

RBF BASED QUADRATURE ON THE SPHERE

MANUEL KINDELAN ^{a,*}, PEDRO GONZÁLEZ-RODRÍGUEZ ^a, DIEGO ÁLVAREZ ^a

^aDepartment of Mathematics
Carlos III University of Madrid
Avenida de la Universidad 30, 28911 Leganés, Spain
e-mail: {kinde, pgonzale}@ing.uc3m.es, jdiego@math.uc3m.es

The paper describes a new RBF-FD based technique to compute quadrature weights on the sphere. In the proposed method, the sphere is divided into rectangles in the latitude-azimuth coordinate system, and the function is integrated over each rectangle using RBF interpolation. The method is easy to implement and its accuracy is comparable to that based on SPH expansions. One advantage of the proposed method is its ability to handle non-uniform node distributions. On this respect, we propose a new algorithm to cluster nodes in regions of steep changes in the function. It is a repulsion-based algorithm with a non-uniform distribution of electrical charges. We show that, using node clustering, the accuracy of the method can be significantly improved.

Keywords: RBF-FD, quadrature, sphere.

1. Introduction

The calculation of solutions of PDEs over the surface of the sphere (Fornberg and Piret, 2008; Flyer and Forberg, 2011; Flyer *et al.*, 2012; 2014; Fornberg and Flyer, 2015) has a significant number of applications in geophysics and mathematical biology. Once the solution is obtained, it is often necessary to provide total or average values of the dependent variables in order to interpret the results. These quantities are computed using numerical quadratures over a typically large number of nodes, which may exhibit a variable spatial density to improve the resolution in regions where the function varies rapidly. Another application of the techniques presented in this paper is the numerical solution of boundary integral equations. Since the seminal work by Atkinson (1982) a significant number of works on this subject has been carried out. For instance, Bruno and Kunyansky (2001), Klöckner *et al.* (2013) as well as Klinteberg and Tornberg (2016) address the case of singular boundary integral equations.

For the integration of general functions, quadrature methods perform better when the nodes are near-uniformly distributed. Thus, much of the work carried out in the past was focused on near-uniform node sets specially designed for numerical quadrature on the

sphere (Stroud, 1971; Bazant and Oh, 1986; Ahrens and Beylkin, 2009), and their weights were tabulated for different values of the total number of nodes. On the sphere, these quadrature weights are determined by requiring exact results for as high order spherical harmonics (SPH) as possible.

There are also near-uniform node sets that were not specifically designed for quadrature on the sphere. The most common ones are the *minimal energy* (ME) nodes and the *maximal determinant* (MD) nodes. The former correspond to the equilibria locations of mutually repelling point charges, while the latter are constructed optimizing the conditioning of spherical harmonics interpolation. Both types of node sets have been used to calculate and tabulate weights using spherical harmonics (Womersley and Sloan, 2003). Although Hesse *et al.* (2010) showed that computation of quadrature weights for ME nodes using spherical harmonics often led to numerical instabilities, Fornberg and Martel (2014) proved that the instabilities associated with computing quadrature weights for ME nodes using SPH were due to rank deficiency, and this could be easily avoided by using a least-squares approach with a slightly reduced SPH order.

However, computing quadrature weights using SPH expansions has two main drawbacks: high computational

*Corresponding author

cost ($\mathcal{O}(N^3)$) and the requirement of quasi-uniform node distributions. These two limitations spurred the development of alternative approaches, such as RBF based interpolation and quadrature (Sommariva and Womersley, 2005; Fuselier *et al.*, 2014). Recently, Reeger and Fornberg (2016) as well as Reeger *et al.* (2016) proposed a new approach that borrows its concept from RBF generated finite differences (RBF-FD) (Fornberg and Flyer, 2015). With this method, the computational cost becomes order $\mathcal{O}(N \log N)$ and it allows for local node refinement. The method starts by creating a Delaunay spherical triangulation on the surface of the sphere, and projecting each triangle and some nearby nodes to a tangent plane. Quadrature weights are then computed for each triangle, and these weights are finally combined to obtain the full weight set for the sphere.

In this paper we propose an alternative and simpler RBF-FD based technique that does not need Delaunay triangularization or projection to a tangent plane. In the proposed method, the sphere is divided into rectangles in the latitude-azimuth coordinate system, and the function is integrated over each rectangle using RBF interpolation. For comparison purposes, we also compute quadrature on the sphere using a global RBF method. Numerical tests of the proposed method are carried out on three different integrands that have been used in the literature (Fornberg and Martel, 2014; Sommariva and Womersley, 2005; Fuselier *et al.*, 2014; Reeger and Fornberg, 2016; Reeger *et al.*, 2016; Beentjes, 2015).

We also analyze the behavior of the method in the case of a non-uniform distribution of nodes. To this end we propose a new algorithm to cluster nodes in regions of steep variation in the function. It is a repulsion-based algorithm with non-uniform distribution of electrical charges. We find that there is a significant improvement in accuracy when nodes are concentrated in regions where the function changes abruptly.

The paper is organized as follows. Section 2 describes the method used to compute quadrature weights in the sphere using global RBF interpolation. Section 3 describes the corresponding method using local RBF interpolation. These sections also contain numerical results related to application of each of the methods to a set of reference functions. Section 4 describes an algorithm to distribute nodes in the sphere which are nearly equispaced locally but with different node densities in order to cluster nodes in the regions of interest. These nodes are used to compute quadrature with the proposed method. Section 5 summarizes the main conclusions of this work.

2. Global RBF based quadrature

In this section we describe the procedure to compute the integral of a function, $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, on the surface of the

unit sphere, $\mathcal{S}_2$, using global RBF interpolation. Given a set of N scattered points, $\{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathcal{S}_2$, our goal is to compute the quadrature weights, w_i , to approximate the value of the integral as

$$\int_{\mathcal{S}_2} f \, dS \approx \sum_{j=1}^N w_j f(\mathbf{x}_j). \quad (1)$$

We start by computing the RBF interpolant of the function

$$f(\mathbf{x}) \approx s(\mathbf{x}) = \sum_{i=1}^N c_i \psi_i(\|\mathbf{x} - \mathbf{x}_i\|), \quad (2)$$

where $\psi_i(\|\mathbf{x} - \mathbf{x}_i\|)$ is an RBF.

