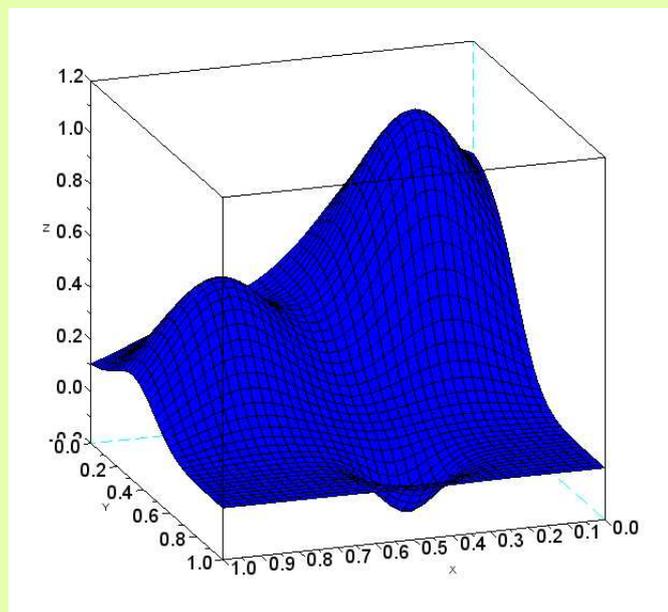


LECTURES ON MULTIVARIATE POLYNOMIAL APPROXIMATION

Stefano De Marchi

Department of Mathematics
University of Padua



Padova, October 2017

These lecture notes are a collections of arguments that briefly present the problem of multivariate polynomial interpolation (and approximation) starting from the univariate setting. The notes have then been used during a short teaching-visit of the author to the University of Göttingen, for the Erasmus Teaching Staff Mobility and also for a part of the course *Approximation Theory and its Applications* given at the University of Padova, for master's students in Mathematics.

People interested on the topic, can refer to the wide literature available that, especially in the last two decades, has grown very fast. If in the univariate polynomial interpolation almost all problems are well settled and solved, in the multivariate setting many things are still unknown. These few pages represent a short introduction to the topic and to some open problems.

The lectures are intended for students with a basic knowledge of numerical methods and advanced preparation on mathematical analysis.

Every lecture provides also a set of exercises solvable by using Matlab. This choice has been done with the aim of making the discussion more interesting from both the numerical and geometrical point of view. Hope the students will be able to solve the proposed excercises for better understanding the topic.

I do hope that after this brief introduction, interested students will be encouraged and also interested in getting into this fascinating world.

Stefano De Marchi
June 2017.

Contents

1	Learning from the univariate case	9
1.1	Univariate polynomial interpolation	9
1.1.1	Barycentric form of the interpolant	11
1.1.2	Interpolation error	12
1.1.3	Characterizations of the polynomial of best approximation	17
1.1.4	How to find the BPA to a function?	22
2	Lebesgue constant	25
2.1	Lebesgue constant	25
2.1.1	Properties of the Lebesgue function $\lambda_n(x)$	27
2.1.2	The Lebesgue constant for specific set of points	28
2.1.3	Optimal interpolation points	34
3	The multivariate case	37
3.1	The Haar-Mairhuber-Curtis theorem	37
3.1.1	Unisolvent set of functions and point sets	38
3.2	Optimal and near-optimal interpolation points	41
3.2.1	Tensor-product Chebyshev-Lobatto and Leja points	41
3.2.2	Dubiner metric and near-optimal interpolation points on $[-1, 1]^2$	42
4	The Padua points	51

4.1	The generating curve of the Padua points	51
4.1.1	Reproducing kernel in Hilbert spaces	53
4.2	The basic Lagrange polynomials and Lebesgue constant	54
4.3	Fast interpolation and cubature at the Padua points	57
4.3.1	Interpolation	57
4.3.2	Cubature	60
4.3.3	Some numerical experiments	64
5	Weakly Admissible Meshes	67
5.1	Weakly Admissible Meshes	68
5.2	Approximate Fekete Points and Discrete Leja Points	69
5.2.1	Extensions and some applications.	76
A	The Stone-Weierstrass theorem	77

List of Figures

1.1	Error $e(x)$ and $e(x) - c$	18
1.2	The function $\sin(4x) \in [-\pi, \pi]$	20
1.3	Plot of the first Chebyshev polynomials	21
2.1	Plot of the Lebesgue functions for $n \leq 10$ for equispaced (left) and Chebyshev-Lobatto (right) points	28
2.2	The set T of Chebyshev points for $n = 9$	30
2.3	Chebyshev and Extended Chebyshev points	31
2.4	Plot of the Lebesgue constants for $n \leq 9$ for equispaced on $[-1,1]$ (left) and $n \leq 10$ for the Chebyshev extrema points U (right)	32
3.1	A closed path between two points	38
3.2	The Chung-Yao construction for unisolvency in \mathbb{R}^2 for $m = 2$ and $N = 6$	40
3.3	Lebesgue constants for tensor-product Chebyshev-Lobatto (TPC) and Leja (TPL) points up to degree 60, compared with the theoretical bound for TPC and with a least-square fitting for TPL.	43
3.4	Left: 496 (i.e. degree 30) quasi-uniform Dubiner (DUB) points for the square; Right: Lebesgue constants for DUB points, quasi-uniform Euclidean (EUC) points and random (RND) points.	45
3.5	Left: Morrow-Patterson (MP), Extended Morrow-Patterson (EMP) and Padua (PD) points, for degree $n = 8$. Right: Padua points for degree $n = 30$	47
3.6	The behaviour of the Lebesgue constants for Morrow-Patterson (MP), Extended Morrow-Patterson (EMP), Padua (PD) points up to degree 60, and their least-squares fitting curves.	48

4.1	Padua points and their generating curve for $n = 4$. The grids of odd and even indices are indicated with different colours and style.	52
4.2	Relative cubature errors versus the number of cubature points (CC = Clenshaw-Curtis, GLL = Gauss-Legendre-Lobatto, OS = Omelyan-Solovyan) for the Gaussian $f(\mathbf{x}) = \exp(- \mathbf{x} ^2)$ (above) and the C^2 function $f(\mathbf{x}) = \mathbf{x} ^3$ (below); the integration domain is $[-1, 1]^2$, the integrals up to machine precision are, respectively: 2.230985141404135 and 2.508723139534059.	65
5.1	Symmetric polar WAMs of the disk for degree $n = 10$ (left) and $n = 11$ (right).	70
5.2	A WAM of the quadrant for even polynomials of degree $n = 16$ (left), and the corresponding WAM of the simplex for degree $n = 8$ (right).	70
5.3	$N = 45$ Approximate Fekete Points (circles) and Discrete Leja Points (asterisks) for degree $n = 8$ extracted from two WAMs of a nonregular convex hexagon (dots).	72
5.4	$N = 28$ AFP (Approximate Fekete Points, circles) and DLP (Discrete Leja Points, asterisks) for degree $n = 6$ extracted from an AM (Admissible Mesh) of a circular sector.	75
5.5	$N = 136$ AFP (circles) and DLP (asterisks) for degree $n = 15$ in a hand-shape polygon.	76

List of Tables

3.1	Tensor-product interpolation errors for the Franke function.	42
3.2	Tensor-product interpolation errors for the function $f_2(x_1, x_2) = (x_1^2 + x_2^2)^{5/2}$	43
3.3	Tensor-product interpolation errors for the function $f_3(x_1, x_2) = (x_1^2 + x_2^2)^{1/2}$	44
3.4	Interpolation errors for the Franke function.	48
3.5	Interpolation errors for the function $f_2(x_1, x_2) = (x_1^2 + x_2^2)^{5/2}$	48
3.6	Interpolation errors for the function $f_3(x_1, x_2) = (x_1^2 + x_2^2)^{1/2}$	49
4.1	Matlab code for the fast computation of the coefficient matrix.	59
4.2	Matlab code for the evaluation of the cubature formula by moments.	61
4.3	Matlab code for the computation of the cubature weights.	63
4.4	CPU time (in seconds) for the computation of the interpolation coefficients at a sequence of degrees (average of 10 runs).	64
4.5	CPU time (in seconds) for the computation of the cubature weights at a sequence of degrees (average of 10 runs).	64
5.1	Lebesgue constants for AFP and DLP extracted from two WAMs of a non-regular convex hexagon (WAM1: barycentric triangulation, WAM2: minimal triangulation; see Fig. 5.3).	73
5.2	Max-norm of the interpolation errors with AFP and DLP extracted from WAM2 for two test functions: $f_1 = \cos(x_1 + x_2)$; $f_2 = ((x_1 - 0.5)^2 + (x_2 - 0.5)^2)^{3/2}$	73

Lecture 1

Learning from the univariate case

We start our lectures recalling the fundamental problem of the univariate interpolation with emphasis to some not commonly known facts.

1.1 Univariate polynomial interpolation

Let $p_n(x) = \sum_{i=0}^n a_i x^i$ be a polynomial of degree n written using the monomial basis $\mathcal{M} = \{1, x, \dots, x^n\}$. i.e. an element of the space $\mathbb{P}_n(\mathbb{R})$.

The interpolation problem consists in finding the unique polynomial $p_n \in \mathbb{P}_n$ such that

$$p_n(x_i) = y_i, \quad i = 0, \dots, n \quad (1.1)$$

are fulfilled for any set of $n + 1$ *distinct* points x_i and values $y_i \in \mathbb{R}$ (or $y_i = f(x_i)$ for some function f of real values). This is equivalent to the solution of the linear system $Va = y$, with V the *Vandermonde matrix* of order $n + 1$.

Because we have chosen x_i distinct, then V is invertible. Indeed,

$$\det(V) = \prod_{i,j=0, i>j}^n (x_i - x_j) \neq 0. \quad (1.2)$$

Exercise 1. *Prove by induction formula (1.2).*

The best situation is when $V = I$, so that the solution of the system is immediate, giving $a_i = y_i$, $i = 0, \dots, n$. This happens when instead of the monomial basis \mathcal{M} we consider the so called *Lagrange basis*, $\mathcal{L} = \{l_0, \dots, l_n\}$ such that $l_i(x_j) = \delta_{i,j}$. The elementary (or fundamental) Lagrange polynomials of degree n are given by

$$l_i(x) := \prod_{i \neq j} \frac{x - x_j}{x_i - x_j}, \quad i, j = 0, \dots, n. \quad (1.3)$$

From the definition above, it follows that

$$p_n(x) = \sum_{i=0}^n l_i(x) y_i.$$

Two consequences

1. $\sum_{i=0}^n l_i(x) = 1$. In fact, the interpolation process is exact on constants.
2. From the property $l_i(x_j) = \delta_{i,j}$, it comes easy to show that

$$l_i(x) = \det(V(x_0, \dots, x_{i-1}, x, x_{i+1}, \dots, x_n)) / \det(V(x_0, \dots, x_n)), \forall i, \quad (1.4)$$

where $V(x_0, \dots, x_n)$ is the **Vandermonde matrix at the points** x_0, \dots, x_n . (*Hint*: consider the case $n = 2$)

The way to show formula (1.4) is this. First, we observe that the elementary Lagrange polynomials at a point x can be written in terms of the monomial basis, that is

$$l_j(x) = \sum_{i=0}^n l_{j,i} x^i, \quad j = 0, \dots, n. \quad (1.5)$$

It is easy to see that by asking that $l_j(x_k) = \delta_{j,k}$, the coefficients of the Lagrange polynomials in (1.5) are the columns of the inverse of the Vandermonde matrix (see next Example)

$$V^{-1} = \begin{pmatrix} l_{0,0} & l_{0,1} & \cdots & l_{0,n} \\ l_{1,0} & l_{1,1} & \cdots & l_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{n,0} & l_{n,1} & \cdots & l_{n,n} \end{pmatrix}.$$

Example 1. Consider the case $n = 1$ we have x_0, x_1 and

$$V = \begin{pmatrix} 1 & 1 \\ x_0 & x_1 \end{pmatrix},$$

then

$$V^{-1} = \frac{1}{x_1 - x_0} \begin{pmatrix} x_1 & -1 \\ -x_0 & 1 \end{pmatrix}.$$

The two columns are indeed the coefficients of the elementary Lagrange polynomials. For instance $l_0(x) = \frac{1}{x_1 - x_0}(x_1 \cdot 1 - 1 \cdot x)$ which gives $l_0(x_0) = 1$, $l_0(x_1) = 0$.

Therefore, letting $\mathbf{m}(x) = (1, x, x^2, \dots, x^n)^T$ the monomial basis and $\mathbf{l}(x)$ the Lagrange basis, formula (1.5) shows that $V^{-1}\mathbf{m}(x) = \mathbf{Vl}(x)$. By using the Cramer's rule, we get $l_j(x)$, $j = 0, \dots, n$, which is formula (1.4).

Observation 1. This way to define the elementary Lagrange polynomials, is particularly useful in the univariate case (to derive coefficients estimates for polynomials, see [36]) and in higher dimensions. In fact, if $\Phi = \{\phi_i, i = 1, \dots, N\}$ is a polynomial basis for the polynomial space $\mathbb{P}_n(\mathbb{R}^d)$ (polynomials of degree $\leq n$ in \mathbb{R}^d), then we can define $V_\Phi(x_0, \dots, x_N)$ (the Vandermonde matrix that uses the basis Φ collocated at the d -dimensional point set x_0, \dots, x_N) and consequently the corresponding elementary Lagrange polynomials.

1.1.1 Barycentric form of the interpolant

Let $\omega_n(x) = (x - x_0) \cdots (x - x_n)$, then

$$l_i(x) = \frac{\omega_n(x)}{(x - x_i)\omega'_n(x_i)} = \omega_n(x) \frac{w_i}{x - x_i},$$

with $w_i = 1/\omega'_n(x_i)$. This is usually referred as the *first barycentric form*.

Moreover, the interpolating polynomial becomes

$$p_n(x) = \omega_n(x) \sum_{i=0}^n \left(\frac{w_i}{x - x_i} \right) y_i. \quad (1.6)$$

Formula (1.6) shows the advantage of the barycentric formulation. In fact, if the $w_i \forall i$ are pre-computed (i.e. once for all), then the evaluation at x of the *new basis* element, $\omega_n(x)w_i/(x - x_i)$, requires $\mathcal{O}(n)$ flops (for evaluating $\omega_n(x)$ and $w_j/(x - x_j)$) instead of $\mathcal{O}(n^2)$ for the evaluation of each $l_i(x)$. This will result in a total $\mathcal{O}(n^2)$ operations instead of $\mathcal{O}(n^3)$ required by the Lagrange basis.

Another advantage is that if we add another interpolation point, x_{n+1} , we have

$$\begin{aligned} w_i^* &= \frac{w_i}{(x_i - x_{n+1})}, \quad i = 0, \dots, n \\ w_{n+1} &= \prod_{i=0}^n \frac{1}{x_{n+1} - x_i}. \end{aligned}$$

Finally, if $g(x) \equiv 1$, then $g(x) = \omega_n(x) \sum_{j=0}^n \frac{w_j}{x - x_j}$. Hence, we get the *second barycentric form*

$$p_n(x) = \sum_{i=0}^n \frac{\left(\frac{w_i}{x - x_i} \right) y_i}{\sum_{i=0}^n \left(\frac{w_i}{x - x_i} \right)}. \quad (1.7)$$

where the new elementary basis elements are

$$b_j(x) = \frac{\frac{w_j}{x - x_j}}{\sum_{i=0}^n \frac{w_i}{x - x_i}}, \quad j = 0, \dots, n.$$

Remark. The barycentric form of the interpolant has the advantage of providing a numerically stable algorithm to evaluate univariate polynomials (see [37, 32], and [47] Chapter 5).

1.1.2 Interpolation error

The main theorem says that the interpolation error depends on the choice of the interpolation points and on the smoothness of the function to be interpolated.

Theorem 1. *Letting $f \in \mathcal{C}[a, b]$ and $X_n = \{x_0, \dots, x_n\}$ the set of interpolation nodes. Then*

$$\|f - p_n\|_\infty \leq (1 + \Lambda_n(X_n)) \|f - p_n^*\|_\infty, \quad n \geq 0 \quad (1.8)$$

where p_n^* is the best interpolating polynomial and $\Lambda_n(X_n) = \max_{x \in [a, b]} \sum_{i=0}^n |l_i(x)|$ the Lebesgue constant.

The existence of p_n^* is ensured by the following result.

Theorem 2. *(cf. [39])*

If V is a normed linear space and W a finite dimensional subspace of V . Then, given $v \in V$ there exist $w^ \in W$ s.t.*

$$\|v - w^*\| \leq \|v - w\|, \quad \forall w \in W. \quad (1.9)$$

Proof. Let v fixed in V and $w \in W$ an arbitrary element. The “point” we are looking for is in the set

$$\{h : h \in W, \|v - h\| \leq \|v - w\|, \forall w \in W\}.$$

But this set is compact in a finite dimensional space so it has a minimum h^* . The conclusion follows. \square .

Example 2. Take $V = \mathcal{C}[a, b]$ equipped with $\|f\| = \max_{x \in [a, b]} |f(x)|$ (uniform or Chebyshev norm), $W = \text{span}\{1, x, \dots, x^n\} := \mathbb{P}_n(\mathbb{R})$. Then by the previous Theorem 2, for all $f \in V$ there exists a best approximation in W , say p_n^* s.t.

$$\|f - p_n^*\| \leq \|f - p_n\|, \quad \forall p_n \in \mathbb{P}_n(\mathbb{R}). \quad (1.10)$$

◇◇

Remark. Another way to express (1.10) is

$$\min_{p \in \mathbb{P}_n} \max_{x \in [a, b]} |f(x) - p(x)| = \max_{x \in [a, b]} |f(x) - p_n^*(x)|, \quad (1.11)$$

introducing in such a way the name of **min-max approximation**.

Question 1. Letting $E_n^*(f) = \|f - p_n^*\|_\infty$. What is the behavior of E_n^* ? In other words, for every $f \in \mathcal{C}(\mathbb{R})$, $\lim_{n \rightarrow \infty} E_n^*(f) = 0$?

A first step for the answer is given by the **first Weierstrass theorem**.

Theorem 3. (W1) Given $f \in \mathcal{C}[a, b]$ and $\epsilon > 0$, there exists a $p_n \in \mathbb{P}_n$ s.t.

$$\|f - p_n\| \leq \epsilon \quad (1.12)$$

Remark. Notice that we are talking about “first” Weierstrass theorem, since there exists a second version that applies to $f \in \mathcal{C}[-\pi, \pi]$, s.t. $f(-\pi) = f(\pi)$ and $t \in \Pi_n$, where Π_n the set of trigonometric polynomials. In fact, just using the transformation $x = \cos(t)$ we can consider $f(t) = g(x(t))$ with $g \in \mathcal{C}[-1, 1]$ and apply the first Weierstrass W1. We point out that the multivariate version of the Theorem ?? is the Stone-Weierstrass theorem (see Appendix A).

Example 3. In the case $[a, b] = [0, 1]$ the **Bernstein polynomial** is a constructive solution of the Weierstrass claim.

$$B_n(f; x) = \sum_{k=0}^n f\left(\frac{k}{n}\right) \underbrace{\binom{n}{k} x^k (1-x)^{n-k}}_{\text{Bernstein polynomial basis}}. \quad (1.13)$$

The following theorem shows that the Bernstein polynomial converges at any point of continuity of f .

Theorem 4. If $f(x)$ is bounded in $[0, 1]$, on every point of continuity we have

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

Moreover, if $f \in \mathcal{C}[0, 1]$ then the limit is uniform.

Question 2. How good is the approximation by elements of $\mathbb{P}_n(R)$?

In practise we need a bound for the error

$$\max_{x \in [0, 1]} |f(x) - B_n(f; x)| = \|f - B_n(f)\|_\infty.$$

This can be done by the use of **modulus of continuity** (MoC).

Definition 1. Let f be defined on $[a, b]$. The MoC of f on $[a, b]$, is defined for all $\delta > 0$

$$\omega(\delta) := \sup_{\substack{x_1, x_2 \in [a, b] \\ |x_1 - x_2| \leq \delta}} |f(x_1) - f(x_2)|. \quad (1.14)$$

Remark. As is clear from the definition, the MoC, depends on δ , f and $[a, b]$, that is $\omega(\delta) = \omega(\delta, f, [a, b])$.

These are the **main properties** of the MoC (cf. e.g. [39, Ch. 1])

- (a) Let $0 < \delta_1 \leq \delta_2$, $\omega(\delta_1) \leq \omega(\delta_2)$. That is the MoC is an increasing function of δ .

(b) A function $f(x)$ is uniformly continuous in $[a, b]$ if and only if $\lim_{\delta \rightarrow 0} \omega(\delta) = 0$.

(c) If $\lambda > 0$, $\omega(\lambda\delta) \leq (1 + \lambda)\omega(\delta)$.

Concerning the property (c). Let $n \in \mathbb{N}$ s.t. $n \leq \lambda < n + 1$. By property (a) we know $\omega(n) \leq \omega(n + 1)$. Now suppose $|x_1 - x_2| \leq (n + 1)\delta$ and $x_1 < x_2$. Subdivide $[x_1, x_2]$ in $n + 1$ equal parts of length $h = (x_2 - x_1)/(n + 1)$ by the $n + 2$ points

$$z_j = x_1 + jh, \quad j = 0, \dots, n + 1.$$

Then

$$|f(x_1) - f(x_2)| = \left| \sum_{j=0}^n (f(z_{j+1}) - f(z_j)) \right| \leq \sum_{j=0}^n |f(z_{j+1}) - f(z_j)| \leq (n + 1)\omega(\delta).$$

Thus, $\omega((n + 1)\delta) \leq (n + 1)\omega(\delta)$. But $n + 1 \leq \lambda + 1$ and the conclusion follows. \square

Theorem 5. *If f is bounded on $[0, 1]$ then*

$$\|f - B_n(f)\|_\infty \leq \frac{3}{2}\omega\left(\frac{1}{\sqrt{n}}\right).$$

Proof. Observe that for $f = 1$, $B_n(1; x) = \sum_{k=0}^n \binom{n}{k} x^k (1 - x)^{n-k} = (x + (1 - x))^n = 1$.

