

Meshless cubature by Green's formula ^{*}

Alvise Sommariva and Marco Vianello [†]

Department of Pure and Applied Mathematics

University of Padova

via Belzoni 7, 35131 - Padova (Italy)

Abstract

By coupling the flexibility of Thin-Plate Splines interpolation with Green's integral formula, we obtain a meshless cubature method from scattered samples of small/moderate size, on convex as well as nonconvex and even multiply connected polygonal domains.

2000 AMS subject classification: 65D05, 65D32.

1 Introduction.

Green's integral formula [8, 1], which in one of its formulations is written as

$$\int_{\Omega} f(P) dP = \oint_{\partial\Omega} \mathcal{F}(P) dy, \quad f = \frac{\partial \mathcal{F}}{\partial x}, \quad P = (x, y), \quad (1)$$

gives in principle an appealing tool for numerical cubature, since it transforms a 2-dimensional into a 1-dimensional problem. Its practical use, however, requires the knowledge of a primitive of the integrand, which seems to restrict the field of application to a subclass of analytically known functions. In this paper, we try to exploit the potentialities of Green's formula via RBF (Radial Basis Functions) interpolation, obtaining simultaneously the capability to work with scattered samples and to manage integration on "difficult" geometries, in a *meshless* fashion.

The general basic idea is simple. Given an interpolation/approximation of a continuous function f on a bounded bivariate domain Ω with piecewise smooth boundary, in some function space $span(\phi_1, \dots, \phi_n)$,

$$\sum_{j=1}^n c_j \phi_j(P) = s(P) \approx f(P), \quad (2)$$

^{*}Work supported by the "ex-60%" funds of the University of Padova, by the INdAM GNCS, and by the ARC MASCOS (research assistantship of A. Sommariva at the School of Mathematics, UNSW Sydney, Australia, 2004-2005).

[†]Corresponding author: e-mail: marcov@math.unipd.it

we can approximate the integral of f on Ω via Green's formula as

$$\int_{\Omega} f(P) dP \approx \int_{\Omega} s(P) dP = \sum_{j=1}^n c_j \int_{\Omega} \phi_j(P) dP = \sum_{j=1}^n c_j \oint_{\partial\Omega} \Phi_j(P) dy, \quad (3)$$

provided that the primitives $\Phi_j(P) = \int \phi_j(P) dx$ are explicitly computable and easily integrated along the boundary $\partial\Omega$.

Dealing with scattered data, polynomials (which would have this latter features) are unsuitable, whereas piecewise polynomial functions require the generation and managing of a mesh. In order to obtain a truly meshless cubature formula, RBF, MLS (Moving Least Squares) and PU (Partition of Unity) methods, see e.g. [16], are potentially attractive. However, typical MLS and PU basis functions seem difficult to integrate explicitly, especially if a second level primitive along the boundary is sought. On the contrary, first level primitives of most classical radial basis functions are easily computed, and with TPS (Thin-Plate Splines, which are a good choice for cubature, cf. [14]), even second level primitives along piecewise linear boundaries (arbitrary polygons) are explicitly computable via symbolic integration tools.

In the next section we give a brief survey on cubature by RBF, and in section 3 we show how to construct TPS cubature formulas via Green's formula and symbolic integration. In section 4, finally, we test TPS cubature with scattered data on nonconvex polygons, and we show how to improve efficiency by data splitting.

2 A survey of RBF cubature.

In a recent paper [14], we ventured the little explored territory of numerical cubature from scattered data by RBF (this approach has then been extended to integration over the sphere in [15]). Indeed, the problem itself of cubature on scattered points has received much less attention in the numerical literature, with respect to construction of cubature formulas on nodes with a predefined distribution; see, e.g., [3, 6, 10]. On the other hand, RBF give a well studied and effective tool for the reconstruction of functions from scattered data: we refer the reader to the recent monographs [5, 9, 16] and to the survey papers [4, 13].

Consider the problem of evaluating the integral of a continuous function f on a bivariate compact domain Ω

$$I(f) = \int_{\Omega} f(P) dP, \quad \Omega \subset \mathbb{R}^2, \quad (4)$$

from a *scattered* sample of size n

$$\mathbf{f} = \{f(P_i)\} \text{ at } X = \{P_i\} = \{(x_i, y_i)\} \subset \Omega, \quad i = 1, \dots, n. \quad (5)$$

As it is well-known, given a suitable radial function $\phi(r)$, $\phi : [0, +\infty) \rightarrow \mathbb{R}$, we can construct the RBF interpolant $s(P)$ in the form (2), where $s(P_i) = f(P_i)$,

$i = 1, \dots, n$, as a linear combination of (scaled) translates of $\phi(|\cdot|)$

$$\phi_j(P) = \phi_j(P; \delta) = \phi(|P - P_j|/\delta), \quad (6)$$

by solving the linear system

$$\mathbf{A}\mathbf{c} = \mathbf{f}, \quad \mathbf{A} = A_{X,\phi} = [\phi_j(P_i)], \quad 1 \leq i, j \leq n, \quad (7)$$

in the case that the function ϕ is *positive definite*, i.e., the collocation matrix A is *positive definite* for every occurrence of the interpolation nodes. In (6), $|\cdot|$ denotes the euclidean norm in \mathbb{R}^2 , and δ a scaling parameter (which can be related to the data density).

