Bivariate interpolation at the Padua points: computational aspects

M. CALIARI[†], S. DE MARCHI[‡], R. MONTAGNA[‡] AND M. VIANELLO[†]

[‡]DEPARTMENT OF PURE AND APPLIED MATHEMATICS, UNIVERSITY OF PADUA, ITALY

[†]DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF VERONA, ITALY

Abstract In [2] we gave a simple, geometric and explicit construction of bivariate polynomial interpolation at certain points in the square, experimentally introduced and termed "Padua points" in [5]. Then, we showed that the associated Lebesgue constant has minimal order of growth $\mathcal{O}(\log^2 n)$; see also [4] for an algebraic setting and solution of the interpolation problem. Here we present an accurate and efficient implementation of the interpolation formula, based on the reproducing-kernel algorithm in [6], accompanied by several numerical tests. We also extend the method to polynomial-based interpolation on bivariate compact domains with various geometries, via composition with suitable smooth transformations, and we present an application to surface compression.

1 The Padua points and their properties

For n positive even integer, the Padua points is the set Ξ_n of points on the square $\Omega = [-1,1]^2$, defined by

$$\Xi_n = \{\boldsymbol{\xi} = (z_m, \eta_k),\ 1 \le m \le n+1;\ 1 \le k \le n/2+1\}$$

where

$$z_m = \cos\frac{(m-1)\pi}{n}, \quad \eta_k = \begin{cases} \cos\dfrac{(2k-2)\pi}{n+1} & m \text{ odd} \\ \cos\dfrac{(2k-1)\pi}{n+1} & m \text{ even} \end{cases}$$

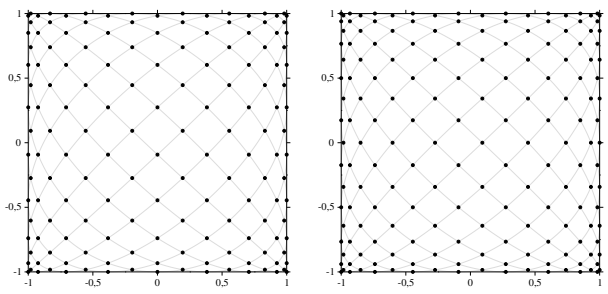
- The cardinality of Ξ_n is $N = (n+1)(n+2)/2 = \dim\left(\mathbb{P}_n(\mathbb{R}^2)\right)$ and they form an unisolvent set for polynomial interpolation in Ω.
- There exists a cubature formula of degree $2n-1$ based on Ξ_n for the product Chebyshev measure on $[-1,1]^2$: weight function $w(x_1,x_2) = 1/\left(\pi^2\sqrt{1-x_1^2}\sqrt{1-x_2^2}\right)$.
- The Padua points lye on the generating parametric curve in $[-1,1]^2$

$$\gamma_n(t) = [-\cos((n+1)t), -\cos(nt)], \quad t \in [0,\pi].$$

They are the self-intersections and the boundary contact points of the curve and they coincide (also for n odd) with the the set of equally spaced points

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

FIGURE 1. *The distribution of $N = 153$ ($n = 16$, left) and $N = 171$ ($n = 17$, right) Padua points and their generating curves.*



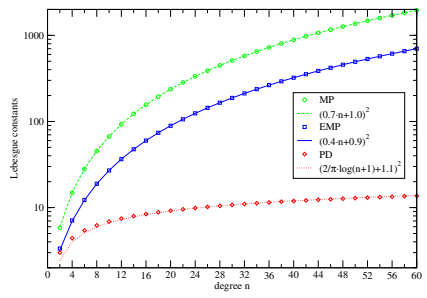
- There are other three families of points generated by the curves $\gamma_n(t) = [-\cos(nt), -\cos((n+1)t)]$, $\gamma_n(t) = [\cos((n+1)t), \cos(nt)]$, $\gamma_n(t) = [\cos(nt), \cos((n+1)t)]$, corresponding to successive 90 degrees clockwise rotations of the square.

2 The Lebesgue constant of the Padua points

It has been rigorously proved in [2] that the Lebesgue constant of the Padua points grows like $\log^2 n$, i.e. there is $C > 0$ such that

$$\Lambda_n \le C \log^2 n, \quad n \ge 2.$$

FIGURE 3. *The experimental behaviour of the Lebesgue constant of Ξ_n, $n \le 60$ in comparison with the ones of Morrow-Patterson (MP) and Extended MP's points [5].*



This behaviour is also shared by interpolation of degree n at Xu points (in an intermediate space between $\mathbb{P}_{n-1}(\mathbb{R}^2)$ and $\mathbb{P}_n(\mathbb{R}^2)$), cf. [9, 3].

