, w \in \mathbb{C}^{mn/2^i} \tag{3.9}$$

iteratively, preconditioning by solving equations of the form

$$T_i y = v, \quad y, v \in \mathbb{C}^{mn/2^i}, \tag{3.10}$$

until the matrix T_i is small enough that T_i^{-1} can be applied directly by means of LU factorization at low cost. Therefore, at level i, each GMRES iteration requires a preconditioning step that in turn calls for iterative solution by GMRES of a *coarser* level equation. At the coarsest level the preconditioner is applied directly using an LU factorization of T_{i+1}. This process is summarized in Algorithm 3.1.

Algorithm 3.1 *Approximate solution of* $T_i y = v$.

(1) *Compute* $\tilde{v}_1 = Q_{i+1} v$, $\tilde{v}_2 = P_{i+1} v$.
(2) *Solve* $\bar{A}_{i+1} \tilde{w}_1 = \tilde{v}_1$ *for* $\tilde{w}_1$.
(3) *Set* $\tilde{w}_2 = \tilde{v}_2 - \bar{C}_{i+1} \tilde{w}_1$.
(4) *Define* $S_{i+1} = T_{i+1} - \bar{C}_{i+1} \bar{A}_{i+1}^{-1} \bar{B}_{i+1}$.
(5) *Solve* $S_{i+1} \tilde{y}_2 = \tilde{w}_2$ *for* $\tilde{y}_2$ *by flexible GMRES iteration, preconditioning with* T_{i+1}, *using Algorithm 3.1 if* $i+1 \leq k$ *and using matrices* L_{i+1}, U_{i+1} *otherwise.*
(6) *Set* $\tilde{y}_1 = \tilde{w}_1 - \bar{A}_{i+1}^{-1} \bar{B}_{i+1} \tilde{y}_2$.
(7) *Set* $y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} Q_{i+1}^T \tilde{y}_1 \\ P_{i+1}^T \tilde{y}_2 \end{pmatrix}$.

To solve equation (2.1), we start the solution process for level $i = 0$ and apply a GMRES iteration with the preconditioner T_1 to the Schur complement of the transformed $T_0 = A$. The overall method is presented in Algorithm 3.2.

Algorithm 3.2 *Solution of* $Ax = b$ *by recursively preconditioned flexible GMRES.*

(1) **Set up**
  (a) *Input matrix A, vector b, tolerance t.*
  (b) *Decide on values for:*
    - *maximum wavelet level, k,*
    - *tolerance* t_i *for inner iterations,*
    - *block band width for approximating the submatrices.*
  (c) *Set* $T_0 = A$ *and* $i = 0$.
  (d) *Recursively, for* $i = 1 \ldots k+1$, *compute* T_i, $\bar{A}_i$, $\bar{B}_i$, $\bar{C}_i$, *and factorize* $\bar{A}_i$.
  (e) *Factorize* T_{k+1} *into* L_{k+1}, U_{k+1}.
(2) **Solve** $T_0 x = b$ *by flexible GMRES preconditioned using Algorithm 3.1.*

Note that the relatively expensive step of computing the BBNS-form matrices $\bar{A}_i, \bar{B}_i, \bar{C}_i, T_i$ is done only once.

4 Numerical results

Here we illustrate the effectiveness of our method, and compare it with some alternative approaches, by considering two example $mn \times mn$ matrices:

$$A_{ni+j,nk+l} = \begin{cases} c & i = k \text{ and } j = l, \\ \frac{1}{2}\log((i-k)^2 + (j-l)^2) & \text{otherwise,} \end{cases} \tag{4.1}$$

for $i, k = 0, 1, \ldots, m-1$; $j, l = 0, 1, \ldots, n-1$; c a constant.

$$B_{ni+j,nk+l} = e^{-\left((i-k)^2+(j-l)^2\right)}, \tag{4.2}$$

for $i, k = 0, 1, \ldots, m-1$; $j, l = 0, 1, \ldots, n-1$.

Tables 1 and 2 give typical results for the matrices A and B respectively. The cost of reducing the relative residual norm to a tolerance of 10^{-6} is shown for matrices of various sizes using the following preconditioners:

P1 simple band preconditioner,
P2 standard BBDWT + band cut preconditioner,
P3 BBDWTPer + band cut preconditioner,
P4 recursive BBDWT-based preconditioner.

In each case GMRES was restarted after 10 iterations. '$*$' denotes non-convergence of GMRES(10). Unpreconditioned GMRES(10) failed to converge to the required tolerance for any size of matrix, so it is omitted from the tables.

m	n	$N = mn$	Preconditioned GMRES								Direct solution
			P1		P2		P3		P4		
			its.	Mflops	its.	Mflops	its.	Mflops	its.	Mflops	Mflops
8	8	64	30	0.65	49	1.2	38	0.99	6	0.32	0.21
16	16	256	58	17	*	*	*	*	7	5.5	12
32	32	1024	86	393	*	*	*	*	7	150	720
64	64	4096	*	*	*	*	*	*	7	6300	46000

TAB. 1. Cost of solving $Ax = b$.

m	n	$N = mn$	Preconditioned GMRES								Direct solution
			P1		P2		P3		P4		
			its.	Mflops	its.	Mflops	its.	Mflops	its.	Mflops	Mflops
8	8	64	8	0.19	8	0.26	8	0.26	4	0.26	0.21
16	16	256	62	19	66	21	63	21	6	5.6	12
32	32	1024	67	310	76	380	74	370	6	120	720
64	64	4096	69	5000	78	6000	78	6000	6	1700	46000

TAB. 2. Cost of solving $Bx = b$.

Clearly the recursive BBDWT approach gives better performance than the alternat-

ive preconditioners that we tested and offers substantial savings compared with direct solution.

5 Conclusion and future work

We have designed a preconditioning method that exploits smoothness between the blocks of a class of dense matrices giving useful savings compared with both direct solution and preconditioned GMRES using band preconditioners. In the future we plan to explore a number of ways of further improving our methods including: (a) using a block DWT, in addition to the BBDWT, to exploit smoothness within each block; (b) using biorthogonal wavelets or multiwavelets (particularly the new supercompact Haar multiwavelets presented in [2]) to give improved compression; (c) preprocessing the matrix to enhance smoothness.

Bibliography

1. O. Axelsson. *Iterative solution methods.* Cambridge University Press, Cambridge, UK, 1996.
2. R. M. Beam and R. F. Warming. Multiresolution analysis and supercompact multiwavelets. *SIAM J. Sci. Comput.*, 22:1238–1268, 2000.
3. G. Beylkin, R. R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Comm. Pure Appl. Math.*, XLIV:141–183, 1991.
4. T. F. Chan and K. Chen. Two-stage preconditioners using wavelet band splitting and sparse approximation. Report CAM 00-26, UCLA, 2000.
5. K. Chen. On a class of preconditioning methods for dense linear systems from boundary elements. *SIAM J. Sci. Comput.*, 20:684–698, 1998.
6. K. Chen. Discrete wavelet transforms accelerated sparse preconditioners for dense boundary element systems. *Electron. Trans. Numer. Anal.*, 8:138–153, 1999.
7. J. Ford and K. Chen. An algorithm for accelerated computation of DWTPer-based band preconditioners. *Num. Alg.*, 26(2):167–172, 2001.
8. J. Ford and K. Chen. Wavelet-based preconditioners for dense matrices with non-smooth local features. *BIT*, 41(2):282–307, 2001.
9. J. Ford, K. Chen, and D. Evans. On a recursive Schur preconditioner for iterative solution of a class of dense matrix problems. *Int. J. Comput. Math.*, 79: to appear.
10. J. Ford, K. Chen, and L. Scales. A new wavelet transform preconditioner for iterative solution of elastohydrodynamic lubrication problems. *Int. J. Comput. Math.*, 75:497–513, 2000.
11. D. Gines, G. Beylkin, and J. Dunn. LU factorization of non-standard forms and direct multiresolution solvers. *Appl. Comput. Harmon. Anal.*, 5:156–201, 1998.
12. Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, 14(2):461–469, 1993.
13. Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS, Boston, 1996.
14. G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesley-Cambridge Press, USA, 1996.

Some properties of the perturbed Haar wavelets

A. L. González

Departmento de Matemática, Universidad Nacional de Mar del Plata, Funes 3350, 7600 Mar del Plata, Argentina.
algonzal@mdp.edu.ar

R. A. Zalik

Department of Mathematics, Auburn University, AL 36849-5310.
zalik@auburn.edu

Abstract

One of the authors has studied the properties of a family of Riesz bases obtained by perturbing the Haar function using B–splines. Although these bases cannot be obtained by multiresolution analyses, they have other interesting properties. The present paper discusses how a discrete signal $\{a_r; 0 \leq r \leq N-1\}$ can be studied by considering a suitable function of the form $f(t) := \sum_{r=0}^{N-1} a_r f_r(t)$, so that the existing theory for functions defined over a continuous domain can be applied.

1 Introduction

In what follows $\mathbf{Z}$ will denote the integers and $\mathbb{R}$ the real numbers; t and x will always denote real variables. The support of a function f will be denoted by $\text{supp}\,(f)$, its quadratic norm by $\|f\|$ and if $f \in L(\mathbb{R})$ its Fourier transform is defined by

$$\widehat{f}(x) := \int_{\mathbb{R}} e^{-txi} f(t)\, dt.$$

In [3] we found a family of affine wavelet Riesz bases of $L^2(\mathbb{R})$, of bounded support and arbitrary degrees of smoothness, obtained by smoothing the discontinuities of the Haar function using B–splines. Although these bases are not orthogonal they are symmetric, a feature that is lacking in orthogonal wavelets. Our bases can be constructed so that the difference between the frame bounds (which are given explicitly) can be made as small as desired. In general, orthogonal wavelets are represented by infinite series, and for computational purposes values are generated over a discrete set using the cascade algorithm [2, 5]. Our bases, on the other hand, are given in closed form. We now briefly describe how these wavelets are defined and introduce additional notation and make assumptions that will be used in the subsequent discussion.

Let $N_m(t)$ denote the B–spline of order m ($m \geq 2$) ([1], Chapter 4), $\chi_{[0,m-1]}(t)$ the

characteristic function of $[0, m-1]$),

$$g(t) := \chi_{[0,m-1]}(t) \sum_{k=0}^{m-2} N_m(t-k), \quad g_1(t) := g(t-m+1), \quad h(t) := (1/2) \sum_{k=0}^{m-2} N_m(t-k),$$

and $q(t) := g_1(t) - h(t)$. For $0 < \delta < 1/2$, let $\alpha_1 = -\alpha_2 = -\alpha_3 = \alpha_4 = 2(m-1)/\delta$, $\beta_1 = 2(m-1)$, $\beta_2 = 2(m-1)(1+\delta)/\delta$, $\beta_3 = -\beta_4 = (m-1)/\delta$,

$$p^{\{i\}}(t) := (-1)^{i-1} q(\alpha_i t + \beta_i), i = 1, 2, 3, 4, \quad p^{\{5\}}(t) := -(\chi_{[1/2-\delta,1/2)}(t) - \chi_{[1/2,1/2+\delta)}(t)),$$

$$p^{\{6\}}(t) := \chi_{[0,1/2)}(t) - \chi_{[1/2,1)}(t), \quad \text{and} \quad \psi(t) := \sum_{i=1}^{6} p^{\{i\}}(t).$$

We will call ψ *the perturbed Haar wavelet.* In [3] we proved that supp $(\psi) \subseteq [-\delta, 1+\delta]$, $\psi \in C^{m-2}(\mathbb{R})$, and that if $\psi_{j,k}(t) := 2^{j/2}\psi(2^j t - k)$, then $\{\psi_{j,k}; j, k \in \mathbf{Z}\}$ is a Riesz basis, and we provided explicit upper and lower frame bounds. Moreover, in [7] we showed that given a function μ, the wavelet coefficients $\langle \mu, \psi_{j,k} \rangle$ can be computed in $O(N)$ steps (where N is the sample size), just as in the orthogonal case.

In this paper we will discuss the application of the perturbed Haar wavelet to the study of discrete signals. Let us first look at the orthogonal case for comparison.

Let μ be an orthogonal wavelet associated with a multiresolution analysis $\{V_j; j \in \mathbf{Z}\}$ and a scaling function ϕ, with the caveat that the definition of multiresolution analysis that we are adopting is that of [1] and [4], and therefore $V_j \subset V_{j+1}, j \in \mathbf{Z}$, whether other authors, like [2] and [5] assume that $V_{j+1} \subset V_j$. If $\mathbf{a} := \{a_r; 0 \leq r \leq N-1\}$ is an arbitrary sequence of real or complex numbers, then this discrete signal is transformed into a continuous one by considering the function $\nu(t) := \sum_{r=0}^{N-1} a_r \phi(t-r)$.

The study of the signal $\nu(t)$ has two stages: the *analysis* stage consists in computing the wavelet coefficients,whereas the *synthesis* stage consists in reconstructing the signal from the wavelet coefficients. If W_j denotes the closure of the linear span of the functions $\mu_{j,k}$, $j \in \mathbf{Z}$, then the W_j are mutually orthogonal and $V_0 = \oplus_{j<0} W_j$. Since $\nu \in V_0$, it turns out that the wavelet coefficients $\langle \nu, \mu_{j,k} \rangle$ vanish for $j > 0$. Moreover, since $\nu(t)$ has compact support, for each $j \leq 0$ there is only a finite number of nonzero wavelet coefficients.

With the perturbed Haar wavelet we face an additional problem: the spaces W_j are no longer orthogonal, and we can therefore no longer assume that all the wavelet coefficients corresponding to positive values of j must vanish. Moreover, we may not even have a scaling function: in [8] we showed that if $\delta = 2^\ell$, where ℓ is a negative integer, then the perturbed Haar wavelet ψ that corresponds to this value of δ cannot be generated by a multiresolution analysis.

To overcome these difficulties, we proceed as follows. Let $n \in \mathbf{Z}$ be such that $2^n \geq 4(m-1)$, $b^{\{1\}}(t) := \chi_{[0,2(m-1))}(t) q(t)$, $b^{\{2\}}(t) := q(4(m-1)-t)$, $b(t) := b^{\{1\}}(t) + b^{\{2\}}(t)$, $f_r(t) := a_r b(2^n t - 4(m-1)r)$, and $f(t) := \sum_{r=0}^{N-1} f_r(t)$. By a direct application of [3] Lemma 6 we obtain the following

Lemma 1 *The function $b(t)$ has the following properties:*

(a) *supp*$(b) \subseteq [0, 4(m-1)]$, (b) $b \in C^{m-2}(\mathbb{R})$, (c) $b(2(m-1)) = 1$,

(d) $\frac{d^k}{dx^k}b(0) = \frac{d^k}{dx^k}b(2(m-1)) = \frac{d^k}{dx^k}b(4(m-1)) = 0,\ 1 \le k \le m-2,$
(e) *The total variation of* b *does not exceed* $4(m-1)$, (f) $|b(t)| \le 1$.

From the preceding lemma we conclude that $\text{supp}(f) \subseteq [0,1]$, and that the functions f_r have disjoint supports. This implies that $\|f\|^2 = \|b\|^2\|\mathbf{a}\|^2 2^{-n}$, where $\|\mathbf{a}\|^2 := \sum_{r=0}^{N-1}|a_r|^2$. We will also use the ℓ_1 norm: $\|\mathbf{a}\|_1 := \sum_{r=0}^{N-1}|a_r|$. Note, moreover, that $f \in C^{m-2}(\mathbb{R})$, and that $f(2^{1-n}(m-1)(2r+1)) = f_r(2^{1-n}(m-1)(2r+1)) = a_r b(2(m-1)) = a_r$.

In theory, given all its wavelet coefficients, the function f can be reconstructed using the frame algorithm or other, even faster, algorithms [5]. However, since there may be an infinite number of nonzero wavelet coefficients, the application of such algorithms may not always be practical. We will adopt an approximation approach. If $A = A(\delta, m)$, and $B = B(\delta, m)$ are respectively the lower and upper frame bounds of the Riesz basis generated by ψ, $h_{j,k} := \langle f, \psi_{j,k}\rangle$, and $Lf := \sum_{j,k,\in\mathbf{Z}} h_{j,k}\psi_{j,k}$, then from the error estimates for the frame algorithm we know that $\|Lf - f\| \le ((B-A)/(B+A))\|f\|$. Since, as remarked above, we can make A and B as close to 1 as we want by making δ sufficiently small, we conclude that for every $\varepsilon > 0$ there is a δ_0 such that if $0 < \delta < \delta_0$, then $\|Lf - f\| < \varepsilon\|f\|$. To approximate f using the wavelet coefficients it will therefore suffice to approximate Lf by an operator of the form

$$Ef = \sum_{j=j_1}^{j_2}\sum_{k\in\mathbf{Z}} h_{j,k}\psi_{j,k}.$$

Observe that since f has bounded support, Ef reduces to a finite sum.

Our objective will be accomplished by showing that there is a constant K such that

$$\left\|\sum_{k\in\mathbf{Z}} h_{j,k}\psi_{j,k}\right\| \le K\|\mathbf{a}\|\,2^{-|j|/2}.$$

But first we need to prove five lemmas, of some independent interest. We begin with

Lemma 2 *Let* $\{a_h; k \in \mathbf{Z}\}$ *and* $\{b_h; k \in \mathbf{Z}\}$ *be increasing sequences such that* $a_k < b_{k-1} < a_{k+1}$, $k \in \mathbf{Z}$. *Assume that* $f_k \in L^2(\mathbb{R})$ *and that supp*$(f_k) \subseteq [a_k, b_k]$, *and let* $f := \sum_{k\in\mathbf{Z}} f_k$. *Then* $\|f\|^2 \le 2\sum_{k\in\mathbf{Z}}\|f_k\|^2$.

Proof: If $r < k-1$ then $b_r \le b_{k-2} < a_k$, whereas if $r > k+1$ then $a_r \ge a_{k+2} > b_k$. This implies that if $r \ne k-1, k$ then $f_r(t) = 0$ on $[a_k, b_k]$, and we readily see that

$$\|f\|^2 \le 2\sum_{k\in\mathbf{Z}}\int_{a_k}^{b_k}|f_k(t)|^2 = 2\sum_{k\in\mathbf{Z}}\|f_k(t)\|^2. \qquad \square$$

Lemma 3 *Let* $u \in L^2(\mathbb{R})$ *be a function with support in an interval* $[a,b]$ *with* $b-a \le 1$. *If* $j \le 0$, *then*

$$\sum_{k\in\mathbf{Z}}|\langle u, \psi_{j,k}\rangle|^2 \le 3\|u\|^2 2^j.$$

Proof: Let $j \le 0$ be arbitrary but fixed, and define $I(k) := \text{supp}\,(\psi_{j,k}) \cap [a,b]$. Then $I(k) \subseteq [2^{-j}(k-\delta), 2^{-j}(k+\delta+1)] \cap [a,b]$. If $I(k) = \emptyset$ then, either $2^{-j}(k+\delta+1) \le a$,

or $2^{-j}(k \dot{-} \delta) \geq b$. This implies that if $I(k) \neq \emptyset$, then $k \in (2^j a - \delta - 1, 2^j b + \delta)$. Since the length of this interval is less than 3, we conclude that there are at most three values of k for which $I(k) \neq \emptyset$. In other words, there are at most three values of k for which $h_{j,k} \neq 0$. Since $|\psi(t)| \leq 1$, for any such k we have:

$$\begin{aligned} |\langle u, \psi_{j,k}\rangle|^2 &= 2^j \left| \int_{I(k)} u(t)\psi(2^k t - k) \right|^2 dt \leq 2^j \int_{I(k)} |u(t)|^2\, dt \int_{I(k)} |\psi(2^k t - k)|^2\, dt \\ &\leq (b-a)2^j \int_{I(k)} |u(t)|^2\, dt \leq 2^j \|u\|^2. \qquad \square \end{aligned}$$

Lemma 4 *Let $\alpha, \beta, \gamma, \sigma \in \mathbb{R}$, with $\alpha, \gamma \neq 0$, and define $c(t) := q(\alpha t + \beta)$, $d(t) := q(\gamma t + \sigma)$, and*

$$K = 2\left\{ \left[25/64 + (25/192)^{2/3} \right] (m-1)^4 + (m-1)^2/1024 \right\}.$$

If $j > 0$ and $i = 5, 6$, then

(a) $\displaystyle\sum_{k \in \mathbf{Z}} |\langle d, c_{j,k}\rangle|^2 \leq 2\left(4\sqrt{K}\,\alpha^{-2} + 1/3\right)^2 2^{-j}$; (b) $\displaystyle\sum_{k \in \mathbf{Z}} \left|\langle d, p^{(i)}_{j,k}\rangle\right|^2 \leq \left(2\sqrt{2} + 1/2\right)^2 2^{-j}$.

Proof: (a) From [3] p. 3367 (bearing in mind the slightly different definition of the Fourier transform), we have

$$\widehat{g}(x) = (i/x)e^{-(m-1)xi/2}\left[e^{-(m-1)xi/2} - ((2/x)\sin x/2)^{m-1} \right].$$

From [1] p. 56 (3.2.16),

$$\widehat{N}_m(x) = e^{-(1/2)mxi}[(2/x)\sin x/2]^m. \tag{1.1}$$

Let

$$s(x) := [(2/x)\sin x/2]^{m-1}.$$

Then

$$\widehat{g_1}(x) = e^{-(m-1)xi}\,\widehat{g}(x) = (i/x)[e^{-2(m-1)xi} - e^{-(3/2)(m-1)xi}s(x)].$$

Since

$$\widehat{h}(x) = \frac{1}{2}\sum_{k=0}^{m-2} e^{-kxi}\widehat{N}_m(x) = (1/2)\frac{1 - e^{-(m-1)xi}}{1 - e^{-xi}}\widehat{N}_m(x),$$

a straightforward computation yields

$$\widehat{h}(x) = -i/(2x)[e^{-(1/2)(m-1)xi} - e^{-(3/2)(m-1)xi}]s(x),$$

whence

$$\begin{aligned} \widehat{q}(x) &= i\frac{1}{x}e^{-(m-1)xi}[\cos(m-1)x - s(x)\cos\frac{1}{2}(m-1)x \\ &\quad + i(2s(x)\sin\frac{1}{2}(m-1)x - \sin(m-1)x)]. \end{aligned} \tag{1.2}$$

This implies that

$$|\widehat{q}(x)|^2 \leq 8x^{-2}, \quad x \neq 0. \tag{1.3}$$

On the other hand,

$$\widehat{q}(x) = ix^{-1}e^{-(m-1)xi}\left[(v_1+v_2)+i(v_3+v_4)\right],$$

where

$$v_1 := \cos(m-1)x - \cos(1/2)(m-1)x, \qquad v_2 := [1-s(x)]\cos(1/2)(m-1)x,$$

$$v_3 := s(x)\left[2\sin(1/2)(m-1)x - \sin(m-1)x\right], \qquad v_4 := [s(x)-1]\sin(m-1)x.$$

A McLaurin expansion shows that $|v_1| \le (5/8)(m-1)^2x^2$. Since $1-u^{m-1} = (1-u)\sum_{k=0}^{m-2}u^k$ and $|\sin u| \le |u|$, we infer that

$$|1-s(x)| \le (m-1)|1-(2/x)\sin x/2| = (m-1)(2/x)|x/2-\sin x/2|.$$

Since $|u-\sin u| \le |u|^3/6$, we conclude that $|1-s(x)| \le (m-1)x^2/48$. Thus,

$$|v_2(x)| \le (m-1)x^2/48, \quad \text{and} \quad |v_4(x)| \le (m-1)x^2/48.$$

Another McLaurin expansion yields $|v_3| \le (5/24)(m-1)^3|x|^3$. Clearly $|v_3| \le 3$; thus $|v_3| = |v_3|^{2/3}|v_3|^{1/3} \le (25/192)^{1/3}(m-1)^2x^2$. Since

$$|\widehat{q}(x)|^2 = x^{-2}[(v_1+v^2)^2+(v_3+v_4)^2] \le 2x^{-2}[v_1^2+v_2^2+v_3^2+v_4^2],$$

we deduce that

$$|\widehat{q}(x)|^2 \le Kx^2. \tag{1.4}$$

From Plancherel's identity we have:

$$\begin{aligned}\langle d, c_{j,k}\rangle &= 2^{j/2}\int_{\mathbb{R}} d(t)c(2^jt-k)\,dt = 2^{j/2}/(2\pi)\int_{\mathbb{R}} e^{kxi}\widehat{c}(x)\widehat{d}(2^jx)\,dx \\ &= 2^{j/2}/(2\pi)\int_0^{2\pi} e^{kxi}\sum_{k\in\mathbf{Z}}\widehat{c}(x+2\pi r)\widehat{d}(2^j(x+2\pi r))\,dx.\end{aligned}$$

This means that $\{2^{-j/2}\langle d, c_{j,k}\rangle; k \in \mathbf{Z}\}$ is the sequence of Fourier coefficients of the function $\sum_{k\in\mathbf{Z}}\widehat{c}(x+2\pi r)\widehat{d}(2^j(x+2\pi r))$. Thus, applying Bessel's identity and then the Cauchy–Schwarz inequality twice (once for sums and once for integrals), we have:

$$2\pi 2^{-j}\sum_{k\in\mathbf{Z}}|\langle d, c_{j,k}\rangle|^2 = \int_0^{2\pi}\left|\sum_{r\in\mathbf{Z}}\widehat{c}(x+2\pi r)\widehat{d}(2^j(x+2\pi r))\right|^2 dx$$

$$\le \int_0^{2\pi}\Big[|\widehat{c}(x)\widehat{d}(2^jx)| + |\widehat{c}(x-2\pi)\widehat{d}(2^j(x-2\pi))| + \Big|\sum_{r\ne 0,-1}\widehat{c}(x+2\pi r)\widehat{d}(2^j(x+2\pi r))\Big|\Big]^2 dx$$

$$\le \Big[\Big(\int_0^{2\pi}|\widehat{c}(x)\widehat{d}(2^jx)|^2\,dx\Big)^{1/2} + \Big(\int_0^{2\pi}|\widehat{c}(x-2\pi)\widehat{d}(2^j(x-2\pi))\,|^2\,dx\Big)^{1/2}$$

$$+\Big(\int_0^{2\pi}\Big|\sum_{r\ne 0,-1}\widehat{c}(x+2\pi r)\widehat{d}(2^j(x+2\pi r))\Big|^2\,dx\Big)^{1/2}\Big]^2$$

$$=: \left(\sqrt{S_1}+\sqrt{S_2}+\sqrt{S_3}\right)^2.$$

Since $\widehat{c}(x) = \alpha^{-1}e^{(\beta/\alpha)xi}\widehat{q}(\alpha^{-1}x)$, (1.3) implies that

$$|\widehat{c}(x+2\pi r)|^2 \le 8|x+2\pi r|^{-2}, \quad x \ne 2\pi r, \tag{1.5}$$

whereas from (1.4) we see that

$$|\widehat{c}(x+2\pi r)|^2 \le K\,\alpha^{-4}|x+2\pi r|^2. \tag{1.6}$$

Since $\widehat{d}(x) = \gamma^{-1}e^{(\tau/\gamma)xi}\widehat{q}(\gamma^{-1}x)$, (1.3) also implies that

$$|\widehat{d}(2^j(x+2\pi r))|^2 \le 4^{-j+1}\,2|x+2\pi r|^{-2}, \quad x \ne 2\pi r. \tag{1.7}$$

Since S_1 is obtained by integrating the product of the left–side members of (1.6) and (1.7) (with $r=0$) over an interval of length 2π, we readily see that

$$S_1 \le 16\pi K\,\alpha^{-4}4^{-j}. \tag{1.8}$$

A similar argument yields

$$S_2 \le 16\pi K\,\alpha^{-4}4^{-j}. \tag{1.9}$$

From Minkowski's inequality

$$S_3 \le \int_0^{2\pi} \sum_{r\ne 0,-1} |\widehat{c}(x+2\pi r)|^2 \sum_{r\ne 0,-1} \left|\widehat{d}(2^j(x+2\pi r))\right|^2 dx.$$

If $x \in [0,2\pi]$ and $r \ge 1$ then from (1.5) we have:

$$\sum_{r\ge 1} |\widehat{c}(x+2\pi r)|^2 \le 2\pi^{-2}\sum_{r\ge 1} r^{-2} = 1/3,$$

whereas (1.7) implies that

$$\sum_{r\ge 1} \left|\widehat{d}(2^j(x+2\pi r))\right|^2 \le 2\,4^{-j}\pi^{-2}\sum_{r\ge 1} r^{-2} = 4^{-j}/3.$$

Similarly,

$$\sum_{r\le -2} |\widehat{c}(x+2\pi r)|^2 \le 2\pi^{-2}\sum_{r\ge 1} r^{-2} = 1/3,$$

and

$$\sum_{r\le -2} \left|\widehat{d}(2^j(x+2\pi r))\right|^2 \le 2\,4^{-j}\pi^{-2}\sum_{r\ge 1} r^{-2} = 4^{-j}/3,$$

whence we conclude that $S_3 \le (4\pi/9)4^{-j}$. Combining (1.8), (1.9) and the preceding inequality, the assertion follows.

(b) Note that $\widehat{p^{\{6\}}}$ is $\widehat{p^{\{5\}}}$ with $\delta = 1/2$. Since $\widehat{p^{\{5\}}}(x) = 2ix^{-1}e^{-(1/2)xi}(1-\cos\delta x)$, we see that

$$|\widehat{p^{\{5\}}}(x+2\pi r)|^2 \le 4|x+2\pi r|^{-2}, \quad x \ne 2\pi r. \tag{1.10}$$

On the other hand, the inequality $|1-\cos\delta x| \le (1/2)\delta^2x^2$ implies that $|\widehat{p^{\{5\}}}(x)| \le \delta^2|x|$; therefore

$$|\widehat{p^{\{5\}}}(x+2\pi r)|^2 \le \delta^4|x+2\pi r|^2. \tag{1.11}$$

We now repeat the argument employed in (a), using (1.10) instead of (1.5), (1.11) instead of (1.6), and bearing in mind that $\delta < 1/2$. □

We now find bounds for the quadratic norms of $q(t)$ and $b(t)$.

Lemma 5 (a) $\|\psi\| \leq 1$; (b) $\|b\| \leq 2(m-1)$.

Proof: (a) [1] Theorem 4.3 implies that the functions N_m are nonnegative. This implies that both g and h are nonnegative. In the proof of [3] Lemma 6(f) we show that

$$\int_{\mathbb{R}} g(t)dt = \int_{\mathbb{R}} h(t)\,dt = (m-1)/2,$$

whence

$$\int_{\mathbb{R}} |q(t)|\,dt \leq m-1.$$

Moreover, $|q(t)| \leq 1$ ([3] Lemma 6(h)). Thus,

$$\int_{\mathbb{R}} |q(t)|^2\,dt \leq \int_{\mathbb{R}} |q(t)|\,dt \leq m-1.$$

Therefore,

$$\int_{\mathbb{R}} |p^{\{i\}}(t)|^2\,dt = (\delta/2(m-1)) \int_{\mathbb{R}} |q(t)|^2\,dt \leq \delta/2, \quad i = 1,2,3,4.$$

This implies that

$$\int_{\mathbb{R}} |\psi(t)|^2\,dt \leq 4\delta/2 + \int_{\mathbb{R}} |p^{\{6\}}(t) - p^{\{5\}}(t)|^2\,dt = 2\delta + (1-2\delta) = 1.$$

(b)

$$\int_{\mathbb{R}} |b(t)|^2\,dt \leq \int_{\mathbb{R}} |b(t)|\,dt = 2\int_{\mathbb{R}} |q(t)|\,dt \leq 2(m-1). \qquad \square$$

Theorem 1

(a) *If* $j \leq 0$,

$$\left\| \sum_{k\in\mathbf{Z}} \langle f, \psi_{j,k}\rangle \psi_{j,k} \right\| \leq 2\sqrt{6}(m-1)\|\mathbf{a}\|\, 2^{(j-n)/2}.$$

(b) *Let* K *be defined as in Lemma 4. If* $j > 0$,

$$\left\| \sum_{k\in\mathbf{Z}} \langle f, \psi_{j,k}\rangle \psi_{j,k} \right\| \leq 8\left[\sqrt{2}\left(K2^{1-n} + 1/3\right) + \sqrt{2} + 1/3\right] \|\mathbf{a}\|_1\, 2^{-j/2}.$$

Proof: Assume first that $j \leq 0$. Applying Lemma 2, Lemma 3, and Lemma 5, we have:

$$\left\| \sum_{k\in\mathbf{Z}} \langle f, \psi_{j,k}\rangle \psi_{j,k} \right\|^2 \leq 2\sum_{k\in\mathbf{Z}} \|\langle f, \psi_{j,k}\rangle \psi_{j,k}\|^2 = 2\|\psi\|^2 \sum_{k\in\mathbf{Z}} |\langle f, \psi_{j,k}\rangle|^2$$

$$\leq 2\sum_{k\in\mathbf{Z}} |\langle f, \psi_{j,k}\rangle|^2 \leq 6\|f\|^2\, 2^j \leq 6\|b\|^2\|\mathbf{a}\|^2 2^{j-n} \leq 24(m-1)^2\|\mathbf{a}\|^2 2^{j-n}.$$

Assume now that $j > 0$. Setting $b_r^{\{i\}}(t) := a_r b^{\{i\}}(2^n t - 4(m-1)r)$, we see that $f_r(t) = b_r^{\{1\}}(t) + b_r^{\{2\}}(t)$. Thus,

$$\left\| \sum_{k\in\mathbf{Z}} \langle f, \psi_{j,k}\rangle \psi_{j,k} \right\| \le \sum_{i=1}^{2} \sum_{\ell=1}^{6} \sum_{r=0}^{N-1} \left\| \sum_{k\in\mathbf{Z}} \langle b_r^{\{i\}}, p_{j,k}^{\{\ell\}}\rangle \psi_{j,k} \right\|.$$

Applying Lemma 2 and Lemma 5 as above, we see that

$$\left\| \sum_{k\in\mathbf{Z}} \langle b_r^{\{i\}}, p_{j,k}^{\{\ell\}}\rangle \psi_{j,k} \right\|^2 \le 2 \sum_{k\in\mathbf{Z}} \left| \langle b_r^{\{i\}}, p_{j,k}^{\{\ell\}}\rangle \right|^2.$$

Since the Fourier transforms of $q(t)$ and $\chi_{[0,2(m-1))}q(t)$ are identical. and the functions $b_r^{\{i\}}$ are of the form $a_r\ q(\alpha t + \beta)$ or $a_r\ \chi_{[0,2(m-1))}(\alpha t + \beta) q(\alpha t + \beta)$ with $|\alpha| = 2^n$, from Lemma 4 we have:

$$\sum_{k\in\mathbf{Z}} \left| \langle b_r^{\{i\}}, p_{j,k}^{\{\ell\}}\rangle \right|^2 \le 2|a_r|^2 \left(2\sqrt{K}2^{-n} + 1/3\right)^2 2^{-j}, \quad \ell = 1,2,3,4,$$

and

$$\sum_{k\in\mathbf{Z}} \left| \langle b_r^{\{i\}}. p_{j,k}^{\{\ell\}}\rangle \right|^2 \le |a_r|^2 2 \left(\sqrt{2} + 1/3\right)^2 2^{-j}, \quad \ell = 5,6,$$

whence the assertion readily follows. □

Bibliography

1. C. K. Chui, *An Introduction to Wavelets*, Academic Press, San Diego, 1992.
2. I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992.
3. N. K. Govil and R. A. Zalik, Perturbations of the Haar Wavelet, *Proc. American Math. Soc.* 125 (1997), 3363–3370.
4. E. Hernández and G. Weiss, *A First Course on Wavelets*, CRC Press, Boca Raton, FL, 1996.
5. S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, 1997.
6. R. M. Young, *An Introduction to Nonharmonic Fourier Series*, Academic Press, New York, 1980.
7. R. A. Zalik, A class of quasi–orthogonal wavelet bases, Wavelets, Multiwavelets and their Applications (A. Aldroubi and E. B. Lin, eds.), Contemporary Mathematics, Vol. 216, American Mathematical Society, Providence, RI, 1998, pp. 81–94.
8. R. A. Zalik, Riesz bases and multiresolution analyses, *Appl. Comput. Harm. Analysis* 7 (1999), 315–331.

An example concerning the L_p-stability of piecewise linear B-wavelets

Peeter Oja

Department of Mathematics, Tartu University, Liivi 2, Tartu, Estonia.
Peeter.Oja@ut.ee[1]

Ewald Quak

SINTEF Applied Mathematics, P.O. Box 124 Blindern, 0314 Oslo, Norway.
Ewald.Quak@math.sintef.no [2]

Abstract

In this paper we consider B-wavelets of order 2, i.e. piecewise linear spline prewavelets of smallest support, over nonuniform knot sequences. We discuss an example showing that for $1 < p \leq \infty$, there is no absolute L_p-stability for these B-wavelets. This means that regardless what specific scaling of the B-wavelets is chosen, the corresponding stability constants cannot be made independent of the knot sequences involved.

1 Introduction

Polynomial splines are fundamental tools in numerous branches of applied mathematics, and for spline spaces defined over a given knot sequence, the basis of choice is provided by B-splines, which possess a lot of attractive properties for numerical computations. One of these important properties of B-splines is their absolute stability. Given a B-spline basis $\{B_i\}_{i\in\mathcal{I}}$ of polynomial order d over a valid knot sequence t, a classical result by de Boor [1] states that properly normalized B-splines are stable in the sense that for each set $\{b_i\}_{i\in\mathcal{I}}$ of real coefficients it holds that

$$C_d^{-1} \|\mathbf{b}\|_p \leq \left\| \sum_{i\in\mathcal{I}} b_i \delta_i^{-\frac{1}{p}} B_i \right\|_p \leq \|\mathbf{b}\|_p . \tag{1.1}$$

Here $\|\cdot\|_p$ denotes the standard integral and discrete p-norms for $1 \leq p \leq \infty$, respectively, and the normalizing factor δ_i for each B-spline is the length of its support divided by the order d. The important point is that the positive constant C_d is dependent on the order d alone, and not in any way on the underlying knot sequence t.

Since nested knot sequences give rise to nested spline spaces, spline functions have also become a focus of attention within the theory of wavelets and multiresolution analysis,

[1]Research supported by the Estonian Science Foundation Grant no. 3926.

[2]Research supported by the EU Research and Training Network MINGLE, RTN1-1999-00117.

starting with cardinal spline wavelets on infinite equally spaced and uniformly refined knot sequences, for which Fourier transform techniques are available, see [3] and the references therein.

The study of spline wavelets on bounded intervals, for arbitrary knot sequences and nonuniform refinement began with the papers [4], [5] and [2], respectively. The construction of so-called minimally supported B-wavelets for a given spline order d and two nested knot sequences to provide a basis of the relative orthogonal complement (wavelet) space is described in detail in [6]. This means that given the coarse and fine knot sequence, there exist explicit algorithms to determine the supports of the B-wavelet functions, the so-called minimal intervals, and also to compute the corresponding wavelet functions, though only up to a normalization constant.

One open problem, however, is how to fix the normalization factor for each B-wavelet function to achieve best possible stability for the whole B-wavelet basis. We provide an example for the case of piecewise linear wavelets, i.e. polynomial order 2, that shows that for $1 < p \leq \infty$ there is no absolute stability of B-wavelets, meaning that there is no choice of normalization that provides absolute stability constants which are completely independent of the underlying knot sequences. L_p-stability estimates involving a quantity dependent on the knot sequences for $1 < p \leq \infty$ and showing absolute stability for $p = 1$ are given in [7].

2 Piecewise linear B-wavelets

The theory of B-wavelets [6] covers general cases of knot refinement, such as situations where several or no knots at all are inserted into an old knot interval, or where the multiplicity of an existing knot is increased. For our purposes, however, it is sufficient to consider what one might call the standard setting, where all knots are simple except at the interval endpoints, which we can count as double knots, and where exactly one new knot is inserted strictly between two old ones.

Our notations are as follows for the closed interval $[0, 1]$. We have a coarse knot sequence with $n - 1$ interior knots, namely

$$\tau : 0 = \tau_0 < \tau_1 < \cdots < \tau_n = 1.$$

Strictly between each pair of coarse knots τ_{i-1} and τ_i we insert a new knot s_i at an arbitrary location, i.e.

$$\tau_{i-1} < s_i < \tau_i \text{ for each } i = 1, \ldots, n.$$

Thus we have a sequence s of new knots

$$s : 0 < s_1 < \cdots < s_n < 1.$$

The fine knot sequence $t = \tau \cup s$, when ordered appropriately, is given as

$$t : 0 = t_0 < t_1 < \cdots < t_{2n} = 1,$$

where the even numbered knots in t correspond to old knots in τ, while the odd numbered knots represent the newly inserted knots from s. To account for the boundary, we treat the interval endpoints as double knots by setting $\tau_{-1} = t_{-1} = 0$ and $\tau_{n+1} = t_{2n+1} = 1$.

For our investigations it is necessary to introduce also some notation related to the knot spacings. We set

$$d_i = t_{i+1} - t_i \text{ for } i = 0, \ldots, 2n-1, \text{ and } \delta_i = t_{i+1} - t_{i-1} \text{ for } i = 0, \ldots, 2n,$$

which means $\delta_0 = d_0 = t_1 - t_0$ and $\delta_{2n} = d_{2n-1} = t_{2n} - t_{2n-1}$ at the boundary. Thus δ_i is the distance between two consecutive old knots if i is odd, and between two consecutive new knots if i is even (and not at the boundary).

We also introduce the index sets

$$\Omega = \{1, 3, \ldots, 2n-1\} \text{ and } \Omega_0 = \{3, 5, \ldots, 2n-3\}.$$

The piecewise linear functions on the knot sequences $\tau \subset t$ form nested linear spaces $V_0 \subset V_1$ of dimensions $n+1$ and $2n+1$, respectively. The corresponding *piecewise linear B-splines* forming a basis of these spaces are simple hat functions. We denote them as φ_j and γ_i for τ and t, respectively, where with the necessary adjustments at the endpoints,

$$\varphi_j(x) = \begin{cases} (x - \tau_{j-1})/\delta_{2j-1} & \text{if } x \in [\tau_{j-1}, \tau_j] \\ (\tau_{j+1} - x)/\delta_{2j+1} & \text{if } x \in [\tau_j, \tau_{j+1}] \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } j = 0, \ldots, n, \tag{2.1}$$

$$\gamma_i(x) = \begin{cases} (x - t_{i-1})/d_{i-1} & \text{if } x \in [t_{i-1}, t_i] \\ (t_{i+1} - x)/d_i & \text{if } x \in [t_i, t_{i+1}] \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i = 0, \ldots, 2n. \tag{2.2}$$

Using for any two functions $f, g \in V_1$ the standard inner product

$$\langle f, g \rangle = \int_0^1 f(t)\, g(t)\, dt,$$

we can write

$$V_1 = V_0 \oplus W,$$

where W is the relative orthogonal complement of V_0 in V_1, and $\oplus$ denotes orthogonal summation. The dimension of W is n, so that there is a basis function ψ_k for every index $k \in \Omega$, or in other words for each newly inserted knot s_k.

Nonzero functions $\psi_k \in W$ with minimal support are called *B-wavelets*. The general theory for B-wavelets developed in [6] establishes in this special case that there are n different piecewise linear B-wavelets which form a basis of the wavelet space W. Each such B-wavelet is uniquely determined up to a constant multiple. There are two boundary B-wavelets ψ_1 and ψ_{2n-1} and $n-2$ interior B-wavelets ψ_k for $k \in \Omega_0$, which we will consider first. Each interior B-wavelet has support $[t_{k-3}, t_{k+3}]$, so that

$$\psi_k(x) = \sum_{i=k-2}^{k+2} q_i^k \gamma_i(x) \qquad \text{for } x \in [0, 1]$$

with the coefficients determined by $\psi_k \in W$, or in other words

$$\langle \psi_k, \varphi_j \rangle = 0 \qquad \text{for } j = 0, \ldots, n.$$

For the boundary wavelets ψ_1 and ψ_{2n-1} we have to make some minor modifications. Their supports are $[t_0, t_4]$ and $[t_{2n-4}, t_{2n}]$, respectively, so that

$$\psi_1(x) = \sum_{i=0}^{3} q_i^1 \gamma_i(x) \text{ and } \psi_{2n-1}(x) = \sum_{i=2n-3}^{2n} q_i^{2n-1} \gamma_i(x) \text{ for } x \in [0,1].$$

In the paper [7] the values of all B-wavelet coefficients q_i^k are given explicitly in terms of the knot locations for the standard setting described here. In the same paper estimates for the coefficients are used to derive L_p-stability estimates for these B-wavelets.

3 Stability of B-wavelets

Our aim in this paper is to establish

Theorem 3.1 *Given the B-wavelet basis $\{\psi_k\}_{k\in\Omega}$, then for $1 < p \leq \infty$, there are no sets of weights $\alpha_{k,p}$, $k \in \Omega$, such that*

$$K_1 \|\mathbf{c}\|_p \leq \left\| \sum_{k\in\Omega} c_k \alpha_{k,p} \psi_k \right\|_p \leq K_2 \|\mathbf{c}\|_p \tag{3.1}$$

holds for any wavelet coefficients $(c_1, c_3, \ldots, c_{2n-1})$ and with absolute constants $K_1 > 0$ and $K_2 > 0$, which are completely independent from the choice of knot sequences τ and s.

Due to the finite dimension of W, it is clear that stability constants K_1 and K_2 exist, as any two norms on W are equivalent. The pertinent question is how the weights could be chosen to achieve that the constants are actually independent of the dimension, the p-norm and, if possible, the choice of new knots s. We will prove the assertion by assuming that the estimate (3.1) holds with constants independent of the knot sequences. Then the following special case serves as a counterexample to this assertion.

The old knot sequence τ consists of the equally spaced points:

$$\tau_0 = 0, \tau_1 = 1/3, \tau_2 = 2/3, \tau_3 = 1.$$

We want to investigate what happens if two newly inserted points are positioned ever more closely, so we introduce the new knots as

$$s_1 = 1/3 - \varepsilon, s_2 = 1/3 + \eta, s_3 = 5/6, \text{ for } 0 < \varepsilon, \eta < 1/3,$$

in order to find out what happens if both $\varepsilon \to 0^+$ and $\eta \to 0^+$.

Thus the fine knot sequence t is

$$t_0 = 0, t_1 = 1/3 - \varepsilon, t_2 = 1/3, t_3 = 1/3 + \eta, t_4 = 2/3, t_5 = 5/6, t_6 = 1.$$

The fine interval lengths are

$$d_0 = 1/3 - \varepsilon, d_1 = \varepsilon, d_2 = \eta, d_3 = 1/3 - \eta, d_4 = 1/6, d_5 = 1/6,$$

while

$$\delta_1 = \delta_3 = \delta_5 = 1/3, \text{and } \delta_0 = 1/3 - \varepsilon, \delta_2 = \varepsilon + \eta, \delta_4 = 1/2 - \eta, \delta_6 = 1/6.$$

In this setting any wavelet

$$\psi = \sum_{i=0}^{6} q_i \gamma_i \in W$$

must be orthogonal to the coarse hat functions $\varphi_0, \cdots, \varphi_3$. This actually means that the column vector $\mathbf{q}$ of coefficients q_i must satisfy the matrix equation

$$\mathbf{A}\mathbf{q} = \mathbf{0}, \tag{3.2}$$

where the entries of $\mathbf{A}$ are the inner products of the coarse and fine hat functions, i.e.

$$a_{j,i} = \langle \varphi_j, \gamma_i \rangle, \quad \text{for } j = 0, \ldots, 3, \quad i = 0, \ldots, 6.$$

Direct computations using (2.1) and (2.2) yield as the only nonzero entries

$$\begin{aligned}
a_{0,0} &= -\frac{1}{2}\varepsilon^2 - \frac{1}{6}\varepsilon + \frac{1}{9}, \quad a_{0,1} = \frac{1}{6}\varepsilon + \frac{1}{18}, \quad a_{0,2} = \frac{1}{2}\varepsilon^2, \\
a_{1,0} &= \frac{1}{2}\varepsilon^2 - \frac{1}{3}\varepsilon + \frac{1}{18}, \quad a_{1,1} = -\frac{1}{6}\varepsilon + \frac{1}{9}, \\
a_{1,2} &= -\frac{1}{2}\varepsilon^2 + \frac{1}{2}\varepsilon - \frac{1}{2}\eta^2 + \frac{1}{2}\eta, \\
a_{1,3} &= -\frac{1}{6}\eta + \frac{1}{9}, \quad a_{1,4} = \frac{1}{2}\eta^2 - \frac{1}{3}\eta + \frac{1}{18}, \\
a_{2,2} &= \frac{1}{2}\eta^2, \quad a_{2,3} = \frac{1}{6}\eta + \frac{1}{18}, \\
a_{2,4} &= -\frac{1}{2}\eta^2 - \frac{1}{6}\eta + \frac{13}{72}, \\
a_{2,5} &= \frac{1}{12}, \quad a_{2,6} = \frac{1}{72}, \\
a_{3,4} &= \frac{1}{72}, \quad a_{3,5} = \frac{1}{12}, \quad a_{3,6} = \frac{5}{72}.
\end{aligned}$$

We now investigate the B-wavelets ψ_1 and ψ_3 in detail, corresponding to $s_1 = t_1$ and $s_2 = t_3$. Specializing the results from [7] then yields all necessary B-wavelet coefficients for this setting up to a scaling factor. Note, however, that it is straightforward to check that the corresponding coefficient vectors satisfy the matrix equation (3.2).

The coefficients of the boundary wavelet ψ_1 are

$$\begin{aligned}
q_0^1 &= -\frac{3}{1-3\varepsilon}, \\
q_1^1 &= 6 - \frac{9\varepsilon\eta}{\varepsilon + \eta + 6\varepsilon\eta}, \\
q_2^1 &= -\frac{1+3\eta}{\varepsilon + \eta + 6\varepsilon\eta}, \\
q_3^1 &= \frac{9\eta^2}{\varepsilon + \eta + 6\varepsilon\eta},
\end{aligned}$$

while the ones for the interior B-wavelet ψ_3 are

$$\begin{aligned} q_1^3 &= \frac{9\varepsilon^2}{\varepsilon+\eta+6\varepsilon\eta}, \\ q_2^3 &= -\frac{1+3\varepsilon}{\varepsilon+\eta+6\varepsilon\eta}, \\ q_3^3 &= 3+\frac{3(\varepsilon+\eta)}{2(\varepsilon+\eta+6\varepsilon\eta)}+\frac{9(1-2\eta)}{2(5-12\eta)}, \\ q_4^3 &= -\frac{9}{5-12\eta}, \\ q_4^3 &= \frac{3}{2\,(5-12\eta)}. \end{aligned}$$

We first provide estimates for the p-norms of these B-wavelets.

Proposition 3.2 *For small enough ε and η, it holds for $1 < p \leq \infty$ that*

$$\begin{aligned} \|\psi_1\|_p &\geq \frac{16}{45}\left(\frac{1}{2}\right)^{1/p}(\varepsilon+\eta)^{1/p-1}, \\ \|\psi_3\|_p &\geq \frac{16}{45}\left(\frac{1}{2}\right)^{1/p}(\varepsilon+\eta)^{1/p-1}. \end{aligned}$$

Proof: For all $0 < \varepsilon, \eta < 1/3$ we find that

$$\begin{aligned} \left|q_2^1\right| &\geq (\varepsilon+\eta)^{-1} \inf_{0<\varepsilon,\eta<1/3} \frac{(1+3\eta)\,(\varepsilon+\eta)}{\varepsilon+\eta+6\varepsilon\eta} \\ &= \frac{8}{9}\,(\varepsilon+\eta)^{-1} \end{aligned}$$

and, similarly,

$$\left|q_2^3\right| \geq \frac{8}{9}\,(\varepsilon+\eta)^{-1}.$$

Note that instead of $8/9$ we may write $1-\sigma$ for any $\sigma > 0$ if ε and η are small enough or even 1 if $\varepsilon = \eta$.

