Meshless cubature over the disk by Thin-Plate Splines ⋆

Alessandro Punzi, Alvise Sommariva [a,*], Marco Vianello [a]

[a] *Department of Pure and Applied Mathematics, University of Padua (Italy)*

Abstract

In some recent papers, the construction of meshless interpolatory cubature formulas by Radial Basis Functions has been studied. In particular, Thin-Plate Splines allow to use conveniently Green's formula and give good results on scattered samples of small/moderate size over polygons. Here, we discuss the extension to meshless cubature over the disk and its practical implementation.

Key words: meshless cubature, disk, scattered data, Thin-Plate Splines, Green's formula.

1 Introduction.

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

$$I(f) = \int_\Omega f(P)\, dP \ , \quad \Omega \subset \mathbb{R}^2 \ , \tag{1}$$

from a *scattered* sample of size n

$$\mathbf{f} = \{f(P_i)\} \ \text{ at } \ X = \{P_i\} = \{(x_i, y_i)\} \subset \Omega \ , \quad i = 1, \ldots, n \ . \tag{2}$$

In a recent paper [9], such a problem has been solved for *polygons* by resorting to the powerful tool of RBF (Radial Basis Functions) interpolation, in connection with a cornerstone of multivariate calculus, Green's integral formula.

⋆ Work supported by the ex-60% funds of the University of Padova and by the GNCS-INdAM.

* Corresponding author. Address: via Trieste 63, 35121 Padova (Italy).
Email address: `alvise@math.unipd.it` (Alvise Sommariva).

Here, we extend the method to numerical cubature from scattered data over the *disk*.

In the next section, we'll give a brief survey of numerical cubature by RBF, which shows that TPS (Thin-Plate Splines) are a good choice in this context. In section 3 we describe the TPS-Green cubature method specialized to the case of the disk, where quadrature along the boundary is a much more delicate problem with respect to polygons. In section 4, finally, we present a set of numerical tests on randomly generated samples, where a comparison is made also with MonteCarlo method.

2 Numerical cubature by RBF.

The topic of numerical cubature from scattered data is much less developed with respect to the construction of cubature formulas on nodes with a predefined distribution (cf., e.g., [3]). Dealing with scattered data, one can try to construct an interpolatory cubature formula using RBF. This approach has been pursued for the first time in [8] for integration over the square, and then extended to general polygons in [9], and to the sphere in [11]. We recall here the general features of RBF cubature, displaying for brevity only the CPD (conditionally positive definite) case. Concerning RBF interpolation theory, we refer the reader to the recent monographs [2,5,14].

The integral (1) can be approximated by integrating an RBF interpolant at the scattered data (2)

$$s(P) = \sum_{j=1}^{n} c_j \, \phi_j(P) + \sum_{k=0}^{\mu} d_k \, \pi_k(P) \; , \quad s(P_i) = f(P_i) \; , \quad i = 1, \dots, n \; , \tag{3}$$

where

$$\phi_j(P) = \phi_j(P; \delta) = \phi(|P - P_j|/\delta) \; , \tag{4}$$

$\phi(r)$ being the underlying radial function and δ a scaling parameter that can be related to the data density. The polynomial term of suitable degree m is added to get unisolvence of the interpolation problem (e.g., $m = 1$ for TPS), and $\{\pi_k\}$ is a basis of the corresponding bivariate polynomial space of dimension $\mu = (m + 1)(m + 2)/2$.

Now, considering the vectors

$$\mathbf{F} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} \; , \quad \mathbf{I} = \begin{bmatrix} \mathbf{I}_R \\ \mathbf{I}_\pi \end{bmatrix} \quad \text{with } \; \mathbf{I}_R = \{I(\phi_j)\} \; , \; \mathbf{I}_\pi = \{I(\pi_k)\} \; . \tag{5}$$

and the (symmetric) augmented matrix

$$\mathcal{A} = \begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \;,\;\; A = A_{X,\phi} = [\phi_j(P_i)] \;,\;\; B = [\pi_k(P_i)] \;, \tag{6}$$

it is easy to see that the cubature formula can be written in the usual form of weighted sum of the function values,

$$I(s) = \langle \mathcal{A}^{-1}\,\mathbf{F}, \mathbf{I}\rangle = \langle \mathcal{A}^{-1}\,\mathbf{I}, \mathbf{F}\rangle = \langle \mathbf{W}, \mathbf{F}\rangle = \langle \mathbf{w}, \mathbf{f}\rangle = \sum_{j=1}^{n} w_j\, f_j \;, \tag{7}$$

