Journal of Scientific Computing manuscript No.
(will be inserted by the editor)

Approximate Integrals Over the Volume of the Ball

Jonah A. Reeger

Received: date / Accepted: date

Abstract A Radial Basis Function Generated Finite-Differences (RBF-FD) inspired technique for evaluating definite integrals over the volume of the ball in three dimensions is described. Such methods are necessary in many areas of Applied Mathematics, Mathematical Physics and myriad other application areas. Previous approaches needed restrictive uniformity in the node set, which the algorithm presented here does not require. By using RBF-FD approach, the proposed algorithm computes quadrature weights for N arbitrarily scattered nodes in only $O(N \log N)$ operations with high orders of accuracy.

Keywords Radial Basis Function · RBF · quadrature · volume · ball

Mathematics Subject Classification (2010) 68Q25 · 65R99

1 Introduction

This article is concerned with the development of a method for the approximate evaluation of definite integrals over the volume of the ball in $\mathbb{R}^3$ of radius $\rho > 0$. That is, consider volume integration over the domain $\Omega = \left\{\mathbf{x} \in \mathbb{R}^3 | \, \|\mathbf{x} - \mathbf{x}_0\|_2 \leq \rho\right\}$, with $\mathbf{x}_0 \in \mathbb{R}^3$. Applications of this common problem in mathematics abound from the more specific scenarios of estimating the volumes of wells containing hydrocarbons beneath the Earth's surface [1] and recovering important diagnostic information in, e.g., Thermoacoustic or Photoacoustic Tomogrpahy [2] to some more generic problems in potential theory, magnetism and other concepts in mathematical physics [3].

The approximation of the values of definite integrals (quadrature when considering integration over an interval, or quadrature/cubature when considering

This work was funded by the Office of Naval Research program Atmospheric Propagation Sciences for High Energy Lasers and the Air Force Office of Scientific Research project Radial Basis Functions for Numerical Simulation.

J. Reeger
Bellbrook, Ohio, United States
Tel.: +1-724-689-3830
E-mail: jonah.reeger@gmail.com

integration over domains in two or more dimensions) is a rich topic dating back many centuries to early attempts to measure, for instance, the area of the circle [4]. Since that time much research has been devoted to developing sophisticated and accurate techniques for estimating the values of integrals over intervals, areas, and volumes. There are many texts devoted to summarizing such methods, see, for example [5–7, 4].

In the simplest case, the rules for quadrature over intervals are often constructed by replacing the integrand with a polynomial interpolant or a polynomial approximation and then integrating, or by enforcing that a rule be exact on a particular class of functions (like polynomials up to a particular degree). This is how to arrive at, for instance, Newton-Cotes or Gaussian Quadrature rules, respectively. These rules for integration over intervals are then often leveraged in the context of iterated integrals by employing one-dimensional quadrature rules for each variable in turn, leading to so-called Cartesian product formulas. Such formulas require structure in the node sets–spacing between nodes that is uniform or tied to the roots of orthogonal polynomials–across each variable of integration. This requirement may be impractical or require an additional interpolation between the structured node set and locations where the integrand is specified.

To overcome these requirements on the structure of the node set, the value of the integral can be approximated utilizing concepts from number theory or through pseudorandom sampling of the integrand as in Monte-Carlo techniques [6]. More common, however, is the alternative method of constructing quadrature rules over areas and volumes by replacing the integrand with a now multivariate polynomial interpolant or approximation and then integrating. It is often the case that the basis set used for interpolation does not depend on the locations of the nodes (e.g. multivariate polynomials). Such basis sets suffer from a question of the existence and uniqueness of an interpolant on unstructured node sets [8, 9], which has prompted the use of Radial Basis Functions (RBFs) in the approximation of the integrand.

The proposed quadrature technique was developed to compute weights at whatever node locations are specified by the user. This is because in applications, numerical quadrature is usually a follow-up to some other task (such as collecting data, or numerically solving PDEs), making it impractical to require node locations that are specific to the quadrature method. The present algorithm is therefore designed to find the quadrature weights given a node set defined by the application. Further, the proposed algorithm allows for node sets featuring spatially varying separation when increased node density is needed to capture fine structure in the integrand. If the reader is only interested in evaluating the definite integral over the volume of the ball and is not constrained to specific node locations (e.g. data samples can be chosen at the precise locations required by a quadrature method or the integrand is available in some, perhaps closed, form to be evaluated), then there are certainly other methods available in most scientific computing environments for this purpose that can achieve high, even spectral, orders of accuracy. The proposed algorithm, on the other hand, will find its greatest utility when the integrand or data is difficult or impossible to sample at the node locations required for a particular quadrature rule, and the method can still achieve high orders of accuracy.

The numerical method described in this paper is a generalization of RBF-FD (radial basis function-generated finite differences). This approach has so far mostly

been used to approximate partial derivatives, with the key difference to regular finite differences being that the node points no longer need to be grid-based (in particular, Cartesian node layouts are now known to be less-than-optimal [10]). For surveys of RBFs and of RBF-FD methods as they are applied to PDEs, see [11,12]. Further, RBFs have been used successfully to construct quadrature rules for an interval in one-dimension, bounded domains in two-dimensions and, more specifically, integrals over bounded two-dimensional (piecewise-)smooth surfaces embedded in three-dimensions [13–15].

The following Section 2 describes the present quadrature method. Section 3 describes some test examples, with illustrations of convergence rates and computational costs. Finally, section 4 outlines some conclusions. A Matlab implementation of the method is available at Matlab Central's File Exchange [16].

2 Description of the key steps in the algorithm

Consider evaluating

$$\iiint_{\Omega} f(\mathbf{x}) dV, \tag{1}$$

where $\Omega = \left\{\mathbf{x} \in \mathbb{R}^3 : \|\mathbf{x} - \mathbf{x}_0\|_2 \leq \rho\right\}$.

Similar to the work presented in [13–15] for surface integrals, the proposed algorithm can be described in four steps:

1. Decompose the domain of integration into $K \in \mathbb{Z}^+$ subdomains.
2. On the k^{th} subdomain construct an interpolant of the integrand.
3. Integrate the interpolant of the integrand to determine weights for integrating a function f over the k^{th} subdomain.
4. Combine the weights for the integrals over the K subdomains to obtain a weight set for approximating the volume integral of f over Ω.

Each of these steps is described in greater detail in what follows.

2.1 Step 1: Decompose the domain of integration

Consider first a set $\mathcal{S}_N = \{\mathbf{x}_i\}_{i=1}^N$ of N points in Ω, with a subset exactly on the boundary surface. On this set of points construct a tessellation $T = \{t_k\}_{k=1}^K$ (via Delaunay tessellation or some other algorithm) of K tetrahedra. These tetrahedra encompass the bulk of the volume of Ω but not the entire volume.