The coefficients c_i are computed by imposing the interpolation conditions, $s(\mathbf{x}_i) = f(\mathbf{x}_i)$, so that

$$\mathbf{c} = A^{-1} \mathbf{f}. \quad (3)$$

Here, $\mathbf{c}$ and $\mathbf{f}$ are vectors containing the values of c_i and $f(\mathbf{x}_i)$, respectively, and A^{-1} is the inverse of the RBF interpolation matrix

$$A = \begin{bmatrix} \psi_1(0) & \psi_2(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \dots & \psi_N(\|\mathbf{x}_1 - \mathbf{x}_N\|) \\ \psi_1(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \psi_2(0) & \dots & \psi_N(\|\mathbf{x}_2 - \mathbf{x}_N\|) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(\|\mathbf{x}_N - \mathbf{x}_1\|) & \psi_2(\|\mathbf{x}_N - \mathbf{x}_2\|) & \dots & \psi_N(0) \end{bmatrix}. \quad (4)$$

Then, from (2), it follows that

$$\begin{aligned} \int_{\mathcal{S}_2} f \, dS &\approx \int_{\mathcal{S}_2} s \, dS \\ &= \sum_{i=1}^N c_i \int_{\mathcal{S}_2} \psi_i(\|\mathbf{x} - \mathbf{x}_i\|) \, dS. \end{aligned} \quad (5)$$

Since all functions $\psi_i(\|\mathbf{x} - \mathbf{x}_i\|)$ are identical (only their centers are displaced to different locations), all the integrals in (5) have the same value. For instance, if we use Gaussians

$$\psi_i(\|\mathbf{x} - \mathbf{x}_i\|) = e^{-\epsilon^2 \|\mathbf{x} - \mathbf{x}_i\|^2}, \quad (6)$$

then

$$\int_{\mathcal{S}_2} \psi_i(\|\mathbf{x} - \mathbf{x}_i\|) \, dS = \frac{\pi}{\epsilon^2} \left(1 - e^{-4\epsilon^2}\right). \quad (7)$$

Values of this integral for other commonly used RBFs are shown in Table 1. It includes infinitely smooth RBFs (GA, MQ, IMQ), piecewise smooth RBFs (PHS) and compact support RBFs (Wendland). Piecewise smooth RBFs have a singularity at the origin, and compactly supported RBFs

Table 1. Exact values of the integral over the unit sphere of some common RBFs: GA (Gaussian), MQ (multiquadric), IMQ (inverse multiquadric), PHS (polyharmonic splines) and CS (Wendland compact support).

	RBF ψ	$\int_{S_2} \psi$
GA	$e^{-\epsilon^2 r^2}$	$\frac{\pi}{\epsilon^2} (1 - e^{-4\epsilon^2})$
MQ	$\sqrt{1 + \epsilon^2 r^2}$	$\frac{2\pi}{3\epsilon^2} ((1 + 4\epsilon^2)^{3/2} - 1)$
IMQ	$1 / \sqrt{1 + \epsilon^2 r^2}$	$\frac{2\pi}{\epsilon^2} ((1 + 4\epsilon^2)^{1/2} - 1)$
PHS	r^k	$\frac{2^k 8\pi}{k+2}$
PHS	$r^k \log r$	$\frac{2^{k+3}\pi}{k+2} \left(\log(2) - \frac{1}{k+2} \right)$
CS	$(1-r)_+^4 (4r+1)$	0.4409

have also a singularity at $r = 1$. This leads to algebraic, rather than spectral accuracy of interpolation.

Using (7) in (5) yields the following result for the surface integral:

$$\int_{S_2} f dS \approx \frac{\pi}{\epsilon^2} (1 - e^{-4\epsilon^2}) \left(\sum_{i=1}^N c_i \right). \quad (8)$$

Finally, if we denote by d_{ij} the (i, j) element of the matrix A^{-1} , from (3)

$$\begin{aligned} \sum_{i=1}^N c_i &= \sum_{i=1}^N \left(\sum_{j=1}^N d_{ij} f(\mathbf{x}_j) \right) \\ &= \sum_{j=1}^N f(\mathbf{x}_j) \left(\sum_{i=1}^N d_{ij} \right), \end{aligned} \quad (9)$$

so that each quadrature weight is just the sum of the corresponding row of A^{-1} multiplied by a constant which is the integral over the sphere of the RBF. This is the main observation of the paper. For instance, in the case of Gaussians, we obtain, using (7),

$$w_i = \left(\sum_{j=1}^N d_{ij} \right) \frac{\pi}{\epsilon^2} (1 - e^{-4\epsilon^2}). \quad (10)$$

Since the integral of the RBFs is exact, the accuracy of the quadrature will only depend on the interpolation accuracy. In this respect, it should be pointed out that some of the RBFs depend on a shape parameter ϵ , and it is well known that the accuracy of the interpolation increases with decreasing ϵ . In fact, it was shown by Fornberg and Piret (2007; 2008) that the $\epsilon \rightarrow 0$ is equivalent to the direct use of spherical harmonics. Numerically, however, there is a critical value of ϵ below which the interpolation matrix

becomes ill-conditioned and the accuracy deteriorates rapidly.

To analyze the accuracy of the proposed methods, we will use the following three test functions employed by other authors (Fornberg and Martel, 2014; Sommariva and Womersley, 2005; Fuselier *et al.*, 2014; Reeger and Fornberg, 2016; Reeger *et al.*, 2016; Beentjes, 2015); see Fig. 1:

$$f_1(x, y, z) = 1 + x + y^2 + x^2 y + x^4 + y^5 \quad (11)$$

$$+ x^2 y^2 z^2, \quad (12)$$

$$f_2(x, y, z) = \frac{1}{9} [1 + \tanh(-9x - 9y + 9z)], \quad (13)$$

$$f_3(x, y, z) = \frac{1}{9} [1 + \text{sign}(-9x - 9y + 9z)]. \quad (14)$$

Function f_1 contains only SPH modes up to degree 6, whereas f_2 and f_3 feature increasingly slowly converging SPH expansions.

The exact values of the integrals over the unit sphere are $216\pi/35$ for function f_1 , and $4\pi/9$ for functions f_2 and f_3 .

In the following, we use two types of node sets: minimal energy (ME) nodes and Halton nodes. The coordinates of ME nodes are taken from the tabulated values of Womersley and Sloan (2003). Halton nodes (Halton, 1960) are quasi-uniformly scattered nodes which are obtained from a set of Halton nodes in a square $(t_i, z_i) \in [-1, 1] \times [-1, 1]$ by

$$x_i = \sqrt{1 - z_i^2} \cos(t_i \pi),$$

$$y_i = \sqrt{1 - z_i^2} \sin(t_i \pi),$$

$$z_i = z_i.$$

Figure 2 shows the quadrature error for function f_1 with ME (solid lines) and Halton nodes (dotted lines)

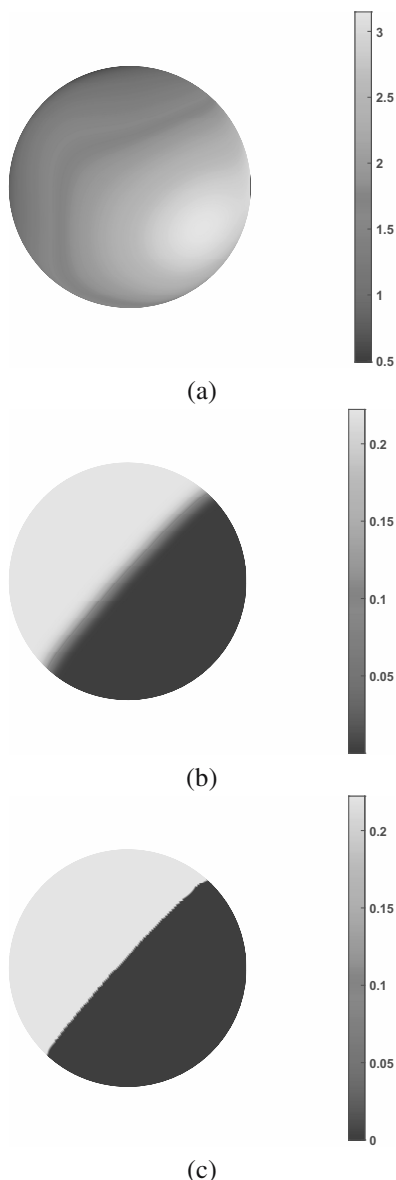


Fig. 1. Test functions f_1 (12), f_2 (13), f_3 (14).