$\langle\cdot,\cdot\rangle$ denoting the scalar product in $\mathbb{R}^n$. Thus, as usual with interpolatory formulas, the weights $\mathbf{w} = \{w_j\}$ are obtained by solving the linear system

$$\mathcal{A}\,\mathbf{W} = \mathbf{I} \;,\;\; \mathbf{W} = \begin{bmatrix} \mathbf{w} \\ \mathbf{z} \end{bmatrix} \;\; \text{(weights equations)} \;. \tag{8}$$

We give now an error estimate, which takes into account the presence of errors in the evaluation of the integrals, $\tilde{\mathbf{I}} \approx \mathbf{I}$, and of possible noise in the sample, $\tilde{\mathbf{f}} \approx \mathbf{f}$; the relevant parameters are the *fill distance* h, and the *separation distance* q of the point set X (recall that $q \leq 2h$). The cubature error can be bounded as

$$\begin{aligned} |I(f) - \langle \tilde{\mathbf{w}}, \tilde{\mathbf{f}}\rangle| &\leq |I(f) - I(s)| + |\langle \mathbf{w} - \tilde{\mathbf{w}}, \mathbf{f}\rangle| + |\langle \tilde{\mathbf{w}}, \mathbf{f} - \tilde{\mathbf{f}}\rangle| \\ \leq \sqrt{\mathrm{meas}(\Omega)}\; \|f - s\|_{L^2(\Omega)} &+ \|\mathcal{A}^{-1}\|_2\, \|\mathbf{f}\|_2\, \|\mathbf{I} - \tilde{\mathbf{I}}\|_2 + \|\mathbf{f} - \tilde{\mathbf{f}}\|_\infty\, \|\tilde{\mathbf{w}}\|_1 \\ = \mathcal{O}\left(\sqrt{\mathcal{F}_\phi(h)}\right) + \mathcal{O}\left(\frac{1}{\mathcal{G}_\phi(q)}\right) &\|\mathbf{I} - \tilde{\mathbf{I}}\|_2 + \|\mathbf{f} - \tilde{\mathbf{f}}\|_\infty \sum_{j=1}^{n} |\tilde{w}_j| \;, \end{aligned} \tag{9}$$

where $\mathcal{F}_\phi(h) \downarrow 0$ as $h \to 0$ and $\mathcal{G}_\phi(q) \downarrow 0$ as $q \to 0$, and thus as $h \to 0$. Technically, such an estimate holds in the "native space" of the RBF, and requires that Ω satisfies the so-called "interior cone condition", cf. [2,5,14].

The simultaneous appearance of the terms $\sqrt{\mathcal{F}_\phi(h)} \downarrow 0$ and $1/\mathcal{G}_\phi(q) \uparrow +\infty$ in (9) is an occurrence, in the context of numerical cubature, of the well-known "*uncertainty principle*" in RBF interpolation, cf. [7]. 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*.

In Table 1, we report the two key parameters related to stability of RBF cubature, 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 ($\delta = 1$).

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).

$\sharp$ of rnd pts	norms	MQ	IMQ	G	$W2$	TPS
$n = 50$	$\lVert\mathcal{A}^{-1}\rVert_2$	2E+12	3E+11	5E+15	5E+03	6E+03
	$\lVert\tilde{\mathbf{w}}\rVert_1$	7E+01	9E+01	3E+02	2E+00	1E+00
$n = 100$	$\lVert\mathcal{A}^{-1}\rVert_2$	2E+16	6E+15	3E+17	5E+04	8E+03
	$\lVert\tilde{\mathbf{w}}\rVert_1$	8E+02	5E+02	1E+03	2E+00	1E+00

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. However, the numerical experiments in [8] have shown that (9) is by far an 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, whereas TPS and W2 give reasonably accurate and more stable cubature formulas. Moreover, the quality of cubature by TPS is weakly sensitive 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. [5]). Clearly, this property makes TPS cubature very attractive for automatic integration, since it avoids the complication of managing/optimizing the scaling parameter.