Let $\mathcal{K}_S \subset \{1, 2, \ldots, K\}$ be the set of indices such that if $k \in \mathcal{K}_S$ then the tetrahedron t_k has a face that is not shared with any of the other tetrahedra. This face has all three vertices on the surface of Ω and, unless the surface is planar, there is a sliver of volume, s_k, between the unshared face and the spherical surface bounding the ball that must be accounted for in decomposing the volume. Conversely, let $\mathcal{K}_I = \{1, 2, \ldots, K\} \setminus \mathcal{K}_S$ be the indices of tetrahedra that do not

have a face with three vertices on the surface. With these definitions (1) can be decomposed as

$$\iiint_{\Omega} f(\mathbf{x})dV = \sum_{k\in\mathcal{K}_I} \iiint_{t_k} f(\mathbf{x})dV + \sum_{k\in\mathcal{K}_S} \left(\iiint_{t_k} f(\mathbf{x})dV + \iiint_{s_k} f(\mathbf{x})dV \right). \quad (2)$$

2.2 Step 2: Construct an interpolant of the integrand

For each tetrahedron in T define the sets $\mathcal{N}_k = \left\{\mathbf{x}_{k,j}\right\}_{j=1}^{n}$ to be the n points in $\mathcal{S}_N$ nearest to the midpoint of t_k (the average of the vertices of t_k). Then for an individual t_k the integral

$$\iiint_{t_k} f(\mathbf{x})dV \quad (3)$$

is evaluated by first approximating $f(\mathbf{x})$ by an RBF interpolant, with interpolation points from the set $\mathcal{N}_k$, and then integrating the interpolant. Often the RBF interpolant is a linear combination of (conditionally-) positive definite RBFs,

$$\phi\left(\left\|\mathbf{x}-\mathbf{x}_{k,j}\right\|_2\right), j = 1, 2, \ldots, n$$

and augmented by multivariate polynomial terms. If $k \in \mathcal{K}_S$, then the same interpolant and set of nodes is used for approximating the integrand over s_k. Define $\{\pi_l(\mathbf{x})\}_{l=1}^{M}$, with $M = \frac{(m+1)(m+2)(m+3)}{6}$, to be the set of all of the trivariate polynomial terms up to degree m. The interpolant is constructed as

$$q(\mathbf{x}) := \sum_{j=1}^{n} c_{k,j}^{\text{RBF}} \phi\left(\left\|\mathbf{x}-\mathbf{x}_{k,j}\right\|_2\right) + \sum_{l=1}^{M} c_{k,l}^{p} \pi_l(\mathbf{x}),$$

where $c_{k,1}^{RBF}, \ldots, c_{k,n}^{RBF}, c_{k,1}^{p}, \ldots, c_{k,M}^{p} \in \mathbb{R}$ are chosen to satisfy the interpolation conditions $q(\mathbf{x}_{k,j}) = f(\mathbf{x}_{k,j})$, $j = 1, 2, \ldots, n$, along with constraints $\sum_{j=1}^{n} c_{k,j}^{RBF} \pi_l(\mathbf{x}_{k,j}) = 0$, for $l = 1, 2, \ldots, M$.

2.3 Step 3: Integrate the interpolant of the integrand

By integrating the interpolant, the approximation of the integral of f is reduced to

$$\iiint_{t_k} f(\mathbf{x})dV \approx \sum_{j=1}^{n} w_{k,j} f(\mathbf{x}_{k,j})$$

for $k \in \mathcal{K}_I$ and

$$\iiint_{t_k} f(\mathbf{x})dV + \iiint_{s_k} f(\mathbf{x})dV \approx \sum_{j=1}^{n} w_{k,j} f(\mathbf{x}_{k,j})$$

for $k \in \mathcal{K}_S$. A simple derivation can be carried out to show that the weights can be found by solving the linear system $A_k \mathbf{w}_k = \mathbf{I}_k$ with $(n+M) \times (n+M)$ matrix

$$A_k = \begin{bmatrix} \Phi_k^T & P_k \\ P_k^T & 0 \end{bmatrix}.$$

The $n \times n$ submatrix Φ_k is made up of the RBFs evaluated at each point in $\mathcal{N}_k$, that is

$$\Phi_{k,ij} = \phi\left(\left\|\mathbf{x}_{k,i} - \mathbf{x}_{k,j}\right\|\right), \text{ for } i,j = 1,2,\ldots,n.$$

Likewise the $n \times M$ matrix $P_{k,il}$ consists of the polynomial basis evaluated at each point in $\mathcal{N}_k$ so that

$$P_{k,il} = \pi_l(\mathbf{x}_{k,i}), \text{ for } i = 1,2,\ldots,n \text{ and } l = 1,2,\ldots,M.$$

If $k \in \mathcal{K}_I$, the right hand side, $\mathbf{I}_k$, includes integrals of the basis functions over t_k only. That is,

$$I_{k,j} = \begin{cases} \iiint\limits_{t_k} \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k,j}\right\|\right) dV & j = 1,2,\ldots,n \\ \iiint\limits_{t_k} \pi_{j-n}(\mathbf{x}) dV & j = n+1, n+2, \ldots, n+M \end{cases}.$$

The integrals of the trivariate polynomial terms can be evaluated exactly via, for instance, the Divergence Theorem or barycentric coordinates. For the RBFs, the integrals can be evaluated by further decomposing t_k into four tetrahedra that share a common vertex. Summing the integrals of the RBFs over the four tetrahedra results in the integral over t_k. This process allows the volume integral over a tetrahedron to be reduced to four integrals in a single dimension. Section 2.3.1 explains this process.

On the other hand, for $k \in \mathcal{K}_S$

$$I_{k,j} = \begin{cases} \iiint\limits_{t_k} \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k,j}\right\|\right) dV + \iiint\limits_{s_k} \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k,j}\right\|\right) dV & j = 1,2,\ldots,n \\ \iiint\limits_{t_k} \pi_{j-n}(\mathbf{x}) dV + \iiint\limits_{s_k} \pi_{j-n}(\mathbf{x}) dV & j = n+1,\ldots,n+M \end{cases}.$$

The integrals over t_k are evaluated using the methods described in the previous paragraph while the integrals over s_k are approximated using a scheme discussed in section 2.3.2.

2.3.1 Integrals of RBFs Over Tetrahedra

Suppose that the tetrahedron t_k has vertices $\mathbf{a}_k$, $\mathbf{b}_k$, $\mathbf{c}_k$ and $\mathbf{d}_k$, all points in $\mathbb{R}^3$. Let $\mathbf{x}_{k,j}$ be some point in $\mathbb{R}^3$, which will be common to the four tetrahedra that will be integrated over to obtain the integral over t_k. Although what follows applies for any point in $\mathbb{R}^3$, the point $\mathbf{x}_{k,j}$ is an interpolation node from the set $\mathcal{N}_k$ in this context. A unit length normal vector to the side of t_k with vertices $\mathbf{a}_k$, $\mathbf{b}_k$ and $\mathbf{c}_k$ is defined by

$$\mathbf{n}_{\mathbf{a}_k\mathbf{b}_k\mathbf{c}_k} := \frac{(\mathbf{b}_k - \mathbf{a}_k) \times (\mathbf{c}_k - \mathbf{a}_k)}{\|(\mathbf{b}_k - \mathbf{a}_k) \times (\mathbf{c}_k - \mathbf{a}_k)\|_2},$$

where the order of the vertices matters and should be taken as the order shown in this definition when defining any normal vector in what follows,. Further, let $\mathbf{e}_{k,j}$, $\mathbf{f}_{k,j}$, $\mathbf{g}_{k,j}$ and $\mathbf{h}_{k,j}$ be the orthogonal projections of $\mathbf{x}_{k,j}$ onto the sides of t_k with vertices $\mathbf{a}_k$, $\mathbf{b}_k$ and $\mathbf{c}_k$; $\mathbf{a}_k$, $\mathbf{d}_k$ and $\mathbf{b}_k$; $\mathbf{a}_k$, $\mathbf{c}_k$ and $\mathbf{d}_k$; and $\mathbf{b}_k$, $\mathbf{d}_k$ and $\mathbf{c}_k$, respectively. For instance,

$$\mathbf{e}_{k,j} = \mathbf{x}_{k,j} + \left[(\mathbf{a}_k - \mathbf{x}_{k,j}) \cdot \mathbf{n}_{\mathbf{a}_k\mathbf{b}_k\mathbf{c}_k}\right] \mathbf{n}_{\mathbf{a}_k\mathbf{b}_k\mathbf{c}_k}. \tag{4}$$