In the process $\varepsilon, \eta \to 0^+$ all other coefficients q_i^1 and q_i^3 have finite limits. This means that for small enough $\varepsilon + \eta$

$$\begin{aligned} \|\psi_1\|_\infty &= \max\left|q_i^1\right| = \left|q_2^1\right| \geq \frac{8}{9}\,(\varepsilon+\eta)^{-1}, \\ \|\psi_3\|_\infty &= \max\left|q_i^3\right| = \left|q_2^3\right| \geq \frac{8}{9}\,(\varepsilon+\eta)^{-1}. \end{aligned}$$

The absolute stability of piecewise linear B-splines (1.1) yields with $C_2 \geq 5/2$ (see [1]) and $\delta_2 = \varepsilon + \eta$

$$\|\psi_1\|_p = \left\|\sum_{i=0}^{3} q_i^1 \gamma_i\right\|_p \geq \frac{2}{5}\left(\frac{1}{2}\right)^{1/p} \left\|\left(q_0^1\delta_0^{1/p}, q_1^1\delta_1^{1/p}, q_2^1\delta_2^{1/p}, q_3^1\delta_3^{1/p}\right)\right\|_p$$

$$\geq \frac{2}{5}\left(\frac{1}{2}\right)^{1/p} |q_2^1|\, \delta_2^{1/p} \geq \frac{16}{45}\left(\frac{1}{2}\right)^{1/p} (\varepsilon+\eta)^{1/p-1}.$$

Analogously we get

$$\|\psi_3\|_p = \left\|\sum_{i=1}^{5} q_i^3 \gamma_i\right\|_p \geq \frac{2}{5}\left(\frac{1}{2}\right)^{1/p} |q_2^3|\, \delta_2^{1/p} \geq \frac{16}{45}\left(\frac{1}{2}\right)^{1/p} (\varepsilon+\eta)^{1/p-1}$$

to complete the proof. □

Proof of Theorem 3.1: Let us now assume that with some scaling factor B-wavelets are absolutely stable in p-norm for $1 < p \leq \infty$, i.e. there exist weights $\alpha_{k,p}$ so that the inequalities (3.1) hold with constants independent of the specific choice of knot sequences. Choosing in the current setting all coefficients equal to zero except for $c_1 = 1$, the stability inequality (3.1) yields

$$\|\alpha_{1,p}\psi_1\|_p \leq K_2$$

or in other words, using Proposition 3.2

$$|\alpha_{1,p}| \leq \frac{K_2}{\|\psi_1\|_p} \leq \frac{45}{16} 2^{1/p} K_2 (\varepsilon+\eta)^{1-1/p} \tag{3.3}$$

and by a similar argument

$$|\alpha_{3,p}| \leq \frac{K_2}{\|\psi_3\|_p} \leq \frac{45}{16} 2^{1/p} K_2 (\varepsilon+\eta)^{1-1/p}. \tag{3.4}$$

On the other hand, the stability estimate (3.1) yields for arbitrary c_1 and c_3, while setting all other c_k to zero, that

$$\|c_1\alpha_{1,p}\psi_1 + c_3\alpha_{3,p}\psi_3\|_p \geq K_1 \left(|c_1|^p + |c_3|^p\right)^{1/p} \geq K_1 \max\left(|c_1|, |c_3|\right).$$

Let us choose specifically

$$c_1 = \alpha_{3,p} \text{ and } c_3 = -\alpha_{1,p},$$

which results in

$$|\alpha_{1,p}\alpha_{3,p}|\, \|\psi_1 - \psi_3\|_p \geq K_1 \max\left(|\alpha_{1,p}|, |\alpha_{3,p}|\right),$$

leading with (3.3) and (3.4) to

$$(\varepsilon+\eta)^{1-1/p} \|\psi_1 - \psi_3\|_p \geq \left(\frac{1}{2}\right)^{1/p} \frac{16 K_1}{45 K_2}. \tag{3.5}$$

On the other hand we derive from the absolute stability of linear B-splines (1.1) that

$$\begin{aligned} &\|\psi_1 - \psi_3\|_p \\ &= \left\|q_0^1\gamma_0 + \left(q_1^1 - q_1^3\right)\gamma_1 + \left(q_2^1 - q_2^3\right)\gamma_2 + \left(q_3^1 - q_3^3\right)\gamma_3 - q_4^3\gamma_4 - q_5^3\gamma_5\right\|_p \\ &\leq \left\|\left(q_0^1\delta_0^{1/p}, \left(q_1^1 - q_1^3\right)\delta_1^{1/p}, \left(q_2^1 - q_2^3\right)\delta_2^{1/p}, \left(q_3^1 - q_3^3\right)\delta_3^{1/p}, q_4^3\delta_4^{1/p}, q_5^3\delta_5^{1/p}\right)\right\|_p \end{aligned}$$

$$\leq 6\max\left(\left|q_0^1\right|\delta_0^{1/p}, \left|q_1^1-q_1^3\right|\delta_1^{1/p}, \left|q_2^1-q_2^3\right|\delta_2^{1/p}, \left|q_3^1-q_3^3\right|\delta_3^{1/p}, \left|q_4^3\right|\delta_4^{1/p}, \left|q_5^3\right|\delta_5^{1/p}\right).$$

All the terms

$$\left|q_0^1\right|\delta_0^{1/p}, \left|q_1^1-q_1^3\right|\delta_1^{1/p}, \left|q_3^1-q_3^3\right|\delta_3^{1/p}, \left|q_4^3\right|\delta_4^{1/p}, \left|q_5^3\right|\delta_5^{1/p}$$

are in fact bounded from above for $\varepsilon+\eta\to 0^+$, so that the expression

$$(\varepsilon+\eta)^{1-1/p}\left|q_0^1\right|\delta_0^{1/p}$$

and the other such terms tend to zero.

Since $\left|q_2^1-q_2^3\right| = 3\left|\varepsilon-\eta\right|/(\varepsilon+\eta+6\varepsilon\eta)$ we obtain for the only remaining term

$$(\varepsilon+\eta)^{1-1/p}\left|q_2^1-q_2^3\right|\delta_2^{1/p} = \frac{3\left|\varepsilon-\eta\right|}{1+6\varepsilon\eta/(\varepsilon+\eta)} \leq 3\left|\varepsilon-\eta\right|,$$

which goes to zero as well for $\varepsilon+\eta\to 0^+$. As a consequence

$$\lim_{\varepsilon+\eta\to 0^+}(\varepsilon+\eta)^{1-1/p}\left\|\psi_1-\psi_3\right\|_p = 0,$$

which contradicts (3.5). □

Remark 3.3 *Although we have chosen an example with one boundary and one interior B-wavelet, let us remark that the lack of absolute stability is in no way due to a boundary effect. A completely analogous reasoning is possible if one chooses knot sequences with more interior knots and studies the behaviour for two interior B-wavelets once two new knots coalesce. Similarly just two boundary B-wavelets could be used on an even shorter knot sequence, where there are no interior B-wavelets at all.*

Bibliography

1. C. deBoor, The quasi-interpolant as a tool in elementary polynomial spline theory, in *Approximation Theory*, G. G. Lorentz et.al. (eds), Academic Press, 1973, 269–276.
2. M. Buhmann and C. A. Micchelli, Spline prewavelets for non-uniform knots, *Numer. Math.* **61** (1992), 455–474.
3. C. K. Chui, *An Introduction to Wavelets*, Academic Press, 1992.
4. C. K. Chui and E. Quak, Wavelets on a bounded interval, in *Numerical Methods in Approximation Theory, ISNM 105*, D. Braess and L. L. Schumaker (eds), Birkhäuser, 1992, 53–75.
5. T. Lyche and K. Mørken, Spline-wavelets of minimal support, in *Numerical Methods in Approximation Theory, ISNM 105*, D. Braess and L. L. Schumaker (eds), Birkhäuser, 1992, 177–194.
6. T. Lyche, K. Mørken and E. Quak, Theory and algorithms for nonuniform spline wavelets, in *Multivariate Approximation Theory* , N. Dyn, D. Leviatan, D. Levin and A. Pinkus (eds), Cambridge University Press, 2001, 152–187.
7. J. Mikkelsen, P. Oja and E. Quak, L_p-stability of piecewise linear B-wavelets, preprint.

How many holes can locally linearly independent refinable function vectors have?

Gerlind Plonka

Institute of Mathematics, University of Duisburg, Germany
plonka@math.uni-duisburg.de

Abstract

In this paper we consider the support properties of locally linearly independent refinable function vectors $\Phi = (\phi_1, \ldots, \phi_r)^T$. We propose an algorithm for computing the global support of the components of Φ. Further, for $\Phi = (\phi_1, \phi_2)^T$ we investigate the supports, especially the possibility of holes of refinable function vectors if local linear independence is assumed. Finally, we give some necessary conditions for local linear independence in terms of rank conditions for special matrices given by the refinement mask. But we are not able to give a final answer to the question whether a locally linearly independent function vector can have more than one hole.

1 Introduction

Let $\Phi = (\phi_1, \ldots, \phi_r)^T$, $r \in \mathbb{N}$, be a vector of compactly supported continuous functions on $\mathbb{R}$. The function vector Φ is said to be *refinable* if it satisfies a vector refinement equation

$$\Phi(x) = \sum_{k \in \mathbb{Z}} A(k)\, \Phi(2x - k), \qquad x \in \mathbb{R}, \tag{1.1}$$

where $\{A(k)\}$ is a finitely supported sequence of real $(r \times r)$-matrices.

Refinable function vectors play a basic role in the theory of multiwavelets. In the last years the properties of refinable function vectors have been investigated very extensively. In fact, it is possible to characterize properties like approximation order and regularity of Φ and L^2-stability of the basis generated by Φ completely by means of the refinement mask $\{A(k)\}$ [1, 6, 7, 11].

We say that Φ is *L^2-stable* if there are constants $0 < A \le B < \infty$ such that for any sequences $c_1, \ldots, c_r \in l^2(\mathbb{Z})$,

$$A \sum_{\nu=1}^{r} \sum_{k \in \mathbb{Z}} |c_\nu(k)|^2 \le \left\| \sum_{\nu=1}^{r} \sum_{k \in \mathbb{Z}} c_\nu(k)\, \phi_\nu(\cdot - k) \right\|_{L^2}^2 \le B \sum_{\nu=1}^{r} \sum_{k \in \mathbb{Z}} |c_\nu(k)|^2.$$

In some applications one needs not only L^2-stability of the basis generated by Φ but other stronger conditions of linear independence. We say that Φ is *globally linearly independent*

if for any sequences $c_1, \ldots, c_r$ on $\mathbb{Z}$

$$\sum_{\nu=1}^{r} \sum_{k \in \mathbb{Z}} c_\nu(k)\, \phi_\nu(\cdot - k) = 0 \quad \text{on} \quad \mathbb{R}$$

implies that $c_\nu(k) = 0$ for all $\nu = 1, \ldots, r$ and all $k \in \mathbb{Z}$ (see [8, 5]).

The following definition is even more restrictive: A function vector Φ is called to be *linearly independent on a nonempty open subset* G of $\mathbb{R}$ if for any sequences $c_1, \ldots, c_r$ on $\mathbb{Z}$

$$\sum_{\nu=1}^{r} \sum_{k \in \mathbb{Z}} c_\nu(k)\, \phi_\nu(\cdot - k) = 0 \quad \text{on} \quad G$$

implies that $c_\nu(k) = 0$ for all $k \in I_\nu(G)$, $\nu = 1, \ldots, r$, where $I_\nu(G)$ contains all $k \in \mathbb{Z}$ with $\phi_\nu(\cdot - k) \not\equiv 0$ on G. Finally, Φ is called to be *locally linearly independent* if it is linearly independent on any nonempty open subset G of $\mathbb{R}$.

Obviously, local linear independence of Φ implies global linear independence and global linear independence of Φ implies L^2-stability. It has been shown by Sun [12], that for compactly supported, refinable functions ($r = 1$) with dilation factor 2 the notions of local and global linear independence are equivalent. However, this is not longer true for function vectors [4].

For (scalar) refinable functions ϕ, local linear independence implies that ϕ has integer support, i.e., $\operatorname{supp} \phi$ starts and ends with an integer, and $\operatorname{supp} \phi$ does not contain holes, i.e., $\operatorname{supp} \phi$ is an interval.

Now, one can ask, 'is this also true for locally linearly independent refinable function vectors?' Unfortunately this is not the case. In [10] it has been shown that a component of Φ can have a hole. However, it is not clear, whether a refinable, locally linearly independent function vector can also have components with finitely many or even infinitely many holes.

In this paper, we want to investigate support properties of locally linearly independent function vectors and consider the 'hole problem' more closely. In the second section we briefly recall a characterization of local linear independence for function vectors in terms of the mask $\{A(k)\}$. In Section 3, we present an algorithm for computing the starting points and endpoints of the support of the components ϕ_ν of Φ.

In the remaining part of the paper we restrict ourselves to the special case $\Phi = (\phi_1, \phi_2)^T$. We collect some observations on function vectors with holes in Section 4 and show that holes can only occur in special situations. In Section 5 we give necessary conditions for local linear independence in terms of rank conditions for matrices formed by the mask $\{A(k)\}$. In Section 6 we prove that the function vector Φ given in Example 4.1 is continuous and locally linearly independent. Finally, we summarize our findings in the conclusion. However, the question put in the title of this paper cannot be answered completely. We conjecture that it is not possible to have locally linearly independent function vectors with more than one hole.

2 Preliminaries

Let us start with some notations. For a compactly supported, continuous function $\phi : \mathbb{R} \to \mathbb{R}$ let $\operatorname{supp}\phi$ be the closed subset of $\mathbb{R}$, where ϕ does not vanish. Further, let the *global support* $\operatorname{gsupp}\phi$ be the smallest interval containing $\operatorname{supp}\phi$. The function ϕ is said to have a *hole* if there is an interval I which is a subset of $\operatorname{gsupp}\phi$ of Lebesgue measure greater than zero, where ϕ is identically zero. The function vector Φ is said to contain a hole if one of its components has a hole.

For a characterization of locally linearly independent function vectors we briefly recall the result of Goodman, Jia and Zhou [4]. Let Φ satisfy the refinement equation (1.1), where the mask matrices $A(k)$ are zero matrices for $k < 0$ and for $k > N$. Considering the vector

$$\boldsymbol{\Phi}(x) = (\Phi(x+k))_{k=0}^{N-1}$$

of length rN and the $(rN \times rN)$-block matrices

$$\mathcal{A}_0 = (A(2k-l))_{k,l=0}^{N-1}, \qquad \mathcal{A}_1 = (A(2k-l+1))_{k,l=0}^{N-1}, \tag{2.1}$$

the refinement equation can equivalently be written as

$$\boldsymbol{\Phi}(x/2) = \mathcal{A}_0\,\boldsymbol{\Phi}(x) \quad \text{and} \quad \boldsymbol{\Phi}((x+1)/2) = \mathcal{A}_1\,\boldsymbol{\Phi}(x), \qquad x \in [0,1].$$

For $\epsilon_1, \ldots, \epsilon_n \in \{0, 1\}$ it follows that

$$\boldsymbol{\Phi}\left(\frac{\epsilon_1}{2} + \cdots + \frac{\epsilon_n}{2^n} + \frac{x}{2^n}\right) = \mathcal{A}_{\epsilon_1} \cdots \mathcal{A}_{\epsilon_n}\,\boldsymbol{\Phi}(x), \qquad x \in [0,1].$$

Now let v_0 be a right eigenvector of $\mathcal{A}_0$ to the eigenvalue 1. This eigenvector is unique (up to multiplication with a constant) if Φ is L^2-stable (see [3]). Let V be the minimal common invariant subspace of $\{\mathcal{A}_0, \mathcal{A}_1\}$ generated by v_0. Then V contains the vectors $\boldsymbol{\Phi}(x)$, $x \in [0,1)$, since $\boldsymbol{\Phi}(0) = c\,v_0$ with some constant c and each $x \in [0,1)$ can be represented as a limit of a sequence of dyadic numbers $l/2^n$, $l \in \mathbb{Z}$, $n = 1, 2, \ldots$. Further, let $\mathcal{M}$ be an $(rN \times \dim V)$-matrix such that the columns of $\mathcal{M}$ form a basis of V. Then we have from [4]

Theorem 2.1 *Let Φ be a refinable vector of compactly supported, continuous functions satisfying (1.1) with $A(k) = 0$ for $k < 0$ and $k > N$. Then we have*

(1) *Φ is linearly independent on $(0,1)$ if and only if all nonzero rows of $\mathcal{M}$ are linearly independent.*

(2) *Φ is locally linearly independent if and only if for all n with $0 \le n \le 2^{rN}$ and all $\epsilon_1, \ldots, \epsilon_n \in \{0, 1\}$ the nonzero rows of $\mathcal{A}_{\epsilon_n} \ldots \mathcal{A}_{\epsilon_1}\mathcal{M}$ are linearly independent.*

Remark 2.2 *A similar characterization of local linear independence is possible also for L^1-solutions of vector refinement equations (1.1) and even for distributions (see [2, 13]). Some examples of locally linearly independent function vectors can be found in [4, 10].*

3 Global support of Φ

Now we want to give an algorithm for computing the global support of the components of refinable function vectors Φ from the mask. To this end let us assume that the $(r \times r)$-matrices $A(k)$ in (1.1) are of the form $A(k) = (A_{i,j}(k))_{i,j=1}^{r}$. We look for α_ν, $\beta_\nu \in \mathbb{R}$

with $\operatorname{gsupp} \phi_\nu = [\alpha_\nu, \beta_\nu]$. Let for all pairs (i,j), $i,j = 1,\dots r$,

$$\begin{aligned} s_{i,j} &:= \min\{k : A_{i,j}(k) \neq 0\}, \\ g_{i,j} &:= \max\{k : A_{i,j}(k) \neq 0\}. \end{aligned}$$

Observe that $s_{i,j}$, $g_{i,j}$ are integers. The numbers α_ν can be found by the following algorithm.

Algorithm 3.1

Input: $s_{i,j}$, $i,j = 1,\dots,r$.

(1) *Let* $p := (p_1,\dots,p_r)$ *be a vector of length* r.
For ν *from* 1 *to* r *do* $\alpha_\nu := s_{\nu,\nu}$; $p_\nu := \nu$ *enddo.*

(2) *For* ν *from* 1 *to* r *do*
For j *from* 1 *to* r *do*
if $s_{\nu,j} < 2\alpha_\nu - \alpha_j$ *then* $\alpha_\nu := (s_{\nu,j} + \alpha_j)/2$; $p_\nu := j$ *endif*
enddo
enddo.

(3) *Repeat step* (2) *as long as the vector* $p = (p_1,\dots,p_r)$ *changes.*

(4) *Form the* $(r \times r)$*-coefficient matrix* P *with*

$$P_{i,j} = \begin{cases} 1 & \text{if } i = j \text{ and } i = p(i), \\ 2 & \text{if } i = j \text{ and } i \neq p(i), \\ -1 & \text{if } i \neq j \text{ and } j = p(i), \\ 0 & \text{elsewhere,} \end{cases}$$

and the vectors $a := (\alpha_1,\dots,\alpha_r)^T$, $s := (s_{1,p_1},\dots,s_{r,p_r})^T$ *and solve the linear equation system* $Pa = s$.

Output: $a = (\alpha_1,\dots,\alpha_r)^T$.

Analogously we obtain the algorithm for the endpoints β_ν:

Algorithm 3.2

Input: $g_{i,j}$, $i,j = 1,\dots,r$.

(1) *Let* $p := (p_1,\dots,p_r)$ *be a vector of length* r.
For ν *from* 1 *to* r *do* $\beta_\nu := g_{\nu,\nu}$; $p_\nu := \nu$ *enddo.*

(2) *For* ν *from* 1 *to* r *do*
For j *from* 1 *to* r *do*
if $g_{\nu,j} > 2\beta_\nu - \beta_j$ *then* $\beta_\nu := (g_{\nu,j} + \beta_j)/2$; $p_\nu := j$ *endif*
enddo
enddo.

(3) *Repeat step* (2) *as long as the vector* $p = (p_1,\dots,p_r)$ *changes.*

(4) *Form the* $(r \times r)$*-coefficient matrix* P *as defined in Algorithm 3.1, and the vectors* $b := (\beta_1,\dots,\beta_r)^T$, $g := (g_{1,p_1},\dots,g_{r,p_r})^T$ *and solve the linear equation system* $Pb = g$.

Output: $b := (\beta_1,\dots,\beta_r)^T$.

Proof: The refinement equation (1.1) implies for each component ϕ_ν that

$$\phi_\nu(x) = \sum_{k\in\mathbb{Z}} \sum_{j=1}^{r} A_{\nu,j}(k)\, \phi_j(2x-k).$$

In particular, it follows from the local linear independence, that for all k with $A_{\nu,j}(k) \neq 0$,

$$\operatorname{gsupp} \phi_j(2\cdot -k) \subseteq \operatorname{gsupp} \phi_\nu, \qquad \nu, j = 1,\dots,r,$$

that is $[(\alpha_j+k)/2, (\beta_j+k)/2] \subseteq [\alpha_\nu, \beta_\nu]$. Using the numbers $s_{i,j}$ and $g_{i,j}$ defined above, we obtain $(\alpha_j + s_{\nu,j})/2 \geq \alpha_\nu$ and $(\beta_j + g_{\nu,j})/2 \leq \beta_\nu$, or equivalently,

$$2\alpha_\nu - \alpha_j \leq s_{\nu,j} \quad \text{and} \quad 2\beta_\nu - \beta_j \geq g_{\nu,j} \tag{3.1}$$

for all $\nu, j = 1,\dots,r$. In particular, for each fixed ν at least one of the r inequalities in (3.1) for the starting points (and for the endpoints, respectively) must be an equality.

Let us look to the first algorithm computing the starting points, the second works analogously. In the first step of the algorithm we just put $\alpha_\nu := s_{\nu,\nu}$. These $s_{\nu,\nu}$ are upper bounds of the true starting points of ϕ_ν since, for $j=\nu$, (3.1) implies $\alpha_\nu \leq s_{\nu,\nu}$. Hence it is clear that, if $2\alpha_\nu - \alpha_j$ is greater than $s_{\nu,j}$ for a fixed ν and some $j \in \{1,\dots,r\}$, then α_ν must be reduced since α_j is already an upper bound for the starting point of ϕ_j. Putting now $\alpha_\nu := (s_{\nu,j} + \alpha_j)/2$ in step 2, we obtain again an upper bound of α_ν. Repeating the second step of the algorithm we obtain decreasing sequences for α_ν (being dyadic rationals, and) approaching the exact starting values. However, if the exact starting values are not dyadic rationals then they cannot be obtained by a finite number of repetitions of step 2. That's why we consider the vector p which stores for each ν an index $j = p_\nu$ for which the inequality in (3.1) is even an equality. Then step 2 must only be repeated a few times in order to find the correct vector p. Now, we can use the r equalities

$$2\alpha_\nu - \alpha_{p_\nu} = s_{\nu,p_\nu}$$

in order to compute α_ν directly. By a suitable rearranging of the equations one obtains an $(r\times r)$-coefficient matrix

$$P := \begin{pmatrix} P_1 & 0 & 0 & \dots & 0 \\ 0 & P_2 & 0 & & \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & & P_\kappa & 0 \\ & & R & & D \end{pmatrix}, \tag{3.2}$$

where P_l, $l = 1,\dots,\kappa$, are circulant matrices of the form

$$\begin{pmatrix} 2 & -1 & \dots & 0 \\ 0 & 2 & -1 & \\ & & \ddots & -1 \\ -1 & 0 & \dots & 2 \end{pmatrix},$$

D is a diagonal matrix with diagonal elements 1 or 2, and R is a matrix of dimension $\dim D \times (r - \dim D)$, with one nonvanishing entry in each row at most. For example, in the case $p = (1, 2, \ldots, r)$, P is just the $(r \times r)$-identity matrix, i.e., $\dim D = r$ and the matrices P_l and R do not occur in P. For $p = (2, 3, \ldots, r, 1)$ we find $P = P_1$ and D as well as R vanish. If p contains smaller 'cycles' of the form $(p_{n_1}, \ldots, p_{n_\mu})$ with $p_{n_j} = n_{j+1}$, $j = 1, \ldots, \mu - 1$ and $p_{n_\mu} = n_1$, then each cycle corresponds to a circulant matrix P_l in P. Since the circulant matrices P_l are invertible, the equation system is uniquely solvable. □

Example 3.3 Let $r = 4$ and let the values $s_{i,j}$, $i, j = 1, 2, 3, 4$ be given by the matrix

$$(s_{i,j})_{i,j=1}^{4} = \begin{pmatrix} 1 & 2 & 1 & 0 \\ 1 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 \\ 3 & 0 & 1 & 1 \end{pmatrix}.$$

Algorithm 3.1 gives
step 1: $a^T = (\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (1, 1, 1, 1)$ and $p = (1, 2, 3, 4)$
step 2: $a^T = (\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (1/2, 3/4, 3/4, 3/8)$ and $p = (4, 1, 1, 2)$
step 3: one repetition of step 2:
$a^T = (\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (3/16, 19/32, 19/32, 19/64)$ and $p = (4, 1, 1, 2)$
Since p did not change no further repetition of step 2 is necessary.
step 4: We obtain

$$P = \begin{pmatrix} 2 & 0 & 0 & -1 \\ -1 & 2 & 0 & 0 \\ -1 & 0 & 2 & 0 \\ 0 & -1 & 0 & 2 \end{pmatrix}$$

which can be simply changed into a matrix of the form (3.2) by rearranging the equations for the vector $a' = (\alpha_1, \alpha_4, \alpha_2, \alpha_3)^T$. The system $Pa = s$ with $s = (0, 1, 1, 0)^T$ gives $a = (1/7, 4/7, 4/7, 2/7)^T$.

Remark 3.4 *In [10] it has been shown that for locally linearly independent refinable function vectors $\Phi = (\phi_1, \ldots, \phi_r)^T$ the starting points and the endpoints of $\operatorname{gsupp} \phi_\nu$, $\nu = 1, \ldots, r$, are rational numbers of the form $k + c_r$, where $k \in \mathbb{Z}$ and $c_r \in J_r$ with*

$$J_r := \left\{ \frac{k}{(2^l - 1)2^{r-l}} : l = 1, \ldots, r,\ k = 0, \ldots, (2^l - 1)2^{r-l} - 1 \right\}.$$

4 Function vectors with holes

In contrast with the scalar case, where a locally linearly independent refinable function cannot have a hole, for function vectors this need no longer to be true.

Example 4.1 Let $\Phi = (\phi_1, \phi_2)^T$ satisfy

$$\begin{aligned} \Phi(x) &= \begin{pmatrix} 1/9 & 2/9 \\ 1/3 & 1/3 \end{pmatrix} \Phi(2x) + \begin{pmatrix} 1/3 & 1/3 \\ 1 & 0 \end{pmatrix} \Phi(2x-1) + \begin{pmatrix} 2/3 & 0 \\ 1/3 & 0 \end{pmatrix} \Phi(2x-2) \\ &\quad + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \Phi(2x-7). \end{aligned}$$

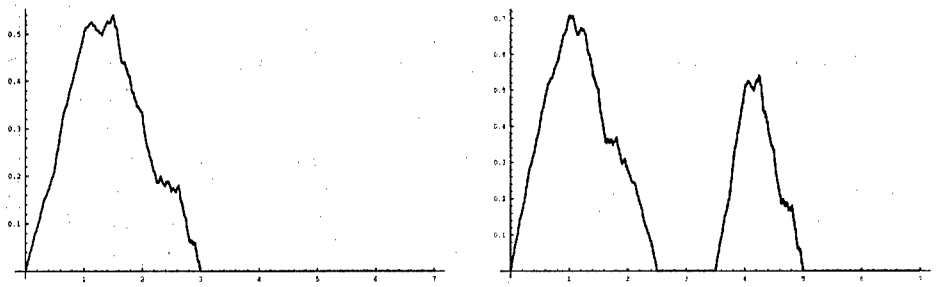


FIG. 1. Locally linearly independent function vector $\Phi = (\phi_1, \phi_2)^T$ with a hole.

Hence $\mathcal{A}_0$ and $\mathcal{A}_1$ in (2.1) are (14×14)-matrices. The function vector Φ is uniquely determined by the refinement equation (up to multiplication by a constant). Further, $\operatorname{gsupp} \phi_1 = [0, 3]$ and $\operatorname{gsupp} \phi_2 = [0, 5]$, and ϕ_2 possesses a hole of length 1, namely $\phi_2(x) = 0$ for $x \in (5/2, 7/2)$ (cf. Figure 1). As we shall show in Section 6, Φ is continuous and locally linearly independent.

Further, one can simply find function vectors Φ with infinitely many holes (but not being locally linearly independent).

Example 4.2 Let $\Phi = (\phi_1, \phi_2)^T$ with

$$\phi_1(x) = \frac{1}{2}\phi_1(2x) + \phi_1(2x-1) + \frac{1}{2}\phi_1(2x-2), \quad \phi_2(x) = \frac{1}{2}\phi_2(2x) + \phi_1(2x-4).$$

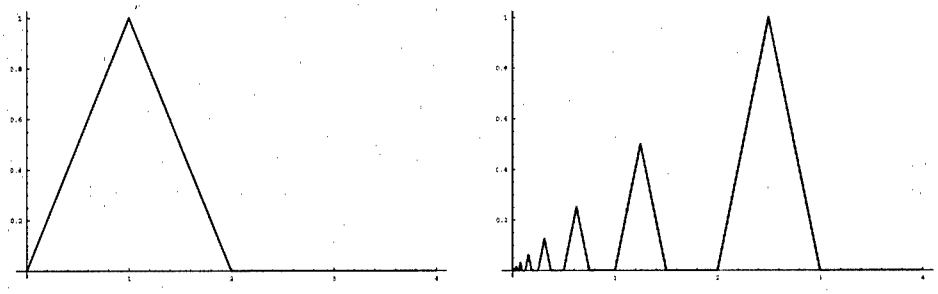


FIG. 2. Function vector $\Phi = (\phi_1, \phi_2)^T$ with infinitely many holes.

Here $\mathcal{A}_0$, $\mathcal{A}_1$ in (2.1) are (8×8)-matrices. Observe that ϕ_1 is just the hat function with $\operatorname{supp} \phi_1 = [0, 2]$ and ϕ_2 is a fractal function with $\operatorname{gsupp} \phi_2 = [0, 3]$, formed by infinitely many 'hats' of support length 2^{-j}, $j = 0, 1, \ldots$, and with infinitely many holes of the form $2^{-j}(3/2, 2)$, $j = 0, 1, \ldots$ (cf. Figure 2). Of course, this function vector is not locally linearly independent, since ϕ_1 is refinable by itself (see also the proof of Theorem 4.3).

We want to consider the support properties of function vectors Φ more closely, and investigate, in which cases the components of Φ can have holes.

In the remaining part of the paper, we only investigate the case $r = 2$, i.e., $\Phi = (\phi_1, \phi_2)^T$.

Theorem 4.3 *Let $\Phi = (\phi_1, \phi_2)^T$ be a refinable, locally linearly independent vector of compactly supported, continuous functions with* $\operatorname{gsupp} \phi_\nu = [\alpha_\nu, \beta_\nu]$ *and let $l_\nu = \beta_\nu - \alpha_\nu$, $\nu = 1, 2$, be the lengths of the global supports with $l_1 \le l_2$. Suppose that Φ contains holes. Then we have*

(1) *The support lengths satisfy $l_2/2 \le l_1 < l_2$.*

(2) *There exist compactly supported, continuous functions f_1, f_2 such that $\phi_2 = f_1 + f_2$ and the vector $(\phi_1, f_1, f_2)^T$ is refinable.*

Proof: Since Φ contains holes, there exists an open interval $I = (\gamma_1, \gamma_2)$ of greatest length and a $\nu \in \{1, 2\}$ with $I \subset \operatorname{gsupp} \phi_\nu$, where ϕ_ν vanishes on I. If there are several intervals of greatest length (biggest holes) we just choose one of them. Refinability implies for $x \in I$

$$\phi_\nu(x) = 0 = \sum_k A_{\nu,1}(k)\, \phi_1(2x - k) + A_{\nu,2}(k)\, \phi_2(2x - k).$$

Since Φ is locally linearly independent, it follows that

$$\begin{aligned} A_{\nu,1}(k) &= 0 \quad \text{for} \quad \operatorname{supp} \phi_1(2 \cdot - k) \cap I \neq \emptyset, \\ A_{\nu,2}(k) &= 0 \quad \text{for} \quad \operatorname{supp} \phi_2(2 \cdot - k) \cap I \neq \emptyset. \end{aligned}$$

The choice of I as the greatest interval now implies that we can replace $\operatorname{supp} \phi_\nu$ by $\operatorname{gsupp} \phi_\nu$, such that

$$\begin{aligned} A_{\nu,1}(k) &= 0 \quad \text{for} \quad 2\gamma_1 - \beta_1 < k < 2\gamma_2 - \alpha_1, \\ A_{\nu,2}(k) &= 0 \quad \text{for} \quad 2\gamma_1 - \beta_2 < k < 2\gamma_2 - \alpha_2. \end{aligned} \tag{4.1}$$

Let now $f_1 := \phi_\nu\, \chi_{[\alpha_\nu, \gamma_1]}$ and $f_2 := \phi_\nu\, \chi_{[\gamma_2, \beta_\nu]}$, where $\chi_{[a,b]}$ denotes the characteristic function of the interval $[a, b]$. Then $\phi_\nu = f_1 + f_2$ and from refinability and from (4.1) it follows that

$$\begin{aligned} f_1(x) &= \sum_{k \le 2\gamma_1 - \beta_1} A_{\nu,1}(k)\, \phi_1(2x - k) + \sum_{k \le 2\gamma_1 - \beta_2} A_{\nu,2}(k)\, \phi_2(2x - k), \\ f_2(x) &= \sum_{k \ge 2\gamma_2 - \alpha_1} A_{\nu,1}(k)\, \phi_1(2x - k) + \sum_{k \ge 2\gamma_2 - \alpha_2} A_{\nu,2}(k)\, \phi_2(2x - k). \end{aligned}$$

If the hole I were in ϕ_1, then at least one of the two functions f_1, f_2 would have a global support length less than $l_1/2$ and hence would vanish since $\operatorname{gsupp} \phi_1(2 \cdot - k)$ and $\operatorname{gsupp} \phi_1(2 \cdot - k)$ have a length $\ge l_1/2$. Thus the hole must be in ϕ_2, i.e., $\phi_2 = f_1 + f_2$.

For $l_1 = l_2$ we obtain a contradiction, since, with the same argument as before, one of the two functions f_1, f_2 vanishes. Hence $l_2 > l_1$. In this case $(\phi_1, f_1, f_2)^T$ is obviously a refinable vector of continuous functions.

It remains to show that $l_2/2 > l_1$ leads to a contradiction. For $l_2/2 > l_1$, ϕ_1 must be refinable by itself, since $\operatorname{gsupp} \phi_2(2 \cdot - k)$ cannot be contained in $\operatorname{gsupp} \phi_1$ for some $k \in \mathbb{Z}$. In particular, from local linear independence we know that then $[\alpha_1, \beta_1]$ is an integer interval and that ϕ_1 has no holes. Further, since at least one of the two functions f_1, f_2

has a global support length less than $l_2/2$, it follows that this function is representable by $\phi_1(2\cdot - k)$, $k \in \mathbb{Z}$, only. Without loss of generality let

$$f_1(x) = \sum_{k \le 2\gamma_1 - \beta_1} A_{2,1}(k)\, \phi_1(2x-k), \qquad x \in \mathbb{R}. \tag{4.2}$$

Considering $\mathbf{\Phi}_1 = (\phi_1(\cdot + k))_{k=\alpha_1}^{\beta_1 - 1}$, local linear independence implies that the space $V_1 = \operatorname{span}\{\mathbf{\Phi}_1(x) : x \in [0,1)\}$ has full dimension l_1. Further, we consider

$$\mathbf{\Phi} = \left(\mathbf{\Phi}_1^T, \left((f_1(\cdot + k))_{k=\lfloor \alpha_2 \rfloor}^{\lceil \gamma_1 \rceil - 1} \right)^T, \left((\phi_2(\cdot + k))_{k=\lceil \gamma_1 \rceil}^{\lceil \beta_2 \rceil - 1} \right)^T \right)^T.$$

(Here, for $x \in \mathbb{R}$, $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x and $\lceil x \rceil$ denotes the smallest integer greater that or equal to x.) Now, choosing a matrix $\mathcal{M}$ of basis vectors of the space $V = \operatorname{span}\{\mathbf{\Phi}(x) : x \in [0,1)\}$, then, because of (4.2), the rows of $\mathcal{M}$ corresponding to f_1 depend on the first l_1 rows (corresponding to ϕ_1). However, not all f_1-rows can be zero rows since f_1 is not a zero function. But this contradicts the local linear independence condition by Theorem 2.1. □

Corollary 4.4 *Let $\Phi = (\phi_1, \phi_2)^T$ be a refinable, locally linearly independent vector of compactly supported, continuous functions with* $\operatorname{gsupp} \phi_\nu = [\alpha_\nu, \beta_\nu]$ *and $l_\nu = \beta_\nu - \alpha_\nu$, $\nu = 1, 2$. Suppose that $l_1 \le l_2$. Then we have: If $l_1 = l_2$ or $l_1 < l_2/2$ then ϕ_1, ϕ_2 do not possess holes.*

Lemma 4.5 *Let $\Phi = (\phi_1, \phi_2)^T$ be a refinable, locally linearly independent vector of compactly supported, continuous functions. Then Φ has no holes that start or end with an integer.*

Proof: Suppose, Φ has a hole which ends with an integer. Choose a hole (γ_1, γ_2) of this type with biggest length. Without loss of generality assume that this hole is in ϕ_2. Then, at least in a small right neighborhood of 0, $\phi_2(\cdot + \gamma_2)$ is representable only by $\phi_1(2\cdot + \alpha_1)$ and $\phi_2(2\cdot + \alpha_2)$. Recall from [10] that the supports $\operatorname{gsupp} \phi_1 = [\alpha_1, \beta_1]$, $\operatorname{gsupp} \phi_2 = [\alpha_2, \beta_2]$ satisfy

$$\alpha_\nu = k + c_2, \quad \beta_\nu = l + c_2, \quad k, l \in \mathbb{Z},\ c_2 \in \{0, 1/2, 1/3, 2/3\}.$$

Now, if both, α_1 and α_2 are integers, then $\phi_1(x + \alpha_1)$, $\phi_2(x + \alpha_2)$, $\phi_2(x + \gamma_2)$ are linearly dependent in some suitable interval $x \in [0, \epsilon)$, $\epsilon > 0$, since they can be represented by the two functions $\phi_1(2x + \alpha_1)$, $\phi_2(2x + \alpha_2)$. This is a contradiction to the local linear independence. If only one α_ν, $\nu \in \{1, 2\}$ is an integer, then $\phi_\nu(x + \alpha_\nu)$ and $\phi_2(x + \gamma_2)$ are representable only by $\phi_\nu(2x + \alpha_\nu)$ in some interval $x \in [0, \epsilon)$ as before and we again obtain a contradiction. If neither α_1 nor α_2 are integers, then $\phi_2(x + \gamma_2)$ cannot be represented by integer translates of $\phi_\nu(2x)$, $\nu = 1, 2$, contradicting the refinability.

Analogously, the contradiction follows for holes starting with an integer. □

Let us call a hole (γ_1, γ_2) in Φ *biggest hole* if there is no other hole in Φ of double size of the form $(2\gamma_1 + k, 2\gamma_2 + k)$ with some $k \in \mathbb{Z}$.

Lemma 4.6 *Let $\Phi = (\phi_1, \phi_2)^T$ be a refinable, locally linearly independent vector of compactly supported, continuous functions. Then there is at most one biggest hole in Φ.*

Proof: Assume that Φ has two biggest holes. Let again l_1, l_2 denote the lengths of the global supports of ϕ_1, ϕ_2 and suppose that $l_1 < l_2$. Then ϕ_1 cannot have a biggest hole by Theorem 4.3. Hence the two holes must be in ϕ_2 and we get a partition $\phi_2 = f_1 + f_2 + f_3$ analogously as in the proof of Theorem 4.3 such that $(\operatorname{gsupp} f_1) \cup (\operatorname{gsupp} f_2) \cup (\operatorname{gsupp} f_3) \subset \operatorname{gsupp} \phi_2$. Further, by refinability, each function f_1, f_2, f_3 can be represented by $\phi_1(2 \cdot -k)$, $\phi_2(2 \cdot -k)$, $k \in \mathbb{Z}$. Moreover, at least one of the three functions f_1, f_2, f_3 must contain a translate of $\phi_2(2\cdot)$, otherwise at least two of the functions f_1, f_2, f_3 would be linearly dependent in a suitable interval inside the starting intervals, since $\phi_1(2 \cdot -k)$ either starts at $\mathbb{Z} + \alpha_1/2$ or at $\mathbb{Z} + (\alpha_1 + 1)/2$ (depending on whether k is even or odd). Hence $\operatorname{gsupp} \phi_2 > (\operatorname{gsupp} \phi_2)/2 + 2(\operatorname{gsupp} \phi_1)/2$. But this contradicts Corollary 4.4. □

Remark 4.7 *All results in this section can be generalized to $r > 2$ and to L^1-integrable functions, if the characterization of local linear independence in [2] is used.*

5 Rank conditions for matrices formed by the refinement mask

We again restrict ourselves to the case that $\Phi = (\phi_1, \phi_2)^T$ is a vector of compactly supported, continuous functions satisfying the refinement equation (1.1) with $A(k) = 0$ for $k < 0$ and $k > N$.

Let us consider the matrices $\mathcal{A}_0$ and $\mathcal{A}_1$ in (2.1) and the minimal common invariant subspace V of $\{\mathcal{A}_0, \mathcal{A}_1\}$ generated by v_0 as defined in Section 2. Recall that V contains $\mathbf{\Phi}(x)$, $x \in [0, 1)$. Let $\mathcal{M}$ be an $(rN \times \dim V)$-matrix such that the columns of $\mathcal{M}$ form a basis of V. Now delete all components in the vector $\mathbf{\Phi} = (\Phi(x+k))_{k=0}^{N-1}$ corresponding to zero rows in $\mathcal{M}$ in order to get $\widetilde{\mathbf{\Phi}}$. Further, delete the corresponding rows and columns in the matrices $\mathcal{A}_0$ and $\mathcal{A}_1$ in (2.1) in order to obtain $\widetilde{\mathcal{A}}_0$ and $\widetilde{\mathcal{A}}_1$ with

$$\widetilde{\mathbf{\Phi}}(x/2) = \widetilde{\mathcal{A}}_0 \widetilde{\mathbf{\Phi}}(x), \qquad \widetilde{\mathbf{\Phi}}((x+1)/2) = \widetilde{\mathcal{A}}_1 \widetilde{\mathbf{\Phi}}(x), \qquad x \in [0, 1]. \tag{5.1}$$

Deleting the zero rows and the corresponding columns in $\mathcal{M}$ we obtain $\widetilde{\mathcal{M}}$.

Example 5.1 Let us consider Example 4.1. Here $\mathbf{\Phi}$ is a vector of length 14 and $V = \operatorname{span}\{\Phi(x+k)_{k=0}^{6} : x \in [0,1)\}$. Since $\operatorname{supp} \phi_1 = [0, 3]$ and $\operatorname{supp} \phi_2 \subset [0, 5]$, it follows that the rows of $\mathcal{M}$ corresponding to $\phi_1(x+j)$, $j = 3, 4, 5, 6$, and $\phi_2(x+j)$, $j = 5, 6$ are zero rows. Indeed, these are all zero rows of $\mathcal{M}$, i.e., V has dimension 8. We delete these components of $\mathbf{\Phi}(x)$ and obtain

$$\widetilde{\mathbf{\Phi}}(x) = (\phi_1(x), \phi_2(x), \phi_1(x+1), \phi_2(x+1), \phi_1(x+2), \phi_2(x+2), \phi_2(x+3), \phi_2(x+4))^T$$

as well as

$$9\widetilde{\mathcal{A}}_0 = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 3 & 3 & 1 & 2 & 0 & 0 \\ 3 & 0 & 9 & 0 & 3 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 3 & 2 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad 9\widetilde{\mathcal{A}}_1 = \begin{pmatrix} 3 & 3 & 1 & 2 & 0 & 0 & 0 & 0 \\ 9 & 0 & 3 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 3 & 3 & 2 & 0 \\ 0 & 0 & 3 & 0 & 9 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 \end{pmatrix}.$$

Let us call a row of $\widetilde{\mathcal{A}}_0$ (resp. $\widetilde{\mathcal{A}}_1$) ϕ_1-row if it corresponds to an ϕ_1-entry in $\widetilde{\mathbf{\Phi}}$ and ϕ_2-row if it correspond to an ϕ_2-entry.

Let n be the length of the new vector $\widetilde{\mathbf{\Phi}}$ and hence $\widetilde{\mathcal{A}}_0$, $\widetilde{\mathcal{A}}_1$ are $(n \times n)$-matrices. If Φ is a locally linearly independent vector then Theorem 2.1 implies that $\widetilde{\mathcal{M}}$ is an invertible $(n \times n)$-matrix.

Deleting the first ϕ_1-row and the first ϕ_2-row and the corresponding columns in $\widetilde{\mathcal{A}}_0$, we obtain a new matrix $\mathcal{B}$ of dimension $(n-2) \times (n-2)$. The same matrix $\mathcal{B}$ is obtained, if we delete the last ϕ_1-row and the last ϕ_2-row and corresponding columns in $\widetilde{\mathcal{A}}_1$. Further, the structure of $\widetilde{\mathcal{A}}_0$, $\widetilde{\mathcal{A}}_1$ implies that

$$\operatorname{spec} \widetilde{\mathcal{A}}_0 = \operatorname{spec} J_0 \cup \operatorname{spec} \mathcal{B}, \qquad \operatorname{spec} \widetilde{\mathcal{A}}_1 = \operatorname{spec} J_1 \cup \operatorname{spec} \mathcal{B},$$

where J_0 (resp. J_1) is a 2×2-matrix containing the entries of $\widetilde{\mathcal{A}}_0$ (resp. $\widetilde{\mathcal{A}}_1$) being at the same time in the first ϕ_1- or ϕ_2-row (resp. last ϕ_1- or ϕ_2-row) and in the first ϕ_1- or ϕ_2-column (resp. last ϕ_1- or ϕ_2-column). (Here spec A denotes the set of eigenvalues of a matrix A.)

Example 5.2 For $\Phi = (\phi_1, \phi_2)^T$ in Example 5.1 we obtain the matrix $\mathcal{B}$ after deleting the first and second row and corresponding columns in $\widetilde{\mathcal{A}}_0$ or by deleting the 5th and 8th row and corresponding columns in $\widetilde{\mathcal{A}}_1$. Hence,

$$\mathcal{B} = \frac{1}{9}\begin{pmatrix} 3 & 3 & 1 & 2 & 0 & 0 \\ 9 & 0 & 3 & 3 & 0 & 0 \\ 0 & 0 & 6 & 0 & 3 & 2 \\ 0 & 0 & 3 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad J_0 = \frac{1}{9}\begin{pmatrix} 1 & 2 \\ 3 & 3 \end{pmatrix}, \quad J_1 = \frac{1}{9}\begin{pmatrix} 0 & 3 \\ 9 & 0 \end{pmatrix}$$

where J_1 and J_2 are invertible.

We obtain

Theorem 5.3 *Let $\Phi = (\phi_1, \phi_2)^T$ be a refinable, locally linearly independent vector of compactly supported, continuous functions. Further, let $\widetilde{\mathcal{A}}_0$, $\widetilde{\mathcal{A}}_1$ and $\mathcal{B}$ be given as above. Then we have*

(1) $\operatorname{rank}(J_0) \geq 1$ *and* $\operatorname{rank}(J_1) \geq 1$,
(2) $\operatorname{rank}(\mathcal{B}) \geq n - 3$,
(3) $\operatorname{rank}(\widetilde{\mathcal{A}}_0) \geq n - 2$ *and* $\operatorname{rank}(\widetilde{\mathcal{A}}_1) \geq n - 2$,
(4) $|\operatorname{rank}(\widetilde{\mathcal{A}}_0) - \operatorname{rank}(\widetilde{\mathcal{A}}_1)| \leq 1$.

Proof: (1) First observe that J_0 and J_1 at least have rank 1, otherwise a component of $\widetilde{\mathbf{\Phi}}(x)$, $x \in [0, 1)$ would completely vanish, contradicting the definition of $\widetilde{\mathbf{\Phi}}$.

Let gsupp $\phi_1 = [\alpha_1, \beta_1]$ and gsupp $\phi_2 = [\alpha_2, \beta_2]$. Then, one simple eigenvalue zero in J_0 implies that $\alpha_1 \in \mathbb{Z}$, $\alpha_2 \in \mathbb{Z} + 1/2$ or vice versa. If J_0 has two eigenvalues 0 then the geometric multiplicity of 0 must be 1 and we obtain $\alpha_1 \in \mathbb{Z} + 1/3$, $\alpha_2 \in \mathbb{Z} + 2/3$ or vice versa. Analogously, a corresponding behavior of J_1 implies $\beta_1 \in \mathbb{Z} + 1/2$, $\beta_2 \in \mathbb{Z}$ or vice versa, and $\beta_1 \in \mathbb{Z} + 2/3$, $\beta_2 \in \mathbb{Z} + 1/3$ or vice versa, respectively.

(2) If the matrix $\mathcal{B}$ possesses the eigenvalue zero, then both, $\widetilde{\mathcal{A}}_0$ and $\widetilde{\mathcal{A}}_1$ possess the eigenvalue zero. Hence, $\widetilde{\mathcal{A}}_0\widetilde{\mathcal{M}}$ and $\widetilde{\mathcal{A}}_1\widetilde{\mathcal{M}}$ are not invertible, while $\widetilde{\mathcal{M}}$ is an invertible matrix. Thus, by Theorem 2.1, $\widetilde{\mathcal{A}}_0$ and $\widetilde{\mathcal{A}}_1$ have a zero row, but being not the first or last ϕ_1- or ϕ_2-row. Hence, also $\mathcal{B}$ has a zero row and, by construction, if $\widetilde{\mathcal{A}}_0$ has the zero row in the l-th ϕ_i-row, $i \in \{1,2\}$, then $\widetilde{\mathcal{A}}_1$ must have a zero row in the $(l-1)$-th ϕ_i-row. This means by (5.1), the two zero rows imply a hole in Φ containing the interval $(k-1/2, k+1/2)$, for some $k \in \mathbb{Z}$. This hole must be a biggest hole. If $\mathcal{B}$ has the eigenvalue zero with geometric multiplicity greater than 1, then with the same arguments one obtains a second biggest hole in Φ. But this contradicts the local linear independence by Lemma 4.6. Hence $\operatorname{rank}(\mathcal{B}) \geq n-3$.

(3) The above considerations directly imply that $\operatorname{rank}(\widetilde{\mathcal{A}}_0) \geq n-2$ and $\operatorname{rank}(\widetilde{\mathcal{A}}_1) \geq n-2$.

(4) Now, if $\widetilde{\mathcal{A}}_0$ has rank $n-2$, then $\mathcal{B}$ has rank $n-3$ and hence $\widetilde{\mathcal{A}}_1$ can have rank $n-1$ at most. Analogously, $\operatorname{rank}(\widetilde{\mathcal{A}}_1) = n-2$ implies $\operatorname{rank}(\widetilde{\mathcal{A}}_0) \leq n-1$. □

From Theorem 5.3 it follows that we have to investigate the following five cases:

(1) $\operatorname{rank}(\widetilde{\mathcal{A}}_0) = \operatorname{rank}(\widetilde{\mathcal{A}}_1) = n$,
(2) $\operatorname{rank}(\widetilde{\mathcal{A}}_0) = \operatorname{rank}(\widetilde{\mathcal{A}}_1) = n-1$,
(3) $\operatorname{rank}(\widetilde{\mathcal{A}}_0) = \operatorname{rank}(\widetilde{\mathcal{A}}_1) = n-2$,
(4) $\operatorname{rank}(\widetilde{\mathcal{A}}_0) = n-1$, $\operatorname{rank}(\widetilde{\mathcal{A}}_1) = n$,
(5) $\operatorname{rank}(\widetilde{\mathcal{A}}_0) = n-1$, $\operatorname{rank}(\widetilde{\mathcal{A}}_1) = n-2$.