3 Meshless cubature over the disk.

In [8,9] it has been possible to integrate the basis functions by means of symbolic computation (see [13]), obtaining closed-form formulas for the integrals $\{I(\phi_j\}$. In fact, concerning integration of RBF over the square, one can resort to radial symmetry (cf. [8]), and more generally integration of TPS over polygons can be accomplished via Green's formula, computing a first-level primitive in one of the variables, and then a second-level primitive along each side (cf. [9]). In the latter case, the success of the procedure is due to the peculiarity of the TPS basis and to the fact that the parametrization of each side is an affine mapping.

Concerning integration of TPS over the disk, the situation is more delicate. We begin by recalling Green's formula (cf. [4] and, e.g., [1]), specialized to the case of the unit disk, i.e.

- $\Omega = D = \{P = (x, y) : x^2 + y^2 \leq 1\}$

which is not restrictive, in view of the obvious "shift and scale" change of variables which maps any disk into the unit disk (interpolation and cubature by TPS are weakly sensitive to scaling, as already observed). Now, recall that we are dealing with

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

Setting for $j = 1, \dots, n$

$$\psi_j(P) = \int \phi(|P - P_j|)\, dx \;, \tag{10}$$

Green's formula over the disk reads as

$$I(\phi_j) = \int_D \phi(|P - P_j|)\, dP = \oint_{\partial D} \psi_j(P)\, dy = \int_{-\pi}^{\pi} \psi_j(\cos\theta, \sin\theta)\; \cos\theta\, d\theta \;, \tag{11}$$

where the first-level primitive $\psi_j(P)$ can be computed explicitly by the powerful symbolic integrator [15],

$$\psi_j(P) = \psi_{j1}(P) + \psi_{j2}(P) + \psi_{j3}(P) \;, \tag{12}$$

where

$$\psi_{j1}(P) = -\frac{1}{9}\,(x - x_j)^3 - \frac{2}{3}\,(x - x_j)(y - y_j)^2 \;,$$

$$\psi_{j2}(P) = \frac{1}{6}(x - x_j)((x - x_j)^2 + 3(y - y_j)^2)\; \log\left((x - x_j)^2 + (y - y_j)^2\right)$$

$$\psi_{j3}(P) = \frac{2}{3}\,(y - y_j)^3\; \arctan\left(\frac{x - x_j}{y - y_j}\right) \;. \tag{13}$$

Unfortunately, here symbolic integrators fail in computing the second-level primitive "along the boundary", i.e. $\int \psi_j(\cos\theta, \sin\theta)\; \cos\theta\, d\theta$, and thus the last integral in (11) has to be computed *numerically*. The difficulties are in the second and third term in (13), whereas the first can be integrated symbolically. On the contrary, integration on the disk of the polynomial part in (3) (here $m = 1$) is trivial in polar coordinates. We stress that, in any case, we have reduced a 2-dimensional to a 1-dimensional integration problem, with a clear computational advantage.

Various choices are possible. In view of the periodicity, the simple trapezoidal rule is appealing, but we have to take into account that both ψ_{j2} and ψ_{j3} have low regularity. Indeed, ψ_{j2} has a critical point at $P = P_j$, and ψ_{j3} has a whole critical line, $y = y_j$. These generate one possible critical point for ψ_{j2} on the circle (when P_j is on or close to the circle), and two critical points for ψ_{j3} on the circle (the intersections with the critical line). Then we chose two different parametrizations in such a way that the critical points are endpoints of the parameter intervals, and quadrature formulas that automatically cluster nodes at the endpoints, like Gauss-Legendre or Clenshaw-Curtis.

The one-dimensional integrals to be computed are

$$\oint_{\partial D} \psi_{j2}(P)\,dy = \int_{\theta_j}^{\theta_j+2\pi} \psi_{j2}(\cos\theta, \sin\theta)\,\cos\theta\,d\theta\ ,\quad \theta_j = \arg(P_j)\ , \tag{14}$$

and by splitting into two arcs

$$\oint_{\partial D} \psi_{j3}(P)\,dy = \int_{\min\{\theta_j^+,\theta_j^-\}}^{\max\{\theta_j^+,\theta_j^-\}} \psi_{j3}(\cos\theta, \sin\theta)\,\cos\theta\,d\theta$$

$$+ \int_{\max\{\theta_j^+,\theta_j^-\}}^{\min\{\theta_j^+,\theta_j^-\}+2\pi} \psi_{j3}(\cos\theta, \sin\theta)\,\cos\theta\,d\theta\ ,\quad \theta_j^{\pm} = \arg\left(\left(\pm\sqrt{1-y_j^2}, y_j\right)\right)\ . \tag{15}$$