Then,

$$\begin{aligned} |f(x) - B_n(f; x)| &\leq \sum_{k=0}^n |f(x) - f\left(\frac{k}{n}\right)| \binom{n}{k} x^k (1 - x)^{n-k} \\ &\leq \sum_{k=0}^n \omega(|x - k/n|) \binom{n}{k} x^k (1 - x)^{n-k}. \end{aligned}$$

By property (c) of the MoC,

$$\omega\left(\left|x - \frac{k}{n}\right|\right) = \omega\left(\underbrace{\sqrt{n}\left|x - \frac{k}{n}\right|}_{\lambda} \underbrace{\frac{1}{\sqrt{n}}}_{\delta}\right) \leq \underbrace{\left(1 + \sqrt{n}\left|x - \frac{k}{n}\right|\right)}_{(*)} \omega\left(\frac{1}{\sqrt{n}}\right)$$

so that

$$|f(x) - B_n(f; x)| \leq \sum_{k=0}^n (*) \binom{n}{k} x^k (1 - x)^{n-k} \leq \omega\left(\frac{1}{\sqrt{n}}\right) \left[1 + \sqrt{n} \sum_{k=0}^n \left|x - \frac{k}{n}\right| \binom{n}{k} x^k (1 - x)^{n-k}\right]$$

and by Cauchy-Schwarz's inequality $(\sum a_i b_i)^2 \leq (\sum a_i^2)(\sum b_i^2)$

$$\begin{aligned} \sum_{k=0}^n \left| x - \frac{k}{n} \right| \binom{n}{k} x^k (1-x)^{n-k} &= \sum_{k=0}^n \left(\left| x - \frac{k}{n} \right| \sqrt{\binom{n}{k} x^k (1-x)^{n-k}} \right) \sqrt{\binom{n}{k} x^k (1-x)^{n-k}} \\ &\leq \left[\sum_{k=0}^n \left(\left(x - \frac{k}{n} \right)^2 \binom{n}{k} x^k (1-x)^{n-k} \right) \right]^{1/2} \sqrt{\sum_{k=0}^n \binom{n}{k} x^k (1-x)^{n-k}} \\ &\leq \sqrt{\sum_{k=0}^n \left(x - \frac{k}{n} \right)^2 \binom{n}{k} x^k (1-x)^{n-k}} \end{aligned}$$

and since

$$\sum_{k=0}^n \left(x - \frac{k}{n} \right)^2 \binom{n}{k} x^k (1-x)^{n-k} \stackrel{(*)}{=} \frac{x(1-x)}{n} \leq \frac{1}{4n}$$

we finally get

$$|f(x) - B_n(f; x)| \leq \omega(1/\sqrt{n}) \left(1 + \sqrt{n} \frac{1}{2\sqrt{n}} \right),$$

that concludes the proof. \square .

Remarks

(i) In the previous theorem we have used in (*) the fact that

$$\sum_{k=0}^n \left(x - \frac{k}{n} \right)^2 \binom{n}{k} x^k (1-x)^{n-k} = \frac{x(1-x)}{n}$$

which can be proved by using the facts that

$$\begin{aligned} \sum_{k=0}^n \binom{n}{k} x^k (1-x)^{n-k} &= 1 \\ \sum_{k=0}^n k \binom{n}{k} x^k (1-x)^{n-k} &= nx \sum_{k=0}^{n-1} \binom{n-1}{k} x^k (1-x)^{n-1-k} = nx \\ \sum_{k=0}^n k(k-1) \binom{n}{k} x^k (1-x)^{n-k} &= n(n-1)x^2 \sum_{k=0}^{n-2} \binom{n-2}{k} x^k (1-x)^{n-2-k} = n(n-1)x^2. \end{aligned}$$

(ii) if $f \in \mathcal{C}[0, 1]$, for property (b), $\omega(1/\sqrt{n}) \rightarrow 0$, getting another proof of the Weierstrass's theorem.

- (iii) $f \in Lip_K \alpha$ (or simply $Lip \alpha$ if K is unimportant) i.e. $|f(x_1) - f(x_2)| \leq K|x_1 - x_2|^\alpha$, $\alpha > 0$ if and only if

$$\omega(\delta) \leq K\delta^\alpha.$$

This is also known as **Hölder condition of order α** .

It follows that

$$\|f - B_n(f)\| \leq \frac{3}{2}Kn^{-\alpha/2}, \quad f \in Lip_K \alpha, \quad K = [0, 1].$$

Notice that, for $\alpha = 0$ the inequality implies that the function are bounded; when $\alpha = 1$ we get the classical Lipschitz condition and for $\alpha > 1$, the functions in $Lip \alpha$ are constants (i.e. with $f' = 0$). In fact, letting $\alpha = 1 + \epsilon$, for a fixed $x_1 \in I$ it follows that

$$\lim_{x_2 \rightarrow x_1} \frac{|f(x_2) - f(x_1)|}{|x_2 - x_1|} \leq K|x_2 - x_1|^{\alpha-1} = 0$$

and the ration is ≥ 0 . Hence the limit, which is f' , exists and has value 0, showing that f is constant.

Therefore the property (ii) is interesting for $0 < \alpha \leq 1$.

- (iv) Suppose $f(x) = |x - \frac{1}{2}|$ (which is $Lip_1 1$). Then, by (ii) $\|f - B_n(f)\| \leq \frac{3}{2}n^{-1/2}$. Take the point $x = 1/2$. Since $B_n(f; x) - |x - 1/2| = (\frac{1}{2})^n \sum_{k=0}^n |\frac{k}{n} - \frac{1}{2}| \binom{n}{k}$. If n is even, we get

$$\sum_{k=0}^n \left| \frac{k}{n} - \frac{1}{2} \right| \binom{n}{k} = 2 \sum_{k=0}^{n/2} \left(\frac{1}{2} - \frac{k}{n} \right) \binom{n}{k} = \frac{1}{2} \binom{n}{n/2}$$

hence

$$|B_n(f; \frac{1}{2}) - 0| = \frac{\binom{n}{n/2}}{2^{n+1}} > \frac{1}{2} \frac{1}{\sqrt{n}}$$

the last inequality coming from the so-called **Stirling's formula**

$$\sqrt{2\pi k} k^k e^{-k} < k! < \sqrt{2\pi k} k^k e^{-k} \left(1 + \frac{1}{4k} \right).$$

◇◇

The last step for the correct asymptotic behaviour of $E_n^*(f)$ is given by the **Jackson's theorem**.

Theorem 6. (Jackson's theorem)

If $f \in \mathcal{C}[-1, 1]$, then

$$\boxed{E_n^*(f; [-1, 1]) \leq 6\omega\left(\frac{1}{n}\right)}. \quad (1.15)$$

Remark. We do not prove the Theorem. The proof is essentially based on Jackson's theorem for functions $f \in \mathcal{C}_0([- \pi, \pi])$, i.e. by using trigonometric polynomials. Hence, in $[-1, 1]$ by using the substitution $x = \cos(t)$ we return to the trigonometric case.

The following Corollaries are easy consequences of the Jackson's theorem and the previous properties.

Corollary 1.

$$\boxed{E_n^*(f; [a, b]) \leq 6\omega\left(\frac{b-a}{2n}\right)}. \quad (1.16)$$

Corollary 2. If $f \in Lip_K \alpha[-1, 1]$

$$\boxed{E_n^*(f; [-1, 1]) \leq 6K n^{-\alpha}}. \quad (1.17)$$

Corollary 3. If $|f'(x)| \leq M$ for $x \in [-1, 1]$ then

$$\boxed{E_n^*(f; [-1, 1]) \leq 6M n^{-1}}. \quad (1.18)$$

Corollary 4. If $f \in \mathcal{C}^1[-1, 1]$ then

$$\boxed{E_n^*(f; [-1, 1]) \leq \frac{6}{n} E_{n-1}^*(f'; [-1, 1])}.$$

Proof. Suppose $\|f' - p_{n-1}^*\| = E_{n-1}^*(f', [-1, 1])$. Let $p_n(x) = \int_0^x p_{n-1}^*(x) dx$. Then

$$E_n^*(f; [-1, 1]) = E_{n-1}^*(f - p_n; [-1, 1]) \leq 6E_{n-1}^*(f'; [-1, 1]) n^{-1}$$

where we have used Corollary 3 and the fact that $p_n' = p_{n-1}^*$. \square

Theorem 7. (no proof)

Let $f \in \mathcal{C}^k[-1, 1]$, then for $n > k$

$$\boxed{E_n^*(f; [-1, 1]) \leq \frac{c_k}{n^k} \omega_k\left(\frac{1}{n-k}\right)}. \quad (1.19)$$

where ω_k is the MoC of $f^{(k)}$ and $c_k = \frac{6^{k+1} e^k}{k+1}$.

This last theorem simply gives information on how well a function (with known properties) can be approximated by polynomials. It does not provide any tool for finding p_n^* .

1.1.3 Characterizations of the polynomial of best approximation

We make some simple notations. Let $e(x) = f(x) - p_n^*(x)$ (the best polynomial approximation error at the point x) and $\|e\| = E_n^*(f; [a, b])$.

Theorem 8. *There exist (at least) two distinct points, $x_1, x_2 \in [a, b]$ s. t.*

$$|e(x_1)| = |e(x_2)| = E_n^*(f; [a, b]),$$

and $e(x_1) = -e(x_2)$.

Proof. We observe that the curve $y = e(x)$ is constrained to lie between the constants (lines) $y = \pm E_n^*(f)$, $x \in [a, b]$, touching at least one of them. We show that $e(x)$ touches both.

If it does not, then there should exist a *better approximation* of f than p_n^* . Assume that $e(x) > -E_n^*(f)$ in $[a, b]$.

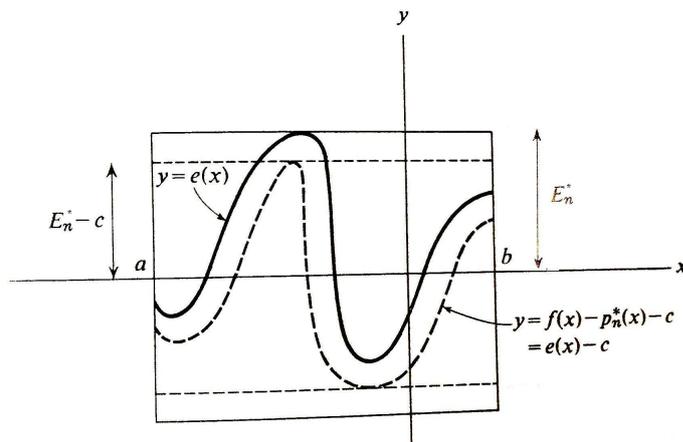


Figure 1.1: Error $e(x)$ and $e(x) - c$

Then,

$$\min_{x \in [a, b]} e(x) = m > -E_n^*(f) \text{ and } c = \frac{E_n^*(f) + m}{2} > 0.$$

Since $q_n = p_n^* + c \in \mathbb{P}_n(\mathbb{R})$, $f(x) - q_n(x) = e(x) - c$ and $-(E_n^*(f) - c) = m - c \leq e(x) - c \leq E_n^*(f) - c$, we have

$$\|f - q_n\| = E_n^*(f) - c,$$

contradicting the definition of $E_n^*(f)$. Thus there must exist a point, say $x_1 \in [a, b]$, such that $e(x_1) = -E_n^*(f)$. Similarly we can say that there exists $x_2 \in [a, b]$ s. t. $e(x_2) = E_n^*(f)$. This concludes the proof. \square

As a consequence we have

Corollary 5. *The best approximating **constant** to $f(x)$ is*

$$p_0^*(x) = \frac{1}{2} \left\{ \max_{x \in [a,b]} f(x) + \min_{x \in [a,b]} f(x) \right\}$$

with error

$$E_0^*(f) = \frac{1}{2} \left\{ \max_{x \in [a,b]} f(x) - \min_{x \in [a,b]} f(x) \right\}.$$

Proof. Again by absurdum. Suppose d is any other constant, then $e(x) = f(x) - d$ can not satisfy the Theorem 8. In fact

$$\begin{aligned} e(x_1) &= f(x_1) - d \\ e(x_2) &= f(x_2) - d \end{aligned}$$

showing that $e(x_1) \neq -e(x_2)$, in contradiction with the property stated in Theorem 8. \square

We need the following **definition**.

Definition 2. *A set of $k + 1$ distinct points $a \leq x_0 < x_1 < \dots < x_k \leq b$ is called an **alternating set** for the error function $e = f - p_n$ if*

$$\begin{aligned} |e(x_j)| &= |f(x_j) - p_n(x_j)| = \|f - p_n\|_\infty, \quad j = 0, \dots, k \\ \text{and} \\ f(x_j) - p_n(x_j) &= -[f(x_{j+1}) - p_n(x_{j+1})], \quad j = 0, \dots, k - 1. \end{aligned}$$

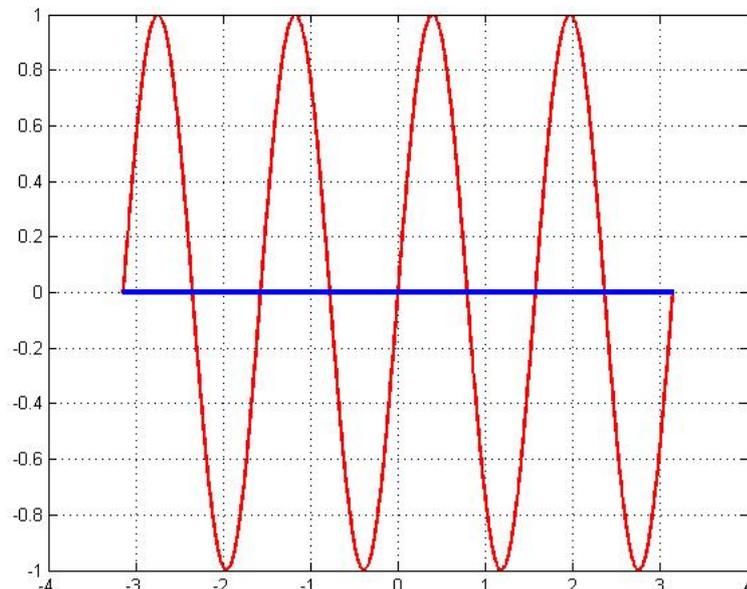
We can finally characterize the best polynomial approximation in the following way

Theorem 9. *Suppose $f \in \mathcal{C}[a, b]$. The polynomial $p_n^* \in \mathbb{P}_n(\mathbb{R})$ is a **best uniform approximation** on $[a, b]$ to f , if and only if there exists an **alternating set** for $e = f - p_n^*$ consisting of $n + 2$ distinct points.*

Remark. The Theorem does not say that the alternating set is unique. Moreover it does not say that the error $f - p_n^*$ could alternate on more than $n + 2$ points.

Instead of providing the proof of Theorem 9, that can be found in [39, pp. 26-27], we give an explicative example.

Example 4. Consider the function $\sin(4x) \in [-\pi, \pi]$ whose plot is displayed in Figure 1.2, together with the best constant approximation (the constant $y = 0$). The error $e = f - p_0^*$ has 8 different alternating sets consisting of 2 points. Indeed, the Theorem 9 says that $p_0^* = p_1^* = \dots = p_5^* = p_6^* = 0$ are best approximations of the indicated degree on $[-\pi, \pi]$ since $\max |\sin(4x)| = 1$ and "alternates" 8 times in the interval $[-\pi, \pi]$. But $p_7 = 0$ is not a best approximation.

Figure 1.2: The function $\sin(4x) \in [-\pi, \pi]$

The Theorem 9 enables to settle the property of uniqueness of the Best Polynomial Approximation (BPA).

Theorem 10. *If p_n^* is the BPA, then it is unique. That is, if p is another BPA and $p \neq p_n^*$ then*

$$\|f - p\| > \|f - p_n^*\|.$$

Proof. Assume $\|f - p\| = \|f - p_n^*\| = E_n(f)$. Then $q = \frac{p+p_n^*}{2}$ is also a BPA, for the convexity property of linear spaces. Now, let x_0, \dots, x_{n+1} be an alternating set for $f - q$, so that for some integer l

$$\frac{f(x_j) - p(x_j)}{2} + \frac{f(x_j) - p_n^*(x_j)}{2} = (-1)^{l+j} E_n^*(f), \quad j = 0, \dots, n+1. \quad (1.20)$$

Since

$$\frac{|f(x_j) - p(x_j)|}{2} \leq \frac{E_n^*(f)}{2}, \quad \text{and} \quad \frac{|f(x_j) - p_n^*(x_j)|}{2} \leq \frac{E_n^*(f)}{2},$$

then, the identity (1.20) holds only if

$$f(x_j) - p(x_j) = f(x_j) - p_n^*(x_j) = (-1)^{l+j} E_n^*(f), \quad j = 0, \dots, n+1,$$

that implies $p(x_j) = p_n^*(x_j)$, $j = 0, \dots, n+1$, that is $p \equiv p_n^*$ \square

There are very few functions f for which a BPA can be found explicitly.

Example 5. This example is indeed a Theorem.

Theorem 11. *The function $f(x) = x^{n+1}$ has BPA p_n^* on $[-1, 1]$ s.t.*

$$x^{n+1} - p_n^* = \frac{1}{2^n} T_{n+1}(x), \quad E_n^*(x^{n+1}; [-1, 1]) = \frac{1}{2^n}$$

where $T_k(x) = \cos(k \arccos(x))$, $k = 0, \dots, n$ (Chebyshev polynomial of first kind)

In Figure 1.3 we provide the plot of the first four Chebyshev polynomials T_0, \dots, T_3 .

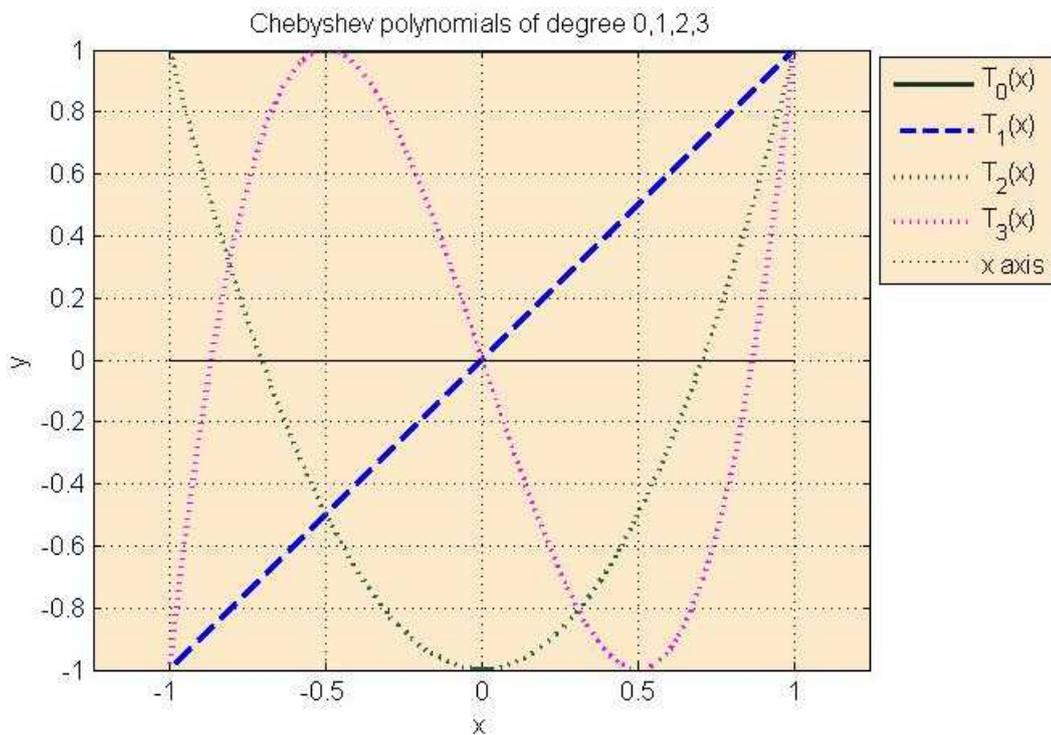


Figure 1.3: Plot of the first Chebyshev polynomials

Hence the above problem is equivalent to find the *monic* polynomial of degree $n + 1$ which deviates least from zero in absolute values on $[-1, 1]$.

The answer to the Theorem 11 is given by this other Theorem.

Theorem 12. *The BPA to x^{n+1} is*

$$\tilde{T}_n(x) = \frac{1}{2^{n-1}} T_n(x),$$

i.e. the normalized Chebyshev polynomial of degree n .

The previous example shows one of the very important properties of the orthogonal Chebyshev polynomials. Here we simply summarize some others for completeness.

1. The Chebyshev polynomials satisfy the recurrence

$$\begin{aligned} T_0(x) &= 1, & T_1(x) &= x, \\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x), & n &\geq 1. \end{aligned} \tag{1.21}$$

2. The zeros of Chebyshev polynomials are real, distinct and inside $[-1, 1]$.
3. Chebyshev polynomials form a (stable) basis for $\mathbb{P}_n(\mathbb{R})$ (we can express the monomial x^k by a combination of Chebyshev polynomials).

1.1.4 How to find the BPA to a function?

This is not always possible! The general procedure works as follows. Consider the function f to be approximated and a set X of $n+2$ points x_1, \dots, x_{n+2} (for example equally spaced or Chebyshev points). Then

- (a) Solve the linear system

$$b_0 + b_1x_i + \dots + b_nx_i^n + (-1)^iE = f(x_i), \quad i = 1, 2, \dots, n+2,$$

for the unknowns $b_i, i = 1, \dots, n+1$ and E .

Use then the values b_i to form the polynomial p_n

- (b) Find the set Y consisting of local maximum error $|p_n(x) - f(x)|$.
- (c) If the errors at every $y \in Y$ are equal in magnitude and alternate in sign, then $p_n = p_n^*$. Otherwise, replace X with Y and repeat the steps above.

The success of this procedure depends on two factors.

1. The possibility of finding a BA on a finite point set
2. The BA on a finite point set should provide the BA on the interval as the number of points increases.

All these things are solved by the so-called **Exchange Method** (cf. [39, §1.4]), implemented in the *Remez (or Remes) algorithm* [38]. An interesting implementation can be found in the package `chebfun`, www.chebfun.org [19] and presented in the paper [35]. Chebfun is an open-source package for computing with functions to 15-digit accuracy. Most Chebfun commands are overloads of familiar MATLAB commands, for example `sum(f)` computes an integral, `roots(f)` finds zeros, and `u = L f` solves a differential equation.

Here the code of the Remez algorithm (as presented in [35]) in the `chebfun` system (slightly but only slightly simplified). The input arguments are a chebfun f and the degree n of the polynomial to be computed and the output arguments are a chebfun p of the best polynomial approximation to f and the error err .

```

%-----
function [p,err] = remez(f,n); % compute deg n BA to chebfun f
iter = 1; delta = 1; deltamin = delta;
[a,b] = domain(f);
xk = chebpts(n+2); xo = xk; % initial reference
sigma = (-1).^[0:n+1]'; % alternating signs
normf = norm(f);
while (delta/normf > 1e-14) & iter <= 20
fk = feval(f,xk); % function values
w = bary_weights(xk); % compute barycentric weights
h = (w'*fk)/(w'*sigma); % levelled reference error
if h==0, h = 1e-19; end % perturb error if necessary
pk = fk - h*sigma; % polynomial vals in the
p=chebfun(@(x)bary(x,pk,xk,w),n+1); % reference chebfun of trial
e = f - p; % polynomial chebfun of the
[xk,err] = exchange(xk,e,h,2); % error replace reference
if err/normf > 1e5 % if overshoot, recompute with
[xk,err] = exchange(xo,e,h,1);% one-point exchange
end
xo = xk;
delta = err - abs(h); % stopping value
if delta < deltamin, % store poly with minimal norm
deltamin = delta;
pmin = p; errmin = err;
end
iter = iter + 1;
end
p = pmin; err = errmin;
%-----

```

We also suggest to look at the documentation at the Chebfun web site

<http://www.chebfun.org/examples/approx/BestApprox.html>

Lecture 2

Lebesgue constant

The interpolation error, as stated in Theorem 1, is composed of two parts: the best polynomial approximation error, already discussed in Chapter 1 and the Lebesgue constant. This second error-part is the content of the present chapter.

