3-24-2016

Radial Basis Function Based Quadrature over Smooth Surfaces

Maloupu L. Watts

Radial Basis Function Based Quadrature over Smooth Surfaces

THESIS

March 2016

Maloupu L. Watts, Second Lieutenant, USAF

AFIT-ENC-MS-16-M-003

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENC-MS-16-M-003

Radial Basis Function Based Quadrature Over Smooth Surfaces

THESIS

Presented to the Faculty

Department of Mathematical Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science (Mathematics)

Maloupu L. Watts, B.S.

Second Lieutenant, USAF

MARCH 2016

AFIT-ENC-MS-16-M-003

Radial Basis Function Based Quadrature Over Smooth Surfaces

Maloupu L. Watts, B.S.
Second Lieutenant, USAF

Committee Membership:

Capt. Jonah A. Reeger, Ph.D. (Chair)

Benjamin F. Akers, Ph.D. (Member)

Lt. Col. Brian McBee, Ph.D. (Member)

AFIT-ENC-MS-16-M-003

Abstract

The numerical approximation of definite integrals, or quadrature, often involves the construction of an interpolant of the integrand and subsequent integration of the interpolant. It is natural to rely on polynomial interpolants in the case of one dimension; however, extension of integration of polynomial interpolants to two or more dimensions can be costly and unstable. A method for computing surface integrals on the sphere is detailed in the literature (Reeger and Fornberg, *Studies in Applied Mathematics*, 2016). The method uses local radial basis function (RBF) interpolation to reduce computational complexity when generating quadrature weights for a particular node set. This thesis expands upon the same spherical quadrature method and applies it to an arbitrary smooth closed surface defined by a set of quadrature nodes and triangulation.

Key words: Quadrature, Radial Basis Functions, RBF, Numerical Integration, Smooth Surface, Interpolation, Closed Surface

I would like to dedicate this work to my husband, who has offered constant support in conducting research, offering an additional technical view, and encouraging my completion of this thesis.

Acknowledgements

I would like to thank *Capt. Jonah A. Reeger, Ph.D.* for working with me as my advisor. His professionalism and academic excellence helped to direct this research. I would also like to thank my two committee members, *Dr. Benjamin Akers* and *Lt. Col. Brian McBee*, for all their contributions and recommendations.

Maloupu L. Watts

Contents

Page

List of Figures

List of Tables

Radial Basis Function Based Quadrature Over

Smooth Surfaces

I. Literature Review

1.1 Background Motivation

An increasing number of applications, arising for example in geophysics, require partial differential equations (PDEs) to be solved on irregular geometries [1]. A numerical solution of PDEs is often paired with numerical quadrature over smooth (closed) surfaces in order to obtain integrated quantities, such as total energy, average temperature, etc. [2]. Accurate approximation of PDEs and integrals on a surface often requires the construction of node sets featuring spatially varying densities in order to capture rapidly changing features (viz. regions of large curvature). This thesis presents an approach for calculating quadrature weights for such node sets to be used in integral approximations over a smooth closed surface S. That is,

$$\mathcal{I}_S(f) := \iint_S f(x,y,z)dS \approx \sum_{i=1}^{N} W_i f(x_i,y_i,z_i) =: \tilde{\mathcal{I}}_S(f) \tag{1}$$

where $f(x,y,z)$ is a scalar function known at some N quadrature nodes and $\mathcal{W}_N$: $\{W_i\}_{i=1}^N$ are the corresponding quadrature weights.

Earlier literature focused on computing surface integrals on the sphere, or spherical quadrature, over very specific node sets, partnered with tabulated weights for select values of N (the total number of nodes) [3, 4, 5]. More recent literature borrowed concepts from radial-basis-function-generated finite differences (RBF-FD) applied to spatially variable node sets for spherical quadrature [2]. This more recent technique

for spherical quadrature is expanded upon in Chapter II and adapted for use in approximating integrals of arbitrary smooth closed surfaces. The need for approximations to surface integrals on surfaces other than the sphere arises in applications that vary from geophysics, where the earth may be treated as an ellipsoid rather than a sphere, to biology, where particle interactions may require models more complicated than colliding spheres (see, for example, [6] and references therein).

This chapter introduces the inherent benefits of using radial basis functions (RBFs) as an approximation basis for node sets featuring spatially variable density. These benefits, which include high orders of accuracy when integrating the interpolant (e.g. $O(N^{-3.5})$ for a sphere on nearly uniformly spaced points) and relatively cheap computation cost (e.g. $O(N\log N)$ for the techniques borrowed from [2]), are why RBFs were chosen for use during interpolation in this thesis.

1.2 Radial Basis Functions

The approximation of a multivariate function $f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^d$ (d the dimension), can be accomplished in many ways - for example, the use of multivariate polynomials as a basis [7]. When moving beyond one dimension, Mairhuber [8] showed that interpolation with a set of basis functions that do not depend on the data locations is an ill-posed problem. That is, an infinite number of configurations of the points where the interpolation conditions are enforced lead to a singular linear system of equations on the interpolation coefficients. Multivariate polynomial interpolation suffers from this problem. Radial basis function interpolation, on the other hand, uses (as the set of basis functions) translates of a single function $\{\phi(r_i)\}_{i=1}^N$, where $r_i = ||\mathbf{x} - \mathbf{x}_i||$ is the distance of any point $\mathbf{x}$ in the observational domain to the point $\mathbf{x}_i$ (where an interpolation condition will be enforced). As such, RBFs do not suffer from the same problem as multivariate polynomials and become useful in this regard.

