BIVARIATE INTERPOLATION AT XU POINTS: RESULTS, EXTENSIONS AND APPLICATIONS*

LEN BOS†, MARCO CALIARI‡, STEFANO DE MARCHI§, AND MARCO VIANELLO¶

Dedicated to Ed Saff on the occasion of his 60th birthday

Abstract. In a recent paper, Y. Xu proposed a set of Chebyshev-like points for polynomial interpolation on the square $[-1,1]^2$. We have recently proved that the Lebesgue constant of these points grows like $\log^2$ of the degree (as with the best known points for the square), and we have implemented an accurate version of their Lagrange interpolation formula at linear cost. Here we construct non-polynomial Xu-like interpolation formulas on bivariate compact domains with various geometries, by means of composition with suitable smooth transformations. Moreover, we show applications of Xu-like interpolation to the compression of surfaces given as large scattered data sets.

Key words. bivariate polynomial interpolation, Xu points, Lebesgue constant, domains transformations, generalized rectangles, generalized sectors, large scattered data sets, surface compression

AMS subject classification. 65D05

1. Introduction. The problem of choosing good nodes on a given compact set is a central one in polynomial interpolation. Besides unisolvence, which is by no means an easy problem (see, e.g., [6, 2, 13]), for practical purposes one needs slow growth of the Lebesgue constant, together with computational stability and efficiency.

Suppose that $K \subset R^d$ is a compact set with non-empty interior. Let V be a subspace of Π_n^d, the polynomials of degree n in d variables, of dimension $\dim(V) =: N$. Then given N points $X := \{\mathbf{x}_k\}_{k=1}^N \subset K$, the polynomial interpolation problem associated to V and X is find for each $f \in C(K)$ a polynomial $p \in V$ such that

$$p(\mathbf{x}_k) = f(\mathbf{x}_k), \quad k = 1, \ldots, N \, .$$

If this is always possible the problem is said to be unisolvent. And if this is indeed the case we may construct the so-called Lagrange fundamental polynomials $\ell_j(\mathbf{x})$ with the property that

$$\ell_j(\mathbf{x}_k) = \delta_{jk},$$

the Kronecker delta. Further, the interpolant itself may be written as

$$(Lf)(\mathbf{x}) = \sum_{k=1}^{N} f(\mathbf{x}_k)\ell_k(\mathbf{x}) \, .$$

The mapping $f \mapsto Lf$ may be regarded as an operator from $C(K)$ (equipped with the uniform norm) to itself, and as such has an operator norm $\|L\|$. Classically, when $K = [-1,1]$ and $V = \Pi_n^1$, this norm is known as the Lebesgue constant and it is known that then

* Received April 18, 2005. Accepted for publication January 17, 2006. Recommended by X. Li. Work supported by MIUR-Prin2003 SCoCoDe project, by the project CPDA028291 of the University of Padova, by the GNCS-INdAM and by the ex-60% funds of the Universities of Padova and Verona.

† Dept. of Mathematics and Statistics, University of Calgary.

‡ Dept. of Computer Science, University of Verona.

§ Dept. of Computer Science, University of Verona, S.da Le Grazie 15, 37134 Verona, Italy, (stefano.demarchi@univr.it); corresponding author.

¶ Dept. of Pure and Applied Mathematics, University of Padova.

$\|L\| \geq C \log n$ and that this minimal order of growth is attained, for example, by the Chebyshev points (see e.g. [7]).

In the multivariate case much less is known. From Berman's Theorem (cf. [18, Thms. 6.4 and 6.5]) it follows that for $K = B^d$, the unit ball in R^d, $d \geq 2$, and $V = \Pi_n^d$, the Lebesgue constant has at least a rate of growth of $\mathcal{O}(n^{(d-1)/2})$. In the tensor product case, when $K = [-1,1]^d$ and $V = \bigotimes_{k=1}^d \Pi_n^1$, then $\|L\| \geq C(\log n)^d$ and this minimal rate of growth is attained for the tensor product of the univariate Chebyshev points. However, even for the cube and the polynomials of total degree n, i.e., for $K = [-1,1]^d$ and $V = \Pi_n^d$, the minimal rate of growth is not known.

Recently Y. Xu [25] introduced a set of Chebyshev-like points for the square $K = [-1,1]^2$, and $V = V_n$ a certain subspace of polynomials such that

$$\Pi_{n-1}^2 \subset V_n \subset \Pi_n^2, \quad N = \dim(V_n) = \dim(\Pi_{n-1}^2) + \left\lfloor \frac{n}{2} \right\rfloor \ . \tag{1.1}$$

It should be remarked that V_n, although not a total degree space of polynomials, is much closer to Π_{n-1}^2 than to the corresponding tensor-product space $\bigotimes_{k=1}^2 \Pi_{n-1}^1$ which has dimension n^2.

In [3] we investigated numerical aspects of the Xu polynomial interpolation formula in the square. The numerical experiments gave us good evidence that the Lebesgue constant for these Xu points has growth of the order $(\log n)^2$ (just as in the tensor product case, and in contrast to the case of the ball where the minimal growth would be of order $\sqrt{n}$). This has been rigorously proved in [4]. Moreover, we have been able to implement the Xu interpolation formula in a stable way, with a computational cost which is in practice linear in the number N of interpolation points. From this we may conclude that the Xu points are excellent points for practical polynomial interpolation.

In the present paper, we first give a survey of the known results on polynomial interpolation at the Xu points. Then, we extend the interpolation method to bivariate compact domains, which are smooth transformations of the square. This leads to non-polynomial Xu-like interpolation formulas, which work on domains with quite different geometries, like generalized rectangles (in cartesian coordinates), generalized sectors and starlike domains (in polar coordinates). Finally, we show an application of Xu-like interpolation to the compression of sufficiently regular surfaces, given as large scattered data sets. Remarkable compression ratios are obtained, simply by interpolating a suitable Shepard-like interpolant at the Xu points.

2. A survey on polynomial interpolation at Xu points. We start by recalling briefly the construction of the Xu interpolation formula of degree n on the square $[-1,1]^2$. In what follows we restrict, for simplicity's sake, to even degrees n. Considering the Chebyshev-Lobatto points on the interval $[-1,1]$

$$\xi_k = \xi_{k,n} = \cos\frac{k\pi}{n}, \quad k = 0,\dots,n, \quad n = 2m \ , \tag{2.1}$$

the Xu interpolation points on the square are defined as the two dimensional Chebyshev array $X_N = \{\mathbf{z}_{r,s}\}$ of dimension $N = n(n+2)/2$

$$\begin{aligned} \mathbf{z}_{2i,2j+1} &= (\xi_{2i}, \xi_{2j+1}), \quad 0 \leq i \leq m, \ 0 \leq j \leq m-1 \ , \\ \mathbf{z}_{2i+1,2j} &= (\xi_{2i+1}, \xi_{2j}), \quad 0 \leq i \leq m-1, \ 0 \leq j \leq m \ . \end{aligned} \tag{2.2}$$