2.1 Lebesgue constant

Firstly some notations. Let $X_n = \{x_k = x_{k,n}, k = 0, \dots, n, n = 1, 2, \dots\}$ a triangular array of nodes s.t. $a \leq x_0 < x_1 < \dots < x_n \leq b$. We consider the Banach space of continuous functions in $[a, b]$, $\mathcal{C}[a, b]$, equipped with the sup-norm

$$\|f\|_\infty = \max_{x \in [a, b]} |f(x)|.$$

Let $p_n(f; X_n; x) = p_n(f; x) = \sum_{k=0}^n l_k(x) f(x_k)$ the interpolating polynomial, in Lagrange form, of f in $[a, b]$ at the point x . Moreover, we consider

$$\lambda_n(x) = \sum_{k=0}^n |l_k(x)|, \tag{2.1}$$

which is the **Lebesgue function**, whose maximum value in $[a, b]$ gives the **Lebesgue constant**, that we'll indicate by

$$\Lambda_n := \max_{x \in [a, b]} \lambda_n(x). \tag{2.2}$$

Remarks.

- (i) Consider the operator $\mathcal{L}_n : \mathcal{C}[a, b] \rightarrow \mathbb{P}_n(\mathbb{R})$. This operator is indeed a **projection** and $\mathcal{L}_n(q) = q, \forall q \in \mathbb{P}_n(\mathbb{R})$. Then

$$\Lambda_n := \sup_{g \in \mathcal{C}[a, b], g \neq 0} \frac{\|\mathcal{L}_n(g)\|_\infty}{\|g\|_\infty}$$

that is, the Lebesgue constant is the sup-norm of \mathcal{L}_n .

- (ii) A second interpretation of Λ_n is as **stability constant**. In fact, let $f_i = f(x_i)$ and the corresponding perturbed points $\tilde{f}_i = f_i + \delta_i$. Letting p_n and \tilde{p}_n the corresponding interpolating polynomials (at the same interpolation points!). Since

$$|p_n(x) - \tilde{p}_n(x)| = \left| \sum_{k=0}^n (f_i - \tilde{f}_i) l_i(x) \right|$$

we get

$$\|p_n - \tilde{p}_n\|_\infty \leq \Lambda_n \|f - \tilde{f}\|_\infty$$

from which follows a lower bound for the Lebesgue constant

$$\Lambda_n \geq \frac{\|p_n - \tilde{p}_n\|_\infty}{\|f - \tilde{f}\|_\infty}.$$

◇◇

The importance of the Lebesgue constant is given by the following Theorem

Theorem 13. *Let $f \in \mathcal{C}[a, b]$ and $p_n \in \mathbb{P}_n(\mathbb{R})$ the interpolating polynomial of degree n on x_0, \dots, x_n . Then*

$$\|f - p_n\|_\infty \leq (1 + \Lambda_n) \underbrace{\|f - p_n^*\|_\infty}_{E_n^*(f)}. \quad (2.3)$$

Proof. For all $q_n \in \mathbb{P}_n$

$$f - p_n = (f - q_n) - \underbrace{(p_n - q_n)}_{\in \mathbb{P}_n} = (f - q_n) - \mathcal{L}_n(f - q_n)$$

Hence

$$\|f - p_n\|_\infty \leq \|f - q_n\|_\infty + \|\mathcal{L}_n(f - q_n)\|_\infty \leq \|f - q_n\|_\infty + \|\mathcal{L}_n\|_\infty \|f - q_n\|_\infty \leq (1 + \Lambda_n) \|f - q_n\|_\infty.$$

To conclude, just take p_n^* instead of q_n . \square

Formula (2.3) shows again that the interpolation error depends on two elements: (a) the Lebesgue constant Λ_n , that depends only on the point set X_n and (b) the BPA error, $E_n^*(f)$, which depends only on f .

In the next subsection we want to understand more about the growth of the Lebesgue constant for different choices of the points set.

2.1.1 Properties of the Lebesgue function $\lambda_n(x)$

Here we enumerate some of the most important properties of the Lebesgue function. To better understand them, we have plotted in Figure 2.1 the Lebesgue functions for two set of points, equispaced and Chebyshev-Lobatto. The computation have been done with the following two Matlab functions that we enclose here for completeness.

Given the set \mathbf{x} of nodes in the given interval, we then provide a set of target points, say \mathbf{t} , where to evaluate λ_n . We construct the matrix $L = (l_0(t), \dots, l_n(t))^T$ (this means that every row is the i -th Lagrange polynomial evaluated at all target points). Then $\lambda_n = \text{sum}(\text{abs}(L), 1)$, while $\Lambda_n = \text{norm}(L, 1)$.

```
function [leb,funLeb]=CostLebesgue(x)
%-----
% Input:
%   x = ordered vector of interpolation nodes
% Output:
%   leb= the value of the Lebesgue constant
%   funLeb=Lebesgue function, vector of dimension M
%           as the target points
%-----
a=x(1); b=x(end); M=1000;
t=linspace(a,b,M)'; % "target" points (column vector)
N=length(x);
for s=1:N
    L(s,:)=lagrai_target(x,t,s);
end
leb=norm(L,1);
funLeb=sum(abs(L),1);
end

function l = lagrai_target(z,x,i)
%-----
% Compute the i-th elem Lagrange polynomial
% on a vector of target points
%-----
% inputs
% z = interpolation nodes
% x = vector (column!)of targets on which evaluate l_i
% i = index
%
% output
% l = vector of the values of l_i at the targets
%-----
n = length(z); m = length(x);
l = prod(repmat(x,1,n-1)-repmat(z([1:i-1,i+1:n]),m,1),2)/...
    prod(z(i)-z([1:i-1,i+1:n]));
return
```

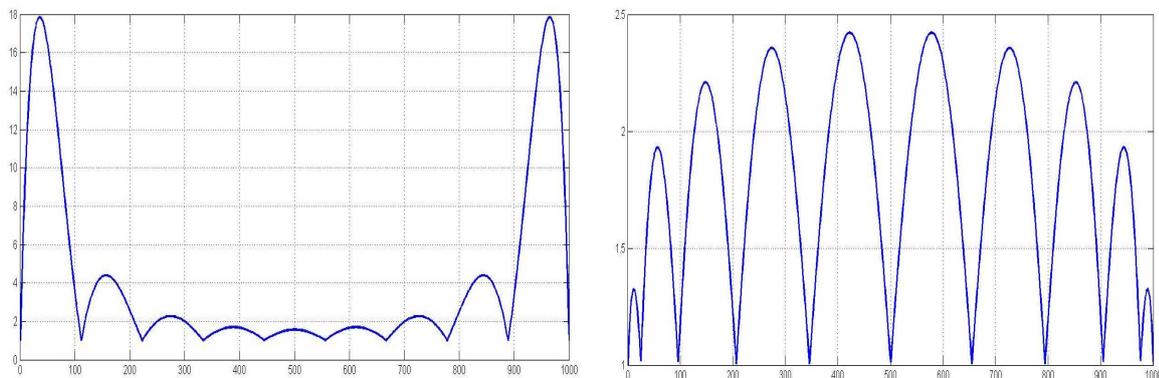


Figure 2.1: Plot of the Lebesgue functions for $n \leq 10$ for equispaced (left) and Chebyshev-Lobatto (right) points

From the above graphs now we understand these properties (presented and discussed in the paper [14]).

- (i) for $n \geq 2$, $\lambda_n(x)$ is a piecewise polynomial s.t. $\lambda_n \geq 1$ (with $\lambda_n(x_i) = 1$). This is a consequence of the definition of λ_n .
- (ii) λ_n has precisely one local maximum on (x_{k-1}, x_k) , $k = 1, \dots, n$ (denoted by μ_k)
- (iii) λ_n is monotone: decreasing and convex in (a, x_0) and increasing on (x_n, b)
- (iv) $\lambda_n(x) = \lambda_n(-x)$, $x \in [a, b]$ if and only if $x_{n-k} = -x_k$, $k = 0, \dots, n$.
- (v) Let $Z = \{z_k\}_{k=0}^n$ and $Y = \{y_k\}_{k=0}^n$ two set of points such that $y_k = \alpha z_k + \beta$, then $\lambda_n(Z; z) = \lambda_n(Y; \alpha z + \beta)$.

2.1.2 The Lebesgue constant for specific set of points

Following the paper [14], we illustrate a brief history of the bounds of the Lebesgue constant on different set of points of the interval $[a, b]$.

(A) We start with the set

$$E = \left\{ x_k = a + k \frac{(b-a)}{n}, k = 0, \dots, n \right\}$$

of **equidistant nodes**. In the below Figure 2.4 (left), we show the growth of the Lebesgue constant for $n \leq 10$ equispaced points on $[0, 1]$.

Runge (1901) observed that the set E is a **bad choice** for Lagrange interpolation.

Tietze (1917). T. Rivlin in 1994, noticed that the study of the maxima $\mu_k(E)$ of λ_n , was already presented in the work by Heinrich and Tietze 1917. In particular they noticed that $M_n(E) = \max_k \mu_k(E) := \Lambda_n(E)$. Moreover, Landau in 1913, already showed that the minimum of these maxima, i.e. $m_n(E) := \min_k \mu_k(E)$, behaves like

$$G_n \approx m_n(E) \sim \frac{1}{\pi} \log(n) \quad n \rightarrow \infty.$$

where G_n is known as **Landau constant**.

Turetskii (1940) found the following asymptotic behavior for the largest maximum

$$\Lambda_n(E) \sim \frac{2^{n+1}}{e n \log n}, \quad n \rightarrow \infty.$$

Schoenhage (1961) proved

$$\Lambda_n \sim \frac{2^{n+1}}{e n (\log n + \gamma)}, \quad n \rightarrow \infty \quad (2.4)$$

with $\gamma = 0.577\dots$ corresponding to the **Euler constant**. Moreover he showed

$$m_n(x) < \frac{2}{\pi} (\log(n+2) + \log(2) + \gamma).$$

Trefethen & Weideman (1991) provided upper and lower bounds for $\Lambda_n(E)$

$$\frac{2^{n-2}}{n^2} < \Lambda_n(E) < \frac{2^{n+3}}{n}, \quad n \geq 1. \quad (2.5)$$

Exercise 2. Using the Matlab code above, show the validity of the previous results by Schoenage (2.4) and the inequalities (2.5).

◇

(B) The second set of points we consider is the set

$$T = \left\{ x_k = -\cos\left(\frac{(2k-1)\pi}{2n}\right), \quad k = 1, \dots, n \right\} \quad (2.6)$$

of **Chebyshev nodes** (which belong to $[-1, 1]$) which are the **roots** of Chebyshev polynomials of first kind (see their definition in (1.21)).

In fact, letting $T_n(x) = \cos(n \arccos(x))$, $|x| \leq 1$ and the fact that $\cos\left((2k+1)\frac{\pi}{2}\right) = 0$ then we get $x_k = \cos\left((2k-1)\frac{\pi}{2n}\right)$.

In the set T we have chosen the $-\cos$ in order to get the nodes ordered in ascending way.

The Chebyshev points (of first kind) can also be seen as the projection of the equispaced points on the half circle, along the diameter, as shown in Figure 2.2.

Remark. In many books, the Chebyshev points (of first kind) are indicated as the points $x_k = \cos\left(\frac{k\pi}{n}\right)$, $k = 0, \dots, n$. These are not the Chebyshev points, even if their behavior, in terms of stability and convergence of polynomial interpolation is the same of the Chebyshev points. These points are called *Chebyshev-Lobatto* points which are related to the Gauss quadrature and correspond to the *Chebyshev extrema* (see below).

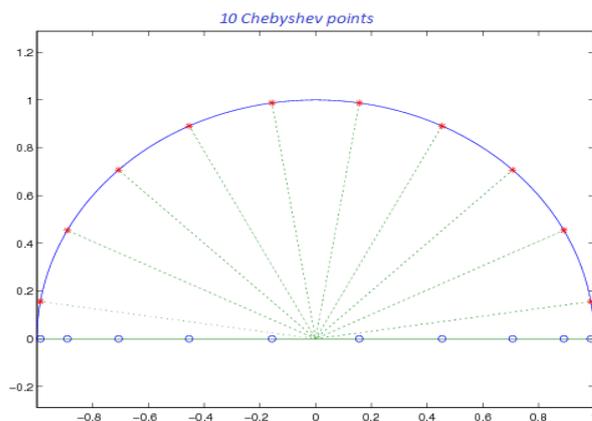


Figure 2.2: The set T of Chebyshev points for $n = 9$

In Figure 2.4 (right), we show the growth of the Lebesgue constant for $n \leq 10$ Chebyshev points on $[-1, 1]$.

Bernstein (1918) proved that

$$\Lambda_n(T) \sim \frac{2}{\pi} \log(n+1), \quad n \rightarrow \infty.$$

Natanson (1965) in his monograph, show that for all finite n

$$\Lambda_n(T) < 8 + \frac{4}{\pi} \log(n+1).$$

Berman (1963) improved the results by Natanson showing that

$$\Lambda_n(T) < 4\sqrt{2} + \frac{2}{\pi} \log(n+1).$$

Ivanov & Zadiraka (1966) made a further improvement

$$\Lambda_n(T) < 2 + \frac{2}{\pi} \log(n+1).$$

Luttmann & Rivlin (1965) provided computational results (after the advent of computers)

$$\Lambda_n(T) = \lambda_n(T; 1),$$

using the fact that $\lambda_n(T; 1) = \frac{1}{n+1} \sum_{k=0}^n \cotan\left(\frac{2k+1}{4(n+1)}\pi\right)$. They then proved

$$\lim_{n \rightarrow \infty} [\lambda_n(T; 1) - \frac{2}{\pi} \log(n+1)] = \frac{2}{\pi} (\gamma + \log \frac{8}{\pi}) := a_0 = 0.9625\dots$$

Ehlich & Zeller (1966), gave the bounds

$$\boxed{a_0 + \frac{2}{\pi} \log(n+1) < \Lambda_n(T) < 1 + \frac{2}{\pi} \log(n+1), \quad n = 0, 1, \dots} \quad (2.7)$$

Remark. It is worth noticing that the bounds provided in (2.7) hold for the $n+1$ Chebyshev points given as

$$T = \left\{ x_k = -\cos\left(\frac{(2k+1)\pi}{2n+2}\right), \quad k = 0, 1, \dots, n \right\}.$$

Since we have used the points (2.6) for Matlab reasons, the above bounds hold with n instead of $n+1$.

(C) The third set of interpolation points are the so-called *Extended Chebyshev points* defined as

$$\hat{T} = \left\{ x_k = \frac{x_k^{(C)}}{\cos(\frac{\pi}{2n})}, \quad k = 1, \dots, n \right\} \quad (2.8)$$

where the $x_k^{(C)}$ are the Chebyshev points set T . They are *stretched* Chebyshev points, so that they include the endpoints of $[-1, 1]$ (cf. Figure 2.3).

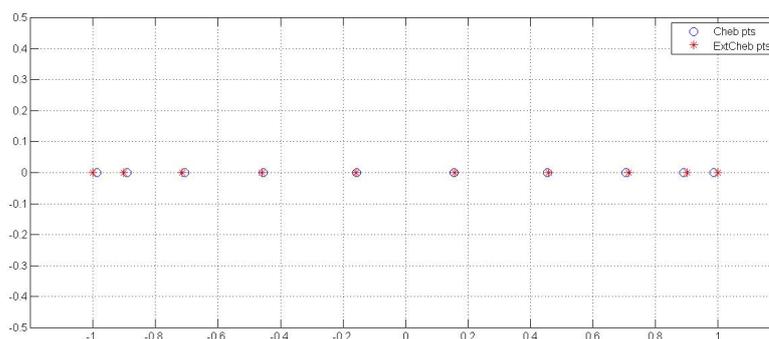


Figure 2.3: Chebyshev and Extended Chebyshev points

It was proved by Brutman [14]

$$\boxed{\Lambda_n(\hat{T}) < \Lambda_n(T)}. \quad (2.9)$$

In particular

$$\Lambda_n(\hat{T}) < 0.7219 + \frac{2}{\pi} \log(n+1)$$

establishing

$$\delta_n(\hat{T}) = M_n(\hat{T}) - m_n(\hat{T}) < 0.201... \forall n = 1, 2, \dots$$

improved as $\delta_n(\hat{T}) \leq 0.0196$, $n \geq 70$ and asymptotically

$$\lim_{n \rightarrow \infty} \delta_n(\hat{T}) = \frac{2}{\pi} \log 2 - \frac{3}{4\pi} = 0.01686...$$

(D) Another set useful for univariate polynomial interpolation on the interval $[-1, 1]$ is the set of the *Chebyshev extrema* which correspond to the *Chebyshev points* in many books and software. For example, `chebfun` (cf. [19]), define the Chebyshev points in this manner.

$$U = \{x_k = -\cos(k\pi/n), k = 0, \dots, n\}, \quad (2.10)$$

which provides the following bounds for the Lebesgue constant

$$\Lambda_n(U) = \begin{cases} \Lambda_{n-1}(T), & \text{for } n \text{ odd} \\ \Lambda_{n-1}(T) - \alpha_n, & 0 \leq \alpha_n < \frac{1}{n^2}, \text{ for } n \text{ even} \end{cases} \quad (2.11)$$

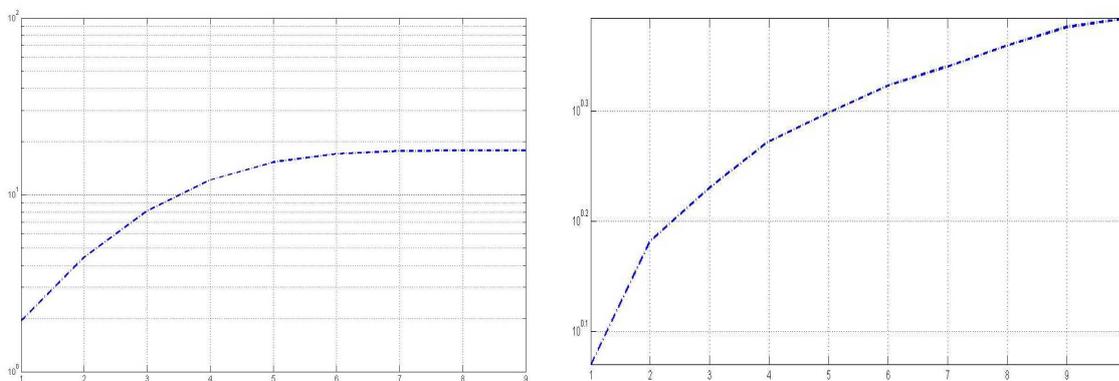


Figure 2.4: Plot of the Lebesgue constants for $n \leq 9$ for equispaced on $[-1, 1]$ (left) and $n \leq 10$ for the Chebyshev extrema points U (right)

(E) Other important sets are

(a) The set J of the roots of the Jacobi polynomials of indexes $\alpha, \beta > -1$ and $\gamma = \max\{\alpha, \beta\}$. Asymptotically holds

$$\Lambda_n(J) = \begin{cases} \mathcal{O}(n^{\gamma+1/2}), & \gamma > -1/2 \\ \mathcal{O}(\log n), & \gamma \leq -1/2 \end{cases} \quad (2.12)$$

For instance, the Chebyshev polynomials are Jacobi polynomials with $\alpha = \beta = -1/2$. From (2.12) we see once again that the Lebesgue constant of the Chebyshev points grows asymptotically as $\log n$.

- (b) **Fekete points:** F . On the interval $[-1, 1]$ they are the roots of $(1-x^2)P'_{n-1}(x)$ where P_n is the Legendre polynomial of degree n . Actually *they correspond to the Chebyshev-Lobatto points (or Chebyshev extrema) of the interval* (cf. [12, 10]).

Among all the properties of these points, we highlight the most important. Féjer in 1932 [14] proved that the Fekete points are the solution of the following optimization problem.

$$\text{Find the nodes } X \text{ that minimize } \Phi_n(X) = \max_{-1 \leq x \leq 1} \sum_{k=0}^n (l_k(X; x))^2 .$$

He proved that this set is the set F such that $\Phi_n(F) = 1$ so that $\Lambda_n(F) \leq \sqrt{n+1}$ (using Schwarz's inequality).

Another way to see the Fekete points, is as the set that **maximize** the modulus of the Vandermonde determinant, $|VDM(X)|$, among all set X of $n+1$ distinct points of $[-1, 1]$.

Luttmann and Rivlin numerically proved that

- i. $\Lambda_n(F) < \Lambda_n(T)$, $3 \leq n \leq 40$.
- ii. the maxima of $\lambda_n(F; x)$ on subinterval between nodes, decrease in $[0, 1]$ and therefore $\lambda_n(F; x)$ attains its maximum on $[-1, 1]$ at $x = 0$ for n even and "near" $x = 0$ for n odd.

Remark. Both the previous properties are still open problems!

Exercise 3. Write a Matlab code that compute the Fekete points and the Chebyshev points, then check the previous two properties of Luttmann and Rivlin.

\leftrightarrow Sündermann in 1983, proved that $\Lambda_n(F) = \mathcal{O}(\log n)$ (that is, they behave asymptotically like the Chebyshev points on the interval). \leftarrow

$\diamond\diamond$

- (c) **Leja sequences**

Definition 3. On $[a, b]$ take the point x_1 . The point $x_s \in [a, b]$, $s = 2, 3, \dots, N$ is s.t.

$$\prod_{k=1}^{s-1} |x_s - x_k| = \max_{x \in [a, b]} \prod_{k=1}^{s-1} |x - x_k|. \quad (2.13)$$

This set is called a **Leja sequence** for $[a, b]$.

We will call L_N this set.

Remark. Consider the set $F_N = \{f_1, \dots, f_N\}$ of Fekete points. The set F_N solve globally the multidimensional optimization problem

$$\max_{X_N \in [a, b]} |VDM(X_N)|.$$

Since $VDM(X_N) = VDM(X_{N-1}) \prod_{i=1}^{N-1} (X_N - x_i)$, then to determine the k -th Leja point, once we have computed x_1, \dots, x_{k-1} , we have to solve the one dimensional problem $\max_{x \in [a, b]} \prod_{i=1}^{k-1} |x - x_i|$.

Both F_N and L_N minimize $\Lambda_N(X)$ reducing the size of the Lagrange polynomials.

As a final note, we observe that Baglama, Calvetti, Reichel introduced the so-called *Fast Leja Points* (cf. [2]). Fast Leja points are obtained by maximization over *adaptive* discretization of the interval $[a, b]$. This method allows to compute m Leja points with a *complexity* of roughly $\frac{1}{2}m^2$ flops.

2.1.3 Optimal interpolation points

Consider again the linear operator $\mathcal{L}_n : \mathcal{C}[a, b] \rightarrow \mathbb{P}_n(\mathbb{R})$. We have seen that

$$\|f - \mathcal{L}_n(f; x)\|_\infty \leq (1 + \Lambda_n(X)) \text{dist}(f, \mathbb{P}_n(\mathbb{R})). \quad (2.14)$$

Question. Is there a set of node $X_n^* = X^*$ s.t. $\min_X \Lambda_n(X) = \Lambda_n(X^*) = \Lambda_n^*$?

