Numerical quadrature over the surface of a sphere

By Jonah A. Reeger [1] and Bengt Fornberg [2]

Large-scale simulations in spherical geometries require associated quadrature formulas. Classical approaches based on tabulated weights are limited to specific quasi-uniform distributions of relatively low numbers of nodes. By using a radial basis function-generated finite differences (RBF-FD) based approach, the proposed algorithm creates quadrature weights for N arbitrarily scattered nodes in only $O(N \log N)$ operations.

Key words quadrature, sphere, radial basis function, RBFs, RBF-FD

1 Introduction

An increasing number of applications, arising for example in geophysics, require PDEs to be solved in spherical geometries. Such calculations often need to be supplemented by numerical quadrature over spherical surfaces, in order to obtain integrated quantities, such as total energy, average temperature, etc. The node sets are typically very large, and feature spatially varying density for improved resolution in critical areas. This article presents a novel approach for calculating quadrature weights for such node sets.

Much of the literature on spherical quadrature has been focused on specific node sets, together with tabulated weights for select values of N (the total number of nodes) [1, 2, 3, 4, 5]. Among node sets not especially designed for quadrature, Minimal Energy (ME) is a common choice. These sets mimic the equilibrium positions of freely moving, repelling point charges.

[1] Captain, United States Air Force, supported by the Department of Defense. The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or U.S. Government. Address for correspondence: Air Force Institute of Technology, Department of Mathematics and Statistics, 2950 Hobson Way, Wright-Patterson Air Force Base, OH 45433-7765 USA.

[2] Address for correspondence: University of Colorado, Department of Applied Mathematics, 526 UCB, Boulder, CO 80309 USA.

Access to the computing resources of Andrew Geyer (Air Force Institutue of Technology) is gratefully acknowledged.

The calculation of quadrature weights based on spherical harmonics (SPH) interpolation often led to numerical instabilities [4, 6], prompting the development of 'Maximal Determinant' (MD) node sets. It was however noted in [7] that the issue with ME nodes could readily be overcome. Except for test functions that are described by extremely rapidly convergent SPH expansions, ME node sets then give comparable accuracy to Gaussian Quadrature approaches.

Following the works just cited, two remaining limitations were (i) the weight sets typically require about $O(N^3)$ operations and $O(N^2)$ memory to calculate, limiting N to around $N = 10^4$, and (ii) using SPH expansions, the node densities needed to be quasi-uniform. Analogously to truncated Fourier expansions in 1-D, SPH expansions can not feature spatially variable resolution abilities.

The RBF Lagrange function approach in [8] provides two major advances: the cost is brought down to $O(N^2)$ operations (still $O(N^2)$ memory) and, with no SPH involved, the resolution can be spatially variable. However, the order of accuracy is no longer spectral, but algebraic, with quadrature errors for sufficiently smooth functions typically $O(N^{-2})$; i.e., fourth order $O(h^4)$ accuracy in cases of quasi-uniform node sets with h a 'typical' node spacing.

The approach introduced here borrows its concept from RBF-generated finite differences (RBF-FD) [9, 10, 11, 12]. It again applies to spatially variable node sets, but with the cost further reduced to $O(N \log N)$ operations and $O(N)$ memory. This lowered operation count arises from having replaced the solution of a linear system of size $N \times N$ by instead solving $O(N)$ separate systems of size $n \times n$, where n is fixed (typically around 80). By changing some parameters, the quadrature accuracy for smooth functions can be adjusted. With our default setting, it becomes $O(N^{-3.5})$, corresponding to $O(h^7)$.

Section 2 starts with an outline of the four key steps in the proposed algorithm and then describes in subsections 2.1-2.4 each of these in more detail. Section 3 illustrates the performance of the proposed method in terms of accuracy and computational cost, and in comparison to the SPH and RBF Lagrange function approaches described in this introduction. To simplify comparisons, two of the three tests cases in Section 3 have been used in previous literature. Section 4 contains some concluding remarks.

2 A Description of the Method

We wish to approximate the surface integral of the scalar function $f(x, y, z)$ over the sphere surface $\mathbb{S}^2$ via

$$\mathcal{I}_{\mathbb{S}^2}(f) := \iint_{\mathbb{S}^2} f(x, y, z) dS \approx \sum_{i=1}^{N} W_i f(x_i, y_i, z_i)$$

where the node set $\mathcal{S}_N = \{(x_i, y_i, z_i)\}_{i=1}^N \subset \mathbb{S}^2$ is a set of (scattered) data sites located on the surface of the sphere. The proposed quadrature method can be summarized in four steps:

1. Given nodes ($\mathcal{S}_N$) on the sphere, create a spherical Delaunay triangulation.
2. For each of the triangles, project it, 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.

2.1 Step 1: Partition the Sphere Surface With a Spherical Delaunay Triangulation

A Delaunay triangulation $\mathcal{T} = \{\tau_k\}_{k=1}^K$ of $\mathbb{S}^2$ on the set $\mathcal{S}_N$ is a set of *spherical* triangles such that

- the triangle vertices are the elements of $\mathcal{S}_N$,
- the triangle edges are geodesics between the vertices of the triangles,
- no triangle contains an element of $\mathcal{S}_N$ other than its vertices,
- the interiors of the triangles are pairwise disjoint,
- the union of the set $\mathcal{T}$ covers $\mathbb{S}^2$, and
- no circumcircle of a triangle τ_k contains an element of $\mathcal{S}_N$ on its interior, guarding against triangles featuring poor aspect ratio

The number of triangles in a spherical Delaunay triangulation over $\mathcal{S}_N$ is $2N - 4$. This number follows immediately from Euler's polyhedron formula $V - E + F = 2$ (relating the number of vertices, edges, and faces, respectively). A more comprehensive definition of the spherical Delaunay triangulation and an $O(N \log N)$ algorithm for its construction can be found in [13].