The Xu interpolant in Lagrange form of a given function f on the square $[-1,1]^2$ is

$$L_n^{\mathrm{Xu}} f(\mathbf{x}) = \sum_{\mathbf{z}_{r,s} \in X_N} f(\mathbf{z}_{r,s}) \ell_n(\mathbf{x}, \mathbf{z}_{r,s}), \quad \ell_n(\mathbf{x}, \mathbf{z}_{r,s}) := \frac{K_n^*(\mathbf{x}, \mathbf{z}_{r,s})}{K_n^*(\mathbf{z}_{r,s}, \mathbf{z}_{r,s})} \ , \tag{2.3}$$

where the polynomials $K_n^*(\cdot, \mathbf{z}_{r,s})$ are given by

$$K_n^*(\mathbf{x}, \mathbf{z}_{r,s}) := \frac{1}{2}\left(K_n(\mathbf{x}, \mathbf{z}_{r,s}) + K_{n+1}(\mathbf{x}, \mathbf{z}_{r,s})\right) + - \frac{1}{2}(-1)^r (T_n(x_1) - T_n(x_2)) \; ; \tag{2.4}$$

here x_1, x_2 are the coordinates of the generic point $\mathbf{x} = (x_1, x_2)$ and T_n is the Chebyshev polynomial of the first kind of degree n, $T_n(x) = \cos(n \arccos x)$. In particular when $\mathbf{x} = \mathbf{z}_{r,s}$ (cf. [25, p. 229, (2.18)])

$$K_n^*(\mathbf{z}_{r,s}, \mathbf{z}_{r,s}) = \frac{1}{2}\left(K_n(\mathbf{z}_{r,s}, \mathbf{z}_{r,s}) + K_{n+1}(\mathbf{z}_{r,s}, \mathbf{z}_{r,s})\right) - 1 \; . \tag{2.5}$$

The polynomials $K_n(\mathbf{x}, \mathbf{y})$ can be represented in the form

$$\begin{aligned} K_n(\mathbf{x}, \mathbf{y}) = {} & D_n(\theta_1 + \phi_1, \theta_2 + \phi_2) + D_n(\theta_1 + \phi_1, \theta_2 - \phi_2) + \\ & + D_n(\theta_1 - \phi_1, \theta_2 + \phi_2) + D_n(\theta_1 - \phi_1, \theta_2 - \phi_2) \; , \\ & \mathbf{x} = (\cos\theta_1, \cos\theta_2), \quad \mathbf{y} = (\cos\phi_1, \cos\phi_2) \; , \end{aligned} \tag{2.6}$$

where the function D_n is defined by

$$D_n(\alpha, \beta) = \frac{1}{2} \frac{\cos((n - 1/2)\alpha)\cos(\alpha/2) - \cos((n - 1/2)\beta)\cos(\beta/2)}{\cos\alpha - \cos\beta} \; . \tag{2.7}$$

As shown in [25], the values $K_n^*(\mathbf{z}_{r,s}, \mathbf{z}_{r,s})$ are explicitly known in terms of the degree n, that is

$$K_n^*(\mathbf{z}_{r,s}, \mathbf{z}_{r,s}) = \begin{cases} n^2 & \begin{cases} r = 0 \quad \text{or} \quad r = n, \quad s \text{ odd} \\ s = 0 \quad \text{or} \quad s = n, \quad r \text{ odd} \end{cases} \\ n^2/2 & \text{in all other cases} \; . \end{cases} \tag{2.8}$$

Observe that this constructive approach immediately yields unisolvence of the interpolation problem, since for any given basis of the underlying polynomial space V_n the corresponding Vandermonde system has a solution for every N-dimensional vector $\{f(\mathbf{z}_{r,s})\}$, and thus the Vandermonde matrix is invertible.

2.1. Computational aspects. Rearranging (2.7) in the case that $\cos(\alpha) = \cos(\beta)$, allows us to give a form of the interpolation formula with pointwise evaluation cost $\mathcal{O}(N)$. However, the interpolation formula (2.3)-(2.6) evaluated via (2.7) turns out to be severely ill-conditioned, as has been shown in [3]. Stabilization can be obtained rewriting D_n by simple trigonometric manipulations

$$D_n(\alpha, \beta) = \frac{1}{4}\left(U_{n-1}(\cos\phi)U_{n-1}(\cos\psi) + U_{n-2}(\cos\phi)U_{n-2}(\cos\psi)\right) \; , \tag{2.9}$$

where $\phi = (\alpha - \beta)/2$, $\psi = (\alpha + \beta)/2$, and U_n denotes the usual Chebyshev polynomial of the second kind. Now, computing the polynomials U_n by the well-known three-term recurrence relation

$$\begin{cases} U_0(\cos\theta) = 1, \quad U_1(\cos\theta) = 2\cos\theta, \\ U_n(\cos\theta) = 2\cos\theta\, U_{n-1}(\cos\theta) - U_{n-2}(\cos\theta), \quad n \geq 2 \; , \end{cases} \tag{2.10}$$

the evaluation of $D_n(\alpha, \beta)$ becomes stable, but the computational cost is $\mathcal{O}(n)$ instead of $\mathcal{O}(1)$. Then, it is not difficult to see that the dominant term in the final complexity for the pointwise evaluation of $L_n^{\mathrm{xu}} f(\mathbf{x})$ is $8nN \sim 8\sqrt{2}N^{3/2} \sim 4n^3$ flops.

An effective way to reduce the computational cost of the stabilized formula (2.9), still preserving high accuracy, is to compute the Chebyshev polynomials of the second kind U_n by the three-term recurrence relation (2.10) only when the representation $U_n(\cos\theta) = \sin(n+1)\theta/\sin\theta$ (whose cost is $\mathcal{O}(1)$ in n and θ) is ill-conditioned, say when $|\theta - k\pi| \le \varepsilon$ for a "small" value of ε. In this case, it is important to estimate the average use percentage of the recurrence in evaluating all the Lagrange basis polynomials. In [3], we resorted to some probabilistic considerations. Indeed, taking random, uniformly distributed evaluation points, such a percentage becomes a random variable (function of a uniform random variable), whose expectation, say η, depends on the threshold ε but not on the degree n. This is clearly seen in Tables 2.1 and 2.2, where it is shown that the averages up to one million random points converge to a value, that does not depend on the degree n.

TABLE 2.1

Averages of the use percentage of recurrence relation (2.10), up to one million uniform random points, in evaluating all the Lagrange basis polynomials at degree $n = 20$.

# of random points	% recurr. (averages)	
	$\varepsilon = 0.01$	$\varepsilon = 0.1$
1.0E+01	0.50	7.00
1.0E+02	0.75	6.25
1.0E+03	0.69	6.27
1.0E+04	0.63	6.34
1.0E+05	0.64	6.36
1.0E+06	0.64	6.37

TABLE 2.2

Average use percentage η of recurrence relation (2.10), in evaluating all the Lagrange basis polynomials at different degrees.

degree n	percentage η	
	$\varepsilon = 0.01$	$\varepsilon = 0.1$
20	0.64	6.37
40	0.64	6.37
80	0.64	6.37

The evaluation of $K_n^*(\mathbf{x}, \mathbf{z}_{r,s})$ using only the trigonometric representation of $U_n(\cos\theta)$ costs about $8 \times 4 = 32$ evaluations of the sine function, recalling that D_n and D_{n+1} appear with the same arguments in (2.4), (2.6). Denoting by $\mathrm{c_{sin}}$ the average evaluation cost of the sine function (which actually depends on its internal implementation), the average complexity for the evaluation of the Xu interpolant $L_n^{\mathrm{Xu}} f(\mathbf{x})$ is of the order of

$$C(n,\varepsilon) := 8n\tau N + 32\mathrm{c_{sin}}(1-\tau)N \sim 4n^3\tau + 16\mathrm{c_{sin}}(1-\tau)n^2 \text{ flops}\,, \tag{2.11}$$

where $\tau = \eta/100$. Using the experimental value $\mathrm{c_{sin}} = 10$ (obtained with GNU Fortran, but consistent with the usual implementations), we can conclude that, for $\varepsilon \le 0.01$ (i.e., $\tau \le 0.0064$), the size of the ratio $C(n,\varepsilon)/N$ remains constant up to degrees of the order of hundreds, that is in practical applications the computational cost can be considered linear in the number N of Xu points.