3 The interpolant

Let $\boldsymbol{\xi} = (\xi_1,\xi_2) \in \Xi_n$ and $\boldsymbol{x} = (x_1,x_2) \in \Omega$.

THEOREM 1 *The compact interpolation formula (cf. [2]). The Lagrange polynomial at $\boldsymbol{\xi}$ is*

$$L_{\boldsymbol{\xi}}(\boldsymbol{x}) := w_{\boldsymbol{\xi}}[K_n(\boldsymbol{x},\boldsymbol{\xi}) - T_n(x_1)T_n(\xi_1)].$$

$K_n(\cdot,\cdot)$ is the reproducing kernel w.r.t. the inner product associated with the product Chebyshev measure. The interpolant is then

$$L_n^{\mathrm{Pd}} f(\boldsymbol{x}) = \sum_{\boldsymbol{\xi}\in\Xi_n} w_{\boldsymbol{\xi}}\,[K_n(\boldsymbol{x},\boldsymbol{\xi}) - T_n(x_1)T_n(\xi_1)]\, f(\boldsymbol{\xi}).$$

3.1 Matrix formulation

$$D(\Xi_n, f) = \mathrm{diag}\left([w_{\boldsymbol{\xi}} f(\boldsymbol{\xi}),\ \boldsymbol{\xi} = (\xi_1,\xi_2) \in \Xi_n]\right) \in \mathbb{R}^{N\times N},$$

$$T^{(i)}(\Xi_n) = \underbrace{\begin{bmatrix} \cdots & T_0(\xi_i) & \cdots \\ \vdots & \hat{T}_1(\xi_i) & \vdots \\ \vdots & \vdots & \vdots \\ \cdots & \hat{T}_n(\xi_i) & \cdots \end{bmatrix}}_{\boldsymbol{\xi}\in\Xi_n,\ i=1,2} \quad B_0(\Xi_n, f) = \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & \cdots & b_{1,n+1} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,n} & 0 \\ \vdots & \vdots & \iddots & \iddots & \vdots \\ b_{n,1} & b_{n,2} & 0 & \cdots & 0 \\ b_{n+1,1}/2 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

$$\hat{T}_j(\xi_i) = \sqrt{2}T_j(\xi_i),\ j \ge 1, \quad (b_{i,j}) = B(\Xi_n, f) = T^{(1)}(\Xi_n) D(\Xi_n, f) \left(T^{(2)}(\Xi_n)\right)'$$

Given a set $X \subset \Omega$ of target points, then

$$L_n^{\mathrm{Pd}} f(X) = \underbrace{\begin{bmatrix}\cdots & L_n f(\boldsymbol{x}) & \cdots\end{bmatrix}}_{\boldsymbol{x}\in X} = \mathrm{diag}\left(\left(T^{(1)}(X)\right)' B_0(\Xi_n, f)\, T^{(2)}(X)\right)'.$$

A Matlab® code (`Hyperint2d`) for interpolation at Padua points (and hyperinterpolation at Xu points) is available (cf. [6, 7]).

4 Beyond the square: extensions

Consider a surjective map $\sigma\colon \Omega \to K$, $\boldsymbol{t} = (t_1,t_2) \mapsto \boldsymbol{x} = (x_1,x_2)$, with "inverse" $\sigma^{-1}\colon K \to \Omega$, $\sigma^{-1}(\boldsymbol{x}) = \boldsymbol{t}(\boldsymbol{x}) \in \overleftarrow{\sigma}(\boldsymbol{x})$, where $\boldsymbol{t}(\boldsymbol{x})$ denotes a point selected from the inverse image $\overleftarrow{\sigma}(\boldsymbol{x})$. By interpolating the $g = f\circ\sigma$ at the Padua points in Ω, we get a (in general) non-polynomial interpolation

$$\mathcal{L}_n^{\mathrm{Pd}} f(\boldsymbol{x}) = L_n^{\mathrm{Pd}} g(\sigma^{-1}(\boldsymbol{x})), \quad g = f\circ\sigma, \quad \boldsymbol{x} \in K.$$

Thus, f will be sampled at the Padua-like points $K \ni \boldsymbol{x} = \sigma(\boldsymbol{\xi})$.
Key feature: smoothness of the transformation σ (cf. [1]).

4.1 Generalized rectangles

$$K = \{\boldsymbol{x} = (x_1,x_2) :\ a \le x_1 \le b,\ \phi(x_1) \le x_2 \le \psi(x_1)\},$$

ϕ and ψ being suitable functions.