When the radial function is only *conditionally positive definite* (of order $m + 1$), like e.g.

- Duchon's Thin-Plate Splines (TPS): $\phi(r) = r^2 \log(r)$

(which are a good choice for cubature, cf. [14]), as it is well known the RBF interpolant is sought in the form

$$s(P) = \sum_{j=1}^n c_j \phi_j(P) + \pi(P), \quad (8)$$

where $\pi \in \mathbb{P}_m^2$ is a suitable bivariate polynomial of degree $\leq m$ ($m = 1$ for TPS). The coefficients of the RBF and of the polynomial term are then obtained by solving the augmented system of dimension $n + \mu$, $\mu = (m + 1)(m + 2)/2$,

$$\begin{aligned} \mathcal{A}\mathbf{C} = \mathbf{F}, \quad \text{with } \mathcal{A} = \begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix}, \quad B = [\pi_k(P_i)], \\ \mathbf{C} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}, \end{aligned} \quad (9)$$

cf. (5), where $\{\pi_1, \dots, \pi_\mu\}$ is a basis of \mathbb{P}_m^2 .

An *interpolatory cubature formula* is immediately obtained by

$$I(f) \approx I(s) = \sum_{j=1}^n c_j I(\phi_j) + I(\pi), \quad (10)$$

which can be rewritten in vector form as

$$I(s) = \langle \mathbf{C}, \mathbf{I} \rangle, \quad \mathbf{I} = \begin{bmatrix} \mathbf{I}_R \\ \mathbf{I}_\pi \end{bmatrix} \quad \text{with } \mathbf{I}_R = \{I(\phi_j)\}, \quad \mathbf{I}_\pi = \{I(\pi_k)\}. \quad (11)$$

Recalling that $\mathbf{C} = \mathcal{A}^{-1} \mathbf{F}$ and the symmetry of the system matrix \mathcal{A} , we get the usual form of weighted sum of the sampling values (here $\langle \cdot, \cdot \rangle$ denotes the scalar product in the corresponding dimension)

$$I(s) = \langle \mathcal{A}^{-1} \mathbf{F}, \mathbf{I} \rangle = \langle \mathcal{A}^{-1} \mathbf{I}, \mathbf{F} \rangle = \langle \mathbf{W}, \mathbf{F} \rangle = \langle \mathbf{w}, \mathbf{f} \rangle = \sum_{j=1}^n w_j f_j, \quad (12)$$

where the weights are obtained as solution of the linear system

$$\mathcal{A} \mathbf{W} = \mathbf{I}, \quad \mathbf{W} = \begin{bmatrix} \mathbf{w} \\ \mathbf{z} \end{bmatrix} \quad (\text{weights equations}). \quad (13)$$

The derivation of the cubature formula in the positive definite case is completely analogous, with small substituting capital letters for the vectors, and the original collocation matrix A substituting the augmented matrix \mathcal{A} .

In [14], we gave some convergence and stability estimates for RBF cubature, where two basic parameters in RBF interpolation appear, the *fill distance* (the radius of the largest inner empty disk) and the *separation distance* of the cubature points

- fill distance: $h = \max_{P \in \Omega} \min_{1 \leq j \leq n} |P - P_j|$
- separation distance: $q = \min_{i \neq j} \{|P_i - P_j|\} \leq 2h$

Here we report only one of such estimates, which tries to take into account also the effect of the errors made in evaluating the integrals of the RBF (we use $\tilde{\mathbf{I}} \approx \mathbf{I}$ and compute perturbed weights $\tilde{\mathbf{W}}$ and $\tilde{\mathbf{w}}$ by (13)), as well as of possible noise in the sample, $\tilde{\mathbf{f}} \approx \mathbf{f}$. Such an estimate is