A more sophisticated implementation may take into account that, for low degrees, the recurrence relation costs less than the trigonometric representation in evaluating $U_n(\cos\theta)$. Comparing the dominant costs, the former should be used when $4n^3 < 16\mathrm{c_{sin}}n^2$, i.e., $n < 4\mathrm{c_{sin}}$. Our Fortran implementation of the Xu interpolation formula resorts to all the tricks just described, in particular the last one with the experimental value $\mathrm{c_{sin}} = 10$, i.e., a threshold degree $n = 40$; see [9].

2.2. The Lebesgue constant of the Xu points. In this section we report some numerical and theoretical results, concerning another key feature of the Xu interpolation formula, that is the behavior of its Lebesgue constant; see [3, 4]. First, it comes easy to bound the Lebesgue constant linearly in the dimension of the polynomial space V_n, which already shows that the Xu points are good candidates for interpolation purposes. Indeed, from the well-known bound for Chebyshev polynomials of the second kind $|U_n(\cos\theta)| \le n+1$, we get easily

$$|\ell_n(\mathbf{x}, \mathbf{z}_{r,s})| = \left| \frac{K_n^*(\mathbf{x}, \mathbf{z}_{r,s})}{K_n^*(\mathbf{z}_{r,s}, \mathbf{z}_{r,s})} \right| \le \frac{(n+1)^2 + 2n^2 + (n-1)^2}{n^2} = 4 + \frac{2}{n^2} \,. \tag{2.12}$$

Defining, in the usual way, the Lebesgue function for the Xu interpolation points

$$\lambda_n^{\mathrm{Xu}}(\mathbf{x}) := \sum_{\mathbf{z}_{r,s} \in X_N} |\ell_n(\mathbf{x}, \mathbf{z}_{r,s})| \,, \tag{2.13}$$

we finally obtain the following bound of the Lebesgue constant

$$\Lambda_n^{\mathrm{Xu}} := \|L_n^{\mathrm{Xu}}\| = \max_{\mathbf{x} \in [-1,1]^2} \lambda_n^{\mathrm{Xu}}(\mathbf{x}) \le \left(4 + \frac{2}{n^2}\right) N \sim 4N \sim 2n^2 \,. \tag{2.14}$$

However, (2.14) is a substantial overestimate of the actual Lebesgue constant. In fact, the Lebesgue function turns out to be symmetric and seems to attain its maximum at the four vertices of the square. A wide set of large-scale numerical experiments on the maximization of the Lebesgue function, performed in [3], confirmed this fact and gave the results summarized in Fig. 2.1 (right), where we compare the Lebesgue constant of Xu points up to degree $n = 100$ with the least-square fitting function $(0.95 + 2/\pi \, \log(n+1))^2$ and the theoretical bound for tensor-product Chebyshev-Lobatto interpolation of degree n (cf. [7]), i.e., $(1 + 2/\pi \, \log(n+1))^2$. These computations gave a sound basis for the following

CONJECTURE. *The Lebesgue function λ_n^{Xu} of the Xu interpolation points can be bounded as*

$$\max_{\mathbf{x} \in [-1,1]^2} \lambda_n^{\mathrm{Xu}}(\mathbf{x}) = \Lambda_n^{\mathrm{Xu}} \le A_n \sim (2/\pi \log(n+1))^2, \quad n \to \infty \,. \tag{2.15}$$

Moreover, the maximum is attained at the four vertices of the square.

The conjecture has been partially proved in [4], at least concerning the actual order of growth of the Lebesgue constant of the Xu points. Indeed, we have obtained the following rigorous estimate

THEOREM 2.1. *The Lebesgue constant of the Xu interpolation points, Λ_n^{Xu}, is bounded by*

$$\Lambda_n^{\mathrm{Xu}} \le 8 \left(\frac{2}{\pi} \log n + 5\right)^2 + \frac{3}{4} \,. \tag{2.16}$$

This means that the Lebesgue constant of the Xu points has the same order of growth as that of the best known interpolation nodes for the square, namely the "Padua points" recently introduced in [8] (for which, however, only numerical results are known). Such points, as another important nodal set for polynomial interpolation and cubature on the square, the Morrow-Patterson points [17, 24], are equally spaced with respect to the Dubiner metric ([10], see also [5]), which on the square $K = [-1,1]^2$ turns out to be $\delta_K(\mathbf{x}, \mathbf{y}) = \max\{|\arccos x_1 - \arccos y_1|, |\arccos x_2 - \arccos y_2|\}$. Now, it is worth stressing that also the Xu points are equally spaced in the Dubiner metric. This fact confirms once more the conjecture stated in [8] concerning near-optimality of nodal sets: "*Nearly optimal points for polynomial interpolation on a compact K are asymptotically equidistributed with respect to the Dubiner metric on K*".

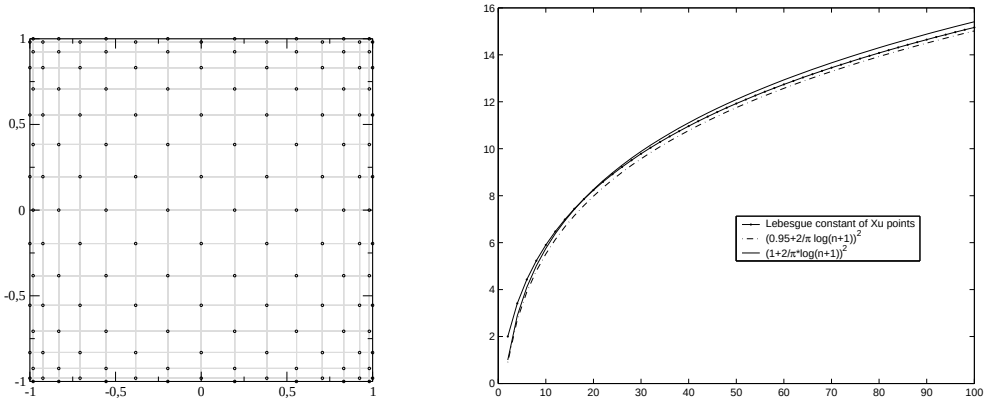

FIG. 2.1. *Left: the distribution of $N = 144$ Xu-like points (degree $n = 16$) in the square $[-1,1]^2$. Right: the Lebesgue constant of the Xu points up to degree $n = 100$.*

2.3. Convergence and approximation. The above results about the Lebesgue constant of the Xu interpolation points, allow us to immediately derive convergence estimates for the corresponding interpolation formula. Recall that the reference polynomial space for the Xu points, V_n in (1.1), is not a total degree space of polynomials; see [25] for its rigorous definition, which is strictly related to the construction of minimal cubature formulas for the product Chebyshev measure on $[-1,1]^2$.