We will call X^* **optimal set of nodes** or **extremal points set**.

How to characterize the set X^* (and Λ_n^*) is one of the most intriguing problem of (univariate) interpolation.

In 1931, Bernstein conjectured (cf. [14]): "*it seems plausible that the greatest of the relative maxima of $\lambda_n(X; x)$ will be minimized if all these maxima will be equal.*" But he was not able to prove it.

In other words, if

$$\mu_1(X^*) = \mu_2(X^*) = \dots = \mu_n(X^*),$$

then X^* is an optimal set of nodes.

Erds in 1947 proved that *there exists a unique canonical X^* for which holds* (canonical means $a = x_0 \leq x_1 \leq \dots \leq x_n = b$)

$$m_n(X^*) = \min_i \mu_i(X) \leq \Lambda_n(X^*) \leq \max_i \mu_i(X) = M_n(X).$$

This conjecture was proved only in 1978 by Kilgore and by de Boor and Pinkus.

Remark. The set X^* and the optimal $\Lambda_n(X^*)$ are not known (except for $n = 2$ and for $n = 3$, see <http://www.math.u-szeged.hu/~vajda/Leb/>). Brutman in 1978 proved that the set \hat{T} (the set of extended Chebyshev points) provides a very good approximation to X^* , showing

$$\frac{1}{2} + \frac{2}{\pi} \log(n+1) < m_n(\hat{T}) < \Lambda_n(X^*) < M_n(\hat{T}) < \frac{3}{4} + \frac{2}{\pi} \log(n+1). \quad (2.15)$$

That is, the set \hat{T} is as useful as the optimal nodes.

Lecture 3

The multivariate case

In the multivariate case the situation is quite complicated and often not clear.

3.1 The Haar-Mairhuber-Curtis theorem

We start our discussion of interpolation on the multivariate setting by polynomials, we need to introduce the space $\mathbb{P}_n(\mathbb{R}^d)$, which is the vector space of d -variate polynomials of degree $\leq n$, which has dimension $N := \binom{n+d}{n}$.

The Haar-Mairhuber-Curtis theorem, gives its name to three mathematicians that independently discovered it: Haar in 1901, Mairhuber in 1956 and Curtis in 1959 (cf. [22, Th. 2.4.1] and [29]). We need firstly a definition

Definition 4. (Haar space)

Let $B \subset \mathcal{C}(\Omega)$ be a finite dimensional subspace with basis $\{b_1, \dots, b_N\}$. Then, B is a Haar space on $\Omega \subseteq \mathbb{R}^d$ if

$$\det(b_k(x_j)) \neq 0,$$

for any set of distinct points x_1, \dots, x_N of Ω .

Remarks.

1. The existence of Haar spaces guarantees the invertibility of the matrix $A := (b_k(x_j))$.
2. Univariate polynomials of degree $\leq n$ form a $(n+1)$ -dimensional Haar space for data given at x_1, \dots, x_{n+1} .

Theorem 14. (cf. [22, Th. 2.4.1] and [?, Th. 2.3]) *Suppose that $\Omega \subseteq \mathbb{R}^d$, $d \geq 2$ contains an interior point. There exists no Haar space on Ω for dimension $d \geq 2$.*

Proof. Suppose $d \geq 2$ and $B \subseteq \Omega$ is a Haar space with basis $\{b_1, \dots, b_N\}$, $N \geq 2$. We show that this leads to a contradiction. Let x_1, \dots, x_N be a set of distinct points in $\Omega \subseteq \mathbb{R}^d$ and A the matrix as defined above. Since B is a Haar space, then $\det(A) \neq 0$. Consider the closed path between x_1 and x_2 (as displayed in Figure 3.1). The path exists since the closeness of Ω . If we exchange x_1 with x_2 (without interfering other points), the corresponding rows in

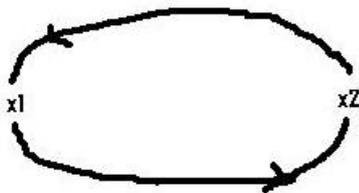


Figure 3.1: A closed path between two points

A will change and $\det(A)$ will change sign. But $\det : \mathcal{M} \rightarrow \mathbb{R}$ is a continuous function. Then, there exists a point such that $\det(A) = 0$ contradicting the fact that $\det(A) \neq 0$. \square

Exercise 4. Write a Matlab program that gives evidence of the HMC Theorem.

3.1.1 Unisolvent set of functions and point sets

We start by the definition of **unisolvency** (for functions) in the *univariate* case, since in higher dimension, due to Theorem 14, it is not easy to find unisolvent sets.

Definition 5. A system of N functions, ϕ_1, \dots, ϕ_N defined on a set $S \subseteq \mathbb{R}$ is called **unisolvant** on S , if

$$\det(\phi_i(x_j)) \neq 0$$

holds for every selection of N distinct points lying in S .

A unisolvent set of functions is also known as a *Chebyshev system*.

Another way to see the previous definition is that the matrix $A = (\phi_i(x_j))_{i,j=1}^n$ has linearly independent columns.

Notice, that the pointwise interpolation can always carry out uniquely with a unisolvent system. In other words, the set $\{\phi_1, \dots, \phi_N\}$ is unisolvent in S if and only if $\sum_{i=1}^N \alpha_i \phi_i(x) = g(x) \equiv 0$.

Example 6. These examples are taken from [22, Ch. 2.4].

1. $\{1, x^2\}$ is unisolvent in $[0, 1]$ not in $[-1, 1]$.

2. $\{1, x, x^2, \dots, x^n\}$ is unisolvent over any interval $[a, b]$.
3. Suppose $w(x)$ does not vanish on $[a, b]$. Then the set $\{w(x), w(x)x, w(x)x^2, \dots, w(x)x^n\}$ is unisolvent in $[a, b]$.
4. $\{1, \sin x, \cos x, \dots, \sin nx, \cos nx\}$ is unisolvent in $[-\pi, \pi]$.
5. Let a_i distinct on $[a, b]$, then

$$\left\{ \frac{1}{x + a_1}, \dots, \frac{1}{x + a_n} \right\}$$

is unisolvent in $[a, b]$. In fact, its determinant known as *Cauchy determinant*, is

$$\det \left(\frac{1}{x_i + a_j} \right) = \frac{\prod_{j < i} (x_i - x_j)(a_i - a_j)}{\prod_{i,j=1}^n (x_i + a_j)}$$

We introduce now the definition of *unisolvent point sets*, which is mostly used in higher dimensions.

Definition 6. (in \mathbb{R}^d)

The set $X \subset \mathbb{R}^d$ is unisolvent for a space W , if any $w \in W$ is completely determined by its values on X .

For example, X is unisolvent for \mathbb{P}_m^d (the polynomials of degree $\leq m$ in \mathbb{R}^d), if there exists a unique polynomial in \mathbb{P}_m^d of lowest possible degree that interpolates the data X .

To better understand we give an example

Example 7. Take $N = 7$ points in \mathbb{R}^2 . We can not interpolate these points uniquely. In fact, the polynomials of degree 2 has dimension $M_2 = \binom{2+2}{2} = 6$ (providing an over-determined system) and the polynomials of degree 3 has dimension $M_3 = \binom{2+3}{2} = 10$ (under-determined system).

A sufficient condition for unisolvence in \mathbb{R}^2 is the following theorem

Theorem 15. (Chung and Yao 1977 [20])

*Suppose $\{L_0, \dots, L_m\}$ are $m + 1$ distinct lines on \mathbb{R}^2 and $U = \{u_1, \dots, u_N\}$ a set of distinct points, with $N = \binom{2+m}{2}$ equal to the dimension of the polynomials of degree m in \mathbb{R}^2 , such that $u_1 \in L_0$, $u_2, u_3 \in L_1 \setminus L_0$ and $u_{N-m}, \dots, u_N \in L_m \setminus \{L_0, \dots, L_{m-1}\}$. Then, there exists a **unique** polynomial of degree $\leq m$ that interpolates on U . Moreover, if $X = \{x_1, \dots, x_M\}$ and $U \subset X$, then X is m -unisolvent on \mathbb{R}^2 .*

Example 8. Here we provide two simple examples.

- 3 collinear points on \mathbb{R}^2 are not 1-unisolvent.

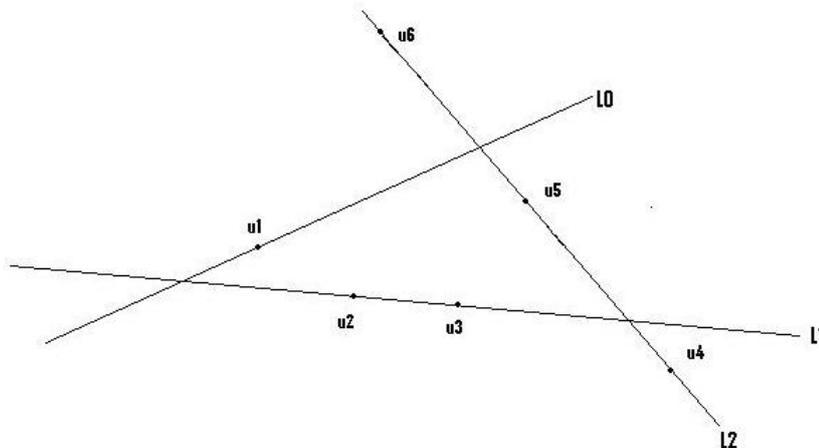


Figure 3.2: The Chung-Yao construction for unisolvency in \mathbb{R}^2 for $m = 2$ and $N = 6$.

- $m = 2$ and $N = 6$: see Figure 3.2.

From these results and examples, we have understood that also in the multidimensional setting, the choice of interpolation points is key issue. We want now to present some solutions to the problem given recently by the CAA-research group between the Universities of Padova and Verona (to which I belong to) <http://www.math.unipd.it/~marcov/CAA.html>.

In the next section we present a first solution in the bivariate setting and on the square, that allowed to introduce the most important set of points known as **Padua points**, that we will introduce in the next chapter.

Parts of the next section, are taken from [16].

3.2 Optimal and near-optimal interpolation points

Let $\Omega \subset \mathbb{R}^d$ be compact. We call *optimal* polynomial interpolation points a set $X_N^* \subset \Omega$ of cardinality N , such that the *Lebesgue constant*

$$\Lambda_n(X_N) = \max_{\mathbf{x} \in \Omega} \lambda_n(\mathbf{x}; X_N), \quad \lambda_n(\mathbf{x}; X_N) := \sum_{i=1}^N |\ell_i(\mathbf{x}; X_N)|, \quad (3.1)$$

defined for all sets $X_N = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \Omega$ which are unisolvent for polynomial interpolation of degree n , attains its minimum on $X_N = X_N^*$. Here, $\lambda_n(\mathbf{x}; X_N)$ is the *Lebesgue function* of X_N , the ℓ_i are the fundamental Lagrange polynomials of degree n , and N is the dimension of the corresponding polynomial space, i.e. $N = \binom{n+d}{d}$, or $N = (n+1)^d$ for the tensor-product

case (cf. e.g. [4, 12]). To be more precise, the fundamental Lagrange polynomials are defined as the ratio

$$\ell_i(\mathbf{x}; X_N) = \frac{VDM(X_N^{(i)})}{VDM(X_N)}, \quad (3.2)$$

where VDM denotes the Vandermonde determinants with respect to *any given basis* of the corresponding polynomial space, and where $X_N^{(i)}$ represents the set X_N in which \mathbf{x} replaces \mathbf{x}_i . It comes easy to see that tensor-product Lagrange polynomials are simply the product of univariate Lagrange polynomials.

As well-known optimal points are not known explicitly, therefore in applications people consider near-optimal points, i.e. roughly speaking, points whose Lebesgue constant increases asymptotically like the optimal one. Moreover, letting $E_n(X_N) = \|f - P_n\|_{\infty, \Omega}$, where P_n is the interpolating polynomial of degree $\leq n$ on X_N of a given continuous function f , and $E_n^* = \|f - P_n^*\|_{\infty, \Omega}$ the *best uniform approximation error*, then

$$E_n(X_N) \leq (1 + \Lambda_n(X_N))E_n^*,$$

which represents an estimate for the *interpolation error*. Thus, near-optimal nodes minimize also (asymptotically) the interpolation error.

In the *one-dimensional case*, as well-known, Chebyshev, Fekete, Leja as well as the zeros of Jacobi orthogonal polynomials are near-optimal points for polynomial interpolation, and their Lebesgue constants increase *logarithmically* in the dimension N of the corresponding polynomial space (cf. [14, 33]). All these points have asymptotically the *arc-cosine distribution*, that is they are asymptotically equidistributed w.r.t. the arc-cosine metric.

Here we will present different set of points that are **near-optimal** in the sense we have just clarify.

3.2.1 Tensor-product Chebyshev-Lobatto and Leja points

Here we consider two sets of tensor-product nodes in the square $[a, b] \times [a, b]$, i.e. the *tensor-product Chebyshev-Lobatto* and *tensor-product Leja points*, which have the same asymptotic distribution of the tensor-product Fekete points. Tensor-product Leja points are generated by using the so-called *Fast Leja Points*, introduced by Baglama, Calvetti, Reichel in [2]. Fast Leja points are obtained by maximization over *adaptive* discretization of the interval $[a, b]$. This method allows to compute m Leja points with a *complexity* of roughly $\frac{1}{2}m^2$ flops.

In Fig. 3.3 we compare the growth of Lebesgue constants for tensor-product Chebyshev-Lobatto points (shortly TPC) and tensor-product fast Leja points (shortly TPL) with the theoretical bound $(1 + 2/\pi \log(n))^2$ for near-optimal points (tensor-product Chebyshev points) (cf. [14]). In fact, it is immediate to see that the Lebesgue constant for tensor-product interpolation points is the square of the univariate constant. In practice, we have estimated the Lebesgue constants by maximizing the Lebesgue function (cf. (3.1)) on a grid of 101×101 equally spaced points on the reference square. In Tables 3.1-3.3 we then show

the errors of tensor-product interpolation with degrees $n = 24, 34, 44, 54$, corresponding to three test functions with different degree of regularity: the well-known *Franke* function

$$f_1(x_1, x_2) = \frac{3}{4} e^{-\frac{1}{4}((9x_1-2)^2+(9x_2-2)^2)} + \frac{3}{4} e^{-\frac{1}{49}(9x_1-2)^2-\frac{1}{10}(9x_2-2)^2} \\ + \frac{1}{2} e^{-\frac{1}{4}((9x_1-7)^2+(9x_2-3)^2)} - \frac{1}{5} e^{-((9x_1-4)^2+(9x_2-7)^2)}$$

considered as usual on $[0, 1]^2$, $f_2(x_1, x_2) = (x_1^2 + x_2^2)^{5/2}$ and $f_3(x_1, x_2) = (x_1^2 + x_2^2)^{1/2}$. Observe that f_2 and f_3 are not regular at the origin, in particular f_2 is C^4 with lipschitzian fourth partial derivatives and fifth partial derivatives discontinuous at the origin, while f_3 is lipschitzian with first partial derivatives discontinuous at the origin.

Even if the behavior of TPL Lebesgue constant is worse than that of TPC (see again Fig. 3.3), in the numerical tests the TPL approximation errors turn out to be closer to TPC errors than predicted by the ratio of Lebesgue constants (the errors have been computed on the same uniform control grid used to estimate the Lebesgue constant). Moreover, one can notice that the approximation performs better when the singularity is located at a corner of the square, since both TPC and TPL cluster by construction at the sides and especially at the corners.

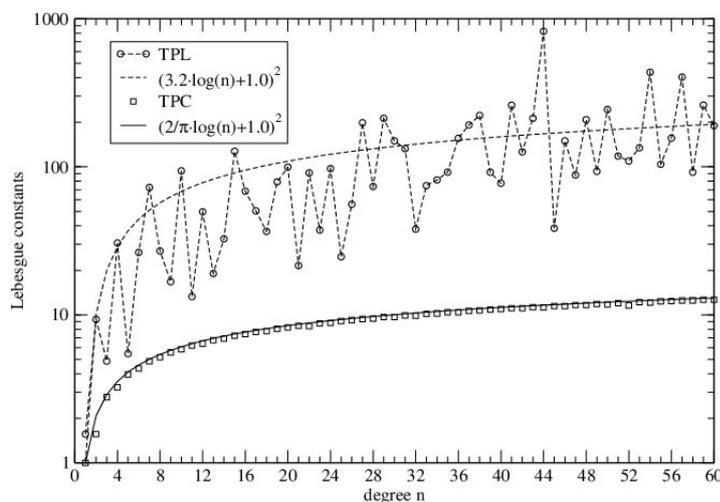


Figure 3.3: Lebesgue constants for tensor-product Chebyshev-Lobatto (TPC) and Leja (TPL) points up to degree 60, compared with the theoretical bound for TPC and with a least-square fitting for TPL.

N	25^2	35^2	45^2	55^2
TPC	$1.2 \cdot 10^{-3}$	$2.3 \cdot 10^{-6}$	$1.5 \cdot 10^{-9}$	$1.9 \cdot 10^{-13}$
TPL	$2.5 \cdot 10^{-3}$	$6.4 \cdot 10^{-6}$	$8.9 \cdot 10^{-9}$	$1.4 \cdot 10^{-12}$

Table 3.1: Tensor-product interpolation errors for the Franke function.

N	25 ²	35 ²	45 ²	55 ²
TPC on $[-1, 1]^2$	$6.0 \cdot 10^{-5}$	$8.2 \cdot 10^{-6}$	$1.8 \cdot 10^{-6}$	$5.4 \cdot 10^{-7}$
TPC on $[0, 2]^2$	$8.5 \cdot 10^{-9}$	$1.7 \cdot 10^{-10}$	$1.4 \cdot 10^{-11}$	$1.1 \cdot 10^{-11}$
TPL on $[-1, 1]^2$	$8.4 \cdot 10^{-5}$	$1.6 \cdot 10^{-5}$	$9.4 \cdot 10^{-6}$	$8.3 \cdot 10^{-7}$
TPL on $[0, 2]^2$	$2.3 \cdot 10^{-8}$	$6.3 \cdot 10^{-10}$	$1.4 \cdot 10^{-11}$	$1.8 \cdot 10^{-11}$

Table 3.2: Tensor-product interpolation errors for the function $f_2(x_1, x_2) = (x_1^2 + x_2^2)^{5/2}$.

N	25 ²	35 ²	45 ²	55 ²
TPC on $[-1, 1]^2$	$2.1 \cdot 10^{-1}$	$1.1 \cdot 10^{-1}$	$6.8 \cdot 10^{-2}$	$4.6 \cdot 10^{-2}$
TPC on $[0, 2]^2$	$2.8 \cdot 10^{-3}$	$5.8 \cdot 10^{-4}$	$1.1 \cdot 10^{-4}$	$8.9 \cdot 10^{-5}$
TPL on $[-1, 1]^2$	$5.7 \cdot 10^{-1}$	$5.6 \cdot 10^{-1}$	$6.2 \cdot 10^{-1}$	$1.1 \cdot 10^{-1}$
TPL on $[0, 2]^2$	$3.9 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$	$5.8 \cdot 10^{-5}$	$2.8 \cdot 10^{-5}$

Table 3.3: Tensor-product interpolation errors for the function $f_3(x_1, x_2) = (x_1^2 + x_2^2)^{1/2}$.

3.2.2 Dubiner metric and near-optimal interpolation points on $[-1, 1]^2$

Generalized arc-cosine metric

In [27], M. Dubiner proposed a metric which encapsulates the local properties of polynomial spaces on a given multivariate compact set, and in one dimension coincides with the *arc-cosine metric*:

$$\mu_{[-1,1]}(x, y) := |\cos^{-1}(x) - \cos^{-1}(y)|, \quad \forall x, y \in [-1, 1].$$

As an ingredient we need the *van der Corput-Schaake inequality* [40], valid for a trigonometric polynomial $T(\theta)$ of degree $\leq m$ with $|T(\theta)| \leq 1$

$$|T'(\theta)| \leq m\sqrt{1 - T^2(\theta)} \quad (3.3)$$

from which follows

$$\left| \frac{T'(\theta)}{\sqrt{1 - T^2(\theta)}} \right| \leq m \quad (3.4)$$

$$\left| \frac{d}{d\theta} \cos^{-1}(T(\theta)) \right| \leq m \quad (3.5)$$

Following [27], it can be proven by means of this inequality that

Proposition 1.

$$\mu_{[-1,1]}(x, y) = \sup_{\|P\|_{\infty, [-1,1]} \leq 1} (\deg P)^{-1} |\cos^{-1}(P(x)) - \cos^{-1}(P(y))|, \quad (3.6)$$

where P varies in $\mathbb{P}([-1, 1])$.

Proof. Let $T(\theta) = P(\cos(\theta))$ and $a = \cos(\theta_a)$, $b = \cos(\theta_b)$. Then

$$|\cos^{-1}(T(\theta_a)) - \cos^{-1}(T(\theta_b))| = \left| \int_{\theta_a}^{\theta_b} \frac{d}{d\theta} \cos^{-1}(T(\theta)) d\theta \right| \underbrace{\leq}_{(3.5)} \deg P \int_{\theta_a}^{\theta_b} d\theta$$

But $\cos^{-1} a = \theta_a$ and $\cos^{-1} b = \theta_b$. Hence, we get

$$|\cos^{-1}(T(\theta_a)) - \cos^{-1}(T(\theta_b))| \leq \deg P |\cos^{-1} a - \cos^{-1} b|$$

from which follows the claim. \square .

By generalizing, we define

$$\mu_{\Omega}(\mathbf{x}, \mathbf{y}) = \sup_{\|P\|_{\infty, \Omega} \leq 1} (\deg P)^{-1} |\cos^{-1}(P(\mathbf{x})) - \cos^{-1}(P(\mathbf{y}))|, \quad \mathbf{x}, \mathbf{y} \in \Omega \subset \mathbb{R}^d, \quad (3.7)$$

where P varies in $\mathbb{P}(\Omega)$, which is the *Dubiner metric* on the compact Ω .

In view of the properties of such a metric (cf. [27]), one may state [5] the following

\hookrightarrow **conjecture:** *nearly-optimal interpolation points on a compact Ω are asymptotically equidistributed with respect to the Dubiner metric on Ω .* \leftarrow

This suggests a general way to produce candidates to be good interpolation points, *once we know the Dubiner metric* for the compact set Ω . Unfortunately the Dubiner metric is explicitly known only in very few cases, for $d = 2$ namely the square and the circle

- $\Omega = [-1, 1]^2$, $\mathbf{x} = (x_1, x_2)$, $\mathbf{y} = (y_1, y_2)$:

$$\mu_{\Omega}(\mathbf{x}, \mathbf{y}) = \max\{|\cos^{-1}(x_1) - \cos^{-1}(y_1)|, |\cos^{-1}(x_2) - \cos^{-1}(y_2)|\}.$$

- $\Omega = \{\mathbf{x} : |\mathbf{x}| \leq 1\}$, $\mathbf{x} = (x_1, x_2)$, $\mathbf{y} = (y_1, y_2)$:

$$\mu_{\Omega}(\mathbf{x}, \mathbf{y}) = \left| \cos^{-1} \left(x_1 x_2 + y_1 y_2 + \sqrt{1 - x_1^2 - y_1^2} \sqrt{1 - x_2^2 - y_2^2} \right) \right|.$$

Quasi-uniform points in the Dubiner metric

We tested the conjecture above on four sets of points on the square which are (asymptotically) equidistributed with respect to the Dubiner metric. The first one is obtained numerically using a reasonable definition of asymptotic equidistribution in a given metric. The other three are given by explicit formulas and are exactly equidistributed in the Dubiner metric.