as a function of the shape parameter ϵ . We plot the average quadrature error using 50 different node sets which are obtained by rotating the original node set a random angle around the z axis. We do this in order to eliminate misleading sharp *dips* that appear when using a single node set and which simply reflect a change of sign in the error for a certain value of ϵ . Here, and throughout the paper (unless explicitly specified), we have used Gaussians as RBFs. Notice that the error decreases with decreasing ϵ until the interpolation matrix becomes ill-conditioned and the errors start to grow. These small ϵ instabilities that appear in this and in all other calculations in the paper can be eliminated using one of several *stable algorithms* that have been proposed in the past. In

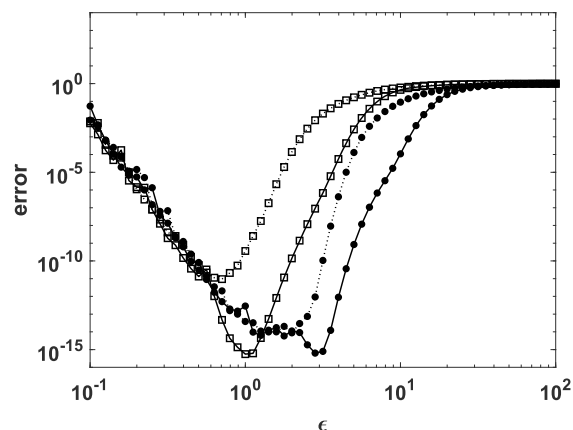


Fig. 2. Average relative quadrature error in 50 rotated node sets for function f_1 as a function of ϵ (solid-squares: 225 ME nodes, dotted-squares: 225 Halton nodes, solid-circles: 1296 ME nodes, dotted-circles: 1296 Halton nodes).

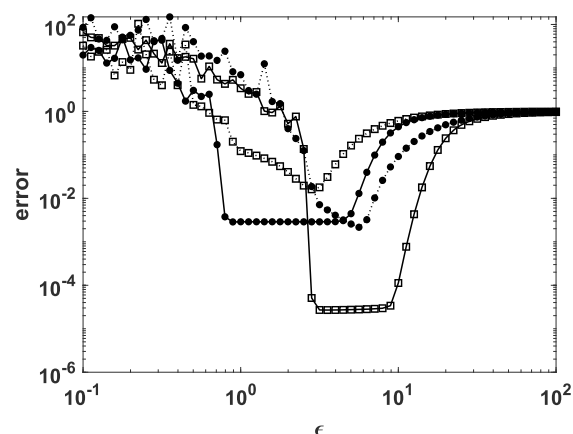


Fig. 3. Average quadrature relative error in 50 rotated node sets for function f_2 as a function of ϵ (solid-squares: 225 ME nodes, dotted-squares: 225 Halton nodes, solid-circles: 1296 ME nodes, dotted-circles: 1296 Halton nodes).

particular, the RBF-QR by Fornberg and Piret (2007) was shown to be very efficient for interpolation in a spherical surface. We do not use them in the computations of this paper because that of ϵ for which the minimum error is reached is higher than the value of ϵ for which the method becomes unstable. Notice also that with the appropriate value of the shape parameter ϵ the accuracy, in the case of ME nodes, is close to machine precision. This fact could be expected since function $f_1(x, y, z)$ (12) contains SPH only up to order 6 and, therefore, whenever $N \geq 49$, the SPH interpolation error should be zero. Since $\epsilon \rightarrow 0$ is equivalent to SPH, the error for function $f_1(x, y, z)$ should be zero in this limit.

Figures 3 and 4 show similar results for functions f_2

Table 2. Quadrature relative errors for ME nodes.

	225			1296		
	f_1	f_2	f_3	f_1	f_2	f_3
SPH	0	$2.7 \cdot 10^{-3}$	$5.2 \cdot 10^{-3}$	0	$6.9 \cdot 10^{-5}$	$2.4 \cdot 10^{-3}$
RBF global (8)	$5.6 \cdot 10^{-16}$	$2.9 \cdot 10^{-3}$	$9.0 \cdot 10^{-3}$	$6.4 \cdot 10^{-16}$	$2.7 \cdot 10^{-5}$	$2.2 \cdot 10^{-3}$
RBF local (Reeger and Fornberg, 2016)	$1.2 \cdot 10^{-3}$	$4.2 \cdot 10^{-3}$	$2.7 \cdot 10^{-3}$	$4.4 \cdot 10^{-7}$	$4.2 \cdot 10^{-5}$	$3.1 \cdot 10^{-3}$
RBF local (22)	$1.3 \cdot 10^{-5}$	$2.2 \cdot 10^{-3}$	$7.5 \cdot 10^{-3}$	$9.6 \cdot 10^{-7}$	$1.7 \cdot 10^{-4}$	$2.8 \cdot 10^{-3}$

Table 3. Quadrature relative errors for Halton nodes.

	225			1296		
	f_1	f_2	f_3	f_1	f_2	f_3
SPH	$1.8 \cdot 10^{-16}$	$6.8 \cdot 10^{-2}$	$3.5 \cdot 10^{-1}$	0	$9.6 \cdot 10^{-3}$	$1.7 \cdot 10^{-1}$
RBF global (8)	$9.5 \cdot 10^{-12}$	$1.6 \cdot 10^{-2}$	$6.9 \cdot 10^{-2}$	$9.2 \cdot 10^{-15}$	$2.1 \cdot 10^{-3}$	$4.6 \cdot 10^{-2}$
RBF local (Reeger and Fornberg, 2016)	$1.5 \cdot 10^{-3}$	$1.3 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	$3.7 \cdot 10^{-7}$	$4.3 \cdot 10^{-4}$	$4.4 \cdot 10^{-3}$
RBF local (22)	$2.9 \cdot 10^{-5}$	$6.7 \cdot 10^{-3}$	$1.7 \cdot 10^{-2}$	$6.0 \cdot 10^{-7}$	$5.4 \cdot 10^{-4}$	$4.8 \cdot 10^{-3}$

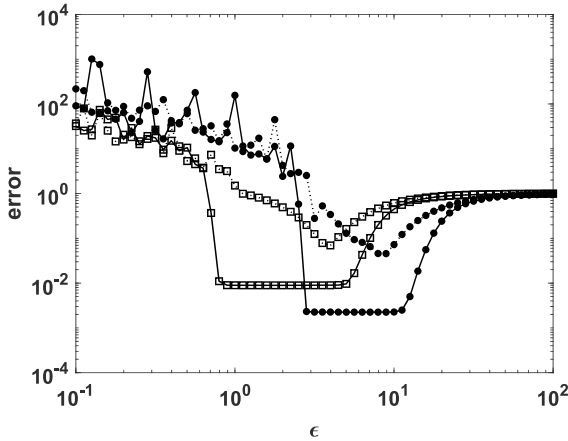


Fig. 4. Average quadrature relative error in 50 rotated node sets for function f_3 as a function of ϵ (solid squares: 225 ME nodes, dotted squares: 225 Halton nodes, solid circles: 1296 ME nodes, dotted circles: 1296 Halton nodes).