All further cases can be reduced to one of the above. However, some of these cases may contradict the local linear independence assumption for Φ.

Considering the first two cases, we obtain a partial answer to the question of whether the support of ϕ_i, $i = 1,2$, can have holes. Moreover, we obtain sufficient conditions for the local linear independence of Φ in terms of rank conditions for $\widetilde{\mathcal{A}}_0$, $\widetilde{\mathcal{A}}_1$.

For the first case we obtain:

Theorem 5.4 *Let $\Phi = (\phi_1, \phi_2)^T$ be a refinable vector of compactly supported, continuous functions. Let the space $\widetilde{V} = \operatorname{span}\{\widetilde{\mathbf{\Phi}}(x) : x \in [0,1)\}$ have full dimension, i.e. $\widetilde{\mathcal{M}}$, formed by basis vectors of $\widetilde{V}$ is an invertible $(n \times n)$-matrix. Let $\widetilde{\mathcal{A}}_0$, $\widetilde{\mathcal{A}}_1$ be given as above. Then $\operatorname{rank}(\widetilde{\mathcal{A}}_0) = \operatorname{rank}(\widetilde{\mathcal{A}}_1) = n$ implies that Φ is locally linearly independent and has no holes.*

Proof: The assertion on local linear independence is already proved in [4], Theorem 3.2. Since $\widetilde{\mathcal{A}}_0$, $\widetilde{\mathcal{A}}_1$ are invertible, the matrix $\widetilde{\mathcal{A}}_{\epsilon_1} \cdots \widetilde{\mathcal{A}}_{\epsilon_n} \widetilde{\mathcal{M}}$ never has a zero row, hence from

$$\widetilde{\mathbf{\Phi}}\left(\frac{\epsilon_1}{2} + \cdots + \frac{\epsilon_n}{2^n} + \frac{x}{2^n}\right) = \widetilde{\mathcal{A}}_{\epsilon_1} \cdots \widetilde{\mathcal{A}}_{\epsilon_n} \widetilde{\mathbf{\Phi}}(x), \qquad x \in [0,1), \tag{5.2}$$

it follows that there is no dyadic interval where ϕ_1 or ϕ_2 vanishes. Thus Φ has no holes. □

For the second case we find

Theorem 5.5 *Let $\Phi = (\phi_1, \phi_2)^T$ be a refinable vector of compactly supported, continuous functions. Let the space $\widetilde{V} = \operatorname{span}\{\widetilde{\mathbf{\Phi}}(x) : x \in [0,1)\}$ have full dimension, i.e. $\widetilde{\mathcal{M}}$,*

formed by basis vectors of $\widetilde{V}$ is an invertible $(n \times n)$-matrix. Let $\widetilde{\mathcal{A}}_0$, $\widetilde{\mathcal{A}}_1$ and $\mathcal{B}$ be given as above. Further, let $\operatorname{rank}(\widetilde{\mathcal{A}}_0) = \operatorname{rank}(\widetilde{\mathcal{A}}_1) = n-1$ *and each of these matrices has one zero row. Then we have*

(1) *If* $\operatorname{rank}(\mathcal{B}) = n-2$ *and the four matrices $\widetilde{\mathcal{A}}_0\widetilde{\mathcal{A}}_0$, $\widetilde{\mathcal{A}}_0\widetilde{\mathcal{A}}_1$, $\widetilde{\mathcal{A}}_1\widetilde{\mathcal{A}}_0$, $\widetilde{\mathcal{A}}_1\widetilde{\mathcal{A}}_1$ have rank $n-1$, then Φ is locally linearly independent and has no holes.*

(2) *If* $\operatorname{rank}(\mathcal{B}) = n-3$ *and the four matrices $\widetilde{\mathcal{A}}_0\widetilde{\mathcal{A}}_0$, $\widetilde{\mathcal{A}}_0\widetilde{\mathcal{A}}_1$, $\widetilde{\mathcal{A}}_1\widetilde{\mathcal{A}}_0$, $\widetilde{\mathcal{A}}_1\widetilde{\mathcal{A}}_1$ have rank $n-1$, then Φ is locally linearly independent and has one hole of the form $(k-1/2,\, k+1/2)$ for some $k \in \mathbb{Z}$.*

Proof: (1) We consider the first case. Since $\operatorname{rank}(\mathcal{B}) = n-2$, it follows that $\mathcal{B}$ is invertible and the zero row of $\widetilde{\mathcal{A}}_0$ must be the first ϕ_1-row or the first ϕ_2-row. Analogously, the zero row of $\widetilde{\mathcal{A}}_1$ must be the last ϕ_1- or ϕ_2-row. Since $\operatorname{rank}(\widetilde{\mathcal{A}}_0\widetilde{\mathcal{A}}_0) = \operatorname{rank}(\widetilde{\mathcal{A}}_1\widetilde{\mathcal{A}}_1) = n-1$, it follows that J_0 and J_1 only have a simple eigenvalue zero and the assumptions (1) of the theorem imply that all matrix products $\widetilde{\mathcal{A}}_{\epsilon_1} \cdots \widetilde{\mathcal{A}}_{\epsilon_n}\widetilde{\mathcal{M}}$, $n \in \mathbb{N}$, have rank $n-1$ and one zero row, namely the same as $\widetilde{\mathcal{A}}_0$ if $\epsilon_1 = 0$ and the same as $\widetilde{\mathcal{A}}_1$ if $\epsilon_1 = 1$. The assumption on $\widetilde{V}$ in the theorem already ensures that Φ is linearly independent on $(0,1)$. Now the above observations also imply that, by Theorem 2.1, Φ is locally linearly independent.

The zero row in $\widetilde{\mathcal{A}}_0$ implies that the support of one component of Φ starts with an integer and the support of the other with a half integer. Considering the zero row in $\widetilde{\mathcal{A}}_1$ we also find that the support of one component ends with an integer and the support of the other with a half integer. In particular, from (5.2) it follows that Φ cannot have holes. (2) We consider the second case. Since $\operatorname{rank}(\mathcal{B}) = n-3$, it follows that $\mathcal{B}$ possesses the eigenvalue zero and the zero rows of $\widetilde{\mathcal{A}}_0$ and $\widetilde{\mathcal{A}}_1$ are not the first or the last ϕ_1- or ϕ_2-rows. Moreover, as shown in the proof of Theorem 5.3, if the l-th ϕ_i-row, $i \in \{1,2\}$, of $\widetilde{\mathcal{A}}_0$ is a zero row then the $(l-1)$-th ϕ_i-row of $\widetilde{\mathcal{A}}_1$ is also a zero row, and this implies by (5.1) a hole of the form $(k-1/2,\, k+1/2)$ for some $k \in \mathbb{Z}$ in ϕ_i. Further, the rank conditions (2) of the theorem imply that all matrix products $\widetilde{\mathcal{A}}_{\epsilon_1} \cdots \widetilde{\mathcal{A}}_{\epsilon_n}\widetilde{\mathcal{M}}$, $n \in \mathbb{N}$, have rank $n-1$ and either a zero row in the l-th or in the $(l-1)$-th row. Thus, by Theorem 2.1, Φ is locally linearly independent and has only one hole. □

Remark 5.6 *Example 4.1 satisfies the assumptions of Theorem 5.5 (2). An example satisfying Theorem 5.5 (1) can be found in [10].*

Observe that the case (2) is not completely settled by Theorem 5.5 since for $\operatorname{rank}(\widetilde{\mathcal{A}}_0) = \operatorname{rank}(\widetilde{\mathcal{A}}_1) = n-1$ some of the four matrices $\widetilde{\mathcal{A}}_0\widetilde{\mathcal{A}}_0$, $\widetilde{\mathcal{A}}_0\widetilde{\mathcal{A}}_1$, $\widetilde{\mathcal{A}}_1\widetilde{\mathcal{A}}_0$, $\widetilde{\mathcal{A}}_1\widetilde{\mathcal{A}}_1$ can also have rank $n-2$. Indeed, there exist locally linearly independent function vectors, where $\operatorname{rank}(\widetilde{\mathcal{A}}_0) = \operatorname{rank}(\widetilde{\mathcal{A}}_1) = n-1$ and $\operatorname{rank}(\widetilde{\mathcal{A}}_0\widetilde{\mathcal{A}}_0) = \operatorname{rank}(\widetilde{\mathcal{A}}_1\widetilde{\mathcal{A}}_1) = n-2$, see [10]. The remaining cases are more complicated to handle and we cannot give a final answer to the question of whether a locally linearly independent refinable vector Φ can have more than one hole.

6 Proof of the example

In this section we want to verify the assertion that the function vector Φ given by the refinement mask in Example 4.1 is continuous and locally linearly independent. Let us

first prove that Φ is continuous. To this end we use the following observation by Jia, Riemenschneider and Zhou [9]:

Let $\{A(k)\}_{k=0}^N$ be a real refinement mask satisfying the following properties:

(1) $\frac{1}{2}\sum_{k=0}^N A(k)$ has one eigenvalue 1 and all further eigenvalues are inside the unit circle.

(2) The matrices $\mathcal{A}_0$ and $\mathcal{A}_1$ both have the simple eigenvalue 1 and there is a vector $e_1 \in \mathbb{R}^{Nr}$ with $e_1^T \mathcal{A}_0 = e_1^T \mathcal{A}_1 = e_1^T$.

(3) Considering the space $U = \{u \in \mathbb{R}^{rN} : e_1^T u = 0\}$ the joint spectral radius of $\mathcal{A}_0|_U$ and $\mathcal{A}_1|_U$ satisfies $\rho(\mathcal{A}_0|_U\, \mathcal{A}_0|_U) < 1$.

Then the subdivision scheme associated with $\{A(k)\}_{k=0}^N$ converges in the maximum norm, and hence the solution vector Φ of the refinement equation is continuous.
Here the joint spectral radius satisfies for any matrix norm

$$\rho(\mathcal{A}_0|_U\, \mathcal{A}_0|_U) = \inf_{n\geq 1}(\max\{\|\mathcal{A}_{\epsilon_1}|_U \cdots \mathcal{A}_{\epsilon_n}|_U\| : \epsilon_i \in \{0,1\},\, i = 1,\ldots,n\})^{1/n}.$$

For our example we find:

1) $\frac{1}{2}\sum_{k=0}^{7} A(k) = \begin{pmatrix} 5/9 & 5/18 \\ 4/3 & 1/6 \end{pmatrix}$ possesses the eigenvalues 1 and $-5/18$.

2) The matrices $\mathcal{A}_0$ and $\mathcal{A}_1$ both have the simple eigenvalue 1 with the left eigenvector $e_1^T = (3,1,3,1,3,1,3,1,3,1,3,1,3,1)$.

3) The space $U = \{u \in \mathbb{R}^{14} : e_1^T u = 0\}$ has dimension 13 and we find the orthonormal basis of U:

$$\begin{aligned}
u_1 &= 28^{-1/2}\,(4,0,0,0,-3,-1,0,0,0,-1,0,0,0,-1)^T,\\
u_2 &= 110^{-1/2}\,(0,0,0,0,-3,-1,0,0,0,0,0,0,0,10)^T,\\
u_3 &= 130^{-1/2}\,(-3,0,0,0,-3,-1,-3,0,0,-1,0,0,10,-1)^T,\\
u_4 &= 132^{-1/2}\,(0,0,0,0,-3,-1,0,0,0,11,0,0,0,-1)^T,\\
u_5 &= 70^{-1/2}\,(-3,0,0,0,-3,-1,7,0,0,-1,0,0,0,-1)^T,\\
u_6 &= 208^{-1/2}\,(-3,0,0,0,-3,-1,-3,0,0,-1,13,0,-3,-1)^T,\\
u_7 &= 3540^{-1/2}\,(-3,-1,-3,0,-3,-1,-3,59,0,-1,-3,-1,-3,-1)^T,\\
u_8 &= 3660^{-1/2}\,(-3,-1,-3,60,-3,-1,-3,-1,0,-1,-3,-1,-3,-1)^T,\\
u_9 &= 2352^{-1/2}\,(-3,48,0,0,-3,-1,-3,0,0,-1,-3,0,-3,-1)^T,\\
u_{10} &= 3422^{-1/2}\,(-3,-1,-3,0,-3,-1,-3,0,0,-1,-3,58,-3,-1)^T,\\
u_{11} &= 4270^{-1/2}\,(-9,-3,-9,-3,-9,-3,-9,-3,61,-3,-9,-3,-9,-3)^T,\\
u_{12} &= 10^{-1/2}\,(0,0,0,0,1,-3,0,0,0,0,0,0,0,0)^T,\\
u_{13} &= 2842^{-1/2}\,(-9,-3,49,0,-9,-3,-9,0,0,-3,-9,0,-9,-3)^T.
\end{aligned}$$

The matrix representations of $\mathcal{A}_0|_U$, $\mathcal{A}_1|_U$ under this basis are $\mathcal{A}_0|_U = ((\mathcal{A}_0\, u_j)^T\, u_k)_{j,k=1}^{13}$ and $\mathcal{A}_1|_U = ((\mathcal{A}_1\, u_j)^T\, u_k)_{j,k=1}^{13}$, and a computation with Maple gives for the spectral norm

$$(\max\{\|\mathcal{A}_{\epsilon_1}|_U\, \mathcal{A}_{\epsilon_2}|_U\, \mathcal{A}_{\epsilon_3}|_U\|_2 : \epsilon_1,\epsilon_2,\epsilon_3 \in \{0,\,1\}\})^{1/3} < 0.95.$$

Hence Φ is continuous.

Let us prove the local linear independence of Φ. Here we use Theorem 2.1 and a procedure proposed by Goodman, Jia and Zhou [4]. The space $V \subset \mathbb{R}^{14}$ (as given in Section 2) is spanned by the vector $v_0 = (0,0,9/5,38/15,6/5,1,0,0,0,9/5,0,0,0,0)^T$ and by A_1v_0, $A_0A_1v_0$, $A_1A_1v_0$, $A_0A_0A_1v_0$, $A_1A_0A_1v_0$, $A_0A_0A_0A_1v_0$, $A_1A_0A_0A_1v_0$. Here v_0 is a right eigenvector of A_0 to the eigenvalue 1. Hence $\dim V = 8$. Forming the matrix $\mathcal{M}$, we observe that the 7-th, the 9-th and the last four rows of $\mathcal{M}$ are zero rows. Hence $\operatorname{gsupp}\phi_1 = [0,3]$ and $\operatorname{gsupp}\phi_2 = [0,5]$. The remaining 8 rows of $\mathcal{M}$ are linearly independent. Thus Φ is linearly independent on $(0,1)$ by Theorem 2.1.

We can restrict our considerations to the shortened matrices $\widetilde{A}_0$, $\widetilde{A}_1$ as given in Example 5.1. Further, we can choose the matrix $\widetilde{\mathcal{M}}$ as the identity matrix. The procedure proposed in [4] gives $\operatorname{rank}\widetilde{A}_0 = \operatorname{rank}\widetilde{A}_0\widetilde{A}_0 = \operatorname{rank}\widetilde{A}_0\widetilde{A}_1 = 7$ and the 7-th rows are zero; $\operatorname{rank}\widetilde{A}_1 = \operatorname{rank}\widetilde{A}_1\widetilde{A}_1 = \operatorname{rank}\widetilde{A}_1\widetilde{A}_0 = 7$ and the 6-th rows are zero.

Hence, Φ is locally linearly independent. Moreover, ϕ_2 possesses a hole of length 1, namely $\phi_2(x) = 0$ for $x \in (5/2,\ 7/2)$.

7 Conclusions

In Section 3 we have presented an algorithm to compute the global supports of the r components of a compactly supported refinable function vector Φ from the refinement mask. The rest of the paper was restricted to $r = 2$.

While for the scalar case local linear independence of a refinable function ϕ guarantees that the support of ϕ is an integer interval without holes, this is not longer the case for $r > 1$. As we have seen in Section 4, a function vector $\Phi = (\phi_1,\ \phi_2)^T$ can only have holes if the lengths l_1 and l_2 of the global supports of ϕ_1, ϕ_2 satisfy $l_2/2 \le l_1 < l_2$. As another property, it has been shown that the endpoints of a hole cannot be integers. Further, Φ can have at most one biggest hole.

In Section 5 we have investigated matrices derived from the refinement mask. In Theorem 5.3 some results on the rank of these matrices are obtained leaving five different cases to be investigated. The first case has been solved completely in Theorem 5.4. The second case has been settled partially in Theorem 5.5. For the other cases we cannot give a final answer. However, if $\widetilde{A}_0$ and $\widetilde{A}_1$ have different rank (as in case (4) and case (5)) then one can show by Theorem 2.1 that Φ must have infinitely many holes. In case (4) this can be seen as follows. Since $\operatorname{rank}(\widetilde{A}_0) = n-1$ it follows that $\operatorname{rank}(\widetilde{A}_1^k\widetilde{A}_0) = n-1$ for $k = 0,1,\ldots$. Hence, by Theorem 2.1, $\widetilde{A}_1^k\widetilde{A}_0$ has a zero row for all $k = 0,1,\ldots$ implying that Φ contains vanishing intervals of the form $(l_k + (2^k-1)/2^k,\ l_k + (2^k - 1/2)/2^k)$ with suitable integers l_k. Here l_k cannot be the same integer for all $k = 0,1,2,\ldots$, in particular one finds $l_k \neq l_{k+1}$, $k \in \mathbb{N}$. Hence Φ has infinitely many holes. This observation leads to the following

Conjecture 7.1 *Let $\Phi = (\phi_1, \phi_2)^T$ be a refinable, locally linearly independent vector of compactly supported, continuous functions. Then Φ cannot have more than one but finitely many holes.*

Our numerical computations however lead to the hypothesis that the cases (3), (4) and (5) contradict the property of local linear independence. So we obtain

Conjecture 7.2 *Let $\Phi = (\phi_1, \phi_2)^T$ be a refinable, locally linearly independent vector of compactly supported, continuous functions. Then Φ cannot have infinitely many holes.*

Acknowledgment The author thanks the referees for their valuable suggestions to improve the paper.

Bibliography

1. C. de Boor, R. A. DeVore, and A. Ron, Approximation orders of FSI spaces in $L_2(\mathbb{R}^d)$, *Constr. Approx.* **14** (1998), 411–427.
2. H. L. Cheung, C. Tang, and D.-X. Zhou, Supports of locally linearly independent M-refinable functions, attractors of iterated function systems and tilings, preprint, 2001.
3. W. Dahmen and C. A. Micchelli, Biorthogonal wavelet expansions, *Constr. Approx.* **13** (1997), 293–328.
4. T. N. T. Goodman, R. Q. Jia, and D.-X. Zhou, Local linear independence of refinable function vectors of functions, *Proc. R. Soc. Edinb.* **130** (2000), 813–826.
5. T. A. Hogan, Stability and independence of the shifts of finitely many refinable functions, *J. Fourier Anal. Appl.* **3** (1997), 757-774.
6. K. Jetter and G. Plonka, A survey on L_2-Approximation order from shift-invariant spaces, in *Multivariate Approximation and Applications*, N. Dyn, D. Leviatan, D. Levin, and A. Pinkus (eds.), Cambridge University Press, 2001, 73–111.
7. R. Q. Jia, Shift-invariant spaces on the real line, *Proc. Amer. Math. Soc.* **125** (1997) 785–793.
8. R. Q. Jia and C. A. Micchelli, On linear independence of integer translates of a finite number of functions, *Proc. Edinburgh Math. Soc.* **36** (1992), 69–85.
9. R. Q. Jia, S. D. Riemenschneider and D.-X. Zhou, Vector subdivision schemes and multiple wavelets, *Math. Comp.* **67** (1998), 1533–1563.
10. G. Plonka and D.-X. Zhou, Properties of locally linearly independent refinable function vectors, preprint, 2001.
11. A. Ron and Z. Shen, The sobolev regularity of refinable functions, *J. Approx. Theory* **106** (2000), 185–225.
12. Q. Y. Sun, Two-scale difference equation: local and global linear independence, manuscript, 1991.
13. J.-Z. Wang, Linear independence relations of the shifts of a vector-valued distribution, manuscript, 2001.

The correlation between the convergence of subdivision processes and solvability of refinement equations

Vladimir Protasov

Department of Mechanics and Mathematics, Moscow State University, Moscow.
protasov@dionis.iasnet.ru

Abstract

We consider the univariate two-scale refinement equation $\varphi(x) = \sum_{k=0}^{N} c_k \varphi(2x-k)$, where $c_0, \cdots, c_N$ are complex values and $\sum c_k = 2$. This paper analyses the correlation between the existence of smooth compactly supported solutions of this equation and the convergence of the corresponding cascade algorithm/subdivision scheme. In the work [11] we have introduced a criterion that expresses this correlation in terms of the mask of the equation. It is shown that the convergence of subdivision scheme depends on values that the mask takes at the points of its *generalized cycles*. In this paper we show that the criterion is sharp in the sense that an arbitrary generalized cycle causes the divergence of a suitable subdivision scheme. To do this we construct a general method to produce divergent subdivision schemes having smooth refinable functions. The criterion therefore establishes a complete classification of divergent subdivision schemes.

1 Introduction

Refinement equations have been studied by many authors in great detail in connection with their role in the theory of wavelets and of subdivision schemes in approximation theory and design of curves and surfaces (see [1–14]). In this paper we study a criterion of convergence of subdivision processes having smooth refinable functions. This criterion was presented in the work [11]. In particular we show that the criterion is sharp in the sense that each if its cases is realized. To do this we provide a general procedure for constructing divergent subdivision schemes (or cascade algorithms) corresponding to smooth refinable functions.

We restrict ourselves to univariate equations with a compactly supported mask. Through the paper we denote by $\mathbb{T} = \mathbb{R}/2\pi\mathbb{Z}$ the unit circle, by $\mathcal{H}$ the space of entire functions on $\mathbb{C}$, by C^l the space of l times continuously differentiable functions on $\mathbb{R}$, by $C^0 = C$ the space of continuous functions, by C_0^l the space of compactly supported functions from C^l, and by C_0 the space of compactly supported continuous functions on $\mathbb{R}$. A sequence $\{f_k\}$ converges to zero in C_0^l if it converges to zero in C^l and the supports of f_k, $k \in \mathbb{N}$ are uniformly bounded.

Consider a refinement equation

$$\varphi(x) = \sum_{k=0}^{N} c_k \varphi(2x-k), \tag{1.1}$$

where $c_k \in \mathbb{C}$, $\sum_k c_k = 2$. The trigonometric polynomial $m(\xi) = \frac{1}{2}\sum_{k=0}^{N} c_k e^{-ik\xi}$ is the *mask* of this equation. It is well known that a C_0-solution of this equation (*refinable function*), if it exists at all, is unique up to normalization and has its support on the segment $[0, N]$. For a given mask m we denote by $[m]$ the corresponding refinement equation. Let us also define the following subspaces of the space C_0:

$$\mathcal{M}^l = \{f \in C_0 \mid \widehat{f}(\xi)(1-e^{-i\xi})^{-l-1} \in \mathcal{H}\}, \qquad \mathcal{L}^l = \{f \in C_0^l \mid \widehat{f^{(l)}} \in \mathcal{M}^l\},\ l \geq 0.$$

In other words the Fourier transform of a function from $\mathcal{M}^l$ has zeros of order $\geq l+1$ at all the points $2\pi k$, $k \in \mathbb{Z}$. The Fourier transform of a function from $\mathcal{L}^l$ has zero at the point $\xi = 0$ and has zeros of order $\geq l+1$ at all the points $2\pi k$, $k \in \mathbb{Z} \setminus \{0\}$. Let us also denote $\mathcal{L} = \mathcal{L}^0 = \mathcal{M}^0$.

The cascade algorithm for refinement equations is the construction of the sequence $f_n = Tf_{n-1}$ for some initial function $f_0 \in C_0$, where $Tf(x) = \sum_k c_k f(2x-k)$ is the *subdivision operator* associated to equation (1.1). This operator is defined on the space C_0 and preserves all the subspaces C^l, $\mathcal{L}^l$. If f_n converges in the space C_0^l to a function $\varphi \in C_0^l$ ($l \geq 0$), then obviously it converges in C_0^l and φ is the solution of (1.1). Moreover, in that case the function $g = f_0 - \varphi$ necessarily belongs to $\mathcal{L}^l$ (see [1], [5]). Thus we say that the cascade algorithm converges in C^l if $T^n g \to 0$, $n \to \infty$ for any $g \in \mathcal{L}^l$. Properties of the cascade algorithms have been studied by many authors in various contexts. This algorithm gives a simple way for approximation of refinable functions and wavelets. On the other hand the convergence of the cascade algorithm is equivalent to the convergence of the corresponding subdivision scheme ([4]). For a given mask $m(\xi)$ we say that the *subdivision process* $\{m\}$ *converges in* C^l if the corresponding cascade algorithm or the corresponding subdivision scheme converges in that space.

It is clear that if a subdivision process converges in C^l, then the corresponding refinement equation has a C_0^l-solution. In general the converse is not true, corresponding examples are well-known (see [1], [2], [13] for general discussions of this aspect). A natural question arises; under which extra conditions the solvability of a refinement equation implies the convergence of the subdivision process?

1) A necessary condition (first introduced in [6]):

If a subdivision process $\{m\}$ *converges in* C^l, *then its mask can be factored as*

$$m(\xi) = \Big(\frac{1+e^{-i\xi}}{2}\Big)^{l+1} a(\xi) \tag{1.2}$$

for some trigonometric polynomial $a(\xi)$. *In particular the condition*

$$m(\xi) = \Big(\frac{1+e^{-i\xi}}{2}\Big) a(\xi) \quad \Leftrightarrow \sum_k c_{2k} = \sum_k c_{2k+1} = 1 \tag{1.3}$$

is necessary for the convergence of the subdivision process in C. Let us remember that for the *existence* of smooth solutions of refinement equation this condition is not necessary (there is a weaker condition for this, see [10]).

For a given mask m denote by $\mathbf{l}(m)$ the maximal integer l such that condition (1.2) is satisfied. So if a subdivision process $\{m\}$ converges in C^k, then $k \leq \mathbf{l}(m)$.

2) A sufficient condition (introduced in [1], developed in [8],[14],[7],[9]):

Suppose a mask m satisfying 1.2 for some $l \geq 0$ has neither symmetric roots nor cycles; then if the equation $[m]$ has a C_0^l-solution, then the process $\{m\}$ converges in C_l.

Let us recall the notation used in this statement. If, for a trigonometric polynomial $p(\xi)$ and for some $\alpha \in \mathbb{T}$, we have $p(\alpha/2) = p(\pi + \alpha/2) = 0$, then $\{\alpha/2, \pi + \alpha/2\}$ is a pair of symmetric roots for $p(\xi)$. In order to be defined we set that for any $\alpha \in \mathbb{T}$ the element $\alpha/2 \in \mathbb{T}$ has the corresponding real value from the half-interval $[0, \pi)$. Further, a given set $\mathbf{b} = \{\beta_1, \cdots, \beta_n\} \subset \mathbb{T}$, where $n \geq 2$, is called cyclic if $2\mathbf{b} = \mathbf{b}$, i.e., $2\beta_j = \beta_{j+1}$ for $j = 1, \cdots, n$ (we set $\beta_{n+1} = \beta_1$). We consider only irreducible cyclic sets, for which all the elements are different. Note that if two cyclic sets do not coincide, then they are disjoint. A cyclic set $\mathbf{b}$ is called a *cycle* of a trigonometric polynomial p if $p(\mathbf{b} + \pi) = 0$, i.e., $p(\beta + \pi) = 0$ for all $\beta \in \mathbf{b}$.

It is well known that the sufficient condition (2) for a mask m is equivalent to the stability of the corresponding refinable function (i.e., integer translates of the refinable function possess Riesz basis property in $L_2(\mathbb{R})$). It is also equivalent to say that the mask satisfies Cohen's criterion (see for example [5, Proposition 2.4]). Actually condition (2) was formulated for the case $l = 0$ only, but it can be easily extended to general l. It is seen, for instance, from Theorem 2.2 of this paper.

Thus we have one necessary and one sufficient condition for the convergence of subdivision processes having smooth refinable functions. It was a natural problem to fill this gap and to elaborate a criterion in terms "if and only if". In 1998 two attempts were made independently from each other and almost simultaneously. They were the work [9] by M. Neamtu and my work [11]. Those two criteria were very similar, but different. Moreover, it turned out that our results were actually incompatible. We will discuss this aspect after formulating the main result of the work [11].

2 A criterion for convergence

We give a criterion of convergence of a subdivision process under the condition that the corresponding refinement equation has a smooth solution. We will see that symmetric roots of mask do not influence the convergence of subdivision processes. This means in particular that the stability of solutions is not necessary for the convergence. The convergence entirely depends on values of the mask at the points of so-called *generalized cycles.*

Everywhere below we consider trigonometric polynomials without positive powers, i.e., polynomials of the form $p(\xi) = \sum_{k=0}^{N} a_k e^{-ik\xi}$. As usual we set $\deg p = N$ (assuming $a_0 a_N \neq 0$). To a given value $\alpha \in \mathbb{T}$ we assign a binary tree denoted in the sequel by $\mathcal{T}_\alpha$. To every vertex of this tree we associate a value from $\mathbb{T}$ as follows: put α at the root, then put $\alpha/2$ and $\pi + \alpha/2$ at the vertices of the first level (the *level* of the vertex is the distance from this vertex to the root. The root has level 0). If a value γ is associated to a vertex on the n-th level, then the values $\gamma/2$ and $\pi + \gamma/2$ are associated to its neighbors on the $(n+1)$-st level. Thus there are the values $\frac{\alpha}{2^n} + \frac{2k\pi}{2^n}$, $k = 0, \cdots, 2^n - 1$ on the n-th level of the tree $\mathcal{T}_\alpha$. A set of vertices $\mathcal{A}$ of the tree $\mathcal{T}_\alpha$ is called a *minimal cut set* if every infinite path (all the paths are without backtracking) starting at the root includes

exactly one element of $\mathcal{A}$. For instance the one-element set $\mathcal{A} = \{root\}$ is a minimal cut set. Every minimal cut set is finite.

Definition 2.1 *A set $\{\beta_1, \cdots, \beta_n\} \subset \mathbb{T}$ is called a generalized cycle of a polynomial $p(\xi)$ if this set is cyclic and for any $j = 1, \cdots, n$ the tree $\mathcal{T}_{\beta_j+\pi}$ possesses a minimal cut set $\mathcal{A}_j$ such that $p(\mathcal{A}_j) = 0$.*

The family $\{\mathcal{A}_1, \cdots, \mathcal{A}_n\}$ is said to be sets of zeros of the generalized cycle $\mathbf{b}$. Let us remark that for a given generalized cycle the set of zeros may not be defined in a unique way. Any (regular) cycle of $p(\xi)$ is also a generalized cycle, in this simplest case each minimal cut set $\mathcal{A}_j$ is the root of the corresponding tree $\mathcal{T}_{\beta_j+\pi}$. On the other hand, not any generalized cycle is a regular cycle. For example, the polynomial $p(\xi) = (e^{-i\xi} - e^{-\frac{\pi i}{3}})(e^{-2i\xi} - e^{\frac{\pi i}{3}})$ has no regular cycles, but is has a generalized cycle $\mathbf{b} = \{\beta_1, \beta_2\} = \{2\pi/3, 4\pi/3\}$. Indeed, this polynomial has three zeros on the period: $\pi/3$, $-\pi/6$, $5\pi/6 \in \mathbb{T}$. The set $\mathcal{A}_1 = \{-\pi/6, 5\pi/6\}$ is a minimal cut set for the point $\beta_1 + \pi$, $\mathcal{A}_2 = \{\pi/3\}$ is a minimal cut set for $\beta_2 + \pi$, and $p(\mathcal{A}_1) = p(\mathcal{A}_2) = 0$. Roughly speaking, each cyclic set $\{\beta_1, \ldots, \beta_n\}$ has a unique corresponding cycle (the family of zeros is $\{\beta_1+\pi, \ldots, \beta_n+\pi\}$) and a variety of generalized cycles (all possible sets of zeros $\{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$, where $\mathcal{A}_j$ is an arbitrary minimal cut set of the tree $\mathcal{T}_{\beta_j+\pi}$, $j = 1, \ldots, n$). Note, that if at least one set $\mathcal{A}_j$ differs from the root $\beta_j + \pi$, then it necessarily contains a pair of symmetric roots of p. Therefore, if the polynomial p has no symmetric roots, then all its generalized cycles, if there are any, are regular cycles.

For any trigonometric polynomial p and any finite subset $Y = \{\alpha_1, \cdots, \alpha_n\} \subset \mathbb{T}$ we denote $\rho_p(Y) = (\prod_{q=1}^n |p(\alpha_q)|)^{1/n}$. This is a multiplicative function on the set of trigonometric polynomials.

Now we formulate the criterion of stability of subdivision process.

Theorem 2.2 *Suppose a refinement equation $[m]$ has a C_0^l-solution for some $l \geq 0$; then the process $\{m\}$ converges in C^l if and only if the mask m satisfies (1.2) and for any generalized cycle $\mathbf{b}$ of the mask m we have $\rho_m(\mathbf{b}) < 2^{-l}$.*

In particular, for $l = 0$, this means that a subdivision process $\{m\}$, whose refinement equation has a continuous solution, converges if and only if $\rho_m(\mathbf{b}) < 1$ for every generalized cycle $\mathbf{b}$ of the mask. Another corollary is Condition (2) from the Section 1. Indeed, if a mask has neither symmetric roots nor cycles, then it has no generalized cycles either. Hence, by Theorem 2.2, the subdivision process must converge.

Example 2.3 Consider a mask

$$m(\xi) = (0.2 + 0.5e^{-i\xi} + 0.3e^{-2i\xi})(e^{-i\xi} - e^{-\frac{\pi i}{3}})^2(e^{-2i\xi} - e^{\frac{\pi i}{3}})^2 \qquad (2.1)$$

The corresponding equation $[m]$ has a C_0-solution, this is shown in Example 4.5. The polynomial m has a unique generalized cycle $\mathbf{b} = \{2\pi/3, 4\pi/3\}$, the same as in the previous example, with the same sets of zeros $\mathcal{A}_1 = \{-\pi/6, 5\pi/6\}$, $\mathcal{A}_2 = \{\pi/3\}$. Actually this is not one, but two coinciding generalized cycles, if we count roots with multiplicity. We have $(\rho_m(\mathbf{b}))^2 =$

$$\left|m(\frac{2\pi}{3})\right| \cdot \left|m(\frac{4\pi}{3})\right| = \left|(-0.2-0.1\sqrt{3}i)\cdot 1\cdot 1\right| \cdot \left|(-0.2+0.1\sqrt{3}i)\cdot 4e^{\frac{4\pi i}{3}}\cdot 4e^{-\frac{4\pi i}{3}}\right| = 1.12 > 1.$$

Hence the subdivision process $\{m\}$ diverges.

3 Statement of the problem

Most examples of divergent subdivision schemes (having smooth refinable functions) are constructed for some special class of masks. These are either "unload" masks of the form $m(\xi) = p(n\xi)$ for some polynomial p and an odd integer n, or, at least, masks whose associated matrix $B = \{c_{2i-j}\}_{i,j\in\{0,\dots,N\}}$ have a multiple eigenvalue 1. The divergence of such schemes is well known and does not require any special criterion. A natural question arises; whether one really needs the criterion of Theorem 2.2 to determine divergent processes? Maybe the family of generalized cycles is too wide to describe unstable subdivision schemes. In general there is no evidence that the condition $\rho_m(\mathbf{b}) > 1$ can be combined with the existence of a smooth solution for the mask m. In this paper we are going to show that Theorem 2.2 indeed characterizes the family of unstable subdivision processes properly. We show that each generalized cycle can cause the divergence of a suitable scheme. On the other hand, we will see that every converging subdivision scheme can be "spoiled" by some generalized cycle.

4 Preliminary results. Reductions of masks

To construct examples of divergent processes we need some auxiliary results. The first of them establishes two properties of cyclic sets. The proof of this lemma is an easy exercise for the reader.

Lemma 4.1 *a) Let* $\mathbf{b}$ *be a cyclic set and* $\alpha \in \mathbb{T}$. *Then for the polynomials* $p_1(\xi) = e^{-i\xi} - e^{-i\alpha}$ *and* $p_2(\xi) = e^{-2i\xi} - e^{-i\alpha}$ *we have* $\rho_{p_1}(\mathbf{b}) = \rho_{p_2}(\mathbf{b})$.
b) Let $\mathbf{b}_1$ *and* $\mathbf{b}_2$ *be cyclic sets and* $p(\xi) = \prod_{\beta\in\mathbf{b}_1}(e^{-i\xi}+e^{-i\beta})$. *Then we have:* $\rho_p(\mathbf{b}_2) = 1$ *if* $\mathbf{b}_1 \neq \mathbf{b}_2$, *and* $\rho_p(\mathbf{b}_2) = 2$ *if* $\mathbf{b}_1 = \mathbf{b}_2$.

Now turn back to the subdivision schemes. For a given integer $l \geq 0$, a mask m, and a function $f \in \mathcal{C}^l$, denote $\nu_l(m, f) = -\lim_{n\to\infty} \log_2 \|T^n[f^{(l)}]\|c/n$, where T is the subdivision operator associated to m (we set $\log_2 0 = -\infty$). The value $\nu_l(m) = \inf_{f\in\mathcal{C}^l} \nu_l(m, f)$ is the *degree of convergence of the process* $\{m\}$ *in the space* $\mathcal{C}^l$.

For every mask m we have $\nu_l(m) \leq l + 1$ (see [3]). Furthermore, it was shown in [3] and [2] that a process $\{m\}$ converges in $\mathcal{C}^l$ if and only if $\nu_l(m) > l$. In particular, the inequality $\nu_0(m) > 0$ means that $\{m\}$ converges in $\mathcal{C}$. Let L be the maximal integer such that $\{m\}$ converges in $\mathcal{C}^L$ (if the process $\{m\}$ does not converge in $\mathcal{C}$, then we nevertheless set $L = 0$). The value $\nu_L(m)$ is said to be the *degree of convergence of the process* $\{m\}$ and denoted in the sequel by $\nu(m)$. If $\nu(m_1) = \nu(m_2)$, then $\nu_l(m_1) = \nu_l(m_2)$ for any $l \geq 0$.

For a given refinement equation $[m]$ denote by $L(m)$ the maximal integer L such that the corresponding refinable function φ belongs to C_0^L. If this equation has no continuous compactly-supported solution, we set $L(m) = -1$. The smoothness of the refinable function φ is the value $s(m) = L + h$, where h is the Holder exponent of the Lth derivative $\varphi^{(L)}$ on $\mathbb{R}$. It is well known that a refinable function belongs to $\mathcal{C}^l$ if and only if $s(m) > l$ (the equality $s(m) = l$ is impossible). In particular, a refinement equation has a C_0-solution if and only if $s(m) > 0$.

Now we can describe the procedure of reduction of subdivision schemes introduced in [11]. This reduction makes it possible to get rid of both symmetric roots and cycles.

4.1 Eliminating of symmetric roots

Let $p(\xi)$ be a given trigonometric polynomial (let us remember that we consider polynomials without positive powers). Assume that p possesses a pair of symmetric roots $\{\alpha/2, \pi+\alpha/2\}$. The transfer from $p(\xi)$ to the polynomial $p_\alpha(\xi) = \frac{p(\xi)(e^{-i\xi}-e^{-i\alpha})}{e^{-2i\xi}-e^{-i\alpha}}$ is said to be a *transfer to the previous level.* The inverse transfer from p_α to p is a *transfer to the next level.* So a transfer to the previous level reduces a pair of symmetric roots $\{\alpha/2, \pi+\alpha/2\}$ to the one root α.

Proposition 4.2 *Let a mask $\tilde{m}$ be obtained from a mask m by a transfer to the previous level. Then $s(\tilde{m}) = s(m)$. Moreover, $\nu(\tilde{m}) = \nu(m)$, whenever $\mathbf{l}(\tilde{m}) = \mathbf{l}(m)$.*

(The constant $\mathbf{l}(m)$ responsible for condition 1.2 was defined in Section 1). This implies, in particular, that the reduced equation $[\tilde{m}]$ possesses a smooth compactly supported solution if and only if the initial equation $[m]$ does; and the same true for the convergence of the corresponding subdivision schemes. Thus, a transfer to the next (previous) level does not change the smoothness of solutions. It also respects the rate of convergence of subdivision processes, unless this transfer does not violate condition 1.2 (a transfer to the previous level may increase the value $\mathbf{l}(m)$). Using this Proposition one can consequently eliminate all symmetric roots of a given mask.

4.2 Elimination of regular cycles

Let a polynomial p possess a cycle $\mathbf{b}$. The transfer from $p(\xi)$ to the polynomial $\tilde{p}(\xi) = p(\xi)/\prod_{\beta\in\mathbf{b}}(e^{-i\xi}+e^{-i\beta})$ is called an *eliminating of a cycle.*

Proposition 4.3 *Let a mask $\tilde{m}$ be obtained from a mask m by eliminating of a cycle $\mathbf{b}$. Then $s(\tilde{m}) = s(m)$ and $\nu(m) = \max\{\nu(\tilde{m}), \rho_m(\mathbf{b})\}$.*

Thus the equation $[m]$ possesses a smooth compactly supported solution if and only if the equation $[\tilde{m}]$ does. Moreover, the process $\{m\}$ converges in C^l if and only if the process $\{\tilde{m}\}$ does and in addition $\rho_m(\mathbf{b}) < 2^{-l}$.

See [11] for the proofs of Propositions 4.2 and 4.3. Now it becomes clear how to establish Theorem 2.2. First we consequently eliminate all symmetric roots. By Proposition 4.2 it does not change neither the smoothness of solution nor the rate of convergence (if the initial mask satisfied condition 1.2). Moreover, by Lemma 4.1 this process respects the constants $\rho_m(\mathbf{b})$ for all cyclic sets $\mathbf{b}$. The final mask has no symmetric roots, hence it can have only regular cycles. Then we eliminate all regular cycles (refereeing to Proposition 4.2) and obtain a mask satisfying Cohen's criterion, whose subdivision process does converge. This line of reasoning also allow us to eliminate directly all generalized cycles as follows.

4.3 Eliminating of generalized cycles

Let a polynomial p possess a generalized cycle $\mathbf{b}$ with corresponding sets of zeros $\mathcal{A}_1, \ldots, \mathcal{A}_n$. The transfer from $p(\xi)$ to the polynomial $\tilde{p}(\xi) = p(\xi)/\prod_{\alpha\in\mathcal{A}_j, j=1,\ldots,n}(e^{-i\xi} - e^{-i\alpha})$ is called an *eliminating of a generalized cycle.*

Proposition 4.4 *Let a mask $\tilde{m}$ be obtained from a mask m by eliminating of a generalized cycle* $\mathbf{b}$. *Then* $s(\tilde{m}) = s(m)$ *and* $\nu(m) = \max\{\nu(\tilde{m}), \rho_m(\mathbf{b})\}$.

Proof: After a suitable sequence of transfers to the previous level all the sets of zeros $\mathcal{A}_1, \dots, \mathcal{A}_n$ drop to the corresponding roots $\beta_1 + \pi, \dots, \beta_n + \pi$, and $\mathbf{b}$ becomes a regular cycle. By Lemma 4.1 this does not change the value $\rho_m(\mathbf{b})$. Now it remains to apply Proposition 4.3. □

Example 4.5 Consider again the mask $m(\xi)$ from Example 2.3. After eliminating the generalized cycle $\mathbf{b} = \{\frac{2\pi}{3}, \frac{4\pi}{3}\}$ we obtain the mask $\tilde{m}(\xi) = 0.2 + 0.5e^{-i\xi} + 0.3e^{-2i\xi}$. Since all the coefficients of $\tilde{m}$ are positive, it follows that the equation $[\tilde{m}]$ has a C_0-solution and, moreover, the corresponding subdivision process $\{\tilde{m}\}$ converges (see, for instance [1]). Now applying Proposition 4.4 we see that the initial process $\{m\}$ diverges, since $\rho_m(\mathbf{b}) = \sqrt{1.12}$. Let us note, that the matrix B corresponding to the mask m ($B = \{c_{2i-j}\}_{i,j\in\{0,\dots,8\}}$) has the eigenvalue 1 with multiplicity one and has no other eigenvalues on the unit circle. So the divergence of the subdivision scheme in this case does not follow from the well-known argument of multiple eigenvalues.

5 Unimprovability of criterion. Examples of divergent schemes

Now we are going to see that Theorem 2.2 gives a full description of divergent subdivision schemes having smooth refinable functions. This means that all possible cases of the criterion of convergence are realized on suitable masks. For the sake of simplicity we formulate this result for the convergence in the space C, i.e., for the case $l = 0$.

Theorem 5.1 *Let* $\mathbf{b} = \{\beta_1, \dots, \beta_n\}$ *be a cyclic set and let* $\mathcal{A}_1, \dots, \mathcal{A}_n$ *be arbitrary minimal cut sets of the trees* $\mathcal{T}_{\beta_1+\pi}, \dots, \mathcal{T}_{\beta_n+\pi}$ *respectively. Then there exists a mask* $m(\xi)$ *such that*

1) $m(\mathcal{A}_j) = 0$, $j = 1, \dots, n$, *i.e.,* $\mathbf{b}$ *is a generalized cycle of the mask* m, *and* A_j *are its sets of zeros;*

2) the equation $[m]$ *has a* C_0*-solution, but the subdivision process* $\{m\}$ *does not converge in* C;

3) after eliminating of the generalized cycle $\mathbf{b}$ *this process becomes converging in* C.

Proof: Consider a mask $p(\xi) = (1 + e^{-i\xi})/2a(\xi)$ such that $\deg a \geq 2$, and the subdivision process $\{p\}$ converges in C. To obtain such a mask it suffices to take an arbitrary polynomial $a(\xi)$ with positive coefficients such that $a(0) = 1$. Now we use the fact that if the process $\{p\}$ converges in C, then it will still converge in this space after all sufficiently small perturbations of the coefficients of $a(\xi)$ preserving the condition $a(0) = 1$ (see [3]). Thus, with possible perturbation of the coefficients, we assume that the trigonometric polynomial a has no real roots and that the value $\rho_a(\mathbf{b})$ is irrational. Such a perturbation exists by the mean value theorem, because $\rho_a(\mathbf{b})$ is a continuous function of the coefficients of $a(\xi)$. This implies, in particular, that $\rho_a(\mathbf{b}) > 0$ and hence $\rho_p(\mathbf{b}) > 0$. Now take the polynomial $q(\xi) = \prod_{\alpha\in\mathcal{A}_j, j=1,\dots,n}(e^{-i\xi} - e^{-i\alpha})$. By Lemma 4.1 we have $\rho_{pq^r}(\mathbf{b}) = 2^r\rho_p(\mathbf{b})$ for every $r \geq 0$. Consequently there exists a nonnegative integer r such that $\rho_{pq^r}(\mathbf{b}) > 1$. Take the smallest such integer r_0 and denote $\tilde{a} = aq^{r_0-1}$ and $\tilde{p} = pq^{r_0-1}$ (if $r_0 = 0$, then we put $\tilde{a} = a, \tilde{p} = p$). Let us remark that the case $\rho_{\tilde{p}}(\mathbf{b}) = 1$ is impossible, because this value is not rational, therefore $\rho_{\tilde{p}}(\mathbf{b}) < 1$. Since $\mathbf{b}$ is the only

generalized cycle of the polynomial $\tilde{p}$, therefore, by Proposition 4.4, the subdivision process $\{\tilde{p}\}$ converges. Now make a small perturbation of the coefficients of the polynomial $\tilde{a}$ after which the process $\{\tilde{p}\}$ still converges, and the value $\rho_{\tilde{p}q}(\mathbf{b})$ is still bigger than 1, but the polynomial $\tilde{a}$ does not have real roots. Then denote $\tilde{m} = \tilde{p}, m = \tilde{m}q$. We see that the mask m has a unique generalized cycle $\mathbf{b}$, and this cycle has sets of zeros $\mathcal{A}_1, \ldots, \mathcal{A}_n$. Since $\rho_m(\mathbf{b}) > 1$, the process $\{m\}$ diverges, however removing this generalized cycle we obtain the converging process $\{\tilde{m}\}$. This proves the theorem. □

Bibliography

1. D. Cavaretta, W. Dahmen, C. Micchelli, *Stationary subdivision,* Mem. Amer. Math. Soc. **93** (1991), 1–186.
2. D. Collela and C. Heil, *Characterization of scaling functions. I. Continuous solutions,* SIAM J. Matrix Anal. Appl. **15** (1994), 496–518.
3. I. Daubechies and J. Lagarias, *Two-scale difference equations. I. Global regularity of solutions,* SIAM. J. Math. Anal. **22** (1991), 1388–1410.
4. I. Daubechies and J. Lagarias, *Two-scale difference equations. II. Local regularity, infinite products of matrices and fractals,* SIAM. J. Math. Anal. **23** (1992), 1031–1079.
5. S. Durand, *Convergence of the cascade algorithms introduced by I. Daubechies*, Numer. Algorithms **4** (1993), 307–322
6. N. Dyn, J. A. Gregory and D. Levin, *Analysis of linear binary subdivision schemes for curve design*, Constr. Approx. **7** (1991), 127–147.
7. L. Herve, *Régularité et conditions de bases de Riesz por les fonctions d'échelle,*, C. R. Acad. Sci., Paris, Ser. I **335** (1992), 1029–1032.
8. R. Q. Jia and J. Wang, *Stability and linear independence associated with wavelet decomposition*, Proc. Amer. Math. Soc. **117** (1993), 1115–1124.
9. M. Neamtu *Convergence of subdivisions versus solvability of refinement equations*, East J. Approx **5**, 1999, 183–210.
10. V. Protasov, *A complete solution characterizing smooth refinable functions*, SIAM J. Math. Anal. **31** (1999), 1332–1350.
11. V. Protasov, *The stability of subdivision operator at its fixed point*, SIAM J. Math. Anal. **33** (2001), 448–460.
12. L. Villemoes, *Wavelet analysis of refinement equations*, SIAM J. Math. Anal. **25** (1994), 1433–1460.
13. Y. Wang, *Two-scale dilation equations and the cascade algorithm*, Random Comput. Dynamic **3** (1995), 289–307.
14. D.-X. Zhou, *Stability of refinable functions, multiresolution analysis, and Haar bases.* SIAM J. Math. Anal. **27** (1996), 891–904.

Accurate approximation of functions with discontinuities, using low order Fourier coefficients

R. K. Wright

Department of Mathematics and Statistics, UVM, Burlington, VT, 05445 USA.
wright@emba.uvm.edu

Abstract

In previous work we introduced a method of using polynomial splines with appropriate discontinuities to approximate a piecewise smooth function f with jump discontinuities of f and f'. The information used is location of discontinuities, and low order, possibly noisy Fourier coefficients. The number of discontinuities was limited to two at most, and the discontinuities needed to lie at meshpoints in a uniform mesh. We showed that the linear operator corresponding to the method is L_2-bounded with a modest bound, and thus that the method is L_2-robust in the presence of noise. In the present paper we develop a new method of analysis which enables us to determine operator bounds that are valid for arbitrarily many discontinuities. The new analysis allows discontinuities to be placed arbitrarily. Given a placement, an initially uniform spline mesh of width h must be used such that nearest meshpoints to discontinuities are at least $4h$ apart (discontinuities then replace these meshpoints); the number of available Fourier coefficients must be at least three times the number of mesh intervals in a period. The previous work was restricted to quadratic splines; the present work includes cubic splines. Much of the analysis uses exact computations with a computer algebra system. We give an example to illustrate the accuracy of the method using noisy Fourier coefficients.

1 Introduction

We consider approximating a function f when the information consists of low order, possibly noisy Fourier coefficients, and knowledge that f is smooth except for jumps of f or f' at known locations but unknown magnitudes. We will work with a method, introduced in [10], which amounts to linear least squares fitting of the available coefficients with the coefficients of splines with appropriately placed discontinuities. Since we anticipate applications to ill-posed problems where boundedness of the solution operator is crucial, we develop a method for bounding the norm of this operator. The bounding method depends heavily on exact computations in certain spline spaces. These computations are fundamentally finite dimensional linear algebra with rational integer coefficients. Their goal is to develop upper bounds for the norms of certain projector operators whose norms are naturally expressed in terms of generalized eigenvalues, and to prove by exact computation that the bounds are correct. A computer algebra system is used for the computations. The programming is detailed in [9].