The integral over the sphere is the sum of the integrals over the triangles: $\mathcal{I}_{\mathbb{S}^2}(f) = \sum_{k=1}^K \iint_{\tau_k} f(x, y, z) dS$. The proposed method will compute the surface integrals over each of

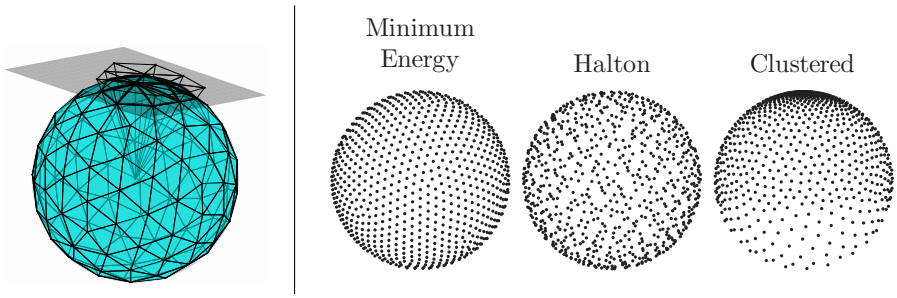

Figure 1: Left: An illustration of the change of variables (1), which projects points on $\mathbb{S}^2$ radially into a plane tangent to the sphere. Right: Various node distributions over $\mathbb{S}^2$. The distributions are shown for $N = 1,444$ in the case of the minimum energy and Halton node sets and $N = 1,458$ in the case of the clustered node set.

the spherical triangles τ_k, $k = 1, 2, \ldots, K$, with a local approximation.

2.2 Step 2: Project Locally to a Tangent Plane

The left frame of Figure 1 illustrates the gnomonic projection. This is a well known map projection technique, used already by the Greek Thales of Miletus in the seventh and sixth centuries B.C. [14]. It projects any geodesic (great-circle arc) as a straight line into a plane tangent to the sphere at $(\hat{x}_k, \hat{y}_k, \hat{z}_k)$.

For each triangle τ_k (with midpoint $(\hat{x}_k, \hat{y}_k, \hat{z}_k)$) the projection can be realized by a change of variables. Suppose that the sphere has radius ρ and that it has been rotated so that the midpoint $(\hat{x}_k, \hat{y}_k, \hat{z}_k)$ is at its top. Then the coordinates $(x_k^{'}, y_k^{'})$ in the tangent plane are given by

$$(x_k^{'}, y_k^{'}) = \begin{cases} \left(\frac{\rho x}{z}, \frac{\rho y}{z}\right) & \hat{x}_k^2 + \hat{y}_k^2 = 0 \\ \left(\frac{\rho\left[\hat{z}_k(\hat{x}_k x + \hat{y}_k y) - z(\hat{x}_k^2 + \hat{y}_k^2)\right]}{\sqrt{\hat{x}_k^2 + \hat{y}_k^2}(\hat{x}_k x + \hat{y}_k y + \hat{z}_k z)}, \frac{\rho^2(\hat{x}_k y - \hat{y}_k x)}{\sqrt{\hat{x}_k^2 + \hat{y}_k^2}(\hat{x}_k x + \hat{y}_k y + \hat{z}_k z)}\right) & \text{otherwise} \end{cases} \tag{1}$$

The changes of variables (1) transform the integral

$$\mathcal{I}_{S^2}(f) = \sum_{k=1}^{K} \iint_{\tau_k} f(x, y, z) dS = \sum_{k=1}^{K} \iint_{t_k} f(x_k^{'}, y_k^{'}) \frac{\rho^3}{(\rho^2 + (x_k^{'})^2 + (y_k^{'})^2)^{\frac{3}{2}}} dx_k^{'} dy_k^{'}$$

where t_k is the triangular region resulting from the projection of the edges (geodesics) of τ_k using (1). Note that a separate projection is performed for each of the spherical triangles τ_k.

2.3 Step 3: Local Weight Calculations

Considering a spherical triangle $\tau_k \in \mathcal{T}$, define $\mathcal{N}_k^n \subset \mathcal{S}_N$ to be the set of n points nearest its midpoint, $(\hat{x}_k, \hat{y}_k, \hat{z}_k)$ (in Euclidean distance). The midpoint is taken to be the average of the vertices of τ_k projected radially to the sphere surface; however, any point contained in the spherical triangle would be suitable. For each k, let $\left((x_k^{'})_j, (y_k^{'})_j\right)$ be the transformation of each (x_j, y_j, z_j), $j = 1, 2, \ldots, n$, in $\mathcal{N}_k^n$ via (1). The projection of each spherical triangle $\tau_k \in \mathcal{T}$ is a triangle t_k in the two dimensional space of $x_k^{'}$ and $y_k^{'}$. The proposed method approximates

$$\iint_{t_k} f(x_k^{'}, y_k^{'}) \frac{\rho^3}{(\rho^2 + (x_k^{'})^2 + (y_k^{'})^2)^{\frac{3}{2}}} dx_k^{'} dy_k^{'} \tag{2}$$

for each projected triangle t_k by interpolating the integrand over the points in $\mathcal{N}_k^n$ and then integrating the interpolant.