$$\begin{aligned} |I(f) - \langle \tilde{\mathbf{w}}, \tilde{\mathbf{f}} \rangle| &\leq |I(f) - I(s)| + |\langle \mathbf{W} - \tilde{\mathbf{W}}, \mathbf{F} \rangle| + |\langle \tilde{\mathbf{W}}, \mathbf{F} - \tilde{\mathbf{F}} \rangle| \\ &\leq \sqrt{\text{meas}(\Omega)} \|f - s\|_{L^2(\Omega)} + \|\mathcal{A}^{-1}\|_2 \|\mathbf{f}\|_2 \|\mathbf{I} - \tilde{\mathbf{I}}\|_2 + \|\mathbf{f} - \tilde{\mathbf{f}}\|_\infty \|\tilde{\mathbf{w}}\|_1 \\ &= \mathcal{O}\left(\sqrt{\mathcal{F}_\phi(h)}\right) + \mathcal{O}\left(\frac{1}{\mathcal{G}_\phi(q)}\right) \|\mathbf{I} - \tilde{\mathbf{I}}\|_2 + \|\mathbf{f} - \tilde{\mathbf{f}}\|_\infty \sum_{j=1}^n |\tilde{w}_j|, \end{aligned} \quad (14)$$

where $\mathcal{F}_\phi(h) \downarrow 0$ as $h \rightarrow 0$ and $\mathcal{G}_\phi(q) \downarrow 0$ as $q \rightarrow 0$, and thus as $h \rightarrow 0$, see [5, 9, 16] (in the positive definite case, (14) holds again with small substituting capital letters, and A substituting \mathcal{A}). We recall that, technically, this estimate is valid for functions belonging to the so-called “native space” of the radial function ϕ . A more complete error analysis, which discusses also the connection of RBF cubature formulas with optimal recovery in the corresponding reproducing kernel Hilbert spaces, can be found in [14].

The simultaneous appearance of the terms $\sqrt{\mathcal{F}_\phi(h)} \downarrow 0$ and $1/\mathcal{G}_\phi(q) \uparrow +\infty$ in (14) is an occurrence, in the context of numerical cubature, of the well-known “*uncertainty principle*” in RBF interpolation, which can be summarized directly with R. Schaback’s words [12]: “*There is no case known where the errors and the sensitivity are both reasonably small*”. In practice, however, there are two distinct situations: for *smooth* RBF, like Gaussians and (inverse) multiquadrics, the rates of $\mathcal{F}_\phi(h)$ and $\mathcal{G}_\phi(q)$ are both *exponential*, while for *less regular* RBF, like TPS, the rates are both *algebraic*.

The situation seems hopeless concerning the use of smooth RBF like Gaussians for numerical cubature, in view of the expected exponential magnification of the integration errors, especially with scattered data where typically $q \ll h$. However, the numerical experiments in [14] have shown that (14) is largely an

Table 1: RBF cubature with sets of $n = 50$ and $n = 100$ uniform random points in $[0, 1]^2$: spectral norm of the inverses of the collocation matrices and 1-norm of the computed weights vectors (average values on 50 independent trials, rounded to the first significant digit).

# of rnd pts	norms	<i>MQ</i>	<i>IMQ</i>	<i>G</i>	<i>W2</i>	<i>TPS</i>
$n = 50$	$\ \mathcal{A}^{-1}\ _2$	2E+12	3E+11	5E+15	5E+03	6E+03
	$\ \tilde{\mathbf{w}}\ _1$	7E+01	9E+01	3E+02	2E+00	1E+00
$n = 100$	$\ \mathcal{A}^{-1}\ _2$	2E+16	6E+15	3E+17	5E+04	8E+03
	$\ \tilde{\mathbf{w}}\ _1$	8E+02	5E+02	1E+03	2E+00	1E+00

overestimate. In any case, it turns out that smooth RBF are much more sensible to integration errors, increase of data density and perturbations in the data, while TPS give reasonably accurate and much more stable cubature formulas. Moreover, the quality of cubature by TPS is weakly sensible to the scaling parameter, a phenomenon which can be related to the well-known scale independence of the condition number of interpolation by polyharmonic splines (cf. [9]). Clearly, this property makes TPS cubature very attractive for automatic integration, since it avoids the complication of managing/optimizing the scaling parameter.

In order to appreciate the possible magnitude of the important quantities related to stability of the cubature formulas, in Table 1 we report from [14] the values of $\|\mathcal{A}^{-1}\|_2$ (where \mathcal{A} is the augmented collocation matrix for conditionally positive definite ϕ and $\mathcal{A} = A$ in the positive definite case), and of $\|\tilde{\mathbf{w}}\|_1 = \sum_{j=1}^n |\tilde{w}_j|$, for (inverse) multiquadrics (MQ and IMQ), Gaussians (G), Wendland's compactly supported (W2) and thin-plate splines (TPS). For simplicity, we consider only unscaled RBF ($\delta = 1$).