Then by applying the divergence theorem it can be shown that

$$\begin{aligned}
\iiint_{t_k} \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k,j}\right\|_2\right) dV = \\
\Bigg\{ &\operatorname{sign}\left(\left(\mathbf{x}_{k,j} - \mathbf{e}_{k,j}\right) \cdot \mathbf{n}_{\mathbf{a}_k\mathbf{b}_k\mathbf{c}_k}\right) \iiint_{t_{\mathbf{x}_{k,j}\mathbf{a}_k\mathbf{b}_k\mathbf{c}_k}} \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k,j}\right\|_2\right) + \cdots \\
&\operatorname{sign}\left(\left(\mathbf{x}_{k,j} - \mathbf{f}_{k,j}\right) \cdot \mathbf{n}_{\mathbf{a}_k\mathbf{d}_k\mathbf{b}_k}\right) \iiint_{t_{\mathbf{x}_{k,j}\mathbf{a}_k\mathbf{d}_k\mathbf{b}_k}} \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k,j}\right\|_2\right) + \cdots \\
&\operatorname{sign}\left(\left(\mathbf{x}_{k,j} - \mathbf{g}_{k,j}\right) \cdot \mathbf{n}_{\mathbf{a}_k\mathbf{c}_k\mathbf{d}_k}\right) \iiint_{t_{\mathbf{x}_{k,j}\mathbf{a}_k\mathbf{c}_k\mathbf{d}_k}} \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k,j}\right\|_2\right) + \cdots \\
&\operatorname{sign}\left(\left(\mathbf{x}_{k,j} - \mathbf{h}_{k,j}\right) \cdot \mathbf{n}_{\mathbf{b}_k\mathbf{d}_k\mathbf{c}_k}\right) \iiint_{t_{\mathbf{x}_{k,j}\mathbf{b}_k\mathbf{d}_k\mathbf{c}_k}} \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k,j}\right\|_2\right) \Bigg\}.
\end{aligned}$$

This expression for the integral over t_k contains integrals over the four tetrahedra that share $\mathbf{x}_{k,j}$ as a common vertex. Consider

$$\iiint_{t_{\mathbf{x}_{k,j}\mathbf{a}_k\mathbf{b}_k\mathbf{c}_k}} \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k,j}\right\|_2\right) dV$$

since the remaining integrals over the tetrahedra are analogous. The integrand is radially symmetric about $\mathbf{x}_{k,j}$ and depends only on the distance from $\mathbf{x}_{k,j}$ suggesting the change of variables

$$\mathbf{x}(\sigma, \lambda_1, \lambda_2) = \mathbf{x}_{k,j} + \sigma(\lambda_1\mathbf{a}_k + \lambda_2\mathbf{b}_k + (1 - \lambda_1 - \lambda_2)\mathbf{c}_k - \mathbf{x}_{k,j}).$$

In this change of variables, triangles similar to the side of t_k with vertices $\mathbf{a}_k$, $\mathbf{b}_k$ and $\mathbf{c}_k$ are parameterized using barycentric coordinates and scaled by the nonnegative parameter σ, which also accounts for the distance of the triangle from the point $\mathbf{x}_{k,j}$. Under this change of variables the integral becomes

$$\int_0^1 \int_0^{1-\lambda_1} \int_0^{\sigma_{\mathbf{a}_k\mathbf{b}_k\mathbf{c}_k}} \phi\left(\sigma\left\|\lambda_1\mathbf{a}_k + \lambda_2\mathbf{b}_k + (1 - \lambda_1 - \lambda_2)\mathbf{c}_k - \mathbf{x}_{k,j}\right\|_2\right) \sigma^2 V_{k,j} d\sigma d\lambda_2 d\lambda_1,$$

where $V_{k,j} = \left|(\mathbf{a}_k - \mathbf{x}_{k,j}) \cdot \left[(\mathbf{b}_k - \mathbf{x}_{k,j}) \times (\mathbf{c}_k - \mathbf{x}_{k,j})\right]\right|$ is six times the volume of the tetrahedron $t_{\mathbf{x}_{k,j}\mathbf{a}_k\mathbf{b}_k\mathbf{c}_k}$. This volume appears in the Jacobian determinant from the change of variables. Also,

$$\sigma_{\mathbf{a}_k\mathbf{b}_k\mathbf{c}_k} = \frac{\left(\frac{1}{3}(\mathbf{a}_k + \mathbf{b}_k + \mathbf{c}_k) - \mathbf{x}_{k,j}\right) \cdot \mathbf{n}_{\mathbf{a}_k\mathbf{b}_k\mathbf{c}_k}}{\left(\mathbf{c}_k - \mathbf{x}_{k,j}\right) \cdot \mathbf{n}_{\mathbf{a}_k\mathbf{b}_k\mathbf{c}_k}}$$

is the value of σ corresponding to the side of t_k with vertices $\mathbf{a}_k$, $\mathbf{b}_k$ and $\mathbf{c}_k$.

Now, in the case of $\phi(r) = r^{2p+1}$, $p = 0, 1, 2, \ldots$, the iterated integrals in σ and then λ_2 can be computed in closed form. However, exploring the integration over λ_1 in Mathematica indicates the cost of a closed form expression for the integral is computationally too expensive, so the proposed algorithm uses standard pseudospectral methods for evaluating the integrals over λ_1.

2.3.2 Integrals Over Slivers of Volume at the Surface

When assigning a sliver of volume to a particular tetrahedron, t_k, care must be taken so that there are no gaps or overlaps between adjacent slivers. Let $\tau_{k,i}$, $i = 1, 2, 3, 4$, be the triangular faces of t_k. At least one of these faces has all three vertices on the surface of the sphere. In most cases, this will be only one face of t_k (particularly when the volume is well resolved by small enough tetrahedra), call it $\tau_{k,*}$. It turns out that if the three edges of $\tau_{k,*}$ are projected radially from the center of the sphere to the surface, gaps and overlaps will be prevented. For each edge of $\tau_{k,*}$ the area between the arc on the sphere surface and the edge of the triangle forms a side of the sliver of volume. The boundary of the sliver volume is formed by all three of these sides, the spherical triangle on the surface of the sphere between the three sides, and the triangle $\tau_{k,*}$. Figure 1 illustrates one of these volumes.