2.3.1 Creation of RBF-FD type weight set for a planar triangle

For the simplicity of discussion consider approximately evaluating the double integral of a function $g(x^{'}, y^{'})$ over a planar triangle t_{ABC} with vertices $A : (x_A^{'}, y_A^{'})$, $B : (x_B^{'}, y_B^{'})$, and $C : (x_C^{'}, y_C^{'})$. We denote this integral by

$$I_{t_{ABC}}(g) := \iint_{t_k} g(x_k^{'}, y_k^{'}) dx_k^{'} dy_k^{'} \tag{3}$$

and evaluate it approximately by integrating the RBF interpolant of $g(x^{'}, y^{'})$ with basis functions $\phi\left(\sqrt{(x^{'} - x_j^{'})^2 + (y^{'} - y_j^{'})^2}\right)$ centered at the points $(x_i^{'}, y_i^{'})$, $i = 1, 2, \ldots, n$, in a neighborhood of t_{ABC}. RBF interpolation has been used successfully in the approximation of differential operators over subsets of scattered data through the concept of RBF-FD [9, 10, 11, 12].

Following common RBF/RBF-FD procedures, we let $\{\pi_l(x^{'}, y^{'})\}_{l=1}^{M}$, with $M = \frac{(m+1)(m+2)}{2}$ be the set of all of the bivariate polynomial terms up to degree m. We construct the inter-

polant

$$s(x^{'}, y^{'}) := \sum_{j=1}^{n} c_j^{\mathrm{RBF}} \phi\left(\sqrt{(x' - x_j')^2 + (y' - y_j')^2}\right) + \sum_{l=1}^{M} c_l^p \pi_l(x^{'}, y^{'})$$

where $c_1^{RBF}, \ldots, c_n^{RBF}, c_1^p, \ldots, c_M^p \in \mathbb{R}$ are chosen to satisfy the interpolation conditions $s(x_j^{'}, y_j^{'}) = g(x_j^{'}, y_j^{'})$, $j = 1, 2, \ldots, n$, along with constraints $\sum_{i=1}^{n} c_i^{RBF} \pi_l(x_i^{'}, y_i^{'}) = 0$, for $l = 1, 2, \ldots, M$. By integrating the interpolant we wish to reduce the approximation of the integral of g to $I_{t_{ABC}}(g) \approx \sum_{i=1}^{N} w_i^{RBF} g(x_i^{'}, y_i^{'})$. A simple derivation can be carried out to show that the weights can be found by solving the linear system $\tilde{A}W = \tilde{I}$ with

$$\tilde{A} = \begin{bmatrix} A^T & P \\ P^T & 0 \end{bmatrix}, \tilde{I} = \begin{bmatrix} I^{RBF} \\ I^p \end{bmatrix}, \text{ and } W = \begin{bmatrix} w_1^{RBF} & \cdots & w_n^{RBF} & w_1^p & \cdots & w_M^p \end{bmatrix}^T,$$

where $A_{ij} = \phi\left(\sqrt{(x_i' - x_j')^2 + (y_i' - y_j')^2}\right)$, $I_j^{RBF} = I_{t_{ABC}}\left(\phi\left(\sqrt{(x' - x_j')^2 + (y' - y_j')^2}\right)\right)$, $P_{il} = \pi_l(x_i^{'}, y_i^{'})$, and $I_l^p = I_{t_{ABC}}(\pi_l(x^{'}, y^{'}))$, for $i, j = 1, 2, \ldots, n$ and $l = 1, 2, \ldots, M$ [11, Section 5.1.4]. The system of linear equations is uniquely solvable in our present context [15, Theorem 8.21].

The integrals $I_l^p = I_{t_{ABC}}(\pi_l(x^{'}, y^{'}))$, $l = 1, 2, \ldots, M$, can be evaluated exactly via, for instance, Green's theorem or through the conversion of the integral to barycentric coordinates. Exact evalutations of $I_j^{RBF} = I_{t_{ABC}}\left(\phi\left(\sqrt{(x' - x_j')^2 + (y' - y_j')^2}\right)\right)$, $j = 1, 2, \ldots, n$, are described next. In the context of global RBF approximations based on thin-plate splines (TPS) this was studied in [16] for planar polygonal domains and in [17] for a sphere (with implementations requiring $O(N^3)$ operations).

2.3.2 Integrals of RBFs Over Arbitrary Planar Triangles

Based on numerical accuracy/stability comparisons, we use for this sub-task a different approach than the one described in [16]. Suppose that t_{ABC} is any planar triangle and that O is an arbitrary point. We wish to compute $I_{ABC}(\phi)$, where ϕ is an RBF centered at O. For instance, in the case shown in Figure 2(a), it is clear that $I_{ABC}(\phi)$ could be computed by adding the integrals of ϕ over each of the six right triangles shown. That is, we can