Now, even using TPS or W2 for the construction of cubature formulas, it is intuitable from the table above that the numerical approximations of the integrals of the RBF, $\tilde{I}(\phi_j) \approx I(\phi_j)$ (and of the polynomial part for TPS) have to be computed with high accuracy, since a substantial loss of precision occurs. Since the intensive use of a numerical integrator at high precision can become a bottleneck of the implementation, in [14] we have derived *explicit formulas* for the integrals of TPS and W2 on the unit square $\Omega = [0, 1]^2$ (which we do not report for brevity), by exploiting radial symmetry and *symbolic integration*.

Albeit such formulas could be extended from the square to general convex polygons, in the next section we show that with TPS a much simpler approach is given by Green's formula, which leads naturally via symbolic integration to TPS cubature formulas on *arbitrary polygons*.

3 Implementing Green's formula for TPS.

We focus now on the construction of cubature formulas by TPS, and in particular on the evaluation of the integrals of the basis functions that are required in the weights equations (13). All the symbolic integrations below have been performed by the effective online integrator at [17].

From (2)-(6), it is clear that the core of the Green's formula approach is given by computing

$$\int_{\Omega} \phi(|P - Q|) dP = \oint_{\partial\Omega} \left(\int \phi(|P - Q|) dx \right) dy, \quad P = (x, y), \quad Q = (u, v), \quad (15)$$

as a function of the point Q . We have put the scaling parameter $\delta = 1$ for notation simplicity, but with TPS, as already observed, this is not really restrictive. In the case of TPS the x -primitive above is (since $\phi(r) = r^2 \log r$)

$$\begin{aligned} \Phi_Q(P) := \int \phi(|P - Q|) dx &= \frac{1}{9} (u-x)^3 + \frac{2}{3} (u-x)(v-y)^2 - \frac{2}{3} (v-y)^3 \arctan\left(\frac{u-x}{v-y}\right) \\ &\quad - \frac{1}{6} (u-x)((u-x)^2 + 3(v-y)^2) \log((u-x)^2 + (v-y)^2). \end{aligned} \quad (16)$$

Now, suppose that the domain Ω is a *polygon* (convex or nonconvex, but simply connected), whose boundary is described *counterclockwise* by a sequence of vertices $V_j = (\xi_j, \nu_j)$, $j = 1, \dots, p$ with $p \geq 3$,

$$\partial\Omega = [V_1, V_2] \cup [V_2, V_3] \cup \dots \cup [V_p, V_{p+1}], \quad V_{p+1} = V_1. \quad (17)$$

Then, we get the integral (15) as a sum of line integrals along the sides

$$\begin{aligned} \int_{\Omega} \phi(|P - Q|) dP &= \sum_{j=1}^p \int_{[V_j, V_{j+1}]} \Phi_Q(P) dy \\ &= \sum_{\xi_j \neq \xi_{j+1}} \frac{\Delta\nu_j}{\Delta\xi_j} \int_{\xi_j}^{\xi_{j+1}} \Phi_Q\left(x, \frac{\Delta\nu_j}{\Delta\xi_j} x + \nu_j\right) dx + \sum_{\xi_j = \xi_{j+1}} \int_{\nu_j}^{\nu_{j+1}} \Phi_Q(\xi_j, y) dy, \end{aligned} \quad (18)$$

where Δ denotes the forward difference operator. Assume that the side $[V_j, V_{j+1}]$ is not parallel to the y -axis. By putting $u - x = t$, so that $v - y = at + b$ along the side for $a = -\Delta\nu_j/\Delta\xi_j$ and $b = \nu_j - v + (u - \xi_j)\Delta\nu_j/\Delta\xi_j$, the problem is eventually reduced to computing explicitly a *second level primitive* like

$$\int \Phi_Q(u - t, v - at - b) dt = A(t) + B(t) + C(t), \quad (19)$$

where by [17] we get

$$A(t) = \int \left(\frac{1}{9} t^3 + \frac{2}{3} t(at + b)^2 \right) dt = \frac{t^2}{9} \left(3b^2 + 4abt + (6a^2 + 1) \frac{t^2}{4} \right), \quad (20)$$

$$B(t) = -\frac{2}{3} \int (at + b)^3 \arctan\left(\frac{t}{at + b}\right) dt = \sum_{i=1}^4 B_i(t), \quad (21)$$

with

$$B_1(t) = -\frac{t}{6} (4b^3 + 6ab^2t + 4a^2bt^2 + a^3t^3) \arctan\left(\frac{t}{at + b}\right),$$

$$B_2(t) = \frac{abt}{18(1+a^2)^3} (3b^2(6+3a^2+a^4) + 3ab(2+3a^2+a^4)t + a^2(1+a^2)^2t^2),$$

$$B_3(t) = -\frac{b^4(a^2-1)}{3(1+a^2)^4} \log(b^2 + 2abt + (1+a^2)t^2),$$

$$B_4(t) = -\frac{ab^4}{6(1+a^2)^4} (10 + 5a^2 + 4a^4 + a^6) \arctan\left(\frac{ab + (1+a^2)t}{b}\right),$$