Assigning the slivers of volume in this way provides for a transformation of the coordinates of the sliver which allows the integral over the volume to be written as an iterated integral over a triangular area and a parameter, σ, which relates to the projection from the origin. Consider any point $\mathbf{x}$ in the volume. The vector $\mathbf{x}$ intersects the plane containing $\tau_{k,*}$ at a point $\mathbf{x}'$ that is inside the triangle $\tau_{k,*}$. All of the points inside $\tau_{k,*}$ can be parameterized by, for instance,

$$\mathbf{x}'(\lambda, \mu) = (1 - \lambda)\mathbf{a}_k + \lambda\left((1 - \mu)\mathbf{b}_k + \mu\mathbf{c}_k\right), \ 0 \le \lambda \le 1 \text{ and } 0 \le \mu \le 1,$$

where $\mathbf{a}_k$, $\mathbf{b}_k$ and $\mathbf{c}_k$ are now representing the vertices of $t_{k,*}$. With this parameterization of $t_{k,*}$ any point $\mathbf{x}$ in the volume of the sliver can be represented as

$$\mathbf{x}(\lambda, \mu, \sigma) = \left(1 + \frac{\sigma}{\|\mathbf{x}'(\lambda, \mu)\|_2}\right)\mathbf{x}'(\lambda, \mu),$$

where $0 \le \sigma \le \rho - \left\|\mathbf{x}'(\lambda, \mu)\right\|_2$. Here σ measures the distance from $\mathbf{x}'$ to $\mathbf{x}$. With this parameterization, for instance,

$$\iiint_{s_k} \phi\left(\left\|\mathbf{x} - \mathbf{x}_{k,j}\right\|\right) dV = \int_0^1 \int_0^1 \int_0^{\rho - \left\|\mathbf{x}'(\lambda,\mu)\right\|_2} \phi\left(\left\|\mathbf{x}(\lambda, \mu, \sigma) - \mathbf{x}_{k,j}\right\|\right) |J(\sigma, \lambda, \mu)| \, d\sigma d\lambda d\mu,$$

where J is the Jacobian determinant of $\mathbf{x}$ with respect to σ, λ and μ. The integrals in σ, λ, and μ can be easily treated with any number of quadrature methods over an interval.

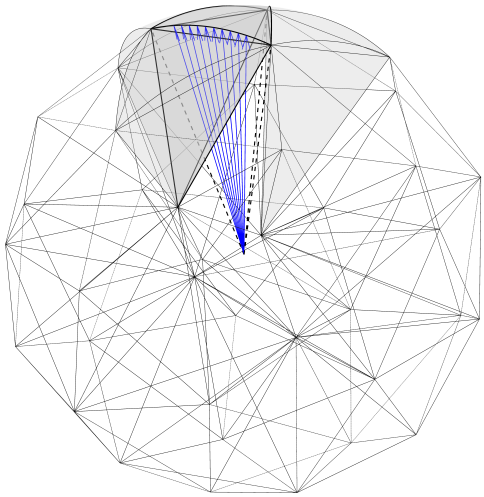

Fig. 1 An illustration of the tetrahedra in the set T. The volume of one of the tetrahedra near the surface is outlined by thicker curves. Call the outlined tetrahedron t_k and let $\tau_{k,*}$ be the face of t_k with three vertices on the surface of the sphere. The arrows indicate the projection of one of $\tau_{k,*}$'s edges to the surface from the origin. The dashed lines indicate the projection of $\tau_{k,*}$'s vertices from the origin. When decomposing the volume of the sphere, the area of the triangle $\tau_{k,*}$, the areas between the arcs on the sphere and the edges of $\tau_{k,*}$ and the area of the spherical triangle between these arcs make of the boundary of the sliver of volume associated with t_k. The projection ensures that between adjacent slivers there are no gaps or overlaps, illustrated by the slivers of volume associated with three adjacent tetrahedra.

2.4 Step 4: Combine weights from the subdomains

Summing over all $k \in \{1, 2, \ldots, K\}$ leads to the approximation of the volume integral over Ω

$$\iiint_{\Omega} f(\mathbf{x}) dV \approx \sum_{k=1}^{K} \sum_{j=1}^{n} w_{k,j} f(\mathbf{x}_{k,j}).$$

Let $\mathcal{K}_i$, $i = 1, 2, \ldots, N$, be the set of all pairs (k, j) such that $\mathbf{x}_{k,j} \mapsto \mathbf{x}_i$. Then the volume integral over Ω can be rewritten as

$$\iiint_{\Omega} f(\mathbf{x}) dV \approx \sum_{i=1}^{N} W_i f(\mathbf{x}_i). \tag{5}$$

3 Test Examples

To demonstrate the performance of the method described herein, the algorithm will be applied to four different test integrands featuring varying degrees of smoothness. The fourth of these test integrands includes an extremely localized feature. Weights are computed on quasi-uniformly spaced nodes, pseudo-randomly generated nodes, and a clustered node set with increased density near the localized feature of the

fourth test integrand. This clustered node set is used demonstrate the performance of the algorithm under localized node refinement. In all of these tests, the radius of the ball is fixed to $\rho = \frac{6^{\frac{1}{3}}}{2\pi^{\frac{1}{3}}}$ so that the volume of the ball is equal to one.

3.1 Node Sets

In the cases of quasi-uniformly spaced nodes and the clustered node sets, quadrature nodes were generated using a modification of the algorithm presented in [17]. This algorithm iteratively places nodes inside an implicitly defined surface by prescribing the lengths of the edges in a sequence of tesselations of the point set. The desired lengths of the edges are given in part by the value of an "edge length function" evaluated at the averages of the two vertices (i.e. nodes) on the edges. In the generation of quasi-uniformly spaced nodes a uniform (constant) edge length function was used. When considering clustered node sets, the edge length function was given by $\sqrt{x^2+y^2+z^2} + \frac{1}{4}\rho$.

The pseudo-randomly spaced node sets were generated by first drawing a set of points from the two-dimensional Halton sequence and mapping the set to the surface of the ball. Then points were drawn from the three-dimensional Halton sequence, mapping the set to the domain $(x, y, z) \in [-\rho, \rho] \times [-\rho, \rho] \times [-\rho, \rho]$ and keeping only points satisfying $x^2 + y^2 + z^2 \leq \rho - \frac{h}{10}$, where h is prescribed to be the average spacing between the nodes on the surface. Examples of the node sets are displayed in figure 2.

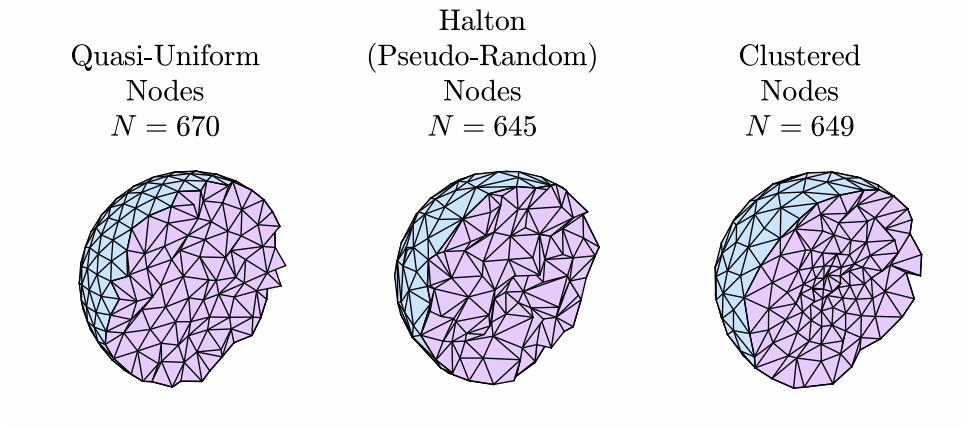

Fig. 2 Examples of the quasi-unifomly spaced, pseudo-randomly spaced, and clustered node sets.