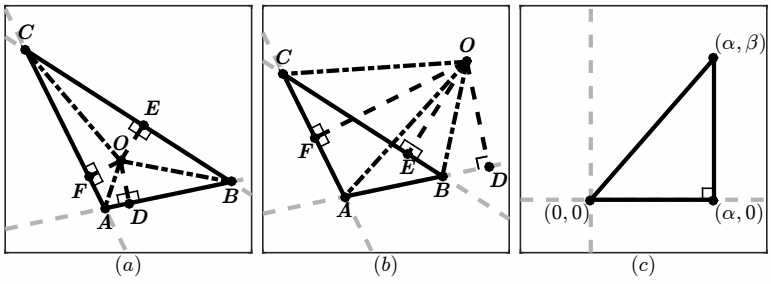

Figure 2: (a) The integral $I_{ABC}(\phi)$ can be computed by adding $I_{OAD}(\phi)$, $I_{ODB}(\phi)$, $I_{OBE}(\phi)$, $I_{OEC}(\phi)$, $I_{OCF}(\phi)$ and $I_{OFA}(\phi)$. (b) The integral $I_{ABC}(\phi)$ can be computed by adding the integrals $I_{OAD}(\phi)$, $I_{OFA}(\phi)$ and $I_{OCF}(\phi)$ and by subtracting $I_{ODB}(\phi)$, $I_{OBE}(\phi)$, and $I_{OEC}(\phi)$. (c) The integrals over each of the right triangles in frames (a) and (b) can be computed by integrating over a triangle like the one in this frame where α is the distance from O to the perpendicular vertex D, E or F and β is the distance from the perpendicular vertex to the appropriate triangle vertex A, B or C.

evaluate

$$\begin{aligned} I_{ABC}(\phi) =& s_{ABC}\left(s_{OAD}I_{OAD}(\phi) + s_{ODB}I_{ODB}(\phi) + s_{OBE}I_{OBE}(\phi) + \right. \\ & \left. s_{OEC}I_{OEC}(\phi) + s_{OCF}I_{OCF}(\phi) + s_{OFA}I_{OFA}(\phi)\right). \end{aligned} \tag{4}$$

with $s_{ABC} = s_{OAD} = s_{ODB} = s_{OBE} = s_{OEC} = s_{OCF} = s_{OFA} = +1$, where D, E, and F are the orthogonal projections of O onto the lines connecting the vertices of t_{ABC}. Similarly, in frame (b) we can choose $s_{ABC} = s_{OAD} = s_{OCF} = s_{OFA} = +1$ and $s_{ODB} = s_{OBE} = s_{OEC} = -1$. In either case the choice of signs corresponds to the orientations of the vertices of each of the triangles. A similar result can be shown if the vertices of t_{ABC} are oriented clockwise and $s_{ABC} = -1$ correspondingly. The sign s_{ABC}, for example, can be defined concisely by (minding the order of ABC)

$$s_{ABC} := \text{sign}\left(\begin{bmatrix} y_A^{'} - y_B^{'} \\ x_B^{'} - x_A^{'} \end{bmatrix} \cdot \begin{bmatrix} x_C^{'} - x_A^{'} \\ y_C^{'} - y_A^{'} \end{bmatrix}\right).$$

2.3.3 Integrals of RBF's Over Planar Right Triangles

Evaluation of (4) requires the double integrals over each of the right triangles t_{OAD}, t_{ODB}, t_{OBE}, t_{OEC}, t_{OCF} and t_{OFA}. Consider the right triangle t_{OAD}, since the rest are completely

analogous. After a translation and a rotation, the integral takes the form

$$I_{t_{OAD}}(\phi) = \int_0^{\alpha} \int_0^{\frac{\beta}{\alpha}x'} \phi\left(\sqrt{(x')^2 + (y')^2}\right) dy' dx',$$

with α the length of the side OD and β the length of the side AD (an example of this result is shown in Figure 2(c)).

Many commonly used RBFs can be integrated exactly over right triangles in the plane. For instance, when using $\phi(r) = r^3, r^5, r^7$:

$$\begin{aligned}
&\int_0^{\alpha} \int_0^{\frac{\beta}{\alpha}x'} \left((x')^2 + (y')^2\right)^{\frac{3}{2}} dy' dx' = \\
&\qquad \frac{1}{40}\alpha\left(3\alpha^4 \sinh^{-1}\left(\frac{\beta}{\alpha}\right) + \beta\sqrt{\alpha^2+\beta^2}\left(5\alpha^2 + 2\beta^2\right)\right), \\
&\int_0^{\alpha} \int_0^{\frac{\beta}{\alpha}x'} \left((x')^2 + (y')^2\right)^{\frac{5}{2}} dy' dx' = \\
&\qquad \frac{1}{336}\alpha\left(15\alpha^6 \sinh^{-1}\left(\frac{\beta}{\alpha}\right) + \beta\sqrt{\alpha^2+\beta^2}\left(33\alpha^4 + 26\alpha^2\beta^2 + 8\beta^4\right)\right), \text{ and} \\
&\int_0^{\alpha} \int_0^{\frac{\beta}{\alpha}x'} \left((x')^2 + (y')^2\right)^{\frac{7}{2}} dy' dx' = \\
&\qquad \frac{\alpha\left(105\alpha^8 \sinh^{-1}\left(\frac{\beta}{\alpha}\right) + \beta\sqrt{\alpha^2+\beta^2}\left(279\alpha^6 + 326\alpha^4\beta^2 + 200\alpha^2\beta^4 + 48\beta^6\right)\right)}{3456}.
\end{aligned}$$

We choose these types of RBFs (and in particular $\phi(r) = r^7$) here since these are free from a 'shape parameter' ε and also provide both good accuracy and well conditioned linear systems used to compute the quadrature weights [18].