Following [25] we can construct a sequence of points which are asymptotically equidistributed in a compact Ω with respect to a given metric ν , by means of the following *geometric greedy algorithm*:

- Let Ω be a compact set in \mathbb{R}^d , and consider $X_0 = \{\mathbf{x}_0\}$ where $\mathbf{x}_0 \in \partial\Omega$.
- If $X_j \subset \Omega$ is finite and consisting of j points, choose $\mathbf{x}_{j+1} \in \Omega \setminus X_j$ so that its distance to X_j is maximal; then, form $X_{j+1} := X_j \cup \{\mathbf{x}_{j+1}\}$.

In Fig. 3.4 we show the distribution of $N = 496$ (which correspond to polynomial degree 30) quasi-uniform Dubiner (shortly quD) points on the square, computed by the geometric greedy algorithm starting from a sufficiently dense random discretization of the square. We chose a discretization with N^3 random points. Why N^3 ? This will become more evident when we will introduce the so-called WAM (see last Chapter of these lecture notes).

Notice. As distances we used the Dubiner on the square, getting the DUB from the RND points, or the Euclidean distance, getting the EUC points.

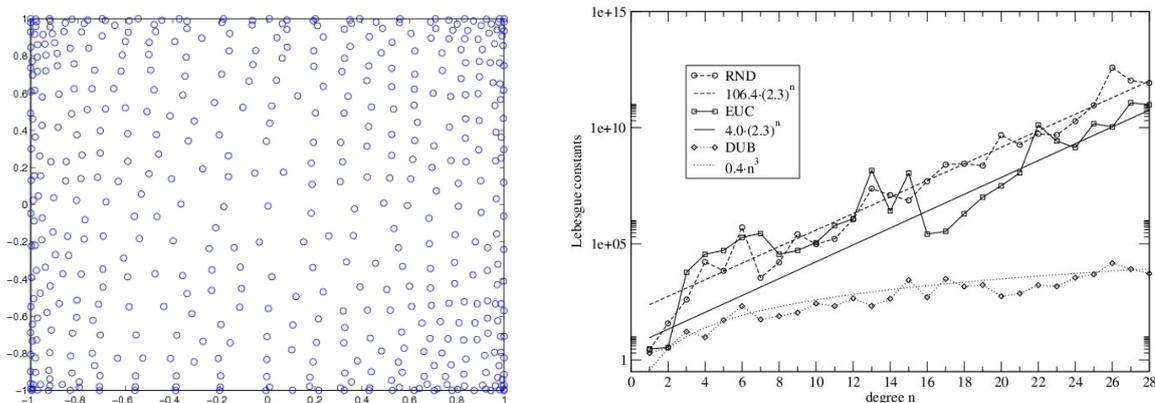


Figure 3.4: Left: 496 (i.e. degree 30) quasi-uniform Dubiner (DUB) points for the square; Right: Lebesgue constants for DUB points, quasi-uniform Euclidean (EUC) points and random (RND) points.

The comparison of the Lebesgue constants in Fig. 3.4 shows that the quasi-uniform Dubiner points are much better for polynomial interpolation than the quasi-uniform Euclidean (shortly EUC) and the random ones (shortly RND), since the growth of their Lebesgue constant is polynomial instead of exponential in the degree. However, they are not still satisfactory since the growth is of order $N^{3/2}$, which is bigger than the theoretical bound of the Fekete points, i.e. $\Lambda_n(F_N) \leq N$.

The Morrow-Patterson (MP) points

C. R. Morrow and T. N. L. Patterson (cf. [34]), proposed for cubature purposes the following set of nodes on the square. For n , a positive *even* integer, consider the points $X_N^{MP} = \{(x_m, y_k)\} \subset [-1, 1]^2$ given by

$$x_m = \cos\left(\frac{m\pi}{n+2}\right), \quad y_k = \begin{cases} \cos\left(\frac{2k\pi}{n+3}\right) & m \text{ odd} \\ \cos\left(\frac{(2k-1)\pi}{n+3}\right) & m \text{ even} \end{cases} \quad (3.8)$$

$1 \leq m \leq n+1$, $1 \leq k \leq \frac{n}{2} + 1$. It is easily seen that these points are exactly equally spaced w.r.t. the Dubiner metric, i.e. they have a constant pointwise separation distance, cf. Section 3.1. This set consists of $N = \binom{n+2}{2}$ points, which is equal to $\dim(\mathbb{P}_n(\mathbb{R}^2))$, and is unisolvent for polynomial interpolation on the square.

\Leftrightarrow As for the growth of the Lebesgue constant, in [24] we proved that $\Lambda_n^{MP} = \mathcal{O}(n^3)$, improving a significantly the bound $\Lambda_n^{MP} = \mathcal{O}(n^6)$ in [5]. From our experiments we obtained $\Lambda_n^{MP} = \mathcal{O}(n^2)$ as can be seen in Fig. 3.6. In particular we found that Λ_n^{MP} can be least-square fitted with the quadratic polynomial $(0.7n+1)^2$, which is smaller than N , i.e. than the theoretical bound for Fekete points.

Extended Morrow-Patterson (EMP) points

In analogy with the one-dimensional setting [14], we tried to improve the Lebesgue constant by considering the extended Morrow-Patterson points, which correspond to using extended Chebyshev nodes in (3.8), i.e. $X_N^{EMP} = \{(x_m, y_k)\} \subset [-1, 1]^2$ given by

$$x_m = \frac{1}{\alpha_n} \cos\left(\frac{m\pi}{n+2}\right), \quad y_k = \begin{cases} \frac{1}{\beta_n} \cos\left(\frac{2k\pi}{n+3}\right) & m \text{ odd} \\ \frac{1}{\beta_n} \cos\left(\frac{(2k-1)\pi}{n+3}\right) & m \text{ even} \end{cases} \quad (3.9)$$

$1 \leq m \leq n+1$, $1 \leq k \leq \frac{n}{2} + 1$, where the *dilation coefficients* $1/\alpha_n$ and $1/\beta_n$ correspond to

$$\alpha_n = \cos(\pi/(n+2)), \quad \beta_n = \cos(\pi/(n+3)).$$

\Leftrightarrow As the Morrow-Patterson points, the EMP points are exactly equally spaced w.r.t. the Dubiner metric and are again unisolvent for polynomial interpolation of degree n . Indeed, the Vandermonde matrix of X_N^{EMP} w.r.t. the canonical basis of $\mathbb{P}_n(\mathbb{R}^2)$, is given by the Vandermonde matrix of the Morrow-Patterson points, where each column is scaled by a suitable constant. In particular, the column corresponding to the monomial $x^i y^j$ is multiplied by $\alpha_n^{-i} \beta_n^{-j}$: hence, $|VDM(X_N^{EMP})|$ is strictly greater than $|VDM(X_N^{MP})|$, i.e. it cannot vanish. Moreover, as clear from Figure 3.6, $\Lambda_n(X_N^{EMP}) < \Lambda_n(X_N^{MP})$.

Modified Morrow-Patterson points or **Padua (PD) points**

Here we recall how the Padua points were discovered.

For n a positive *even* integer consider the points $(x_m, y_k) \in [-1, 1]^2$ given by

$$x_m = \cos\left(\frac{(m-1)\pi}{n}\right), \quad y_k = \begin{cases} \cos\left(\frac{(2k-2)\pi}{n+1}\right) & m \text{ odd} \\ \cos\left(\frac{(2k-1)\pi}{n+1}\right) & m \text{ even} \end{cases} \quad (3.10)$$

$1 \leq m \leq n+1$, $1 \leq k \leq \frac{n}{2} + 1$. These are *modified* Morrow-Patterson points that were firstly discussed in Padua by the authors with L. Bos and S. Waldron in 2003, and so we have decided to call them **Padua points** (shortly PD points); again, they are exactly equispaced w.r.t. the Dubiner metric on the square. For a sketch of the distribution of PD points and for a comparison with MP and EMP at small degree, see Fig. 3.5.

The Padua points were, to our knowledge, the best known nodes for polynomial interpolation on the square. In fact, from our experiments, $\Lambda_n^{PD} = \mathcal{O}(\log^2 n)$ (see Fig. 3.6 below).

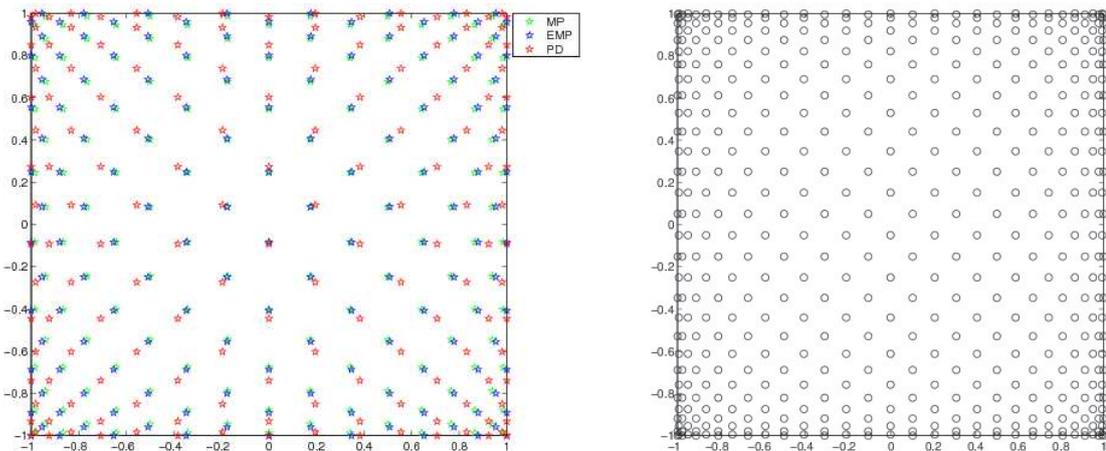


Figure 3.5: Left: Morrow-Patterson (MP), Extended Morrow-Patterson (EMP) and Padua (PD) points, for degree $n = 8$. Right: Padua points for degree $n = 30$.

As a final result, in Table 3.4–3.6, we show the interpolation errors (in the sup-norm) for the three test functions considered. We note that the errors have been computed on the same uniform control grid used to estimate the Lebesgue constants.

Exercise 5. Write a Matlab toolbox that performs the following tasks

- Determine all families of near-optimal interpolation points on the square presented in this chapter.
- Determine the corresponding Lebesgue constants
- Compute the interpolation errors

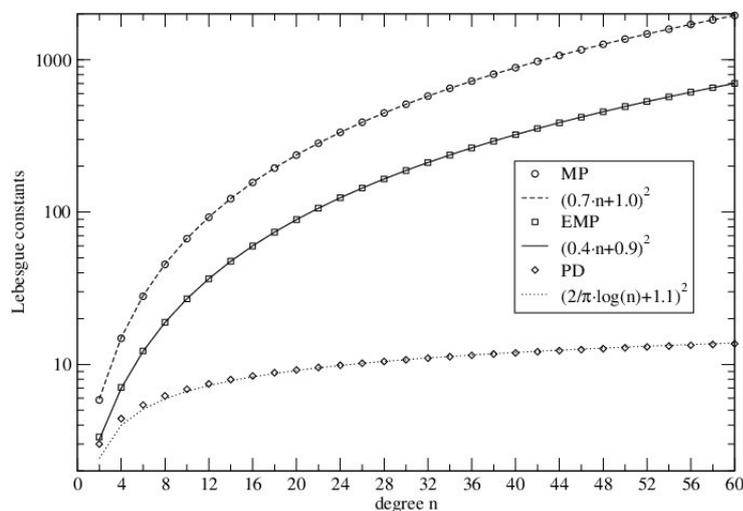


Figure 3.6: The behaviour of the Lebesgue constants for Morrow-Patterson (MP), Extended Morrow-Patterson (EMP), Padua (PD) points up to degree 60, and their least-squares fitting curves.

	$n = 34$	Λ_{34}	$n = 48$	Λ_{48}	$n = 62$	Λ_{62}	$n = 76$	Λ_{76}
MP	$1.3 \cdot 10^{-3}$	649	$2.6 \cdot 10^{-6}$	1264	$1.1 \cdot 10^{-9}$	2082	$2.0 \cdot 10^{-13}$	3102
EMP	$6.3 \cdot 10^{-4}$	237	$1.3 \cdot 10^{-6}$	456	$5.0 \cdot 10^{-10}$	746	$5.4 \cdot 10^{-14}$	1106
PD	$4.3 \cdot 10^{-5}$	11	$3.3 \cdot 10^{-8}$	13	$5.4 \cdot 10^{-12}$	14	$1.9 \cdot 10^{-14}$	15

Table 3.4: Interpolation errors for the Franke function.

n	34	48	62	76
MP on $[-1, 1]^2$	$1.8 \cdot 10^{-4}$	$5.1 \cdot 10^{-5}$	$1.9 \cdot 10^{-5}$	$8.8 \cdot 10^{-6}$
MP on $[0, 2]^2$	$1.0 \cdot 10^{-8}$	$3.8 \cdot 10^{-10}$	$3.7 \cdot 10^{-11}$	$2.3 \cdot 10^{-11}$
EMP on $[-1, 1]^2$	$6.5 \cdot 10^{-5}$	$1.8 \cdot 10^{-5}$	$6.7 \cdot 10^{-6}$	$3.0 \cdot 10^{-6}$
EMP on $[0, 2]^2$	$7.2 \cdot 10^{-9}$	$2.6 \cdot 10^{-10}$	$2.4 \cdot 10^{-11}$	$8.6 \cdot 10^{-12}$
PD on $[-1, 1]^2$	$3.6 \cdot 10^{-6}$	$6.5 \cdot 10^{-7}$	$1.8 \cdot 10^{-7}$	$6.5 \cdot 10^{-8}$
PD on $[0, 2]^2$	$2.8 \cdot 10^{-9}$	$9.3 \cdot 10^{-11}$	$9.4 \cdot 10^{-12}$	$6.4 \cdot 10^{-12}$

Table 3.5: Interpolation errors for the function $f_2(x_1, x_2) = (x_1^2 + x_2^2)^{5/2}$.

n	34	48	62	76
MP on $[-1, 1]^2$	$4.4 \cdot 10^{-1}$	$4.4 \cdot 10^{-1}$	$4.4 \cdot 10^{-1}$	$4.4 \cdot 10^{-1}$
MP on $[0, 2]^2$	$8.8 \cdot 10^{-4}$	$2.8 \cdot 10^{-4}$	$2.6 \cdot 10^{-4}$	$1.7 \cdot 10^{-5}$
EMP on $[-1, 1]^2$	$1.4 \cdot 10^{-1}$	$1.4 \cdot 10^{-1}$	$1.4 \cdot 10^{-1}$	$1.4 \cdot 10^{-1}$
EMP on $[0, 2]^2$	$8.3 \cdot 10^{-4}$	$2.6 \cdot 10^{-4}$	$2.1 \cdot 10^{-4}$	$2.1 \cdot 10^{-5}$
PD on $[-1, 1]^2$	$3.7 \cdot 10^{-2}$	$2.7 \cdot 10^{-2}$	$2.1 \cdot 10^{-2}$	$1.7 \cdot 10^{-2}$
PD on $[0, 2]^2$	$7.3 \cdot 10^{-4}$	$3.7 \cdot 10^{-4}$	$7.0 \cdot 10^{-6}$	$4.6 \cdot 10^{-6}$

Table 3.6: Interpolation errors for the function $f_3(x_1, x_2) = (x_1^2 + x_2^2)^{1/2}$.

Lecture 4

The Padua points

It is impossible to discuss all the aspects and discoveries done on the Padua points. For completeness, the interested readers, can refer to the link

<http://www.math.unipd.it/~marcov/CAApadua.html>

which contains an up-to-date bibliography (papers, preprints, presentations and software), concerning this important family of interpolation points.

The points are also indexed in Wikipedia

https://en.wikipedia.org/wiki/Padua_points.

4.1 The generating curve of the Padua points

We provide here a slightly different definition of Padua points.

Let start by taking the $n + 1$ Chebyshev-Lobatto points on $[-1, 1]$

$$C_{n+1} := \left\{ z_j^n = \cos \left(\frac{(j-1)\pi}{n} \right), j = 1, \dots, n+1 \right\}.$$

We then consider two subsets of points with *odd* and *even* indices

$$\begin{aligned} C_{n+1}^o &:= \{ z_j^n, j = 1, \dots, n+1, j \text{ odd} \} \\ C_{n+1}^e &:= \{ z_j^n, j = 1, \dots, n+1, j \text{ even} \} \end{aligned}$$

Then, the Padua points (of the first family) are the set

$$\text{Pad}_n := (C_{n+1}^o \times C_{n+2}^o) \cup (C_{n+1}^e \times C_{n+2}^e) \subset C_{n+1} \times C_{n+2}. \quad (4.1)$$

These points have cardinality of the space of bivariate polynomials of degree $\leq n$, i.e. $N = (n+1)(n+2)/2$.

For a given $n \geq 1$, consider the parametric curve

$$\gamma_n(t) = (\cos nt, \cos(n+1)t), \quad 0 \leq t \leq \pi. \quad (4.2)$$

This curve is obviously all contained in the square $[-1, 1]^2$ (the curve is made by cosines). The curve is also an algebraic curve given by $T_{n+1}(x) = T_n(y)$ where T_n denotes the Chebyshev polynomial of the first kind of degree n . In Figure 4.1 we plot the Padua points and the corresponding parametric curve. It is also a **Lissajous** curve. Indeed, all Lissajous curves are algebraic curves and implicit equations of Lissajous curves can be expressed by Chebyshev polynomials. On the other hand, Chebyshev polynomials are Lissajous curves. In fact a parametrization of $y = T_n(x)$, $|x| \leq 1$ is

$$\begin{cases} x = \cos t \\ y = -\sin\left(nt - \frac{\pi}{2}\right) \end{cases} \quad 0 \leq t \leq \pi$$

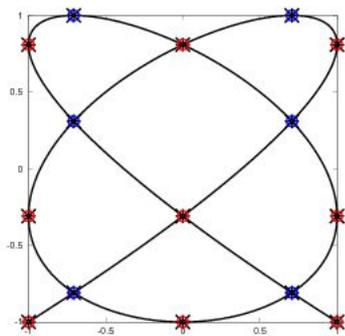


Figure 4.1: Padua points and their generating curve for $n = 4$. The grids of odd and even indices are indicated with different colours and style.

On $\gamma_n(t)$ we consider the *equally spaced* points

$$\mathcal{A}_n = \left\{ \gamma_n\left(\frac{k}{n(n+1)}\pi\right) : k = 0, 1, \dots, n(n+1) \right\}.$$

Remark. As an example, consider \mathcal{A}_5 . It has $30 + 1 = 31$ points, but only $21 = \dim(\mathbb{P}_5(\mathbb{R}^2))$ are **different** points.

Notice that each of the interior points lies on a self-intersection point of the curve. Indeed, it is easy to see that

$$\gamma_n\left(\frac{jn + m(n+1)}{n(n+1)}\pi\right) = \gamma_n\left(\frac{jn - m(n+1)}{n(n+1)}\pi\right)$$

for any integers j, m . Hence we may naturally describe the set of distinct points \mathcal{A}_n as

$$\mathcal{A}_n = \left\{ A_{jm} := \gamma_n\left(\frac{jn + m(n+1)}{n(n+1)}\pi\right) : j, m \geq 0 \text{ and } j + m \leq n \right\}. \quad (4.3)$$

More specifically, we may classify the points of \mathcal{A}_n by

Interior Points:

$$A_{jm} = \gamma_n \left(\frac{jn + m(n+1)}{n(n+1)}\pi \right) = \left((-1)^m \cos \left(\frac{jn}{n+1}\pi \right), (-1)^j \cos \left(\frac{m(n+1)}{n}\pi \right) \right),$$

$j, m > 0$ and $j + m \leq n$.

Vertex Points: $A_{00} = (1, 1)$ and $A_{0,n} = ((-1)^n, (-1)^{(n+1)})$.

Vertical Edge Points: $A_{0m} = \left((-1)^m, \cos \left(\frac{m(n+1)}{n}\pi \right) \right), m = 1, 2, \dots, n$.

Horizontal Edge Points: $A_{j0} = \left(\cos \left(\frac{jn}{n+1}\pi \right), (-1)^j \right), j = 1, 2, \dots, n$.

It follows then that the number of distinct points in \mathcal{A}_n is indeed $\binom{n+2}{2}$ and $\mathcal{A}_n = \text{Pad}_n$.

4.1.1 Reproducing kernel in Hilbert spaces

We provide here the basic tools on RKHS used to prove some results in many multivariate interpolation problems, polynomial or not. We have used these concepts also for proving some results for Padua points.

Definition 7. Let \mathcal{H} be a real Hilbert space with inner product $(\cdot, \cdot)_{\mathcal{H}}$. A function $K : \Omega \times \Omega \rightarrow \mathbb{R}$, $\Omega \subseteq \mathbb{R}^d$, is called a **reproducing kernel** for \mathcal{H} if the following two properties hold:

- (i) $K(\cdot, x) \in \mathcal{H}, \forall x \in \Omega$
- (ii) $f(x) = (f, K(\cdot, x))_{\mathcal{H}}, \forall f \in \mathcal{H}$ and all $x \in \Omega$

Remark: property (ii) is known as the **reproduction property**.

The existence of a RK is equivalent to the fact that the point evaluation functionals δ_x on Ω are bounded. That is, there exists a positive constant $M = M_x$ s.t.

$$|\delta_x f| = |f(x)| \leq M \|f\|_{\mathcal{H}}, \quad \forall f \in \mathcal{H}, x \in \Omega,$$

where the inequality is due to the Riesz representation in Hilbert spaces.

Moreover, a reproduction kernel, if exists, is **unique**.

The main properties of a RKHS are

1. $K(x, y) = (K(\cdot, y), K(\cdot, x))_{\mathcal{H}}, \quad \forall x, y \in \Omega.$
2. $K(x, y) = K(y, x), \quad \forall x, y \in \Omega.$
3. Convergence in Hilbert space norm implies pointwise convergence, i.e. if $\|f - f_n\|_{\mathcal{H}} \rightarrow 0$ then $|f(x) - f_n(x)| \rightarrow 0$ for all $x \in \Omega.$
4. A RKHS is **positive definite**.

Proof.