In [10] we obtained bounds under much more restrictive conditions than in the present paper. In [10] the splines were quadratic only, while here results also are given for cubic splines. The analysis in [10] required all knots of the approximating splines to be uniformly spaced, and since the discontinuities are at the knots, the location of discontinuities was limited. Further, in [10] the estimation process is linear in the total number of discontinuities, and produces results unacceptably large for cases with more than one discontinuity of f and two of f'.

Others ([2, 3, 4, 5]) have addressed questions of accurate approximations to functions with discontinuities given Fourier coefficients as information. In [8] we give examples which show that those methods can substantially magnify noise in the coefficients; our main concern here is to prove robustness of our method. We illustrate with an example in Section 5.

2 General linear space-theoretic results

Let $\mathcal{V}$ be a real Hilbert space with inner product $\langle\ ,\ \rangle$. We will denote the norm associated with $\langle\ ,\ \rangle$ by $||\ ||$. Let $\mathcal{P}$ and $\mathcal{Q}$ be closed subspaces of $\mathcal{V}$; suppose P is the orthogonal projector on $\mathcal{P}$. Here, as in [10], we deal with the approximation f^* obtained as the solution to the constrained least squares problem

$$\min ||Pf^* - Pf||, f^* \in \mathcal{Q}.$$

Assuming that P is invertible as a mapping on $\mathcal{Q}$, we denote by P^+ the mapping from $P(\mathcal{Q})$ to $\mathcal{Q}$ which inverts P. It is not hard to verify that $f^* = P^+RPf$ where R is the orthogonal projector on $P(\mathcal{Q})$. Let A denote the operator that takes f to f^*.

Theorem 2.1 *Let C be a mapping from V to $\mathcal{Q}$. Let ϵ be T-periodic and in $L_2(0,T)$. Then*

$$||A(Pf + \epsilon) - f|| \leq (||P^+|| + 1)||Cf - f|| + ||P^+||\,||\epsilon||.$$

Proof: $A(Pf + \epsilon) = Af + A\epsilon$. $||Af - f|| \leq ||Af - Cf|| + ||Cf - f|| = ||A(f - Cf)||$ $+||Cf - f|| \leq (||A|| + 1)||f - Cf||$. $||A|| = ||P^+RP|| \leq ||P^+||$ because P and R are orthogonal projections. □

A main objective of the following work will be to bound $||P^+||$. This will be done by establishing upper bounds for $||I - P||$ as a mapping on $\mathcal{Q}$. From these, bounds can easily be derived for $||P^+||$.

Theorem 2.2 *Let $\eta < 1$ exist such that $||(I - P)q|| \leq \eta||q||$, for all $q \in \mathcal{Q}$. Then P is injective as a mapping on $\mathcal{Q}$ and for all $h \in P(\mathcal{Q})$, P^+, the inverse of the restriction of P to $\mathcal{Q}$, satisfies*

$$||P^+h||^2 \leq \frac{1}{1 - \eta^2}||h||^2.$$

We will obtain bounds for $||I - P||$ by considering the projector perpendicular to a spline space $\mathcal{G}$ which is more tractable than $P\mathcal{V}$, and on which $I - P$ is small. In the next section, $\mathcal{Q}$ is the approximating spline space, $\mathcal{S}$ a subspace of maximally continuous splines, and $\mathcal{G}$ is a space of maximally continuous splines whose knots are in a mesh refining the mesh for the members of $\mathcal{S}$. $\mathcal{S}$ and $\mathcal{G}$ have orthogonal projectors S and G, respectively. The following estimates $||I - P||$ in terms of $||I - G||$.

Theorem 2.3 *Suppose* $||(I-P)g|| \leq \eta_0||g||$ *for all* $g \in \mathcal{G}$. *Suppose* $||(I-G)q|| \leq \eta_1||q||$ *for all* $q \in \mathcal{Q}$. *Then* $||(I-P)q|| \leq (\eta_0 + \eta_1)||q||$ *for all for all* $q \in \mathcal{Q}$.

Proof: For $q \in \mathcal{Q}$, $||(I-P)q|| \leq ||(I-P)Gq|| + ||(I-P)(I-G)q||$.
$||(I-P)Gq|| \leq \eta_0||Gq|| \leq \eta_0||q||$, and $||(I-P)(I-G)q|| \leq ||(I-G)q|| \leq \eta_1||q||$. □

Theorem 2.4 enables us to bound $||I-G||$ on $\mathcal{Q}$ by instead bounding projectors orthogonal to small subspaces of $\mathcal{G}$, restricted to small subspaces of $\mathcal{Q}$.

Theorem 2.4 *Let* $\mathcal{G}$ *and* $\mathcal{S}$ *be closed subspaces of* $\mathcal{V}$ *with* $\mathcal{S} \subseteq \mathcal{G} \cap \mathcal{Q}$. *Let* $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_r$ *be nonzero mutually orthogonal subspaces of* $\mathcal{V}$. *Let* $\mathcal{Q}_i \subseteq \mathcal{Q} \cap V_i$, $1 \leq i \leq r$ *be nonzero closed subspaces such that* $\mathcal{Q} = \mathcal{S} + \mathcal{Q}_1 + \mathcal{Q}_2 + \cdots + \mathcal{Q}_r$. *Let* $\mathcal{G}_i \subseteq \mathcal{G} \cap V_i$, $\mathcal{H}_i \subseteq \mathcal{S}^\perp \cap V_i$, $1 \leq i \leq r$ *be nonzero closed subspaces with orthogonal projectors* G_i, H_i. *Let* ν *be a constant such that* $||(I - G_i)q_i||^2 \leq \nu||H_iq_i||^2$ *for all* $q_i \in \mathcal{Q}_i$, $1 \leq i \leq r$. *Then* $||(I - G)q||^2 \leq \nu||q||^2$ *for all* $q \in \mathcal{Q}$.

Proof: $q \in \mathcal{Q}$ can be written $q = s + v$ where $s \in \mathcal{S}$ and $v = q_1 + q_2 + \cdots + q_r$, $q_i \in \mathcal{Q}_i$, $1 \leq i \leq r$. $||(I - G)q|| = ||(I - G)v||$ since $\mathcal{S} \subseteq \mathcal{G}$. Let $F = G_1 + G_2 + \cdots + G_r$. Since $\mathcal{G}_1 + \mathcal{G}_2 + \cdots + \mathcal{G}_r \subseteq \mathcal{G}$, $||(I - G)v||^2 \leq ||(I - F)v||^2 = \sum_{i=1}^r ||(I - G_i)q_i||^2$, the latter equality because of orthogonality of the $\mathcal{G}_i$. $||q||^2 \geq ||(I - S)v||^2 \geq ||\sum_{i=1}^r H_iv||^2 = \sum_{i=1}^r ||H_iq_i||^2$, since $\sum_{i=1}^r \mathcal{H}_i \subseteq \mathcal{S}^\perp$, and the $\mathcal{H}_i$ are orthogonal. If all $H_iq_i = 0$ the hypothesis implies all $(I - G_i)q_i = 0$. The above then implies $(I - G)q = 0$, and the conclusion is true. We proceed assuming $H_iq_i \neq 0$ for some i and let $\mathcal{N}$ be the set of all those i. Then

$$\frac{||(I - G)q||^2}{||q||^2} \leq \frac{\sum_{i \in \mathcal{N}} ||(I - G_i)q_i||^2}{\sum_{i \in \mathcal{N}} ||H_iq_i||^2}.$$

An elementary argument shows the quotient of sums is $\leq \nu$ since for each $i \in \mathcal{N}$, $||(I - G_i)q_i||^2/||H_iq_i||^2 \leq \nu$. □

3 Bounds for restricted projectors

Below, we specialize the spaces of the last section, and get our main results. Let $T > 0$ be a fixed period. We take $\mathcal{V}$ to be the space of real-valued T-periodic functions which belong $L_2(I)$ for some, and thus every, period interval I. On $\mathcal{V}$ and its subspaces we define the inner product $\langle f, g \rangle = \int_I f(t)g(t)\,dt$, I a period interval. The other realizations are defined in the statements and proofs of the following results. Lemma 3.1 sets up an application of Theorem 2.4; Theorem 3.2 uses this, together with Theorem 2.2, to get our main result.

Lemma 3.1 *Let* X *be a finite set of points in* $[0, T)$. *Let* $N \geq 4$ *be an integer. Let* $K = \{iT/N, 0 \leq i \leq N\}$: *for each* $x \in X$, *let* k_x *be a member of* K *closest to* x *where*

0 is identified with T. Assume N large enough that between any two distinct k_x are at least three other members of K. Let K_X result from substituting in K each $x \in X$ for its k_x. For $m = 3, 4$ let $\mathcal{Q}$ be the space of m-th order T-periodic polynomial splines with K_X as knots and with continuity C^{m-2} at all knots except the $x \in X$, where no continuity is required. Let $\mathcal{G}$ be the space of m-th order periodic splines with knots in $[0, T)$ at the points $\{iT/(3N), 0 \le i \le 3N\}$, and let G be the orthogonal projector on $\mathcal{G}$. Then $I - G$ restricted to $\mathcal{Q}$ satisfies $||I - G||_2^2 \le .69$ if $m = 3$, and $||I - G||_2^2 \le .9$ if $m = 4$.

Proof: Let $\mathcal{S}$ be the subspace of $\mathcal{Q}$ consisting of those splines which are C^∞ at the k_x. Clearly $\mathcal{S} \subseteq \mathcal{G}$. Let $h = T/N$. Fix $x = x_i \in X = \{x_1, x_2, \ldots, x_r\}$ and let $y_0 = x_i$, $y_\alpha = k_{x_i} - \alpha h$, $\alpha = -2, -1, 1, 2$. Take $\mathcal{V}_i$ to be the subspace of $\mathcal{V}$ consisting of those functions with support in $[y_{-2}, y_2]$ and its T-translates.

For $m = 3$ let j_1 and j_2 be B-splines with knots y_{-1},y_0,y_0,y_0 and y_0,y_0,y_0,y_1; let j_3 be the difference of the B-splines with knots y_{-2},y_{-1},y_0, y_1 and y_{-1},y_0,y_1,y_2 (see [1] for explanation of multiplicity versus degree of continuity). For $m = 4$ let j_1 and j_2 be B-splines with knots $y_{-1}, y_0, y_0, y_0, y_0$ and y_0,y_0,y_0,y_0,y_1; let j_3 be the difference of the B-splines with knots y_{-2},y_{-1},y_0,y_0,y_1 and y_{-1},y_0,y_0,y_1,y_2; and let j_4 be the B-spline with knots y_{-2},y_{-1},y_0, y_1,y_2. Since $y_2 - y_{-2} < T$ we may identify the j_α with their T-periodic extensions.

Let Q_i be the space of splines whose generic member is $q_i = \sum_{\alpha=1}^{m} c_\alpha j_\alpha$ for constants c_α. For each i, nonzero members of $\mathcal{Q}_i$ have continuity from C^{m-2} through full discontinuity at x_i, while members of $\mathcal{S}$ are C^∞ at x_i. It follows that $\mathcal{S} \cap (\mathcal{Q}_1 + \mathcal{Q}_2 + \cdots + \mathcal{Q}_r) = 0$ and $\mathcal{Q} = \mathcal{S} + \mathcal{Q}_1 + \cdots + \mathcal{Q}_r$.

Let $\mathcal{G}_i$ be the subspace of G with basis the C^{m-2} periodic B-splines whose knots in the period containing $[y_{-2}, y_2]$ are length $m + 1$ sublists of consecutive knots from the list $(\alpha h/3 + k_x, -6 \le \alpha \le 6)$. Let $\mathcal{H}_i$ be the space of those m-th order periodic splines which in $[-T/2 + k_x, T/2 + k_x]$ have support in $[y_{-2}, y_2]$, which have knots at the y_i, $i \ne 0$ and at x, are C^{m-2} at y_{-1} and y_1, which may be fully discontinuous at y_{-2}, y_2, and x, and which are orthogonal to all members of $\mathcal{S}$. $||(I - G_i)q_i||^2 / ||H_i q_i||^2$ is a ratio of quadratic forms in the c_α. An upper bound ν for it can be obtained as an upper bound for the eigenvalues of the pencil $A - \lambda B$ where $a_{\alpha\beta} = \langle (I - G_i)j_\alpha, (I - G_i)j_\beta \rangle$, $b_{\alpha\beta} = \langle H_i j_\alpha, H_i j_\beta \rangle$, $1 \le \alpha, \beta, \le m$.

In [9] explicit bases for the spaces $\mathcal{G}_i$ and $\mathcal{H}_i$ are calculated as m-th order splines. From their definitions ([1]), B-splines are rational functions of the knots, and thus are also inner products of B-splines. The null-basis and orthogonal projection calculations in [9] use standard methods which involve only rational operations. Thus the $(I - G_i)j_\alpha$ and $H_i j_\alpha$ and then the $a_{\alpha\beta}$ and $b_{\alpha\beta}$ are rational functions of the knots of q_i, so long as x remains in $[k_x, k_x + h/3]$. When x crosses into $[k_x + h/3, k_x + h/2]$, thus crossing knots for splines in G_i, the rational functions change, so in general the matrix entries are piecewise rational functions of x.

Let ν be a conjectured upper bound for the maximum eigenvalue λ_{max} of $A - \lambda B$ (in [9] a floating point approximation to λ_{max} is plotted as a function of x; ν is determined from inspecting this plot). For computational convenience in [9] we represent x as $2\epsilon h/3 + k_x$, $0 \le \epsilon \le 1/2$ for $x \le k_x + h/3$, and as $(1 + \epsilon)h/3 + k_x$, $0 \le \epsilon \le 1/2$ for $k_x + h/3 \le x \le$

$k_x + h/2$. For further convenience we take $k_x = 0$, clearly losing no generality. We have represented only $x \geq k_x$, but because of symmetry, $x \leq k_x$ produces the same bounds.

Since h is a linear factor in all knots in the calculation, we see that $a_{\alpha\beta}$ and $b_{\alpha\beta}$ can be written as h multiplying piecewise rational functions of ϵ (with integer rational coefficients). The determinant of $A - \nu B$ is thus h^m times a piecewise rational function of ϵ. The MAXRAT algorithm ([9]) proves that its reciprocal is bounded as a function of ϵ in the appropriate ranges, so the determinant itself is bounded away from 0. In [9], ϵ is then set equal to 0 in $A - \tau B$, and the determinant of that matrix is then shown to have m sign changes as τ decreases from ν. Thus the conjectured value ν bounds all eigenvalues of $A - \lambda B$ for all values of x. The upper bounds thus obtained are $\nu = .69$ for $m = 3$ and $\nu = .9$ for $m = 4$. We emphasize that the B-splines, matrix entries, and determinants all are calculated exactly, using the Maple ([6, 7]) computer algebra system, so the bounding property of ν is rigorously proven. Since the bounds we obtain apply to the spaces $\mathcal{G}_i$ and B_i associated with any one of the x_i, they satisfy the hypotheses of Theorem 2.4 which now provides our conclusions. □

Our main result now follows.

Theorem 3.2 *Let the hypotheses be those of Lemma 3.1. In addition, let P be the orthogonal projector onto the space of n-th order real-valued T-periodic trigonometric polynomials, where $n \geq 3N$. If $m = 3$, we have $||P^+||_2 \leq 2.4$, while if $m = 4$, we have $||P^+||_2 \leq 4.5$.*

Proof: The space $\mathcal{G}$ in Lemma 3.1 consists of periodic splines with uniformly spaced knots. Theorem 3.1 of [10] implies that $||I - P||_2 \leq (\alpha/(1+\alpha))^{1/2}$ where

$$\alpha = 4\sum_{r=1}^{\infty}(1/(1+2r))^{2m}.$$

In [9] we use this formula to get upper bounds of .076 when $m = 3$ and .025 when $m = 4$ for $||I - P||_2$. Taking these bounds as η_0 in Theorem 2.3 and taking the bounds from Lemma 3.1 as η_1 in Theorem 2.3, we obtain from that theorem bounds for $||I - P||_2$ of .907 for $m = 3$ and .974 for $m = 4$. Theorem 2.2 now applies to produce the present results. □

Above, we required $n \geq 3N$; under this condition we can get our simplest and most comprehensive results. Since we contemplate applying our results where the number n of useful coefficients may be limited, we have tried to get versions of Theorem 3.2 where n is smaller compared with N. We have no useful versions for $n < 3N$ and $m = 4$ (cubic splines). The following result for quadratic splines may be useful. To formulate it, let $\epsilon_1 = \max\{|x - k_x|N/T\}$. In the previous results, the separation of the values x from their nearest uniform mesh points k_x was unrestricted, which corresponds to $\epsilon_1 = 1/2$. Here, we can get results for quadratic splines, and $n \geq 2N$, provided the x are more restricted; our methods of analysis "blow up" for $n \geq 2N$ as ϵ_1 approaches a number slightly larger than .25.

Theorem 3.3 *Let $m = 3$ (quadratic splines); let $n \geq 2N$. Otherwise, let the hypotheses be those of Theorem 3.2. Corresponding to the list* $0, .1, .2, .25$ *for values of* ϵ_1*, we have the list of values* $1.7, 2.1, 3.9, 16$ *as bounds for* $||P^+||$.

Proof: For each of the cases for ϵ, an argument similar to the proof of Lemma 3.1 applies to produce a bound η_1 for $||I - G||_2$ where $\mathcal{G}$ now is defined using the uniform knot spacing $1/(2N)$ rather than $1/(3N)$. The only difference in the argument is that here, a discontinuity location x always stays in the interval $[k_x, k_x + \epsilon_1 h]$ where $h = T/N$, so the matrix entries and determinants can be treated as functions of ϵ in $[0, \epsilon_1]$. Each bound η_1 now is used just as in the proof of Theorem 3.2, to get the present bounds for $||P^+||_2$. □

4 Uniform norm bounds

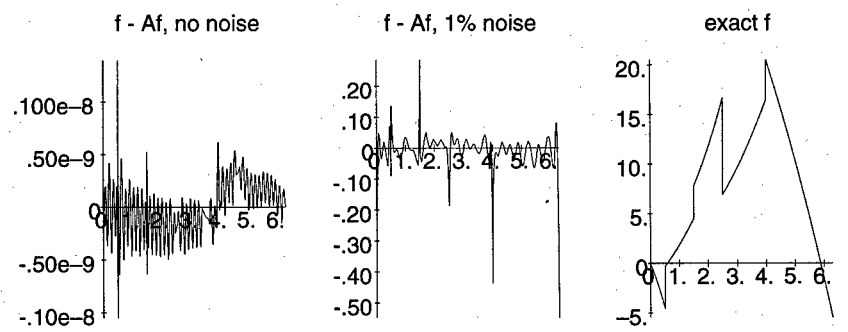
Using representers of point evaluation, as in [8], we can get uniform norm bounds for P^+, and thus for A. The arguments are similar to those in [8]. The main difference is that there the mesh is uniform and the order m is 3. The constructions of representers extend fairly easily to the present case: here the norms of representers are functions both of the evaluation point and the location of the discontinuity nearest to the evaluation point. One can show that for each point $t \in [0, T)$, a spline r_t exists in a space $\mathcal{U}$ containing $\mathcal{Q}$, such that $\langle r_t, q \rangle = q(t)$ for each $q \in \mathcal{Q}$, and such that $||r_t||_2 \leq k/\sqrt{h}$ where $k = 5$, $m = 3$ and $k = 7, m = 4$; $h = T/N$ as before. The computations for the construction and bound calculations are in [9]. Noting that $\sqrt{T}/\sqrt{h} = \sqrt{N}$, we have

$$||Af||_\infty \leq \max_t ||r_t||_2 ||Af||_2 \leq (k/\sqrt{h})||P^+||_2 \sqrt{T}||f||_\infty \leq k\sqrt{N}||P^+||_2||f||_\infty.$$

When $N \leq 100$ and the hypotheses are those of Lemma 3.1, this gives $||Af||_\infty \leq 120||f||_\infty$ for $m = 3$, and $||Af||_\infty \leq 315||f||_\infty$ for $m = 4$.

5 Example

FIG. 1.



We illustrate the method using an example where the function f is 2π-periodic and on $[0, 2\pi)$ consists of the function $e^{-x/6}$ with a piecewise quadratic added, so as to produce discontinuities at 0, .5, 1.5, 2.5, and 4. f is a modification of an example in [2]; for convenience we have shifted that example left by 1 unit, and we have added the exponential term because our method can represent a piecewise quadratic exactly in the absence of noise. Exact (up to 17-decimal digit floating point error) Fourier coefficients are derived from f by exact integration using the Maple ([6, 7]) system. Noisy approximate coefficients are also derived by sampling f at 1024 equidistant values in $[0, 2\pi]$, adding uniformly distributed pseudo-random noise to the samples, and taking the discrete Fourier transform of the samples. In effect, we work with $f + \epsilon$ where ϵ is a perturbing function. The level of the noise is set so that the discrete L_2-norm of the noise vector is 1% of the discrete L_2-norm of the vector of samples of f. $N = 45$ and thus $n = 135$ are the smallest values of n and N for which the hypotheses of the previous section are satisfied. Using these values, we proceed with $m = 4$ (cubic splines) for each of these cases for Fourier coefficients. Plots of f and of the error for the two cases appear in the figure. The ratio $||f - A(f + \epsilon)||_2/||f||_2$ is about .005 for the case of 1% noise. In [9] we develop a probabilistic estimate of .0037 for the ratio of $||\epsilon||_2/||f||_2$. This estimate indicates an L_2-norm noise magnification of about 1.35-fold, compared with the upper bound of 4.5 given in Theorem 3.2. The uniform error, for noise-free coefficients, is about 10^{-9}; computational experiments show this is dominated by truncation error in approximating the exponential term. In [9] we do the corresponding calculations for $m = 3$, and find similar results for 1% noise, with larger, but still small, error for noise-free coefficients.

In [9], we implement Eckhoff's method as described in [3], used on the above data. For noiseless data, the results are comparable to those reported by Eckhoff for similar examples. The uniform norm error seems to be about .06, with errors at jumps somewhat smaller. For 1% noise, the results of Eckhoff's method are about 750-fold in error.

Bibliography

1. C. de Boor, *Practical guide to splines*, Springer Verlag, New York (1978).
2. K. Eckhoff, *Accurate and efficient reconstruction of discontinuous functions from truncated series expansions*, Math. Comp. **61** (1993), 745–763.
3. K. Eckhoff, *Accurate reconstructions of functions of finite regularity from truncated Fourier series expansions*, Math. Comp. **64** (1995), 671–690.
4. D. Gottlieb and C.-W. Shu, *On the Gibbs phenomenon and its resolution*, SIAM Review **39** (1997), 644–667.
5. D. Gottlieb, C.-W. Shu, A. Solomonoff and H. Vandeven, *On the Gibbs phenomenon I: Recovering exponential accuracy from the Fourier partial sum of a nonperiodic analytic function*, J. Comput. Appl. Math. **43** (1992), 81–98.
6. K. M. Heal, M.L. Hansen, and K.M. Rickard, *Maple V Learning Guide*, Springer-Verlag New York (1998).
7. M. B. Monagan, K. O. Geddes, K. M. Heal, G. Labahn and S. M. Vorkoetter, *Maple V Programming Guide*, Springer Verlag, New York (1998).
8. R. K. Wright, *A robust method for accurately representing nonperiodic functions*

given Fourier coefficient information, J. Comput. Appl. Math. **140**, (2002) 837–848.

9. R. K. Wright *Computations and examples for spline approximation of discontinuous functions using low order Fourier coefficients*, UVM Math/Stat Department Technical Report 2001.2
10. R. K. Wright, *Spline fitting discontinuous functions given just a few Fourier coefficients*, Numerical Algorithms **9** (1995), 157–169.

Chapter 7

General Approximation

Preceding Page Blank

Remarks on delay approximations based on feedback

Alessandro Beghi and Antonio Lepschy

Dipartimento di Elettronica e Informatica, Università di Padova, Padova, Italy.
{beghi,lepschy}@dei.unipd.it

Wieslaw Krajewski

Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland.
krajewsk@ibspan.waw.pl

Umberto Viaro

Dipartimento di Ingegneria Elet., Mecc. e Gest., Università di Udine, Italy.
viaro@uniud.it

Abstract

The response of a unity-feedback system with a delay element in the forward path exhibits a periodic component that can be approximated by truncating its harmonic expansion. Rational approximants of the transfer function e^{-Ts} of such element can simply be obtained from this closed-loop approximation. A unifying approach to recent methods based on this criterion [2, 3] is presented, which allows us to point out their respective features. The standard Padé technique and a heuristic method described in [5] are also considered.

1 Introduction and problem statement

In modelling dynamic systems for control purposes, it is often necessary to account for time delays due, e.g., to transport phenomena or distributed-parameter components.

The response of an ideal delay element (delayor) to an input $u(t)$, identically equal to 0 for $t < 0$, is $y(t) = u(t-T)$, $T > 0$, where T indicates the time delay. By denoting with $U(s)$ the Laplace transform of $u(t)$, the Laplace transform of $y(t)$ is $Y(s) = e^{-Ts}U(s)$. Therefore the transfer function of the delayor is the transcendental function e^{-Ts}.

The problem of approximating e^{-Ts} by means of a rational function has a long history (see, e.g., [1]) but is still important from both the computational and the conceptual point of view; a few recent contributions on the subject are quoted in [2]. In many practical applications the physical realizability and the stability of the approximant limit the choice of the approximant to proper rational functions with real coefficients and a Hurwitz denominator. These requirements are satisfied by Blaschke products, i.e., functions of the form:

$$B(s) = \frac{\prod_{i=1}^{n}(s-a_i)}{\prod_{i=1}^{n}(s+a_i)}, \qquad \mathbb{R}e\,[a_i] > 0\,. \tag{1.1}$$

This has the desirable property that $|B(j\omega)| = |e^{-jT\omega}| = 1$, $\forall\omega$, and arg[B(jω)] is monotonically decreasing with ω like arg[$e^{-jT\omega}$] = $-T\omega$. On the other hand, the step response of a system with transfer function $B(s)$ starts from $+1$ or -1, whereas the step response of an ideal delayor obviously starts from 0.

The most widely adopted method to form a rational approximant of a delay element is based on the Padé technique which does not always guarantee stability (even if biproper Padé models are necessarily stable). Since such a technique leads to the retention of the first Maclaurin expansion coefficients of e^{-Ts}, the resulting approximation is the best in the neighbourhood of $\omega = 0$. In different frequency bands, other types of models may be preferred.

In [3] a unity-feedback system whose forward path consists of a delayor is analysed.

In the case of negative feedback, the unit step response is a piecewise constant function taking on the value 0 for $2kT < t < (2k+1)T$ and the value 1 for $(2k+1)T < t < (2k+2)T$, $k \geq 0$, which can be decomposed into a step of amplitude $\frac{1}{2}$, and a square wave of amplitude $\frac{1}{2}$ starting from $-\frac{1}{2}$ at $t = 0$.

In the case of positive feedback, similar considerations allow us to decompose the unit step response into a linear ramp of slope $\frac{1}{T}$, a step of amplitude $-\frac{1}{2}$, and a saw-tooth wave that linearly decreases from $\frac{1}{2}$ to $-\frac{1}{2}$ in every period from kT to $(k+1)T$.

In both cases, the periodic component can easily be expressed as a series of harmonic terms (for $t > 0$). It is therefore natural to approximate the step response of the unity-feedback system by retaining the non-periodic component together with a suitable number of the first harmonics of the periodic component.

A rational approximation $W_a(s)$ of the transcendental transfer function $W(s)$ of the above-mentioned feedback system is obtained by dividing the Laplace transform of the approximate step response by the Laplace transform $\frac{1}{s}$ of the step input. The rational approximant $G_a(s)$ of the delayor transfer function is then determined as

$$G_a(s) = \frac{W_a(s)}{1 \mp W_a(s)}\,, \tag{1.2}$$

where the minus sign applies to the case of negative feedback and the plus sign to that of positive feedback. It turns out [3] that $G_a(s)$ is a stable biproper rational function having the form of a Blaschke product; precisely, negative feedback supplies even-order approximants and positive feedback produces odd-order approximants.

Obviously, the same result could be achieved by referring to different inputs (even an impulse), but the choice of the unit step is particularly convenient. According to the terminology suggested in [4], the rationale of such a procedure consists in retaining the "input component" (and the "resonant component", if any) and in truncating the periodic "system component" of the response.

In [2] a feedback structure is used as well, but another approximation criterion is adopted, which leads to different models depending on the chosen input. In particular, the family of inputs considered in [2] is $\{u(t) = t^m\,, m \in \mathbb{N}\,, t > 0\}$, and the procedure exploits several properties of Bernoulli numbers and polynomials.

In the following, the above approaches are presented in a unified form which allows us to point out their respective features and to derive the related approximants in an

easier way. Finally, criteria are given to choose the approximation that is most suited to the application at hand, also taking into account the standard Padé approximation and a further approximation presented in [5].

2 Derivation of the approximant

For the sake of simplicity, we shall almost exclusively refer to the case of negative feedback; only a brief mention will be made of the case of positive feedback.

2.1 Negative feedback

The transfer function $W(s)$ of the negative feedback system with forward-path transfer function $G(s) = e^{-Ts}$ is

$$W(s) = \frac{G(s)}{1+G(s)} = \frac{1}{e^{Ts}+1}, \tag{2.1}$$

whose singularities (poles) are the roots of $e^{Ts} = -1$, i.e.

$$s = \pm jp_k := \pm j(2k-1)\frac{\pi}{T},\ k \in \mathbb{Z}_+ .$$

$W(s)$ can also be interpreted as the Laplace transform of the sequence of positive and negative impulses forming the *derivative* of the step response described in the introduction. Therefore, it is the sum of a constant equal to $\frac{1}{2}$ (corresponding to the step component in the just-mentioned step response) and a series of "harmonic" terms associated with the above poles:

$$W(s) = \frac{1}{2} + \sum_{k=1}^{\infty}\left[\frac{r_k}{s-jp_k} + \frac{\bar{r}_k}{s+jp_k}\right],$$

where the bar denotes conjugate and, using the standard formula for the residues,

$$r_k = \lim_{s\to jp_k}(s-jp_k)W(s) = -\frac{1}{T}.$$

It follows that

$$W(s) = \frac{1}{2} - \frac{2}{T}\sum_{k=1}^{\infty}\frac{s}{s^2+(2k-1)^2\frac{\pi^2}{T^2}}. \tag{2.2}$$

In order to compare the results in [2] and [3], let us consider a canonical input of the form

$$u_i(t) = \frac{t^{i-1}}{(i-1)!},\ t>0, \tag{2.3}$$

whose Laplace transform is

$$U_i(s) = \frac{1}{s^i}.$$

(In [3] only the case of $i = 1$ is considered, whereas the inputs used in [2] differ from (2.3) by a scaling factor which is irrelevant for the following considerations.)

On the basis of (2.2) the Laplace transform of the (forced) response to (2.3),

$$Y_i(s) = \frac{1}{s^i} W(s)\,,$$

can be rewritten as

$$Y_i(s) = \frac{1}{s^i} \sum_{h=0}^{i-1} c_h s^h + \sum_{k=1}^{\infty} \frac{\alpha_{ki} + \beta_{ki} s}{s^2 + (2k-1)^2 \frac{\pi^2}{T^2}}\,,$$

where for i even,

$$\alpha_{ki} = 0\,, \qquad \beta_{ki} = (-1)^{\frac{i}{2}} \frac{2}{T} \left[\frac{T}{(2k-1)\pi}\right]^i\,, \tag{2.4i}$$

and for i odd,

$$\alpha_{ki} = (-1)^{\frac{i-1}{2}} \frac{2}{T} \left[\frac{T}{(2k-1)\pi}\right]^{(i-1)}\,, \qquad \beta_{ki} = 0\,. \tag{2.4ii}$$

Therefore, $W(s)$ can also be presented in the alternative form

$$W(s) = \frac{Y_i(s)}{U_i(s)} = \sum_{h=0}^{i-1} c_h s^h + \sum_{k=1}^{\infty} \frac{\alpha_{ki} s^i + \beta_{ki} s^{i+1}}{s^2 + (2k+1)^2 \frac{\pi^2}{T^2}}\,. \tag{2.5}$$

Each term of the series in (2.5) is given by the sum of a polynomial of degree $i-1$ (quotient of the division of its numerator by its denominator) and a strictly proper rational function (whose numerator is the remainder of the division). Therefore, (2.5) becomes

$$W(s) = \sum_{h=0}^{i-1} c_h s^h + \sum_{k=1}^{\infty} \left\{ \sum_{h=0}^{i-1} d_{ki,h} s^h + \frac{\gamma_{ki} + \delta_{ki} s}{s^2 + (2k-1)^2 \frac{\pi^2}{T^2}} \right\} \tag{2.6}$$

which can be rewritten as

$$W(s) = \sum_{h=0}^{i-1} \left(c_h + \sum_{k=1}^{\infty} d_{ki,h} \right) s^h + \sum_{k=1}^{\infty} \frac{\gamma_{ki} + \delta_{ki} s}{s^2 + (2k-1)^2 \frac{\pi^2}{T^2}}\,. \tag{2.7}$$

By comparing (2.7) with (2.2), one finds that

$$c_0 + \sum_{k=1}^{\infty} d_{ki,0} = \frac{1}{2}\,, \tag{2.8}$$

$$c_h + \sum_{k=1}^{\infty} d_{ki,h} = 0\,, \quad h > 0, \tag{2.9}$$

$$\gamma_{ki} = 0\,, \quad \forall\, k, i,$$

$$\delta_{ki} = -\frac{2}{T}\,, \quad \forall\, k, i\,.$$

The procedure suggested in [3] could alternatively be presented with reference to expression (2.7) where coefficients related to the specific input appear. Precisely, the approximant $W_a(s)$ is obtained in this case by adding to the exact value $\frac{1}{2}$ of the first

sum (cf. (2.8) and (2.9)) the first K (harmonic) terms of the second summation

$$W_a(s) = \frac{1}{2} - \frac{2}{T}\sum_{k=1}^{K} \frac{s}{s^2 + (2k-1)^2\frac{\pi^2}{T^2}},$$

which is *independent* of the input $u_i(t)$.

The procedure suggested in [2] refers instead to expressions (2.5) or (2.6), and the approximation consists in truncating the summation over k, where each addendum is formed by a polynomial and a strictly proper harmonic term. Therefore the resulting $W_a(s)$ is

$$W_a(s) = \sum_{h=0}^{i-1} c_h s^h + \sum_{k=1}^{K} \frac{\alpha_{ki} + \beta_{ki} s}{s^2 + (2k-1)^2\frac{\pi^2}{T^2}}, \tag{2.10}$$

which does depend on i and it is not proper because the part added to the harmonic terms does not reduce to the constant $\frac{1}{2}$, as is instead the case in $W(s)$. Nevertheless, the approximant $G_a(s) = W_a(s)/(1 - W_a(s))$ of e^{-Ts} turns out to be biproper.

As concerns the computation of the above approximants, the suggested approach seems to be preferable to that adopted in [2] because

(i) coefficients c_h, which correspond to the first i Maclaurin expansion coefficients of

$$W(s) = \frac{1}{1 + \sum_{h=0}^{\infty} \frac{(Ts)^h}{h!}},$$

can be easily be evaluated using the classic Padé procedure, and

(ii) formulae (2.4*i*) and (2.4*ii*) immediately supply coefficients α_{ki}, β_{ki}.

2.2 Positive feedback

Considerations analogous with those of Section 2.1 lead to the following transfer function in the case of positive feedback

$$W(s) = \frac{1}{Ts} - \frac{1}{2} + \frac{2}{T}\sum_{k=1}^{\infty} \frac{s}{s^2 + \left(\frac{2k\pi}{T}\right)^2}, \tag{2.11}$$

so that $Y_i(s) = W(s)U_i(s)$ can be separated into a (harmonic) series associated with the imaginary conjugate poles of $W(s)$ and a strictly proper fraction with denominator s^{i+1}. Using the terminology in [4], the mentioned series corresponds to the "system component" of the forced response and the fraction corresponds to its "interaction component" because the poles of the latter are common to $W(s)$ and $U_i(s)$ (no "input component" is present in this case since $U_i(s)$ does not exhibit poles different from those of $W(s)$).

As shown in [3], the truncation of the series in (2.2) results in even-order biproper approximants $G_a(s)$, whereas the truncation of the series in (2.11) results in odd-order biproper approximants $G_a(s)$.

Instead, as shown in [2], truncating the series in (2.5) leads to odd-order approximants, whereas truncating the analogous series corresponding to positive feedback leads to even-order approximants.

2.3 Stability and approximation error

It has been proved [3] that the even-order rational approximations $G_a(s)$ of e^{-Ts} obtained from (2.1), as well as the odd-order ones obtained by truncating (2.11), are stable. Instead, as explicitly stated in [2] for inputs t^m, $m > 2$ (i.e., using the previous notation, $u_i(t)$ with $i > 3$) the "alternating sign of the Bernoulli numbers makes the approximation in general unstable [...]. Hence, from a practical point of view, any improvement with respect to the approximants obtained in [3] is to be found with $p = 1$", i.e., $i = 2, 3$.

The approximation accuracy can be evaluated by referring, e.g., to the "closed-loop error"

$$E(s) := W(s) - W_a(s) .$$

From (2.1) we get

$$E(s) = E_1(s) := -\frac{2}{T} \sum_{k=K+1}^{\infty} \frac{s}{s^2 + p_k^2} ,$$

whereas from (2.10) we have

$$E(s) = E_2(s) := \sum_{h=0}^{i-1} \sum_{k=K+1}^{\infty} d_{ki,h} s^h + E_1(s) .$$

Since $E(s)$ is a complex quantity, $|E_2(s)|$ may well be smaller than $|E_1(s)|$ for certain values of s (or $j\omega$).

3 Alternative approximants

As already pointed out, the procedure suggested in [2] leads to approximants that depend on the chosen canonical input. To improve the approximation within suitable frequency bands not centred at the origin, it is reasonable to resort to non-canonical inputs whose spectrum has larger amplitude there. A simple choice corresponds, e.g., to

$$U(s) = \frac{1}{1 + 2\xi \dfrac{s}{\omega_n} + \dfrac{s^2}{\omega_n^2}} ,$$

in which ω_n is at the centre of the band and ξ is suitably small.

The choice of the form of the input (as well as the order of the canonical input) is somewhat arbitrary and is influenced, in practice, by empiric considerations. Therefore, it makes sense to compare the results of the above procedures with those obtained in [5] using a heuristic procedure based on the direct approximation of the phase Bode diagram of $e^{-jT\omega}$ by means of a Blaschke product $B_n(j\omega)$ of order n. For n odd, the first factor of $B_n(s)$ has the form

$$G_1(s) = \frac{1 - \tau s}{1 + \tau s} , \qquad \tau > 0,$$

and the others have the form

$$G_i(s) = \frac{1 - 2\xi_i \dfrac{s}{\omega_{ni}} + \dfrac{s^2}{\omega_{ni}^2}}{1 + 2\xi_i \dfrac{s}{\omega_{ni}} + \dfrac{s^2}{\omega_{ni}^2}}, \quad 1 > \xi_i > 0, \quad \omega_{ni} > 0, \tag{3.1}$$

whereas for n even all factors have form (3.1).

All the considered techniques produce unit-magnitude all-pass frequency responses so that the approximation they afford can be judged with reference to the phase deviation $\Delta(j\omega)$ from $-T\omega$ only. As $\omega \to \infty$, $\Delta(j\omega) \to \infty$ in all cases. Therefore, reasonable criteria for choosing the method most suited to the specific application are: (i) the bandwidth B_ϵ where $|\Delta(j\omega)|$ is less than a specified value ϵ, or (ii) the maximum Δ_B of $|\Delta(j\omega)|$ in a prescribed band B.

By way of example, Fig. 1 shows $\Delta(j\omega)$ vs ω for the 4-th order all-pass approximants of $e^{-j\omega}$, $(T = 1)$ obtained according to (2.1) with $K = 2$ (curve a), to the procedure suggested in [2] for $u_3(t) = t^2$ (curve b), to the standard Padé procedure (curve c), and to the heuristic method in [5] (curve d). For instance, with reference to criterion (i) above, the Padé approximant is best for ϵ very small, the method suggested in [2] is optimal for $\epsilon \simeq 10°$, and the heuristic method and the method suggested in [3] are preferable for $\epsilon \geq 45°$.

Analogous results are obtainable for approximants of different order.

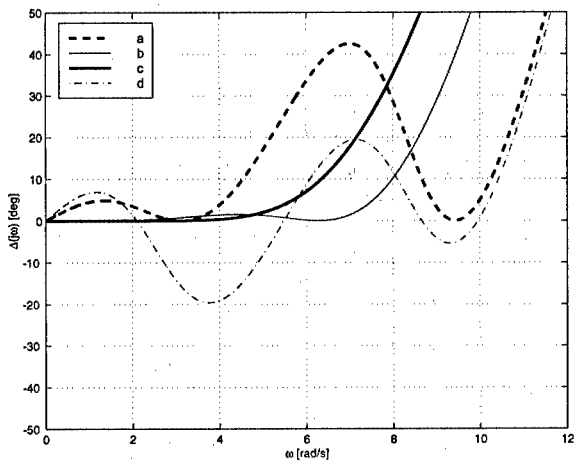


FIG. 1. Phase deviations $\Delta(j\omega)$ for the considered 4-th order approximants.

4 Conclusions

The approximation procedure presented in [2] and [3] have been embedded in a unified frame which points out well their respective features and allows us to determine the

parameters of the approximants in an easier way. Criteria have been provided for choosing the approximation method that is most suited to the specific application.

Bibliography

1. O. Perron, *Die Lehre von den Kettenbrüchen.* Stuttgart: Teubner, 1913. 3rd ed. 1957. In German.
2. C. Battle and A. Miralles, "On the approximation of delay elements by feedback," *Automatica*, vol. 36, pp. 659–664, 2000.
3. A. Beghi, A. Lepschy, and U. Viaro, "Approximating delay elements by feedback," *IEEE Trans. Circ. Sys. I*, vol. 44, pp. 824–828, 1997.
4. P. Dorato, A. Lepschy, and U. Viaro, "Some comments on steady-state and asymptotic responses," *IEEE Trans. Education*, vol. 37, pp. 264–268, 1994.
5. A. Beghi, A. Lepschy, and U. Viaro, "On the simplification of the mathematical model of a delay element," in E. Kuljanic, ed., *Advanced Manufacturing Systems and Technology*: Springer Verlag, 1996, pp. 617–624.

Point shifts in rational interpolation with optimized denominator

Jean–Paul Berrut

Département de Mathématiques, Université de Fribourg, Switzerland
jean-paul.berrut@unifr.ch

Hans D. Mittelmann

Department of Mathematics, Arizona State University, Tempe, USA
mittelmann@asu.edu

Abstract

In previous work we have suggested obtaining rational interpolants of a function f by attaching optimally placed poles to its interpolating polynomials. For a large number of interpolation points these polynomials are well-known to be good approximants only if the nodes tend to cluster near the endpoints of the interval, as with Čebyšev or Legendre points. In practice, however, one would prefer to have them closer to equidistant. This will in particular be the case when the difficult portion of f lies well within the interior of the interval, or when approximating derivatives of f, as in the solution of differential equations. To address this difficulty, we use here a conformal change of variable to shift the points from the Čebyšev position toward a more equidistant distribution in a way that should maintain the exponential convergence when f is analytic. Numerical examples demonstrate the resulting improvement in the quality of the approximation.

1 Introduction

We are concerned here with rational approximation of a continuous function f on an interval $[a, b]$, which we may take as $[-1, 1] =: I$, after a linear change of variable when necessary. We further assume that the approximant r should interpolate f between a finite number, say $N + 1$, of distinct points (nodes) $x_0, x_1, \ldots, x_N$ in I. In a similar way as in [5], r will be constructed by attaching a certain number of poles to an interpolating polynomial.

In some applications, such as the numerical solution of two-point boundary value problems (see, e.g., [6]), one may choose the points more or less at will; in that case, one will place them so as to reach the best compromise between two often conflicting goals: points good for interpolation, on one side, and points favourable for the condition of the problem to be solved, on the other side. In [5], we have considered equidistant and Čebyšev points, the first for their regularity, the second for the condition of the interpolation and for the fast convergence of the interpolant for very smooth functions. For the solution of two-point boundary problems in [6] we have merely used Čebyšev points.

There is in general no reason besides the problem condition for accumulating the nodes toward the boundary, as with Čebyšev or Legendre points. Moreover, one of the reasons for using rational instead of polynomial interpolation is its better suitability for approximating functions with large slopes. Here too, shifting the points away from the center may not be appropriate.

Another odd consequence of accumulating interpolation points toward the extremities is the consequent ill-conditioning of the derivatives of the interpolating polynomials [7, 1]. This worsens the stability properties of time-stepping in the solution of time evolution problems with the method of lines [13] as well as the convergence of iterative methods for solving discretized stationary problems [3].

To address these difficulties, we will take advantage here of the fact that the fast convergence of the interpolant can be maintained while shifting the points with a conformal map g (independent of N) toward an equidistant position. This, however, requires an important change to the method in [5], because this point shift ruins the exponential convergence of the Čebyšev interpolating polynomial. We therefore use here as the starting interpolant the polynomial interpolating $f(g^{-1})$ in the domain of the inverse g^{-1} of the conformal map employed for the point shift, and attach poles to this polynomial.

Section 2 reviews the formulae and advantages of shifting Čebyšev points conformally toward the center of the interval when interpolating functions, and Section 3 briefly recalls the method of optimally attaching poles to the interpolating polynomial introduced in our earlier work. In Section 4 we describe how to take advantage of the better conditioning of derivatives induced by the conformal point shift; the corresponding practical improvements are finally documented with numerical examples.

2 Rational interpolation with a variable change for point shifts

Let $\mathcal{P}_m$ and $\mathcal{R}_{m,n}$, respectively, denote the linear space of all polynomials of degree at most m and the set of all rational functions with numerator in $\mathcal{P}_m$ and denominator in $\mathcal{P}_n$; furthermore, denote by f_k the interpolated values $f(x_k)$, $k = 0(1)N$, of f. Then, the unique polynomial $p \in \mathcal{P}_N$ that interpolates f at the x_k's,

$$p(x) = \sum_{k=0}^{N} f_k L_k(x), \qquad L_k(x) := \prod_{i \neq k} (x - x_i) \Big/ \prod_{i \neq k} (x_k - x_i),$$

can be written in its *barycentric form* [9]

$$p(x) = \sum_{k=0}^{N} \frac{w_k}{x - x_k} f_k \Big/ \sum_{k=0}^{N} \frac{w_k}{x - x_k}, \tag{2.1}$$

where the so–called *weight* w_k corresponding to the point x_k is given by

$$w_k := 1 \Big/ \prod_{i=0,\ i \neq k}^{N} (x_k - x_i).$$

Despite its appearance, (2.1) determines a polynomial of degree at most N: the w_k are precisely the numbers which guarantee this [4]. By choosing other w_k's, a rational

interpolant is constructed.

The barycentric formula has several advantages over other representations of the interpolating polynomial ([4] p. 357). One of them is the fact that the weights appear in both the numerator and the denominator, so that they can be divided by any common factor. For example, simplified weights for Čebyšev points of the first kind $x_k^{(1)} := \cos\phi_k$, where $\phi_k := \frac{2k+1}{2(n+1)}\pi$ and $k = 0, \ldots, N$, are given by $w_k^{(1)} = (-1)^k \sin\phi_k$ ([9] p. 249), while for the Čebyšev points of the second kind $x_k^{(2)} := \cos k\frac{\pi}{N}$ – which will be used here – one simply has Salzer's formula ([9] p. 252)

$$w_k^{(2)} = (-1)^k \delta_k, \qquad \delta_k = \begin{cases} 1/2, & k = 0 \text{ or } k = N, \\ 1, & \text{otherwise.} \end{cases}$$

These points are, together with Legendre's, the most used nodes for global polynomial interpolation and large N. They achieve exponential convergence of p toward f if the latter is analytic in an ellipse E_ρ with foci at ± 1 and sum of its axes equal to 2ρ, $\rho > 1$. However, this fast convergence comes at the cost of a concentration of the nodes in the vicinity of the extremities of I. As mentioned above, this accumulation may have drawbacks, such as poor spreading of the information about f over the interval and ill-conditioning of the derivatives near the endpoints.

With a suitable choice of the interpolant, one may conformally shift the nodes toward an equidistant position (though not all the way) without losing the exponential convergence. For that purpose, one considers, beside the x-space in which f is to be approximated, another space, denoted by y, say, and the $N+1$ Čebyšev points of the second kind

$$y_k = x_k^{(2)}$$

in the interval $J := [-1, 1]$ in this y-space. Let g be a conformal map from a domain $\mathcal{D}_1$ containing J (in the y-space) to a domain $\mathcal{D}_2$ containing I (in the x-space); moreover, suppose that f is a function $\mathcal{D}_2 \mapsto \mathbb{C}$ such that the composition $f \circ g : \mathcal{D}_1 \mapsto \mathbb{C}$ is analytic in an ellipse E_ρ, as defined above. With this map we may define new interpolation points on I, $x_k = g(y_k)$, as well as the conformal transplantation $F(y) := f(x)$ [10] of f into the y-space.

Then, with the polynomial interpolating $F(y)$ at the y_k

$$A_N(y) := \sum_{k=0}^{N} F(y_k) L_k(y) = \sum_{k=0}^{N} f(x_k) L_k\big(g^{-1}(x)\big) =: a_N(x), \tag{2.2}$$

one has

$$|a_N(x) - f(x)| = \mathcal{O}(\rho^{-N}), \qquad x \in [-1, 1].$$

Rational interpolation with all poles prescribed is very simple in the barycentric setting [5]: the P poles z_i are attached to (2.1) by replacing w_k with

$$b_k = w_k d_k, \qquad d_k := \prod_{i=1}^{P} (x_k - z_i).$$

If $N \geq P$ this results in a rational interpolant in $\mathcal{R}_{N,P}$ with poles at z_i, $i = 1, \ldots, P$ (when such an interpolant exists, see [5]).

Remark 2.1 *Exponential convergence of interpolation at the shifted points is also attained with the rational function given by (2.1) with* $w_k = w_k^{(2)}$ *[2]. However, this is in general a rational function in* $\mathcal{R}_{N,\nu}$, $\nu > N - P$: *there is not enough defect in the denominator degree for the weights* $w_k^{(2)} d_k$ *to warrant the presence of the* P *poles* z_i.

We then use a_N as the starting interpolant to which we attach the poles v_i in the y-space. This yields

$$R(y) := \frac{\displaystyle\sum_{k=0}^{N} \frac{w_k \prod_{i=1}^{P}(y_k - v_i)}{y - y_k} f_k}{\displaystyle\sum_{k=0}^{N} \frac{w_k \prod_{i=1}^{P}(y_k - v_i)}{y - y_k}} = \frac{\displaystyle\sum_{k=0}^{N} \frac{w_k \prod_{i=1}^{P}\left(g^{-1}(x_k) - g^{-1}(z_i)\right)}{g^{-1}(x) - g^{-1}(x_k)} f_k}{\displaystyle\sum_{k=0}^{N} \frac{w_k \prod_{i=1}^{P}\left(g^{-1}(x_k) - g^{-1}(z_i)\right)}{g^{-1}(x) - g^{-1}(x_k)}} =: r(x).$$

If a rational interpolant with these poles exists, it is given in the y-space by R, and r is a rational function in the argument $g^{-1}(x)$. Its poles are at $z_i = g(v_i)$.

3 Construction of the optimal interpolant

Our method consists in optimizing the position of the v_i's so as to minimize

$$\|R - F\|_\infty = \|r - f\|_\infty,$$

as described in §3 of [5]. Optimal v_i's always exist, but these are not unique in general. Whether the optimal R is unique is an open question; however, for every optimized pole v_i an indicator may be calculated which, if nonzero, guarantees that v_i is indeed a pole of R.