2.4 Step 4: Combine the Weights Over the Entire Sphere

Applying step 3 to the double integral (2) over each of the triangles t_k gives

$$\iint_{t_k} f(x_k', y_k') \frac{\rho^3}{(\rho^2 + (x_k')^2 + (y_k')^2)^{\frac{3}{2}}} dx_k' dy_k' \approx \sum_{j=1}^{n} (w_k)_j f((x_k')_j, (y_k')_j) \frac{\rho^3}{(\rho^2 + (x_k')_j^2 + (y_k')_j^2)^{\frac{3}{2}}}.$$

Hence,

$$\mathcal{I}_{\mathbb{S}^2}(f) \approx \sum_{k=1}^{K} \sum_{j=1}^{n} (w_k)_j f((x_k')_j, (y_k')_j) \frac{\rho^3}{(\rho^2 + (x_k')_j^2 + (y_k')_j^2)^{\frac{3}{2}}}, \tag{5}$$

i.e. weights in the plane and on the sphere differ only by the last factor in (5). Let $\mathcal{K}_i$, $i = 1, 2, \ldots, N$ be the set of all pairs (k, j) such that $((x_k^{'})_j, (y_k^{'})_j) \mapsto (x_i, y_i, z_i)$. Then the surface integral over $\mathbb{S}^2$ can be written as

$$\mathcal{I}_{\mathbb{S}^2}(f) \approx \sum_{i=1}^{N} \left(\sum_{(k,j)\in\mathcal{K}_i} (w_k)_j \frac{\rho^3}{(\rho^2 + (x_k^{'})_j^2 + (y_k^{'})_j^2)^{\frac{3}{2}}} \right) f(x_i, y_i, z_i) = \sum_{i=1}^{N} W_i f(x_i, y_i, z_i). \quad (6)$$

3 Test cases

Numerical tests of the proposed quadrature method are performed on a variety of integrands that appear elsewhere in the literature. In particular, test integrands from [7] and [8] are considered. In all cases, the method utilizes its current default settings for the number of nearest neighbors ($n = 80$) and maximum order of the bivariate polynomial terms used ($m = 7$ for a total of 36 bivariate terms). Figure 3 shows computation time and the error in the case of a very smooth test function. For more realistic ones, the accuracy increase becomes negligible past our default choices for n and m. Such accuracy tests will also show that the RBF component is essential in the local interpolant, both for providing good accuracy and to guarantee non-singularity of all the linear systems used to compute the quadrature weights. The default setting uses the RBF $\phi(r) = r^7$, since the cost of computing weights is similar to that of r^3, but features improved accuracy.

Comparisons were performed over a variety of node sets featuring varying degrees of regularity. The right frame of Figure 1 illustrates these node sets. There are two types of sets of nodes denoted ME in this paper. The node distribution of the first type, ME (F), are the quasi-minimum energy nodes discussed in [8], which are obtained by treating the nodes as point charges and nearly minimizing the Riesz energy over the possible configurations of the nodes [19, 20]. These are available for a power of 3 in the Riesz energy only for $N = 2,500,\ 10,000,\ 22,500,\ 40,000,\ 62,500,\ 90,000,\ 160,000,$ and $250,000$ as in [8]. The node distributions of the second type called ME (W) are those made available in [4]. These nodes locally minimize potential energy on the sphere and are available for $N = (M+1)^2$, $M = 1, 2, \ldots, 100$ [4] (the present work considers $M = 20, 21, \ldots, 100$). The second plot illustrates the Halton node set with $N = 1,444$, which exhibits far less uniformity. These points are generated by mapping the first N points from the 2 dimensional pseudorandom

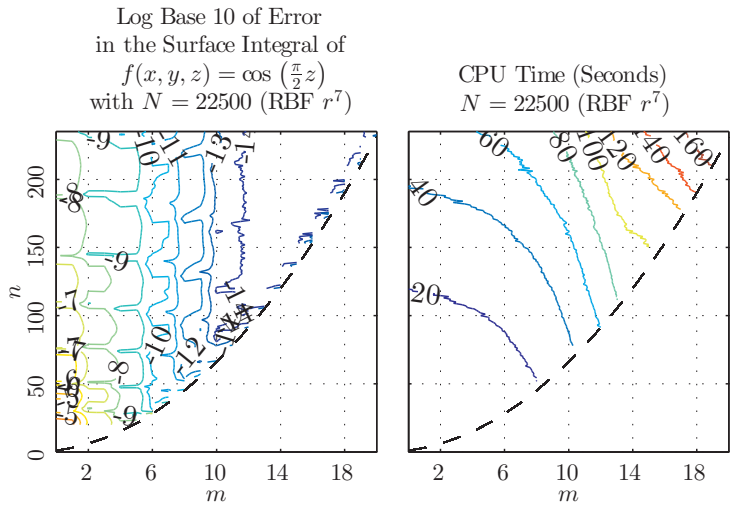

Figure 3: Left: Contour plot of log base 10 of the error when computing the surface integral of $f(x,y,z) = \cos\left(\frac{\pi}{2}z\right)$ ($\mathcal{I}_{\mathbb{S}_2}(f) = 8$) for various m and n. Right: Contour plots of computation time in seconds when computing the quadrature weights for various m and n. The dashed parabola in both plots represent the boundary $n = M$ below which the linear system $\tilde{A}W = \tilde{I}$ becomes singular. The vertical axes are the same in the two plots.

Halton sequence [21] of points contained in $[0,1] \times [0,1]$ onto $\mathbb{S}^2$. The last node set (with $N = 1,458$) is a clustered node set generated as described in [22] with a node separation function $r(x,y,z) = \sigma(1.2-z)$, ($\sigma = 0.0555$ in the figure). Changing the parameter σ varies the density of the nodes. In this study σ ranges from 0.005 to 0.1 in increments of 0.001. The ME (both types) node sets considered in this paper have corresponding quadrature weights available at [5] (discussed in [8]) and [4]. These weights are used for comparison with those generated by the proposed method.