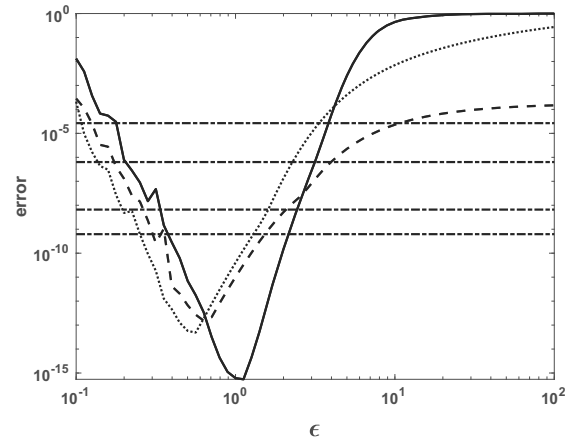


Fig. 5. Average quadrature relative error in 50 rotated node sets for function f_1 as a function of ϵ using 225 ME nodes (solid: GA, dashed: MQ, dotted: IQ, dot-dashed: PHS (r^7 , r^5 , r^3 , $r \log r$ from less to more error)).

and f_3 . Notice that, in the case of ME nodes, the error decreases with decreasing ϵ until a minimum is reached and the error remains constant for a wide range of ϵ . For still lower values of ϵ the interpolation matrix becomes ill-conditioned and the error increases. For Halton nodes there is a specific value of ϵ for which the error is minimum and, for lower values, the error increases.

The second row of Tables 2 and 3 shows the relative error using the RBF global method with ME and Halton nodes. For comparison purposes the first row shows the corresponding results using SPH interpolation. Notice that the accuracies obtained with appropriate values of the shape parameter are similar to those obtained using SPH interpolation (Fornberg and Martel, 2014).

Figure 5 compares the average quadrature error

using 50 rotated node sets for different RBFs. The integral is computed using (8) with the value of the integral of each RBF shown in Table 1. In the case of polyharmonic splines, PHS (defined in Table 1 and shown with dot-dashed lines in Fig. 5), the error decreases with increasing power. Also notice that Gaussian RBFs behave best.

3. Local RBF based quadrature

When the function to be integrated is not smooth but changes over small scales, a large number of nodes are needed in order to achieve good accuracy. The global RBF is computationally too expensive in these situations, so we adapt the ideas described in the previous section to

a local method that divides the surface into small patches and finds the weights corresponding to each of them.

Reeger and Fornberg (2016) as well as Reeger *et al.* (2016) recently proposed a method to compute quadrature weights in the sphere using an RBF-FD approach. The method can be summarized in the following four steps:

1. Create a Delaunay spherical triangulation on the surface of the sphere.
2. Project each triangle, together with some nearby nodes, to a tangent plane.
3. Find quadrature weights over the local tangent plane node set for the projected planar triangle.
4. Combine the weights for the individual triangles to obtain the full weight set for the sphere.

The third row of Tables 2 and 3 shows the relative error using this method with ME and Halton nodes.

These results have been obtained using Reeger's programs (Reeger, 2015). Their accuracy is similar to that obtained with the proposed method (8), which uses global quadrature weights.

We propose an alternative algorithm that avoids projecting on the tangent plane. The main steps of the algorithm are as follows:

1. Divide the sphere into N_R rectangles in the latitude-azimuth coordinate system (θ, ϕ) with approximately the same area.
2. For each rectangle find the N_v nearest neighbors to the midpoint of the rectangle.
3. Integrate over each rectangle using RBF interpolation to find local quadrature weights.
4. Combine the integrals for the individual rectangles to obtain the total weights for complete integral over the sphere.

We describe these steps in more detail below.

3.1. Partition of the sphere into N_R rectangles.

We divide the sphere into N_R rectangles in the latitude, azimuth coordinate system. The number of rectangles N_R should be selected so that the average number of nodes in each rectangle is reasonable: large enough to allow for an accurate interpolation, but small enough for the size of the interpolation matrix to be small. In fact, we fix the average number of nodes per rectangle, N_p , and we compute the number of rectangles as $N_R = N/N_p$, where N is the total number of nodes. For a unit sphere, the area of each rectangle is $A_R = 4\pi/N_R$. We consider rectangles of equal width in the latitude direction $\Delta\theta = \pi/2 - \arcsin(1 - A_R/(2\pi))$. In the azimuth direction, we take

$(\Delta\phi)_i = A_R/|\sin(\theta_{i,0}) - \sin(\theta_{i,1})|$. Figure 6 shows the latitude-azimuth coordinate system for an example of Step 1 of the algorithm for the case $N = 225$ and $N_p = 3$. There are 74 rectangles covering the surface of the sphere, and a set of $N = 225$ ME nodes marked with dots.

3.2. Finding N_v nearest neighbours to the midpoint of a rectangle. Figure 6 also shows $N_v = 15$ nearest neighbors to the midpoint of one of the rectangles. The N_v nearest neighbors are marked with circles.

3.3. Local quadrature weights. We integrate over each rectangle using RBF interpolation to find local quadrature weights. Let $v(k, i)$ with $k = 1, \dots, N_R$ and $i = 1, \dots, N_v$ be the set of indices of the N_v nearest nodes to the midpoint of rectangle k . The local interpolator of f over this rectangle takes the form

$$s_k(\mathbf{x}) = \sum_{i=1}^{N_v} c_{k,v(k,i)} \psi_{v(k,i)}(\|\mathbf{x} - \mathbf{x}_{v(k,i)}\|), \quad (15)$$

where the coefficients, $c_{k,v(k,i)}$, are obtained by imposing the interpolation conditions: $s_k(\mathbf{x}_{v(k,i)}) = f(\mathbf{x}_{v(k,i)})$. Then, similarly to what was done in the global case (5), the integral over this rectangle can be approximated as

$$I_k = \int_{S_k} f \, dS \approx \int_{S_k} s_k \, dS = \sum_{i=1}^{N_v} c_{k,v(k,i)} \int_{S_k} \psi_{v(k,i)}(\|\mathbf{x} - \mathbf{x}_{v(k,i)}\|) \, dS, \quad (16)$$

where S_k is the domain of rectangle k . To approximate the values of these integrals we note that the RBFs depend

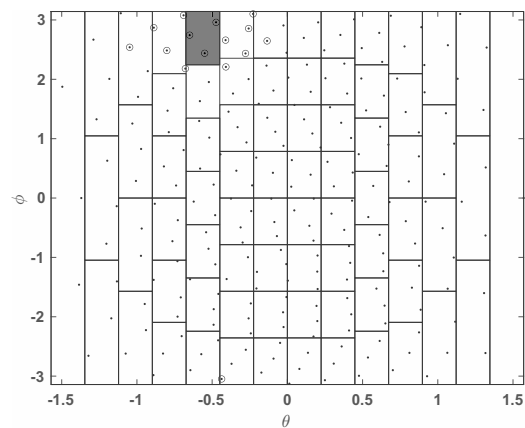


Fig. 6. Portrayal of 225 nodes (dots) represented on a latitude-azimuth coordinate system, along with the division of the sphere into 74 rectangles and $N_v = 15$ nearest neighbors (circles) of the midpoint of one of the rectangles (shaded).