1. By (i) of the Definition 7, $K(\cdot, y) \in \mathcal{H}, \quad \forall y \in \Omega.$ Then (ii) gives $K(x, y) = (K(\cdot, y), K(\cdot, x))_{\mathcal{H}}, \quad \forall x, y \in \Omega.$
2. Follows from 1. by the symmetry of the inner product.
3. Using the reproduction property of K and the Cauchy-Schwarz inequality

$$|f(x) - f_n(x)| = |(f - f_n, K(\cdot, x))_{\mathcal{H}}| \leq \|f - f_n\|_{\mathcal{H}} \|K(\cdot, x)\|_{\mathcal{H}}.$$

4. In fact, for given $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ points of \mathbb{R}^d and a non-zero vector $\mathbf{c} \in \mathbb{R}^N,$ we have

$$\begin{aligned} \sum_{j=1}^N \sum_{k=1}^N c_j c_k K(\mathbf{x}_j, \mathbf{x}_k) &= \sum_{j=1}^N \sum_{k=1}^N c_j c_k (K(\cdot, \mathbf{x}_j), K(\cdot, \mathbf{x}_k))_{\mathcal{H}} \\ &= \left(\sum_j c_j K(\cdot, \mathbf{x}_j), \sum_k c_k K(\cdot, \mathbf{x}_k) \right)_{\mathcal{H}} \\ &= \left\| \sum_j c_j K(\cdot, \mathbf{x}_j) \right\|_{\mathcal{H}}^2 \geq 0. \end{aligned}$$

4.2 The basic Lagrange polynomials and Lebesgue constant

Going back to the Padua points. The next step is a closed form for the basic Lagrange polynomials with the aim to find an upper bound for the Lebesgue constant. The results in this section can be found in [9].

Consider the weighted integral

$$I(f) := \frac{1}{\pi^2} \int_{[-1,1]^2} f(x, y) \frac{1}{\sqrt{1-x^2}} \frac{1}{\sqrt{1-y^2}} dx dy$$

and the associated inner product

$$(f, g) := \frac{1}{\pi^2} \int_{[-1,1]^2} f(x, y) g(x, y) \frac{1}{\sqrt{1-x^2}} \frac{1}{\sqrt{1-y^2}} dx dy. \quad (4.4)$$

Note that the polynomials $\{T_j(x)T_m(y) : j + m \leq n\}$ form an orthogonal basis for $\mathbb{P}_n(\mathbb{R}^2)$ with respect to this inner product. Further, the finite dimensional space $\mathbb{P}_n(\mathbb{R}^2)$ has a reproducing kernel with respect to this inner product, which we denote by $K_n(\cdot; \cdot)$. Specifically, for each $A, B \in [-1, 1]^2$, $K_n(A; \cdot) \in \mathbb{P}_n(\mathbb{R}^2)$, $K_n(\cdot; B) \in \mathbb{P}_n(\mathbb{R}^2)$ and

$$(p, K_n(A; \cdot)) = p(A)$$

for all $p \in \mathbb{P}_n(\mathbb{R}^2)$.

We will show that there is a remarkable quadrature formula for $I(f)$ based on the Padua points (Theorem 16 below).

Lemma 1. *For all $p \in \mathbb{P}_{2n}(\mathbb{R}^2)$ we have*

$$\frac{1}{\pi^2} \int_{[-1,1]^2} p(x, y) \frac{1}{\sqrt{1-x^2}} \frac{1}{\sqrt{1-y^2}} dx dy = \frac{1}{\pi} \int_0^\pi p(\gamma_n(t)) dt.$$

Proof. We need only show that this holds for

$$p(x, y) = T_j(x)T_m(y), \quad j + m \leq 2n.$$

But, when $j = m = 0$, $p(x, y) = 1$, and clearly both sides are equal to 1. Otherwise, if $(j, m) \neq (0, 0)$, the left side equals 0 while the right side equals

$$\begin{aligned} \frac{1}{\pi} \int_0^\pi T_j(\cos(nt))T_m(\cos((n+1)t)) dt &= \frac{1}{\pi} \int_0^\pi \cos(jnt) \cos(m(n+1)t) dt \\ &= 0 \end{aligned}$$

provided $jn \neq m(n+1)$. But since n and $n+1$ are relatively prime, $jn = m(n+1)$ implies that $j = \alpha(n+1)$ and $m = \alpha n$ for some positive integer α . Hence $j+m = \alpha(n+(n+1)) > 2n$. \square

Theorem 16. *Consider the following weights associated to the Padua points $A \in \mathcal{A}_n$:*

$$w_A := \frac{1}{n(n+1)} \begin{cases} 1/2 & \text{if } A \in \mathcal{A}_n \text{ is a vertex point} \\ 1 & \text{if } A \in \mathcal{A}_n \text{ is an edge point} \\ 2 & \text{if } A \in \mathcal{A}_n \text{ is an interior point} \end{cases}.$$

Then, if $p(x, y) \in \mathbb{P}_{2n}(\mathbb{R}^2)$ is such that $(p(x, y), T_{2n}(y)) = 0$, we have

$$\frac{1}{\pi^2} \int_{[-1,1]^2} p(x, y) \frac{1}{\sqrt{1-x^2}} \frac{1}{\sqrt{1-y^2}} dx dy = \sum_{A \in \mathcal{A}_n} w_A p(A).$$

In particular, since $T_{2n}(y) = 2(T_n(y))^2 - 1$, this quadrature formula holds for $p = fg$ with $f, g \in \mathbb{P}_n(\mathbb{R}^2)$ and either $(f(x, y), T_n(y)) = 0$ or $(g(x, y), T_n(y)) = 0$.

For $A \in \mathcal{A}_n$ we will let y_A denote the y -coordinate of A .

Theorem 17. (The basic Lagrange polynomials)

For $B \in \mathcal{A}_n$, set

$$L_B(x, y) := w_B[K_n(B; (x, y)) - T_n(y)T_n(y_B)], \quad B \in \mathcal{A}_n. \quad (4.5)$$

Then we have the interpolation formula

$$\sum_{B \in \mathcal{A}_n} L_B(A)p(B) = p(A), \quad \forall p \in \Pi_n(\mathbb{R}^2), \quad \forall A \in \mathcal{A}_n. \quad (4.6)$$

The proof is again in [9].

The polynomials L_B are indeed the fundamental Lagrange polynomials, i.e., they satisfy

$$L_B(A) = \delta_{A,B}, \quad A, B \in \mathcal{A}_n. \quad (4.7)$$

This can be easily verified using the following compact formula for the reproducing kernel K_n proved in [49].

Lemma 2. (cf. [9])

For $A, B \in [-1, 1]^2$ write $A = (\cos(\theta_1), \cos(\theta_2))$, $B = (\cos(\phi_1), \cos(\phi_2))$. Then

$$\begin{aligned} K_n(A; B) &= D_n(\theta_1 + \phi_1, \theta_2 + \phi_2) + D_n(\theta_1 + \phi_1, \theta_2 - \phi_2) \\ &\quad + D_n(\theta_1 - \phi_1, \theta_2 + \phi_2) + D_n(\theta_1 - \phi_1, \theta_2 - \phi_2) \end{aligned}$$

where

$$D_n(\alpha, \beta) = \frac{1}{2} \frac{\cos((n+1/2)\alpha) \cos(\alpha/2) - \cos((n+1/2)\beta) \cos(\beta/2)}{\cos(\alpha) - \cos(\beta)}. \quad (4.8)$$

Remark. Notice that the computation of D_n in (4.8) is very unstable when $\cos \alpha \approx \cos \beta$ being like a first divided difference, and thus has to be stabilized. A stable formula to compute D_n can be obtained by simple trigonometric manipulations leading to a formula in terms of *Chebyshev polynomials of second kind*, as done in [6, §2] for interpolation at Xu points.

Finally we have the required bound

Theorem 18. (cf. [9])

There is a constant $C > 0$ such that the Lebesgue function is bounded,

$$\Lambda_n(x, y) \leq C(\log n)^2, \quad n \geq 2, \quad (x, y) \in [-1, 1]^2.$$

The proof of this Theorem has been omitted.

4.3 Fast interpolation and cubature at the Padua points

In this section we show fast algorithms to compute and evaluate the interpolant and the evaluation of integrals at the Padua points. The implementation of these tasks has been done in Matlab/Octave. The software `Padua2DM` is available at the CAA-research group web page <http://www.math.unipd.it/~marcov/CAAssoft.html>.

The results of this section are mainly taken from the paper [18].

4.3.1 Interpolation

In what follows, $\mathbf{x} = (x_1, x_2) \in [-1, 1]^2$ is a target point and $\xi = (\xi_1, \xi_2) \in \text{Pad}_n \subset [-1, 1]^2$ is an element of the set of the Padua points of order n .

The polynomial interpolation formula can be written in the bivariate Chebyshev orthonormal basis as

$$\begin{aligned} \mathcal{L}_n f(\mathbf{x}) &= \sum_{\xi \in \text{Pad}_n} f(\xi) w_\xi \left(K_n(\xi, \mathbf{x}) - \frac{1}{2} \hat{T}_n(\xi_1) \hat{T}_n(x_1) \right) \\ &= \sum_{k=0}^n \sum_{j=0}^k c_{j,k-j} \hat{T}_j(x_1) \hat{T}_{k-j}(x_2) - \frac{1}{2} \sum_{\xi \in \text{Pad}_n} f(\xi) w_\xi \hat{T}_n(\xi_1) \hat{T}_0(\xi_2) \hat{T}_n(x_1) \hat{T}_0(x_2) \\ &= \sum_{k=0}^n \sum_{j=0}^k c_{j,k-j} \hat{T}_j(x_1) \hat{T}_{k-j}(x_2) - \frac{c_{n,0}}{2} \hat{T}_n(x_1) \hat{T}_0(x_2) \end{aligned} \quad (4.9)$$

where the coefficients are defined as

$$c_{j,k-j} = \sum_{\xi \in \text{Pad}_n} f(\xi) w_\xi \hat{T}_j(\xi_1) \hat{T}_{k-j}(\xi_2), \quad 0 \leq j \leq k \leq n \quad (4.10)$$

and can be computed once and for all.

The MM algorithm

We can define the $(n+1) \times (n+2)$ matrix computed corresponding to the Chebyshev-like grid $C_{n+1} \times C_{n+2}$ with entries

$$\mathbb{G}(f) = (g_{r,s}) = \begin{cases} w_\xi f(\xi) & \text{if } \xi = (z_r^n, z_s^{n+1}) \in \text{Pad}_n \\ 0 & \text{if } \xi = (z_r^n, z_s^{n+1}) \in (C_{n+1} \times C_{n+2}) \setminus \text{Pad}_n \end{cases}$$

In [17] we have computed the coefficients (4.10) by a double matrix-matrix product involving the matrix $\mathbb{G}(f)$. The construction was as follows.

Given a vector $S = (s_1, \dots, s_m) \in [-1, 1]^m$, first we define the rectangular Chebyshev matrix

$$\mathbb{T}(S) = \begin{pmatrix} \hat{T}_0(s_1) & \cdots & \hat{T}_0(s_m) \\ \vdots & \cdots & \vdots \\ \hat{T}_n(s_1) & \cdots & \hat{T}_n(s_m) \end{pmatrix} \in \mathbb{R}^{(n+1) \times m} \quad (4.11)$$

Then it is easy to check that the coefficients $c_{j,l}$, $0 \leq j \leq n$, $0 \leq l \leq n - j$ are the entries of the upper-left triangular part of the matrix

$$\mathbb{C}(f) = \mathbb{T}(C_{n+1}) \mathbb{G}(f) (\mathbb{T}(C_{n+2}))^t \quad (4.12)$$

where, with a little abuse of notation, $C_{n+1} = (z_1^n, \dots, z_{n+1}^n)$ is the *vector* of the Chebyshev–Gauss–Lobatto points, too.

\hookrightarrow A slightly more refined algorithm can be obtained, by exploiting the fact that the Padua points are union of two Chebyshev subgrids.

Indeed, defining the two matrices

$$\mathbb{G}_1(f) = (w_\xi f(\xi), \xi = (z_r^n, z_s^{n+1}) \in C_{n+1}^E \times C_{n+2}^O)$$

$$\mathbb{G}_2(f) = (w_\xi f(\xi), \xi = (z_r^n, z_s^{n+1}) \in C_{n+1}^O \times C_{n+2}^E)$$

then we can compute the coefficient matrix as

$$\mathbb{C}(f) = \mathbb{T}(C_{n+1}^E) \mathbb{G}_1(f) (\mathbb{T}(C_{n+2}^O))^t + \mathbb{T}(C_{n+1}^O) \mathbb{G}_2(f) (\mathbb{T}(C_{n+2}^E))^t \quad (4.13)$$

by multiplying matrices of smaller dimension than those in (4.12). In fact, in (4.13), the \mathbb{T} matrices have dimensions $n \times \frac{n}{2}$ and the matrices G_1, G_2 about $\frac{n}{2} \times \frac{n}{2}$.

We term this approach *MM (Matrix Multiplication)* in the numerical tests.

The FFT-based algorithm

The spectral structure of the Padua points allows to use a FFT-approach. Indeed, the coefficients $c_{j,l}$ can be rewritten as

$$\begin{aligned} c_{j,l} &= \sum_{\xi \in \text{Pad}_n} f(\xi) w_\xi \hat{T}_j(\xi_1) \hat{T}_l(\xi_2) = \sum_{r=0}^n \sum_{s=0}^{n+1} g_{r,s} \hat{T}_j(z_r^n) \hat{T}_l(z_s^{n+1}) \\ &= \beta_{j,l} \sum_{r=0}^n \sum_{s=0}^{n+1} g_{r,s} \cos \frac{jr\pi}{n} \cos \frac{ls\pi}{n+1} = \beta_{j,l} \sum_{s=0}^{M-1} \left(\sum_{r=0}^{N-1} g_{r,s}^0 \cos \frac{2jr\pi}{N} \right) \cos \frac{2ls\pi}{M} \end{aligned}$$

where $N = 2n$, $M = 2(n+1)$ and

$$\beta_{j,l} = \begin{cases} 1 & j = l = 0 \\ 2 & j \neq 0, l \neq 0 \\ \sqrt{2} & \text{otherwise} \end{cases} \quad g_{r,s}^0 = \begin{cases} g_{r,s} & 0 \leq r \leq n \text{ and } 0 \leq s \leq n+1 \\ 0 & r > n \text{ or } s > n+1 \end{cases}$$

```

Input: Gf ↔ G(f)

Gfhat = real(fft(Gf, 2*n));
Gfhat = Gfhat(1:n+1, :);
Gfhathat = real(fft(Gfhat, 2*(n+1), 2));
Cof = Gfhathat(:, 1:n+1);
Cof = 2*Cof;
Cof(1, :) = Cof(1, :)/sqrt(2);
Cof(:, 1) = Cof(:, 1)/sqrt(2);
Cof = fliplr(triu(fliplr(Cof)));
Cof(n+1, 1) = Cof(n+1, 1)/2;

```

Output: Cof ↔ C₀(f)

Table 4.1: Matlab code for the fast computation of the coefficient matrix.

Then, it is possible to recover the coefficients $c_{j,l}$ by a double Discrete Fourier Transform, namely

$$\begin{aligned} \hat{g}_{j,s} &= \text{REAL} \left(\sum_{r=0}^{N-1} g_{r,s}^0 e^{-2\pi i jr/N} \right), \quad 0 \leq j \leq n, \quad 0 \leq s \leq M-1 \\ \frac{c_{j,l}}{\beta_{j,l}} = \hat{g}_{j,l} &= \text{REAL} \left(\sum_{s=0}^{M-1} \hat{g}_{j,s} e^{-2\pi i ls/M} \right), \quad 0 \leq j \leq n, \quad 0 \leq l \leq n-j \end{aligned} \quad (4.14)$$

we call $\mathbb{C}_0(f)$ the interpolation coefficients matrix

$$\mathbb{C}_0(f) = (c'_{j,l}) = \begin{pmatrix} c_{0,0} & c_{0,1} & \cdots & \cdots & c_{0,n} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,n-1} & 0 \\ \vdots & \vdots & & & \vdots \\ c_{n-1,0} & c_{n-1,1} & 0 & \cdots & 0 \\ \frac{c_{n,0}}{2} & 0 & \cdots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{(n+1) \times (n+1)} \quad (4.15)$$

which is essentially the upper-left triangular part of the matrix $\mathbb{C}(f)$ in (4.12), but the modification on the last element of the first column. The Matlab code for the computation of the coefficient matrix by a double Fast Fourier Transform is reported in Table 4.1. We just notice that all the indexes starting from 0 are shifted by 1 (as required in Matlab).

Remark. The complexity $c(n)$ of the presented methods are as follows: $c(n) \sim 2n^3$ for the MM-algorithm, $c(n) \sim \mathcal{O}(n^2 \log n)$ for the FFT-algorithm.

Evaluation of the interpolant

As we have seen, we firstly compute $C_0(f)$ **once and for all**, then the interpolant of a function f , at every point $\mathbf{x} = (x_1, x_2) \in [-1, 1]^2$ can be evaluated by the matrix product

$$\mathcal{L}_n f(\mathbf{x}) = (\mathbb{T}(x_1))^t \mathbb{C}_0(f) \mathbb{T}(x_2).$$

It is also possible to evaluate the polynomial interpolation formula on a set \mathbf{X} of target points, at the same time. Given the vector X_1 of the first components of a set of target points and the vector X_2 of the corresponding second components, then

$$\mathcal{L}_n f(\mathbf{X}) = \text{diag} \left((\mathbb{T}(X_1))^t \mathbb{C}_0(f) \mathbb{T}(X_2) \right) \quad (4.16)$$

The result $\mathcal{L}_n f(\mathbf{X})$ is a (column) vector containing the evaluation of the interpolation polynomial at the corresponding target points.

If the target points are a Cartesian grid $\mathbf{X} = X_1 \times X_2$, then it is possible to evaluate the polynomial interpolation in a more compact form

$$\mathcal{L}_n f(\mathbf{X}) = \left((\mathbb{T}(X_1))^t \mathbb{C}_0(f) \mathbb{T}(X_2) \right)^t \quad (4.17)$$

The result $\mathcal{L}_n f(\mathbf{X})$ is a matrix whose i -th row and j -th column contains the evaluation of the interpolation polynomial at the point with first component the j -th element in X_1 and second component the i -th element in X_2 .

4.3.2 Cubature

The cubature at the PD points is a nontensorial Clenshaw-Curtis (CC) cubature formula. In order to understand, we recall what is the one-dimensional CC cubature formula.

$$\int_{-1}^1 f(x) dx \underset{x=\cos \theta}{=} \int_0^\pi f(\cos \theta) \sin \theta d\theta,$$

and this can be performed if we compute the cosine series

$$f(\cos \theta) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(k\theta)$$

with

$$a_k = \frac{2}{\pi} \int_0^\pi f(\cos \theta) \cos(k\theta) d\theta. \quad (4.18)$$

Then,

$$\int_0^\pi f(\cos \theta) \sin(\theta) d\theta = a_0 + \sum_{k=1}^{\infty} \frac{2a_{2k}}{1 - (2k)^2}$$

instead of (4.18) we use the Discrete Cosine Transform

$$a_k \approx \frac{2}{N} \left(\frac{f(1)}{2} + \frac{f(-1)}{2} (-1)^k + \sum_{k=1}^{N-1} f \left(\cos \frac{k\pi}{N} \right) \cos \left(\frac{ks\pi}{N} \right) \right), \quad s = 0, \dots, N.$$

```

Input: COF ↔ C0(f)

k = [0:2:n];
mom = 2*sqrt(2)./(1-k.^2);
mom(1) = 2;
[M1,M2] = meshgrid(mom);
M = M1.*M2;
COFM = COF(1:2:n+1,1:2:n+1).*M;
Int = sum(sum(COFM));

```

Output: Int ↔ $I_n(f)$

Table 4.2: Matlab code for the evaluation of the cubature formula by moments.

Formula for the moments

We then can write

$$\begin{aligned}
 \int_{[-1,1]^2} f(x) dx &\approx I_n(f) = \int_{[-1,1]^2} \mathcal{L}_n f(x) dx = \sum_{k=0}^n \sum_{j=0}^k c'_{j,k-j} m_{j,k-j} \\
 &= \sum_{j=0}^n \sum_{l=0}^n c'_{j,l} m_{j,l} = \sum_{j \text{ even}}^n \sum_{l \text{ even}}^n c'_{j,l} m_{j,l}
 \end{aligned} \tag{4.19}$$

where the *moments* $m_{j,l}$ are

$$m_{j,l} = \left(\int_{-1}^1 \hat{T}_j(t) dt \right) \left(\int_{-1}^1 \hat{T}_l(t) dt \right)$$

We have

$$\int_{-1}^1 \hat{T}_j(t) dt = \begin{cases} 2 & j = 0 \\ 0 & j \text{ odd} \\ \frac{2\sqrt{2}}{1-j^2} & j \text{ even} \end{cases}$$

and then, the cubature formula (4.19) can be evaluated by the Matlab code reported in Table 4.2, where we have used the fact that only the (even,even) pairs of indexes are active in the summation process.

Formula for the weights

On the other hand, it is often desirable to have a cubature formula that involves only the function values at the nodes and the corresponding cubature weights. Again, a simple

matrix formulation is available, using the fact that the Padua points are the union of two subgrids of product Chebyshev points. First, observe that

$$I_n(f) = \sum_{j \text{ even}}^n \sum_{l \text{ even}}^n c'_{j,l} m_{j,l} = \sum_{j \text{ even}}^n \sum_{l \text{ even}}^n c_{j,l} m'_{j,l}$$

with

$$\mathbb{M}_0 = (m'_{j,l}) = \begin{pmatrix} m_{0,0} & m_{0,2} & \cdots & \cdots & m_{0,p_n} \\ m_{2,0} & m_{2,2} & \cdots & m_{2,p_n-2} & 0 \\ \vdots & \vdots & & & \vdots \\ m_{p_n-2,0} & m_{p_n-2,2} & 0 & \cdots & 0 \\ m'_{p_n,0} & 0 & \cdots & 0 & 0 \end{pmatrix} \in \mathbb{R}^{(\lfloor \frac{n}{2} \rfloor + 1) \times (\lfloor \frac{n}{2} \rfloor + 1)}$$

where $p_n = n$ and $m'_{p_n,0} = m_{p_n,0}/2$ for n even, $p_n = n - 1$ and $m'_{p_n,0} = m_{p_n,0}$ for n odd.