Since the computation of Gauss-Legendre nodes and weights is expensive at high degree of exactness, whereas Clenshaw-Curtis formula can be implemented with low cost by the FFT and has in practice a comparable accuracy (cf. [12]), we have used the latter. We recall that when several different functions have to be integrated (as here), the formulation of Clenshaw-Curtis formula involving explicitly the weights is much more convenient. The weights can be computed once and for all by the fast algorithm recently proposed in [13] (based on the IFFT). We have numerical evidence that the Clenshaw-Curtis formula of degree 32 is sufficient to compute all the three integrals in (14)-(15) (and then $I(\phi_j)$) with 10 correct figures. Observe that 2-dimensional numerical integration of ϕ_j would require a much larger number of nodes to get the same accuracy.

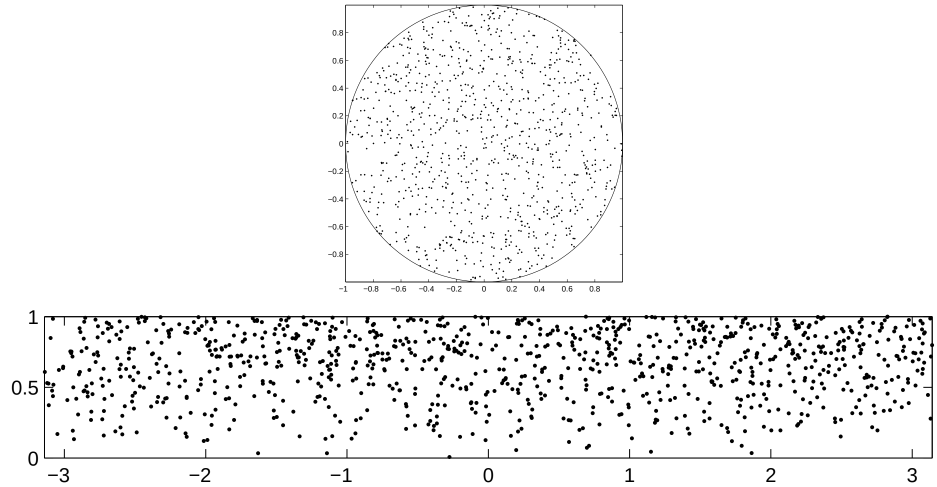

Fig. 1. The nonuniformity effect on a uniform random sample of 1000 points on the unit disk by mapping in the polar rectangle $[-\pi, \pi] \times [0, 1]$.

Remark 1 (*Polar or cartesian coordinates?*) Following [9], we term the approach described above TPS-Green meshless cubature (in cartesian coordinates). One might think, in principle, that the whole construction is not necessary, since integration over the disk is integration over a rectangle in polar coordinates, and thus one could simply work from the beginning in such coordinates using the cubature method developed in [8]. In such a case, clearly, the TPS would be themselves TPS in polar coordinates. However, if we start for example from a random sample generated by the uniform probability distribution in cartesian coordinates, the mapping from the disk to the polar rectangle causes a distortion (see Figure 1): the density of the mapped points decreases approaching the center of the disk, and this deteriorates the quality of the approximation. The effect of such nonuniformity will be manifest in the numerical experiments of the next section (compare "TPS-Green polar" with "TPS-Green cartesian").

Remark 2 (*Possible extensions.*) The TPS-Green meshless cubature method is not restricted to polygons (cf. [9]), or disks. It could be extended to any bounded domain, even multiply connected, with piecewise regular external and internal boundaries ($\partial\Omega = \Gamma^{\text{ext}} \cup \Gamma^{\text{int}}$), namely in computing

$$\begin{aligned} I(\phi_j) &= \oint_{\partial\Omega} \psi_j(P)\, dy = \oint_{\Gamma^{\text{ext}}} \psi_j(P)\, dy - \oint_{\Gamma^{\text{int}}} \psi_j(P)\, dy \\ &= \sum_k \int_{\Gamma_k^{\text{ext}}} \psi_j(P)\, dy - \sum_s \int_{\Gamma_s^{\text{int}}} \psi_j(P)\, dy \;, \end{aligned} \tag{16}$$