on a single variable which is the distance to the RBFs. In spherical coordinates, the square of the distance to this center, $\mathbf{x}_{v(k,i)} = (\phi_{v(k,i)}, \theta_{v(k,i)})$, is

$$r^2 = 2 - 2[\cos(\theta) \cos(\theta_{v(k,i)}) \cos(\phi - \phi_{v(k,i)}) + \sin(\theta) \sin(\theta_{v(k,i)})]. \quad (17)$$

For a rectangle $[\theta_{k,0}, \theta_{k,1}] \times [\phi_{k,0}, \phi_{k,1}]$, we have to compute the integrals

$$I_{k,v(k,i)} = \int_{\theta_{k,0}}^{\theta_{k,1}} \int_{\phi_{k,0}}^{\phi_{k,1}} \psi_{v(k,i)}(r(\theta, \phi)) \cos \theta d\theta d\phi. \quad (18)$$

Thus, following a similar approach to that used to derive (9), the integral of f over the rectangle k , I_k , can be approximated as

$$\begin{aligned} I_k &\approx \sum_{i=1}^{N_v} w_{k,v(k,i)} f(\mathbf{x}_{v(k,i)}) \\ &= \sum_{i=1}^{N_v} c_{k,v(k,i)} I_{k,v(k,i)}. \end{aligned} \quad (19)$$

Similarly to (10), the local quadrature weights, $w_{k,v(k,i)}$, are

$$w_{k,v(k,i)} = \sum_{j=1}^{N_v} d_{k,ij} I_{k,v(k,j)}, \quad (20)$$

where $d_{k,ij}$ is now the (i, j) element of the inverse of the RBF interpolation matrix restricted to the nearest nodes to rectangle k whose elements are given by $A_k = [a_{k,ij}] = [\psi_{v(k,i)}(\|\mathbf{x}_{v(k,j)} - \mathbf{x}_{v(k,i)}\|)]$ ($i, j = 1, \dots, N_v$). Unfortunately, the integrals (18) cannot be computed analytically for the RBFs in Table 1. Therefore, we compute them numerically using Matlab's `quad2d` function. However, since $\epsilon^2 r^2 \ll 1$ because we use stencils with closest nodes, we can compute approximations to the integrals (18) analytically by approximating the RBFs with a Taylor polynomial in powers of $(\epsilon r)^2$. For instance, in the case of Gaussians,

$$\begin{aligned} I_{k,v(k,i)} &\approx \int_{\theta_{k,0}}^{\theta_{k,1}} \int_{\phi_{k,0}}^{\phi_{k,1}} 1 - (\epsilon r)^2 \\ &\quad + \frac{1}{2}(\epsilon r)^4 \dots \cos \theta d\theta d\phi \end{aligned} \quad (21)$$

with r^2 in terms of θ and ϕ given by (17). The integrals of powers of r^2 appearing in (21) can be

computed analytically with, for instance, Matlab's symbolic capabilities.

The main advantage of using (21) instead of the numerical integration of (18) is a significant increase in computational efficiency.

Table 4 shows the speedups obtained for different orders of approximation of the integral in (21) in the case of 1296 ME nodes, where it is necessary to use order $\mathcal{O}(\epsilon r)^6$ to obtain accuracies comparable to those obtained with `quad2d`. Thus, the speedup is 36.

3.4. Combining the integrals for the individual rectangles to obtain the total weights for the complete integral over the sphere. Finally, the total quadrature weights over the whole surface of the sphere are obtained by adding up the contribution of all rectangles; thus,

$$\begin{aligned} \int_{S_2} f dS &= \sum_{k=1}^{N_R} I_k \approx \sum_{k=1}^{N_R} \sum_{i=1}^{N_v} w_{k,v(k,i)} f(\mathbf{x}_{v(k,i)}) \\ &= \sum_{i=1}^N f(\mathbf{x}_i) \left(\sum_{k=1}^{N_R} w_{k,v(k,i)} \right), \end{aligned} \quad (22)$$

where $w_{k,v(k,i)} = 0$ if the node $\mathbf{x}_i$ is not contained in the vector $v(k, \cdot)$. Notice that the weight for the whole integral,

$$w_i = \sum_{k=1}^{N_R} w_{k,v(k,i)}, \quad (23)$$

is the sum of the local weights of rectangles for which the node $\mathbf{x}_i$ is considered a neighbor.

Figure 7 shows the error in the integration of function f_1 as a function of ϵ using $N_p = 3$, $N_v = 15$, and Gaussian RBFs.

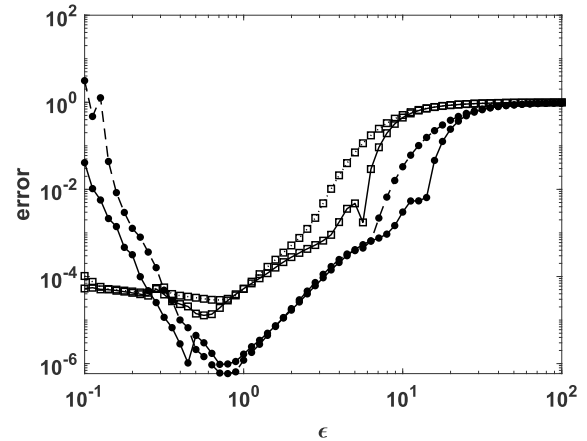


Fig. 7. Average quadrature relative error in 50 rotated node sets for function f_1 as a function of ϵ using $N_p = 3$, $N_v = 15$ (solid-squares: 225 ME nodes, dotted-squares: 225 Halton nodes, solid-circles: 1296 ME nodes, dotted-circles: 1296 Halton nodes).

Table 4. Speedups for different orders of approximation.

	<code>quad2d</code>	$\mathcal{O}(2)$	$\mathcal{O}(4)$	$\mathcal{O}(6)$	$\mathcal{O}(8)$
speedup	1	137	113	36	15

We plot the average quadrature error using 50 different node sets which are obtained by rotating the original node set randomly. Solid lines correspond to ME nodes and dotted lines to Halton nodes. Circles correspond to 1296 nodes and squares to 225 nodes. These results show that the method converges but is less accurate than the global one.

Figure 8 shows the corresponding results for functions f_2 (left) and f_3 (right). The fourth row of Tables 2 and 3 shows the relative error using the proposed local RBF method with ME and Halton nodes. These relative errors are computed using the value of ϵ that corresponds to the minimum error in Figs. 7 and 8. The accuracies are similar to those obtained with global RBFs (second row) and with the local RBF one proposed by Reeger and Fornberg (2016) (third row). It would appear that for Halton nodes the proposed method performs slightly better than the method of Reeger and Fornberg (2016). However, in the proposed method there are several parameters that have to be chosen (N_p , N_v and ϵ), and both the accuracy and the computational cost of the integral depend on their values. In fact, the computational cost increases with decreasing N_p (cost $\propto 1/N_p$). Therefore, the relevant item that should be considered to analyze the optimal parameters of the method is the error divided by N_p .

Figure 9 shows the average relative error divided by N_p as a function of N_p for the case $\epsilon = 1$ and $N_v = 15$ using 30 node sets obtained by random rotations of a set of 1296 ME nodes. Notice that the minimum of this relevant parameter occurs for $N_p = 4$ which, therefore, should be considered as the optimal operating point of the method in the case of 1296 ME nodes. Similar results are obtained with 225 ME nodes and with 225 and 1296 Halton nodes. Thus, we assume that using $N_p = 4$ is, in general, the best choice for the proposed method since it represents a good balance between computational cost and accuracy.