and finally

$$C(t) = -\frac{1}{6} \int t(t^2 + 3(at + b)^2) \log(t^2 + (at + b)^2) dt = \sum_{i=1}^4 C_i(t), \quad (22)$$

with

$$C_1(t) = \frac{2\sqrt{3}ab^4}{(1+3a^2)^3} \arctan\left(\frac{3ab + (1+3a^2)t}{\sqrt{3}b}\right),$$

$$C_2(t) = -t \left(\frac{a(5+3a^2)b^3}{3(1+3a^2)^2} + \frac{(1+a^2)b^2}{6(1+3a^2)} t + \frac{5ab}{36} t^2 + \frac{(1+3a^2)}{38} t^3 \right),$$

$$C_3(t) = \frac{9b^4(-1+6a^2+3a^4)}{13(1+3a^2)^3} \log(3b^2 + 6abt + (1+3a^2)t^2),$$

$$C_4(t) = -\frac{t^2}{24} (6b^2 + 8abt + (1+3a^2)t^2) \log(3b^2 + 6abt + (1+3a^2)t^2).$$

When the side $[V_j, V_{j+1}]$ is parallel to the y -axis (i.e., $\xi_j = \xi_{j+1}$, cf. (18)), the problem is completely analogous, since by the substitution $v - y = t$ it is reduced to compute symbolically $\int \Phi_Q(\xi_j, v - t) dt$ (we do not report the formulas, for brevity). Observe also that when the side is parallel to the x -axis, the corresponding integral in (18) is clearly null, since $\Delta\nu_j = 0$. We do not even describe integration of the polynomial part in (10), which is trivial via Green's formula.

Remark 3.1 The TPS cubature formula can be extended easily to *multiply connected* domains, via the corresponding extension of Green's theorem. Indeed, assume that the boundary of Ω be the union of an external boundary Γ^{ext} with a finite number of internal boundaries Γ_k^{int} , $k = 1, \dots, m$ (describing holes). Then we have

$$\int_{\Omega} \phi(|P - Q|) dP = \oint_{\Gamma^{\text{ext}}} \Phi_Q(P) dy - \sum_{k=1}^m \oint_{\Gamma_k^{\text{int}}} \Phi_Q(P) dy, \quad (23)$$

where all line integrals are taken counterclockwise and can be computed as in (18)-(22).

Remark 3.2 It is worth observing that the cubature formulas above can be used also when f has been sampled in a larger domain, $\{P_j\} \subset \Omega'$, $\Omega' \supseteq \Omega$. This could be of some interest in applications, for example when, given a sample of a density function in some region, one is interested in evaluating the “mass” of a specific subregion. Moreover, albeit obvious it is worth stressing the following fact, which can be very useful in practical applications: when one has samples of different functions at a fixed set of scattered points on a polygonal domain, the weights of TPS cubature can be computed “a priori” once and for all. This could be important, for example, in the presence of a fixed net of sampling points for a density function which varies in time, and the problem is that of computing the integral of the density on the given domain at several different times.

4 Numerical results.

In this section we present several numerical tests of cubature from scattered data by the formulas obtained in section 3 (TPS-Green cubature). First, we show the behavior of the formulas by three test functions on two test polygons with “difficult” geometry. Then, we show how to improve efficiency of TPS-Green cubature by splitting the data into rectangular subcells.

All the tests have been done in Matlab (version 6.1), with an Intel-Centrino 1.4 processor and 512 Mb RAM.

4.1 TPS-Green cubature over difficult geometries.

We have considered the following test functions

$$\begin{aligned} f_1(x, y) &= \exp(x - y), \quad f_2(x, y) = \exp(5(x - y)), \\ f_3(x, y) &= \sqrt{(x - 0.5)^2 + (y - 0.5)^2}, \end{aligned} \quad (24)$$

and the two nonconvex polygons in Figure 1 (both with area $1/2$). Observe that f_1 and f_2 are C^∞ (with f_2 varying more rapidly), whereas f_3 has a singularity of the gradient in $(0.5, 0.5)$.

In Tables 2-3 we show the errors of TPS interpolation (in the max norm on a suitable control grid) and of TPS-Green cubature, corresponding to a sequence of scattered samples obtained by the uniform random distribution. The displayed values are the averages on 50 independent trials. In addition, we report also the error of Monte Carlo (MC) integration on the same data. For TPS interpolation and cubature equations (cf. (9)-(13)) we have used the standard direct solver of Matlab (ultimately Gaussian elimination).