¿From (1.1), however, we get trivially that $E_n(f) \le \inf_{p\in V_n} \|f-p\|_{\infty,K} \le E_{n-1}(f)$, $E_n(f)$ denoting as usual the best uniform approximation error to f on $K = [-1,1]^2$ by polynomials in Π_n^2, and thus the convergence estimate

$$\|f - L_n^{\mathrm{Xu}} f\|_{\infty,K} \le (1+\Lambda_n^{\mathrm{Xu}}) \inf_{p\in V_n} \|f-p\|_{\infty,K} \le (1+\Lambda_n^{\mathrm{Xu}}) E_{n-1}(f) \ . \tag{2.17}$$

The rate of decay of $E_n(f)$ as $n\to\infty$ depends on the degree of smoothness of f, in view of multivariate generalizations of Jackson's theorem (cf. [1]). In particular, from (2.17) and [1] we obtain the convergence estimate

$$\|f - L_n^{\mathrm{Xu}} f\|_{\infty,K} = \mathcal{O}(n^{-\alpha}\ \log^2 n) \ , \quad f\in C^\alpha(K) \ , \quad 0<\alpha<\infty \ . \tag{2.18}$$

The actual approximation behavior of the Xu interpolation formula has still to be investigated thoroughly. Theoretical results and numerical tests, however, have shown that it can be considered among the best approximation tools with polynomials on the square, especially if one considers its low computational cost.

For the purpose of illustration, we report some numerical results taken from [8]. In Table 2.3, we display the Lebesgue constants (rounded to the nearest integer) of several nodal sets at a sequence of degrees. The degrees have been chosen in such a way that the dimension N of polynomial spaces, and thus the number of function evaluations in the interpolation process, is as close as possible to the dimension of some tensor-product polynomial spaces. As already observed, PD (Padua) and Xu points have the smallest Lebesgue constants, which are very close to $(1+2/\pi\log(n+1))^2$. In Table 2.4, we compare interpolation at Xu points with the other nodal sets and with tensor-product Chebyshev-Lobatto interpolation, in the recovery of the well-known Franke test function [12]. Notice that Xu interpolation errors are very close to those at Padua points, and much smaller than the errors given by the other sets of points (except for the highest degree). But it is also important to recall that, with the present state of

the art, the computational complexity is $\mathcal{O}(N)$ for the Xu points, whereas it is $\mathcal{O}(N^3)$ for the Padua points (due to direct solution of a suitable Vandermonde system).

TABLE 2.3
Lebesgue constants (rounded to the nearest integer) of different nodal sets: Morrow-Patterson (MP), Extended Morrow-Patterson (EMP), Padua points (PD), Xu points.

interp. pts.	Λ_{34}	Λ_{48}	Λ_{62}	Λ_{76}
MP	649	1264	2082	3102
EMP	237	456	746	1106
PD	11	13	14	15
XU	10	12	13	14

3. Beyond the square: extension to other domains. In this section, we construct non-polynomial (but polynomial based) interpolation formulas at Xu-like points on bivariate domains with different geometric structures, by means of suitable transformations. A similar approach has already been used, e.g., in [22] concerning the extension of adaptive approximation with bivariate Chebyshev series.

Consider a sufficiently regular function f defined on a bivariate compact domain K, that corresponds to the square $[-1,1]^2$ through a smooth surjective transformation

$$\sigma : [-1,1]^2 \to K \ , \ \mathbf{t} = (t_1, t_2) \mapsto \mathbf{x} = (x_1, x_2) \ . \tag{3.1}$$

Moreover, even though σ is not one-to-one in general, assume that we can define a global "inverse-like" mapping (which for convenience we shall still denote by σ^{-1})

$$\sigma^{-1} : K \to [-1,1]^2 \ , \ \ \sigma^{-1}(\mathbf{x}) = \mathbf{t}(\mathbf{x}) \in \overleftarrow{\sigma}(\mathbf{x}) \ , \tag{3.2}$$

where $\mathbf{t}(\mathbf{x})$ denotes a point selected in some manner from the inverse image $\overleftarrow{\sigma}(\mathbf{x})$. The latter choice will be made explicit, in instances of non-injective transformations (see below, e.g., the case of polar coordinates).

Now, by interpolating the composition $g = f \circ \sigma$ at the Xu points in $[-1,1]^2$, we get a (in general) non-polynomial interpolation formula

$$\mathcal{L}_n^{\mathrm{Xu}} f(\mathbf{x}) = L_n^{\mathrm{Xu}} g(\sigma^{-1}(\mathbf{x})) \ , \ \ g = f \circ \sigma \ , \ \ \mathbf{x} \in K \ , \tag{3.3}$$

cf. (2.3). This means that f will be sampled at the Xu-like points (cf. (2.2))

$$K \ni \mathbf{x}_{r,s} = \sigma(\mathbf{z}_{r,s}) \ . \tag{3.4}$$

Observe that theoretically, in view of multivariate extensions of Jackson's theorem (cf., e.g., [1]), when the function f is globally Hölder-continuous in K, a Hölder-continuous transformation suffices to ensure convergence of Xu-like interpolation. On the other hand, singularities of σ lead in general to nonconvergence.

However, a key point in order to avoid loss of smoothness in this process and thus an artificial slowing down of convergence, is to choose a transformation as smooth as possible, and in any case with at least the same degree of regularity as the function f. This role of the transformation will be clarified in the examples below. We stress that, whereas the smoothness of σ is a key feature for the effectiveness of the interpolation method, the regularity of σ^{-1} plays practically no role (indeed σ^{-1} can be singular without problems, see again the case of polar coordinates).

Now we are ready to describe three important classes of domain geometries, with corresponding transformations (the terminology being usual in the field of numerical cubature).

TABLE 2.4
Interpolation errors for the Franke test function on $[0,1]^2$, by different nodal sets (n is the degree, N the number of nodes): Tensor Product Chebyshev (TPC), Morrow-Patterson (MP), Extended Morrow-Patterson (EMP), Padua points (PD), Xu points.

TPC $n,\ N=(n+1)^2$	1E-03 24, 625	3E-06 34, 1225	1E-09 44, 2025	2E-13 54, 3025
MP $n,\ N=(n+1)(n+2)/2$	1E-03 34, 630	3E-06 48, 1225	1E-09 62, 2016	2E-13 76, 3003
EMP $n,\ N=(n+1)(n+2)/2$	6E-04 34, 630	1E-06 48, 1225	5E-10 62, 2016	5E-14 76, 3003
PD $n,\ N=(n+1)(n+2)/2$	4E-05 34, 630	3E-08 48, 1225	5E-12 62, 2016	2E-14 76, 3003
XU $n,\ N=n(n+2)/2$	3E-05 34, 612	5E-08 48, 1200	8E-12 62, 1984	2E-13 76, 2964

In all the tables below the interpolation errors have been computed in the max-norm, on the σ-image of a 50×50 control grid in $[-1,1]^2$, and are rounded to the first significant digit. In all the figures, together with the Xu-like interpolation points, we show how the grid of lines (in grey) where the original Xu points lie in the square (see Fig. 2.1) is deformed by the transformation σ.