In the practical computations documented in §5 the optimization of the v_i's was performed using the same two algorithms as in [5]: for small N we used a discrete differential correction algorithm according to [11], while for larger N the simulated annealing method of [8] was applied. Both methods will in principle locate a desired global maximum. The first method achieves it in a systematic and guaranteed way evaluating the error not continuously but on a fine grid; the simulated annealing method cannot be guaranteed to find the global extremum but, when used for an extensive search, will produce a reasonable approximation of it.

As mentioned in [5], our way of attaching poles to the interpolating polynomial has a very nice property: the approximation error can only decrease or at worst stay constant with a growing number of poles, this in sharp contrast with classical rational interpolation; when a new unknown, say v_j, is added to the set of variables, $\{v_1, \ldots, v_{j-1}\}$, the optimal values of the latter are a feasible vector for the higher dimensional optimization.

Let us conclude this section with a comment on the use of the nomenclature "attaching the poles". In classical rational interpolation, the poles of the interpolant are

determined by the data. There too, however, one sometimes wishes to prescribe the location of the poles (with corresponding decrease of the number of degrees of freedom): many authors then speak of "assigning", or "prescribing" the poles. In that sense one cannot "assign" poles to a polynomial, which obviously cannot have poles. We thus start with the interpolating polynomial and its poles at infinity and make it a rational interpolant by bringing the poles into an optimal position in $\mathbb{C}$. We call this procedure "attaching poles", to distinguish it from the process of forcing a rational function to have a pole at a particular place.

4 Derivatives of the optimal interpolant with shifted points

As mentioned in §1, one of the reasons for shifting the points from their Čebyšev position toward the interior of the interval is the improvement of the condition of the derivatives resulting from such a shift. Besides r, we will evaluate also r' and r'' as approximants of f', resp. f'', and estimate $\|r - f'\|_\infty$ and $\|r - f''\|_\infty$.

Schneider and Werner [14] have noticed that every rational interpolant $R \in \mathcal{R}_{N,N}$, written in its barycentric form

$$R(y) = \sum_{k=0}^{N} \frac{u_k}{y - y_k} f_k \Big/ \sum_{k=0}^{N} \frac{u_k}{y - y_k},$$

can easily be differentiated. The formulae for the first two derivatives read

$$R'(y) = \begin{cases} \displaystyle\sum_{k=0}^{N} \frac{u_k}{y - y_k} R[y, y_k] \Big/ \sum_{k=0}^{N} \frac{u_k}{y - y_k}, & y \neq y_i,\ i = 0(1)N, \\ \displaystyle -\Big(\sum_{\substack{k=0 \\ k \neq i}}^{N} u_k R[y_i, y_k] \Big) \Big/ u_i, & y = y_i \end{cases}$$

and

$$R''(y) = \begin{cases} \displaystyle 2\sum_{k=0}^{N} \frac{u_k}{y - y_k} R[y, y, y_k] \Big/ \sum_{k=0}^{N} \frac{u_k}{y - y_k}, & y \neq y_i,\ i = 0(1)N, \\ \displaystyle -2\Big(\sum_{\substack{k=0 \\ k \neq i}}^{N} u_k R[y_i, y_i, y_k] \Big) \Big/ u_i, & y = y_i, \end{cases}$$

with $R[z, z, y_k] = \frac{R'(z) - R[z, y_k]}{z - y_k}$. The chain rule then yields, for $r(x) = R\big(g^{-1}(x)\big)$,

$$r'(x) = R'(y) \cdot \big[g^{-1}(x)\big]' = \frac{R'(y)}{g'(y)}, \qquad r''(x) = \frac{1}{[g'(y)]^2} R''(y) - \frac{g''(y)}{[g'(y)]^3} R'(y). \quad (4.1)$$

Specifically, in our calculations we have used the map suggested by Kosloff and Tal-Ezer [12],

$$g(y) = \frac{\arcsin(\alpha y)}{\arcsin \alpha}, \qquad 0 < \alpha < 1.$$

α	$P=0$	$P=2$	$P=4$	$P=6$	$P=8$
0.0	6.37e − 5	1.42e − 6	5.83e − 8	9.38e − 9	1.30e − 9
0.5	3.11e − 5	6.69e − 7	2.48e − 8	4.21e − 9	4.23e − 10
0.75	8.06e − 6	1.60e − 7	5.50e − 9	9.47e − 10	1.27e − 10
0.9	1.12e − 6	1.97e − 8	5.90e − 10	3.94e − 11	2.05e − 11
0.95	2.78e − 7	4.47e − 9	1.29e − 10	1.36e − 11	3.82e − 12
0.96	1.85e − 7	2.93e − 9	8.27e − 11	4.20e − 12	3.88e − 12

TAB. 1. Errors when approximating f with increasing P and α in Example 1.

In the limiting cases, $\alpha \to 0$ keeps the points at their Čebyšev position, whereas $\alpha \to 1$ renders them equidistant. The derivatives of g are given by

$$g'(y) = \frac{\alpha}{\arcsin \alpha} \frac{1}{\sqrt{1-(\alpha y)^2}}, \qquad g''(y) = \frac{\alpha^3}{\arcsin \alpha} \frac{y}{\sqrt{\left(1-(\alpha y)^2\right)^3}},$$

so that in (4.1)

$$\frac{g''(y)}{[g'(y)]^3} = (\arcsin^2 \alpha) y.$$

5 Numerical evidence

We now report on practical computations, performed on two examples, which demonstrate the efficiency of point shifts for improving the rational interpolants with optimized denominators. These examples share the property that the difficult part of f lies in the center of I, so that the shift of the points toward a more equidistant position naturally improves the quality of the information provided to the interpolation method.

α	$P=0$	$P=2$	$P=4$	$P=6$	$P=8$
0.0	5.27e − 3	1.26e − 3	4.85e − 6	8.69e − 7	1.40e − 7
0.5	2.67e − 3	5.87e − 4	2.33e − 6	4.03e − 7	4.63e − 8
0.75	7.47e − 4	1.49e − 5	5.16e − 7	9.44e − 8	1.30e − 8
0.9	1.14e − 4	2.01e − 6	6.56e − 8	4.28e − 9	2.16e − 9
0.95	2.97e − 5	4.99e − 7	1.48e − 8	1.59e − 9	4.52e − 10
0.96	2.01e − 5	3.24e − 7	9.52e − 9	4.80e − 10	4.70e − 10

TAB. 2. Errors when approximating f' with increasing P and α in Example 1.

The sup-norm $\| \quad \|_\infty$ has thereby been estimated by considering the 1000 equally spaced points $\widehat{x}_\ell = -\frac{5}{4} + \frac{\ell-1}{999}\frac{10}{4}$, $\ell = 1(1)1000$, on the interval $[-5/4, 5/4]$ and computing the maximal absolute value of the error at those $\widehat{x}_\ell$ lying in $[-1, 1]$.

Example 5.1 We have first revisited Example 3 of [5], which displays in the center of

I a slope increasing with a positive parameter, here denoted by ϵ,

$$f(x) = \cos \pi x + \frac{\mathrm{erf}(\delta x)}{\mathrm{erf}(\delta)}, \qquad \delta = \sqrt{.5\epsilon},$$

where erf denotes the error function (see [5] for a graph).

In Table 1 we give the results obtained with $\epsilon = 500$ and $N = 81$, increasing numbers P of poles and increasing α. Tables 2 and 3 display the same information for the approximation of f' and f'' with r' and r'' as given by the formulae (4.1). The combination of extra poles and a point shift brings about 7 digits of accuracy, where the point shift alone makes only for 2–3. The improvement in the derivatives is especially remarkable: the error in the second derivative decreases from the useless value of 9.26 to about 10^{-7}!

α	$P=0$	$P=2$	$P=4$	$P=6$	$P=8$
0.0	9.26	4.05e − 2	4.82e − 4	7.85e − 5	1.46e − 5
0.5	4.26	2.07e − 2	2.18e − 4	3.75e − 5	4.91e − 6
0.75	9.50e − 1	5.48e − 3	6.25e − 5	9.53e − 6	1.26e − 6
0.9	9.30e − 2	6.49e − 4	8.86e − 6	4.93e − 7	2.34e − 7
0.95	1.59e − 2	1.23e − 4	1.88e − 6	1.75e − 7	5.31e − 8
0.96	9.18e − 3	7.36e − 5	1.29e − 6	6.00e − 8	5.57e − 8

TAB. 3. Errors when approximating f'' with increasing P and α in Example 1.

Example 5.2 Example 3 in [5] has demonstrated that the attachment of poles may be very effective in improving the approximation of oscillatory functions. Here we change the function to

$$h(x) = e^{-ax^2} \sin bx, \qquad a > 0,\ b > 0,$$

so that the most oscillatory part lies in the center of the interval.

Results with $a = 5$, $b = 25$, $N = 31$, $P = 0$ and $P = 2$ are given in Table 4. In contrast with the preceding example, here the point shift brings much more improvement than the attachment of poles, about 6–7 digits, an especially heartening fact for the derivatives, to which the interpolants without shift are useless approximants.

Acknowledgement: The authors wish to thank Peter Graves–Morris for his comments which have enhanced the present text.

Bibliography

1. R. Baltensperger and J.–P. Berrut, The errors in calculating the pseudospectral differentiation matrices for Čebyšev-Gauss-Lobatto points, *Comput. Math. Applic.* **37** (1999), 41–48. Errata: **38** (1999), 119.
2. R. Baltensperger, J.–P. Berrut, and B. Noël, Exponential convergence of a linear rational interpolant between transformed Chebyshev points, *Math. Comp.* **68** (1999), 1109–1120.

α	h		h'		h''	
	$P=0$	$P=2$	$P=0$	$P=2$	$P=0$	$P=2$
0.0	4.12e − 2	2.49e − 3	2.03	1.36e − 1	1.43e + 3	9.51e + 1
0.5	1.66e − 2	8.68e − 4	8.90e − 1	6.08e − 2	5.63e + 2	3.84e + 1
0.75	1.97e − 3	7.95e − 5	1.17e − 1	7.73e − 3	5.98e + 1	3.98
0.9	1.91e − 5	4.20e − 7	1.09e − 3	4.56e − 5	3.97e − 1	1.68e − 2
0.92	4.57e − 6	7.78e − 8	2.48e − 4	8.20e − 6	8.24e − 2	2.75e − 3
0.94	6.56e − 7	7.18e − 9	3.26e − 5	5.69e − 7	9.56e − 3	1.66e − 4
0.96	3.03e − 8	2.39e − 9	1.81e − 6	5.49e − 7	4.71e − 4	1.62e − 4

TAB. 4. Change in the errors induced by the introduction of two poles in Example 2.

3. J.–P. Berrut and R. Baltensperger, The linear rational collocation method for boundary value problems, *BIT* **41** (2001), 868–879.
4. J.–P. Berrut and H. D. Mittelmann, Matrices for the direct determination of the barycentric weights of rational interpolation, *J. Comput. Appl. Math.* **78** (1997), 355–370.
5. J.–P. Berrut and H. D. Mittelmann, Rational interpolation through the optimal attachment of poles to the interpolating polynomial, *Numerical Algorithms* **23** (2000), 315–328.
6. J.–P. Berrut and H. D. Mittelmann, The linear rational collocation method with iteratively optimized poles for two-point boundary value problems, *SIAM J. Scient. Comput.* **23** (2001), 961–975.
7. K. S. Breuer and R. M. Everson, On the errors incurred calculating derivatives using Chebyshev polynomials, *J. Comput. Phys.* **99** (1992), 56–67.
8. A. Corana, M. Marchesi, C. Martini, and S. Ridella, Minimizing multimodal functions of continuous variables with the "Simulated Annealing" algorithm, *ACM Trans. Math. Software* **13** (1987) 262–280.
9. P. Henrici, *Essentials of Numerical Analysis*, Wiley, New-York, 1982.
10. P. Henrici, *Applied and Computational Complex Analysis Vol. 3*, Wiley, New York, 1986.
11. E. H. Kaufman Jr, D. J. Leeming, and G. D. Taylor, Uniform rational approximation by differential correction and Remes-differential correction, *Int. J. Numer. Meth. Engin.* **17** (1981), 1273–1278.
12. D. Kosloff and H. Tal-Ezer, A modified Chebyshev pseudospectral method with an $\mathcal{O}(N^{-1})$ time step restriction, *J. Comput. Phys.* **104** (1993), 457–469.
13. S. C. Reddy and L. N. Trefethen, Lax-stability of fully discrete spectral methods via stability regions and pseudo–eigenvalues, *Comput. Methods Appl. Mech. Engrg.* **80** (1990), 147–164.
14. C. Schneider and W. Werner, Some new aspects of rational interpolation, *Math. Comp.* **47** (1986) 285–299.

An application of a mathematical blood flow model

Michael Breuß, Andreas Meister

Department of Mathematics, University of Hamburg, Germany.
breuss@math.uni-hamburg.de, meister@math.uni-hamburg.de

Bernd Fischer

Mathematical Institute, Medical University of Lübeck, Germany.
fischer@math.mu-luebeck.de

Abstract

Mathematical models of blood flow are inevitably embedded in models of human thermoregulation because they take the role of the most significant heat distributor in models of the human thermal system [14, 6]. Models of human thermoregulation have a wide range of applications, e.g. for the prediction of the impact of accidents, diseases and clinical treatments (see [14] and the references therein). The application of our interest is the prediction of the influence of cooling on the heat distribution in premature infants, see Section 2. In Section 3 we discuss the requirements of a reliable thermoregulation model while the governing equation is described in paragraph four. The employed blood flow model is discussed within Section 5. Section 6 deals with numerical results, followed by concluding remarks in the last paragraph.

1 Motivation

Lack of oxygen of the fetus or newborn is known to be an important cause for injuries of the developing brain [9]. Experimental studies have shown that the neuronal loss evolves over several days after such an incident [8]. An important factor influencing the degree and distribution of neuronal loss is the cerebral temperature, i.e. lowering the cerebral temperature can prevent much damage [5].

The question arises, if it is possible to lower the cerebral temperature of an infant by $2-3\,K$ by the manipulation of the environment inside an incubator while the rest of the body maintains a pleasant temperature. The objective of this paper is to discuss the mathematical measurements which can be used to predict an answer to that question by the use of numerical simulations.

2 Modeling the thermoregulation of premature infants

The term thermoregulation stands for the measurements of the body to hold a pleasant temperature [4]. Models for thermoregulation consist of two parts: the active and the passive system [6]. The active system consists of the regulatory mechanisms shivering (heat production within the muscles attached to the skeleton), vasomotion (control over the degree of blood flow within the skin) and sweating (control over the degree of effectiveness of heat transfer between the infant and the surrounding air). The passive system

is the combination of the physical human body and the heat transfer in it and at its surface. The idea behind this distinction is that the active system has a controlling influence over the passive system. Naturally, only results obtained by the complete model can be compared with available real life data.

Concerning premature infants, it is known that shivering and sweating are not of importance for the modelling process [4, 13], while vasomotion should not be of great concern for our special application [13]. The modeling of the passive system demands the discretization of the body and the modeling of metabolic heat production and blood flow. We do not consider phenomena which are related to environmental conditions, namely the response to air convection, the probability to gain or loose heat due to radiation and heat loss due to evaporation in dependence on pressure, temperature and humidity of the surrounding air, assuming that these are controllable by the use of an incubator [13].

In order to give an answer to the defined question by use of numerical simulations, a model needs to deliver detailed temperature profiles within the head and a detailed resolution of the heat transfer processes in the body. It should be applicable to different size neonates whereby aspects like the anatomy and the thermal maturity have to be considered. With the exception of the blood flow model, these aspects can be defined via a suitable geometry and the use of real life data for spatially dependent rates of metabolic heat production within a numerical method [7, 2]. This also incorporates that existing numerical methods made for the simulation of thermoregulation of adults are of no use in the given context since studies have shown [3] that a detailed modeling of geometry and tissue composition is necessary in order to obtain relevant temperature profiles. As it can be shown experimentally [7, 2] in agreement to theoretical discussions concerning thermoregulation models of adults [6, 14], the use of a blood flow model greatly affects the computed numerical solutions.

3 Analysis of the blood flow model

The bio-heat equation derived by Pennes [10] forms the basis of the majority of models for human thermoregulation in use today [14, 6]. It describes the dissipation of heat in a homogeneous, infinite tissue volume. For two spatial dimensions, it can be written in the form

$$c(\mathbf{x})\rho(\mathbf{x})\partial_t T(\mathbf{x},t) = \mathrm{div}\big[\lambda(\mathbf{x})\nabla \mathrm{T}(\mathbf{x},\mathrm{t})\big] + \mathrm{f}(\mathbf{x},\mathrm{t}). \tag{3.1}$$

Thereby, the temperature T depends on the spatial variable $\mathbf{x} = (x_1, x_2)^T$ as well as on time t. Furthermore, $\lambda(\mathbf{x})$, $c(\mathbf{x})$ and $\rho(\mathbf{x})$ denote the heat conductivity, specific heat capacity and density of the tissue, respectively. The term $f(\mathbf{x})$ can be decomposed via $f(\mathbf{x},t) = Q_M(\mathbf{x}) + Q_B(\mathbf{x},t)$ into parts corresponding to metabolic heat production $Q_M(\mathbf{x})$ and blood flow $Q_B(\mathbf{x},t)$.

As already indicated, the term $Q_M(\mathbf{x})$ can be defined by the use of real life data [7]. The formulation of the source term due to blood flow is based on variations of the following procedure [6, 14]. The idea is that the body is supplied from a central pool of blood by the major arteries. Before the tissue is perfused, the temperature of the arterial blood mixes with the temperature of venous blood flowing in adjacent veins. After that, the arterial blood exchanges heat with the tissue in the capillaries and becomes venous

blood. The venous blood is collected in the major veins and its temperature mixes with the temperature of arterial blood in the adjacent arteries before it flows back into the blood pool.

Since equation (3.1) deals with the change of thermal energy per unit volume, the term $Q_B(\mathbf{x})$ takes the form

$$Q_B(\mathbf{x},t) = c_B \rho_B CCX(\mathbf{x}) BF(\mathbf{x}) \big[T_B(t) - T(\mathbf{x},t)\big], \tag{3.2}$$

whereby $T_B(t)$ denotes the time-dependent mean value of the temperature of the blood within the blood pool, we also assume that the specific density of the blood ρ_B and the specific heat capacity of the blood c_B are constant variables.

The described modeling results in a differential equation for the temporal evolution of the temperature within the blood pool, namely in

$$m_B c_B \partial_t T_B(t) = \int_D \rho_B c_B CCX(\mathbf{x}) BF(\mathbf{x})\, d\mathbf{x} \big[T_V(t) - T_B(t)\big]. \tag{3.3}$$

Thereby, the total blood mass m_B, the time dependent mean value of the temperature of the venous blood $T_V(t)$, and locally defined tissue-dependent measures for the blood perfusion $BF(\mathbf{x})$ and the counter-current heat exchange $CCX(\mathbf{x})$ are introduced.

Equation (3.3) shows that the temporal change of the blood pool temperature is proportional to the difference to the temperature of the venous blood. The outlined idea leads to the modeling of the temperature of the venous blood as

$$T_V(t) = \frac{\int_D CCX(\mathbf{x}) BF(\mathbf{x}) T(\mathbf{x},t)\, d\mathbf{x}}{\int_D CCX(\mathbf{x}) BF(\mathbf{x})\, d\mathbf{x}} \tag{3.4}$$

which is also usable when only steady states are considered [7]. The crucial terms in the order of importance are the blood perfusion $BF(\mathbf{x})$ and the counter current heat exchange $CCX(\mathbf{x})$.

There is much debate about the choice of these functions in literature [14, 6]. This debate arises because the representation of blood circulation is substituted by a rather simple model formulation. The cure to this disadvantage is generally sought by exploring more and more detailed models of microstructure, organs, etc., or it is sought by a better modeling of control mechanisms of the actice system in the case of adults [14, 6].

The main drawback of the described blood flow model is given by the blood pool idea itself. This is up to now to our knowledge not outlined in any mathematical description of this model within the literature and can be illustrated as follows. Let a detailed geometry be given with a stationary temperature distribution together with a homogeneous neutral temperature at the whole boundary as initial state. Let us assume that we start a numerical computation where a selective cooling at the neck is employed. By heat conduction of the tissue, the effect of cooling computed with the help of the discretization of heat gradient and heat conductivity of the local tissue propagates into the inner part of the domain. Concerning the blood flow, the averaging step within (3.4) captures the local cooling effect which results in a slightly cooler average temperature of the venous blood within the whole domain than in the initial state. Employing this value in (3.3) results in a slight negative change of the blood pool temperature. Taking account of the

evaluation of the source term (3.2) for the control volumes located in the vicinity of the neck, we notice that a strong cooling is locally equalized by the combination of a) the source term due to blood flow which is mostly influenced by the neutral blood temperature in the rest of the body and b) of the source term due to metabolic heat production which was not influenced at all by the change in the boundary temperature. The result is that the effect of a local cooling mechanism is instantly distributed over the whole domain while a weighted mean value of the temperature over the domain equalizes local cooling mechanisms. The validity of this reasoning is verified by numerical results [7, 2] and by an exemplary result shown in Section 6.

The non-local nature of the described blood flow model can directly be seen by applying an implicit time stepping strategy. Due to the integration over the whole computational domain in (3.4), one ends up with a fully occupied matrix after the usual linearization step which was already recognized in [7] in the context of steady state calculations.

We now illuminate a further property of the bloodflow model. Therefore, let the abbreviations $\alpha = \rho_B c_B$, $\beta = \int_D K_B(\mathbf{x})B(\mathbf{x})\,d\mathbf{x}$ and $\gamma = \rho_B/m_B$ hold. A straightforward computation gives

$$T_B(t) = T_V(t) - \frac{1}{\gamma\beta}\frac{d}{dt}T_B(t). \tag{3.5}$$

Note that α, β and γ are positive constants. Consider a steady state situation as initial state, i.e. $T_B = T_V$ holds. If the body is heated, the temperature within the body increases and so T_V will increase. This has the effect that the bloodpool temperature T_B will increase in the near future, i.e. $T_B'(t) > 0$. We now investigate the net effect of the bloodflow. Integration of the source over the computational domain D results in

$$\int_D Q_B(\mathbf{x},t)\,d\mathbf{x} = \alpha\left[\beta T_B(t) - \int_D K_B(\mathbf{x})B(\mathbf{x})T(\mathbf{x},t)\,d\mathbf{x}\right] \overset{(3.5)}{=} -\frac{\alpha}{\gamma}\frac{d}{dt}T_B(t).$$

When employing $T_B'(t) > 0$ we see that the total of all sources in the body is negative, i.e. while the blood in the bloodpool cools the increasingly warm body in the mean if the body is exposed to heat, it also takes over heat from it. The bloodpool and the body are to be seen as two separate systems which are connected via heat fluxes and so one can consider the bloodpool as a regulator.

4 Numerical method and experiments

The following numerical approximation of the unsteady bio-heat equation (3.1) represents a convenient extension of the finite volume method developed in [7], which has been proven to be a robust, accurate and reliable algorithm in the context of steady state temperature distributions. However, finite volume schemes are categorically based on the integral form of the governing equation. In order to apply Gauss's integral theorem it is neccessary to write the equation in divergence form. Therefore, we introduce the auxiliary variable $k(\mathbf{x}) = \rho(\mathbf{x})c(\mathbf{x})$ and the auxiliary temperature $\overline{T}(\mathbf{x},t) = k(\mathbf{x})T(\mathbf{x},t)$

into the governing equation and consequently the bio-heat equation (3.1) writes

$$\frac{d}{dt}\int_\sigma \overline{T}(\mathbf{x},t)\,d\mathbf{x} = \int_{\partial\sigma}\left[\frac{\lambda(\mathbf{x})}{k(\mathbf{x})}\nabla\overline{T}(\mathbf{x},t) - \frac{\lambda(\mathbf{x})\overline{T}(\mathbf{x},t)}{k(\mathbf{x})^2}\nabla k(\mathbf{x})\right]\cdot\mathbf{n}(\mathbf{x})\,\mathbf{ds} + \int_\sigma \mathbf{f}(\mathbf{x},\mathbf{t})\,\mathbf{dx} \tag{4.1}$$

for all control volumes $\sigma \subset D$, see [2]. In order to solve equation (4.1) numerically, the space part $\overline{D}$ is decomposed into a finite number of sub-domains. We start from an

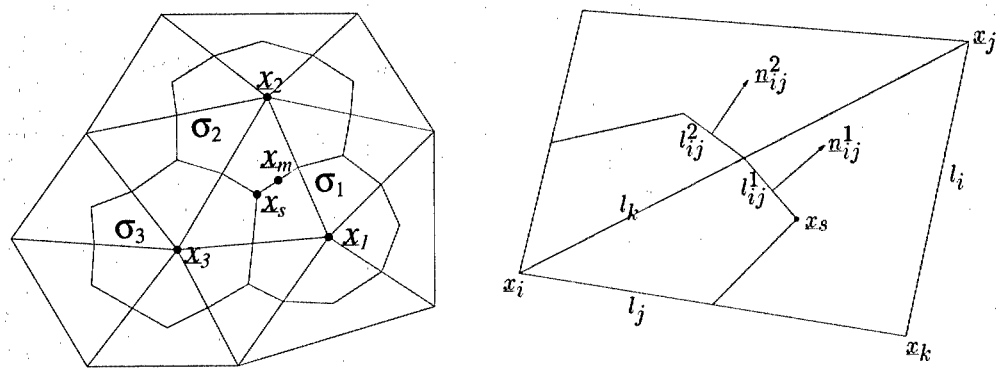


FIG. 1. General form of a control volume of the triangulation (left) and its boundary (right).

arbitrary conforming triangulation $\mathcal{D}^h$ of the domain $\overline{D}$ which is called the primary mesh and consisting of finitely many triangles $\mathcal{D}_i$ and the corresponding nodes are abbreviated by $\mathbf{x}_i \in \overline{D}$. Based on the triangulation a discrete control volume σ_i is defined as the open set of $\mathbf{R^2}$ including the node $\mathbf{x}_i$ and bounded by straight lines which are determined by the connection of the midpoints of the edges of the corresponding triangles $\mathcal{D}_j$ (i.e. $\mathbf{x}_i \in \partial\mathcal{D}_j$) and their barycentre (see Figure 1). The union $\mathcal{B}^h$ of all boxes is called the secondary mesh. A finite volume method represents a discretization of the evolutionary equation (4.1) for cell averages defined by $\left(\mathcal{M}\overline{T}\right)(t)\big|_\sigma = (1/|\sigma|)\int_\sigma \overline{T}(\mathbf{x},t)\,d\mathbf{x}$, where $|\sigma|$ denotes the volume of the box σ. With respect to the secondary mesh $\mathcal{B}^h$ we can write the integral form (4.1) as

$$\begin{aligned}\frac{d}{dt}\left(\mathcal{M}\overline{T}\right)(t)\big|_{\sigma_i} &= \frac{1}{|\sigma_i|}\left\{\int_{\partial\sigma_i}\left[\frac{\lambda(\mathbf{x})}{k(\mathbf{x})}\nabla\overline{T}(\mathbf{x},t) - \frac{\lambda(\mathbf{x})\overline{T}(\mathbf{x},t)}{k(\mathbf{x})^2}\nabla k(\mathbf{x})\right]\cdot\mathbf{n}(\mathbf{x})\,\mathbf{ds}\right.\\ &\left. + \int_{\sigma_i} Q_B(\mathbf{x},t)\,d\mathbf{x} + \int_{\sigma_i} Q_M(\mathbf{x})\,d\mathbf{x}\right\},\ \forall\sigma_i \in \mathcal{B}^h.\end{aligned} \tag{4.2}$$

Corresponding to a finite element method the evaluation of the boundary integral is performed by using a piecewise constant distribution of the heat coefficient λ and a piecewise linear distribution of the auxilary temperature $\overline{T}$. with respect to the triangles of the trianglutaion used. Note that the source term remains unchanged and the calculation is

given by

$$\int_{\sigma_i} Q_M(\mathbf{x})\, d\mathbf{x} = |\sigma_i| Q_M(\mathbf{x}_i)$$

and

$$\int_{\sigma_i} Q_B(\mathbf{x}, t)\, d\mathbf{x} = |\sigma_i| c_B \rho_B CCX(\mathbf{x}_i) BF(\mathbf{x}_i) \left[T_B(t) - T(\mathbf{x}_i, t)\right].$$

The computation of the blood pool temperature is directly performed by an explicit time discretization of equation (3.3). Thereby, the temperature of the venous blood is given by equation (3.4).

It is remarkable that the method degenerates to the scheme presented in [7] in the context of a steady state solution and therefore the excellent properties like the discrete min-max principle are maintained in such a situation. Due to the space available

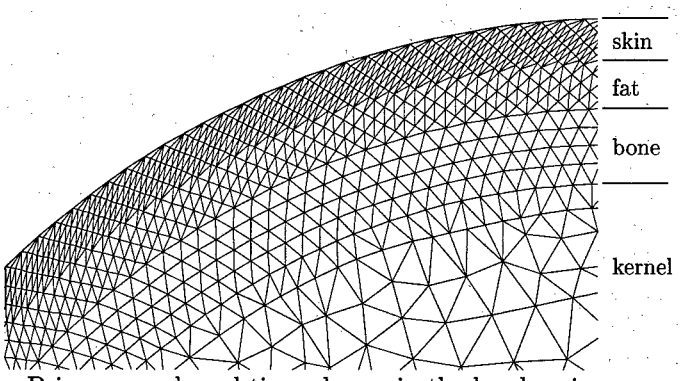


FIG. 2. Primary mesh and tissue layers in the head region.

we restrict ourself to the consideration of steady state calculations using the described method. Thereby, we distinguish layers of skin, fat, bone and kernel by different rates of metabolism, specific heat capacity and blood perfusion associated with the regions depicted in Figure 2. As boundary conditions we employ a comfortable boundary temperature of $309.15\,K$ at head, back, legs, and belly while we set $299.15\,K$ at the neck, i.e. we selectively cool the neck. In reality, this corresponds to the situation where the infant is wearing a water-filled collar with the purpose of cooling the blood flowing into the brain through the arteries adjacent to the skin.

In Figure 3 (a) we can see the temperature distribution in the two-dimensional discretized idealization of the body of a premature infant. Thereby, no blood flow and no metabolic heat production is applied, so that the depicted distribution of heat is only influenced by the heat conductivity of the employed tissues. The situation where tissue dependent metabolic heat production is taken into account is shown in Figure 3 (b). Note that the heat sources visualized within the picture not only have local effects, they also influence the mean value of the temperature of the blood pool. Within Figure 3 (c), blood flow is additionally given.

It is evident that the blood flow has the effect outlined in Section 5. Especially, the numerical solution incorporates no hint of the fact, that in reality there is a transport of cool blood to the brain and also a transport of blood by the veins coming from the brain.

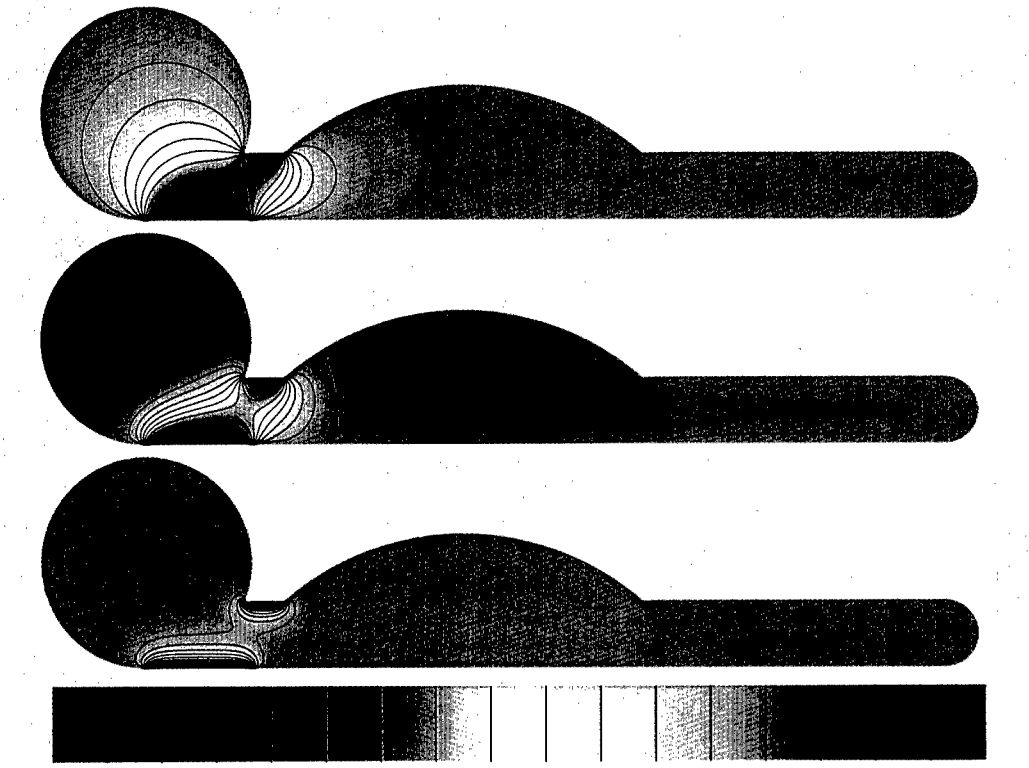


FIG. 3. Comparison of steady state situations (a) only with heat conduction (b) with heat conduction and metabolic heat production and (c) with blood flow additionally taken into account (from top to bottom).

5 Concluding remarks

The range of applicability of the described blood flow model is restricted to situations where it makes sense to employ a mean value of the whole blood, e.g. if the whole body is exposed for a longer time to the same temperature. For a clinical application where the effects of local cooling or heating have to be studied, caution is required when dealing with the results achieved by employing variations of the described model.

Bibliography

1. B. Fischer, M. Breuß and A. Meister, The unsteady thermoregulation of premature infants — a model and its application, in *Discrete Modelling and Discrete Algorithms in Continuum Mechanics*, Proceedings of the GAMM Workshop, Th. Sonar and I. Thomas (eds.), 2000.
2. B. Fischer, M. Breuß and A. Meister, The numerical simulation of unsteady heat conduction in a premature infant, in *Numerical Methods for Fluid Dynamics, M.J. Baines (editor), ICFD, Oxford University Computing Laboratory* **7** (2001).
3. M. Buse and J. Werner, Heat balance of the human body: influence of variations of locally distributed parameters, in *Journal of Theoretical Biology* **114** (1985), 34–51.
4. O. Bußmann, A model for the thermoregulation of premature infants and neonates under consideration of the thermal maturity, *PhD Thesis*, Medical University of Lübeck, (2000), in German.
5. R. Busto et al., The importance of brain temperature in cerebral ischemic injury, in *Stroke* **20** (1989), 1114–1134.
6. D. Fiala, K.J. Lomas and M. Stohrer, A computer model of human thermoregulation for a wide range of environmental conditions: the passive system, in *Journal of Applied Physiology* **87** No. 5 (1999), 1957–1972.
7. B. Fischer, M. Ludwig and A. Meister, The thermoregulation of infants: Modeling and numerical simulation, in *BIT* **41** No. 5 (2001), 950–966.
8. P.D. Gluckman and C.E. Williams, When and why do brain cells die?, in *Dev. Med. Child Neurol.* **34** (1992), 1010–1014.
9. E.C. Mallard et al., Neuronal damage in the developing brain following intrauterine asphyxia, in *Reprod. Fertil. Dev.* **7** (1995), 647–653.
10. H.H. Pennes, Analysis of Tissue and Arterial Blood Temperatures in the Resting Human Forearm, in *Journal of Applied Physiology* **1** (1948), 93–122.
11. G. Simbruner, *Thermodynamic Models for Diagnostic Purposes in the Newborn and Fetus*, Facultas Verlag, Wien, 1983.
12. T. Sonar, On the Construction of Essentially Non-Oscillatory Finite Volume Approximations to Hyperbolic Conservation Laws on General Triangulations: Polynomial Recovery, Accuracy, and Stencil Selection, in *Comp. Meth. Appl. Mech. Eng.* **140** (1997), 157–181.
13. K. Thomas, Back to Basics: Thermoregulation in Neonates, in *Neonatal Network* **13** No. 2 (1994), 15–22.
14. J. Werner, Thermoregulatory models. Recent research, current applications and future development, in *Scand. J. Work Environ. Health* **15** Suppl. 1 (1989), 34–46.
15. J. Werner and P. Webb, A six-cylinder model of human thermoregulation for general use on personal computers, in *Ann. Physiol. Anthrop.* **12** No. 3 (1993), 123–134.
16. E.H. Wissler, A mathematical model of the human thermal system, in *Bulletin of the human thermal system* **26** (1964), 147–166.

Zeros of the hypergeometric polynomial $F(-n, b;\ c;\ z)$

K. Driver* and K. Jordaan

School of Mathematics, University of the Witwatersrand, Johannesburg, South Africa.
036kad@cosmos.wits.ac.za, 036jord@cosmos.wits.ac.za

Abstract

Our interest lies in describing the zero behaviour of Gauss hypergeometric polynomials $F(-n, b;\ c;\ z)$ where b and c are arbitrary parameters. In general, this problem has not been solved and even when b and c are both real, the only cases that have been fully analysed impose additional restrictions on b and c. We review recent results that have been proved for the zeros of several classes of hypergeometric polynomials $F(-n, b;\ c;\ z)$ where b and c are real. We show that the number of real zeros of $F(-n, b;\ c;\ z)$ for arbitrary real values of the parameters b and c, as well as the intervals in which these zeros (if any) lie, can be deduced from corresponding results for Jacobi polynomials.

1 Introduction

The Gauss hypergeometric function, or ${}_2F_1$, is defined by

$$F(a, b;\ c;\ z) = 1 + \sum_{k=1}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{z^k}{k!}, \quad |z| < 1,$$

where a, b and c are complex parameters and

$$(\alpha)_k = \alpha(\alpha + 1) \ldots (\alpha + k - 1) = \Gamma(\alpha + k)/\Gamma(\alpha)$$

is Pochhammer's symbol. When $a = -n$ is a negative integer, the series terminates and reduces to a polynomial of degree n, called a hypergeometric polynomial. Our focus lies in the location of the zeros $F(-n, b;\ c;\ z)$ for real values of b and c.

Hypergeometric polynomials are connected with several different types of orthogonal polynomials, notably Chebyshev, Legendre, Gegenbauer and Jacobi polynomials. In the cases of Chebyshev and Legendre polynomials, the connection demands fixed special values of the parameters b and c, namely, (cf. [1], p.561)

$$F\left(-n, n;\ \frac{1}{2};\ z\right) = T_n(1 - 2z)$$

and

$$F\left(-n, n + 1;\ 1;\ z\right) = P_n(1 - 2z),$$

*Research of the first author is supported by the John Knopfmacher Centre for Applicable Analysis and Number Theory, University of the Witwatersrand.

respectively. However, in the cases of Gegenbauer and Jacobi polynomials, we have

$$F\left(-n, n+2\lambda;\, \lambda+\frac{1}{2};\, z\right) = \frac{n!}{(2\lambda)_n} C_n^{\lambda}(1-2z) \tag{1.1}$$

and

$$F\left(-n, \alpha+\beta+1+n;\, \alpha+1;\, z\right) = \frac{n!}{(\alpha+1)_n} \mathcal{P}_n^{(\alpha,\beta)}(1-2z), \tag{1.2}$$

respectively. Since the zeros of orthogonal polynomials are well understood, we expect the connections (1.1) and (1.2) to be very useful in analysing the zeros of $F(-n, b;\, c;\, z)$. Conversely, if the zeros of $F(-n, b;\, c;\, z)$ are known, this leads to new information about the zero distribution of Gegenbauer or Jacobi polynomials for values of their parameters that lie outside the range of orthogonality of these polynomials.

This paper is organized as follows. In Section 2 we give a self-contained review of recent results regarding the zeros of several special classes of hypergeometric polynomials. Section 3 contains results originally due to Klein [9] which detail the numbers and location of real zeros of $F(-n, b;\, c;\, z)$ for arbitrary real values of b and c. We provide simple proofs using results proved in [13].

2 Zeros of special classes of hypergeometric polynomials

We begin with a few general remarks. Since we shall assume throughout our discussion that b and c are real parameters, we know that all zeros of $F(-n, b;\, c;\, z)$ must occur in complex conjugate pairs. In particular, if n is odd, F must always have at least one real zero. Further, if $b = -m$ where $m < n$, $m \in \mathbb{N}$, $F(-n, b;\, c;\, z)$ reduces to a polynomial of degree m. However, since we are interested in the behaviour of the zeros of $F(-n, b;\, c;\, z)$ as b and/or c vary through real values, we shall adopt the convention that $F(-n, -m;\, c;\, z) = \lim_{b\to -m} F(-n, b;\, c;\, z)$. This ensures that the zeros of F vary continuously with b and c. Note also that $F(-n, b;\, c;\, z)$ is not defined when $c = 0, -1, \ldots, -n+1$. Regarding the multiplicity of zeros, a hypergeometric function $w = F(a, b;\, c;\, z)$ satisfies the differential equation

$$z(1-z)w'' + \left[c - (a+b+1)z\right]w' - abw = 0,$$

so if $w(z_0) = w'(z_0) = 0$ at some point $z_0 \neq 0$ or 1, it would follow that $w \equiv 0$. Thus multiple zeros of $F(-n, b;\, c;\, z)$ can only occur at $z = 0$ or 1.

2.1 Quadratic transformations

The class of hypergeometric polynomials that admit a quadratic transformation is specified by a necessary and sufficient condition due to Kummer (cf. [1], p.560). There are

twelve polynomials in this class (cf. [14], p.124)

$$F(-n,b;\,2b;\,z) \qquad F(-n,b;\,-n-b+1;\,z) \qquad F\left(-n,b;\,\tfrac{-n+b+1}{2};\,z\right)$$

$$F\left(-n,b;\,\tfrac{1}{2};\,z\right) \qquad F\left(-n,-n+\tfrac{1}{2};\,c;\,z\right) \qquad F\left(-n,b;\,-n+b+\tfrac{1}{2};\,z\right)$$

$$F\left(-n,b;\,\tfrac{3}{2};\,z\right) \qquad F\left(-n,-n-\tfrac{1}{2};\,c;\,z\right) \qquad F\left(-n,b;\,-n+b-\tfrac{1}{2};\,z\right)$$

$$F(-n,b;\,-2n;\,z) \qquad F(-n,b;\,b+n+1;\,z) \qquad F(-n,n+1;\,c;\,z)\,.$$

The most important polynomial in this class is $F(-n,b;\,2b;\,z)$ because complete analysis of its zero distribution for all real values of b (cf. [4], [5]) leads to corresponding results for the zeros of the Gegenbauer polynomials $C_n^{\lambda}(z)$ for all real values of the parameter λ (cf. [6]).

Theorem 2.1. *Let $F = F(-n,b;\,2b;\,z)$ where b is real.*

(i) For $b > -\frac{1}{2}$, all zeros of $F(-n,b;\,2b;\,z)$ are simple and lie on the circle $|z-1| = 1$.

(ii) For $-\frac{1}{2} - j < b < \frac{1}{2} - j$, $j = 1,2,\ldots\left[\frac{n}{2}\right] - 1$, $(n-2j)$ zeros of F lie on the circle $|z-1| = 1$. If $j = 2k$ is even, there are k non-real zeros of F in each of the four regions bounded by the circle $|z-1| = 1$ and the real axis. If $j = 2k+1$ is odd, there are k non-real zeros of F in each of the four regions described above and the remaining two zeros are real.

(iii) If n is even, for $-\left[\frac{n}{2}\right] < b < -\left[\frac{n}{2}\right] + \frac{1}{2}$, no zeros of F lie on $|z-1| = 1$. If $n = 4k$, all zeros of F are non-real whereas if $n = 4k+2$, two zeros of F are real and $4k$ are non-real. If n is odd, for $-1 - \left[\frac{n}{2}\right] < b < -\left[\frac{n}{2}\right] + \frac{1}{2}$, only the fixed real zero of F at $z = 2$ lies on $|z-1| = 1$. If $n = 4k+1$, $n-1 = 4k$ zeros of F are non-real whereas if $n = 4k+3$, two further zeros are real and the remaining $4k$ are non-real.

(iv) For $j - n < b < j - n + 1$, $j = 1,2,\ldots\left[\frac{n}{2}\right] - 1$, $(n-2j)$ zeros of F are real and greater than 1. If $j = 2k$ is even, all remaining $2j$ zeros of F are non-real with k zeros in each of the regions described above; while if $j = 2k+1$, $4k$ zeros are non-real as before and 2 are real.

(v) For $b < 1-n$, all zeros of $F(-n,b;\,2b;\,z)$ are real and greater than 1. As $b \to -\infty$, all the zeros of F converge to the point $z = 2$.

An analogous theorem which describes the behaviour of the zeros of $C_n^{\lambda}(z)$ can be found in [6], Section 3 or [7], Theorem 1.2.

For the polynomial $F\left(-n,b;\,\frac{1}{2};\,z\right)$ the following result has been proved in [7], Theorem 2.3.

Theorem 2.2. *Let $F = F\left(-n,b;\,\frac{1}{2};\,z\right)$ with b real.*

(i) For $b > n - \frac{1}{2}$, all n zeros of F are real and simple and lie in $(0,1)$.

(ii) For $n - \frac{1}{2} - j < b < n + \frac{1}{2} - j$, $j = 1,2,\ldots,n-1$, $(n-j)$ zeros of F lie in $(0,1)$ and the remaining j zeros of F form $\left[\frac{j}{2}\right]$ non-real complex pairs of zeros and one real zero lying in $(1,\infty)$ when j is odd.

(iii) *For* $0 < b < \frac{1}{2}$, F *has* $\left[\frac{n}{2}\right]$ *non-real complex conjugate pairs of zeros with one real zero in* $(1, \infty)$ *when* n *is odd.*

(iv) *For* $-j < b < -j+1$, $j = 1, 2, \ldots, n-1$, F *has exactly* j *real negative zeros. There is exactly one further real zero greater than 1 only when* $(n-j)$ *is odd and all the remaining zeros of* F *are non-real.*

(v) *For* $b < 1-n$, *all zeros of* F *are real and negative and converge to zero as* $b \to -\infty$.

A very similar theorem is proved for the zeros of $F\left(-n, b;\ \frac{3}{2};\ z\right)$ in [7], Theorem 2.4 with only minor differences of detail.

For the hypergeometric polynomial $F(-n, b;\ -2n;\ z)$, less complete results have been proved. We have (cf. [8] Theorem 3.1 and Corollary 3.2) the following.

Theorem 2.3. *Let* $F = F\left(-n, b;\ -2n;\ z\right)$ *with* b *real.*

(i) *For* $b > 0$, F *has* n *non-real zeros if* n *is even whereas if* n *is odd,* F *has exactly one real negative zero and the remaining* $(n-1)$ *zeros of* F *are all non-real.*

(ii) *For* $-n < b < 0$, *if* $-k < b < -k+1$, $k = 1, \ldots, n$, F *has* k *real zeros in the interval* $(1, \infty)$. *In addition, if* $(n-k)$ *is even,* F *has* $(n-k)$ *non-real zeros whereas if* $(n-k)$ *is odd,* F *has one real negative zero and* $(n-k-1)$ *non-real zeros.*

(iii) *For* $-n > b > -2n$, *if* $-n-k > b > -n-k-1$, $k = 0, 1, \ldots, n-1$, F *has* $(n-k)$ *real zeros in the interval* $(1, \infty)$. *In addition, if* k *is even* F *has* k *non-real zeros while if* k *is odd,* F *has one real zero in* $(0, 1)$ *and* $(k-1)$ *non-real zeros.*

(iv) *For* $b < -2n$, *all* n *zeros of* F *are non-real for* n *even whereas for* n *odd,* F *has exactly one real zero in the interval* $(0, 1)$.

The identities (cf. [7], Lemma 2.1)

$$F(-n, b;\ c;\ 1-z) = \frac{(c-b)_n}{(c)_n} F(-n, b;\ 1-n+b-c;\ z) \tag{2.1}$$

and

$$F(-n, b;\ c;\ z) = \frac{(b)_n}{(c)_n}(-z)^n F\left(-n, 1-c-n;\ 1-b-n;\ \frac{1}{z}\right) \tag{2.2}$$

hold for b and c real, $c \neq \{0, -1, \ldots, -n+1\}$. Applying (2.1) and (2.2) to each of the polynomials $F(-n, b;\ 2b;\ z)$, $F\left(-n, b;\ \frac{1}{2};\ z\right)$, $F\left(-n, b;\ \frac{3}{2};\ z\right)$ and $F(-n, b;\ -2n;\ z)$ in turn, we obtain the remaining eight polynomials in the quadratic class. It is then an easy task to deduce analogous results for their zero distribution.

A similar set of results has been proved for the sixteen hypergeometric polynomials in the cubic class. Again, this class arises from a necessary and sufficient condition (cf. [2], p.67) and details can be found in [7].

3 The real zeros of $F(-n, b;\ c;\ z)$ for b and c real

The results proved below are due to Klein [9] who considered the zeros of more general hypergeometric functions (not necessarily polynomials). Klein's proof is geometric and

difficult to penetrate. A more transparent perspective in the polynomial case may be provided by the approach given here.

The classical equation linking the hypergeometric polynomial $F(-n,b;\,c;\,z)$ with Jacobi polynomials $\mathcal{P}_n^{(\alpha,\beta)}(z)$ is given by (1.2). We will find an alternative expression (cf. [12], p.464, eqn. (142))

$$F(-n,b;\,c;\,z)=\frac{n!z^n}{(c)_n}\mathcal{P}_n^{(\alpha,\beta)}\left(1-\frac{2}{z}\right), \tag{3.1}$$

where $\alpha=-n-b$ and $\beta=b-c-n$, more suited to our analysis. The number of real zeros of $\mathcal{P}_n^{(\alpha,\beta)}(x)$ in the intervals $(-1,1)$, $(-\infty,1)$ and $(1,\infty)$ are given by the Hilbert-Klein formulas (cf. [13], p.145, Theorem 6.72), also known to Stieltjes. We use Klein's symbol

$$E(u)=\begin{cases}0 & \text{if } u\le 0\\ [u] & \text{if } u>0,\ u\neq \text{integer}\\ u-1 & \text{if } u=1,2,3,\ldots\end{cases}.$$

Noting that under the linear fractional transformation $w=1-2/z$, the intervals $1<w<\infty$, $-\infty<w<-1$ and $-1<w<1$ correspond to $-\infty<z<0$, $0<z<1$ and $1<z<\infty$ respectively, we can use equation (3.1) to rephrase the Hilbert-Klein formulas for hypergeometric polynomials.

Theorem 3.1. *Let $b,\,c\in\mathbb{R}$ with $b,c,c-b\neq 0,\,-1,\ldots,-n+1$. Let*

$$X = E\left\{\frac{1}{2}\left(|1-c|-|n+b|-|b-c-n|+1\right)\right\} \tag{3.2}$$

$$Y = E\left\{\frac{1}{2}\left(-|1-c|+|n+b|-|b-c-n|+1\right)\right\} \tag{3.3}$$

$$Z = E\left\{\frac{1}{2}\left(-|1-c|-|n+b|+|b-c-n|+1\right)\right\}. \tag{3.4}$$

Then the numbers of zeros of $F(-n,b;\,c;\,z)$ in the intervals $(1,\infty)$, $(0,1)$ and $(-\infty,0)$ respectively are

$$N_1=\begin{cases}2[(X+1)/2] & \text{if } (-1)^n\binom{-b}{n}\binom{b-c}{n}>0\\ 2[X/2]+1 & \text{if } (-1)^n\binom{-b}{n}\binom{b-c}{n}<0\end{cases} \tag{3.5}$$

$$N_2=\begin{cases}2[(Y+1)/2] & \text{if } \binom{-c}{n}\binom{b-c}{n}>0\\ 2[Y/2]+1 & \text{if } \binom{-c}{n}\binom{b-c}{n}<0\end{cases} \tag{3.6}$$

$$N_3=\begin{cases}2[(Z+1)/2] & \text{if } \binom{-c}{n}\binom{-b}{n}>0\\ 2[Z/2]+1 & \text{if } \binom{-c}{n}\binom{-b}{n}<0.\end{cases} \tag{3.7}$$

Proof: The expressions all follow immediately from the Hilbert-Klein formulas (cf. [13], p.145, Thm. 6.72) together with equation (3.1). □

Theorem 3.2. *Let* $F = F(-n, b;\, c;\, z)$ *where* $b,\, c \in \mathbb{R}$ *and* $c > 0$.