FIGURE 2. *The distribution of $N = 153$ Padua-like points ($n = 16$) in two generalized sectors.*

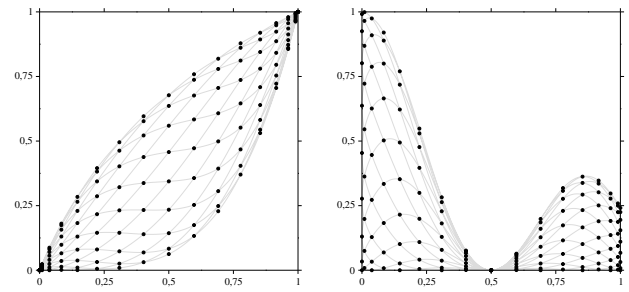


TABLE 2. *Padua-like interpolation errors (in max-norm) of $f(x_1,x_2) = \sin(x_1^2 + x_2^2)$ on the generalized rectangles K_1 (left) and K_2 (right) of Fig. 2; n is the underlying polynomial degree, $N = (n+1)(n+2)/2$ the corresponding number of Padua-like interpolation points.*

	$n=8$	$n=16$	$n=24$	$n=32$	$n=40$
	$N=45$	$N=153$	$N=325$	$N=561$	$N=861$
K_1	7E-3	1E-5	7E-9	2E-12	8E-14
K_2	2E-2	2E-4	1E-6	3E-9	2E-11

4.2 Generalized sectors and starlike domains

Generalized sectors

$$K = \{\boldsymbol{x} = (\rho\cos\theta, \rho\sin\theta) :\ \theta_1 \le \theta \le \theta_2,\ \rho_1(\theta) \le \rho \le \rho_2(\theta)\}$$

Starlike-polar domains

$$K = \{\boldsymbol{x} = (\rho\cos\theta, \rho\sin\theta) :\ 0 \le \theta \le \pi,\ -r(\theta+\pi) \le \rho \le r(\theta)\}$$

FIGURE 3. *The distribution of $N = 153$ Padua-like points ($n = 16$) in polar coordinates in the unit disk (top left) and starlike-polar in the disk (top right), a cardioid (bottom left) and a 4-leaf clover (bottom right).*

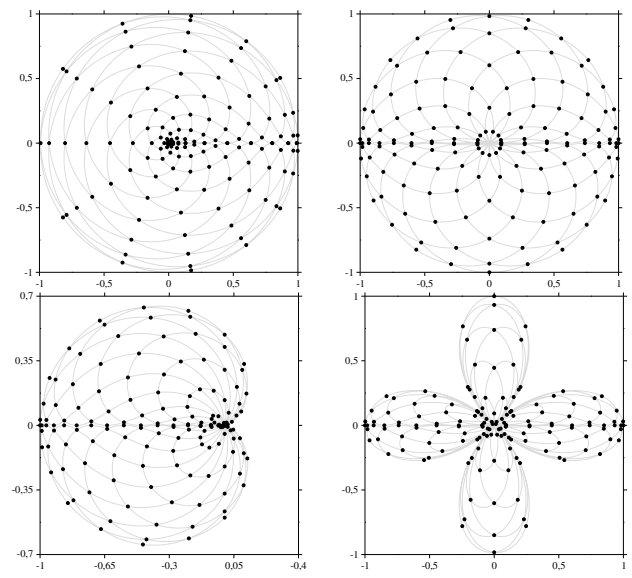


TABLE 3. *Padua-like interpolation errors (in max-norm) of $f(x_1,x_2) = \cos(x_1+x_2)$ on the unit disk in Cartesian, standard polar and starlike-polar coordinates.*

	$n=8$	$n=16$	$n=24$	$n=32$	$n=40$
Cartesian	3E-2	2E-2	4E-3	2E-3	3E-3
polar	9E-2	2E-3	1E-5	7E-8	2E-10
starlike	6E-3	8E-6	1E-9	4E-13	8E-14

5 Surface compression from scattered data by "interpolated interpolations"

We consider the problem of compressing a surface, given as a large scattered data set. This problem can be addressed in several ways, for example by multiresolution methods using splines or radial basis functions. Here we adopted for sufficiently regular surfaces

- Padua-like interpolation of a cubic Shepard-like interpolant (cf. [8])

The compression ratio obtained is

$$\text{compr. ratio} = 3 \times \frac{\text{numb. of scatt. pts.}}{\text{numb. of Padua nodes}} \approx 6 \times \frac{\text{numb. of scatt. pts.}}{n^2},$$

where n is the underlying polynomial degree.