The reference integrals have been computed by the Matlab `dblquad` function (adaptive cubature routine) for the integrand multiplied by the characteristic function of the domain (which can be implemented via the Matlab `inpolygon` function, cf. [11]). This works, however, only by a suitable splitting of the enclosing square into subsquares. In fact, the procedure applied directly to

the whole enclosing square gives unsatisfactory results: even with an input tolerance of $1\text{E-}10$ for `dblquad`, one obtains cubature errors of $2\text{E-}03$, $2\text{E-}02$, $5\text{E-}04$ (polygon in Figure 1-left) and of $4\text{E-}02$, $1\text{E+}00$, $2\text{E-}02$ (polygon in Figure 1-right), for f_1, f_2, f_3 , respectively. The same happens trying to compute simply the polygons area by integrating the constant 1. This unreliability seems to be a consequence of the shape of the polygons, which causes manifest difficulties to the `dblquad` function (release 1.13).

It is worth stressing that our Matlab implementation of TPS-Green cubature is truly meshless, since it needs only the scattered sample and the sequence of vertices of the polygon, in counterclockwise order. This works even in the case of the polygon in Figure 1-right, whose boundary has two “double” points.

Observe that TPS-Green cubature is quite accurate, with errors that are up to three orders of magnitude smaller than those of TPS interpolation, and about two orders of magnitude below those of Monte Carlo integration. The function f_2 is surprisingly the “hardest”, whereas f_3 is integrated satisfactorily in spite of the singularity. Higher accuracy of TPS cubature with respect to interpolation is natural (as with other interpolatory formulas), and has already been mentioned in [14]. Higher accuracy with respect to Monte Carlo is also remarkable, especially thinking to practical applications where sampling is very costly or even cannot be refined, and thus one has to compute integrals from small/moderate size samples.

In Table 4 we display two important quantities related to *stability* of cubature by RBF, cf. estimate (14): the spectral norm of the inverse of the collocation matrix, and the sum of the weights absolute values. It is worth recalling (cf. [14]) that the weights are not all positive, but with TPS the negative ones are few and “small”. Indeed, the sum of the weights absolute values remains always around 0.6, i.e. close to the area of the corresponding polygon which is 0.5 in the examples.

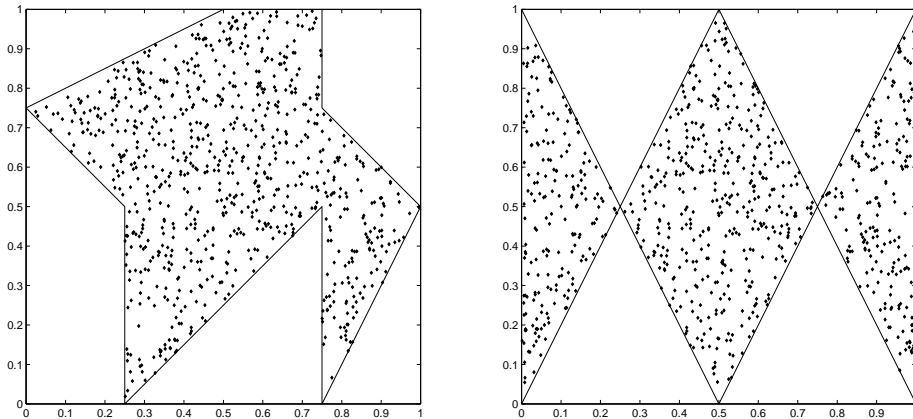


Figure 1: Two nonconvex polygons with a sample of 800 random points.

Table 2: Errors of TPS interpolation and TPS-Green compared to Monte Carlo cubature with $n = 100, 200, 400, 800$ uniform random points on the nonconvex polygon in Fig. 1-left (average values on 50 independent trials, rounded to the 1st significant digit).

function	formula	100 pts	200 pts	400 pts	800 pts
f_1	<i>TPS_intp</i>	5E-02	3E-02	2E-02	1E-02
	<i>TPS_Green</i>	1E-04	4E-05	2E-05	8E-06
	<i>MC</i>	1E-02	9E-03	8E-03	5E-03
f_2	<i>TPS_intp</i>	8E+00	5E+00	3E+00	1E+00
	<i>TPS_Green</i>	2E-02	8E-03	3E-03	9E-04
	<i>MC</i>	2E-01	1E-01	9E-02	7E-02
f_3	<i>TPS_intp</i>	4E-02	2E-02	1E-02	8E-03
	<i>TPS_Green</i>	2E-04	7E-05	2E-05	6E-06
	<i>MC</i>	4E-03	3E-03	2E-03	2E-03

Table 3: As in Table 2 for the nonconvex polygon in Fig. 1-right.