3.1. Generalized rectangles (Cartesian coordinates). The domain K is defined by

$$K=\{\mathbf{x}=(x_1,x_2):\ a\le x_1\le b\,,\ \phi(x_1)\le x_2\le \psi(x_1)\}\,, \tag{3.5}$$

ϕ and ψ being suitable functions (so that double integrals can be iterated). Here the transformation σ can be defined as

$$\sigma(t_1,t_2)=(x_1(t_1,t_2),x_2(t_1,t_2)) \tag{3.6}$$

with

$$\begin{aligned} x_1(t_1,t_2)&=a+(t_1+1)\,\frac{b-a}{2}\,,\\ x_2(t_1,t_2)&=\phi(x_1)+(t_2+1)\,\frac{\psi(x_1)-\phi(x_1)}{2}\,, \end{aligned} \tag{3.7}$$

and "inverse" given by

$$\begin{aligned} t_1(x_1,x_2)&=-1+2\,\frac{x_1-a}{b-a}\,,\\ t_2(x_1,x_2)&=\begin{cases} -1+2\,\frac{x_2-\phi(x_1)}{\psi(x_1)-\phi(x_1)} & \psi(x_1)\ne\phi(x_1)\\ -1 & \psi(x_1)=\phi(x_1)\end{cases} \end{aligned} \tag{3.8}$$

The regularity of this transformation is clearly determined by the regularity of ϕ and ψ. Notice that when $\psi(x_1)=\phi(x_1)$ at some abscissa x_1, the transformation σ is non-injective, but with the (arbitrary) choice made in the inverse image of such points the method works without problems. In Fig. 3.1 we show the distribution of $N=144$ Xu-like points (corresponding to polynomial degree $n=16$) for two generalized rectangles K_1 and K_2 like (3.5) (both with singular points for σ^{-1}, where $\phi(x_1)=\psi(x_1)$)

$$\begin{aligned} &K_1: a=0,\ b=1,\ \phi(x_1)=x_1^4,\ \psi(x_1)=\log(1+4x_1)/\log 5\,,\\ &K_2: a=0,\ b=1,\ \phi(x_1)\equiv 0,\ \psi(x_1)=4(x_1-0.5)^2(1+\sin 4x_1)\,. \end{aligned} \tag{3.9}$$

In Tables 3.1–3.2, finally, we report the interpolation errors on such domains for two functions with different degree of regularity, correspondingly to a sequence of Xu-like nodal sets.

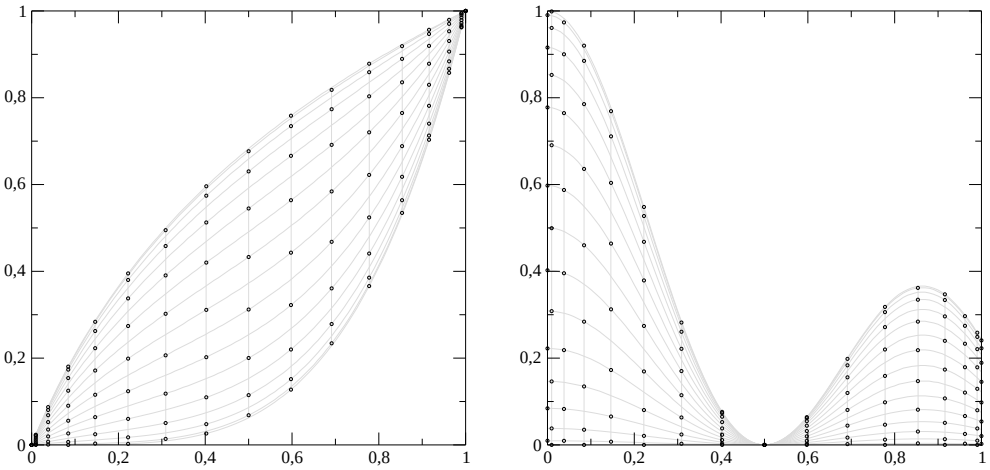

FIG. 3.1. *The distribution of $N = 144$ Xu-like points (degree $n = 16$) in the generalized rectangles K_1 (left) and K_2 (right) defined in (3.9).*

TABLE 3.1

Xu-like interpolation errors of $f(x_1, x_2) = \sin(x_1^2 + x_2^2)$ on the generalized rectangles K_1 and K_2 of Fig. 3.1; n is the underlying polynomial degree, $N = n(n+2)/2$ the corresponding number of Xu-like interpolation points (3.4).

n	8	16	24	32	40
N	40	144	312	544	840
K_1	1E-2	2E-5	1E-8	4E-12	5E-14
K_2	3E-2	2E-4	2E-6	4E-9	3E-11

TABLE 3.2

As in Table 3.1 for the function $f(x_1, x_2) = |x_1 - x_2|^3$.

n	8	16	24	32	40
N	40	144	312	544	840
K_1	3E-4	5E-5	1E-5	5E-6	3E-6
K_2	4E-2	3E-3	9E-4	4E-4	2E-4

3.2. Generalized sectors (polar coordinates). The domain K is defined by

$$K = \{\mathbf{x} = (\rho\cos\theta, \rho\sin\theta) : \theta_1 \le \theta \le \theta_2\,,\ \rho_1(\theta) \le \rho \le \rho_2(\theta)\}\,, \tag{3.10}$$

and the transformation is the composition of one analogous to (3.7) with obvious adjustments from Cartesian to polar coordinates, with inverse

$$t_1(x_1, x_2) = -1 + 2\,\frac{\theta - \theta_1}{\theta_2 - \theta_1}\,,\quad t_2(x_1, x_2) = -1 + 2\,\frac{\rho - \rho_1(\theta)}{\rho_2(\theta) - \rho_1(\theta)}\,, \tag{3.11}$$

where

$$\rho = \rho(x_1, x_2) = \sqrt{x_1^2 + x_2^2}\,,\quad \theta = \theta(x_1, x_2) = \arctan(x_2/x_1)\,.$$

Observe that in practice we interpolate the function $F(\rho, \theta) = f(\rho\cos\theta, \rho\sin\theta)$ at Xu-like points on a generalized rectangle in polar coordinates. The special case of the origin is managed by choosing $\theta(0, 0) = 0$, while the angles where $\rho_1 = \rho_2$ are treated as above. Again, the regularity of the transformation is determined by the functions ρ_1 and ρ_2.

The simplest case is that of a disk of radius r centered at the origin, i.e. $0 \le \theta \le 2\pi$, $0 \le \rho \le r$. Notice that the transformation is analytic in this case, whereas that corresponding to the disk represented directly in Cartesian coordinates is not even C^1, since we have $\phi(x_1) = -\sqrt{r^2 - x_1^2}$, $\psi(x_1) = \sqrt{r^2 - x_1^2}$, which are Hölder-continuous but have singular derivatives at $x_1 = \pm r$. We stress this fact by showing Table 3.3, which illustrates the importance of choosing the right transformation. Notice that, whereas the function is extremely smooth, the choice of representing the unit disk in Cartesian coordinates leads to computational failure, since the singularity of the transformation entails very slow convergence.