Comparisons were also made against two alternative methods for finding quadrature weights on the sphere. The first of these types of methods is SPH based quadrature, where the quadrature weights are obtained directly by SPH based interpolation [4]. Such methods can achieve spectral accuracy at the cost of $O(N^3)$ operations and $O(N^2)$ memory. The interpolation matrix in these methods can become extremely ill conditioned for certain sets of scattered data. To remedy this situation, node sets must be chosen carefully to minimize ill conditioning (as in [23], with the introduction of MD node sets) or alternative methods for solving the linear system in the interpolation problem must be employed (as in [7], where the inversion of the interpolation matrix is replaced by the Moore-Penrose pseudoinverse).

The second method we compare against is a kernel based quadrature method [8]. This method creates local Lagrangian basis functions for an RBF interpolant, then utilizes these functions as a preconditioner [24] to GMRES [25] when solving the linear system required to find the quadrature nodes. As it is implemented, this achieves orders of accuracy around $O(h^4)$ ($O(N^{-2})$ when the node spacing $h \sim O(N^{-\frac{1}{2}})$, which is typical for quasiuniform node sets). The cost of the method (as implemented in [8]) is $O(N^2)$ operations and $O(N^2)$ memory.

The computational costs of the proposed method and the two alternative methods (in terms of time) are illustrated in Figure 4. The main components of the computational cost of the proposed method are

(i) for each of the $2N-4$ triangles, $O(n^3)$ operations to obtain the quadrature weights,

(ii) Delaunay triangulation on the sphere, $O(N \log N)$ operations,

(iii) nearest neighbor search using the kd-tree algorithm, $O(N\ n \log N)$ operations [26].

With n fixed (typically set to $n = 80$), the overall cost will for N extremely large scale as $O(N \log N)$ due to the items (ii) and (iii). However, before this occurs (including all of the present test cases), item (i) will dominate, leading to purely the purely linear trend with N seen in the top-left subplot of Figure 4 and also to the near-perfect linear scaling with the number of cores seen in the top-right subplot.

3.1 Test Integrands and Results

Comparisons are made on three test integrands:

$$\begin{aligned} f_1(x,y,z) =& \frac{1+\tanh(-9x-9y+9z)}{9} \\ f_2(x,y,z) =& -\left(2-2(xx_c+yy_c+zz_c)\right)^{\frac{1}{4}} F(\theta,\phi) \\ f_3(x,y,z) =& \frac{\frac{\pi}{2}+\operatorname{atan}\left(300\left(z-\frac{9999}{10000}\right)\right)}{\pi}, \end{aligned}$$

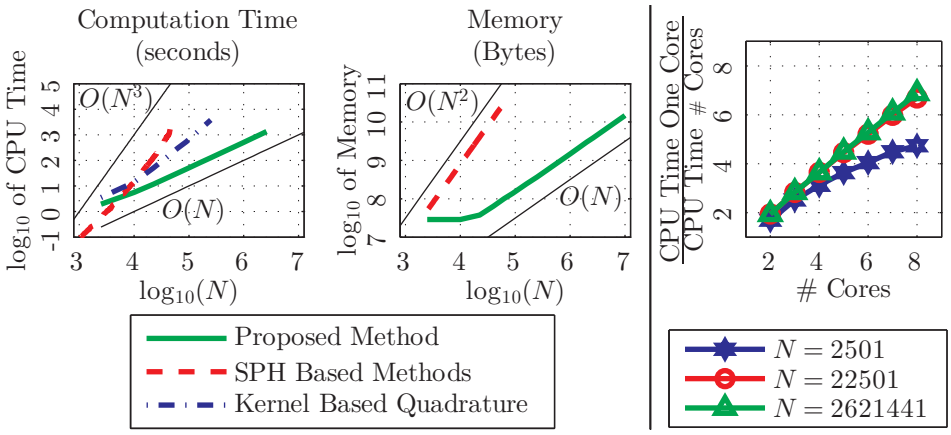

Figure 4: Left: $\log_{10}$ of the computation time in seconds versus $\log_{10}$ of the number of nodes N. Timing results are shown when using 8 cores. Center: $\log_{10}$ of the memory in Bytes versus $\log_{10}$ of the number of nodes N. Memory required for the kernel based quadrature scales like that of SPH based methods and is not shown. Right: Parallel computation scaling results showing the ratio of computation time on a single core to the computation time on # cores versus the number (#) of cores. All computations shown in this figure were performed in Matlab on machines with dual Intel Xeon E5-2687W 3.1 GHz, 8-core processors.

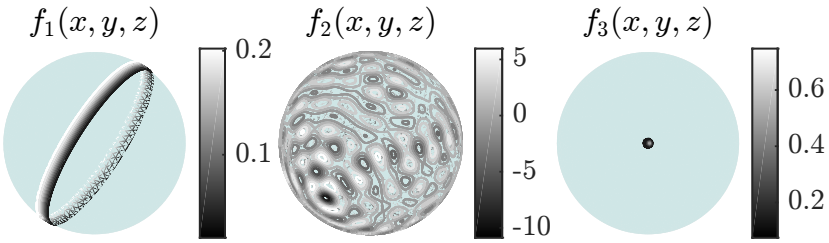

Figure 5: Ten equally spaced contours over $\mathbb{S}^2$ for each of the test functions $f_1(x, y, z)$, $f_2(x, y, z)$ and $f_3(x, y, z)$

where $(x_c, y_c, z_c) = (\cos(-2.0281)\cos(0.76102), \sin(-2.0281)\cos(0.76102), \sin(0.76102))$ and $\theta = \cos^{-1}(z)$ and $\phi = \tan^{-1}\left(\frac{y}{x}\right)$ so that $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$. Further, define

$$F(\theta,\phi) = \sum_{k=-20}^{-1} \sqrt{\frac{41}{2\pi}}\sqrt{\frac{(20+k)!}{(20-k)!}} P_{20}^{-k}(\cos(\theta))\sin(k\phi)\text{sign}(P_{20}^{-k}(\cos(\theta_c))\sin(k\phi_c)) + \sum_{k=0}^{20} \sqrt{\frac{41}{2\pi}}\sqrt{\frac{(20-k)!}{(20+k)!}} P_{20}^{k}(\cos(\theta))\cos(k\phi)\text{sign}(P_{20}^{k}(\cos(\theta_c))\cos(k\phi_c))$$