function	formula	100 pts	200 pts	400 pts	800 pts
f_1	<i>TPS_intp</i>	1E-01	5E-02	3E-02	2E-02
	<i>TPS_Green</i>	3E-04	7E-05	2E-05	1E-05
	<i>MC</i>	2E-02	1E-02	7E-03	6E-03
f_2	<i>TPS_intp</i>	2E+01	2E+01	2E+01	2E+01
	<i>TPS_Green</i>	2E-02	7E-03	4E-03	8E-04
	<i>MC</i>	4E-01	4E-01	2E-01	2E-01
f_3	<i>TPS_intp</i>	5E-02	2E-02	2E-02	9E-03
	<i>TPS_Green</i>	2E-04	8E-05	3E-05	9E-06
	<i>MC</i>	6E-03	5E-03	3E-03	2E-03

Table 4: TPS-Green cubature with $n = 100, 200, 400, 800$ uniform random points on the nonconvex polygons in Fig. 1: spectral norm of the inverses of the collocation matrices and 1-norm of the computed weights vectors (average values on 50 independent trials, rounded to the 1st significant digit).

polygon	norms	100 pts	200 pts	400 pts	800 pts
Fig.1-left	$\ \mathcal{A}^{-1}\ _2$	2E+05	1E+06	3E+06	2E+07
	$\ \tilde{\mathbf{w}}\ _1$	6E-01	6E-01	6E-01	6E-01
Fig.1-right	$\ \mathcal{A}^{-1}\ _2$	2E+05	1E+06	2E+06	2E+07
	$\ \tilde{\mathbf{w}}\ _1$	6E-01	6E-01	6E-01	6E-01

4.2 Improving efficiency by data splitting.

Our tests with TPS-Green cubature have shown that it provides a flexible, reasonably accurate and stable meshless method, which gives good results even with relatively small scattered samples (say a size of the hundreds of points).

Already with “moderate” size samples (say a size of the thousands), however, we face the classical complexity problem of globally supported RBF with direct solvers.

Dealing with interpolation, a number of fast methods have been developed to accelerate the solution of the collocation equations, typically within the frame of iterative solvers; cf., e.g., [5, 16] and references therein. In the case of numerical cubature on polygons, however, there is a simple alternative for improving efficiency, still using a direct solver. Due to additivity of the integral, we can split the polygonal domain into subdomains, obtained by subdividing the least enclosing rectangle into *non-overlapping rectangular subcells* and then intersecting each subcell with the domain. This entails a splitting of data among the corresponding *polygonal subdomains*. The integral on each subdomain can be computed with TPS-Green cubature, where each local system of weights equations can be much smaller than the original global one, depending on the number of subcells, the distribution of the sampling nodes and the shape of the domain.

This approach can be even improved with a little additional cost, by using not only the nodes of a subcell for computing the weights pertaining to the corresponding subpolygon, but also those belonging to a slightly larger subcell (*overlap* approach: the rectangular subcell sides are expanded by a small percentage). In such a way we can partially recover the information lost with data splitting, for example because we reduce the effect of boundaries (cf. [7]). This is reminiscent of a well-known domain decomposition method for RBF interpolation [2], but without the nontrivial complication of interfacing the approximations.

All the procedure is simple to implement in Matlab, by resorting to the already quoted function `inpolygon` and to another computational geometry tool within the Mapping Toolbox [11], `polybool` (a function which computes the vertices of polygons intersections). In Tables 5-6 we show the behavior of TPS-Green cubature with the data splitting technique just described, correspondingly to subdivisions of the least enclosing square for the polygon in Figure 1-left into 4, 9 and 16 equal subsquares, with and without overlap. Overlap percentages refer to the corresponding expansion of the subsquares side. The sample consists of 3000 points, generated with the uniform random distribution.

Observe that with no splitting, the bulk of the computational process is clearly given by the construction and solution of the weights equations. When the number of subsquares increases this cost decreases, but not quadratically as one could expect in principle, also due to the effect of the original polygon shape in the nonuniform distribution of the nodes among the subcells. In fact, if the splitting produces p subpolygons Ω_i containing n_i nodes respectively, $i = 1, \dots, p$, with $n_1 + \dots + n_p = n$, the speed-up with respect to the global construction is roughly given by that of the Gaussian elimination process

$$\text{speed-up for the weights} \approx \frac{n^3}{\sum_{i=1}^p n_i^3},$$

which is less than p^2 (the equality holds when the denominator is minimum, i.e. with $n_i = n/p$ for every i).

The cost of integration of the radial basis functions, which is roughly proportional to the average number of sides of a subpolygon, is negligible with coarse splittings, and in any case is quite stable compared to the other costs. The cost of the geometrical part (managing of subpolygons) seems to increase linearly with the number of subsquares, as somehow expected. On the other hand, too fine splittings would cause a severe loss of precision, for example when subpolygons begin to appear that do not contain nodes of the sample. These observations suggest that there is an “optimal” splitting, which minimizes CPU time at a given error tolerance. We have not been able to perform further subdivisions, since the release of the `polybool` function at our disposal (release 1.3) gives wrong results when the subcells become too small (e.g., already with 25 subsquares in the example). This bug should have been eliminated in more recent releases of the Mapping Toolbox, see [11].