TABLE 4. *Compression errors (in max-norm) for the Franke test function on $[0,1]^2$, sampled at randomly generated point sets; last row and column: actual errors of the Padua and Shepard-like interpolants on the test function.*

random pts.	$n=16$	$n=24$	$n=32$	$n=40$	$n=48$	"true" Shep.
5000	2E-2	2E-3	1E-4	1E-4	1E-4	1E-4
10000	2E-2	2E-3	1E-4	3E-5	1E-4	2E-4
20000	2E-2	2E-3	1E-4	7E-6	5E-6	2E-5
40000	2E-2	2E-3	1E-4	3E-6	6E-7	1E-6
"true" Padua	2E-2	2E-3	1E-4	2E-6	3E-8	

TABLE 5. *The compression ratios corresponding to the example above.*

random pts.	$n=16$	$n=24$	$n=32$	$n=40$	$n=48$
5000	104:1	48:1	28:1	18:1	13:1
10000	208:1	96:1	55:1	36:1	25:1
20000	416:1	192:1	110:1	71:1	50:1
40000	832:1	385:1	221:1	143:1	100:1

5.1 Compression of a Finite Element PDE solution

- Poisson equation with Dirichlet boundary conditions

$$\begin{cases} \Delta f(\boldsymbol{x}) = -10 & \boldsymbol{x} \in K \\ f(\boldsymbol{x}) = 0 & \boldsymbol{x} \in \partial K \end{cases}$$

where K is the "lynx-eye" shaped domain in Fig. 5.

- Finite Element discretization on a Delaunay mesh with 81796 triangles and 41402 nodes.
- Padua-like interpolation of the Finite Element solution.

FIGURE 5. *Left: the distribution of $N = 153$ Padua-like points ($n = 16$) in the "lynx-eye" shaped domain (generalized sector). Right: a detail of the Finite Element mesh in the domain, near the internal boundary.*

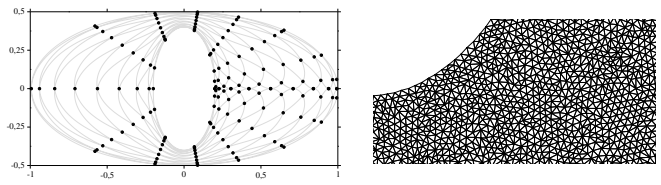
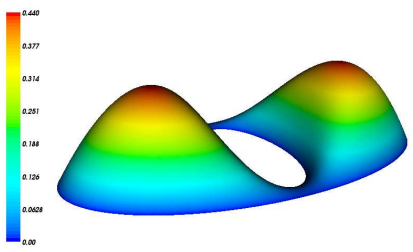


TABLE 6. *Compression errors (in the max-norm) for the Finite Element solution of the Poisson equation above.*

mesh size	$n=8$	$n=12$	$n=16$	$n=20$	$n=24$	$n=28$	$n=32$
41402	6E-2	3E-2	1E-2	5E-3	2E-3	9E-4	7E-4

FIGURE 6. *Plot of the Padua-like interpolated solution ($n = 24$: compression ratio $\approx$ 400:1, compression error $\approx 2\cdot 10^{-3}$).*



References

[1] L. BOS, M. CALIARI, S. DE MARCHI, AND M. VIANELLO, *Bivariate interpolation at Xu points: results, extensions and applications*, ETNA **25** (2006), pp. 1–16.

[2] 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, in press (available online 15 May 2006).

[3] L. BOS, S. DE MARCHI, AND M. VIANELLO, *On the Lebesgue constant for the Xu interpolation formula*, J. Approx. Theory **141** (2006), pp. 134–141.

[4] L. BOS, S. DE MARCHI, M. VIANELLO, AND Y. XU, *Bivariate Lagrange interpolation at the Padua points: the ideal theory approach*, submitted (2006).

[5] M. CALIARI, S. DE MARCHI AND M. VIANELLO, *Bivariate polynomial interpolation on the square at new nodal sets*, Appl. Math. Comput. **162** (2005), pp. 161–174.

[6] M. CALIARI, M. VIANELLO, S. DE MARCHI, AND R. MONTAGNA, *Hyper2d: a numerical code for hyperinterpolation on rectangles*, Appl. Math. Comput., in press (available online July 31, 2006).

[7] M. CALIARI, M. VIANELLO, S. DE MARCHI, AND R. MONTAGNA `Hyperint2d`, code available at `http://www.math.unipd.it/~mcaliari/software`.

[8] R. J. RENKA, *Algorithm 790: CSHEP2D: Cubic Shepard Method for Bivariate Interpolation of Scattered Data*, ACM Trans. Math. Software **25** (1999), pp. 70–73.

[9] Y. XU, *Lagrange interpolation on Chebyshev points of two variables*, J. Approx. Theory **87** (1996), pp. 220–238.