provided that the boundary pieces $\{\Gamma_k^{\text{ext}}\}$ and $\{\Gamma_s^{\text{int}}\}$ are suitably parametrized. Clearly, apart from the case of a linear piece of the boundary (segment) where explicit formulas are known (cf. [9]), integration of ψ_j along the boundary has to be done numerically. For example, extension to ellipses is immediate by integrating $\psi_j(a\cos\theta, b\sin\theta)\cos\theta$, where a and b are the semiaxes. Again, one can use Clensahw-Curtis quadrature as above. In such a way we can work directly with TPS on the ellipse, avoiding the distortion of the sampling point set occurring via transformation of the ellipse into a square or into a disk (change of variables in the integration). We have also implemented the extension in the case of (annular) *circular sectors*, i.e.

$$\Omega = \mathcal{S}(\rho_1, \rho_2, \alpha_1, \alpha_2) = \{P :\ 0 \le \rho_1 \le |P| \le \rho_2,\ \alpha_1 \le \arg(P) \le \alpha_2\} \;, \tag{17}$$

whose boundary consists of the two circular arcs $|P| = \rho_i$ and of the two segments $\arg(P) = \alpha_i$, $i = 1, 2$. Here, integration of $\psi_j(P)$ along the segments is explicit, whereas Clenshaw-Curtis quadrature along each arc (the internal clockwise and the external counterclockwise) is possibly split into two or three subarcs, to manage as above the critical points.

4 Numerical tests.

In this section we present several numerical tests of numerical integration from scattered data over the disk, by "TPS-Green" meshless cubature. First, we show the behavior of the formulas on three test functions, (already used in [9] concerning meshless cubature over polygons), dealing with samples of small size (in the hundreds). Then, we show how to improve efficiency of TPS-Green cubature on samples of larger size (in the thousands) by splitting the disk into sectors or annuli. All the tests have been done in Matlab (cf. [6]), with an Intel-Centrino Duo 2.3Ghz processor and 1Gb RAM. The corresponding Matlab code can be found at [10]. Our implementation is truly *meshless* (no kind of mesh is required or generated), and it needs only the parameters defining the disk (or more generally the annular sector, i.e. the center, and the radial and angular intervals) besides the scattered sample.

We have considered the following test functions

$$f_1(x,y) = \exp(x-y)\ , \quad f_2(x,y) = \exp(5(x-y))\ , \quad f_3(x,y) = \sqrt{x^2+y^2}\ . \tag{18}$$

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 the origin.

In Table 2 we show the parameters relevant to stability of cubature by RBF, i.e. the spectral norm of the collocation matrix inverse, and the sum of the weights absolute values (cf. estimate (9)). It is worth observing that the weights are not all positive, but the sum of the negative ones is, in absolute value, much smaller than that of the positive ones. Thus, the sum of the weights absolute values is always of the order of the sum of the weights, which approximates π (the unit disk area).

In Table 3 we compare the errors of TPS-Green cubature in cartesian and polar coordinates (see Remark 1), and of MonteCarlo formula, on a sequence of scattered samples in the unit disk obtained by the uniform random distribution in the enclosing square. The linear systems involved (weights equations) have been solved by the standard direct solver of Matlab. Observe that TPS-Green cubature in cartesian coordinates is more accurate than that in polar coordinates, due to the distortion effect described in Remark 1, and also than MonteCarlo integration (up to two orders of magnitude). This latter fact is remarkable thinking to practical applications, where sampling is very costly or even cannot be refined, and thus a reasonably accurate value of the integral should be obtained from the available data.

Table 2
Spectral norm of the inverses of the collocation matrices and 1-norm of the computed weights vectors (cartesian coordinates).

norms	100 pts	200 pts	400 pts	800 pts
$\|\mathcal{A}^{-1}\|_2$	2E+04	1E+05	2E+05	3E+05
$\|\tilde{\mathbf{w}}\|_1$	3.56	3.55	3.44	3.65

These and many other numerical tests (not reported for brevity) show that

Table 3
Relative errors of TPS-Green cubature in polar and cartesian coordinates and of MonteCarlo cubature with $n = 100, 200, 400, 800$ uniform random points on the unit disk.