Figure 10 shows the average relative error as a function of N_v , for the case $\epsilon = 1$ and $N_p = 4$ using 30 node sets obtained by random rotations of a set of 1296 ME nodes. The minimum relative error is $3.6 \cdot 10^{-7}$ for function f_1 , $6.0 \cdot 10^{-5}$ for function f_2 , and $1.4 \cdot 10^{-3}$ for function f_3 . Notice that for function f_1 the relative error is smaller for values of N_v between 15 and 20. For functions f_2 and f_3 the error is nearly constant for N_v between 12 and 25. Since the computational cost grows with N_v (cost $\propto N_v^3$) we recommend, as a general rule, to use $N_v = 15$.

Figure 11 compares the relative error using different RBFs for the function f_1 . The data correspond to 225 ME nodes using $N_p = 4$ and $N_v = 15$. Notice that Gaussians, multiquadrics and inverse multiquadric yield comparable results. Also shown is the error using the r^3 PHS. Other PHS (r^5 , r^7 , $r \log r$) are less accurate. Also in the case of functions f_2 and f_3 the different RBFs yield very similar accuracy (including PHS r^3).

4. Node clustering

In this section, we describe an algorithm to compute ME nodes in the sphere in the cases of both equispaced and non-equispaced nodes. It is an iterative algorithm, which starts with a certain node distribution and tries to evolve to a minimum energy distribution in which the repulsive forces on each node are zero.

Uniformly distributed ME nodes are a set of nodes that minimize the I -energy

$$E(1) = \sum_{1 \leq j < k \leq N} \|\mathbf{x}_j - \mathbf{x}_k\|^{-1}, \quad s > 0. \quad (24)$$

In the limit $N \rightarrow \infty$, the minimum of the I -energy is (Saff and Kuijlaar, 1997)

$$\min E(1) \approx \frac{1}{2} N^2 - 0.55305 N^{3/2}. \quad (25)$$

The first step of the proposed algorithm computes the force acting on each node which is the sum of all the repulsion forces exercised by the rest of the nodes. We assume that the repulsion force between nodes i and j is a force in the direction $(\mathbf{x}_j - \mathbf{x}_i)$ with magnitude inversely proportional to the square of the distance between the nodes. We let the force to be proportional to the product of charges (q) in order to allow non-homogeneous node distributions. Thus, the repulsion force between nodes i and j is

$$\mathbf{f}_{i,j} = \frac{1}{\|\mathbf{x}_j - \mathbf{x}_i\|^2} \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|} q_i q_j, \quad (26)$$

$$\mathbf{f}_j = \sum_{i=1, i \neq j}^N \mathbf{f}_{i,j}. \quad (27)$$

To find the displacement of node j , we compute the component of force $\mathbf{f}_j$ on the tangent plane as

$$\mathbf{t}_j = \mathbf{f}_j - (\mathbf{f}_j \cdot \mathbf{x}_j) \mathbf{x}_j. \quad (28)$$

The second step of the algorithm is to find these displacements. To this end, we move each node by an amount of $c \mathbf{t}_j$ and then use a gnomonic projection to bring back the node to the surface of the sphere. Thus,

$$\mathbf{x}_j = \frac{\mathbf{x}_j + c \mathbf{t}_j}{\|\mathbf{x}_j + c \mathbf{t}_j\|}. \quad (29)$$

The iterative procedure stops when the maximum displacement is below a certain threshold or when a maximum number of iterations is reached. The constant c has to be carefully chosen. It should be small enough to prevent crossing between nodes, but high enough to move nodes a significant amount in each iteration. Starting from a random distribution of $N = 225$ equally charged nodes

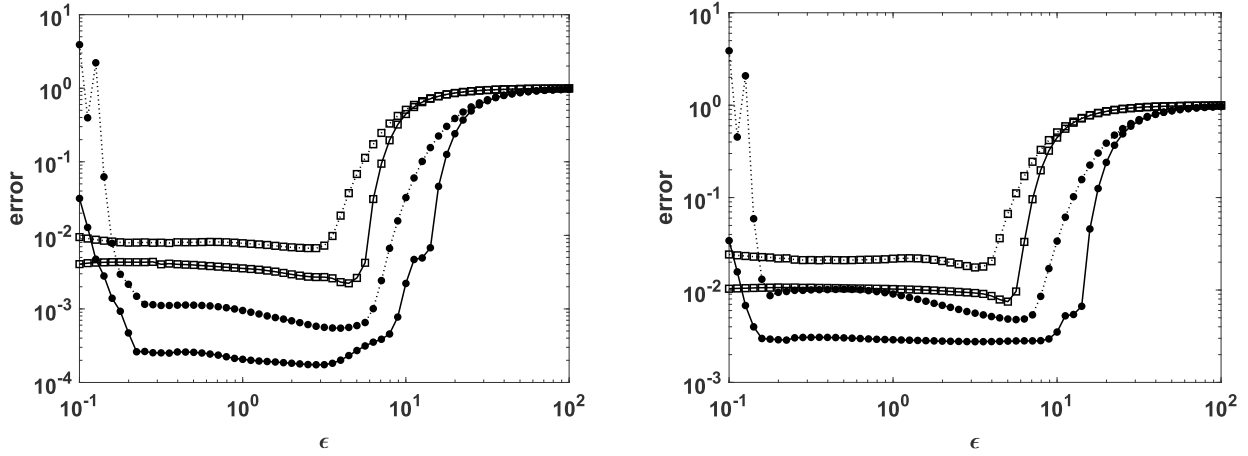


Fig. 8. Average quadrature relative error in 25 rotated node sets for functions f_2 (left) and f_3 (right) as a function of ϵ using $N_p = 3$, $N_v = 15$ (solid squares: 225 ME nodes, dotted squares: 225 Halton nodes, solid circles: 1296 ME nodes, dotted circles: 1296 Halton nodes).

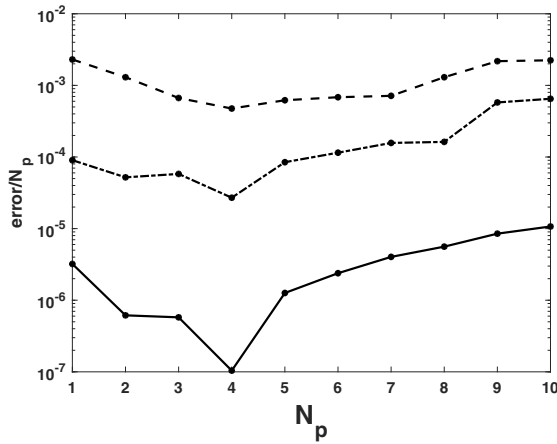


Fig. 9. Average quadrature relative error divided by N_p as a function of N_p in 30 node sets obtained by random rotations of a set of 1296 ME nodes: $\epsilon = 1$, $N_v = 15$, as well as functions f_1 (solid line), f_2 (dashed line), and f_3 (dash-dotted line).