3.2 Performance on Test Integrands

The algorithm was applied to four test integrands. When generating quadrature weights, in all cases the radial basis function was $\phi(r) = r^3$ and the number

of nearest neighbors, $n = (m+1)(m+2)(m+3)$, was based on the trivariate polynomial order m. Some computational experiments in, for instance, [15,18,19] indicated that in the presence of boundaries the number of nearest neighbors must be large enough to overcome effects like Runge phenomenon. The examples given in [15] indicated that the boundary errors were most prominent when nodes were (exactly) uniformly spaced. Therefore, to determine how many nearest neighbors should be included to overcome boundary errors, the algorithm here was modified to compute volume integrals over a cube. When considering a cube the entire volume can be decomposed by tetrahedra, so the algorithm need not consider slivers of volume near the surface. Figure 3 illustrates the absolute error when integrating $f(x,y,z) = \frac{1}{1+((x-x_s)^2+(y-y_s)^2+(z-z_s)^2)}$, with $x_s = 0.234841098236337$, $y_s = 0.048716273957102$ and $z_s = 0.214415743035283$, over the volume of the unit cube centered at the origin for various choices of n and m. The matrix A_k is singular for choices of n below $n = \frac{(m+1)(m+2)(m+3)}{6}$, this is indicated by the lower dashed curve in the figure. Further, for each case of $m = 0, 1, 2, \ldots, 7$ it is clear that choices of n below $n = (m+1)(m+2)(m+3)$ can lead to large errors.

Performing a similar experiment for the unit ball, figure 4 illustrates that even the relaxation from nodes that are exactly uniformly spaced to those that are quasi-uniformly spaced can allow for n as low as $(m+1)(m+2)(m+3)$. However, all further results shown utilize $n = (m+1)(m+2)(m+3)$.

The first of the test integrands is a degree 30 trivariate polynomial. That is, let

$$f_1(x,y,z) = \sum_{\alpha=0}^{30} \sum_{\beta=0}^{\alpha} \sum_{\gamma=0}^{\alpha-\beta} a_{\alpha\beta\gamma} x^{\alpha-\beta-\gamma} y^{\beta} z^{\gamma}.$$

The coefficients of the polynomial are available in a Matlab file at [16]. The exact value of the integral over the ball is

$$\sum_{\alpha=0}^{30} \sum_{\beta=0}^{\alpha} \sum_{\gamma=0}^{\alpha-\beta} a_{\alpha\beta\gamma} \frac{\rho^{\alpha+3}}{8\Gamma\left(\frac{\alpha+5}{2}\right)} \left[\left((-1)^{\beta}+1\right)\left((-1)^{\gamma}+1\right)\left(1+(-1)^{\alpha-\beta-\gamma}\right) \Gamma\left(\frac{\beta+1}{2}\right)\Gamma\left(\frac{\gamma+1}{2}\right)\Gamma\left(\frac{\alpha-\beta-\gamma+1}{2}\right)\right]$$

with Γ the gamma function, and for the set of coefficients used here the expression evaluates to 3.792079311949332. Figure 5 displays convergence of the approximate integral to the exact value at an order better than $\mathrm{O}\left(N^{-\frac{m}{3}}\right)$, where m corresponds to the order of the trivariate polynomial terms used in the approximation. If h refers to a typical node separation distance, this corresponds to a convergence order of better than $\mathrm{O}(h^m)$, especially in the case of quasi-uniformly spaced nodes. The theory in [20] explains that if the multivariate polynomial basis up to degree m is included in the process of RBF interpolation, then all of the terms in the Taylor series up to degree m will be handled exactly for the function being interpolated. The remaining terms in the Taylor series are then approximated by the RBF basis that was included. This is leading to a convergence order of at least $\mathrm{O}(h^m)$.

The second test integrand is the Gaussian

$$f_2(x,y,z) = \exp\left(-10\left((x-x_s)^2+(y-y_s)^2+(z-z_s)^2\right)\right)$$

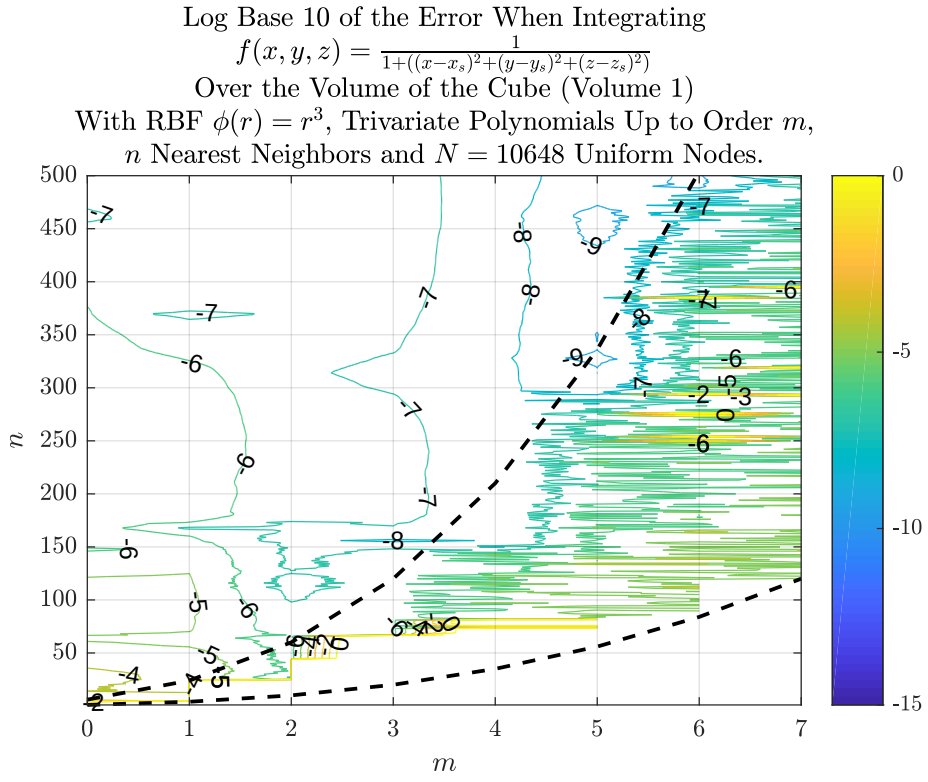

Fig. 3 Log$_{10}$ base 10 of the absolute error when integrating $f(x,y,z) = \frac{1}{1+((x-x_s)^2+(y-y_s)^2+(z-z_s)^2)}$, with $x_s = 0.234841098236337$, $y_s = 0.048716273957102$ and $z_s = 0.214415743035283$, over the volume of the unit cube centered at the origin for various choices of n and m. The lower dashed curve is the function $n = \frac{(m+1)(m+2)(m+3)}{6}$ below which the matrix A_k is guaranteed to be singular. The upper dashed curve is $n = (m+1)(m+2)(m+3)$. Choices of n roughly above this curve lead to more accurate approximations of the integrand in the presence of boundaries.

where

$$(x_s, y_s, z_s) = (0.047056440432708, 0.071766893999009, 0.118950756342700)$$

is a randomly chosen shift of the center of the Gaussian from the origin. In order to have an accurate value to compare to, the volume integral was first approximated by evaluating

$$\int_{-\rho}^{\rho} \int_{-\sqrt{\rho^2-x^2}}^{\sqrt{\rho^2-x^2}} \int_{-\sqrt{\rho^2-x^2-y^2}}^{\sqrt{\rho^2-x^2-y^2}} f_2(x,y,z)\,dz\,dy\,dx,$$

using Matlab's integral3 command with the absolute and relative tolerances both set to ten times machine precision. The resulting approximation of the integral for