(i) For $b > c + n$, *all zeros of* F *are real and lie in the interval* $(0, 1)$.

(ii) For $c < b < c + n$, $c + j - 1 < b < c + j$, $j = 1, 2, \ldots, n$; F *has* j *real zeros in* $(0, 1)$. *The remaining* $(n - j)$ *zeros of* F *are all non-real if* $(n - j)$ *is even while if* $(n - j)$ *is odd,* F *has* $(n - j - 1)$ *non-real zeros and one additional real zero in* $(1, \infty)$.

(iii) For $0 < b < c$, *all the zeros of* F *are non-real if* n *is even, while if* n *is odd,* F *has one real zero in* $(1, \infty)$ *and the other* $(n - 1)$ *zeros are non-real.*

(iv) For $-n < b < 0$, $-j < b < -j + 1$, $j = 1, 2, \ldots, n$, F *has* j *real negative zeros. The remaining* $(n - j)$ *zeros of* F *are all non-real if* $(n - j)$ *is even, while if* $(n - j)$ *is odd,* F *has* $(n - j - 1)$ *non-real zeros and one additional real zero in* $(1, \infty)$.

(v) For $b < -n$, *all zeros of* F *are real and negative.*

Proof: We use the identity (cf. [1], p.559, (15.3.4))

$$F(-n, b;\, c;\, z) = (1 - z)^n F\left(-n, c - b;\, c;\, \frac{z}{z - 1}\right) \tag{3.8}$$

to show that (i) ⇒ (v) and (ii) ⇒ (iv) so that it will suffice to prove (i), (ii) and (iii) above.

(i) ⇒ (v): If $b < -n$ then $c - b > c + n$ and by (i), all zeros of $F(-n, c - b;\, c;\, w)$ are real and lie in the interval $(0, 1)$. Since $w = z/(z - 1)$ maps $(-\infty, 0)$ to $(0, 1)$, (v) follows from (3.8).

(ii) ⇒ (iv): If $-j < b < -j + 1$, $j = 1, 2, \ldots, n$, then $c + j - 1 < c - b < c + j$, $j = 1, 2, \ldots, n$. By (ii), since $w = z/(z - 1)$ maps $(-\infty, 0)$ to $(0, 1)$ and $(1, \infty)$ to $(1, \infty)$, (iv) follows again from (3.8). To prove (i), (ii) and (iii), we note that in each part, $b > 0$ (and of course $c > 0$ by assumption). Then

$$\operatorname{sign}\binom{-b}{n} = (-1)^n, \quad \operatorname{sign}\binom{-c}{n} = (-1)^n. \tag{3.9}$$

(i) Suppose $b > c + n$. Then $b - c > n$ and

$$\operatorname{sign}\binom{b - c}{n} > 0 \text{ for all } n. \tag{3.10}$$

Considering (3.5), (3.6) and (3.7) with (3.9) and (3.10), we observe that

$$N_1 = 2\,[(X + 1)/2], \quad N_3 = 2\,[(Z + 1)/2],$$

$$N_2 = \begin{cases} 2\,[(Y + 1)/2] & \text{for } n \text{ even} \\ 2\,[Y/2] + 1 & \text{for } n \text{ odd} \end{cases}.$$

Assume now that $c > 1$. Then for $b > c + n$, we have from (3.2), (3.3) and (3.4) that $X = 0$, $Y = n$, $Z = 0$. Substituting these values into N_1, N_2 and N_3 yields the result. A similar calculation shows that the same result is obtained when $0 < c < 1$.

(ii) For $c+j-1<b<c+j$, $j=1,2,\ldots,n$, we find that sign $\binom{b-c}{n}=(-1)^{n-j}$. Then from (3.5), (3.6), (3.7) we see that

$$N_1 = \begin{cases} 2[(X+1)/2] & \text{for } (n-j) \text{ even} \\ 2[X/2]+1 & \text{for } (n-j) \text{ odd} \end{cases},$$

$$N_2 = \begin{cases} 2[(Y+1)/2] & \text{for } j \text{ even} \\ 2[Y/2]+1 & \text{for } j \text{ odd} \end{cases},$$

$$N_3 = 2[(Z+1)/2].$$

It follows from (3.2), (3.3) and (3.4) by an easy calculation that $X=0$, $Y=j$, $Z=0$ and we deduce that $N_1 = \begin{cases} 0 & \text{if } (n-j) \text{ is even} \\ 1 & \text{if } (n-j) \text{ is odd} \end{cases}$, $N_2=j$ and $N_3=0$ which proves (ii).

(iii) For $0<b<c$, sign $\binom{b-c}{n}=(-1)^n$. Then $N_1 = \begin{cases} 2[(X+1)/2] & \text{if } n \text{ is even} \\ 2[X/2]+1 & \text{if } n \text{ is odd} \end{cases}$, $N_2=2[(Y+1)/2]$, $N_3=2[(Z+1)/2]$. Also, we find $X=0$, $Y=0$ and $Z=0$ which completes the proof of (iii) and hence the theorem. □

For $c<0$, the range of values of b and c that have to be considered can be reduced if we use the identities (2.1) and (2.2). Since the real zeros of $F(-n,b;c;z)$ are now known for all $c>0$ and $b\in\mathbb{R}$ from Theorem 3.2, it follows from (2.1) that we need only consider $c-b>1-n$. Similarly, from (2.2) and Theorem 3.2, we can assume $b>1-n$. We split the result for $c<0$ into the cases where $b>0$ and $1-n<b<0$.

Theorem 3.3. *Let $F=F(-n,b;c;z)$. Suppose that $c<0$, $b>0$, $c-b>1-n$. Then*

(i) $1-n<c-b<0$ and $0<b<n-1$ and $1-n<c<0$.

(ii) If $-k<c<-k+1$, $k=1,\ldots,n-1$ and

$$-j<c-b<-j+1, \quad j=1,\ldots,n-1,$$

then $F(-n,b;c;z)$ has $(j-k)\geq 0$ real zeros in $(0,1)$. For the remaining $(n-j+k)$ zeros of F

(a) $(n-j+k)$ are non-real if $(n-j)$ and k are even

(b) $(n-j+k-1)$ are non-real and one real zero lies in $(1,\infty)$ if $(n-j)$ is odd and k is even

(c) $(n-j+k-1)$ are non-real if $(n-j)$ is even, k odd and one zero is real and negative

(d) $(n-j+k-2)$ are non-real if $(n-j)$ is odd and k is odd with one real negative zero and one real zero in $(1,\infty)$.

Proof: (i) This follows immediately from $c<0$, $b>0$, $c-b>1-n$.

(ii) For $c < 0$, $b > 0$, $c - b > 1 - n$, we have

$$|1 - c| = 1 - c, \quad |b + n| = b + n, \quad |b - c - n| = c - b + n$$

and it follows from (3.2), (3.3) and (3.4) that

$$X = E(1 - c - n), \quad Y = E(b), \quad Z = E(c - b).$$

Since $1 - c - n < 0$ and $c - b < 0$, $X = Z = 0$. Now sign $\binom{-b}{n} = (-1)^n$ and for $k = 1, \ldots, n-1$, $-k < c < -k + 1 \Rightarrow$ sign $\binom{-c}{n} = (-1)^{n-k}$, while for $-j < c - b < -j + 1$, $j = 1, \ldots, n-1$, sign $\binom{b-c}{n} = (-1)^{n-j}$. Therefore, from (3.5), (3.6) and (3.7),

$$N_1 = \begin{cases} 0 & \text{if } (n-j) \text{ even} \\ 1 & \text{if } (n-j) \text{ odd} \end{cases} \tag{3.11}$$

$$N_2 = \begin{cases} 2[(Y+1)/2] & \text{if } (j-k) \text{ is even} \\ 2[Y/2] + 1 & \text{if } (j-k) \text{ is odd} \end{cases}, \quad Y = E(b) \tag{3.12}$$

$$N_3 = \begin{cases} 0 & \text{if } k \text{ even} \\ 1 & \text{if } k \text{ odd} \end{cases}. \tag{3.13}$$

Now for $j > b - c > j - 1$ and $-k < c < -k + 1$, $b \in (j - k - 1, j - k + 1)$, $j - k = 1, 2, \ldots, n - 2$. If $b \in (j - k - 1, j - k)$, $Y = E(b) = j - k - 1$, whereas if $b \in (j-k, j-k+1)$, $Y = E(b) = j-k$. Considering the cases $(j-k)$ even and $(j-k)$ odd, it is straight-forward to check that for all j, $k \in \mathbb{N}$ with $j - k = 0, 1, \ldots, n-2$, we have

$$N_2 = j - k. \tag{3.14}$$

Equations (3.11), (3.12), (3.13) and (3.14) complete the proof of (ii). □

By virtue of Theorem 3.3 and the identities (2.1), (2.2) and (3.8), it is easy to see that we only have one possibility left that has not been analysed, namely,

$$1 - n < c - b < 0, \quad 1 - n < b < 0, \quad 1 - n < c < 0. \tag{3.15}$$

Theorem 3.4. *Let* $F = F(-n, b; c; z)$ *where* b *and* c *satisfy condition (3.15). If* $-j < b < -j+1$, $j = 1, \ldots, n-1$; $-k < c < -k+1$, $k = 1, \ldots, n-1$ *and* $-\ell < c-b < -\ell+1$, $\ell = 1, \ldots, n-1$, *then* F *has no real zeros if* $n+j+\ell$, $k+\ell$, $j+k$ *are even, one real zero in* $(1, \infty)$ *if* $n + j + \ell$ *is odd, one real zero in* $(0, 1)$ *if* $k + \ell$ *is odd and one real negative zero if* $j + k$ *is odd.*

Proof: Under the restrictions (3.15), we have

$$|1 - c| = 1 - c, \quad |b + n| = b + n, \quad |b - c - n| = c - b + n.$$

Then from (3.2), (3.3) and (3.4),

$$X = E(1 - c - n), \quad Y = E(b), \quad Z = E(c - b),$$

and it follows from (3.15) that $X = Y = Z = 0$. Also, sign $\binom{-b}{n} = (-1)^{n-j}$, sign $\binom{-c}{n} = (-1)^{n-k}$ and sign $\binom{b-c}{n} = (-1)^{n-\ell}$. The stated result then follows immediately from (3.5), (3.6) and (3.7). □

Remark 3.1 *We have not considered the asymptotic zero distribution as $n \to \infty$ of $F(-n, b; c; z)$. There are recent interesting results in this regard using different approaches, namely complex analysis techniques [10], matrix theoretic tools [11], asymptotic analysis of the Euler integral representation [3] and analysis of coefficients [8].*

Bibliography

1. M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*, (Dover, New York, 1965).
2. Bateman Manuscript Project, *Higher Transcendental Functions, Volume I*, (A. Erdélyi, editor; McGraw–Hill, New York, 1953).
3. K. Driver and P. Duren, "Asymptotic zero distribution of hypergeometric polynomials",*Numerical Algorithms*, 21 (1999), 147–156.
4. K. Driver and P. Duren, "Zeros of the hypergeometric polynomials $F(-n, b; 2b; z)$", *Indag. Math.*, 11 (1) (2000), 43–51.
5. K. Driver and P. Duren, "Trajectories of the zeros of hypergeometric polynomials $F(-n, b; 2b; z)$ for $b < -\frac{1}{2}$", *Constr. Approx.*, 17 (2001), 169–179.
6. K. Driver and P. Duren, "Zeros of ultraspherical polynomials and the Hilbert-Klein formulas", *J. Comput. and Appl. Math.*, 135 (2001), 293–301.
7. K. Driver and M. Möller, "Quadratic and cubic transformations and the zeros of hypergeometric polynomials", *J. Comput. and Appl. Math.*, to appear.
8. K. Driver and M. Möller, "Zeros of the hypergeometric polynomials $F(-n, b; -2n; z)$", *J. Approx. Th.*, 110 (2001), 74–87.
9. F. Klein, "Über die Nullstellen der hypergeometrischen Reihe", *Mathematische Annalen*, 37 (1890), 573–590.
10. A.B.J. Kuijlaars and W. van Assche, "The asymptotic zero distribution of orthogonal polynomials with varying weights", *J. Approx. Th.*, 99 (1999), 167-197.
11. A.B.J. Kuijlaars and S. Serra Capizzano, "Asymptotic zero distribution of orthogonal polynomials with discontinuously varying recurrence coefficients", *J. Approx. Th.*, to appear.
12. A.P. Prudnikov, Yu. A. Brychkov and O.I. Marichev, *Integrals and Series, Volume 3*, (Moscow, "Nauka", 1986 (in Russian); English translation, Gordon & Breach, New York, 1988); Errata in *Math. Comp.*, 65 (1996), 1380–1384.
13. G. Szegő, *Orthogonal Polynomials*, (American Mathematical Society, New York, 1959).
14. N. Temme, *Special Functions: An introduction to the classical functions of mathematical physics*, (Wiley, New York, 1996).

Approximation error maps

A. Gomide and J. Stolfi

Institute of Computing, University of Campinas, Brazil.
anamaria@ic.unicamp.br, stolfi@ic.unicamp.br

Abstract

In order to analyze the accuracy of a fixed, finite-dimensional approximation space which is not uniform over its domain Ω, we define *approximation error map*, a description of how the error is distributed over Ω—not for a single test function but for a general class of such functions. We show how to compute such a map from the best approximations to an orthonormal basis of the target function space.

1 Introduction

The expected accuracy of a finite-dimensional approximation space (e.g. a polynomial spline space, or a finite wavelet decomposition) will often vary over its domain Ω. Indeed, adaptive-resolution schemes are based on the premise that refining the element grid in a particular region of Ω will improve the approximation accuracy in that region.

Knowledge of how the expected approximation error varies over the domain Ω is obviously relevant to the evaluation of an approximation space, and to the tuning of knot locations, grid geometry, refinement thresholds and other parameters. Towards that goal, we introduce the concept of *approximation error map*, a description of how the error is distributed over Ω—not for a single test function, but for all functions in some specified space $\mathcal{F}$. We then show how to compute such a map from the best approximations to an orthonormal basis of $\mathcal{F}$.

1.1 Notation and definitions

Let $\mathcal{F}$ and $\mathcal{A}$ be two fixed, finite-dimensional vector spaces, not necessarily disjoint, of functions defined on some domain Ω with values in $\mathbf{R}$. Let $\|\cdot\|$ be a vector semi-norm for the space $\mathcal{A} + \mathcal{F}$. For any function $f \in \mathcal{F}$, we define its *best approximation* as the function $f^{\mathcal{A}} \in \mathcal{A}$ that minimizes the error $\|f - f^{\mathcal{A}}\|$.

We refer to $\mathcal{A}$ and $\mathcal{F}$ as the *approximation* and *gauge spaces*, respectively. We assume that the $\|\cdot\|$-balls *in the subspace* $\mathcal{A}$ are strictly convex, ensuring that the best approximation always exists and is unique. Since $(\alpha f)^{\mathcal{A}} = \alpha(f^{\mathcal{A}})$ and $\|\alpha f\| = |\alpha| \, \|f\|$ for any real constant α, we can confine the analysis of approximation errors to the *unit $\mathcal{F}$-sphere* $\mathcal{F}_1 = \{ f \in \mathcal{F} : \|f\| = 1 \}$.

1.2 Global error measures

Usually, the effectiveness of the approximation space $\mathcal{A}$ is measured by a single number $\|f - f^{\mathcal{A}}\|$—either for the worst-case function $f \in \mathcal{F}_1$, or by the root-mean-power average

over all functions $f \in \mathcal{F}_1$

$$\sigma^*_{p,\mathcal{A},\mathcal{F}} = \left[\int_{\mathcal{F}_1} \|f - f^{\mathcal{A}}\|^p \, df\right]^{1/p} \Bigg/ \left[\int_{\mathcal{F}_1} 1 \, df\right]^{1/p}. \tag{1.1}$$

Note that integrals are taken over the function space $\mathcal{F}_1$, not over the domain Ω. The worst-case error is the limit

$$\mu^*_{\mathcal{A},\mathcal{F}} = \lim_{p \to +\infty} \sigma^*_{p,\mathcal{A},\mathcal{F}} = \sup\left\{ \|f - f^{\mathcal{A}}\| : f \in \mathcal{F}_1 \right\}. \tag{1.2}$$

1.3 Uniform approximation spaces

A global error measure such as $\mu^*_{\mathcal{A},\mathcal{F}}$ or $\sigma^*_{p,\mathcal{A},\mathcal{F}}$ is generally sufficient when all points of Ω are equivalent with respect to the quality of approximation. More formally, we say that a normed function space $\mathcal{X}$ is *uniform* over Ω if there is some family Φ of maps from Ω to Ω that preserves $\mathcal{X}$ and its norm $\|\cdot\|$, and which can take any point of Ω to any other point. A natural example is $\mathcal{Y}^d_n$, the set of all harmonic functions on the sphere $\mathbf{S}^d$ of a given maximum order n, with any L_p norm; this space is preserved by the family of rigid rotations of $\mathbf{S}^d$. Obviously, if both $\mathcal{A}$ and $\mathcal{F}$ are uniform under the same family Φ, then $\mathcal{A}$ approximates $\mathcal{F}$ equally well at all points of Ω. (Of course, for any *specific* function $f \in \mathcal{F}$, the error $f - f^{\mathcal{A}}$ will usually vary over Ω.)

There are however many important approximation spaces $\mathcal{A}$ which are not uniform. A familiar example is the space of polynomials or trigonometric series defined on a bounded region $\Omega \subseteq \mathbf{R}^n$. Another example is the space of the piecewise polynomial splines of fixed order and continuity defined over a fixed grid G. Wavelet spaces truncated to a fixed order provide yet another example. For such spaces, the expected approximation error usually varies over Ω, even when the functions to be approximated are drawn from a uniform space.

2 Approximation error map

We define the *root mean power approximation error map of $\mathcal{F}$ by $\mathcal{A}$* as the function $\sigma_{p,\mathcal{A},\mathcal{F}}$ of Ω to $\mathbf{R}$ defined by

$$\sigma_{p,\mathcal{A},\mathcal{F}}(x) = \left[\int_{\mathcal{F}_1} \left|f(x) - f^{\mathcal{A}}(x)\right|^p \, df\right]^{1/p} \Bigg/ \left[\int_{\mathcal{F}_1} 1 \, df\right]^{1/p}. \tag{2.1}$$

As before, integrals are taken over the function space $\mathcal{F}_1$, not over the domain Ω. Note that $\sigma_{p,\mathcal{A},\mathcal{F}}(x)$ is not the error for a *specific* function f, but rather the *average* error *at the point x* for a *generic* function f in $\mathcal{F}_1$. As a limiting case, we define also the *worst-case approximation error map of $\mathcal{F}$ by $\mathcal{A}$* as the function

$$\mu_{\mathcal{A},\mathcal{F}}(x) = \lim_{p \to +\infty} \sigma_{p,\mathcal{A},\mathcal{F}}(x) = \sup\left\{ \left|f(x) - f^{\mathcal{A}}(x)\right| : f \in \mathcal{F}_1 \right\}. \tag{2.2}$$

Again note that the supremum is taken over $\mathcal{F}_1$, not over Ω, and that $\mu_{\mathcal{A},\mathcal{F}}(x)$ is not the error at x for a *single* function f, but rather the error for the function f in $\mathcal{F}_1$ that is worst *for that particular x*. A plot of $\sigma_{p,\mathcal{A},\mathcal{F}}(x)$ or $\mu_{\mathcal{A},\mathcal{F}}(x)$ over Ω should show at a

glance how well $\mathcal{A}$ approximates $\mathcal{F}$ in different parts of the domain, for all functions of $\mathcal{F}$ at once.

3 Computing the approximation error map

Formulas (2.1)–(2.2) become more tractable when the function metric $\|\cdot\|$ is the L_2 norm $\|f\| = [\int_\Omega |f(x)|^2\, dx]^{1/2}$ defined on the space $\mathcal{A}+\mathcal{F}$—in other words, when $\|f\|^2 = \langle f, f\rangle$ where $\langle f, g\rangle = \int_\Omega f(x)g(x)\, dx$. We make this assumption in the remainder of this section. In that case, $f^{\mathcal{A}}$ is a linear function of f, namely the orthogonal projection of f onto the subspace $\mathcal{A}$; and $\mu_{\mathcal{A},\mathcal{F}}$ is simply $|\sin\theta|$, where θ is the angle between the two subspaces.

3.1 Explicit formula for σ

Let us suppose that $\mathcal{A}$ and $\mathcal{F}$ are disjoint, and let $\phi_1, \ldots, \phi_n$ be an orthonormal basis for $\mathcal{F}$. Let $\alpha_i = \phi_i^{\mathcal{A}}$ for all i, and let $\varepsilon_i = \phi_i - \alpha_i$. We will call ϕ, α, and ε the *gauge*, *approximation*, and *error bases*, respectively (even though α_i and ε_i need not be independent). The average error map $\sigma_{p,\mathcal{A},\mathcal{F}}(x)$ can be expressed in terms of the error basis

$$\begin{aligned}
\sigma_{p,\mathcal{A},\mathcal{F}}(x) &= \left[\int_{\mathbf{S}^{n-1}} \left|\left(\sum_i c_i\phi_i\right)(x) - \left(\sum_i c_i\phi_i\right)^{\mathcal{A}}(x)\right|^p dc\right]^{1/p} \Big/ \left[\int_{\mathbf{S}^{n-1}} 1\, dc\right]^{1/p} \\
&= \left[\frac{1}{A_n}\int_{\mathbf{S}^{n-1}} \left|\sum_i c_i\varepsilon_i(x)\right|^p dc\right]^{1/p}, \qquad (3.1)
\end{aligned}$$

where $A_n = 2\pi^{\frac{n}{2}}/\Gamma(\frac{n}{2})$ is the measure of $\mathbf{S}^{n-1}$.

Note that $\sum_i c_i\varepsilon_i(x)$ is the dot product of the unit vector $c = (c_1, c_2, \ldots, c_n)$ and the vector $\varepsilon(x) = (\varepsilon_1(x), \varepsilon_2(x), \ldots, \varepsilon_n(x))$; it depends only on $|\varepsilon(x)|$ and on the angle θ between those two vectors, and is constant over the slice of $\mathbf{S}^{n-1}$ where θ is constant. The measure of that slice is $A_{n-1}\,|\sin\theta|^{n-1}\, d\theta$. Therefore,

$$\begin{aligned}
\sigma_{p,\mathcal{A},\mathcal{F}}(x) &= \left[\frac{1}{A_n}\int_0^\pi |\varepsilon(x)|\,|\cos\theta|^p\, A_{n-1}\,|\sin\theta|^{n-1}\, d\theta\right]^{1/p} \\
&= |\varepsilon(x)|\left[\frac{A_{n-1}}{A_n}\int_0^\pi |\cos\theta|^p\,|\sin\theta|^{n-1}\, d\theta\right]^{1/p} \\
&= |\varepsilon(x)|\left[\frac{(\Gamma(\frac{n}{2}))^2\Gamma(\frac{p+1}{2})}{\sqrt{\pi}\Gamma(\frac{n-1}{2})\Gamma(\frac{p+1+n}{2})}\right]^{1/p}. \qquad (3.2)
\end{aligned}$$

3.2 Explicit formula for μ

The worst-case error map $\mu_{\mathcal{A},\mathcal{F}}$ can be obtained by taking p to the limit $+\infty$ in formula (3.2), or directly, as follows. From formula (2.2),

$$\mu_{\mathcal{A},\mathcal{F}}(x) = \sup\left\{\left|\left(\sum_i c_i\phi_i\right)(x) - \left(\sum_i c_i\phi_i\right)^{\mathcal{A}}(x)\right| : \left\|\sum_i c_i\phi_i\right\| = 1\right\}$$

$$= \sup\left\{ \left| \sum_i c_i \varepsilon_i(x) \right| : c \in \mathbf{S}^{n-1} \right\}. \tag{3.3}$$

By considering the effect of negating each c_i, it is easy to see that the absolute value in the last formula is superfluous, i.e.

$$\mu_{\mathcal{A},\mathcal{F}}(x) = \sup\left\{ \sum_i c_i \varepsilon_i(x) : c \in \mathbf{S}^{n-1} \right\}. \tag{3.4}$$

Formula (3.4) is the supremum of a linear functional with coefficients $\varepsilon_i(x)$ over the sphere $\mathbf{S}^{n-1}$; which is achieved at the point $c^*(x)$ of $\mathbf{S}^{n-1}$ that is collinear with the coefficient vector, namely $c_i^*(x) = \varepsilon_i(x)/\sqrt{\sum_j (\varepsilon_j(x))^2}$, whence

$$\mu_{\mathcal{A},\mathcal{F}}(x) = \sum_i c_i^*(x)\varepsilon_i(x) = \sqrt{\sum_j (\varepsilon_j(x))^2} = |\varepsilon(x)|. \tag{3.5}$$

In summary, the error maps $\sigma_{p,\mathcal{A},\mathcal{F}}(x)$ and $\mu_{\mathcal{A},\mathcal{F}}(x)$ (which differ only by a constant factor) can be derived from the approximation errors $\varepsilon_i(x)$ for each basis function $\phi_i(x)$, combined with the norm $|\varepsilon(x)| = \sqrt{\sum_i (\varepsilon_i(x))^2}$.

4 Practical considerations

4.1 Connection between the function and point norms

The maps (2.2) and (2.1) will be more useful when there is a direct connection between the function-space norm $\|\cdot\|$ and the absolute value $|\cdot|$, used to compare functions values at a given point x, as in formulas (2.1)–(2.2)—namely, when

$$\|f\| = \left[\int_\Omega |f(x)|^q \, dx \right]^{1/q}. \tag{4.1}$$

More generally, the function values at x could be compared with a norm which could depend on x, or take derivatives of the function into account. We will not pursue such extensions in this paper.

Connection (4.1) is not strictly necessary—at least when $\mathcal{A}$ and $\mathcal{F}$ are finite dimensional. However, it may not make much sense to choose the approximant $f^{\mathcal{A}}$ so as to minimize the function norm $\|\cdot\|$, and then analyze its accuracy using some other norm $|\cdot|$, if there is no connection between the two.

Considering that the error map is relatively easy to compute when $\|\cdot\|$ is the L_2 norm (see Section 3), and probably intractable otherwise, the connection expressed by formula (4.1) will probably hold in practice (with $q = 2$).

4.2 Choice of the gauge space

The approximation error map depends not only on the space $\mathcal{A}$, but also on the gauge space $\mathcal{F}$ and the error metric $\|f\|$. Therefore, the choice of $\mathcal{F}$ and $\|\cdot\|$ must be guided by the intended application.

For example, suppose the domain Ω is the circle or the sphere $\mathbf{S}^d$, and the application does not specify a preferred direction. Then we should choose $\mathcal{F}$ and $\|\cdot\|$ so that they

are invariant under rotations of Ω—otherwise, any inhomogeneity in them may produce irrelevant artifacts in the error map. Also, if the functions to be approximated are expected to be smooth, and/or only their low frequencies are important, then the functions in $\mathcal{F}$ should be smooth too. A natural choice for $\mathcal{F}$, in this case, are the circular or spherical harmonics up to a certain maximum order, and the metric $\|\cdot\|$ can be simply the L_q norm over the sphere $\mathbf{S}^d$.

4.3 Essential dimensions

We will argue next that, for the L_2 function norm, the "interesting" part of the error map is determined by two "essential" subspaces $\mathcal{F}' \subseteq \mathcal{F}$ and $\mathcal{A}' \subseteq \mathcal{A}$, which are disjoint and such that $\dim \mathcal{F}' \geq \dim \mathcal{A}'$.

First, if the spaces $\mathcal{A}$ and $\mathcal{F}$ have a non-trivial intersection $\mathcal{V}$, and we split a function $f \in \mathcal{F}$ into its components $g \in \mathcal{V}$ and $h \perp \mathcal{V}$, we find that $f^{\mathcal{A}} = g + h^{\mathcal{A}}$; and that $h^{\mathcal{A}}$ is itself orthogonal to $\mathcal{V}$. Therefore, we can confine our attention to the complements $\mathcal{F}'$ and $\mathcal{A}'$ of $\mathcal{V}$ relative to $\mathcal{A}$ and $\mathcal{F}$, which are disjoint.

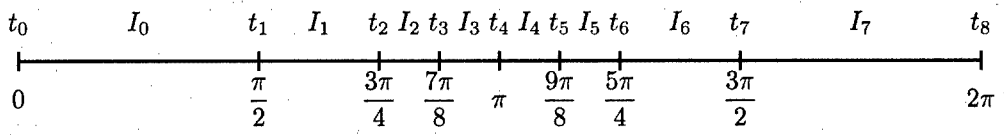
Let us then suppose that $\mathcal{A}$ and $\mathcal{F}$ are disjoint. If $\dim \mathcal{F} < \dim \mathcal{A}$, let $\mathcal{A}' \subset \mathcal{A}$ be the projection of $\mathcal{F}$ onto $\mathcal{A}$, which contains all optimum approximants. Obviously, for any function f, we have $f^{\mathcal{A}} = f^{\mathcal{A}'}$, so we can confine our attention to the space $\mathcal{A}'$, which is still disjoint from $\mathcal{F}$ and satisfies $\dim \mathcal{F} \geq \dim \mathcal{A}'$.

5 Examples

5.1 Trigonometric splines on the circle

Consider the approximation of a function by continuous trigonometric splines, of maximum frequency $r = 2$, defined on a partition T of $\mathbf{S}^1$ into $n = 8$ *unequal* intervals. This space coincides with the space $\mathcal{P}_0^{r,2}[T]$ of non-homogeneous polynomial splines of $\mathbf{R}^2$, restricted to $\mathbf{S}^1$, with C_0 continuity constraints [2].

For the gauge space $\mathcal{F}$, we will use the family of trigonometric series truncated after a suitable maximum frequency $s \geq r$, which coincides with the space of general spherical polynomials (not splines) $\mathcal{P}^{s,2}$ for some $s \geq r$. The norm is $\|f\| = \sqrt{\langle f, f \rangle}$ where $\langle f, g \rangle = \int_{\mathbf{S}^1} f(\theta) g(\theta)\, dp$. Specifically, T consists of the intervals I_0 through I_7 shown below



Within each interval I_j, the generic approximant is a linear combination g_j of the Fourier basis functions ϕ_i, for $-r \leq i \leq +r$. These partial functions are constrained to be continuous across interval boundaries; i.e. $g_{j-1}(t_j) = g_j(t_j)$ for each j in $\{0, \ldots, n-1\}$ (where all indices are taken modulo n). These equations turn out to be independent, therefore the dimension of $\mathcal{A}$ is $n(2r+1) - n = 32$.

For the gauge space $\mathcal{F}$, we will use the trigonometric polynomials of some order $s \geq r$, i.e. linear combinations of the basis functions ϕ_i for $-s \leq i \leq +s$, where $\phi_i(\theta) = (1/\sqrt{\pi}) \sin(i\theta + \pi/4)$. As observed in Section 4.3, we can ignore the subspace $\mathcal{A}' = \mathcal{F} \cap \mathcal{A}$ of

$\mathcal{A}$ generated by $\phi_{-r}, \ldots, \phi_r$. Moreover, in order to use all of $\mathcal{A}$, we need $\dim \mathcal{F} \geq \dim \mathcal{A}$—i.e., $2s+1 \geq 32$, implying $s \geq 16$. See Figure 1. The resulting error map $\mu_{\mathcal{A},\mathcal{F}}(x)$ is shown in Figure 2.

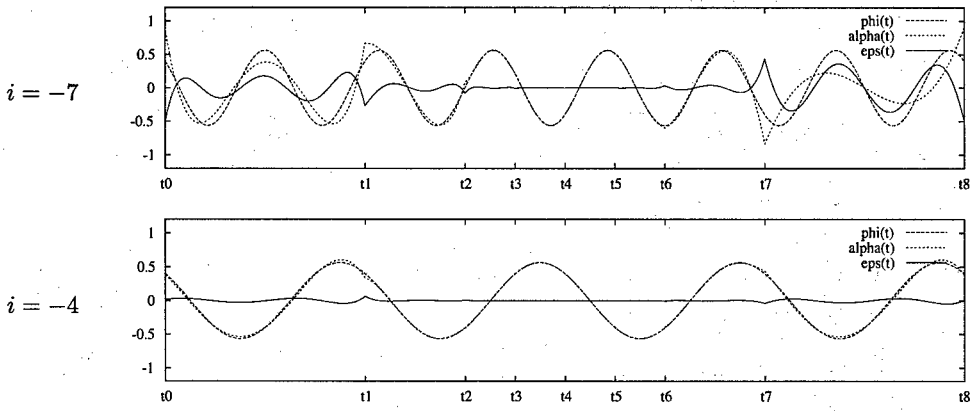


FIG. 1. The functions $\phi_i(t)$, $\alpha_i(t)$, and $\varepsilon_i(t)$, for selected values of i.

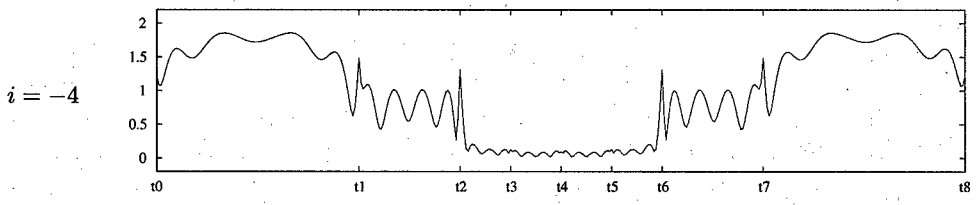


FIG. 2. The error map $\mu_{\mathcal{A},\mathcal{F}}(t)$ for continuous (C_0) trigonometric splines on eight unequal intervals, tested with the space of trigonometric polynomials of order 16.

5.2 Spherical splines on a uniform mesh

For the examples in this section, the approximating functions are spherical polynomial splines [1, 2, 3, 4] of continuity class zero and various degrees, homogeneous and non-homogeneous, defined on some triangulation T of the sphere $\mathbf{S}^2$.

Figure 3 (left) shows the approximation error map $\mu_{\mathcal{A},\mathcal{F}}(p)$ for the *homogeneous* spherical spline space $\mathcal{A} = \mathcal{H}_0^5[T]/\mathbf{S}^2$, which has dimension 252. In Figure 3 (right), $\mathcal{A}$ is the *non-homogeneous* spherical spline space $\mathcal{P}_0^4[T]/\mathbf{S}^2$, which has dimension 254. In both cases, the gauge space $\mathcal{F}$ is the family $\mathcal{Y}_{15}^2$ of spherical harmonics of maximum order 15, which has dimension 256. The intersection $\mathcal{F} \cap \mathcal{H}_0^5[T]/\mathbf{S}^2$ is the family of spherical harmonics of odd order ≤ 5 (dimension 21), whereas $\mathcal{F} \cap \mathcal{P}_0^4[T]/\mathbf{S}^2$ is the full harmonic space $\mathcal{Y}_4^2$ (dimension 25). The level curves are logarithmically spaced, five per decade.

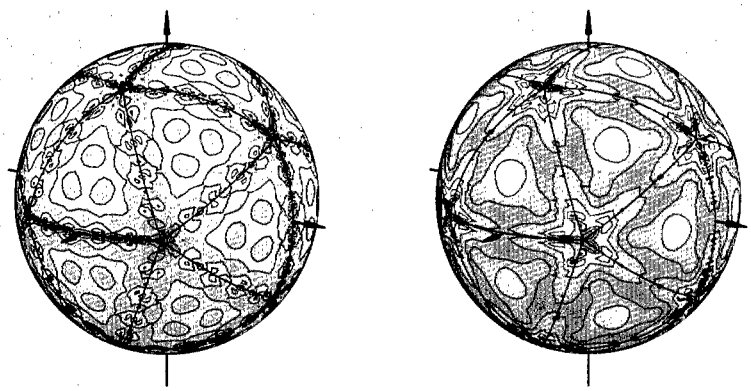


FIG. 3. Error maps $\mu_{\mathcal{A},\mathcal{F}}(p)$ for the approximation spaces $\mathcal{A} = \mathcal{H}_0^5[T]/\mathbf{S}^2$ (left) and $\mathcal{A} = \mathcal{P}_0^4[T]/\mathbf{S}^2$ (right). The maximum errors are 13.5 and 9.37, respectively.

5.3 Spherical splines on a variable mesh

In the following examples, the approximating functions are again spherical polynomial splines, but the vertices of the triangulation T have been displaced so as to create regions of very different sizes (still with icosahedral topology).

Figure 4 (left) shows the approximation error map $\mu_{\mathcal{A},\mathcal{F}}(p)$ for the space of *homogeneous* spherical splines $\mathcal{A} = \mathcal{H}_0^5[T]/\mathbf{S}^2$, which has dimension 252. In Figure 4 (right), $\mathcal{A}$ is the space of *non-homogeneous* spherical splines $\mathcal{P}_0^4[T']/\mathbf{S}^2$, which has dimension 254. In both cases, the gauge space $\mathcal{F}$ is the family $\mathcal{Y}_{15}^2$ of spherical harmonics of maximum order 15, which has dimension 256, as before. The level curves are logarithmically spaced (5 per decade).

6 Conclusion

Asymptotic error analysis is not very helpful when comparing two fixed finite-dimensional approximation spaces of similar dimensions—such as a spline space against a wavelet space, or two spline spaces with different grid geometries. Approximation errors computed for individual test functions are difficult to interpret and may not be representative of the average or worst cases. We expect that the approximation error map will be a useful analysis tool for those situations—especially for domains that admit natural uniform target spaces, such as spheres (including the circle) and tori.

Acknowledgments. This research was supported in part by CAPES, FINEP, and CNPq (PRONEX-SAI).

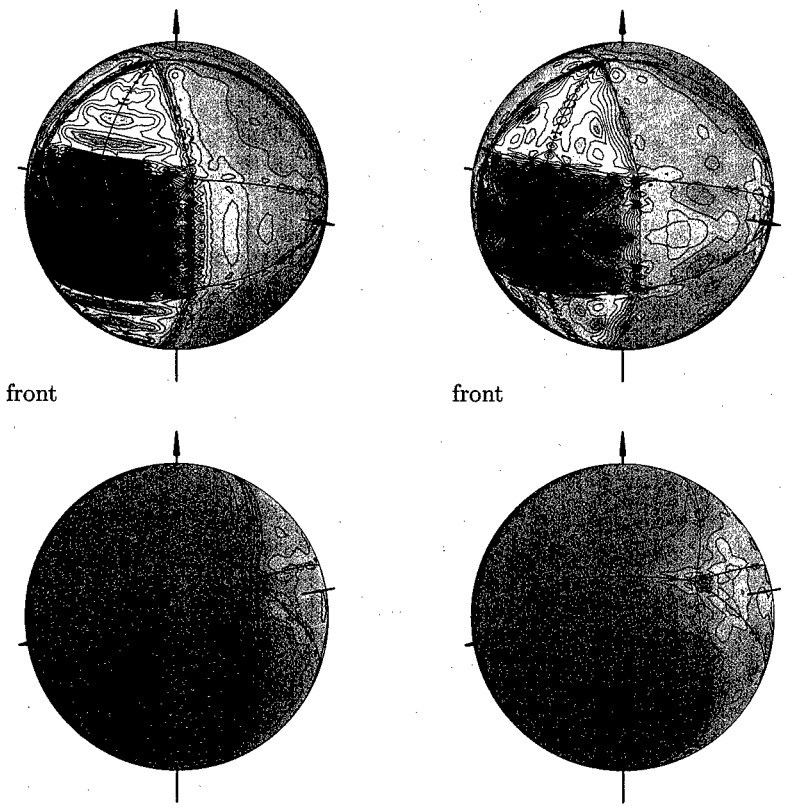


FIG. 4. Error maps $\mu_{\mathcal{A},\mathcal{F}}(p)$ for the approximation spaces $\mathcal{A} = \mathcal{H}_0^5[T]/\mathbf{S}^2$ (left) and $\mathcal{A} = \mathcal{P}_0^4[T]/\mathbf{S}^2$ (right). The maximum errors are 17.1 and 17.9, respectively.

Bibliography

1. P. Alfeld, M. Neamtu, and L. L. Schumaker. Dimension and local bases of homogeneous spline spaces. *SIAM Journal of Mathematical Analysis*, 27(5):1482–1501, Sept. 1996.
2. A. Gomide and J. Stolfi. Non-homogeneous polynomial C_k splines on the sphere S^n. Technical Report IC-00-10, Institute of Computing, Univ. of Campinas, July 2000.
3. A. Gomide and J. Stolfi. Bases for non-homogeneous polynomial $\mathbf{C}_k$ splines on the sphere. In *Lecture Notes in Computer Science 1380: Proc. LATIN'98 — Latin American Theoretical Informatics Conference*, pages 133–140. Springer, Apr. 1998.
4. A. Gomide. *Splines Polinomiais Não Homogêneos na Esfera.* PhD thesis, Institute of Computing, University of Campinas, May 1999. (In Portuguese).

Approximation by perceptron networks

Věra Kůrková

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, P.O. Box 5, 182 07 Prague 8, Czechia
vera@cs.cas.cz

1 Introduction

The classical perceptron proposed by Rosenblatt [22] as a simplified model of a neuron computes a weighted sum of its inputs and after comparing it with a threshold, applies an activation function representing a rate of neuron firing. To model this rate, Rosenblatt used the Heaviside discontinuous threshold function, which still is, together with its various continuous approximations, the most widespread type of activation used in neurocomputing. Formally, a perceptron with the Heaviside activation function computes a characteristic function of a half-space of $\mathcal{R}^d$, which is for practical reasons (all inputs are bounded) restricted to a box, usually $[0,1]^d$. Thus theoretical study of perceptron networks leads to various questions concerning approximation of functions by a special class of plane waves formed by linear combinations of characteristic functions of half-spaces (corresponding to the simplest model of perceptron network called the one-hidden-layer network with a linear output unit).

Although Rosenblatt's model was inspired biologically, plane waves (sometimes called ridge functions) have been studied for a long time by mathematicians motivated by various problems from physics. In contrast to integration theory, where functions are approximated by linear combinations of characteristic functions of boxes (simple functions), the theory of perceptron networks studies approximation of multivariable functions by linear combinations of characteristic functions of half-spaces. Expressions in terms of such functions exhibit the strength and weakness of plane waves methods described by Courant and Hilbert [4], page 676: "But always the use of plane waves fails to exhibit clearly the domains of dependence and the role of characteristics. This shortcoming, however, is compensated by the elegance of explicit results."

In this paper we survey our recent results on properties of approximation by linear combinations of characteristic functions of half-spaces. We focus on existence of best approximation, impossibility of choosing among best approximations a continuous one, estimates of rates of approximation by linear combinations of n characteristic functions of half-spaces and integral representation as a linear combination of a continuum of half-spaces.

This work was partially supported by GA ČR 201/99/0092 and 201/02/0428.

2 Preliminaries

A *perceptron* with an *activation function* $\psi : \mathcal{R} \to \mathcal{R}$ (where $\mathcal{R}$ denotes the set of real numbers) computes real-valued functions on $\mathcal{R}^d \times \mathcal{R}^{d+1}$ of the form $\psi(\mathbf{v} \cdot \mathbf{x} + b)$, where $\mathbf{x} \in \mathcal{R}^d$ is an *input* vector, $\mathbf{v} \in \mathcal{R}^d$ is an *input weight* vector and $b \in \mathcal{R}$ is a *bias*.

The most common activation functions are sigmoidals, i.e., functions with an ess-shaped graph. Both continuous and discontinuous sigmoidals are used. Here, we study networks based on the discontinuous *Heaviside function* ϑ defined by $\vartheta(t) = 0$ for $t < 0$ and $\vartheta(t) = 1$ for $t \geq 0$. Let H_d denote the set of functions on $[0,1]^d$ computable by Heaviside perceptrons, i.e.,

$$H_d = \{f : [0,1]^d \to \mathcal{R} \mid f(\mathbf{x}) = \vartheta(\mathbf{v} \cdot \mathbf{x} + b), \mathbf{v} \in \mathcal{R}^d, b \in \mathcal{R}\}.$$

Notice that H_d is the set of *characteristic functions of half-spaces* of $\mathcal{R}^d$ restricted to $[0,1]^d$.

For all positive integers d, H_d is compact in $(\mathcal{L}_p([0,1]^d), \|.\|_p)$ with $p \in [1,\infty)$ (see, e.g., [8]). This can be verified easily once the set H_d is reparameterized by elements of the unit sphere S^d in $\mathcal{R}^{d+1}$. Indeed, a function $\vartheta(\mathbf{v} \cdot \mathbf{x} + b)$, with a non-zero vector $(v_1, \ldots, v_d, b) \in \mathcal{R}^{d+1}$, is equal to $\vartheta(\hat{\mathbf{v}} \cdot \mathbf{x} + \hat{b})$, where $(\hat{v}_1, \ldots, \hat{v}_d, \hat{b}) \in S^d$ is obtained from $(v_1, \ldots, v_d, b) \in \mathcal{R}^{d+1}$ by normalization.

The simplest type of multilayer feedforward network has one hidden layer and one linear output. Such networks with Heaviside perceptrons in the hidden layer compute functions of the form

$$\sum_{i=1}^{n} w_i \vartheta(\mathbf{v}_i \cdot \mathbf{x} + b),$$

where n is the number of hidden units, $w_i \in \mathcal{R}$ are output weights and $\mathbf{v}_i \in \mathcal{R}^d$ and $b_i \in \mathcal{R}$ are input weights and biases, respectively. The set of all such functions is the *set of all linear combinations of n elements of H_d* and is denoted by $\mathrm{span}_n H_d$.

For all positive integers d, $\cup_{n \in \mathcal{N}_+} \mathrm{span}_n H_d$ (where $\mathcal{N}_+$ denotes the set of all positive integers) is dense in $(C([0,1]^d), \|.\|_C)$, the linear space of all continuous functions on $[0,1]^d$ with the supremum norm, as well as in $(\mathcal{L}_p([0,1]^d), \|.\|_p)$ with $p \in [1,\infty]$ (see, e.g., [5, 9]).

3 Existence of a best approximation

A subset M of a normed linear space $(X, \|.\|)$ is called *proximinal* if for every $f \in X$ the distance $\|f - M\| = \inf_{g \in M} \|f - g\|$ is achieved for some element of M, i.e., $\|f - M\| = \min_{g \in M} \|f - g\|$ (see, e.g., [23]). Clearly, a proximinal subset must be closed.

A sufficient condition for proximinality of a subset M of a normed linear space $(X, \|.\|)$ is compactness or bounded compactness. However, by extending H_d into $\mathrm{span}_n H_d$ for any positive integer n we lose compactness. Nevertheless compactness can be replaced by a weaker property that requires only those sequences that "minimize" a distance from M of an element of X to have convergent subsequences. More precisely, a subset M of a normed linear space $(X, \|.\|)$ is called *approximatively compact* if for each $f \in X$ and any sequence $\{g_i : i \in \mathcal{N}_+\} \subseteq M$ such that $\lim_{i \to \infty} \|f - g_i\| = \|f - M\|$, there exists $g \in M$ such that $\{g_i : i \in \mathcal{N}_+\}$ converges subsequentially to g (see, e.g., [23], p. 368). The following theorem is from [16].

Theorem 3.1 *For all n, d positive integers, $\mathrm{span}_n H_d$ is an approximatively compact subset of $(\mathcal{L}_p([0,1]^d, \|.\|_p)$ with $p \in [1,\infty)$.*

The proof is based on an argument showing that any sequence of elements of $\mathrm{span}_n H_d$ has a

subsequence that either converges to an element of $\mathrm{span}_n H_d$ or to a Dirac delta distribution, and the latter case cannot occur when such a sequence "minimizes" a distance from some function in $\mathcal{L}_p([0,1]^d)$.

It follows directly from the definitions that each approximatively compact subset is proximinal.

Corollary 3.2 *For all n, d positive integers, $\mathrm{span}_n H_d$ is a proximinal subset of $(\mathcal{L}_p([0,1]^d), \|.\|_p)$ with $p \in [1,\infty)$.*

Thus, for any fixed number n, a function in $\mathcal{L}_p([0,1]^d)$ has a best approximation among functions computable by a linear combination of n characteristic functions of half-spaces.

4 Uniqueness and continuity of a best approximation

Let M be a subset of a normed linear space $(X, \|.\|)$ and let $\mathcal{P}(M)$ denote the set of all subsets of M. The set-valued mapping $P_M : X \to \mathcal{P}(M)$ defined by $P_M(f) = \{g \in M : \|f - g\| = \|f - M\|\}$ is called the *metric projection of X onto M* and $P_M(f)$ is called the *projection of f onto M*.

Let $F : X \to \mathcal{P}(M)$ be a set-valued mapping. A *selection* from F is a mapping $\phi : X \to M$ such that for all $f \in X$, $\phi(f) \in F(f)$. A mapping $\phi : X \to M$ is called a *best approximation operator* from X to M if it is a selection from P_M.

When M is proximinal, then $P_M(f)$ is non-empty for all $f \in X$ and so there exists a best approximation mapping from X to M. The best approximation need not be unique. When it is unique, M is called a *Chebyshev set* (or "unicity" set). Thus M is Chebyshev if for all $f \in X$ the projection $P_M(f)$ is a singleton.

Recall that a normed linear space $(X, \|.\|)$ is called *strictly convex* (also called "rotund") if for all $f \neq g$ in X with $\|f\| = \|g\| = 1$ we have $\|(f+g)/2\| < 1$. It is well known that for all $p \in (1,\infty)$, $(\mathcal{L}_p([0,1]^d), \|.\|_p)$ is strictly convex.

The following theorem from [13] implies for p in the open interval $(1,\infty)$ that if among best approximations to $\mathrm{span}_n H_d$ (the existence of which is guaranteed by Corollary 3.2) there is a continuous one, then $\mathrm{span}_n H_d$ must be a Chebyshev set.

Theorem 4.1 *In a strictly convex normed linear space, any subset with a continuous selection from its metric projection is Chebyshev.*

We shall combine this theorem with the following geometric characterization of Chebyshev sets with a continuous best approximation from [24].

Theorem 4.2 *In a Banach space with strictly convex dual, every Chebyshev subset with continuous metric projection is convex.*

It is well known that $\mathcal{L}_p$-spaces with $p \in (1,\infty)$ satisfy the assumptions of this theorem (since the dual of $\mathcal{L}_p$ is $\mathcal{L}_q$ where $1/p + 1/q = 1$ and $q \in (1,\infty)$) (see, e.g., [7], p. 160). Hence, to show the non-existence of a continuous selection, it is sufficient to verify that $\mathrm{span}_n H_d$ is not convex.

Proposition 4.3 *For all n, d positive integers, $\mathrm{span}_n H_d$ is not convex.*

Indeed, consider $2n$ parallel half-spaces with the characteristic functions $g_i(\mathbf{x}) = \vartheta(\mathbf{v} \cdot \mathbf{x} + b_i)$, where $0 > b_1 > \ldots > b_{2n} > -1$ and $\mathbf{v} = (1, 0, \cdots, 0) \in \mathcal{R}^d$. Then $\frac{1}{2}\sum_{i=1}^{2n} g_i$ is a convex combination of two elements of $\mathrm{span}_n H_d$, $\sum_{i=1}^{n} g_i$ and $\sum_{i=n+1}^{2n} g_i$, but it is not in $\mathrm{span}_n H_d$, since its restriction to the one-dimensional set $\{(t, 0, \ldots, 0) \in \mathcal{R}^d : t \in [0,1]\}$ has $2n$ discontinuities.

Summarizing results of this section and the previous one, we get the following corollary.

Corollary 4.4 *In* $(\mathcal{L}_p([0,1]^d), \|.\|_p)$ *with* $p \in (1,\infty)$ *for all* n, d *positive integers there exists a best approximation mapping from* $\mathcal{L}_p([0,1]^d)$ *to* $\mathrm{span}_n H_d$*, but no such mapping is continuous.*

Thus convenient properties of projection operators such as uniqueness and continuity are not satisfied by $\mathrm{span}_n H_d$. These properties would allow one to estimate worst-case errors using methods of algebraic topology (see, e.g., [6]). In linear approximation theory, application of such methods shows that some sets of functions defined by smoothness conditions exhibit the curse of dimensionality: the approximants converge at rate $\mathcal{O}(1/\sqrt[d]{n})$, where d is the number of variables and n is the dimension of the approximating linear space (see, e.g., [20]). Our results show that these arguments are not applicable to approximation by $\mathrm{span}_n H_d$.