(a linear combination of the real parts of a set of spherical harmonics). Illustrations of these test integrands are given in Figure 5. The test integrands here feature varying degrees of

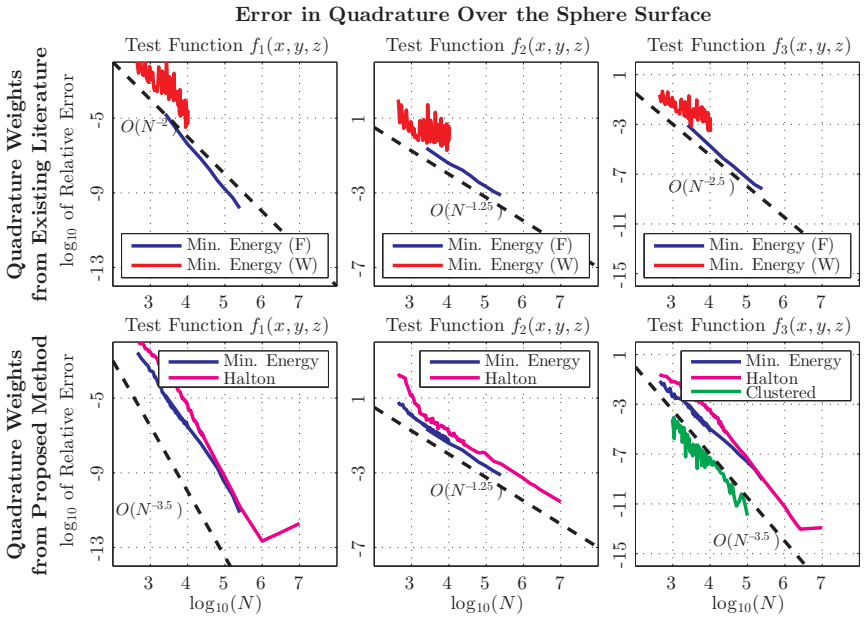

Figure 6: Relative errors when computing the surface integrals of $f_1(x,y,z)$, $f_2(x,y,z)$ and $f_3(x,y,z)$ over $\mathbb{S}^2$. Except for the clustered node set, these are the maximum relative errors over 1,000 random rotations of each test integrand. The relative errors displayed in the top row are for the cases where the quadrature weights W_i in 6 are those available at [4] (generated by an SPH-based method) in the case of ME (W) and those available at [5] (generated by the kernel-based quadrature method) in the case of ME (F). The bottom row then displays the relative errors when the quadrature weights are generated by the proposed method.

smoothness. For instance, the function $f_1(x,y,z)$ has slowly converging spherical harmonic coefficients as indicated in [7], but it is still in $C^\infty(\mathbb{S}^2)$. It has a sharp gradient along a narrow band around the sphere. The second test function is highly oscillatory over whole sphere, with a singularity at the tip of a sharp spike (lowering the convergence rate for all methods). The third test integrand is also $C^\infty(\mathbb{S}^2)$, but with an extreme spike at one location. When integrating over $\mathbb{S}^2$ the values of the surface integrals are $\mathcal{I}_{\mathbb{S}^2}(f_1) = \frac{4\pi}{9}$, $\mathcal{I}_{\mathbb{S}^2}(f_2) \approx 0.014830900415995262852$ [8], and $\mathcal{I}_{\mathbb{S}^2}(f_3) \approx 0.049629692928687$.

Figure 6 displays the relative errors when computing the surface integrals of $f_1(x,y,z)$, $f_2(x,y,z)$ and $f_3(x,y,z)$ over $\mathbb{S}^2$, respectively.

3.2 Discussion

The frames of Figure 6 indicate that the proposed method (at its default settings) can achieve a convergence rate around $O(N^{-3.5})$ even for test functions featuring rapidly changing

features like those in $f_1(x, y, z)$ and $f_3(x, y, z)$. The computational efficiency of the method allows for the computation of quadrature weights for nodes sets tailored to the integrand (rather than being separately tabulated). The bottom right frame of Figure 6 illustrates an improvement of nearly 3 orders of magnitude by employing a simple node clustering strategy around the spike located at the 'north pole'. When the test integrand is no longer smooth (as for $f_2(x, y, z)$) all of the methods discussed here perform comparably well.

To estimate the accuracy of the proposed method in an application with a given function and a fixed node set, we recommend that the obtained values for the integral are compared when the parameters n, m, and RBF type are varied somewhat from their default settings of $n = 80$, $m = 7$ (maximum order of bivariate polynomial terms), and $\phi(r) = r^7$. The code that is available at Matlab Central [27] is designed to trivially allow for such changes.