3.3. Starlike domains in polar coordinates. An important subclass of generalized sectors is given by starlike domains around a given center, i.e., up to a translation,

$$K = \{\mathbf{x} = (\rho\cos\theta, \rho\sin\theta) : \ 0 \le \theta \le 2\,\pi\,,\ 0 \le \rho \le r(\theta)\}\ , \tag{3.12}$$

Here a different transformation can be defined, which allows a better (more symmetric) distribution in comparison to standard polar coordinates of the Xu-like points, which now cluster at both $\theta = 0$ and at $\theta = \pi$ instead of only at $\theta = 0$, and do not cluster at the origin. This is obtained using diameters instead of rays, by varying the angle θ in $[0, \pi]$ and by allowing negative values of ρ, in the following way

$$K = \{\mathbf{x} = (\rho\cos\theta, \rho\sin\theta) : \ 0 \le \theta \le \pi\,,\ -r(\theta+\pi) \le \rho \le r(\theta)\}\ . \tag{3.13}$$

We note that these nonstandard polar coordinates are used, for example, with pseudospectral methods on the disk [11]. Now, we have a different generalized rectangle in (ρ, θ) coordinates. The transformation σ in (3.1) is defined via

$$\theta(t_1,t_2) = \frac{\pi}{2}\,(t_1+1)\ ,\quad \rho(t_1,t_2) = (t_2+1)\,\frac{r(\theta)+r(\theta+\pi)}{2} - r(\theta+\pi)\ , \tag{3.14}$$

and its "inverse" by

$$t_1(x_1,x_2) = -1 + \frac{2\,\theta}{\pi}\ ,\quad t_2(x_1,x_2) = -1 + 2\,\frac{\rho - r(\theta+\pi)}{r(\theta)+r(\theta+\pi)}\ , \tag{3.15}$$

where

$$\rho = \rho(x_1,x_2) = \mathrm{sign}(x_2)\,\sqrt{x_1^2+x_2^2}\ ,\quad \theta = \theta(x_1,x_2) = \arctan\,(x_2/x_1)\ .$$

See Fig. 3.2 for a comparison of distributions of Xu-like points in the case of the unit disk, and Fig. 3.3 for Xu-like points in starlike-polar coordinates, in the case of the cardioid $r(\theta) = (1-\cos\theta)/2$, and of the "four-leaf clover" $r(\theta) = \cos 2\theta$.

The advantage of using the transformation (3.14) for a starlike domain ("starlike-polar" coordinates) instead of standard polar coordinates, is illustrated by Table 3.3. In Tables 3.4–3.5, we give the interpolation errors for two functions with different degrees of regularity on the domains of Fig. 3.3, at a sequence of Xu-like nodal sets.

4. 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; see [14, 15] and references therein.

Here, we adopt a very simple global strategy, based on the fact that the information content of a sufficiently regular function on one of the domains described in the previous sections, is contained, up to the interpolation error, in its values at relatively few Xu points.

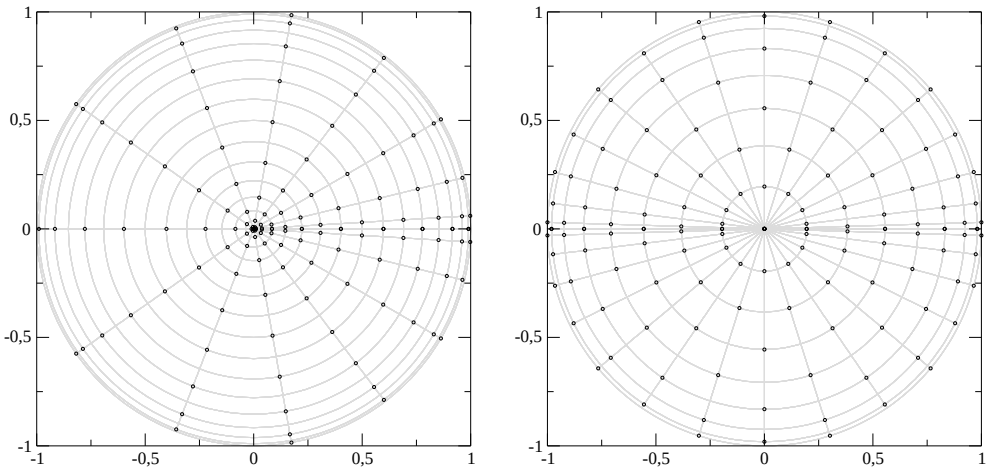

FIG. 3.2. *The distribution of $N = 144$ Xu-like points (degree $n = 16$) in the unit disk in polar (left) and starlike-polar (right) coordinates.*

TABLE 3.3

Xu-like interpolation errors of $f(x_1, x_2) = \cos(x_1 + x_2)$ on the unit disk in Cartesian, standard polar and "starlike polar" coordinates; n is the underlying polynomial degree, $N = n(n+2)/2$ the corresponding number of Xu-like interpolation points (3.4).

n	8	16	24	32	40
N	40	144	312	544	840
Cartesian	6E-2	2E-2	6E-3	3E-3	4E-3
polar	1E-1	3E-3	2E-5	1E-7	3E-10
starlike	1E-2	1E-5	4E-9	5E-13	2E-14

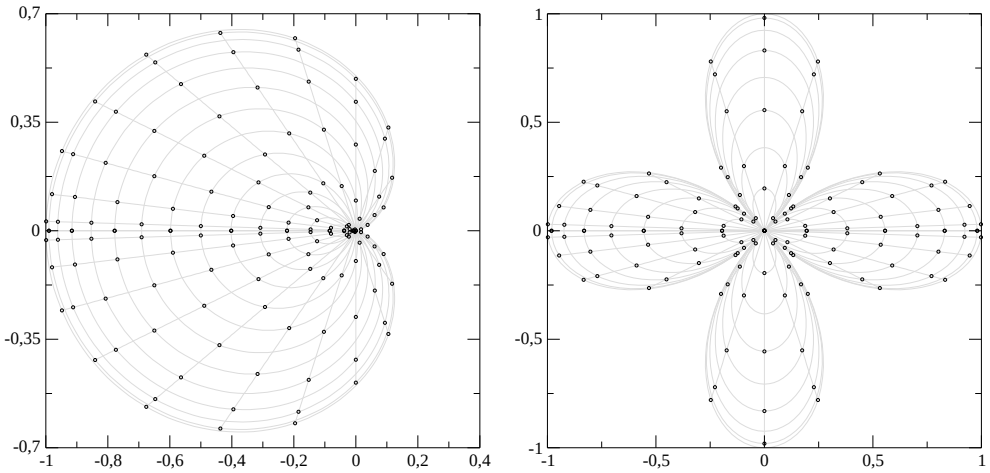

FIG. 3.3. *The distribution of $N = 144$ Xu-like points (degree $n = 16$) in starlike-polar coordinates: cardioid (left), four-leaf clover (right).*

Since the function is assumed to be known only as a large scattered data set, its values at Xu points have to be computed through an auxiliary function.

In our application, we have chosen the cubic Shepard-like interpolant implemented in the

TABLE 3.4
Xu-like interpolation errors of $f(x_1, x_2) = \cos(x_1 + x_2)$ on the cardioid and the four-leaf clover of Fig. 3.3; n is the underlying polynomial degree, $N = n(n+2)/2$ the corresponding number of Xu-like interpolation points (3.4).

n	8	16	24	32	40
N	40	144	312	544	840
cardioid	2E-2	3E-5	3E-8	1E-11	5E-14
4-leaf	2E-1	1E-2	9E-4	1E-5	8E-7

TABLE 3.5
As in Table 3.4 for the function $f(x_1, x_2) = (x_1^2 + x_2^2)^{5/2}$.

n	8	16	24	32	40
N	40	144	312	544	840
cardioid	1E-2	1E-4	2E-5	3E-6	1E-6
4-leaf	4E-1	7E-2	1E-3	2E-4	5E-5

ACM Algorithm 790 (CSHEP2D) by R.J. Renka [19]. As is known, for reasonably dense data sets CSHEP2D is among the most accurate and efficient scattered data algorithms available [20]. It constructs a C^2 interpolant in a moving least-square fashion [16, 23], with a mean complexity which is linear in the cardinality of the data for the preprocessing stage (computation of the parameters defining the interpolant), and $\mathcal{O}(1)$ for each pointwise evaluation (the basis functions being locally supported).