In any case the cubature errors and total timings show that with a suitable number of subsquares and using a modest overlap, we get an approximation close to and often even better than that of the global method, with a *much smaller computational cost*. For example, with 9 and 16 subsquares, and 10% overlap, we get speed-ups of about 11 and 16 times, respectively.

Table 5: Errors of TPS-Green cubature with a sample of 3000 uniform random points on the nonconvex polygon in Fig. 1-left, by splitting into subsquares, with and without overlap.

subsqs	overlap	f_1	f_2	f_3
1		2E-07	3E-05	1E-06
4	0%	1E-06	8E-05	6E-06
	10%	8E-08	2E-05	7E-07
	20%	3E-08	4E-06	8E-07
9	0%	1E-06	4E-05	3E-06
	10%	2E-07	3E-05	3E-07
	20%	5E-08	1E-06	7E-07
16	0%	3E-06	4E-05	8E-06
	10%	2E-07	1E-04	7E-08
	20%	2E-07	4E-05	5E-07

References

- [1] T.M. Apostol, *Calculus*, vol. II, 2nd edition, Blaisdell, 1969.
- [2] R.K. Beatson, W.A. Light and S. Billings, *Fast solution of the radial basis function interpolation equations: domain decomposition methods*. SIAM J. Sci. Comput. **22** (2000), 1717–1740.
- [3] A.Yu. Bezhaev, *Cubature formulae on scattered meshes*, Soviet J. Numer. Anal. Math. Modelling **6** (1991), 95–106.

Table 6: Partial and total CPU times (seconds) of TPS-Green cubature for the example in Table 5: construction and solution of the weights equations, cubature of the RBF, managing of subpolygons.

subsqs	overlap	weights	basis cub	geometry	tot
1		22.68	0.20	0.00	22.88
4	0%	3.14	0.14	0.08	3.36
	10%	3.74	0.15	0.08	3.97
	20%	4.40	0.16	0.09	4.65
9	0%	1.47	0.15	0.16	1.88
	10%	1.75	0.16	0.16	2.07
	20%	2.11	0.18	0.16	2.45
16	0%	0.82	0.12	0.30	1.24
	10%	0.99	0.12	0.30	1.41
	20%	1.19	0.13	0.30	1.62

- [4] M.D. Buhmann, *Radial basis functions*, Acta Numer. **9** (2000), 1–38.
- [5] M.D. Buhmann, *Radial Basis Functions: Theory and Implementation*, Cambridge Monographs on Applied and Computational Mathematics, vol. 12, Cambridge University Press, 2003.
- [6] R. Cools and D. Laurie (Eds.), *Numerical evaluation of integrals*, J. Comput. Appl. Math. **112** (1999), no. 1-2.
- [7] B. Fornberg, T.A. Driscoll, G. Wright and R. Charles, *Observations on the behavior of radial basis function approximations near boundaries*, Comput. Math. Appl. **43** (2002), 473–490.
- [8] G. Green, *An Essay on the Application of Mathematical Analysis to the Theories of Electricity and Magnetism*, Nottingham, 1828.
- [9] A. Iske, *Multiresolution Methods in Scattered Data Modelling*, Lecture Notes in Computational Science and Engineering, vol. 37, Springer, 2004.
- [10] D. Levin, *Stable integration rules with scattered integration points*. Numerical evaluation of integrals. J. Comput. Appl. Math. **112** (1999), 181-187.
- [11] The MathWorks, *MATLAB Documentation Set*, 2006 version (available online at <http://www.mathworks.com>).
- [12] R. Schaback, *Error estimates and condition numbers for radial basis function interpolation*, Adv. Comput. Math. **3** (1995), 251-264.
- [13] R. Schaback and H. Wendland, *Kernel Techniques: From Machine Learning to Meshless Methods*, Acta Numer., 2006, to appear.
- [14] A. Sommariva and M. Vianello, *Numerical cubature on scattered data by radial basis functions*, Computing **76** (2005), 295-310.

- [15] A. Sommariva and R. Womersley, *Integration by RBF over the sphere*, 2005, submitted (preprint UNSW AMR05/17, available online at <http://www.maths.unsw.edu.au>).
- [16] H. Wendland, *Scattered Data Approximation*, Cambridge Monographs on Applied and Computational Mathematics, vol. 17, Cambridge University Press, 2005.
- [17] Wolfram Research, Inc., *The Wolfram Integrator*, 2005 version (available online at <http://integrals.wolfram.com>).