5 Rates of approximation

Let $(X, \|.\|)$ be a normed linear space and G be its subset, then *G-variation* (variation with respect to G) is defined as the Minkowski functional of the set cl conv $(G \cup -G)$, i.e.,

$$\|f\|_G = \inf\{c \in \mathcal{R}_+ : f/c \in \mathrm{cl}\,\mathrm{conv}\,(G \cup -G)\}.$$

Variation with respect to G is a norm on the subspace $\{f \in X : \|f\|_G < \infty\} \subseteq X$. The closure in its definition depends on the topology induced on X by the norm $\|.\|$. When X is finite-dimensional, G-variation does not depend on the choice of a norm on X, since all norms on a finite-dimensional space are topologically equivalent.

Variation with respect to G has been introduced in [17] as an extension of the concept from [1] of H_d-variation called *variation with respect to half-spaces*. For functions of one variable, variation with respect to half-spaces coincides, up to a constant, with the notion of total variation studied in integration theory (see [1]). For G countable orthonormal, it coincides with l_1-norm with respect to G (see [18]).

The following theorem from [17] is a reformulation of Maurey-Jones-Barron Theorem (see [2], [10], [21]) on estimates of rates of approximation of the order of $\mathcal{O}(1/\sqrt{n})$.

Theorem 5.1 *Let* $(X, \|.\|)$ *be a Hilbert space,* G *be its subset and* $s_G = \sup_{g\in G} \|g\|$*. Then for every* $f \in X$ *and for every positive integer* n*,*

$$\|f - \mathrm{span}_n G\| \leq \sqrt{\frac{(s_G\|f\|_G)^2 - \|f\|^2}{n}}.$$

Corollary 5.2 *For all positive integers* d, n *and for every* $f \in (\mathcal{L}_2([0,1]^d, \|.\|_2)$*,*

$$\|f - \mathrm{span}_n H_d\|_2 \leq \frac{\|f\|_{H_d}}{\sqrt{n}}.$$

Thus worst-case error in approximation of functions from the unit ball in H_d-variation by linear combinations of characteristic functions of n half-spaces of $[0,1]^d$ is at most $1/\sqrt{n}$. Estimates derived from Theorem 5.1 are sometimes called "dimension-independent", which is misleading since with increasing number of variables, the condition of being in the unit ball in G-variation becomes more and more constraining. See [19] for examples of smooth functions with H_d-variation growing exponentially with the number of variables d. However, such exponentially growing lower bounds

on variation with respect to half-spaces are merely lower bounds on upper bounds on rates of approximation by $\mathrm{span}_n H_d$, they do not prove that such functions cannot be approximated with faster rates than $\|f\|_{H_d}/\sqrt{n}$. Finding whether these exponentially large upper bounds are tight seems to be a difficult task related to some open problems in the theory of complexity of Boolean circuits.

Some insight into behavior of H_d-variation gives its geometric characterization derived in [19] using the Hahn-Banach Theorem.

Theorem 5.3 *Let $(X, \|.\|)$ be a Hilbert space and G be its nonempty subset. Then for every $f \in X$, $\|f\|_G = \sup_{h \in S} \frac{|f \cdot h|}{\sup_{g \in G} |g \cdot h|}$, where $S = \{h \in X - G^\perp : \|h\| = 1\}$.*

Thus functions that are "almost orthogonal" to H_d (i.e., have small inner products with characteristic functions of half-spaces) have large H_d-variation.

6 Integral representation

The following theorem from [14] shows that a smooth real-valued function on $\mathcal{R}^d$ with compact support can be represented as an integral combination of characteristic functions of half-spaces. By $H^-_{\mathbf{e},b}$ is denoted the half-space $\{\mathbf{x} \in \mathcal{R}^d : \mathbf{e} \cdot \mathbf{x} + b < 0\}$.

Theorem 6.1 *Let d be a positive integer and let $f : \mathcal{R}^d \to \mathcal{R}$ be compactly supported and $d+2$-times continuously differentiable. Then*

$$f(\mathbf{x}) = \int_{S^{d-1} \times \mathcal{R}} w_f(\mathbf{e}, b) \vartheta(\mathbf{e} \cdot \mathbf{x} + b) d\mathbf{e} db,$$

where for d odd

$$w_f(\mathbf{e}, b) = a_d \int_{H^-_{\mathbf{e},b}} \triangle^{k_d} f(\mathbf{y}) d\mathbf{y},$$

$k_d = (d+1)/2$, and a_d is a constant independent of f, while for d even,

$$w_f(\mathbf{e}, b) = a_d \int_{H^-_{\mathbf{e},b}} \triangle^{k_d} f(\mathbf{y}) \alpha(\mathbf{e} \cdot \mathbf{y} + b) d\mathbf{y},$$

where $\alpha(t) = -t \log |t| + t$ for $t \neq 0$ and $\alpha(0) = 0$, $k_d = (d+2)/2$, and a_d is a constant independent of f.

The assumption that f is compactly supported can be replaced by the weaker assumption that f vanishes sufficiently rapidly at infinity. The integral representation also applies to certain nonsmooth functions that generate tempered distributions.

By an approach reminiscent of Radon transform but based directly on distributional techniques from Courant and Hilbert [4], it was shown in [11] that if f is compactly supported function on $\mathcal{R}^d$ with continuous d-th order partial derivatives, where d is odd, then f can be represented as

$$f(\mathbf{x}) = \int_{S^{d-1} \times \mathcal{R}} v_f(\mathbf{e}, b) \vartheta(\mathbf{e} \cdot \mathbf{x} + b) d\mathbf{e} db,$$

where $v_f = a_d \int_{H_{\mathbf{e},b}} (D_{\mathbf{e}}^{(d)} f)(\mathbf{y}) d\mathbf{y}$, $a_d = (-1)^{k-1}(1/2)(2\pi)^{1-d}$ for $d = 2k+1$, $D_{\mathbf{e}}^{(d)} f$ is the directional derivative of f in the direction $\mathbf{e}$ iterated d times, $d\mathbf{e}$ is the $(d-1)$-dimensional volume element on S^{d-1}, and $d\mathbf{y}$ is likewise on a hyperplane. Although the coefficients v_f are obtained by integration over hyperplanes, while the w_f arise from integration over half-spaces, these coefficients can be shown to coincide by an application of the Divergence Theorem [3] p.423 to the half-spaces $H_{\mathbf{e},b}^-$. Theorem 6.1 extends the representation of [11] to even values for d and target functions f which are not compactly supported but which decrease sufficiently rapidly at infinity.

For $w \in \mathcal{L}_1(S^{d-1} \times \mathcal{R})$ and $f \in \mathcal{D}(\mathcal{R}^d)$ define

$$T_H(w)(\mathbf{x}) = \int_{S^{d-1}\times\mathcal{R}^d} w(\mathbf{e}, b)\vartheta(\mathbf{e}\cdot\mathbf{x} + b) d\mathbf{e} db,$$

$$S_H(f)(\mathbf{e}, b) = w_f(\mathbf{e}, b).$$

Theorem 6.1 shows that for each $f \in \mathcal{D}(\mathcal{R}^d)$, $T_H(S_H(f)) = f$. This theorem can be also used to estimate variation with respect to half-spaces by the $\mathcal{L}_1$-norm of the weighting function $w_f = v_f$. It is shown in [11] that for any f to which the above representation applies,

$$\|f\|_{H_d} \leq \int_{S^{d-1}\times\mathcal{R}^d} |w_f(\mathbf{e}, b)| d\mathbf{e}\, db.$$

Combining this upper bound on H_d-variation with Corollary 5.2, we get a smoothness condition that defines sets of functions that can be approximated by $\text{span}_n H_d$ with rates of the order of $1/\sqrt{n}$.

Bibliography

1. Barron, A. R. (1992). Neural net approximation, in *Proceedings of the 7th Yale Workshop on Adaptive and Learning Systems* (pp. 69–72).
2. Barron, A. R. (1993). Universal approximation bounds for superposition of a sigmoidal function, *IEEE Transactions on Information Theory* **39**, 930–945.
3. Buck, R. C. (1965). *Advanced Calculus*, McGraw-Hill: New York.
4. Courant, R. and Hilbert, D. (1962). *Methods of Mathematical Physics*, vol. 2. Wiley: New York.
5. Cybenko, G. (1989). Approximation by superpositions of a single function, *Mathematics of Control, Signal and Systems* **2**, 303–314.
6. DeVore, R., Howard, R. and Micchelli, C. (1989). Optimal nonlinear approximation, *Manuscripta Mathematica* **63**, 469–478.
7. Friedman, A. (1982). *Foundations of Modern Analysis*, Dover: New York.
8. Gurvits, L. and Koiran, P. (1997). Approximation and learning of convex superpositions, *Journal of Computer and System Sciences* **55**, 161–170.
9. Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators, *Neural Networks* **2**, 251–257.
10. Jones, L. K. (1992). A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training, *Annals of Statistics* **20**, 608–613.

11. Kůrková, V., Kainen, P. C. and Kreinovich, V. (1997). Estimates of the number of hidden units and variation with respect to half-spaces, *Neural Networks* **10**, 1061–1068.
12. Kainen, P. C., Kůrková, V. and Vogt, A. (1999). Approximation by neural networks is not continuous, *Neurocomputing* **29**, 47–56.
13. Kainen, P. C., Kůrková, V. and Vogt, A. (2000). Geometry and topology of continuous best and near best approximations, *Journal of Approximation Theory* **105**, 252–262.
14. Kainen, P. C., Kůrková, V. and Vogt, A. (2000). An integral formula for Heaviside neural networks, *Neural Network World* **10** 313–319
15. Kainen, P. C., Kůrková, V. and Vogt, A. (2000). Best approximation by Heaviside perceptron networks. *Neural Networks* **13** 645–647.
16. Kainen, P. C., Kůrková, V. and Vogt, A. (2001). Best approximation by linear combinations of characteristic functions of half-spaces (submitted to J. of Approx. Theory).
17. Kůrková, V. (1997). Dimension-independent rates of approximation by neural networks, in *Computer-Intensive Methods in Control and Signal Processing: Curse of Dimensionality* (Eds. Warwick, K., Kárný, M.) (pp. 261–270). Birkhauser: Boston.
18. Kůrková, V., Sanguineti, M. (2001). Bounds on rates of variable-basis and neural network approximation, *IEEE Trans. on Information Theory* **47**, 2659–2665.
19. Kůrková, V., Savický, P. and Hlaváčková, K. (1998). Representations and rates of approximation of real-valued Boolean functions by neural networks, *Neural Networks* **11**, 651–659.
20. Pinkus, A. (1986). *n-Width in Approximation Theory*, Springer: Berlin.
21. Pisier, G. (1981). Remarques sur un resultat non publié de B. Maurey, in *Seminaire d'Analyse Fonctionelle* I., n.12, Ecole Polytechnique, 1980-81.
22. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization of the brain, *Psychological Review* **65**, 386–408.
23. Singer, I. (1970). *Best Approximation in Normed Linear Spaces by Elements of Linear Subspaces*, Springer: Berlin.
24. Vlasov, L. P. (1970). Almost convex and Chebyshev sets, *Math. Notes Acad. Sci. USSR* **8**, 776–779.
25. Zemanian, A. H. (1987). *Distribution Theory and Transform Analysis*, Dover: New York.

Eye-ball rebuilding using splines with a view to refractive surgery simulation

Mathieu Lamard

Laboratoire de Traitement de l'Information Médicale, Ecole Nationale Supérieure des Télécommunications de Bretagne, F-29609 Brest Cedex, France.
Mathieu.Lamard@enst-bretagne.fr

Béatrice Cochener

CHU de Brest Ophtalmologie, 5 avenue Foch 29609 Brest Cedex, France.
Beatrice.Cochener-Lamard@chu-brest.fr

Alain Le Méhauté

Département de Mathématiques, UMR 6629, CNRS, Université de Nantes, BP 92208, F-44072 Nantes, Cedex 3, France
Alain.Le-Mehaute@math.univ-nantes.fr

Abstract

In this paper we present a use of splines in the biomedical field.

1 Introduction

In the surgical field of ophthalmology, refractive surgery has experiencied an important expansion for about fifteen years. It allows the surgeons to correct different refractive errors (myopia, hyperopia, astigmatism) aiming to decrease or minimize the use of optical equipments such as glasses and lenses. Many surgical techniques are today available for experts; with specific indications for each of them. Development of these methods commonly takes time and requires many research studies on animals before any clinical approach. In overall, abacus are established for all procedures. They provide to the surgeon some rules for the achievement of the surgery. These nomograms are usually based on statistical analysis of first wide series of operated patients. However, up to now, no technique is able to take into account individual variability of eyes (morphology, physiology).

The purpose of the present article is to consider this parameter in building a 3 dimensional numerical model of the eye and then applying to it various simulations of surgical techniques in order to measure their effects.

2 Eye and vision

2.1 The eye anatomy

Schematically the eye-ball has quite a spherical shape with a vertical diameter (approximately 23 mm) and an antero-posterior of 2 mm longer (axial length). Its average volume is 6.5 cm^3 for a weight of 7 grams.

2.2 Refractive errors

When parallels rays reach a normal eye, they are refracted and converge without accommodation on the retina (called emmetropia). Errors of refraction come from a disparity between the refractive capacity of the anterior segment of the eye and the length of the eye; the light rays are no longer focus on the retina. This is called ametropia, and is mainly of three types; myopia, hyperopia, astigmatism.

3 Correction of ametropia

3.1 Optical equipment

Glasses or lenses represent the traditional method. Glasses are safe and reversible for correction of most refractive errors but they can be responsible for visual field reduction and prismatic aberrations. They can also be a source of discomfort and cosmetic impairment for the wearer. Contact lenses have solved most of the problems associated with glasses, but require very strict hygiene to avoid severe complications. Refractive surgeries can bring an answer to these various problems.

3.2 Refractive surgery

Many techniques are available today in refractive surgeries. Most of them plan to reshape the cornea using of an excimer laser (193 nm). This laser (emitting in far UV) is used in two distinct surgeries;
— The Photo Refractive Keratomileusis (PRK)
— Laser Assisted In Situ Keratomileusis (LASIK).
The PRK technique removes cornea tissue on its surface in breaking molecular bindings. The depth and size of the ablation is determined as a function of the attempted correction. In LASIK the ablation is performed after the cut of a thin cornea flap (160 μm). This flap is replaced on the area of stromal ablation. In general PRK is used for correction of low ametropia and LASIK for low and medium corrections. For height corrections other concepts have been developed (additive surgery).

4 Data acquisition

In order to reconstruct the eyeball in 3D, data from the eye under consideration are needed. Numerous modalities allow us to obtain information about the eye anatomy.

4.1 Ultrasound

Ultrasound scan uses ultrasound waves for investigating human tissues in vivo. Nowadays in ophthalmology it is a routine exam for the posterior segment of the eye, especially for the research of foreign intra-ocular body. Reasons for this intense use are multiple,

including non invasive procedure, speed and low cost. But problems remain, which define the current limits ultrasound. Multiple phenomena of reflection (between two internal interfaces, or between an interface and a transducer itself) create false echos. Inaccuracies quickly increase with the deepening of the investigation because of all sources of "background noise", such as diffraction, diffusion and refraction. Advantages of ultrasound allowed us to use it without constraint to obtain maximum image quality. Our first work was to set up an images acquisition protocol of quality. The protocol privileged the underwater method to obtain a good acoustic coupling between the probe and the eye. The patient is lying on his back, he is wearing on his face a submarine mask without pane. This mask is filled with physiological serum. The probe, equipped with a lighting target, is plunged into the liquid. The patient fixes the target, in such conditions the provided images are along the optical axis. The operator turns this probe manually and regularly around the optical axis, and obtains a volume of data. A computer equipped with an image acquisition board can save all images on an hard disk. The images resolution is dependent on the probe and on the frequency of the ultrasound used.

4.2 MRI

The MRI, which tries to localize hydrogen pits by measuring their magnetization, realizes a real grey scale cartography of the proton concentration of the various examined structures [1]. The resultant data volume has a dependent acquisition time resolution, which currently represents one of the main important limitations of this technique. Besides the big quality of images obtained, the MRI has probably no harmful effect because it does not use ionisants beams.

4.3 Computerized corneal topography

The anterior surface of the cornea is one fundamental element of the refraction. Any modification or abnormality of this surface modifies the visual acuity. So the knowledge of this shape is extremely important. In a traditional way the Javal's keratometer is used to know punctually the refractive power of the cornea. In the last few years ophthalmologists have become used to another system, computerized corneal topography [2]. This technique, based on the reflection and the analysis of the Placido's discs deformation, allows us to obtain numerous data on the topology of the cornea. The curvature of the cornea is represented on a colored map.

4.4 Visible Human images

The images of the Visible Human project (the photographic modality) have great space resolution. They allow us to make reconstruction tests without acquisition problems.

5 Data segmentation

The purpose of this section is to addign a weight to each pixel of the image. The greater the weight the greater the contribution of this pixel to the reconstruction of the edge will be.

5.1 Pretreatments

Little pretreatment were done on the images under various modality. The speckle filtering or the use of enhancement contrast filter have a sure visual action but the reconstruction does not seem to be affected in our specific case. The only pretreatment used is an overlooked one. The ophthalmologist places four points on each image to isolate the lens and hence helps the treatment filters.

5.2 Treatments

To affect a weight to each pixel of the image, numerous edge detection filters were tested, using different methods, LOG, Canny-Deriche, Shen-Castan, and the operator based on the geometrical moments. The most convincing results were obtained with the Canny operator. It has been created as the solution of an optimization problem with constraints [3]. This filter is supposed to be an optimal compromise between the following criteria: localization, detection and unicity. We have to note that this filter is optimized for images flooded in a white, Gaussian, additive noise; and it is not the case in most of the used data.

This filter is actually one of the references in the edge detection for its quality of results; it is regularly used in the literature to the evaluation of new filters. A recursive implementation of this operator was developed by [4] allowing an important performance gain. The third dimension filter is obtained by supposing the filter separable and by making a convolution product. This choice is an easy one but it introduces anisotropies. These results images are difficult to use, and as recommended by [5], we extract its local maxima. This method consists in estimating the gradient direction and only keeping its watershed.

5.3 Post treatments

The previous stages can be applied to any type of images without taking into account their contents. Two post-treatments types are presented to take into account peculiarities of the eye contents. The first post-treatment consists to take into account ultrasound sound images and MRI particularities. The center of the eye have got no edges and generally the first visible edge is the good one. The "visible human" project images [10] have specifics characteristics. They are in fact photos of frozen tissues; crystals of ice are clearly visible in the vitrous, while it is uniform in the other modalities. A simple threshold is ineffective. The hysteresis threshold , introduced by [3] takes into account the edges connexity and luminance (levels of grey) and give us good results on such images.

6 Eyeball rebuilding with splines

The most used techniques for edge reconstruction on medical prints are snakes (active contour models) [7, 8]. A shape approaching the organ to be reconstructed is initialized, then deformed locally to fit the data. These deformations use, generally, physical properties of elasticity materials. These various methods allow the organ edge reconstruction of varied forms as bones, heart, brain, etc.. This type of reconstruction is effective but numerous parameters must be set. We opted for a different technique. The edge to be

reconstructed in our case is a quasi-spherical shape, and we reconstruct it by using B-splines. Their mathematical properties allow us to reconstruct the edge in a effective and fast way, and with adjusting only few parameters.

6.1 Principle

For a B-spline (1D) on $R = [a, b]$ we have to set:
— the degree k of the spline,
— the position and numbers of the knots $(\lambda_i, i = 0, ..., g + 1)$,
— the coefficients c_i of the spline representation:

$$s(x) = \sum_{i=-k}^{g} c_i N_{i,k+1}(x),$$

where $N_{i,k+1}(x)$ is the B-spline basis function.

We have chosen to set the degree of the spline to 3. Tests indicate this is a good compromise between computer time and result quality. The other parameter determination depends on the approximation criteria used and the position of control knots.

6.1.1 *The Dierckx criteria [6]*

The Dierckx approximation criteria determine a spline like the solution of a constrained minimization problem:

minimize

$$\tilde{n} := \sum_{i=1}^{g} \left(s^{(k)}(\lambda_i +) - s^{(k)}(\lambda_i -) \right)^2$$

with the constraint

$$\delta := \sum_{r=1}^{m} (w_r(y_r - s(x_r)))^2 \leq S$$

where (x_r, y_r) are the coordinates of the m data points, with w_r the associated weight.

6.1.2 *Control knots number*

As the number of control knots becomes important, the smoothness of the curvature decreases. Using that property we set up an iterative algorithm to perform the calculation of the spline. After an initialization with few control knots (we set for example $\lambda_0 = a$, $\lambda_2 = b$ and $\lambda_1 = (a + b)/2$), the spline is computed. If the smoothness is too important (with the δ estimation) we add some control knots and we start again the estimation of the smoothness. In the other case we stop the algorithm. At each iteration we can insert one or more control knots. The distribution of the control knots is recomputed for each iteration. They can be linearly distributed over $R = [a, b]$.

This method can be generalized to surfaces without difficulty (see [6]) using spherical coordinates and periodic boundary conditions.

6.2 Results

Different results are presented either in 3d or 2d view. In 2d view, the spline is drawn in red, and represents the intersection of the 2d spline and the data volume. The main reconstruction errors are due to segmentation errors. But the more data the better, and the quality of the reconstruction needs to be good. The reconstruction of images issued from visible human (22 slices) is better than from the MRI (8 slices) and the ultrasound images (4 slices).

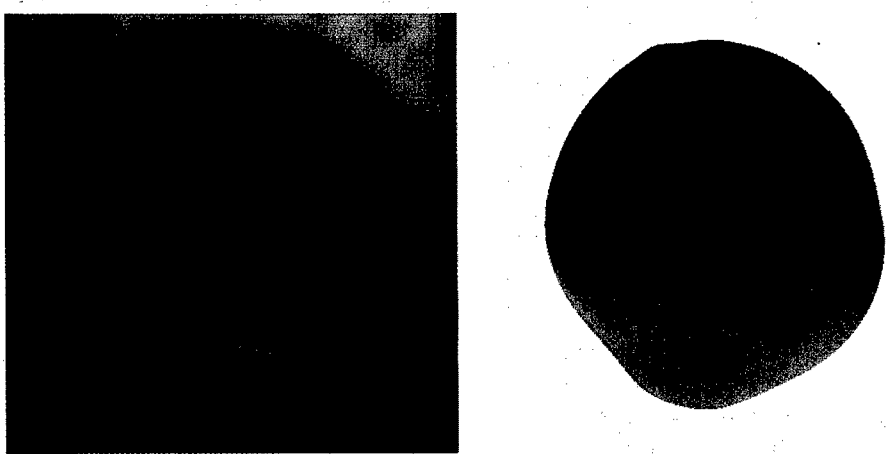


FIG. 1. Reconstruction using photographic images.

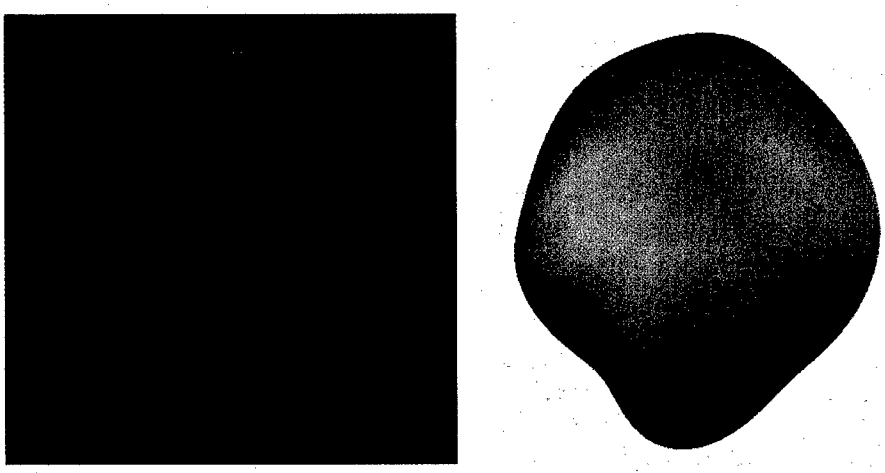


FIG. 2. Reconstruction using MRI.

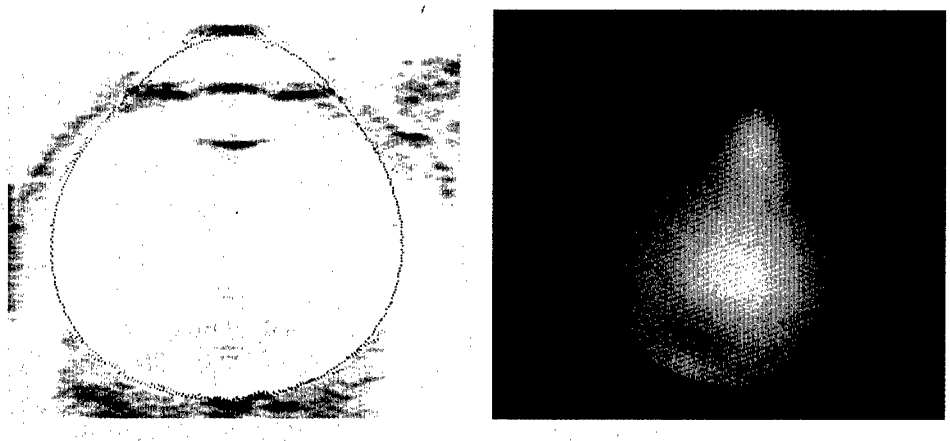


FIG. 3. Reconstruction using ultra sound images.

7 Elastic modelisation of surgery

7.1 Method used

The finite elements method is used to simulate surgery and solve the elasticity problem. Actually the knowledge of the comportment law of the eye ball tissues is the main limitation of this problem. Literature reports a wide range of coefficients to describe these tissues. In fact they seem to have an individual variability. So we use the approximation [9] for the elasticity coefficients which uses three parameters, internal pressure, radius of the eye and width of the edge. The use of complex models does not offer much information because of the low precision of the data that we used.

7.2 Results

Numerous simulations have been done. Results seem good in spite of the comportment law and the duration of the finite element method. The result are represented with a color map of the eye representing the curvature radius like the ophthalmologist does.

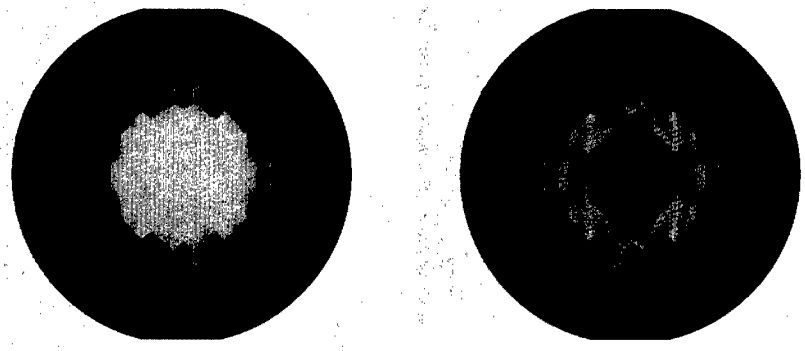


FIG. 4. Excimer Simulation before (left) and after (right).

8 Conclusion

This article presents a very modular path to realize modelisation of refractive surgeries. Each part of this work can be independently modified and can be adapted to an other organ. All this work has been validated by ophthalmologists. The eye ball reconstruction using spline appears to be an efficient method with a low CPU time. The mechanical modelisation provides proper results despite several approximations. This study might be useful for the medical doctor but also for testing new surgical techniques.

Bibliography

1. C. Dupas, La RMN au service de la medecine, *La Recherche* **81** (1977), 778–781.
2. S.D. Klyce, Computer assisted corneal topography : High resolution graphic presentation and analysis of keratoscopy, *Invest. Ophtalmol. Vi SCI.* **25** (1984), 1426–145.
3. J. Canny, A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence.* **6** (1986), 679–698.
4. R. Deriche, Fast algorithms for low level vision, *IEEE Transaction on Pattern Analysis and Machine Intelligence.* **12** (1990), 78–87.
5. R. Deriche, Techniques d'extraction de contours, (http://www-sop.inria.fr/robotvis/personnel/der/der-eng.html) *Cours de l'INRIA Sophia-Antipolis*, 1998.
6. P. Dierckx, Curve and surface fitting with splines, Clarendon press, Oxford, 1995.
7. M. Klass A. Witkin D. Terzopulos, Snakes: active contour models, *Int J Comput Vision.* **1** (1988), 321–331.
8. D. Metaxas, Physics-based deformable models : Application to computer vision, graphics and medical imaging. (1996) Kluwer Academic.
9. P.P. Purslow W.S. Karwatowski, Ocular Elasticity, *Ophtalmology.* **103** (1996), 1686–1692.
10. http://www.nlm.nih.gov/research/visible/visible_human.html.

A robust algorithm for least absolute deviations curve fitting

Dongdong Lei, Iain J Anderson

University of Huddersfield, Huddersfield, UK.
d.lei@hud.ac.uk

Maurice G Cox

National Physical Laboratory, Teddington, UK.
Maurice.Cox@npl.co.uk

Abstract

The least absolute deviations criterion, or the ℓ_1 norm, is frequently used for approximation where the data may contain outliers or 'wild points'. One of the most popular methods for solving the least absolute deviations data fitting problem is the Barrodale and Roberts (BR) algorithm (1973), which is based on linear programming techniques and the use of a modified simplex method [1]. This algorithm is particularly efficient. However, since it is based upon the simplex method it can be susceptible to the accumulation of unrecoverable rounding errors caused by using an inappropriate pivot. In this paper we shall show how we can extend a numerically stable form of the simplex method to the special case of ℓ_1 approximation whilst still maintaining the efficiency of the Barrodale and Roberts algorithm. This extension is achieved by using the ℓ_1 characterization to rebuild the relevant parts of the simplex tableau at each iteration. The advantage of this approach is demonstrated most effectively when the observation matrix of the approximation problem is sparse, as in the case when using compactly supported basis functions such as B-splines. Under these circumstances the new method is considerably more efficient than the Barrodale and Roberts algorithm as well as being more robust.

1 Introduction

Given a set of m data points $\{(x_i, y_i)\}_{i=1}^m$, the ℓ_1, or least absolute deviations curve-fitting problem seeks $c \in \mathbb{R}^n$ to solve the optimization problem

$$\min_c \|y - Ac\|_1 = \sum_{i=1}^m \left| y_i - \sum_{j=1}^n a_{i,j} c_j \right| = \sum_{i=1}^m |r_i|, \tag{1.1}$$

where A is an $m \times n$ observation matrix, and r_i denotes the residual of the ith point.

Another way of stating the ℓ_1, or least absolute deviations curve-fitting problem, is by the characterization theory of an ℓ_1 solution [8], which may be given in different forms. The following is perhaps the most commonly used.

A vector $c \in \mathbb{R}^n$ solves the minimization problem (1.1) if and only if there exist $\lambda \in \mathbb{R}^m$ such that

$$A^T\lambda = 0 \quad \text{with} \quad \begin{cases} |\lambda_i| \leq 1, & \text{for } i \in \mathcal{Z}, \\ \lambda_i = \text{sign}(r_i), & \text{for } i \notin \mathcal{Z}, \end{cases} \tag{1.2}$$

where $\mathcal{Z}$ represents the set of indices for which $r_i = 0$.

One of the popular methods designed for solving the ℓ_1 approximation problem is the Barrodale and Roberts (BR) algorithm. It replaces the unconstrained variables c and r in (1.1) by nonnegative variables c^+, c^-, u and v, and considers the linear programming problem

$$\begin{array}{lll} \min\limits_{c} & e^Tu + e^Tv & \\ \text{subject to} & Ac^+ - Ac^- + u - v & = y, \\ & c^+, c^-, u, v \geq 0. & \end{array} \tag{1.3}$$

Much of the reason for the popularity of the BR algorithm is that it exploits the characteristics of the ℓ_1 approximation in order to solve the problem in a more efficient manner than the general simplex approach. However, it is a simplex based method, and so it is susceptible to numerical instabilities caused by using inappropriate pivots. The new method presented here uses matrix factorization instead of simplex pivoting. This approach allows numerically stable updates to be made, thus avoiding the unnecessary build-up of rounding errors. This method is particularly efficient when the observation matrix is large and sparse [5].

Bartels [2] and Gill and Murray [4] presented methods that concentrate on avoiding the inherent instability of the simplex method. However, these methods are designed for a general linear programming problem and if we were to employ these techniques for the special case of the ℓ_1 problem, the storage requirements and computational workload of the method would be unnecessarily large compared to those of the highly efficient BR algorithm.

The ℓ_1 problem is, in essence, an interpolation problem. The aim of any iterative procedure for the ℓ_1 problem is to find an optimal set of interpolation points. Indeed, this is how the BR algorithm solves the ℓ_1 problem. It begins with all coefficients, c, set to zero (being non-basic variables), and during each iteration of stage one, one of the residuals, r_i, becomes non-basic by making the corresponding point an interpolation point (i.e., the coefficients are altered so that $r_i = 0$). At the end of stage one, the current estimate interpolates n distinct points. During stage two, the interpolation points are exchanged one at a time with a non-interpolation point until an optimal solution is achieved.

In fact, the new algorithm is effectively identical to the BR algorithm in the sense that we use exactly the same pivoting strategy. However, we start with a predetermined set of interpolation points and do not store the simplex tableau directly. In each iteration, we only reconstruct the parts of the simplex tableau that are needed by the more stable approach employed.

2 A more stable computational approach

The linear programming presentation of a least absolute deviations curve-fitting problem is given in (1.3). It is a standard linear programming problem of dimension $m \times (2m+2n)$. The robust approaches of Bartels and Gill and Murray can be applied to solve it. They involve the factorization of an $m \times m$ matrix. On the other hand, the BR algorithm only deals with an $m \times n$ matrix in each iteration, if $m \gg n$, the direct usage of these stable approaches is less efficient. We shall show next that the factorization of an $n \times n$ matrix is all that is required at each iteration.

We split the data points based on the set interpolation $\mathcal{Z}$, and let A_Z, y_Z, u_Z and v_Z be the counterparts of A, y, u and v in (1.3) corresponding to the set $\mathcal{Z}$. Their complementary matrix and vectors are denoted by $\widetilde{A}_Z$ and $\widetilde{y}_Z$, $\widetilde{u}_Z$ and $\widetilde{v}_Z$, so that A_Z and $\widetilde{A}_Z$ comprise A, etc., problem (1.3) can be expressed as

$$\begin{array}{lll} \min\limits_{c} & e^{\mathrm{T}}(u_Z + \widetilde{u}_Z) + e^{\mathrm{T}}(v_Z + \widetilde{v}_Z) & \\ \text{subject to} & A_Z c^+ - A_Z c^- + u_Z - v_Z & = y_Z, \\ & \widetilde{A}_Z c^+ - \widetilde{A}_Z c^- + \widetilde{u}_Z - \widetilde{v}_Z & = \widetilde{y}_Z, \\ & c^+, c^-, u_Z, \widetilde{u}_Z, v_Z, \widetilde{v}_Z & \geq 0. \end{array} \tag{2.1}$$

Since the coefficients for c_j^- are just the negative of the coefficients for c_j^+, $j = 1, 2, \ldots, n$, it is possible to suppress c_j^- and let c represent the unconstrained variable. The initial simplex tableau associated with problem (2.1) can be constructed in matrix form by Table 1, where e_k, $k = m, n, m-n$, are $k \times 1$ vectors with all components equal to one.

BV	c	u_Z	$\widetilde{u}_Z$	v_Z	$\widetilde{v}_Z$	r
u_Z	A_Z	I	0	$-I$	0	y_Z
$\widetilde{u}_Z$	$\widetilde{A}_Z$	0	I	0	$-I$	$\widetilde{y}_Z$
Z	$e_m^{\mathrm{T}} \begin{pmatrix} A_Z \\ \widetilde{A}_Z \end{pmatrix}$	0	0	$-2e_n^{\mathrm{T}}$	$-2e_{m-n}^{\mathrm{T}}$	$e_m^{\mathrm{T}} \begin{pmatrix} y_Z \\ \widetilde{y}_Z \end{pmatrix}$

TAB. 1. The initial simplex tableau of the ℓ_1 fitting problem.

As we know, the simplex method is an iterative procedure in which each iteration is characterized by specifying which m of $2m+n$ variables are basic. For the ℓ_1 approximation, we are only concerned with those vertices which are formed by a set of interpolation points. For n interpolation points, the basic variables consist of n of the coefficient parameters c and $m-n$ of the parameters $\widetilde{u}_Z$ corresponding to the non-interpolation points.

Let B be the $m \times m$ basis matrix whose columns consist of the m columns associated with the basic variables. Then

BV	u_z	r
c	A_Z^{-1}	$A_Z^{-1}y_Z$
$\widetilde{u}_Z$	$-\widetilde{A}_Z A_Z^{-1}$	$\widetilde{r}_Z$
Z	$-e_{m-n}^{\mathrm{T}}(\widetilde{A}_Z A_Z^{-1}) - e_n^{\mathrm{T}}$	$e_{m-n}^{\mathrm{T}}\widetilde{r}_Z$

TAB. 2. The condensed simplex tableau associated with a set of interpolation points.

$$B = \left(\begin{array}{c|c} A_Z & 0 \\ \hline \widetilde{A}_Z & I \end{array}\right). \tag{2.2}$$

It is readily verified that the inverse of B can be written in the form of (2.3) as long as A_Z is invertible.

$$B^{-1} = \left(\begin{array}{c|c} A_Z^{-1} & 0 \\ \hline -\widetilde{A}_Z A_Z^{-1} & I \end{array}\right). \tag{2.3}$$

Equation(2.3) shows that the explicit inverse computation of an $m \times m$ matrix in the form of (2.2) can be achieved by dealing with an inverse of an $n \times n$ matrix, and in general, $n \ll m$.

To make the m non-basic variables become basic, we multiply the whole simplex tableau by B^{-1}, and omit the identity and zero matrices. Then new simplex tableau is given in Table 2.

An arbitrary choice of the interpolation set Z may cause some of the values in the right hand side column to become negative. Although it is permissible for the coefficient parameters c to be negative, for those rows having negative residuals $\widetilde{r}_Z$, we restore feasibility by exchanging the corresponding $\widetilde{u}_Z$ for $\widetilde{v}_Z$. This exchanging can be made by subtracting twice those rows from the objective row and changing the sign of the original rows [1].

Such an exchange process can be expressed in matrix terms by introducing a sign vector

$$\widetilde{\lambda}_Z = \mathrm{sign}(\widetilde{r}_Z).$$

Let $\widetilde{A}_{Z_S}$ represent the matrix which is obtained by multiplying those rows of $\widetilde{A}_Z$ associated with negative residuals by -1,

$$\widetilde{A}_{Z_S} = \mathrm{diag}(\widetilde{\lambda}_Z)\widetilde{A}_Z.$$

BV	u_z	r
c	A_Z^{-1}	$A_Z^{-1}y_Z$
$\widetilde{u}_Z$	$-\widetilde{A}_{Z_S}A_Z^{-1}$	$\lvert\widetilde{r}_Z\rvert$
Z	$-\widetilde{\lambda}_Z^{\mathrm{T}}(\widetilde{A}_Z A_Z^{-1}) - e_n^{\mathrm{T}}$	$\widetilde{\lambda}_Z^{\mathrm{T}}\widetilde{r}_Z$

TAB. 3. Restoration of feasibility of the simplex tableau.

The simplex tableau after restoring feasibility is shown in Table 3.

The point to be removed from $\mathcal{Z}$ is decided by the values of the objective row. Each time the maximum value of the objective row (including the suppressed columns) is chosen, we let the index of this element be k. In order to choose which new point is to join the set $\mathcal{Z}$, we compute the value of the pivotal column, the kth column in the simplex tableau. Since the simplex tableau is in the form of

$$\left[\begin{array}{c} I \\ \hline -\widetilde{A}_{Z_S} \end{array}\right] A_Z^{-1},$$

the kth column can be obtained by using $\widetilde{A}_{Z_S}$ and the kth column of A_Z^{-1}.

The BR algorithm pivoting strategy is adopted to decide which new point is to be added to the interpolation set, when a new set of indices $\mathcal{Z}$ is generated. We repeat the process in an iterative manner until the optimal solution is achieved.

Table 3 is in fact identical to the simplex tableau of the BR algorithm in stage 2. The difference here is that the BR algorithm is implemented by a simplex pivoting approach, while the transformation of the simplex tableau in the form of Table 3 can be accomplished in a numerically more stable manner.

3 The improved method

The improved method starts with a predetermined interpolation set $\mathcal{Z}$, the minimum requirement for $\mathcal{Z}$ being that it forms a well-behaved matrix A_Z. For B-spline basis functions, we can choose any set of points satisfying the Schoenberg-Whitney condition [6]. For a Chebyshev polynomial basis, points close to the n Chebyshev zeros can be regarded as the initial interpolation set. In other cases, we can choose points approximate to them or even uniformly distributed.

If we denote the set of λ_i, $i \in \mathcal{Z}$, as λ_Z, we can rewrite the characterization equation (1.2) as

$$A_Z^{\mathrm{T}}\lambda_Z = -\widetilde{A}_Z^{\mathrm{T}}\widetilde{\lambda}_Z, \tag{3.1}$$

and λ_Z can be obtained mathematically from

$$\lambda_{\mathcal{Z}} = -(A_{\mathcal{Z}}^{\mathrm{T}})^{-1}(\widetilde{A}_{\mathcal{Z}}^{\mathrm{T}}\widetilde{\lambda}_{\mathcal{Z}}). \tag{3.2}$$

Table 3 shows that the objective row can be computed as

$$\text{Objective row} = -(\widetilde{\lambda}_{\mathcal{Z}}^{\mathrm{T}}\widetilde{A}_{\mathcal{Z}})A_{\mathcal{Z}}^{-1} - e_n^{\mathrm{T}}. \tag{3.3}$$

Thus, using (3.2) we conclude that

$$\text{Objective row} = \lambda_{\mathcal{Z}}^{\mathrm{T}} - e_n^{\mathrm{T}}. \tag{3.4}$$

We know that at the ℓ_1 solution all the values in the objective row are in the range $[-2,\ 0]$, and also $|\lambda| \leq 1$. This latter result can be explained in terms of the former by the relationship (3.4).

(3.4) is useful because it can be used to verify whether an interpolation set forms an optimal solution, or to compute λ from the values of the objective row. We use it to compute the values of the objective row.

The improved method can be summarized as follows;

(1) Choose an initial set of interpolation points and form the set $\mathcal{Z}$.

(2) Construct $A_{\mathcal{Z}}$, $y_{\mathcal{Z}}$ and their counterpart $\widetilde{A}_{\mathcal{Z}}$, $\widetilde{y}_{\mathcal{Z}}$ accordingly.

(3) Solve the equation $A_{\mathcal{Z}}c = y_{\mathcal{Z}}$ for c, and compute

$$\widetilde{r}_{\mathcal{Z}} = \widetilde{y}_{\mathcal{Z}} - \widetilde{A}_{\mathcal{Z}}c, \qquad \text{and} \qquad \widetilde{\lambda}_{\mathcal{Z}} = \operatorname{sign}(\widetilde{r}_{\mathcal{Z}}).$$

(4) Obtain the values of $\lambda_{\mathcal{Z}}$ from the equation

$$A_{\mathcal{Z}}^{\mathrm{T}}\lambda_{\mathcal{Z}} = -\widetilde{A}_{\mathcal{Z}}^{\mathrm{T}}\widetilde{\lambda}_{\mathcal{Z}}. \tag{3.5}$$

(5) If $|\lambda_{\mathcal{Z}}| \leq 1$ hold, the current solution is optimal, and the algorithm terminates. Otherwise, continue.

(6) Obtain the objective row of the BR simplex tableau from

$$\text{objective row } = \lambda_{\mathcal{Z}}^{\mathrm{T}} - e_n^{\mathrm{T}}.$$

(7) Examine the values of the objective row; the point associated with the maximum value of the objective row is chosen to leave the set $\mathcal{Z}$.

(8) Decide the point to add by the BR pivoting strategy. Obtain a new set of indices $\mathcal{Z}$, and repeat from step 2.

4 Practical considerations and application to the ℓ_1 spline approximation

The robustness of the above algorithm stems from the reliable updating of the relevant parts of the simplex tableau in each iteration. The major computational work is obtaining (explicitly or implicitly) the inverse of an $n \times n$ matrix $A_{\mathcal{Z}}$. It can be calculated and stored explicitly by using an LU or QR factorization, or preferably it can be expressed as a product of factors. Since $A_{\mathcal{Z}}$ differs from its predecessor by only one row, savings can be made by reusing results from the previous step. Necessary material is available [4, 7] regarding the stable implementation of this row updating procedure.

$m = 512$	Numbers of iterations		Execution Time (seconds)	
$q =$	New	BR	New	BR
44	57	125	1.6	14.7
49	75	111	2.2	13.4
54	71	134	2.4	20.2
59	83	156	3.0	26.8
64	78	160	3.1	32.'
69	88	194	4.0	42.4
74	75	165	3.7	36.0
79	87	189	4.8	48.1

TAB. 4. The number of iterations and execution time taken by the algorithm of this paper and the Barrodale and Roberts algorithm for a set of 512 response data points provided by the National Physical Laboratory.

Sparsity almost always is more important than matrix dimension. Additional savings can be made if the observation matrix A is sparse or structured. Approximation using a B-spline basis often occurs in practical applications. In such cases, A is block banded, and A_Z can be triangularized using $\mathcal{O}(n)$ flops [3]. Similarly, the sparsity of A can be exploited to compute other relevant parts of the simplex tableau efficiently.

We have applied our method to solve the least absolute deviations curve-fitting problems by B-splines using various numbers of interior knots. All software was written in MATLAB and implemented on a Sun Workstation. The initial interpolation points are chosen to be those points corresponding to the maximum value in each column of the observation matrix A.

Some of our computational results are reported in Tables 4 and 5. Each table presents the outcomes of a particular set of data points by the new method and by the BR algorithm.

All the experimental results exhibit the effectiveness of the improved method on large, sparse systems. Although these tables show that the improved method is faster than the BR algorithm, it would be unfair to judge the convergence speed purely based upon the time taken, since the improved method embodies some MATLAB built-in functions, while the BR algorithm uses only user-defined functions. However, on average, the new method requires far fewer iterations than the BR algorithm, and is competitive with the BR algorithm both in efficiency and accuracy for a structured system.

Further work to be addressed by the authors will involve a definitive implementation of this algorithm in Fortran, and development of an error analysis for both the improved method and the BR algorithm.

$m = 1200$	Numbers of iterations		Execution Time (seconds)	
$q =$	New	BR	New	BR
50	82	143	4.0	58.7
56	105	165	5.2	85.8
62	113	190	6.1	110.2
68	131	189	7.6	110.4
74	121	223	7.8	157.9
80	132	216	9.2	163.2
86	155	245	11.8	209.8
92	173	252	14.0	241.8
98	153	272	13.6	292.6

TAB. 5. The number of iterations and execution time taken by the algorithm of this paper and the Barrodale and Roberts algorithm for a set of 1200 data points, generated by MATLAB command $x = \text{linspace}(1, 10, 1200)'$; $y = \log(x) + \text{randn}(1200, 1)$.

Bibliography

1. I. Barrodale and F. D. K. Roberts. An improved algorithm for discrete ℓ_1 linear approximation. *SIAM Journal of Numerical Analysis* **10**, 839–848, 1973.
2. R. H. Bartels. A stabilization of the simplex method. *Numerical Math.* **16**, 414–434, 1971.
3. M. G. Cox. The least squares solution of overdetermined linear equations having band or augmented band structure. *IMA Journal of Numerical Analysis* **1**, 3–22, 1981.
4. P. E. Gill and W. Murray. A numerically stable form of the simplex algorithm. *Linear Algebra and its Applications* **7**, 99–138, 1973.
5. D Lei, I. J. Anderson, and M. G. Cox, An improved algorithm for approximating data in the ℓ_1 norm. In P. Ciarlini, M. G. Cox, E. Filipe, F. Pavese, and D. Richter, editors, *Advanced Mathematical and Computational Tools in Metrology V*, 247–250, Singapore, 2001. World Scientific Publishing.
6. M. J. D. Powell. *Approximation Theory and Methods.* Cambridge University Press, Cambridge, UK, 1981.
7. R. J. Vanderbei. *Linear Programming — Foundations and Extensions.* Kluwer Academic Publishers, Boston, MA, US, 1997.
8. G. A. Watson. *Approximation Theory and Numerical Methods.* Wiley, New York, US, 1980.

Tomographic reconstruction using Cesaro-means and Newman-Shapiro operators

Ulrike Maier

Mathematisches Institut, Justus-Liebig University, 35392 Giessen, Germany
Ulrike.Maier@math.uni-giessen.de

Abstract

Tomography is well known because of its many applications. Although theoretically solved, the numerical implementation of tomographic reconstruction algorithms is still a difficult problem. In this article the numerical implementation of a reconstruction method using Cesaro-means and Newman-Shapiro operators is described. The key point herein is the use of suitable quadrature formulae on the sphere. It turns out that in the context described product Gaussian formulae are best suited. The algorithm is tested at the so called Shepp-Logan phantom which is a three dimensional model of a human head.

1 Introduction and notation

The problem in tomography is to reconstruct a function F from its Radon transform sufficiently well. Since certain classes of functions can be expanded into series of orthogonal polynomials it is essential to exploit the action of the Radon transform on orthogonal polynomials and on polynomials in general.

This approach is the more interesting since the inverse of the Radon transform for polynomials is known explicitly.

The convergence of orthogonal expansions to the given function is often achieved only by applying a summability method. The application of such methods can be interpreted as a kind of "filter technique" which is necessary for sufficiently good reconstructions. The combination of an expansion of the function and the application of suitable summability methods leads to promising reconstruction algorithms.

In this article two examples for a summability method and their implementation are presented — the Cesaro-means and Newman-Shapiro-means. After some introductory remarks on Laplace-series at the end of this section, in Section 2 the theory of summability methods needed here is presented. In Section 3 this theory is applied to the reconstruction of functions from their Radon transform. Section 4 describes the numerical implementation of the reconstruction formula which is tested on the so called Shepp–Logan phantom of a head in Section 5.

In this article the following notation is used. Let B^r denote the unit ball in $\mathbb{R}^r$, S^{r-1} denote the unit sphere and $Z^r := [-1,1] \times S^{r-1}$. xy denotes the Euclidean product of $x, y \in \mathbb{R}^r$.

The spaces of restrictions of r–variate polynomials, homogeneous polynomials and homogeneous harmonic polynomials of degree $\mu \in \mathbb{N}_0$ onto a subspace $X \subset \mathbb{R}^r$ ($X = S^{r-1}$ or $X = B^r$) are denoted by $\mathbb{P}_\mu^r(X)$, $\overset{*}{\mathbb{P}}{}_\mu^r(X)$, $\overset{*}{\mathbb{H}}{}_\mu^r(X)$, respectively. The space $C(S^{r-1})$ of all continuously differentiable functions is provided with the inner product $< F, G >:= \int_{S^{r-1}} F(x)G(x)dx$. The surface measure of the sphere is denoted by $\omega_{r-1} =< 1, 1 >$.

Let C_μ^λ denote the Gegenbauer polynomials of degree μ and index λ and $\tilde{C}_\mu^\lambda = C_\mu^\lambda / C_\mu^\lambda(1)$ the normalized Gegenbauer polynomials. The reproducing kernel function of $\overset{*}{\mathbb{H}}{}_\mu^r(S^{r-1})$ is given by $G_\mu(xy) = \dfrac{2\mu + r - 2}{(r-2)\omega_{r-1}} \cdot C_\mu^{\frac{r-2}{2}}(xy)$, the normalized reproducing kernel $\tilde{G}_\mu$ is defined by $\tilde{G}_\mu := G_\mu / G_\mu(1)$.