In practice, the surface compression algorithm (interpolated interpolation) can be summarized as follows:

- Encoding stage: construction of the parameters defining the Shepard-like interpolant, say $S(\mathbf{x})$, by subroutine CSHEP2 of ACM 790; evaluation of $S(\mathbf{x})$ at a sequence of Xu nodal sets by subroutine CS2VAL of ACM 790, testing the reconstruction error of the Xu-like interpolant on the original data set, until such an error goes below a given tolerance or stagnates, or the compression ratio becomes unacceptable. The array of values $\{S(\mathbf{x}_{r,s})\}$ at the resulting Xu nodal set represents the compressed surface, and is accompanied by an estimated compression error.
- Decoding and reconstruction stage: simply the evaluation of the Xu-like interpolant $\mathcal{L}_n^{\mathrm{Xu}} S$ at any given set of target points, needed for the specific application (e.g., plotting).

Notice that the Xu nodal set is completely known once the domain and the degree are given, and thus there is no need to store and transmit the array of Xu points (while the original scattered data set is an array of 3D points). This means that the compression ratio is given by

$$\text{compr. ratio} \;=\; 3 \times \frac{\text{numb. of scatt. pts.}}{\text{numb. of Xu nodes}} \;\approx\; 6 \times \frac{\text{numb. of scatt. pts.}}{n^2}\,, \tag{4.1}$$

where n is the underlying polynomial degree.

4.1. Example 1: Compression of test functions on the unit square. For the purpose of illustration, in Tables 4.1–4.3 we report the compression errors obtained by Xu interpolation of the quoted Shepard-like interpolant, on the sampling of three test functions at a sequence of large randomly generated point sets (from 5000 up to 40000 points). Since the domain is the unit square, the Xu interpolants are polynomials, constructed at a sequence of degrees. The last column shows the actual errors made by the Shepard-like interpolant on the underlying function, whereas the last row displays the actual errors made by direct Xu interpolation of the underlying function. In Table 4.4 we show the compression ratios, computed as in (4.1)

and rounded to the nearest integer, corresponding to the sequences of scattered point sets and of interpolation degrees above.

Observe that the rows tend to stabilize around the underlying Shepard interpolation error, whereas the columns around the the underlying Xu interpolation error. This can be easily explained by splitting the compression error as

$$S(\mathbf{x}) - L_n^{\mathrm{Xu}} S(\mathbf{x}) = \{S(\mathbf{x}) - f(\mathbf{x})\} + \{f(\mathbf{x}) - L_n^{\mathrm{Xu}} f(\mathbf{x})\} + L_n^{\mathrm{Xu}}(f - S)(\mathbf{x}) \ , \tag{4.2}$$

where $S(\mathbf{x})$ is the Shepard-like interpolant [19] on the scattered point set. Recall that $S(\mathbf{x})$ has only C^2 regularity, and thus quite slow convergence of its Xu interpolants could be expected, in view of (2.18). Nevertheless, the Lebesgue constant of Xu interpolation increases very slowly (cf. (2.15), (2.16)), hence from the splitting (4.2) we can expect an initial convergence stage driven by f, followed by a stagnation around the Shepard interpolation error, as the degree n increases. On the other hand, for the same reasons, increasing the data density for a fixed n, it is natural that the error stagnates around the Xu interpolation error of the underlying function, when this error becomes dominant.

It is worth noticing that, if one is satisfied with an error in the max-norm below 0.1%, which can be considered more than acceptable in many practical applications (e.g., quality plotting), the compression ratios corresponding to the largest scattered data sets are on the order of the hundreds (see Table 4.4).

TABLE 4.1

Compression errors (in the max-norm) for the Franke test function on $[0,1]^2$, sampled at a sequence of scattered (randomly generated) point sets, by Xu interpolation of a cubic Shepard-like interpolant [19]; last row and column: actual errors of the Xu and Shepard-like interpolants on the test function.

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

TABLE 4.2

As in Table 4.1 for the "waterfall" shaped test function $f(x_1, x_2) = (\tanh(9x_2 - 9x_1) + 1)/9$ taken from the testset in [12].

random pts.	$n = 16$	$n = 24$	$n = 32$	$n = 40$	$n = 48$	"true" Shep.
5000	9E-3	2E-3	6E-4	2E-4	3E-4	5E-4
10000	9E-3	2E-3	5E-4	1E-4	6E-5	7E-5
20000	9E-3	2E-3	5E-4	1E-4	3E-5	2E-5
40000	9E-3	2E-3	5E-4	1E-4	3E-5	9E-6
"true" Xu	9E-3	2E-3	5E-4	1E-4	3E-5	

4.2. Example 2: Compression of a Finite Element PDE solution. Another interesting application of compression of regular surfaces via "interpolated interpolations", arises for example within the numerical solution of elliptic PDEs by Finite Elements on large-scale meshes.

Again for the only purpose of illustration, we consider the following Poisson equation with Dirichlet boundary conditions

$$\begin{cases} \Delta f(\mathbf{x}) = -10 \ , & \mathbf{x} \in \Omega \\ f(\mathbf{x}) = 0 \ , & \mathbf{x} \in \partial\Omega \end{cases} \tag{4.3}$$

TABLE 4.3

As in Table 4.1 for the oscillating test function $f(x_1, x_2) = 2\cos(10x_1)\sin(10x_2) + \sin(10x_1x_2)$ *taken from the testset in [20].*

random pts.	$n = 16$	$n = 24$	$n = 32$	$n = 40$	$n = 48$	"true" Shep.
5000	4E-3	8E-4	1E-3	1E-3	2E-3	3E-3
10000	4E-3	1E-3	1E-3	1E-3	8E-4	2E-3
20000	4E-3	2E-4	2E-4	2E-4	2E-4	3E-4
40000	4E-3	3E-5	3E-5	5E-5	3E-5	9E-5
"true" Xu	4E-3	1E-7	2E-13	1E-14	3E-14	

TABLE 4.4

The compression ratios in (4.1) (rounded to the nearest integer), corresponding to Tables 4.1–4.3.

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

where Ω is the "lynx-eye" shaped domain, given by a horizontal elliptical domain with a vertical elliptical hole; see Fig. 4.1. Both the ellipses are centered at the origin, and have semi-axes $a = 1$ and $b = 0.5$ (external), $a = 0.2$ and $b = 0.4$ (internal). The numerical solution has been computed by a standard Galerkin Finite Element discretization with linear basis functions, on a Delaunay mesh with 81796 triangular elements, 41402 nodes, and mesh parameter (maximum triangles side) $h = 0.013$; see Fig. 4.1 (right) for a detail of the mesh. The resulting linear system has been solved in a standard way, by the Conjugate Gradient method preconditioned with incomplete Cholesky factorization [21].

Observe that $K = \overline{\Omega}$ (which is not simply-connected) is a generalized sector (see Section 3.2), with boundaries defined by the polar equation of the ellipses

$$\rho(\theta) = ab\sqrt{\frac{1+\tan^2\theta}{b^2+a^2\tan^2\theta}}\ , \quad 0 \le \theta \le 2\pi\ , \tag{4.4}$$

where a and b are the horizontal and vertical semi-axes, respectively; the distribution of $N = 312$ Xu-like points in K (degree $n = 24$), is shown in Fig. 4.1 (left).

TABLE 4.5

Compression errors (in the max-norm) for the Finite Element solution of the Poisson equation (4.3), by Xu-like interpolation of a cubic Shepard-like interpolant [19].