Now, using the formula for the coefficients (4.10) we can write

$$\begin{aligned} I_n(f) &= \sum_{\xi \in \text{Pad}_n} \lambda_\xi f(\xi) \\ &= \sum_{\xi \in C_{n+1}^E \times C_{n+2}^O} \lambda_\xi f(\xi) + \sum_{\xi \in C_{n+1}^O \times C_{n+2}^E} \lambda_\xi f(\xi) \end{aligned}$$

where

$$\lambda_\xi = w_\xi \sum_{j \text{ even}}^n \sum_{l \text{ even}}^n m'_{j,l} \hat{T}_j(\xi_1) \hat{T}_l(\xi_2) \quad (4.20)$$

Defining the Chebyshev matrix corresponding to even degrees (cf. (4.11))

$$\mathbb{T}^E(S) = \begin{pmatrix} \hat{T}_0(s_1) & \cdots & \hat{T}_0(s_m) \\ \hat{T}_2(s_1) & \cdots & \hat{T}_2(s_m) \\ \vdots & \cdots & \vdots \\ \hat{T}_{p_n}(s_1) & \cdots & \hat{T}_{p_n}(s_m) \end{pmatrix} \in \mathbb{R}^{(\lfloor \frac{n}{2} \rfloor + 1) \times m}$$

and the matrices of interpolation weights on the subgrids of Padua points,

$$\mathbb{W}_1 = (w_\xi, \xi \in C_{n+1}^E \times C_{n+2}^O)^t,$$

$$\mathbb{W}_2 = (w_\xi, \xi \in C_{n+1}^O \times C_{n+2}^E)^t,$$

it is then easy to show that the cubature weights $\{\lambda_\xi\}$ can be computed in the matrix form

$$\mathbb{L}_1 = (\lambda_\xi, \xi \in C_{n+1}^E \times C_{n+2}^O)^t = \mathbb{W}_1 \cdot (\mathbb{T}^E(C_{n+1}^E))^t \mathbb{M}_0 \mathbb{T}^E(C_{n+2}^O)^t$$

```

Input: W1 ↔ (wξ, ξ ∈ Cn+1E × Cn+2O)t, W2 ↔ (wξ, ξ ∈ Cn+1O × Cn+2E)t

argn1 = linspace(0,pi,n+1);
argn2 = linspace(0,pi,n+2);
k = [0:2:n]';
l = (n-mod(n,2))/2+1;
TE1 = cos(k*argn1(1:2:n+1));
TE1(2:1,:) = TE1(2:1,:)*sqrt(2);
T01 = cos(k*argn1(2:2:n+1));
T01(2:1,:) = T01(2:1,:)*sqrt(2);
TE2 = cos(k*argn2(1:2:n+2));
TE2(2:1,:) = TE2(2:1,:)*sqrt(2);
T02 = cos(k*argn2(2:2:n+2));
T02(2:1,:) = T02(2:1,:)*sqrt(2);
mom = 2*sqrt(2)./(1-k.^2);
mom(1) = 2;
[M1,M2] = meshgrid(mom);
M = M1.*M2;
M0 = fliplr(triu(fliplr(M)));
if (mod(n,2) == 0)
    M0(n/2+1,1) = M0(n/2+1,1)/2;
end
L1 = W1.*(TE1'*M0*T02)';
L2 = W2.*(T01'*M0*TE2)';

```

Output: L1 ↔ (λ_ξ, ξ ∈ C_{n+1}^E × C_{n+2}^O)^t, L2 ↔ (λ_ξ, ξ ∈ C_{n+1}^O × C_{n+2}^E)^t

Table 4.3: Matlab code for the computation of the cubature weights.

$$\mathbb{L}_2 = (\lambda_\xi, \xi \in C_{n+1}^O \times C_{n+2}^E)^t = \mathbb{W}_2 \cdot (\mathbb{T}^E(C_{n+1}^O))^t \mathbb{M}_0 \mathbb{T}^E(C_{n+2}^E)^t$$

where the dot means that the final product is made componentwise. The corresponding Matlab code is reported in Table 4.3. The definition of the weights matrices \mathbb{L}_i , $i = 1, 2$, makes use of transposes in order to be compatible with the Matlab meshgrid-like structure of the matrices \mathbb{W}_i (see also (4.17)).

Remark. We can use a FFT-based implementation in analogy with the univariate CC-quadrature.

It is worth recalling that the cubature weights are not all positive, but the negative ones are few and of small size. Indeed, the cubature formula is stable and convergent for every continuous integrand, since

$$\lim_{n \rightarrow \infty} \sum_{\xi \in \text{Pad}_n} |\lambda_\xi| = 4$$

as it has been proved in [44].

4.3.3 Some numerical experiments

Here we present the CPU times for the computation of the interpolation coefficients and the cubature weights. We did these experiments on a Laptop with Intel Core2 Duo 2.2GHz processor. The code we have run is the Matlab implementation. We note that using Octave 3.2.3 (the available version at that time, we obtained similar results.

n	20	40	60	80	100	300	500	1000
FFT	0.001	0.001	0.001	0.002	0.003	0.034	0.115	0.387
MM	0.002	0.003	0.003	0.003	0.008	0.101	0.298	1.353

Table 4.4: CPU time (in seconds) for the computation of the interpolation coefficients at a sequence of degrees (average of 10 runs).

n	20	40	60	80	100	300	500	1000
FFT	0.001	0.001	0.002	0.002	0.004	0.028	0.111	0.389
MM	0.001	0.001	0.001	0.002	0.003	0.027	0.092	0.554

Table 4.5: CPU time (in seconds) for the computation of the cubature weights at a sequence of degrees (average of 10 runs).

Exercise 6. We invite the readers to write a code that implement the interpolant and the cubature at the Padua points. Then, try to reproduce similar CPU results as obtained in Tables 4.4 and 4.5, as well as the relative cubature errors displayed in Figure 4.2.

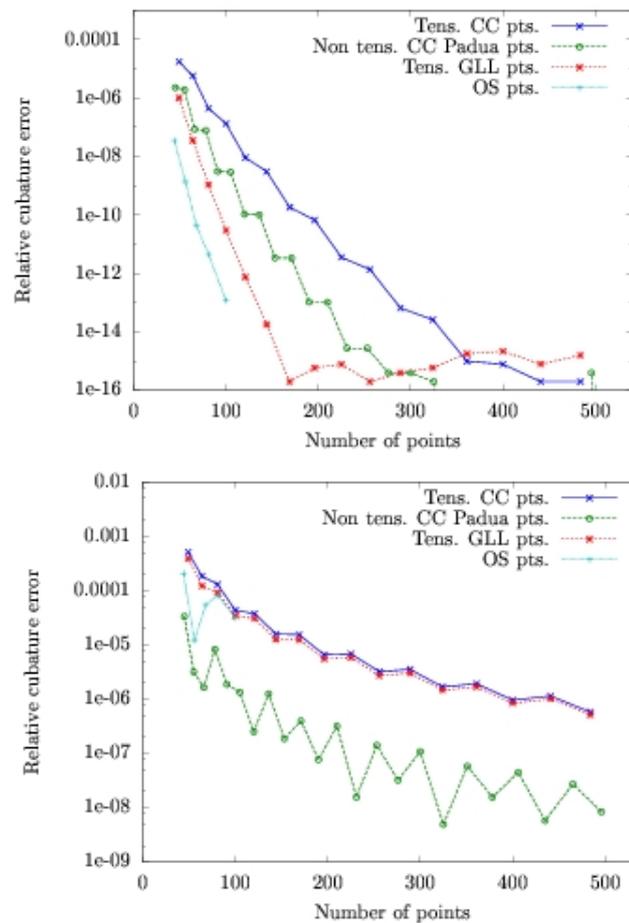


Figure 4.2: Relative cubature errors versus the number of cubature points (CC = Clenshaw-Curtis, GLL = Gauss-Legendre-Lobatto, OS = Omelyan-Solovyan) for the Gaussian $f(\mathbf{x}) = \exp(-|\mathbf{x}|^2)$ (above) and the C^2 function $f(\mathbf{x}) = |\mathbf{x}|^3$ (below); the integration domain is $[-1, 1]^2$, the integrals up to machine precision are, respectively: 2.230985141404135 and 2.508723139534059.

Lecture 5

Weakly Admissible Meshes

In this last lecture we discuss another very important problem connected to polynomial interpolation on subsets $K \subseteq \mathbb{R}^d$, $d > 1$.

We start by recalling some properties of Fekete points that will be one of the reason why we are interested to the construction of suitable discretizations of K . Fekete points have been studied so far mainly in the univariate (complex) case. The only instance where they are analytically known are the following ones-

- On the interval $[-1, 1]$ they are the $n + 1$ **Gauss-Lobatto** points (indeed they are the zeros of $(1 - x^2)P'_n(x)$ with P_n the Legendre polynomial of degree n). Moreover, $\Lambda_n = \mathcal{O}(\log n)$
- On the complex circle they are $2n + 1$ equispaced points and $\Lambda_n = \mathcal{O}(\log n)$.
- On the cube $[-1, 1]^d$ they coincide with the tensor product of Gauss-Lobatto points with $\Lambda_n = \mathcal{O}(\log^d n)$.

In general, as we already observed in §1.1.6, the computation of Fekete points becomes rapidly a very large scale problem namely a *nonlinear optimization problem* in $N \times d$ variables ($N = \dim(\mathbb{P}_n^d(K))$). From the computational point of view, recently there were done some hard computation obtaining true Fekete points for small n . Two important examples are

- the triangle where they have been computed for $n \leq 19$ (cf. [46]);
- the sphere where they have been computed for $n \leq 191$ (cf. [45]).

From these examples, it comes clear that we can eventually solve the problem by a suitable discretization of the domain (on which we want to solve the optimization problem).

The question to which we try to answer is this chapter is

Which is the "best" discretization of K from which we can extract good interpolation points?

5.1 Weakly Admissible Meshes

This section is partially taken from [10].

Given a *polynomial determining* compact set $K \subset \mathbb{R}^d$ or $K \subset \mathbb{C}^d$ (i.e., polynomials vanishing there are identically zero), a Weakly Admissible Mesh (WAM) is defined in [15] to be a sequence of discrete subsets $\mathcal{A}_n \subset K$ such that

$$\|p\|_K \leq C(\mathcal{A}_n) \|p\|_{\mathcal{A}_n}, \quad \forall p \in \mathbb{P}_n^d(K) \quad (5.1)$$

where both $\text{card}(\mathcal{A}_n) \geq N$ and $C(\mathcal{A}_n)$ grow at most *polynomially* with n . When $C(\mathcal{A}_n)$ is bounded we speak of an Admissible Mesh (AM). Here and below, we use the notation $\|f\|_X = \sup_{x \in X} |f(x)|$, where f is a bounded function on the compact X .

We sketch below the main features of WAMs in terms of ten properties (cf. [7, 15]):

- P1:** $C(\mathcal{A}_n)$ is invariant under affine mapping
- P2:** any sequence of unisolvent interpolation sets whose Lebesgue constant grows at most polynomially with n is a WAM, $C(\mathcal{A}_n)$ being the Lebesgue constant itself
- P3:** any sequence of supersets of a WAM whose cardinalities grow polynomially with n is a WAM with the same constant $C(\mathcal{A}_n)$
- P4:** a finite union of WAMs is a WAM for the corresponding union of compacts, $C(\mathcal{A}_n)$ being the maximum of the corresponding constants
- P5:** a finite cartesian product of WAMs is a WAM for the corresponding product of compacts, $C(\mathcal{A}_n)$ being the product of the corresponding constants
- P6:** in \mathbb{C}^d a WAM of the boundary ∂K is a WAM of K (by the maximum principle)
- P7:** given a polynomial mapping π_s of degree s , then $\pi_s(\mathcal{A}_{ns})$ is a WAM for $\pi_s(K)$ with constants $C(\mathcal{A}_{ns})$ (cf. [7, Prop.2])
- P8:** any K satisfying a Markov polynomial inequality like $\|\nabla p\|_K \leq Mn^r \|p\|_K$ has an AM with $\mathcal{O}(n^{rd})$ points (cf. [15, Thm.5])
- P9:** least-squares polynomial approximation of $f \in C(K)$: the least-squares polynomial $\mathcal{L}_{\mathcal{A}_n} f$ on a WAM is such that

$$\|f - \mathcal{L}_{\mathcal{A}_n} f\|_K \lesssim C(\mathcal{A}_n) \sqrt{\text{card}(\mathcal{A}_n)} \min \{\|f - p\|_K, p \in \mathbb{P}_n^d(K)\}$$

(cf. [15, Thm.1])

- P10:** Fekete points: the Lebesgue constant of Fekete points extracted from a WAM can be bounded like $\Lambda_n \leq NC(\mathcal{A}_n)$ (that is the elementary classical bound of the continuum Fekete points times a factor $C(\mathcal{A}_n)$); moreover, their asymptotic distribution is the same of the continuum Fekete points, in the sense that the corresponding discrete probability measures converge weak-* to the pluripotential equilibrium measure of K (cf. [7, Thm.1])

The properties above give the basic tools for the construction of WAMs.

We focus to the bivariate case.

Property **P8**, applied for example to **convex compacts** where a Markov inequality with exponent $r = 2$ always holds, says that it is possible to obtain an **Admissible Mesh** with $\mathcal{O}(n^4)$ points, which is too big for practical uses!

In order to avoid such a large cardinality, we use WAMs, which can have cardinality typically of $\mathcal{O}(n^2)$ points.

Example 9. On $[-1, 1]^2$ there are known WAMs with $\mathcal{O}(n^2)$ points. These are the **Padua points** with cardinality $\mathcal{O}(n^2/2)$, **Xu points** with cardinality $\mathcal{O}(n^2/2)$ and **tensor-product points** with cardinality $\mathcal{O}(n^2)$.

In [7] a WAM on the disk with approximately $2n^2$ points and $C(\mathcal{A}_n) = \mathcal{O}(\log^2 n)$ has been constructed with standard polar coordinates, using essentially property **P2** for univariate Chebyshev and trigonometric interpolation. Moreover, using property **P2** and **P7**, WAMs for the triangle and for linear trapezoids, again with approximately $2n^2$ points and $C(\mathcal{A}_n) = \mathcal{O}(\log^2 n)$, have been obtained simply by mapping the so-called Padua points of degree $2n$ from the square with standard quadratic transformations (the first known optimal points for bivariate polynomial interpolation, with a Lebesgue constant growing like log-squared of the degree, cf. [9]).

In [13] these results have been improved, showing that there are WAMs for the disk and the triangle with approximately n^2 points and still the same growth of the relevant constants. In particular, a symmetric polar WAM of the unit disk is made by equally spaced angles and Chebyshev-Lobatto points on the corresponding diameters

$$\mathcal{A}_n = \{(r_j \cos \theta_k, r_j \sin \theta_k)\}$$

$$\{(r_j, \theta_k)\}_{j,k} = \left\{ \cos \frac{j\pi}{n}, 0 \leq j \leq n \right\} \times \left\{ \frac{k\pi}{m}, 0 \leq k \leq m-1 \right\}$$

where $m = n + 2$ for even n and $m = n + 1$ for odd n (see Figure 5.1).

Exercise 7. Construct for various n the WAMs displayed in Figures 5.1 and the transformed ones in the quadrant and simplex as in Figures 5.2.

5.2 Approximate Fekete Points and Discrete Leja Points

We need a few notations. Given a compact set $K \subset \mathbb{R}^d$ (or \mathbb{C}^d), a finite-dimensional space of linearly independent continuous functions, $S_N = \text{span}(p_j)_{1 \leq j \leq N}$, and a finite set $\{\xi_1, \dots, \xi_N\} \subset K$, ordering in some manner the points and the basis we can construct the Vandermonde-like matrix $V(\xi; \mathbf{p}) = [p_j(\xi_i)]$, $1 \leq i, j \leq N$. If $\det V(\xi; \mathbf{p}) \neq 0$ the set

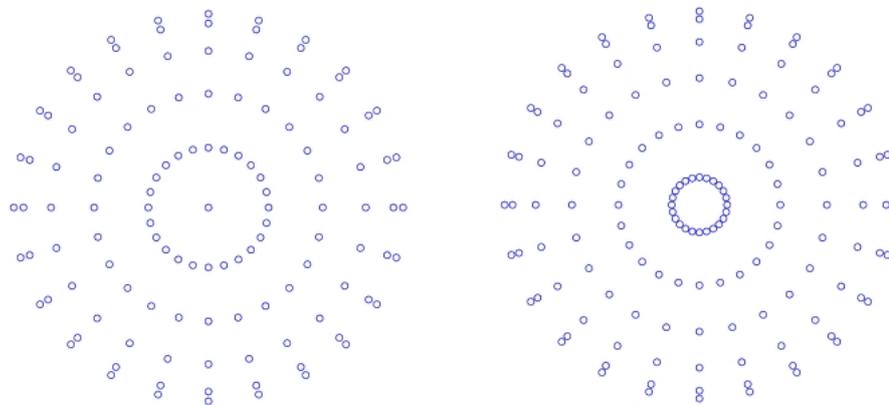


Figure 5.1: Symmetric polar WAMs of the disk for degree $n = 10$ (left) and $n = 11$ (right).

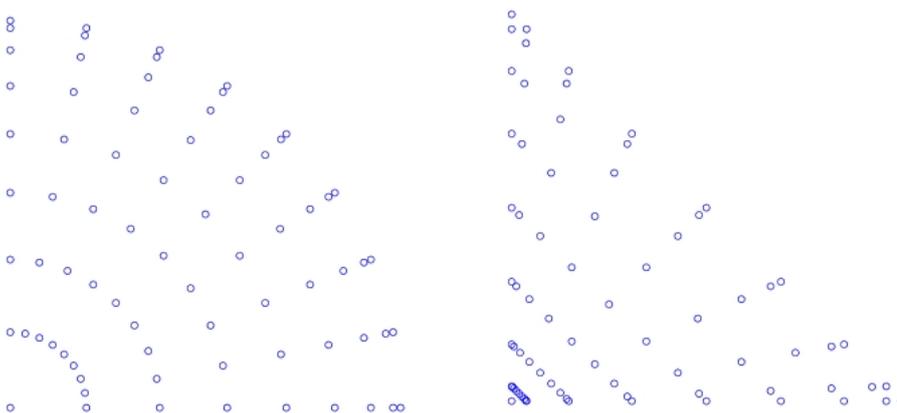


Figure 5.2: A WAM of the quadrant for even polynomials of degree $n = 16$ (left), and the corresponding WAM of the simplex for degree $n = 8$ (right).

$\{\xi_1, \dots, \xi_N\}$ is unisolvent for interpolation in S_N , and

$$\ell_j(x) = \frac{\det V(\xi_1, \dots, \xi_{j-1}, x, \xi_{j+1}, \dots, \xi_N; \mathbf{p})}{\det V(\xi_1, \dots, \xi_{j-1}, \xi_j, \xi_{j+1}, \dots, \xi_N; \mathbf{p})}, \quad j = 1, \dots, N, \quad (5.2)$$

is a cardinal basis, i.e. $\ell_j(\xi_k) = \delta_{jk}$ and $L_{S_N} f(x) = \sum_{j=1}^N f(\xi_j) \ell_j(x)$ interpolates any function f at $\{\xi_1, \dots, \xi_N\}$. In matrix terms, the cardinal basis $\ell = (\ell_1, \dots, \ell_N)^t$ is obtained from the original basis $\mathbf{p} = (p_1, \dots, p_N)^t$ as $\ell = L\mathbf{p}$, $L := (V(\xi; \mathbf{p}))^{-t}$.

A reasonable approach for the computation of Fekete points is to use a discretization of the domain.

Property **P10** gives a first guideline on the fact that WAMs are good candidates as starting meshes.

Since the rows of the Vandermonde matrix $V = V(\mathbf{a}; \mathbf{p})$, correspond to the mesh points and the columns to the basis elements, then

computing the Fekete points of a WAM amounts to selecting N rows of V such that the absolute value of the determinant of the resulting $N \times N$ submatrix (its volume!), is maximum.

An approximate solution can be given by one of the following two *greedy* algorithms, given as pseudo-codes in a Matlab-like notation, which compute what we call “Discrete Extremal Sets”; cf. [7].

algorithm greedy 1 (Approximate Fekete Points):

- $V = V(\mathbf{a}, \mathbf{p}); \text{ind} = [];$
- for $k = 1 : N$ “select i_k : $\text{vol } V([\text{ind}, i_k], 1 : N)$ is maximum”; $\text{ind} = [\text{ind}, i_k];$ end
- $\xi = \mathbf{a}(i_1, \dots, i_N)$

algorithm greedy 2 (Discrete Leja Points):

- $V = V(\mathbf{a}, \mathbf{p}); \text{ind} = [];$
- for $k = 1 : N$ “select i_k : $|\det V([\text{ind}, i_k], 1 : k)|$ is maximum”; $\text{ind} = [\text{ind}, i_k];$ end
- $\xi = \mathbf{a}(i_1, \dots, i_N)$

These algorithms are genuinely different. In both, the selected points (as opposed to for the continuum Fekete points) depend on the choice of the polynomial basis. But in the second algorithm, which is based on the notion of determinant, the selected points depend also on the ordering of the basis. In the univariate case with the standard monomial basis, it is not difficult to recognize that the selected points are indeed the Leja points extracted from the mesh.

The two greedy algorithms above correspond to basic procedures of numerical linear algebra. Indeed, in algorithm 1 the core “select i_k : $\text{vol } V([\text{ind}, i_k], 1 : N)$ is maximum” can be implemented as “select the largest norm row $\text{row}_{i_k}(V)$ and remove from every row of V its orthogonal projection onto row_{i_k} ”, since the corresponding orthogonalization process does not affect volumes (as can be understood geometrically applying the method to a collection of 3-dimensional vectors and thinking in terms of parallelograms and parallelepipeds). Working for convenience with the transposed Vandermonde matrix, this process is equivalent to the QR factorization with column pivoting implemented in Matlab by the standard “backslash” operator (cf. [8] for a full discussion of the equivalence).

On the other hand, it is clear that in algorithm 2 the core “**select** i_k : $|\det V([ind, i_k], 1:k)|$ **is maximum**” can be implemented by one column elimination step of the **Gaussian elimination** process with standard **row pivoting**, since such process automatically seeks the maximum keeping invariant the absolute value of the relevant subdeterminants.

This is summarized in the following Matlab-like scripts:

algorithm AFP (Approximate Fekete Points):

- $W = (V(\mathbf{a}, \mathbf{p}))^t$; $\mathbf{b} = (1, \dots, 1)^t \in \mathbb{C}^N$; $\mathbf{w} = W \setminus \mathbf{b}$; $ind = \text{find}(\mathbf{w} \neq \mathbf{0})$; $\xi = \mathbf{a}(ind)$

algorithm DLP (Discrete Leja Points):

- $V = V(\mathbf{a}, \mathbf{p})$; $[L, U, \sigma] = \text{LU}(V, \text{“vector”})$; $ind = \sigma(1, \dots, N)$; $\xi = \mathbf{a}(ind)$

where σ is the permutation vector.

Example 10. Hexagon. To give an example of computation of Discrete Extremal Sets, we consider the nonregular convex hexagon in Figure 5.3, with the WAMs generated by two different triangulations, and the Chebyshev product basis of the minimal including rectangle. In Figure 5.3 on the left the hexagon is split by using the so-called *barycenter splitting*, which has k sides and k triangles. On the right we used the so-called *ear-clipping splitting* that gives k sides and $k - 2$ triangles. From Table 5.1 we see that, concerning Lebesgue constants, DLP are of lower quality than AFP: this is not surprising, since the same phenomenon is well-known concerning continuous Fekete and Leja points. Nevertheless, both provide reasonably good interpolation points, as it is seen from the interpolation errors on two test functions of different regularity in Table 5.2.