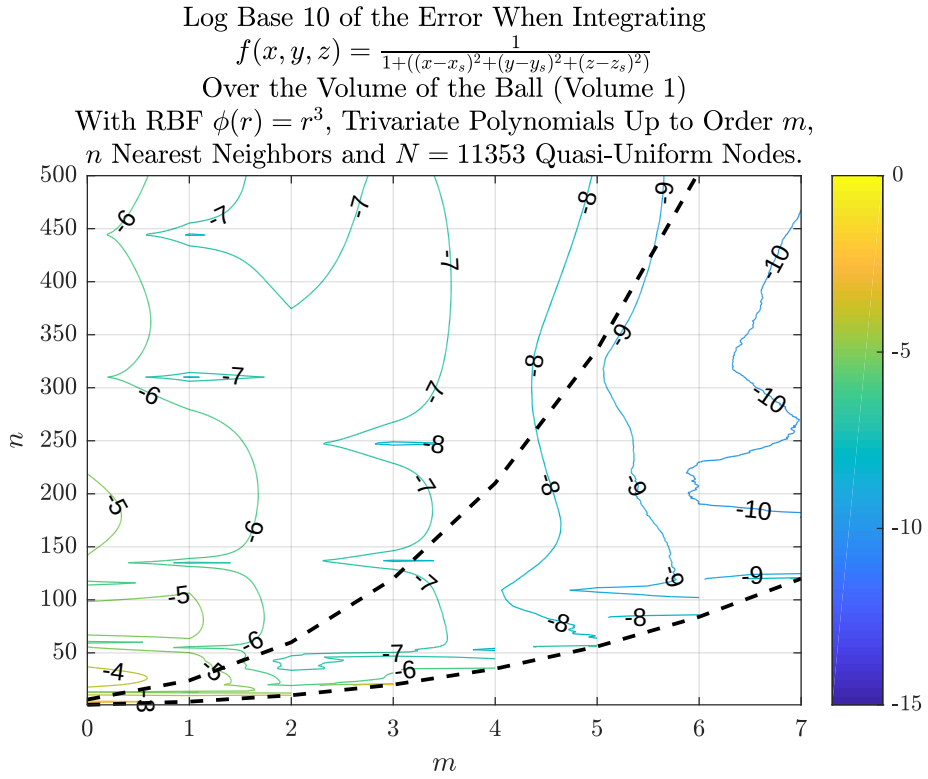

Fig. 4 Log base 10 of the absolute error when integrating $f(x,y,z) = \frac{1}{1+((x-x_s)^2+(y-y_s)^2+(z-z_s)^2)}$, with $x_s = 0.234841098236337$, $y_s = 0.048716273957102$ and $z_s = 0.214415743035283$, over the volume of the unit ball centered at the origin for various choices of n and m. The lower dashed curve is the function $n = \frac{(m+1)(m+2)(m+3)}{6}$ below which the matrix A_k is guaranteed to be singular. The upper dashed curve is $n = (m+1)(m+2)(m+3)$.

f_2 is 0.161965667295343. Figure 6 illustrates the error in the integral of f_2 over the ball when compared to the result from Matlab after rotating the integrand randomly 100 times. It is clear again that the order of the error is most dependent on the degree of the polynomials used in the interpolation.

The third test integrand is

$$f_3(x,y,z) = \text{sign}\,(z)$$

with sign the signum function. This function is discontinuous at the plane $z = 0$, so any method based on a continuous approximation of the integrand across the discontinuity should not be expected to achieve better than $\text{O}\left(N^{-\frac{1}{3}}\right)$ (i.e. $\text{O}(h)$) error. Figure 7 illustrates this for the present method.

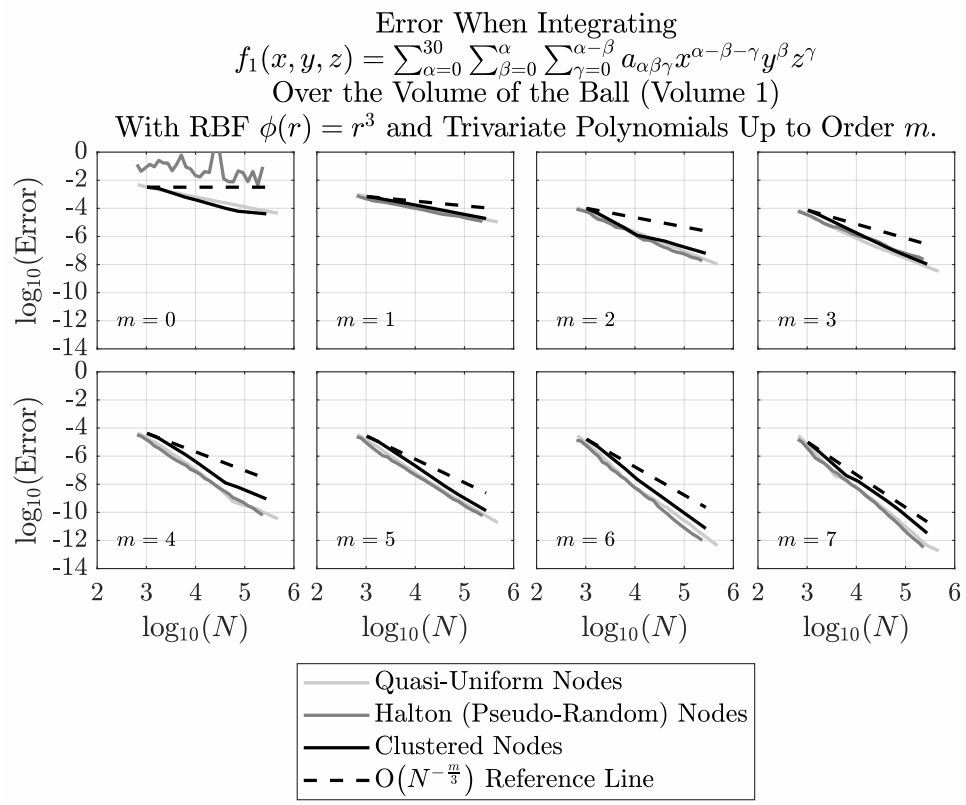

Fig. 5 Log base 10 of the absolute error when approximating the volume integral of f_1 over the ball with radius $\rho = \frac{6^{\frac{1}{3}}}{2\pi^{\frac{1}{3}}}$ centered at the origin. The errors shown here are the largest after rotating the integrand 100 times.

3.3 Performance When Utilizing Clustered Node Sets

To illustrate further utility of the proposed method, the algorithm was also applied to a test integrand featuring a steep and localized gradient. To capture the rapid change in the integrand node sets were generated that feature more densely clustered nodes near the local feature. The test integrand was

$$f_4(x,y,z) = \tan^{-1}\left(5000(x^2+y^2+z^2)\right),$$

which has a steep gradient near the origin. Figure 8 illustrates that in cases where quasi-uniformly spaced or pseudo-randomly spaced node sets cannot capture the changes in the integrand, weights generated for node sets with clustering near features improve the approximation under refinement.