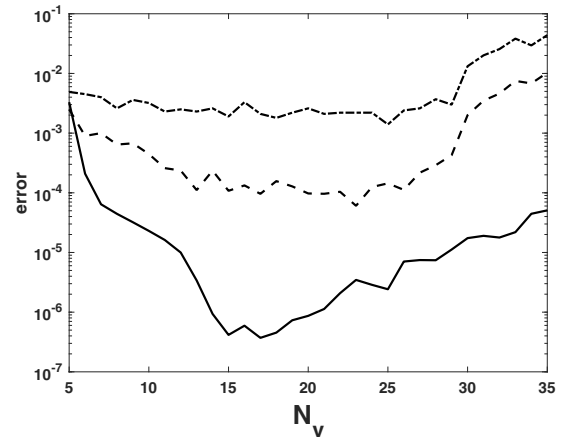


Fig. 10. Average quadrature relative error as a function of N_v in 30 node sets obtained by random rotations of a set of 1296 ME nodes: $\epsilon = 1$, $N_p = 4$, as well as functions f_1 (solid line), f_2 (dashed line), and f_3 (dash-dotted line).

($q_j = 1$), with a I -energy of 24788 and using the above algorithm, we reach a I -energy local minimum of 23451, which is very close to the asymptotic limit $\min E(1) = 23449$ (25). For this configuration of nodes, the distances to the closest node lie in the interval $[0.2232, 0.2509]$, and the maximum force ($\max \mathbf{t}_j$) is 0.0063.

The values of the charges q_j can be used to achieve a non-uniform distribution of nodes. For instance, in functions f_2 and f_3 there is a rapid variation in the values of the function for points in the vicinity of the plane $x + y - z = 0$. Thus, it could be useful to concentrate

nodes in this zone in order to increase the accuracy of the RBF interpolation. To this end, we have associated to each node j a charge q_j depending on the variation in the function $f(\mathbf{x})$ in its vicinity. This variation has been estimated by the quantity

$$\Theta_j = \frac{1}{N_\delta} \sum_{k=1}^{N_\delta} \frac{|f(\mathbf{x}_j) - f(\mathbf{x}_{\delta(k)})|}{\|\mathbf{x}_j - \mathbf{x}_{\delta(k)}\|}, \quad (30)$$

where $\mathbf{x}_{\delta(k)}$, $k = 1, 2, \dots, N_\delta$, are the coordinates of the N_δ neighbor points, such as $\|\mathbf{x}_j - \mathbf{x}_{\delta(k)}\| \leq \delta$, and, finally,

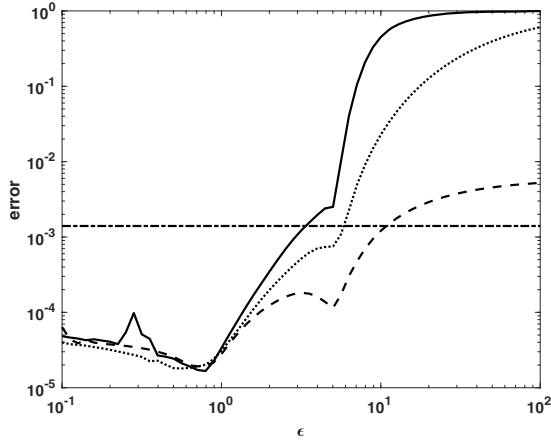


Fig. 11. Average quadrature relative error as a function of ϵ in 20 node sets obtained by random rotations of a set of 225 ME nodes for function f_1 . Solid: GA, dashed: MQ, dotted: IQ, dot-dashed: PHS (r^3).

the charge q_j has been calculated as

$$q_j = 1 - \mu \frac{\Theta_j - \Theta_{\min}}{\Theta_{\max} - \Theta_{\min}}. \quad (31)$$

Here, $\Theta_{\min}$ and $\Theta_{\max}$ are the minimum and maximum values of Θ_j , respectively. The parameters δ and μ control the degree of clustering: the bigger the variation of the function, the lower the charge, and, hence, the density of points increases. It should be pointed out that there is a trade-off between the stability of matrix inversion and the optimal node selection. In fact, increasing node clustering in the vicinity of discontinuities leads to an increase in the accuracy of the interpolation. However, if nodes are very close together, the condition number increases and, therefore, accuracy deteriorates due to round-off errors.

Figure 12 shows the result of applying the algorithm just described to a set of 1296 nodes for function f_2 . The charges in each node are computed through (31) using $\delta = 0.1$ and $\mu = 0.5$, so that $q_i q_j \approx 1$ for points far the plane $x + y - z = 0$ and $q_i q_j \approx 0.16$ for points lying in the plane. The constant c in (29) is taken as

$$c = 0.1 \frac{\min_{i,j,i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|}{\max_j \|\mathbf{t}_j\|}. \quad (32)$$

Figure 13 shows the corresponding histogram of distances to the nearest neighbor.

Applying the method described in Section 3 to compute the quadrature of the function f_3 (14) results in the error versus ϵ dependence shown in Fig. 14. The results shown with a dashed line represent the average error using the nodes in Fig. 12 and other 50 node sets, which are obtained by rotating it around the axis

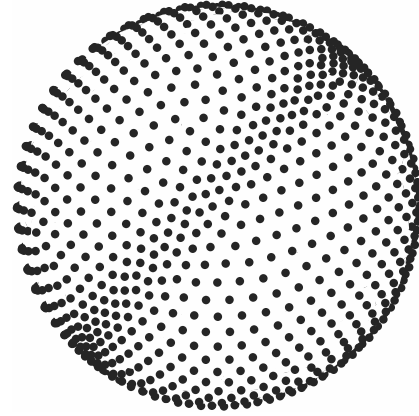


Fig. 12. Portrayal of 1296 nodes distributed using Coulomb's law with node charges according to (31).

$[1, 1, -1]$ which is normal to the plane of discontinuity of function f_3 . Notice that there is a significant improvement in accuracy using the nodes in Fig. 14 in comparison with standard ME nodes (Fig. 8, right). In fact, the relative error is $7.2 \cdot 10^{-4}$ versus $2.9 \cdot 10^{-3}$ without node clustering (see the fourth row of Table 2). For the case of 225 nodes the improvement is smaller.

5. Conclusions

In this paper we presented a new RBF-FD based technique to compute quadrature and quadrature weights in the sphere. In the proposed method, the sphere is divided into rectangles in the latitude-azimuth coordinate system, and the function is integrated over each rectangle using RBF interpolation. The method is easy to implement and its accuracy is comparable to that based on SPH expansions. For comparison purposes, we also computed quadrature on the sphere using a global RBF method. One advantage of the proposed method over SPH expansions-based techniques is its ability to handle non-uniform node distributions. With respect to that we proposed a new algorithm to cluster nodes in regions of steep changes in the function. It is a repulsion-based algorithm with a non-uniform distribution of electrical charges. We applied the proposed method to compute quadrature using this type of clustered node distributions and we found that there was a significant improvement in accuracy when nodes were concentrated in regions where the function changes abruptly. It should be pointed out that the proposed method can be readily applied to other closed surfaces that can be parametrized.