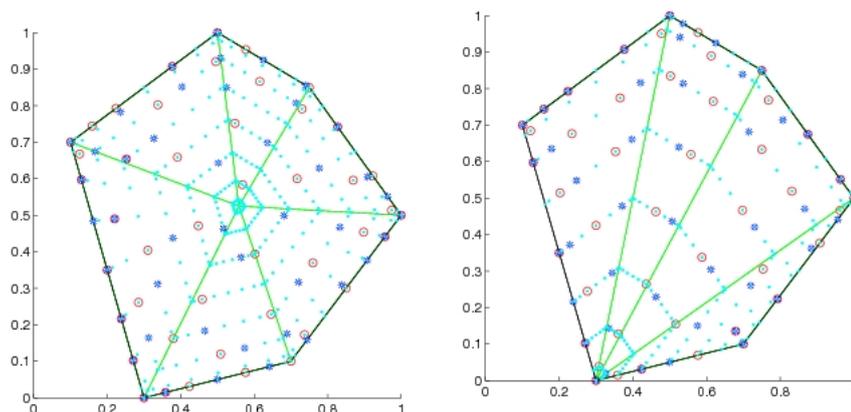


Figure 5.3: $N = 45$ Approximate Fekete Points (circles) and Discrete Leja Points (asterisks) for degree $n = 8$ extracted from two WAMs of a nonregular convex hexagon (dots).

Example 11. Circular sector. We consider the $3/4$ of the unit disk, that is

$$K = \left\{ (\rho, \theta) : 0 \leq \rho \leq 1, -\frac{\pi}{2} \leq \theta \leq \pi \right\}.$$

mesh	points	$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$
WAM1	AFP	6.5	18.9	20.4	40.8	73.3	73.0
	DLP	7.1	19.6	49.8	58.3	108.0	167.0
WAM2	AFP	6.8	12.3	34.2	52.3	49.0	80.4
	DLP	10.7	48.4	62.0	91.6	86.6	203.0

Table 5.1: Lebesgue constants for AFP and DLP extracted from two WAMs of a nonregular convex hexagon (WAM1: barycentric triangulation, WAM2: minimal triangulation; see Fig. 5.3).

function	points	$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$
f_1	AFP	6E-06	5E-13	3E-15	3E-15	3E-15	4E-15
	DLP	8E-06	2E-12	2E-15	4E-15	3E-15	4E-15
f_2	AFP	3E-03	2E-04	1E-04	4E-05	2E-05	1E-05
	DLP	3E-03	3E-04	1E-04	3E-05	2E-05	5E-06

Table 5.2: Max-norm of the interpolation errors with AFP and DLP extracted from WAM2 for two test functions: $f_1 = \cos(x_1 + x_2)$; $f_2 = ((x_1 - 0.5)^2 + (x_2 - 0.5)^2)^{3/2}$.

For this compact set we wish to construct an **admissible mesh**, AM, for extracting AFP and DLP. By property **P4**, the set K is the union of 3 quadrants, which are convex components. Therefore the problem is:

How to construct an AM on a convex compact $K \subset \mathbb{R}^2$?

Two facts.

1. A well-known fact is that every convex compact of \mathbb{R}^2 admits the *Markov inequality* of order 2

$$\max_{x \in K} \|\nabla p(x)\|_2 \leq M n^2 \|p\|_K, \quad M = \frac{\alpha(K)}{w(K)}, \quad \forall p \in \mathbb{P}_n^2(K), \quad (5.3)$$

where $\alpha(K) \leq 4$ and $w(K)$ is the minimal distance between two parallel support lines for K (see [11]).

2. Take the cartesian grid $\{(ih, jh), i, j \in \mathbb{Z}\}$ (h stepsize). Then follows these steps
 - On every small square of the grid with nonempty intersection with K take a point on this intersection
 - Let \mathcal{A}_n the mesh formed by these points
 - For all $x \in K$, let $a \in \mathcal{A}_n$ the point closest to x (that by construction both belong to the same square)

- Using the mean-value theorem, Cauchy-Schwarz inequality and Markov inequality (5.3) we get

$$|p(x) - p(a)| \leq \underbrace{\|\nabla p(y)\|_2}_{\leq (5.3)} \underbrace{\|x - a\|}_{\leq \sqrt{2}h} \leq M\sqrt{2}hn^2 \|p\|_K$$

because y belongs to the open segment connecting x and a , lying on K .

Since

$$|p(x)| \leq |p(x) - p(a)| + |p(a)| \leq M\sqrt{2}hn^2 \|p\|_K + |p(a)|$$

then

$$\|p\|_K \leq \frac{1}{1 - \mu} \|p\|_{\mathcal{A}_n}$$

provided $h = h_n$ s.t.

$$M\sqrt{2}h_n n^2 \leq \mu < 1. \quad (5.4)$$

Hence, \mathcal{A}_n is an AM with constant $C = 1/(1 - \mu)$.

In the case of the *first quadrant*, $w(K) = 1$ and we take the upper bound for convex compact $\alpha(K) = 4$ (since sharper bounds do not seem to exist). Hence by (5.3) and (5.4), an AM exists as soon as we consider $\{(ih_n, jh_n), i, j \in \mathbb{Z}\}$ with

$$4\sqrt{2}h_n n^2 < 5.66n^2 h_n \leq \mu < 1$$

for some fixed μ . For example, taking $h_n = 1/(6n^2)$ we get $\mu = 5.66/6 < 1$ and $C \approx 17.65$. Since we can partition the set of grid squares into subsets of four adjacent squares and, apart from a neighborhood of the boundary of the quadrant, take as mesh point their common vertex, then the cardinality of the mesh is roughly estimated as 1/4 of the number of grid points in the unit square times the area of the quadrant. Then the cardinality of \mathcal{A}_n in the first quadrant is

$$\text{card}(\mathcal{A}_n) \approx \frac{(6n^2)^2 \pi}{4} \approx 7n^4$$

and therefore for the whole sector (3/4 of the disk) $\text{card}(\mathcal{A}_n) \approx 21n^4$.

As a remark, we notice that for computing the Vandermonde matrix, we have used the *Koornwinder orthogonal basis of the unit disk* called also *ball polynomials* (cf. [30]) associated to the weight function of two variables $\omega^{(\alpha)}(x, y) = (1 - x^2 - y^2)^\alpha$, $\alpha > -1$ on the ball, defined as

$$P_{n-k,k}^{(\alpha)}(x, y) = P_{n-k}^{(\alpha+k+\frac{1}{2}, \alpha+k+\frac{1}{2})}(x, y) (1-x^2)^{\frac{k}{2}} P_k^{\alpha, \alpha} \left(\frac{y}{\sqrt{1-x^2}} \right), \quad n \geq k \neq 0, \quad (5.5)$$

where as usual $P_k^{\alpha, \beta}(x)$ indicates the Jacobi polynomials.

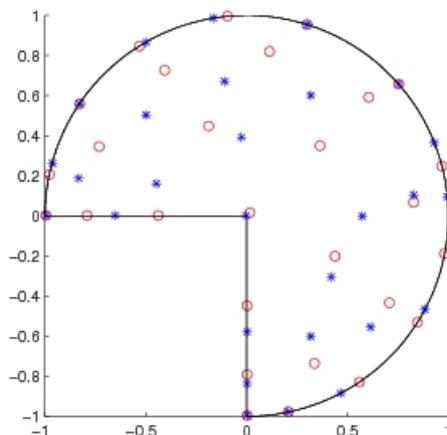


Figure 5.4: $N = 28$ AFP (Approximate Fekete Points, circles) and DLP (Discrete Leja Points, asterisks) for degree $n = 6$ extracted from an AM (Admissible Mesh) of a circular sector.

Polygons. The recent paper [13] gives a construction of a WAM for the two-dimensional unit simplex, and thus for any triangle by affine mapping (Property P1 of WAMs). This WAM, say \mathcal{A}_n , has $n^2 + n + 1$ points for degree n , and constant $C(\mathcal{A}_n) = \mathcal{O}(\log^2 n)$. The mesh points lie on a grid of intersecting straight lines, namely a pencil from one vertex (image of the point $(0, 0)$ of the simplex) cut by a pencil parallel to the opposite side (image of the hypotenuse of the simplex). The points on each segment of the pencils, and in particular the points on each side, are the corresponding Chebyshev-Lobatto points.

Property **P4** allows then to obtain WAMs for any triangulated polygon. The constant of any such WAM can be bounded by the maximum of the constants corresponding to the triangular elements, and thus is $\mathcal{O}(\log^2 n)$, irrespectively of the number of sides of the polygon, or of the fact that it is convex or concave. Notice that a rough triangulation is better in the present framework, to keep the cardinality of the mesh low (which will be of the order of n^2 times the number of triangles).

Example 12. As a first example, we have already seen a nonregular convex hexagon (cf. Figure 5.3), either trivially triangulated by the barycenter, or by the so-called “ear-clipping” algorithm. The latter constructs a minimal triangulation of any *simple* polygon with k vertices, obtaining $k - 2$ triangles. Referring to Figure 5.3, in the first mesh the point $(0, 0)$ of the simplex is mapped to the barycenter for each triangle. The cardinality of the barycentric-based mesh is $6(n^2 + n + 1) - 6(n + 1) - 5 = 6n^2 - 5$, whereas that of the other mesh is $4(n^2 + n + 1) - 3(n + 1) - 2 = 4n^2 + n - 1$ (one has to subtract the repetitions of points along the contact sides).

Example 13. In order to emphasize the flexibility of the Algorithms AFP and DLP, as another example we show the points computed for a polygon in a shape of a hand, with 39 sides (obtained from the screen sampled hand of one of the authors, by piecewise linear

interpolation); see Figure 5.5. In this example we have used the ear-clipping triangulation which gives 37 triangles and a WAM with approximately $37n^2$ points for degree n .

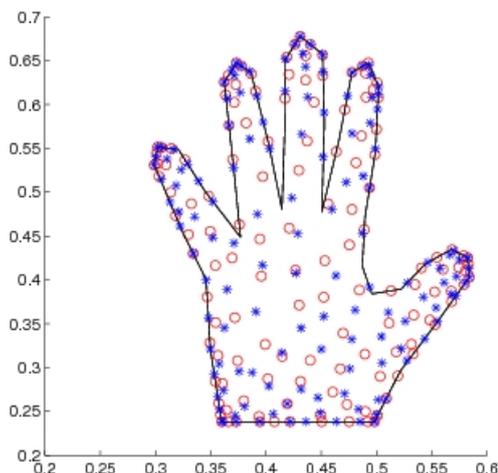


Figure 5.5: $N = 136$ AFP (circles) and DLP (asterisks) for degree $n = 15$ in a hand-shape polygon.

5.2.1 Extensions and some applications.

We have already constructed WAM on three dimensional domains, like pyramids, cones and solid of rotations in [26].

Polynomial interpolation and approximation are at the core of many important numerical techniques, and we are of the opinion that Weakly Admissible Meshes and Discrete Extremal Sets could give new useful tools in several applications. We mention here three of them.

As a first natural application, we can consider *numerical cubature*. In fact, if in algorithm AFP we take as right-hand side $\mathbf{b} = \mathbf{m} = \int_K \mathbf{p}(x) d\mu$ (the *moments* of the polynomials basis with respect to a given measure), the vector $\mathbf{w}(ind)$ gives directly the weights of an algebraic cubature formula at the corresponding Approximate Fekete Points. The same can clearly be done with Discrete Leja Points, solving the system $V(\xi; \mathbf{p})\mathbf{w} = \mathbf{m}$ by the LU factorization, with the additional feature that we get a nested family of cubature formulas (Leja points being a sequence). This approach has recently applied to the approximation of integrals in the quasi Monte Carlo method (cf. [3]).

The possibility of locating good points for polynomial approximation on polygonal regions/elements, could also be useful in the numerical solution of PDEs by *spectral and high-order methods*, for example in the emerging field of discontinuous Galerkin methods (see, e.g., [31])

Appendix A

The Stone-Weierstrass theorem

This short appendix provides the extension of the *Weierstrass theorem* (valid for functions of one real variable).

Let $K \subset \mathbb{R}^d$ be compact and $\mathcal{C}(K)$ be the set of continuous functions on K equipped with the sup-norm $\|f\|_K = \sup_{x \in K} |f(x)|$. Take $A \subset \mathcal{C}(K)$. When is A dense in $\mathcal{C}(K)$? In other words, when $\bar{A} = \mathcal{C}(K)$?

The Stone-Weierstrass theorem provides *sufficient* conditions for A to be dense in $\mathcal{C}(K)$.

Theorem 19. *A is dense in $\mathcal{C}(K)$ (i.e. $\bar{A} = \mathcal{C}(K)$) if the following conditions holds*

1. *A is an algebra. That is for any $u, v \in A$ and any $a, b \in \mathbb{R}$, $au + bv \in A$ and $uv \in A$ (A is close by linear combinations and multiplication).*
2. *The function $c = 1$ belongs to A .*
3. *A separates points. That is for any couple of distinct points $x, y \in K$ there exists $f \in A$ s.t. $f(x) \neq f(y)$.*

The multivariate equivalent to the polynomials $\mathfrak{P}_m(\mathbb{R})$ is the set $\mathfrak{P}_m(\mathbb{R}^d)$, i.e. the set of polynomials of total degree less and equal to m in d variables. This multivariate set is generated by the monomials $\text{span}\{x^{\mathbf{k}}, x \in \mathbb{R}^d, \mathbf{k} = (k_1, \dots, k_d), k_j \in \mathbb{N}, |\mathbf{k}| = k_1 + k_2 + \dots + k_d = m\}$.

We want to underline some facts about the sufficient conditions of the previous theorem.

- About condition 2. Take $K = [0, 1]$ and the space $P_0 = \text{span}\{x, x^2, \dots, x^n\}$. It is clear that for all $p \in P_0$, $p(0) = 0$. Therefore if $f \in \mathcal{C}[0, 1]$ is such that $f(0) \neq 0$, then it can not be recovered by elements of $P = 0$. This shows also that condition 1. is necessary.
- About condition 3. It is also necessary. In fact consider the set of trigonometric polynomials \mathbb{T}_n , it is an algebra that do not separate the points $-\pi$ and π .

- We want to show that condition 1. is not necessary. In fact there exist sets of polynomials that are not an algebra.
 1. *Lacunary polynomials.* It is the set $M = \text{span}\{x^{n_k}, n_0 = 0 < n_1 < n_2 < \dots, \lim_{k \rightarrow \infty} n_k = \infty\}$. The set M satisfies properties 2. and 3. but it is not closed by multiplication. In order to be dense, it has to satisfy the Muntz's theorem (cf. e.g. [1]) that is $\sum_{k=0}^{\infty} \frac{1}{n_k} = \infty$.
 2. *Incomplete polynomials.* The set $M^* = \{\sum_{n \leq k \leq 2n} c_k x^k, \forall n\} \oplus 1$ is not dense in $\mathcal{C}[0, 1]$. The algebra property of closeness by multiplication is satisfied, but not that of addition since M^* is not a linear space. But it has been proved that $\bar{M}^* = \mathcal{C}[\frac{1}{4}, 1]$.

Bibliography

- [1] J. M. Almira, *Muntz Type Theorems I*, <https://arxiv.org/pdf/0710.3570.pdf>.
- [2] J. Baglama, D. Calvetti and L. Reichel, *Fast Leja points*, *Electron. Trans. Numer. Anal.* **7** (1998), 124–140.
- [3] C. Bittante, S. De Marchi and A. Sommariva, *A new quasi-Monte Carlo technique based on nonnegative least-squares and approximate Fekete points*, submitted 2014.
- [4] L. Bos, *On certain configurations of points in \mathbb{R}^n which are unisolvent for polynomial interpolation*, *J. Approx. Theory*, **64**(3) (1991), 271–280.
- [5] L. Bos, *Multivariate interpolation and polynomial inequalities*, Ph.D. course held at the University of Padua (2001), unpublished.
- [6] L. Bos, M. Caliari, S. De Marchi, M. Vianello, *A numerical study of the Xu interpolation formula*, *Computing* **76**(3-4) (2006), 311–324.
- [7] L. Bos, J.-P. Calvi, N. Levenberg, A. Sommariva and M. Vianello, *Geometric Weakly Admissible Meshes, Discrete Least Squares Approximation and Approximate Fekete Points*, *Math. Comp.* **80** (2011), 1601–1621.
- [8] L. Bos and N. Levenberg, *On the Approximate Calculation of Fekete Points: the Univariate Case*, *Elect. Trans. Numer. Anal.* **30** (2008), 377–397.
- [9] L. Bos, M. Caliari, S. De Marchi, M. Vianello and Y. Xu: *Bivariate Lagrange interpolation at the Padua points: the generating curve approach*, *J. Approx. Theory* **143** (2006), 15–25.
- [10] L. Bos, S. De Marchi, A. Sommariva and M. Vianello: *Weakly Admissible Meshes and Discrete Extremal Sets*, *Numer. Math. Theory Methods Appl.* **4** (2011), 1–12.
- [11] L. Bos, S. De Marchi, A. Sommariva and M. Vianello: *Computing multivariate Fekete and Leja points by numerical linear algebra*, *SIAM J. Num. Anal.* **48**(5) (2010), 1984–1999.
- [12] L. Bos, M. A. Taylor and B. A. Wingate, *Tensor product Gauss-Lobatto points are Fekete points for the cube*, *Math. Comp.* **70** (2001), 1543–1547.

- [13] L. Bos, A. Sommariva and M. Vianello, Least-squares polynomial approximation on weakly admissible meshes: disk and triangle, *J. Comput. Appl. Math.* 235 (2010), 660–668.
- [14] L. Brutman, *Lebesgue functions for polynomial interpolation: a survey*, *Ann. Numer. Math.* 4 (1997) 111–127.
- [15] J.P. Calvi and N. Levenberg, Uniform approximation by discrete least squares polynomials, *J. Approx. Theory* 152 (2008), 82–100.
- [16] M. Caliari, S. De Marchi and M. Vianello: Bivariate polynomial interpolation on the square at new nodal sets, *Appl. Math. Comput.* 165(2) (2005), 261–274
- [17] Caliari, M., De Marchi, S., Vianello, M.: Algorithm 886: Padua2D: Lagrange Interpolation at Padua Points on Bivariate Domains. *ACM Trans. Math. Software* 35-3 (2008).
- [18] M. Caliari, S. De Marchi, A. Sommariva and M. Vianello: Padua2DM: fast interpolation and cubature at the Padua points in Matlab/Octave, *Numer. Algorithms* 56 (2011), 45–60.
- [19] Numerical computing with functions: Chebfun. www.chebfun.org
- [20] Chung K. C. and Yao, T. H.: On lattices admitting unique Lagrange interpolations, *SIAM J. Numer. Anal.* 14 (1977), 735–743.
- [21] C. de Boor, *A Practical Guide to Splines*, revised edition, Springer, New York 2001.
- [22] P. Davis, *Interpolation and Approximation*, Blaisdell Pub Company, New York 1963.
- [23] S. De Marchi: On Leja sequences: some results and applications, *Appl. Math. Comput.* 152(3) (2004), 621–647.
- [24] S. De Marchi, A. Sommariva and M. Vianello: Multivariate Christoffel functions and hyperinterpolation, *Dolomites Res. Notes Approx.* 7(2014), 36–33.
- [25] S. De Marchi, R. Schaback and H. Wendland: Near-Optimal Data-Independent Point Locations for Radial Basis Function Interpolation, *Adv. Comput. Math.* 23(3) (2005), 317–330.
- [26] S. De Marchi and M. Vianello: Polynomial approximation on pyramids, cones and solids of rotation *Dolomites Res. Notes Approx*, Proceedings DWCAA12, Vol. 6 (2013), 20–26.
- [27] M. Dubiner, *The theory of multi-dimensional polynomial approximation*, *J. Anal. Math.* 67 (1995), 39–116.
- [28] Gregory E. Fasshauer, *Meshfree Approximation Methods with Matlab*, World Scientific Publishing, Interdisciplinary Mathematical Sciences - Vol 6, 2007.
- [29] Gregory E. Fasshauer, *Meshfree Approximation Methods with Matlab*, Lecture 1, slides. *Dolomites Res. Notes Approx.* - Vol 1, 2008.

- [30] Lidia Fernández, Teresa E. Pérez, Miguel A. Piñar, On Koornwinder classical orthogonal polynomials in two variables, *J. Comput. Appl. Math.* **236** (2012), 3817–3826.
- [31] G.J. Gassner, F. Lörcher, C.-D. Munz and J.S. Hesthaven, Polymorphic nodal elements and their application in discontinuous Galerkin methods, *J. Comput. Phys.* **228** (2009), 1573–1590.
- [32] N. J. Higham: The numerical stability of barycentric Lagrange interpolation, *IMA J. Numer. Anal.* **24** (2004), 547–556.
- [33] G. Mastroianni and D. Occorsio, *Optimal systems of nodes for Lagrange interpolation on bounded intervals. A survey.*, *J. Comput. Appl. Math.* **134**(1-2) (2001), 325–341.
- [34] C.R. Morrow and T.N.L. Patterson: Construction of Algebraic Cubature Rules Using Polynomial Ideal Theory, *SIAM J. Numer. Anal.* **15**(5) (1978), 953–976.
- [35] Ricardo Pachón and LLOYD N. Trefethen: Barycentric-Remez algorithms for best polynomial approximation in the chebfun system, *BIT Numer. Math.* **49** (2009), 721–741.
- [36] Heinz-Joachim Rack: Variation on a Theme of Chebyshev: Sharp Estimates for the Leading Coefficients of Bounded Polynomials, *Dolomites Res. Notes Approx Vol.* **6**(2013), 101–119.
- [37] H.-J. Rack and M. Reimer: The numerical stability of evaluation schemes for polynomials based on the Lagrange interpolation form, *BIT Numer. Math.* **77** (1982), 101–107.
- [38] Remes, E.: Sur la détermination des polynomes d’approximation de degré donnée. *Commun. Soc. Math. Kharkov* **10** (1934).
- [39] T. Rivlin: *An Introduction to the Approximation of Functions*, Dover Pub. Inc, 1969.
- [40] J.G. van der Corput, G. Schaake, *Ungleichungen für Polynome und trigonometrische Polynome*, *Compositio Math.*, **2** (1935), 321–361.
- [41] Schoenberg I. J.: Metric spaces and completely monotone functions, *Ann. of Math.* **39** (1938), 811–841.
- [42] L. L. Schumaker, *Spline Functions - Basic Theory*, Wiley-Interscience, New York 1981.
- [43] A. Sommariva and M. Vianello: Computing approximate Fekete points by QR factorizations of Vandermonde matrices, *Comp. Math. App.* **57** (2009), 1324–1336.
- [44] Sommariva, A., Vianello, M., Zanovello, R.: Nontensorial Clenshaw–Curtis cubature. *Numer. Algorithms* **49**, 409–427 (2008).
- [45] Sloan, Ian H.; Womersley, Robert S. Extremal systems of points and numerical integration on the sphere. *Adv. Comput. Math.* **21** (2004), no. 1-2, 107–125.
- [46] Taylor, M. A.; Wingate, B. A.; Vincent, R. E. An algorithm for computing Fekete points in the triangle. *SIAM J. Numer. Anal.* **38** (2000), no. 5, 1707–1720.

- [47] L. N. Trefethen: *Approximation Theory and Approximation Methods*, SIAM 2013
- [48] P. Vértesi, On the Lebesgue function and Lebesgue constant: a tribute to Paul Erdős, *Bolyai Society of Mathematical Studies*, Vol. 11, Budapest, Janos Bolyai Math. Soc., 2002, pp. 705–728.
- [49] Y. Xu, Christoffel functions and Fourier series for multivariate orthogonal polynomials, *J. Approx. Theory* **82** (1995), 205–239.