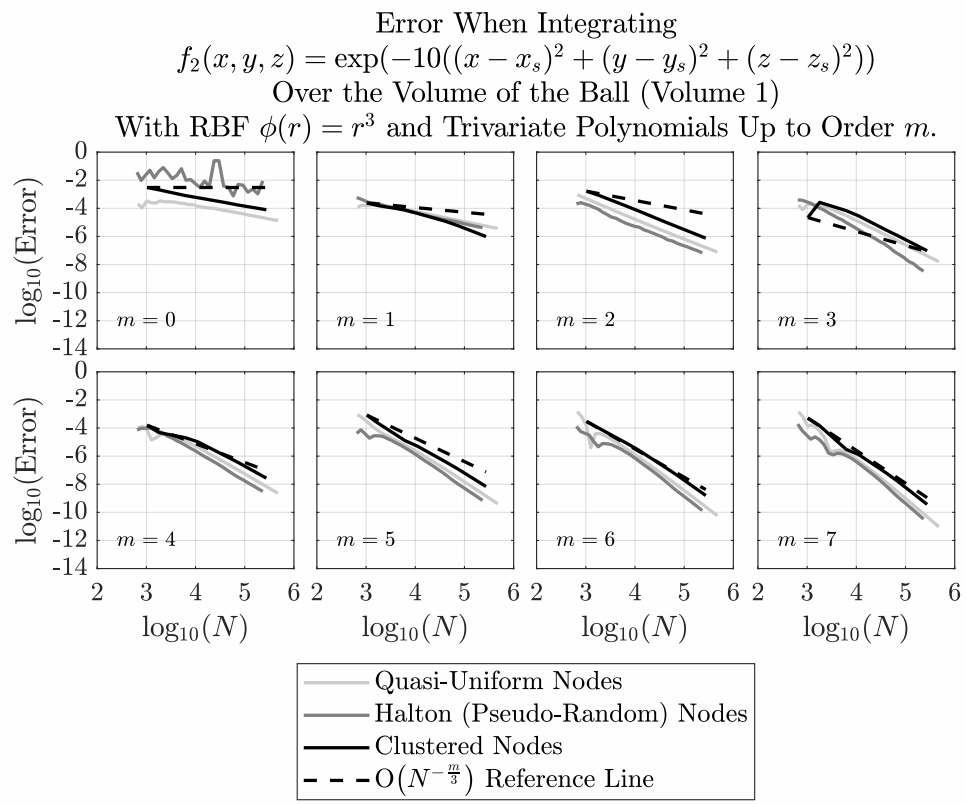

Fig. 6 Log base 10 of the absolute error when approximating the volume integral of f_2 over the ball with radius $\rho = \frac{6^{\frac{1}{3}}}{2\pi^{\frac{1}{3}}}$ centered at the origin. The errors shown here are the largest after rotating the integrand 100 times.

3.4 Computational Expense

Just like the algorithms presented in [13–15], the ability to consider each tetrahedron individually allows the time to compute a set of quadrature weights and the use of memory both to scale like $O(N)$. Figure 9 illustrates the time to compute the set of quadrature weights on N nodes for various choices of the polynomial order, m. Since the choice of m affects the sizes of the systems of linear equations that need to be solved at each iteration, the figure shows an increase in the computational cost as m increases. Further, except for the identification of nearest neighbors in order to construct the local weight set for each tetrahedron/sliver of volume and for the combination of weights in step 4 the algorithm is pleasingly parallel. The parallelization tests in [13] illustrate that these two steps do not have a significant impact on the scalability of the algorithm with the number of cores when considering evaluating surface integrals, and the same is true for the volume integration algorithm described herein.

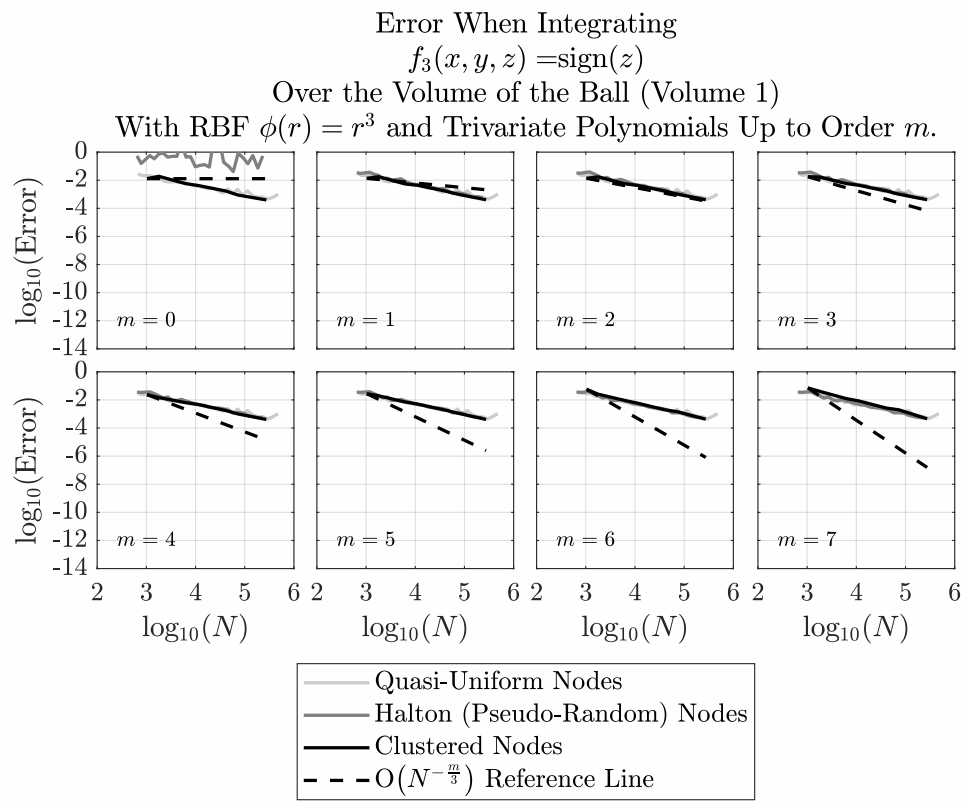

Fig. 7 Log base 10 of the absolute error when approximating the volume integral of f_3 over the ball with radius $\rho = \frac{6^{\frac{1}{3}}}{2\pi^{\frac{1}{3}}}$ centered at the origin. The errors shown here are the largest after rotating the integrand 100 times.

4 Conclusions

This study has supplemented the previous RBF-FD based approach for evaluating definite integrals [13–15] with an extension to integrals over volumes. The computational tests illustrate an algorithm that can achieve at least $O(h^m)$ accuracy, with h the typical node separation distance and m the order of trivariate polynomial basis functions included in the approximation. On a set of N nodes in the ball, the computational cost is only $O(N)$ and the algorithm is pleasingly parallel. A key feature of the algorithm is that it is able to compute quadrature weights on even irregularly spaced or clustered node sets.