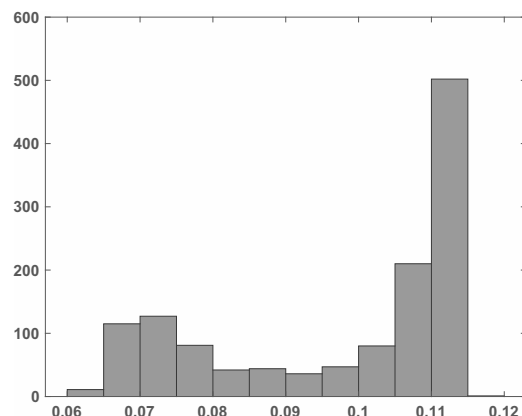


Fig. 13. Histogram of distances to nearest neighbor for the case of 1296 nodes and function f_2 . We used $\delta = 0.1$ and $\mu = 0.5$.

Acknowledgment

The authors acknowledge the support from the Spanish Ministry of Economy and Business (project FIS2016-77892-R).

References

- Ahrens, C. and Beylkin, G. (2009). Rotationally invariant quadratures for the sphere, *Proceedings of the Royal Society A* **465**(2110): 3103–3125.
- Atkinson, K. (1982). Numerical integration on the sphere, *Journal of the Australian Mathematical Society B* **23**(3): 332–347.
- Bazant, Z. and Oh, B. (1986). Efficient numerical integration on the surface of a sphere, *Journal of Applied Mathematics and Mechanics* **66**(11): 37–49.
- Beentjes, C. (2015). Quadrature on a spherical surface, <https://cbeentjes.github.io/files/Ramblings/QuadraturesSphere.pdf>.
- Bruno, O.P. and Kunyansky, L.A. (2001). A fast, high order algorithm for the solution of surface scattering problems: Basic implementation, tests and applications, *Journal of Computational Physics* **169**(1): 80–110.
- Flyer, N. and Fornberg, B. (2011). Radial basis functions: Developments and applications to planetary scale flows, *Computers and Fluids* **46**(1): 23–32.
- Flyer, N., Lehto, E., Blaise, S., Wright, G. and St-Cyr, A. (2012). A guide to RBF-generated finite differences for nonlinear transport: Shallow water simulations on a sphere, *Journal of Computational Physics* **231**(11): 4078–4095.
- Flyer, N., Wright, G. and Fornberg, B. (2014). Radial basis function generated finite differences: A mesh-free method for computational geosciences, in M. Freeden *et al.* (Eds), *Handbook of Geomathematics*, Springer-Verlag, Berlin, pp. 2535–2669.
- Fornberg, B. and Flyer, N. (2015). *A Primer on Radial Basis Functions with Applications to the Geosciences*, Society for Industrial and Applied Mathematics, Philadelphia.
- Fornberg, B. and Martel, J. (2014). On spherical harmonics based numerical quadrature over the surface of a sphere, *Advances in Computational Mathematics* **40**(5–6): 1169–1184.
- Fornberg, B. and Piret, C. (2007). A stable algorithm for flat radial basis function on a sphere, *SIAM Journal on Scientific Computing* **30**(1): 60–80.
- Fornberg, B. and Piret, C. (2008). On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere, *Journal of Computational Physics* **227**(5): 2758–2780.
- Fuselier, E., Hangelbroek, T., Narcowich, F., Ward, J. and Wright, B. (2014). Kernel based quadrature on spheres and other homogeneous spaces, *Numerische Mathematik* **127**(1): 57–92.
- Halton, J. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, *Numerische Mathematik* **1**(1): 84–90.
- Hesse, K., Sloan, I. and Womersley, R. (2010). Numerical integration on the sphere, in M. Freeden *et al.* (Eds), *Handbook of Geomathematics*, Springer-Verlag, Berlin, pp. 1187–1219.
- Klöckner, A., Barnett, A., Greengard, L. and O’Neil, M. (2013). Quadrature by expansion: A new method for the evaluation of layer potentials, *Journal of Computational Physics* **252**: 332–349.
- Klinteberg, L. and Tornberg, A.K. (2016). A fast integral equation method for solid particles in viscous flow using quadrature by expansion, *Journal of Computational Physics* **326**: 420–445.

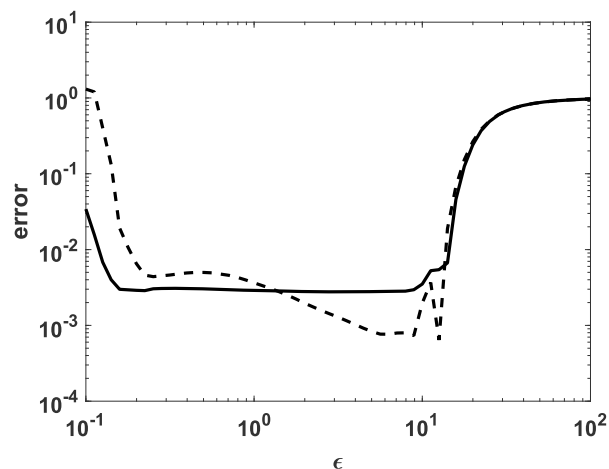


Fig. 14. Average quadrature relative error in 50 rotated node sets as a function of ϵ using the method described in Section 3 for function f_3 (14) with 1296 ME nodes (solid line) and the nodes shown in Fig. 12 (dashed line).

- Reeger, J. (2015). *Spherical_Quadrature_RBF (Quadrature_Nodes)*, <https://es.mathworks.com/matlabcentral/fileexchange/51214>.
- Reeger, J. and Fornberg, B. (2016). Numerical quadrature over the surface of a sphere, *Studies in Applied Mathematics* **137**(1): 174–188.
- Reeger, J., Fornberg, B. and Watts, L. (2016). Numerical quadrature over smooth, closed surfaces, *Proceedings of the Royal Society A* **472**(2194): 20160401.
- Saff, E. and Kuijlaar, A. (1997). Distributing many points on a sphere, *Proceedings of the Royal Society A* **19**(2): 5–11.
- Sommariva, A. and Womersley, R. (2005). Integration by RBF over the sphere, *Applied Mathematics Report amr05/17*, University of New South Wales, Sydney.
- Stroud, A. (1971). *Approximate Calculation of Multiple Integrals*, Prentice-Hall, Englewood Cliffs.
- Womersley, R. and Sloan, H. (2003). Interpolation and cubature on the sphere, <https://web.maths.unsw.edu.au/~rsw/Sphere/>.



Manuel Kindelan graduated in 1971 as an aeronautical engineer from Universidad Politécnica de Madrid. He received his PhD in applied mathematics from the University of California in San Diego in 1975. He worked at INTA (National Institute of Aerospace Technique) in 1975–1978, and at IBM in 1978–1992. In 1992 he joined Universidad Carlos III de Madrid, where he is now a full professor. His main research interest is in numerical methods for solving PDEs.



Pedro González-Rodríguez had studied for his bachelor's degree at Universidad Autónoma de Madrid and received his PhD in mathematics from Carlos III University in Madrid in 2005. Currently he is a visiting professor in the Mathematics Department there. His main research interests cover inverse problems related to medical imaging and numerical methods for solving PDEs.



Diego Álvarez graduated in physics from Universidad Autónoma de Madrid in 1993. He received his PhD in mathematical engineering from Carlos III University in Madrid in 2001. Currently he is an associate professor in the Mathematics Department there. His main research interests are in numerical methods for the solution of PDEs, inverse problems and mathematical methods related to medical imaging.

Received: 4 October 2021

Revised: 4 March 2022

Accepted: 11 April 2022