4 Conclusions

The proposed algorithm for calculating quadrature weights features operation and memory counts for N scattered nodes of only $O(N \log N)$ and $O(N)$, respectively. The present implementation uses $\phi(r) = r^7$ and consequently exhibits $O(N^{-3.5})$ convergence for smooth test functions (when the node spacing is $O(N^{-0.5})$). Increasing the power "7" in the RBF can lead to higher still formal orders of accuracy at the expense of numerical stability when solving the linear system for the quadrature weights.

References

[1] C. AHRENS and G. BEYLKIN. Rotationally invariant quadratures for the sphere. *Proc. Roy. Soc. A*, 465:3103–3125, 2009.

[2] Z. P. BAŽANT and B. H. OH. Efficient numerical integration on the surface of a sphere. *Z. angew. Math. u. Mech.*, 66:37–49, 1986.

[3] A. H. STROUD. *Approximate Calculation of Multiple Integrals.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1971.

[4] R. S. WOMERSLEY and I. H. SLOAN. Interpolation and cubature on the sphere, http://web.maths.unsw.edu.au/˜rsw/sphere/. Website. Accessed 30 October 2014.

[5] G.B. WRIGHT. http://math.boisestate.edu/~wright/quad_weights/. Website. Accessed 30 October 2014.

[6] K. HESSE, I. H. SLOAN, and R. S. WOMERSLEY. Numerical integration on the sphere. In W. Freeden, M.Z. Nashed, and T. Sonar, editors, *Handbook of Geomathematics*, pages 1187–1219, Berlin, 2010. Springer Verlag.

[7] B. FORNBERG and J. M. MARTEL. On spherical harmonics based numerical quadrature over the surface of a sphere. *Adv. Comput. Math.*, 40:1169–1184, 2014.

[8] E. FUSELIER, T. HANGELBROEK, F. J. NARCOWICH, J. D. WARD, and G. B. WRIGHT. Kernel based quadrature on spheres and other homogeneous spaces. *Numer. Math.*, 127:57–92, 2014.

[9] N. FLYER, E. LEHTO, S. BLAISE, G. B. WRIGHT, and A. ST-CYR. A guide to RBF-generated finite-differences for nonlinear transport: Shallow water simulations on a sphere. *J. Comput. Math.*, 231(11):4078–4095, 2012.

[10] N. FLYER, G.B. WRIGHT, and B. FORNBERG. Radial basis function-generated finite differences: A mesh-free method for computational geosciences. In W. Freeden, M. Z. Nashed, and T. Sonar, editors, *Handbook of Geomathematics*. Springer Verlag Berlin Heidelberg. doi: 10.1007/978-3-642-27793-1 61-1.

[11] B. FORNBERG and N. FLYER. *A Primer on Radial Basis Functions with Applications to the Geosciences*. SIAM, 2015.

[12] B. FORNBERG and N. FLYER. Solving PDEs with radial basis functions. *Acta Numerica*, 24:215–258, 2015.

[13] R. J. RENKA. Algorithm 772: STRIPACK: Delaunay triangulation and Voronoi diagram on the surface of a sphere. *ACM Trans. Math. Softw.*, 23(3):416–434, Sept. 1997.

[14] J. P. SNYDER. *Map Projections: A Working Manual*. Geological Survey (U.S.), Washington, DC, U.S., 1987.

[15] H. WENDLAND. *Scattered Data Approximation*. Cambridge University Press, Cambridge, U.K., 2005.

[16] A. SOMMARIVA and M. VIANELLO. Meshless cubature by Green's formula. *Appl. Math. Comput.*, 183(2):1098–1107, 2006.

[17] I.H. SLOAN and A. SOMMARIVA. Approximation on the sphere using radial basis functions plus polynomials. *Adv. Comput. Math.*, 29:147–177, 2008.

[18] N. FLYER, B. FORNBERG, G.A. BARNETT, and V. BAYONA. On the role of polynomials in RBF-FD approximations: I. interpolation and accuracy. submitted.

[19] D.P. HARDIN and E.B. SAFF. Discretizing manifolds via minimum energy points. *Notices Amer. Math. Soc.*, 51:1186–1194, 2004.

[20] S.V. BORODACHOV, D.P. HARDIN, and E.B. SAFF. Low complexity methods for discretizing manifolds via Riesz energy minimization. *Found. Comp. Math.*, 14:1173–1208, 2014.

[21] J. H. HALTON. On the efficiency of certain quasi-random sequences of points in evaluation multi-dimensional integrals. *Numer. Math.*, 2(1):84–90, 1960.

[22] B. FORNBERG and N. FLYER. Fast generation of 2-D node distributions for mesh-free PDE discretizations. *Comp. Math. Applic.*, 69:531–544, 2015.

[23] I.H. SLOAN and R.S. WOMERSLEY. Extremal systems of points and numerical integration on the sphere. *Adv. Comput. Math.*, 21:102–125, 2004.

[24] E.J. FUSELIER, T. HANGELBROEK, F.J. NARCOWICH, J.D. WARD, and G.B. WRIGHT. Localized bases for kernel spaces on the unit sphere. *SIAM J. Numer. Anal.*, 51(5):2538–2562, 2013.

[25] Y. SAAD and M.H. SCHULTZ. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 7:856–869, 1986.

[26] J. H. FRIEDMAN, J. L. BENTLEY, and R. A. FINKEL. An algorithm for finding best matches in logarithmic expected time. *ACM Trans Math Softw*, 3(3), 1977.

[27] J. A. REEGER. Spherical_Quadrature_RBF(Quadrature_Nodes) (2015). (http://www.mathworks.com/matlabcentral/fileexchange/51214), MATLAB Central File Exchange. Accessed 20 June 2015.