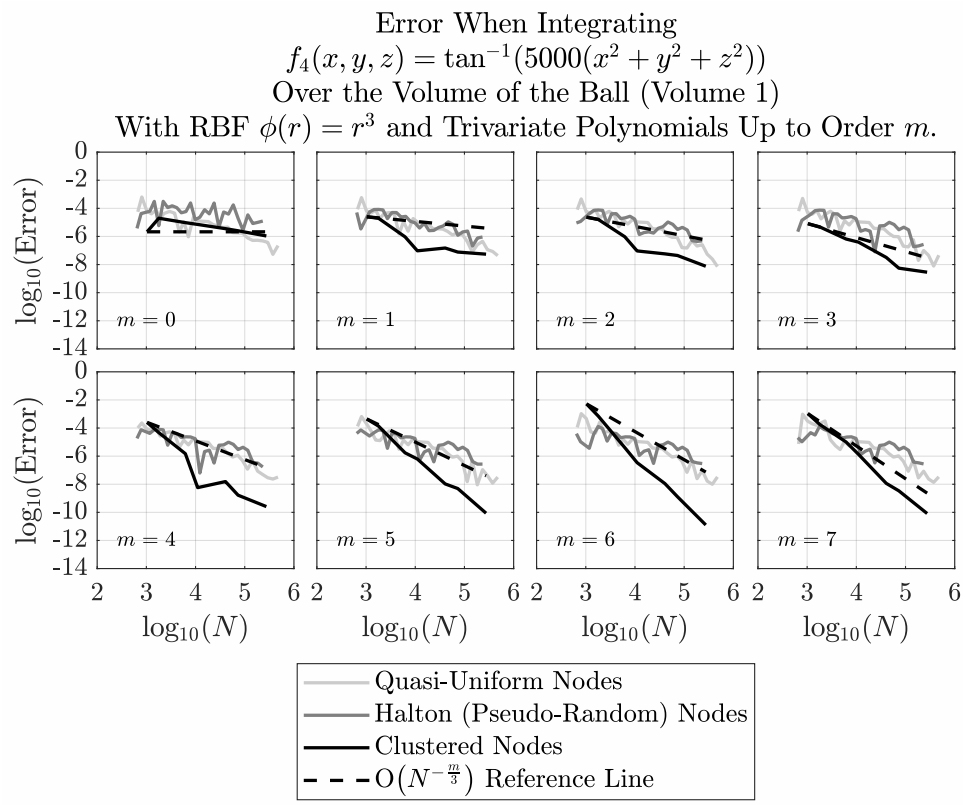

Fig. 8 Log base 10 of the absolute error when approximating the volume integral of f_4 over the ball with radius $\rho = \frac{6^{\frac{1}{3}}}{2\pi^{\frac{1}{3}}}$ centered at the origin.

5 Declarations

Funding

This work was funded by the Office of Naval Research program Atmospheric Propagation Sciences for High Energy Lasers and the Air Force Office of Scientific Research project Radial Basis Functions for Numerical Simulation.

Conflicts of interest/Competing interests

The author declares that he has no conflict of interest.

Availability of data and material

All data including node sets for the ball and cube and all parameters for each test function are available at the link in [16].

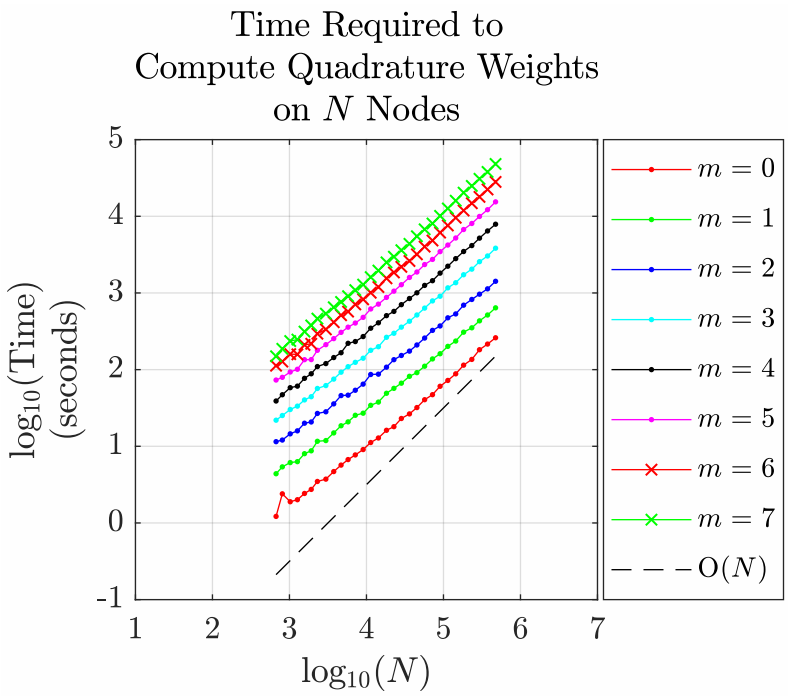

Fig. 9 Log base 10 of the time it takes to compute quadrature weights on N nodes when including trivariate polynomial up to order k. The (black) dashed line is an $\mathrm{O}(N)$ reference line. Computation times were recorded on a desktop workstation running Matlab R2018b on Windows 10 Professional with 256 GB of Random Access Memory and dual Intel Xeon E5-2697 v3 Central Processing Units with 14 cores each running at 2.60 GHz.

Code availability

The proposed algorithm and associated test functions are available at the link in [16]

References

1. P. Slavinić and M. Cvetković. Volume calculation of subsurface structures and traps in hydrocarbon exploration - a comparision between numerical integration and cell based models. *Open Geosci*, 8:14–21, 2016.
2. P. Kuchment and L. Kunyansky. *Mathematics of photoacoustic and thermoacoustic tomography*, volume 1, pages 1117–1167. Springer, 2 edition, 2015.
3. J. G. Leathem. *Volume and surface integrals used in physics*. Number 1. University Press, 1922.
4. W. Freeden and M. Gutting. *Integration and cubature methods: A geomathematically oriented course*. CRC Press, Boca Raton, Florida, United States, 2018.
5. A. H. Stroud. *Approximate calculation of multiple integrals*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, United States, 1971.
6. A. R. Krommer and C. W. Ueberhuber. *Computational integration*. SIAM, Philadelphia, Pennsylvania, United States, 1998.
7. P. K. Kythe and M. R. Schäferkotter. *Computational methods for integration*. Chapman & Hall/CRC, Boca Raton, Florida, United States, 2005.

8. Jr. P. C. Curtis. n-parameter families and best approximation. *Pacific J. Math*, 93:1013–1027, 1959.
9. J. C. Mairhuber. On Haar's theorem concerning Chebyshev approximation problems having unique solutions. *Prc. Amer. Math. Soc.*, 7:609–615, 1956.
10. N. Flyer, G. A. Barnett, and L. J. Wicker. Enhancing finite differences with radial basis functions: Experiments on the Navier–Stokes equations. *J. Comput. Phys.*, 316:39–62, 2016.
11. B. Fornberg and N. Flyer. Solving PDEs with radial basis functions. *Acta Numerica*, 24:215–258, 2015.
12. B. Fornberg and N. Flyer. *A primer on radial basis functions with applications to the geosciences*. SIAM, Philadelphia, U.S., 2015.
13. J. A. Reeger and B. Fornberg. Numerical quadrature over the surface of a sphere. *Stud. Appl. Math.*, 137(2):174–188, 2016.
14. J. A. Reeger, B. Fornberg, and M. L. Watts. Numerical quadrature over smooth, closed surfaces. *P. Roy. Soc. Lon. A Mat.*, 472, 2016. doi: 10.1098/rspa.2016.0401.
15. J. A. Reeger and B. Fornberg. Numerical quadrature over smooth surfaces with boundaries. *J. Comput. Phys.*, 355:176–190, 2018.
16. J. A. Reeger. Volume_Quadrature_RBF_Ball (2019). (https://www.github.com/jareeger/Volume_Quadrature_RBF_Ball), GitHub. Retrieved March 27, 2019.
17. P. Persson and G. Strang. A simple mesh generator in Matlab. *SIAM Review*, 46(2):329–345, June 2004.
18. V. Bayona, N. Flyer, B. Fornberg, and G. A. Barnett. On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs. *J. Comput. Phys*, 332:257–273, 2017.
19. V. Bayona, N. Flyer, and B. Fornberg. On the role of polynomials in RBF-FD approximations: III. Behavior near domain boundaries. *J. Comput. Phys.*, 380:378–399, 2019.
20. V. Bayona. An insight into RBF-FD approximations augmented with polynomials. *Comput. Math. Appl.*, 77(9):2337–2353, 2019.