function	formula	100 pts	200 pts	400 pts	800 pts
f_1	TPS-Green (polar)	3E-03	2E-03	2E-03	3E-04
	TPS-Green (cart.)	1E-03	1E-04	1E-04	6E-06
	MonteCarlo	4E-02	1E-02	2E-02	1E-03
f_2	TPS-Green (polar)	2E-02	1E-02	1E-02	6E-03
	TPS-Green (cart.)	3E-02	2E-02	2E-03	8E-04
	MonteCarlo	3E-01	1E-01	4E-02	4E-03
f_3	TPS-Green (polar)	3E-02	1E-02	1E-02	6E-03
	TPS-Green (cart.)	5E-04	4E-04	5E-05	4E-05
	MonteCarlo	1E-01	4E-02	2E-02	1E-02

TPS-Green cubature provides a flexible, reasonably accurate and stable meshless method on disks, sectors and annuli, which gives good results even with relatively small scattered samples (say a size in the hundreds of points). Already with “moderate” size samples (say a size in 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., [2,14] and references therein.

In the case of numerical cubature, however, there is a simple alternative for improving efficiency, even using a direct solver, at least until conditioning remains acceptable. Due to additivity of the integral, we can split the integration domain, and hence the data, into subdomains, like (sub)sectors or (sub)annuli. 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 subdomains and on the distribution of the sampling points.

In Table 4 we show an example of the behavior of TPS-Green cubature with the data splitting technique just described, by subdividing the unit disk into 16 equal area annuli. Observe that with no splitting, the bulk of the computational process is clearly given by the assembling and solution of the weights equations. When the splitting is applied, the cost of the weights equations is pulled down, whereas the cost of integration of the TPS is roughly doubled (each ψ_j is integrated along two circles on the average, namely the boundaries of an annulus). The resulting effect is a remarkable reduction of the global CPU time (speed-up ratio about 9 in this example).

Table 4
Relative errors and CPU times (seconds): TPS-Green cubature of $f(x,y) = \exp(5(x^2+y^2))$ by 3000 uniform random points on the unit disk, directly and by subdividing into 16 equal area annuli.

subdivisions	error	CPU weights	CPU basis cub	CPU tot	speed-up
none	5E-4	17.3	1.1	18.4	
16	1E-4	0.3	1.8	2.1	8.8

References

[1] T.M. Apostol, *Calculus*, vol. II, 2nd edition, Blaisdell, 1969.

[2] M.D. Buhmann, *Radial Basis Functions: Theory and Implementation*, Cambridge Monographs on Applied and Computational Mathematics, vol. 12, Cambridge University Press, 2003.

[3] R. Cools and D. Laurie (Eds.), *Numerical evaluation of integrals*, J. Comput. Appl. Math. **112** (1999), no. 1-2.

[4] G. Green, *An Essay on the Application of Mathematical Analysis to the Theories of Electricity and Magnetism*, Nottingham, 1828.

[5] A. Iske, *Multiresolution Methods in Scattered Data Modelling*, Lecture Notes in Computational Science and Engineering, vol. 37, Springer, 2004.

[6] The MathWorks, *MATLAB Documentation Set*, 2006 version (available online at http://www.mathworks.com).

[7] R. Schaback, *Error estimates and condition numbers for radial basis function interpolation*, Adv. Comput. Math. **3** (1995), 251–264.

[8] A. Sommariva and M. Vianello, *Numerical cubature on scattered data by radial basis functions*, Computing **76** (2005), 295–310.

[9] A. Sommariva and M. Vianello, *Meshless cubature by Green's formula*, Appl. Math. Comput., in press (available online 28 August 2006).

[10] A. Sommariva and M. Vianello, *TPS-Green-D: Matlab code for meshless cubature over disks by Thin-Plate Splines*, November 2006 (downloadable at http://www.math.unipd.it/~marcov/software.html).

[11] A. Sommariva and R. Womersley, *Integration by RBF over the sphere*, 2006, submitted (preprint UNSW AMR05/17, available online at http://www.maths.unsw.edu.au).

[12] L.N. Trefethen, *Is Gauss quadrature better than Clenshaw-Curtis?*, SIAM Rev., to appear.

[13] J. Waldvogel, *Fast construction of the Fejér and Clenshaw-Curtis quadrature rules*, BIT 46 (2006), 195–202.

[14] H. Wendland, *Scattered Data Approximation*, Cambridge Monographs on Applied and Computational Mathematics, vol. 17, Cambridge University Press, 2005.

[15] Wolfram Research, Inc., *The Wolfram Integrator*, 2006 version (available online at http://integrals.wolfram.com).