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

In Table 4.5 we report the compression errors, corresponding to Xu-like interpolation of the Shepard-like interpolant quoted above, at a sequence of degrees. Notice the expected stagnation of the error around the size of the Finite Element discretization error. It is worth stressing that we are then able to compress the $3 \times 41402 = 124206$ Finite Element solution data into the array of $N = 24 \times 13 = 312$ values at Xu-like points ($n = 24$), which means a compression ratio of about 400:1. See Fig. 4.2 for a plot of the Xu-like interpolated solution at degree $n = 24$.

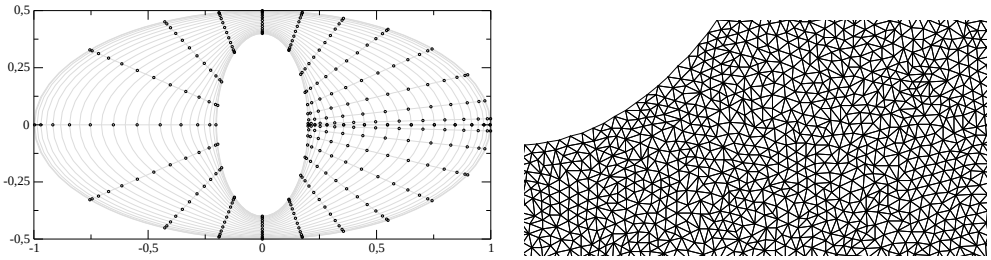

FIG. 4.1. *The distribution of $N = 312$ Xu-like points (degree $n = 24$) in the "lynx-eye" shaped domain of the Poisson equation (4.3) (left), and a detail of the Finite Element mesh in the domain above, near the internal boundary (right).*

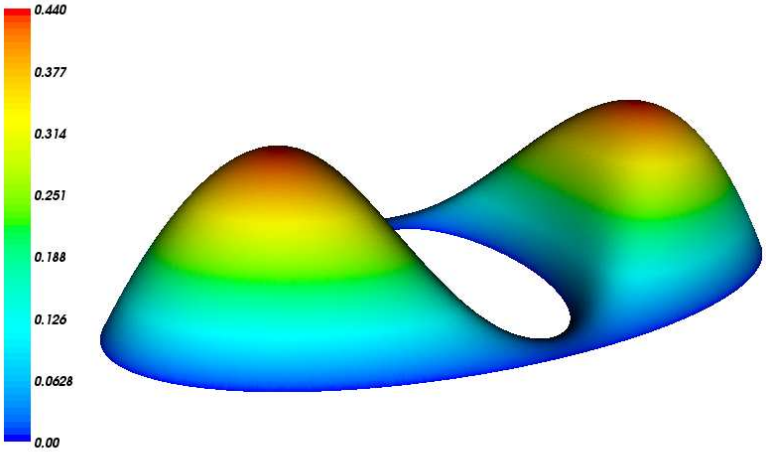

FIG. 4.2. *Plot of the Xu-like interpolated solution at degree $n = 24$ (compression ratio $\approx 400 : 1$, compression error $\approx 2 \cdot 10^{-3}$).*

REFERENCES

[1] T. BAGBY, L. BOS, AND N. LEVENBERG, *Multivariate simultaneous approximation*, Constr. Approx., 18 (2002), pp. 569–577.

[2] L. BOS, *On certain configurations of points in R^n which are unisolvent for polynomial interpolation*, J. Approx. Theory, 64 (1991), pp. 271–280.

[3] L. BOS, M. CALIARI, S. DE MARCHI AND M. VIANELLO, *A numerical study of the Xu polynomial interpolation formula*, Computing, 76(3-4) (2005), pp. 311–324.

[4] L. BOS, S. DE MARCHI, AND M. VIANELLO, *On the Lebesgue constant for the Xu interpolation formula*, J. Approx. Theory, available online 17 April 2006.

[5] L. BOS, N. LEVENBERG, AND S. WALDRON, *Metrics associated to multivariate polynomial inequalities*, in Advances in constructive approximation: Vanderbilt 2003, Mod. Methods Math., Nashboro Press, Brentwood, TN, 2004, pp. 133–147,

[6] B. BOJANOV AND Y. XU, *On polynomial interpolation of two variables*, J. Approx. Theory, 120 (2003), pp. 267–282.

[7] L. BRUTMAN, *Lebesgue functions for polynomial interpolation - a survey*, Ann. Numer. Math., 4 (1997), pp. 111–127.

[8] M. CALIARI, S. DE MARCHI AND M. VIANELLO, *Bivariate polynomial interpolation on the square at new nodal sets*, Appl. Math. Comput., published online, October 2004.

[9] M. CALIARI, S. DE MARCHI AND M. VIANELLO, *Xi2: a numerical code for Xu-like bivariate polynomial interpolation*, preliminary version, September 2004; http://prof.sci.univr.it/$\sim$caliari/software.htm.

[10] M. DUBINER, *The theory of multi-dimensional polynomial approximation*, J. Anal. Math., 67 (1995), pp. 39–116.
[11] B. FORNBERG, *A Practical Guide to Pseudospectral Methods*, Cambridge Monographs on Applied and Computational Mathematics, Vol. 1, Cambridge University Press, Cambridge, 1996.
[12] R. FRANKE, *Scattered data interpolation: Test of some methods*, Math. Comp., 38 (1982), pp. 181–200.
[13] M. GASCA AND T. SAUER, *Polynomial interpolation in several variables*, Adv. Comput. Math., 12 (2000), pp. 377–410.
[14] D. HONG AND L. L. SCHUMAKER, *Surface compression using a space of C^1 cubic splines with a hierarchical basis*, Computing, 72 (2004), pp. 79–92.
[15] A. ISKE, *Multiresolution Methods in Scattered Data Modelling*, Lecture Notes in Computational Science and Engineering, Vol. 37, Springer, 2004.
[16] D. LEVIN, *The approximation power of moving least squares*, Math. Comp., 67 (1998), pp. 1517–1531.
[17] C. R. MORROW AND T. N. L. PATTERSON, *Construction of algebraic cubature rules using polynomial ideal theory*, SIAM J. Numer. Anal., 15 (1978), pp. 953–976.
[18] M. REIMER, *Multivariate Polynomial Approximation*, International Series of Numerical Mathematics, Vol. 144, Birkh¨auser, 2003.
[19] R.J. RENKA, *Algorithm 790: CSHEP2D: cubic shepard Method for bivariate interpolation of scattered data*, ACM Trans. Math. Software, 25 (1999), pp. 70–73.
[20] R.J. RENKA AND R. BROWN, *Algorithm 792: accuracy tests of ACM algorithms for interpolation of scattered data in the plane*, ACM Trans. Math. Software, 25 (1999), pp. 78–94.
[21] Y. SAAD, *Iterative methods for sparse linear systems*. 2nd edition. SIAM, Philadelphia, 2003.
[22] A. SOMMARIVA, M. VIANELLO, AND R. ZANOVELLO, *Adaptive bivariate Chebyshev approximation*, Numer. Algorithms, 38 (2005), pp. 79–94.
[23] H. WENDLAND, *Scattered Data Approximation*, Cambridge Monographs on Applied and Computational Mathematics, Vol. 17, Cambridge University Press, Cambridge, 2005.
[24] Y. XU, *Gaussian cubature and bivariate polynomial interpolation*, Math. Comp., 59 (1992), pp. 547–555.
[25] Y. XU, *Lagrange interpolation on Chebyshev points of two variables*, J. Approx. Theory, 87 (1996), pp. 220–238.