Let $Y \in \{C(S^{r-1}), L^2(S^{r-1}), L^p(S^{r-1})\}$. For $f \in Y$ let

$$L(f,x) = \sum_{\nu=0}^{\infty} \big(\Lambda_\nu f\big)(x) = \sum_{\nu=0}^{\infty} \int_{S^{r-1}} f(y) G_\nu(xy) dy \tag{1.1}$$

be the Laplace–series of f, where $\big(\Lambda_\nu f\big)(x) := \int_{S^{r-1}} f(y) G_\nu(x,y) dy$ is the orthogonal projection of f onto $\overset{*}{\mathbb{H}}{}_\nu^r(S^{r-1})$ and the partial sums $L_\mu(f,x) = \sum_{\nu=0}^{\mu} \big(\Lambda_\nu f\big)(x)$ are the orthogonal projections of f onto $\mathbb{P}_\mu^r(S^{r-1})$.

Whereas for $Y = L^2(S^{r-1})$ it is known that the partial sums $L_\mu(f,x)$ converge to f in norm, no convergence is obtained for $Y = C(S^{r-1})$ or $Y = L^p(S^{r-1})$ for $p \geq 2 + \dfrac{r}{r-2}$ and $p \leq 2 - \dfrac{2}{r}$ (see e.g. [1]p.211). Applying a summability method the situation changes.

2 Summability methods

Let $A = (a_{\mu\nu})_{\mu,\nu \in \mathbb{N}_0}$ be an infinite matrix for which the elements $a_{\mu\nu} \in \mathbb{R}$ fulfil the following properties.

(i) $a_{\mu\nu} = 0$ for $\nu > \mu$,
(ii) $\lim_{\mu \to \infty} a_{\mu\nu} = 1$ for $\nu \in \{0, 1\}$,
(iii) $K_\mu(\xi) \geq 0$ for $-1 \leq \xi \leq 1$, where $K_\mu := \sum_{\nu=0}^{\mu} a_{\mu\nu} G_\nu$.

If with the aid of a summability method the kernel G_ν in (1.1) is substituted by a kernel

$$K_\mu = \sum_{\nu=0}^{\mu} a_{\mu\nu} G_\nu \tag{2.1}$$

then the operator L^A defined by the transformed series

$$L^A(f,x) = \lim_{\mu \to \infty} \int_{S^{r-1}} f(y) K_\mu(x,y) dy \tag{2.2}$$

can be shown to converge pointwise to the identity provided that for the kernel K_μ the properties (i)–(iii) of the matrix A are valid.

Remark 2.1 *The coefficients $a_{\mu\nu}$ can be obtained from*

$$a_{\mu\nu} = \left(L_\mu^A \tilde{G}_\nu(t.)\right)(t) = \int_{S^{r-1}} \tilde{G}_\nu(tx) K_\mu(tx) d\omega(x), \quad t \in S^{r-1}.$$

For A being the matrix of the Cesaro-means the proof was given by Kogbetlianz [4] first. Berens et al. [1] give a proof for Cesaro-means as well as for Abel-Poisson-means. They also prove results on the order of convergence and the corresponding saturation classes. The convergence proof for Newman-Shapiro operators ($Y = C(S^{r-1})$) can be found in Reimer [7].

2.1 Cesaro-means

For Cesaro–means the coefficients $a_{\mu\nu}$ in the summability method have to be chosen as

$$a_{\mu\nu} = \frac{(1)_\mu}{(k+1)_\mu} \frac{(k+1)_{\mu-\nu}}{(1)_{\mu-\nu}}, \tag{2.3}$$

where $(p)_q = p \cdot (p+1) \cdot \ldots \cdot (p+q-1)$ denotes the Pochammer symbol. Then the kernels K_μ in (iii) take on the form

$$K_\mu = \frac{(1)_\mu}{(k+1)_\mu} \sum_{\nu=0}^{\mu} \frac{(k+1)_{\mu-\nu}}{(1)_{\mu-\nu}} G_\nu. \tag{2.4}$$

Convergence of the transformed Laplace-series (2.2) is valid for $k > (r-2)/2$; for $k \geq r-1$ the operators even are positive (see Kogbetlianz [4]).

2.2 Newman-Shapiro summability method

In [8] Reimer considers kernel polynomials

$$K_{2\nu+1}(\xi) := K_{2\nu}(\xi) := g_{\nu+1} \left[\frac{G_{\nu+1}(\xi)}{\xi - \eta_{\nu+1}}\right]^2 \tag{2.5}$$

as used by Newman-Shapiro [5]. Here, $\eta_{\nu+1}$ is the largest root of $G_{\nu+1}$ and

$$g_{\nu+1} = (r-2)\omega_{r-1} \cdot \frac{1-\eta_{\nu+1}^2}{(2\nu+r)^2} \binom{\nu+r-2}{r-3}^{-1} = \frac{1-\eta_{\nu+1}^2}{2\nu+r} \cdot \frac{1}{G_{\nu+1}(1)}. \tag{2.6}$$

The coefficients $a_{\mu\nu}$ in the Newman-Shapiro operators can be calculated to be

$$\begin{aligned} a_{\mu\nu} \;=\; & g_{\nu+1} \cdot \sum_{j=0}^{\nu} \sum_{l=0}^{\nu} \frac{(2\nu+r)^2}{(\nu+1)^2} \cdot \frac{\tilde{G}_j(\eta_{\nu+1}) \tilde{G}_l(\eta_{\nu+1})}{\left(\tilde{G}_\nu(\eta_{\nu+1})\right)^2} \cdot \frac{(j+\lambda)(l+\lambda)}{\omega_{r-1}\lambda^2} \\ & \cdot \sum_{k=0}^{\min\{j,l\}} \frac{(\lambda)_k}{(1)_k} \frac{(\lambda)_{j-k}}{(1)_{j-k}} \frac{(\lambda)_{l-k}}{(1)_{l-k}} \frac{(1)_{j+l-2k}}{(2\lambda)_{j+l-2k}} \frac{(2\lambda)_{j+l-k}}{(\lambda+1)_{j+l-k}} \cdot \delta_{\nu,j+l-2k}, \end{aligned} \tag{2.7}$$

where $\delta_{\nu,j+l-2k}$ denotes the Kronecker delta and $\lambda = \frac{r-2}{2}$.

The matrix A defined by the Newman-Shapiro operators fulfils the properties (i)–(iii) (see Reimer [8]).

Remark 2.2 *The corresponding partial sum operators L_μ^A are nonnegative with positive $a_{\mu\nu}$. For continuous and differentiable functions even more is valid (see Reimer [8]): whereas for continuous functions the approximation error is of order $O(\mu^{-1})$, functions $F \in C^j(S^{r-1})$, $j \in \{1,2\}$, have an error of order $O(\mu^{-j})$.*

3 Application to tomography

The Radon transform $\mathcal{R} : C(B^r) \to C(Z^r)$ is defined by

$$\big(\mathcal{R}F\big)(s,t) := \int\limits_{\substack{v \perp t \\ v^2 \le 1-s^2}} F(st+v)dv, \quad F \in C(B^r), \quad (s,t) \in Z^r, \tag{3.1}$$

which means that the Radon transform $\mathcal{R}$ of F is determined by integrating F over all hyperplanes of dimension $r-1$. This map can also be defined for functions in $L^1(\mathbb{R}^r)$, $L^2(B^r)$, the Schwartz space $\mathcal{S}(\mathbb{R}^r)$ or some Sobolev spaces. $\mathcal{R}$ is continuous on all of these spaces, whereas the inverse $\mathcal{R}^{-1}$ is only continuous on $\mathcal{S}(\mathbb{R}^r)$ and on the Sobolev spaces.

For polynomials it is known that

$$\big(\mathcal{R}C_\mu^{\frac{r}{2}}(a.)\big)(s,t) = \tilde{C}_\mu^{\frac{r}{2}}(s)C_\mu^{\frac{r}{2}}(at), \quad a \in S^{r-1}, \quad (s,t) \in Z^r \tag{3.2}$$

(see Davison, Grünbaum [2]) and, more generally,

$$\big(\mathcal{R}P_m\big)(s,t) = \tilde{C}_\mu^{\frac{r}{2}}(s)P_m(t), \quad (s,t) \in Z^r, \tag{3.3}$$

where the polynomials $P_m \in \overset{*}{I\!P}{}_\mu^r(S^{r-1})$ are generated by the Gegenbauer polynomials, i.e. $\frac{1}{\omega_{r-1}}C_\mu^{\frac{r}{2}}(ax) = \sum_{|m|=\mu} a^m P_m(x)$. These polynomials P_m, $|m| = \mu$, are known to constitute a basis for $\overset{*}{I\!P}{}_\mu^r(S^{r-1})$.

Let $V_\mu^r := span\{P_m : |m| = \mu\}$. Since the Gegenbauer polynomials $C_\nu^{\frac{r}{2}}$ can also be interpreted as the reproducing kernel of $\overset{*}{I\!H}{}_\mu^{r+2}(S^{r+1})$, the orthogonal projection F_ν of $F \in C(B^r)$ onto $V_\nu^r(B^r)$ can be identified with the orthogonal projection of F onto $\overset{*}{I\!H}{}_\nu^{r+2}(S^{r+1})$ (see Reimer [7] for details). Thus the theory of Laplace series can be used here for the reconstruction of F from its Radon transform.

Let A be a matrix transformation as introduced in Section 2 and let F_ν be the orthogonal projection of F onto $V_\nu^r(B^r)$. Then according to the summability theory of Laplace series $F = \lim_{\mu\to\infty} \sum_{\nu=0}^{\mu} a_{\mu\nu}F_\nu$. Since the Radon tranform is linear and continuous there is $\mathcal{R}F = \lim_{\mu\to\infty} \sum_{\nu=0}^{\mu} a_{\mu\nu}\mathcal{R}F_\nu$.

It can be shown that (see Reimer [7])

$$F_\nu(x) = \lambda_{\nu,r}\,\frac{\omega_{r-2}}{r-1}\int_{Z^r} \big(\mathcal{R}F\big)(s,t)C_\nu^{\frac{r}{2}}(s)C_\nu^{\frac{r}{2}}(tx)d(s,t), \tag{3.4}$$

where

$$\lambda_{\nu,r} = \frac{(r-1)\,C_\nu^{\frac{r}{2}}(1)}{\omega_{r-1}\cdot\omega_{r-2}}\int_{-1}^{1}\Big(C_\nu^{\frac{r}{2}}(s)\Big)^2(1-s^2)^{\frac{r-1}{2}}ds = \frac{2\nu+r}{\omega_{r-1}^2}. \tag{3.5}$$

¿From this, after some lengthy calculation using the adjoint operator of $\mathcal{R}$ (which essentially is the inverse operator of $\mathcal{R}$), the reconstruction formula follows

$$F(x) = \lim_{\mu\to\infty} \sum_{\nu=0}^{\mu} a_{\mu\nu}\lambda_{\nu,r} \int_{S^{r-1}} \int_{-1}^{1} (\mathcal{R}F)(s,t) C_\nu^{\frac{r}{2}}(s) C_\nu^{\frac{r}{2}}(tx) ds dt. \tag{3.6}$$

Because of the identification of the orthogonal projection of F onto $V_\nu^r(B^r)$ and onto $\overset{*}{I\!H}_\nu^{r+2}(S^{r+1})$, convergence of the Cesaro-means follows for $k > r/2$, and positivity of the operators is valid for $k \geq r+1$. For the same reason the coefficients $a_{\mu\nu}$ in the Newman-Shapiro summability method have to be calculated for $\lambda = \frac{(r+2)-2}{2} = \frac{r}{2}$.

4 Numerical implementation

For the reconstruction of F formula (3.6) was used. As soon as the Radon transform of F is known, the numerical implementation in principle reduces to a stable evaluation of the Gegenbauer polynomials and a suitable approximation of the integrals in (3.6). The Gegenbauer polynomials were evaluated by their recurrence relation (see Szegö [11]) which is known to be numerically very stable. The coefficients $a_{\mu\nu}$ for the Cesaro-means and the Newman-Shapiro operators were computed with the aid of formula (2.3) and (2.7), respectively. The factor $\lambda_{\nu,r}$ was obtained by (3.5). Since the calculation of $a_{\mu\nu}$ for the Newman-Shapiro operators is very time consuming (more than 10 hours for $\mu > 100$) these coefficients were stored before the main computation was started.

Since the integrand in (3.6) is a polynomial of degree $\nu + 2$ with respect to s (see (3.6) together with (5.1)), $\int_{-1}^{1} .. ds$ was approximated by a Gaussian-Legendre quadrature of degree $\mu/2 + 1$. This choice ensures that for the evaluation of $\mathcal{R}F(s,t)$ enough evaluations with respect to s are performed and that the integral is evaluated exactly within numerical precision.

For the quadrature on S^{r-1} first an interpolatory quadrature as introduced in [6]p.132 was used. The weights of such a quadrature formula are obtained as solutions of a linear system of equations $GA = e$, where $e = (1,\ldots,1)^T \in I\!R^N$, $N = \dim \overset{*}{I\!P}_\mu^r(S^{r-1})$, $A = (A_1,\ldots,A_N)^T$ the vector of weights and

$$G = \frac{1}{\omega_{r-1}} (C_\mu^{\frac{r}{2}}(x_j x_k) + C_{\mu-1}^{\frac{r}{2}}(x_j x_k))_{j,k=1}^N.$$

The points were chosen to be regularly distributed on latitudes of the sphere.

For $\mu \geq 70$ in the computation of the weights computational problems occured because of a lack of memory. Apart from this problem, several weights turned out to be negative which led to oscillations of the reconstruction. Therefore, this interpolatory quadrature was substituted by a product–Gauss formula for the sphere S^{r-1} as suggested by Stroud [10]p. 41. The points and weights of the Gaussian quadrature were computed by the MATLAB program `qrule.m` which is available via internet from the Mathworks Inc. The number of points of the product Gauss formula is $N = 2M^{r-1}$ where $M = \mu/2 + 1$ is the number of points used in each direction, i.e. $N = 2M^2$ for $r = 3$.

All codes for computation were written in MATLAB 6. The actual computation took place on a SUN Ultra10 with 256 MB main memory, 691 MB virtual memory and SUN OS operating system release 5.7. To increase the computatinal speed all parts of the MATLAB code were written with as few for-loops as possible. This gave an improvement in speed of a factor > 500.

5 Computational results

The theoretical results have been applied to the so called Shepp-Logan phantom which is usually used as a test function for tomographic reconstrution algorithms. It is a three dimensional model of a human head consisting of 10 ellipsoids (see Shepp [9]) which were shrinked here to fit into the unit sphere S^2. Figure 1 shows a cut at $x_3 = 0.2721$.

Let $a_1^{(j)}, a_2^{(j)}, a_3^{(j)}$, $j = 1, \ldots, 10$, denote the axes of the j-th ellipsoid, $d^{(j)}$ denote its density value and $s_2^{(j)} - s_1^{(j)}$ the diameter of the ellipsoid in the direction of $t \in S^2$. Since the Radon transform is linear, the Radon transform of the Shepp-Logan phantom can be calculated to be

$$\mathcal{R}F(s,t) = \sum_{j=1}^{10} \pi d^{(j)} a_1^{(j)} a_2^{(j)} a_3^{(j)} (s - s_1^{(j)})(s_2^{(j)} - s) \left(\frac{s_2^{(j)} - s_1^{(j)}}{2} \right)^{-3/2} \tag{5.1}$$

Figure 2 shows the reconstruction results according to formula (3.6) for Cesaro-means of index $k = 4$ and for Newman–Shapiro operators.

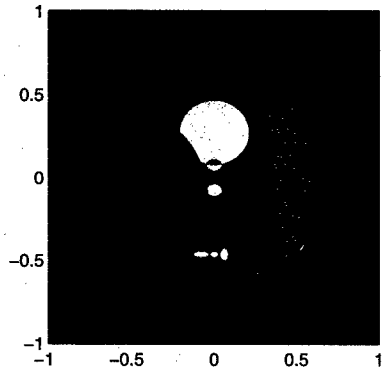


FIG. 1. Shepp-Logan phantom.

The values $k = 1.6$ and $k = 2$ were tested, too, but for high degrees of μ no convergent behaviour could be observed.
For Cesaro-means with $k = 4$ and for Newman-Shapiro operators Figure 2 clearly shows an improving behaviour of the reconstructions for increasing μ.
The Newman-Shapiro operators show a better convergence and for $\mu \geq 150$ even the small structures in the original head can be detected in the reconstruction. It can be expected that for higher degrees of μ this behaviour will become more evident.

Unfortunately, for $\mu \geq 170$ the computation of the coefficients $a_{\mu\nu}$ for the Newman-Shapiro operators caused some numerical problems so that the calculations were stopped with $\mu = 160$. Although the numerical results look quite promising, the drawback in the reconstruction is the computational time. For $\mu = 160$ the computation took 27.5 hours for the Radon transform and 31 hours for the evaluation at the points $x \in [-1, 1]^2$. The evaluation was done on an equidistant grid of 200×200 points.

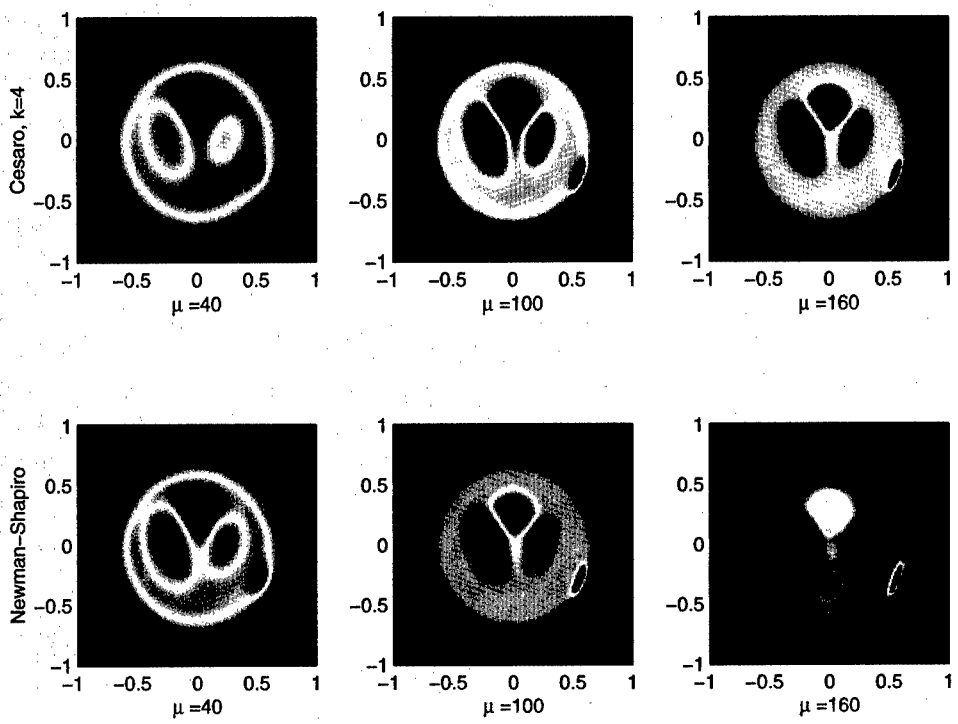


FIG. 2. reconstruction of the Shepp-Logan phantom.

In principle there is no problem to produce three dimensional reconstructions. The evaluation points x only have to be chosen from a grid in $[-1,1]^3$. Because of the time consuming calculations this was not done here, yet.

Bibliography

1. H. Berens, P.L. Butzer, S. Pawelke, Limitierungsverfahren von Reihen mehrdimensionaler Kugelfunktionen und deren Saturationsverhalten, Publ. Res. Inst. Math. Sci. Ser. A 4 (1969) 201–268.
2. M.E. Davidson, F.A. Grünbaum, Tomographic reconstruction with arbitrary directions, Comm. Pure Appl. Math. 34 (1981) 77–119.
3. S.R. Deans, *The Radon transform and some of its applications*, Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1983.
4. E. Kogbetlianz, Recherches sur la summabilité des séries ultrasphériques par la méthode de moyenne arithmétiques, J. de Math. pure et appl. 9(3) (1924) 107–187.
5. D.J. Newman, H.S. Shapiro, Jackson's theorem in higher dimensions. In: *On approximation theory*, eds. P.L. Butzer, J. Korevaar, Birkhäuser Verlag, Basel, 1964, pp. 208–219.
6. M. Reimer, *Constructive theory of multivariate functions.* BI Wissenschaftsverlag,

Mannheim, Wien, Zürich, 1990.
7. M. Reimer, Radon–transform, Laplace–series and matrix–transforms, Comm. Appl. Analysis 1 (1997) 337–349.
8. M. Reimer, Generalized hyperinterpolation on the sphere and the Newman–Shapiro operators, submitted.
9. L.A. Shepp, Computerized tomography and nuclear magnetic resonance, J. Comp. Ass. Tomography 4 (1980) 94–107.
10. A.H. Stroud, *Approximate calculation of multiple integrals.* Englewood Cliffs, NJ: Prentice Hall 1971.
11. G. Szegö, *Orthogonal polynomials*, Amer. Math. Soc., Providence 1991.

A unified approach to fast algorithms of discrete trigonometric transforms

Manfred Tasche

University of Rostock, Department of Mathematics, D-18051 Rostock, Germany.
manfred.tasche@mathematik.uni-rostock.de

Hansmartin Zeuner

Medical University of Lübeck, Institute of Mathematics, Wallstraße 40, D-23560 Lübeck, Germany.
zeuner@math.mu-luebeck.de

Abstract

We present a unified approach to fast algorithms of various discrete trigonometric transforms. With the help of so-called Euler formulas we describe an elegant and useful connection between Fourier matrices and trigonometric matrices. It is known that FFTs are closely related to the factorizations of the unitary Fourier matrix into a product of unitary sparse matrices. Using these Euler formulas and FFTs, we obtain fast algorithms of discrete trigonometric transforms. As a further consequence of these Euler formulas and Gaussian sums, we compute all eigenvalues of some trigonometric matrices.

1 Introduction.

The fast Fourier transform (FFT) and related algorithms for orthogonal trigonometric transforms are essential tools for practical computations. Special discrete trigonometric transforms are the discrete Hartley transforms (DHT), discrete cosine transforms (DCT), and the discrete sine transforms (DST) of various types. These transforms have found important applications in approximation methods with Chebyshev polynomials, quadrature methods of Clenshaw–Curtis type (see [3]), signal processing, and image compression (see [4, 6, 9]).

Euler formulas describe the algebraic connection between Fourier matrices of a certain type and corresponding cosine and sine matrices. Using these formulas, FFTs can be transformed into fast and stable algorithms for the DCT and DST. Further, from these Euler formulas the orthogonality of various trigonometric matrices follows immediately. For simplicity we consider only symmetric trigonometric matrices, i.e. Fourier and Hartley matrices of type I and IV as well as cosine and sine matrices of type I, IV, V and VIII.

This paper is organized as follows; first we introduce generalized Fourier matrices. New Euler formulas for these matrices describe a close connection with various orthogonal Hartley, cosine and sine matrices. These results simplify and extend former results

of [9], pp. 83–96. Applying these Euler formulas and FFTs, we obtain fast algorithms of discrete trigonometric transforms. As a further consequence of these formulas and Gaussian sums, we can compute all eigenvalues of orthogonal symmetric trigonometric matrices.

2 Euler formulas for Fourier matrices of type I

Let $N \geq 2$ be a given integer. The *Fourier matrix of type* I is the classical Fourier matrix defined in unitary form

$$\boldsymbol{F}_N^{\mathrm{I}} := \frac{1}{\sqrt{N}} \big(\omega_N^{jk}\big)_{j,k=0}^{N-1}$$

with $\omega_N := \exp(-2\pi \mathrm{i}/N)$. Note that the Gaussian sum (see [5], pp. 326–330) yields the trace of $\boldsymbol{F}_N^{\mathrm{I}}$:

$$\operatorname{tr} \boldsymbol{F}_N^{\mathrm{I}} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{j^2} = \frac{1+\mathrm{i}^N}{1+\mathrm{i}}. \tag{2.1}$$

Closely related with type I Fourier matrices are the *cosine* and *sine matrices of types* I and V:

$$\begin{aligned}
\boldsymbol{C}_{N+1}^{\mathrm{I}} &:= \sqrt{\frac{2}{N}} \Big(\varepsilon_j^N \varepsilon_k^N \cos \frac{jk\pi}{N}\Big)_{j,k=0}^{N}, \\
\boldsymbol{S}_{N-1}^{\mathrm{I}} &:= \sqrt{\frac{2}{N}} \Big(\sin \frac{(j+1)(k+1)\pi}{N}\Big)_{j,k=0}^{N-2}, \\
\boldsymbol{C}_{N+1}^{\mathrm{V}} &:= \frac{2}{\sqrt{2N+1}} \Big(\varepsilon_j^{N+1} \varepsilon_k^{N+1} \cos \frac{2jk\pi}{2N+1}\Big)_{j,k=0}^{N}, \\
\boldsymbol{S}_N^{\mathrm{V}} &:= \frac{2}{\sqrt{2N+1}} \Big(\sin \frac{2(j+1)(k+1)\pi}{2N+1}\Big)_{j,k=0}^{N-1}.
\end{aligned}$$

Here we set $\varepsilon_j^N := \sqrt{2}/2$ for $j \in \{0, N\}$ and $\varepsilon_j^N := 1$ for $j \in \{1, \ldots, N-1\}$. In this notation a subscript of a matrix denotes the order, while a superscript signifies the type of the matrix. In the following, $\boldsymbol{I}_N$ denotes the identity matrix and $\boldsymbol{J}_N$ the counteridentity matrix, which has the columns of $\boldsymbol{I}_N$ in reverse order. Blanks in a block matrix indicate blocks of zeros. The direct sum of matrices $\boldsymbol{A}, \boldsymbol{B}$ will be denoted by $\boldsymbol{A} \oplus \boldsymbol{B}$. Defining the orthogonal matrices

$$\boldsymbol{P}_{2N}^{\mathrm{I}} := \frac{1}{\sqrt{2}} \begin{pmatrix} \sqrt{2} & & 0 & \\ & \boldsymbol{I}_{N-1} & & \boldsymbol{I}_{N-1} \\ 0 & & \sqrt{2} & \\ & \boldsymbol{J}_{N-1} & & -\boldsymbol{J}_{N-1} \end{pmatrix}, \quad \boldsymbol{P}_{2N+1}^{\mathrm{V}} := \frac{1}{\sqrt{2}} \begin{pmatrix} \sqrt{2} & & \\ & \boldsymbol{I}_N & \boldsymbol{I}_N \\ & \boldsymbol{J}_N & -\boldsymbol{J}_N \end{pmatrix},$$

we obtain for Fourier matrices of type I the following Euler formulas:

Theorem 2.1 *Depending on whether the order of the Fourier matrix of type* I *is even or odd, we have*

$$(\boldsymbol{P}_{2N}^{\mathrm{I}})^{\mathrm{T}} \boldsymbol{F}_{2N}^{\mathrm{I}} \boldsymbol{P}_{2N}^{\mathrm{I}} = \boldsymbol{C}_{N+1}^{\mathrm{I}} \oplus (-\mathrm{i}) \boldsymbol{S}_{N-1}^{\mathrm{I}}, \tag{2.2}$$

$$(\boldsymbol{P}_{2N+1}^{\mathrm{V}})^{\mathrm{T}}\boldsymbol{F}_{2N+1}^{\mathrm{I}}\boldsymbol{P}_{2N+1}^{\mathrm{V}} = \boldsymbol{C}_{N+1}^{\mathrm{V}} \oplus (-\mathrm{i})\boldsymbol{S}_{N}^{\mathrm{V}}. \tag{2.3}$$

Proof: It is obvious that $(\boldsymbol{P}_{2N}^{\mathrm{I}})^{\mathrm{T}}\boldsymbol{P}_{2N}^{\mathrm{I}} = \boldsymbol{I}_{2N}$. Splitting $\boldsymbol{F}_{2N}^{\mathrm{I}}$ into four blocks

$$\boldsymbol{F}_{2N}^{\mathrm{I}} = \frac{1}{\sqrt{2N}} \begin{pmatrix} \left(\omega_{2N}^{jk}\right)_{j,k=0}^{N} & \left(\omega_{2N}^{j(N+k+1)}\right)_{j,k=0}^{N,N-2} \\ \left(\omega_{2N}^{(N+j+1)k}\right)_{j,k=0}^{N-2,N} & \left(\omega_{2N}^{(N+j+1)(N+k+1)}\right)_{j,k=0}^{N-2} \end{pmatrix}$$

and using the classical Euler formula $\exp(-\mathrm{i}x) = \cos x - \mathrm{i}\sin x$, we obtain (2.2) by blockwise computation of $(\boldsymbol{P}_{2N}^{\mathrm{I}})^{\mathrm{T}}\boldsymbol{F}_{2N}^{\mathrm{I}}\boldsymbol{P}_{2N}^{\mathrm{I}}$. The proof of (2.3) is similar. □

Remark 2.2 An analogous result to (2.2) can be found in [9], pp. 85–90, but with a complex matrix instead of $\boldsymbol{P}_{2N}^{\mathrm{I}}$. Compare also with [1]. The Euler formula (2.3) is new. Note that the results and their proofs are simpler than in [9], pp. 85–90 and [1].

Corollary 2.3 *The matrices* $\boldsymbol{C}_{N+1}^{\mathrm{I}}, \boldsymbol{S}_{N-1}^{\mathrm{I}}, \boldsymbol{C}_{N+1}^{\mathrm{V}}, \boldsymbol{S}_{N}^{\mathrm{V}}$ *are orthogonal.*

Proof: Since $\boldsymbol{F}_{2N}^{\mathrm{I}}$ is unitary and $\boldsymbol{P}_{2N}^{I}$ is orthogonal, $\boldsymbol{C}_{N+1}^{\mathrm{I}} \oplus (-\mathrm{i})\boldsymbol{S}_{N-1}^{\mathrm{I}}$ is unitary by (2.2). Hence the real matrices $\boldsymbol{C}_{N+1}^{\mathrm{I}}$ and $\boldsymbol{S}_{N-1}^{\mathrm{I}}$ are orthogonal. Other proofs can be found in [4], pp. 12–16 and [6].

The proof for the type V matrices uses (2.3) and follows similar lines. □

Remark 2.4 Results analogous to (2.2) and (2.3) are true for the *Hartley matrix of type* I (see [9], pp. 77–80 and [8], pp. 224–227)

$$\boldsymbol{H}_{N}^{\mathrm{I}} := \frac{1}{\sqrt{N}}\left(\operatorname{cas}\frac{jk\pi}{N}\right)_{j,k=0}^{N-1}$$

with $\operatorname{cas} x := \cos x + \sin x$. Then we obtain the formulas

$$(\boldsymbol{P}_{2N}^{\mathrm{I}})^{\mathrm{T}}\boldsymbol{H}_{2N}^{\mathrm{I}}\boldsymbol{P}_{2N}^{\mathrm{I}} = \boldsymbol{C}_{N+1}^{\mathrm{I}} \oplus \boldsymbol{S}_{N-1}^{\mathrm{I}}, \tag{2.4}$$

$$(\boldsymbol{P}_{2N+1}^{\mathrm{V}})^{\mathrm{T}}\boldsymbol{H}_{2N+1}^{\mathrm{I}}\boldsymbol{P}_{2N+1}^{\mathrm{V}} = \boldsymbol{C}_{N+1}^{\mathrm{V}} \oplus \boldsymbol{S}_{N}^{\mathrm{V}}. \tag{2.5}$$

The Euler formula (2.2) can be used for fast and numerically stable computations of DCTs and DSTs of type I: Let $\boldsymbol{x} \in \mathbb{R}^{N+1}$ and $\boldsymbol{y} \in \mathbb{R}^{N-1}$ with $N = 2^t$ $(t \geq 2)$ and set $\boldsymbol{z} := \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} \in \mathbb{R}^{2N}$. Since $\boldsymbol{P}_{2N}^{\mathrm{I}}\boldsymbol{z}$ is real, we can apply Edson's algorithm for the FFT of *real* data (see [8], pp. 215–223 and [7]). The output of the conjugate even result is in the form $\boldsymbol{U}_{2N}\boldsymbol{F}_{2N}^{\mathrm{I}}(\boldsymbol{P}_{2N}^{\mathrm{I}}\boldsymbol{z})$ where $\boldsymbol{U}_{2N} := (\boldsymbol{I}_{N+1} \oplus (-\mathrm{i})\boldsymbol{I}_{N-1})\,(\boldsymbol{P}_{2N}^{\mathrm{I}})^{\mathrm{T}}$. Therefore by

$$\boldsymbol{U}_{2N}\boldsymbol{F}_{2N}^{\mathrm{I}}\boldsymbol{P}_{2N}^{\mathrm{I}}\boldsymbol{z} = (\boldsymbol{C}_{N+1}^{\mathrm{I}} \oplus (-1)\boldsymbol{S}_{N-1}^{\mathrm{I}})\boldsymbol{z} = \begin{pmatrix} \boldsymbol{C}_{N+1}^{\mathrm{I}}\boldsymbol{x} \\ -\boldsymbol{S}_{N-1}^{\mathrm{I}}\boldsymbol{y} \end{pmatrix}$$

we have calculated $\boldsymbol{C}_{N+1}^{\mathrm{I}}\boldsymbol{x}$ and $\boldsymbol{S}_{N-1}^{\mathrm{I}}\boldsymbol{y}$ simultaneously using $5Nt$ flops.

If we have to use an FFT with *complex* data, we combine real data vectors $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{N+1}$ resp. $\boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^{N-1}$ into the complex vector $\boldsymbol{z}' := \begin{pmatrix} \boldsymbol{x} + \mathrm{i}\boldsymbol{x}' \\ \boldsymbol{y} + \mathrm{i}\boldsymbol{y}' \end{pmatrix}$. Then we can compute two DCTs $\boldsymbol{C}_{N+1}^{\mathrm{I}}\boldsymbol{x}, \boldsymbol{C}_{N+1}^{\mathrm{I}}\boldsymbol{x}'$ and two DSTs $\boldsymbol{S}_{N-1}^{\mathrm{I}}\boldsymbol{y}, \boldsymbol{S}_{N-1}^{\mathrm{I}}\boldsymbol{y}'$ simultaneously via an FFT of length $2N$ applied to the complex input vector $\boldsymbol{P}_{2N}^{\mathrm{I}}\boldsymbol{z}'$.

In a similar way, the Euler formula (2.3) can be used for fast computations of DCTs and DSTs of type V: For given $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{N+1}$ and $\boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^{N}$ the transformed vectors

$\boldsymbol{C}_{N+1}^{\mathrm{V}}\boldsymbol{x}, \boldsymbol{C}_{N+1}^{\mathrm{V}}\boldsymbol{x}', \boldsymbol{S}_{N}^{\mathrm{V}}\boldsymbol{y}$, and $\boldsymbol{S}_{N}^{\mathrm{V}}\boldsymbol{y}'$ can be calculated at the same time as components of $(\boldsymbol{P}_{2N+1}^{\mathrm{V}})^{\mathrm{T}}\boldsymbol{F}_{2N+1}^{\mathrm{I}}\boldsymbol{P}_{2N+1}^{\mathrm{V}}\boldsymbol{z}' = (\boldsymbol{C}_{N+1}^{\mathrm{V}} \oplus (-\mathrm{i})\boldsymbol{S}_{N}^{\mathrm{V}})\boldsymbol{z}'$ where we use an FFT of length $2N+1$ with complex data $\boldsymbol{P}_{2N+1}^{\mathrm{V}}\boldsymbol{z}'$. If $2N+1 = 3^t$ or more generally, if $2N+1$ is a product of small primes (see [8], pp. 76–101 and [7]) the FFT of length $2N+1$ can be computed very efficiently.

3 Euler formulas for Fourier matrices of type IV

The *Fourier matrix of type* IV, defined by

$$\boldsymbol{F}_{N}^{\mathrm{IV}} := \frac{1}{\sqrt{N}}\Big(\omega_{4N}^{(2j+1)(2k+1)}\Big)_{j,k=0}^{N-1}$$

is related to the Fourier matrix of type I by the formula

$$\boldsymbol{F}_{N}^{\mathrm{IV}} = \omega_{4N}\boldsymbol{W}_{N}\boldsymbol{F}_{N}^{\mathrm{I}}\boldsymbol{W}_{N} \tag{3.1}$$

with $\boldsymbol{W}_{N} := \operatorname{diag}(\omega_{2N}^{k})_{k=0}^{N-1}$ and is therefore unitary. If N is a power of 2 or 3, then $\boldsymbol{F}_{N}^{\mathrm{IV}}$ can be factorized into a product of sparse unitary matrices.

Lemma 3.1 *The trace of the Fourier matrix of type* IV *is equal to*

$$\operatorname{tr}\boldsymbol{F}_{N}^{\mathrm{IV}} = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\omega_{4N}^{(2k+1)^2} = \frac{1-\mathrm{i}^N}{1+\mathrm{i}}. \tag{3.2}$$

Proof: We begin with the generalized Gaussian sum (see [5], p. 330)

$$\frac{1}{\sqrt{N}}\sum_{j=0}^{2N-1}\omega_{4N}^{j^2} = 1-\mathrm{i}$$

which we split into two sums containing even and odd j respectively. Then

$$\frac{1}{\sqrt{N}}\sum_{j=0}^{2N-1}\omega_{4N}^{j^2} = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\omega_{N}^{k^2} + \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\omega_{4N}^{(2k+1)^2} = \operatorname{tr}\boldsymbol{F}_{N}^{\mathrm{I}} + \operatorname{tr}\boldsymbol{F}_{N}^{\mathrm{IV}},$$

and the results follows by (2.1). □

Now we introduce *cosine* and *sine matrices of type* IV and VIII which are closely related with the Fourier matrix of type IV:

$$\begin{aligned}
\boldsymbol{C}_{N}^{\mathrm{IV}} &:= \sqrt{\frac{2}{N}}\Big(\cos\frac{(2j+1)(2k+1)\pi}{4N}\Big)_{j,k=0}^{N-1},\\
\boldsymbol{S}_{N}^{\mathrm{IV}} &:= \sqrt{\frac{2}{N}}\Big(\sin\frac{(2j+1)(2k+1)\pi}{4N}\Big)_{j,k=0}^{N-1},\\
\boldsymbol{C}_{N}^{\mathrm{VIII}} &:= \frac{2}{\sqrt{2N+1}}\Big(\cos\frac{(2j+1)(2k+1)\pi}{2(2N+1)}\Big)_{j,k=0}^{N-1},\\
\boldsymbol{S}_{N+1}^{\mathrm{VIII}} &:= \frac{2}{\sqrt{2N+1}}\Big(\varepsilon_{j+1}^{N+1}\varepsilon_{k+1}^{N+1}\sin\frac{(2j+1)(2k+1)\pi}{2(2N+1)}\Big)_{j,k=0}^{N}.
\end{aligned}$$

As above we define orthogonal matrices

$$\boldsymbol{P}_{2N}^{\mathrm{IV}} := \frac{1}{\sqrt{2}} \begin{pmatrix} \boldsymbol{I}_N & \boldsymbol{I}_N \\ -\boldsymbol{J}_N & \boldsymbol{J}_N \end{pmatrix}, \quad \boldsymbol{P}_{2N+1}^{\mathrm{VIII}} := \frac{1}{\sqrt{2}} \begin{pmatrix} \boldsymbol{I}_N & \boldsymbol{I}_N & \\ & & \sqrt{2} \\ -\boldsymbol{J}_N & \boldsymbol{J}_N & \end{pmatrix}.$$

Theorem 3.2 *For the Fourier matrix of type* IV *and even resp. odd order, we obtain the following Euler formulas:*

$$\left(\boldsymbol{P}_{2N}^{\mathrm{IV}}\right)^{\mathrm{T}} \boldsymbol{F}_{2N}^{\mathrm{IV}} \boldsymbol{P}_{2N}^{\mathrm{IV}} = \boldsymbol{C}_N^{\mathrm{IV}} \oplus (-\mathrm{i}) \boldsymbol{S}_N^{\mathrm{IV}}, \tag{3.3}$$

$$\left(\boldsymbol{P}_{2N+1}^{\mathrm{VIII}}\right)^{\mathrm{T}} \boldsymbol{F}_{2N+1}^{\mathrm{IV}} \boldsymbol{P}_{2N+1}^{\mathrm{VIII}} = \boldsymbol{C}_N^{\mathrm{VIII}} \oplus (-\mathrm{i}) \boldsymbol{S}_{N+1}^{\mathrm{VIII}}. \tag{3.4}$$

Proof: Similar to that of Theorem 2.1. □

Corollary 3.3 *The matrices* $\boldsymbol{C}_N^{\mathrm{IV}}, \boldsymbol{S}_N^{\mathrm{IV}}, \boldsymbol{C}_N^{\mathrm{VIII}}$ *and* $\boldsymbol{S}_{N+1}^{\mathrm{VIII}}$ *are orthogonal.*

Remark 3.4 An analogous result to (3.3) can be found in [9], pp. 94–96. Compare also with [1]. Formula (3.4) is new. A different proof of the orthogonality of $\boldsymbol{C}_N^{\mathrm{IV}}$ and $\boldsymbol{S}_N^{\mathrm{IV}}$ can be found in [6].

Remark 3.5 Similar formulas as in Theorem 3.2 are true for the *Hartley matrix of type* IV (see [1, 2])

$$\boldsymbol{H}_N^{\mathrm{IV}} := \frac{1}{\sqrt{N}} \left(\operatorname{cas} \frac{(2j+1)(2k+1)\pi}{2N} \right)_{j,k=0}^{N-1}.$$

Then we have

$$\left(\boldsymbol{P}_{2N}^{\mathrm{IV}}\right)^{\mathrm{T}} \boldsymbol{H}_{2N}^{\mathrm{IV}} \boldsymbol{P}_{2N}^{\mathrm{IV}} = \boldsymbol{C}_N^{\mathrm{IV}} \oplus \boldsymbol{S}_N^{\mathrm{IV}}, \tag{3.5}$$

$$\left(\boldsymbol{P}_{2N+1}^{\mathrm{VIII}}\right)^{\mathrm{T}} \boldsymbol{H}_{2N+1}^{\mathrm{IV}} \boldsymbol{P}_{2N+1}^{\mathrm{VIII}} = \boldsymbol{C}_N^{\mathrm{VIII}} \oplus \boldsymbol{S}_{N+1}^{\mathrm{VIII}}. \tag{3.6}$$

The Euler formulas can be used for a fast and numerically stable computation of DCT and DST of types IV and VIII:

Using (3.3) and (3.1), for arbitrary $\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^N$ the DCTs $\boldsymbol{C}_N^{\mathrm{IV}} \boldsymbol{x}$, $\boldsymbol{C}_N^{\mathrm{IV}} \boldsymbol{x}'$ and DSTs $\boldsymbol{S}_N^{\mathrm{IV}} \boldsymbol{y}$ and $\boldsymbol{S}_N^{\mathrm{IV}} \boldsymbol{y}'$ can be calculated via one FFT of length $2N$ with complex data $\boldsymbol{P}_{2N}^{\mathrm{IV}} \boldsymbol{z}'$ and $\boldsymbol{z}' := \begin{pmatrix} \boldsymbol{x} + \mathrm{i}\boldsymbol{x}' \\ \boldsymbol{y} + \mathrm{i}\boldsymbol{y}' \end{pmatrix}$. If $N = 2^t$, this procedure requires about $10Nt$ operations. Likewise by (3.4), for $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^N, \boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^{N+1}$ the DCTs of type VIII, $\boldsymbol{C}_N^{\mathrm{VIII}} \boldsymbol{x}$, $\boldsymbol{C}_N^{\mathrm{VIII}} \boldsymbol{x}'$ and the DSTs $\boldsymbol{S}_{N+1}^{\mathrm{VIII}} \boldsymbol{y}$, $\boldsymbol{S}_{N+1}^{\mathrm{VIII}} \boldsymbol{y}'$ can be calculated via one FFT of length $2N+1$ with complex data $\boldsymbol{P}_{2N+1}^{\mathrm{VIII}} \boldsymbol{z}'$.

Remark 3.6 The sine, cosine, Hartley, and Fourier matrices considered above enjoy the interesting intertwining relations (see [2]):

$$\begin{aligned} \boldsymbol{C}_{N+1}^{\mathrm{I}} \boldsymbol{J}_{N+1} &= \boldsymbol{\Sigma}_{N+1} \boldsymbol{C}_{N+1}^{\mathrm{I}}, \quad \boldsymbol{S}_{N-1}^{\mathrm{I}} \boldsymbol{J}_{N-1} = \boldsymbol{\Sigma}_{N-1} \boldsymbol{S}_{N-1}^{\mathrm{I}}, \\ \boldsymbol{C}_N^{\mathrm{IV}} \boldsymbol{J}_N &= \boldsymbol{\Sigma}_N \boldsymbol{S}_N^{\mathrm{IV}}, \\ \boldsymbol{H}_N^{\mathrm{I}} \boldsymbol{J}_N' &= \boldsymbol{J}_N' \boldsymbol{H}_N^{\mathrm{I}}, \qquad \boldsymbol{H}_N^{\mathrm{IV}} \boldsymbol{J}_N = \boldsymbol{J}_N \boldsymbol{H}_N^{\mathrm{IV}}, \\ \boldsymbol{F}_N^{\mathrm{I}} \boldsymbol{J}_N' &= \boldsymbol{J}_N' \boldsymbol{F}_N^{\mathrm{I}}, \qquad \mathbf{F}_N^{\mathrm{IV}} \boldsymbol{J}_N = \boldsymbol{J}_N \mathbf{F}_N^{\mathrm{IV}}, \end{aligned} \tag{3.7}$$

with the diagonal matrix $\boldsymbol{\Sigma}_{N+1} := \operatorname{diag}((-1)^k)_{k=0}^N$ and the reflection matrix $\boldsymbol{J}_N' := 1 \oplus \boldsymbol{J}_{N-1}$. Therefore applying (3.7) in the above algorithm, it is also possible to compute

four DCTs (or four DSTs) of type IV and order N via one FFT of length $2N$ with complex data.

4 Eigenvalues of trigonometric matrices

Finally we determine the eigenvalues of trigonometric matrices introduced above. Since the cosine and sine matrices of type I, IV, V and VIII, and the Hartley matrices of type I and IV are real, symmetric and orthogonal, only 1 and -1 are possible eigenvalues. For $x \in \mathbb{R}$ we denote by $\lfloor x \rfloor$ resp. $\lceil x \rceil$ the integer $k \in \mathbb{Z}$ with $k \le x < k+1$ resp. $k-1 < x \le k$.

Theorem 4.1 *The sine and cosine matrices* $\boldsymbol{C}_N^{\mathrm{I}}, \boldsymbol{S}_N^{\mathrm{I}}, \boldsymbol{C}_N^{\mathrm{IV}}, \boldsymbol{S}_N^{\mathrm{IV}}, \boldsymbol{C}_N^{\mathrm{V}}, \boldsymbol{S}_N^{\mathrm{V}}, \boldsymbol{C}_N^{\mathrm{VIII}}$ *and* $\boldsymbol{S}_N^{\mathrm{VIII}}$ *of order* $N \ge 2$ *possess the eigenvalues* 1 *and* -1 *with multiplicities*

$$m(1) = \lceil N/2 \rceil, \quad m(-1) = \lfloor N/2 \rfloor.$$

Proof: Since $\boldsymbol{C}_N^{\mathrm{I}}$ is symmetric and orthogonal, only 1 and -1 can be eigenvalues. Their multiplicities fulfil

$$m(1) + m(-1) = N.$$

On the other hand, since $\boldsymbol{C}_N^{\mathrm{I}}$ and $\boldsymbol{S}_{N-2}^{\mathrm{I}}$ are real, it follows from (2.2) and the trace formula (2.1) that

$$m(1) - m(-1) = \operatorname{tr} \boldsymbol{C}_N^{\mathrm{I}} = \operatorname{Re}\bigl(\operatorname{tr} \boldsymbol{F}_{2N-2}^{\mathrm{I}}\bigr) = \operatorname{Re} \frac{1 + \mathrm{i}^{2N-2}}{1+\mathrm{i}} = \begin{cases} 1 & \text{for odd } N, \\ 0 & \text{for even } N. \end{cases}$$

From these two linear equations we obtain $m(1) = \lceil N/2 \rceil$ and $m(-1) = \lfloor N/2 \rfloor$. In the other cases, the proof is similar. □

From Theorem 4.1 and the Euler formulas (2.2)–(2.3) and (3.3)–(3.4) it follows immediately:

Corollary 4.2 *The Fourier matrices of type* I *and* IV *have only eigenvalues* $1, -1, \mathrm{i}, -\mathrm{i}$ *with multiplicities:*

	$\boldsymbol{F}_{2N}^{\mathrm{I}}$	$\boldsymbol{F}_{2N+1}^{\mathrm{I}}$	$\boldsymbol{F}_{2N}^{\mathrm{IV}}$	$\boldsymbol{F}_{2N+1}^{\mathrm{IV}}$
$m(1)$	$\lfloor N/2 \rfloor + 1$	$\lfloor N/2 \rfloor + 1$	$\lceil N/2 \rceil$	$\lceil N/2 \rceil$
$m(-1)$	$\lceil N/2 \rceil$	$\lceil N/2 \rceil$	$\lfloor N/2 \rfloor$	$\lfloor N/2 \rfloor$
$m(\mathrm{i})$	$\lceil N/2 \rceil - 1$	$\lfloor N/2 \rfloor$	$\lfloor N/2 \rfloor$	$\lceil N/2 \rceil$
$m(-\mathrm{i})$	$\lfloor N/2 \rfloor$	$\lceil N/2 \rceil$	$\lceil N/2 \rceil$	$\lfloor N/2 \rfloor + 1$

From Theorem 4.1 and formulas (2.4)–(2.5) and (3.5)–(3.6) it follows:

Corollary 4.3 *The Hartley matrices of type* I *and* IV *have only eigenvalues* 1 *and* -1 *with the following multiplicities:*

	$\boldsymbol{H}_{2N}^{\mathrm{I}}$	$\boldsymbol{H}_{2N+1}^{\mathrm{I}}$	$\boldsymbol{H}_{2N}^{\mathrm{IV}}$	$\boldsymbol{H}_{2N+1}^{\mathrm{IV}}$
$m(1)$	$2\lfloor N/2 \rfloor + 1$	$N+1$	$2\lceil N/2 \rceil$	$N+1$
$m(-1)$	$2\lceil N/2 \rceil - 1$	N	$2\lfloor N/2 \rfloor$	N

Bibliography

1. V. Britanak and K. R. Rao, *The fast generalized discrete Fourier transform: A unified approach to the discrete sinusoidal transform computation*, Signal Process., 79 (1999), pp. 135–150.
2. G. Heinig and K. Rost, *Hartley transform representations of inverses of real Toeplitz-plus-Hankel matrices*, Numer. Funct. Anal. Optim., 21 (2000), pp. 175–189.
3. J. C. Mason and E. Venturino, *Integration methods of Clenshaw–Curtis type, based on four kinds of Chebyshev polynomials*, in Multivariate Approximation and Splines, G. Nürnberger, J. W. Schmidt, and G. Walz, eds., Basel, 1997, Birkhäuser, pp. 153–165.
4. K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, and Applications*, Academic Press, San Diego, 1990.
5. R. Remmert, *Funktionentheorie* I, Springer, Berlin, 1992.
6. G. Strang, *The discrete cosine transform*, SIAM Rev., 41 (1999), pp. 135–147.
7. M. Tasche and H. Zeuner, *Roundoff error analysis for fast trigonometric transforms*, in Handbook of Analytic–Computational Methods in Applied Mathematics, G. Anastassiou, ed., CRC Press, Boca Raton, 2000, pp. 357–406.
8. C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, 1992.
9. M. V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, A K Peters, Wellesley, 1994.

This volume contains the proceedings of an International Symposium on Algorithms for Approximation Four (A4A4), held at University of Huddersfield from July 15th to 20th, 2001, attended by 106 people from no less than 32 countries. The 54 papers submitted cover a broad range of topics in approximation theory, metrology, orthogonal polynomials, splines, wavelets, radial basis functions, approximation on manifolds, and applications in medical modelling, and the solution of integral and differential equations. All papers were refereed meticulously.