Not only can radial basis functions be applied for any number of dimensions, but they also have excellent approximation properties. They can be applied on meshless data sets, allowing a user to easily enhance interpolation properties over certain data features. For example, there may be unique results occurring for a certain region of interest. Clustering data nodes near the region of interest reduces uniformity in data fidelity. However, radial basis functions have the ability to work regardless of this constraint. Examples of RBFs can be found in table 1 [9, 10], where ε is a non-negative parameter for the infinitely smooth RBFs that affects the RBF shape.

Table 1. Types of Radial Basis Functions $\phi(r)$

Piecewise Smooth (Conditionally Positive Definite)		
MN	Monomial	$\|r\|^{2m+1}$
TPS	thin plate spline	$\|r\|^{2m}\ln\|r\|$
Infinitely Smooth (Positive Definite)		
MQ	Multiquadric	$\sqrt{1+(\varepsilon r)^2}$
IQ	Inverse quadric	$\frac{1}{1+(\varepsilon r)^2}$
IMQ	Inverse MQ	$\frac{1}{\sqrt{1+(\varepsilon r)^2}}$
GA	Gaussian	$e^{-(\varepsilon r)^2}$
BE	Bessel	$\frac{J_{\frac{d}{2}-1}(\varepsilon r)}{(\varepsilon r)^{\frac{d}{2}-1}}$

1.3 History of RBFs

Radial basis functions are a more recent development in mathematics. Unconditional nonsingularity of the interpolation problem for a specific case was given by German scientist Bochner in 1933 [11]. This was followed by the use of positive definite functions, a class of functions to which many RBFs belong, in specific metric

spaces in 1938 by Schoenberg [12]. It was not until 1971, however, that Hardy applied multiquadric (MQ) functions to applications in topography given coordinate data for irregular surfaces [13]. The multiquadric functions are radial functions of the form indicated in table 1.

In Hardy's work, RBFs were used to interpolate irregular smooth surfaces given discrete data [13]. In 1990, he published another article summarizing MQ and multiquadric-biharmonic (MQ-B) methods for use in such applications as geodesy, geophysics, surveying and mapping, geography, remote sensing, signal processing, and more [1]. After this emergence of MQ equations, radial basis functions became increasingly popular.

Also in 1990, Kansa published a paper using multiquadrics as a basis for approximating the solution to partial differential equations [14]. He pointed out that these functions are continuously differentiable and integrable over a meshless grid, and are therefore useful for representing other functions in the setting of integral and differential equations. Their simple extension to higher dimensions makes multiquadrics and other radial basis functions great candidates for solving PDEs.

The construction of a RBF interpolant can be computationally intensive, particularly if the node set $\mathcal{S}_N$: $\{\mathbf{x}_i\}_{i=1}^N$ (where the interpolation conditions are to be enforced) is large. In the presence of N interpolation conditions on N nodes, a dense linear system of equations of size $N \times N$ must be solved. Further complicating the matter is that these interpolants are often used to construct approximate differential operators (as in the case of [14]) whose application is through a dense matrix multiplication. This computational complexity inspired the development of RBF-generated finite differences (RBF-FD) [15, 16, 17, 18], which considers for each node $\mathbf{x}_i \in \mathcal{S}_N$ a subset $\mathcal{N}_n \subset \mathcal{S}_N$ of n points nearest $\mathbf{x}_i$ and constructs an interpolant (and subsequent approximate differential operator) on that set. This reduces computational cost to N

linear systems of size $n \times n$ and an approximate differential operator that is sparse [19]. The differences between global RBF approximations and RBF-FD has analogy to those between pseudo-spectral methods and finite differences for approximating derivatives.

1.4 Numerical Quadrature over Surfaces

The strong approximation properties of radial basis functions in interpolation and the solution of PDEs led to their consideration for approximations to surface integrals. Reeger and Fornbeg discuss numerical quadrature over the surface of a sphere [2], forming the building blocks for this thesis. As a result, much of their work is reiterated in Chapter II for completeness. Various integration methods on the sphere are written by [5, 20, 21, 22, 23]. For instance, [23] proposes spherical harmonics as a set of basis functions for interpolation and analyzes the variability in results for using different quasi-uniform node sets on $\mathbb{S}^2$. On the other hand, when applying the associated quadrature weights to approximate a surface integral, Womersley and Sloan [5] suggest the use of exact cubature weights (quadrature weights in $\mathbb{R}^3$) associated with polynomials of certain degrees, while also analyzing the use of various quasi-uniform node sets on $\mathbb{S}^2$.

Some of the methods for numerically computing surface integrals on the sphere have been adapted to other surfaces [24, 25]. Just as this thesis adapts from a spherical quadrature method [2] to approximate an integral over a smooth closed surface $S \subset \mathbb{R}^3$, Atkinson [24] also adapts spherical quadrature methods to solve integral equations defined over simple smooth surfaces. He presents two general methods: product Gaussian quadrature and finite element integration. The product Gaussian quadrature uses Gauss-Legendre nodes and quadrature weights for the unit sphere $\mathbb{S}^2$, with convergence rate $O((2N^{-1/2}-1)^{-u})$ on nearly uniform node sets (u the num-

ber of times f in (1) is continuously differentiable). The finite element integration discretizes $\mathbb{S}^2$ into triangles, with convergence rate $O(K^{-1})$ (K the number of triangles) when using spherical triangles, and $O(K^{-2})$ when mapping spherical triangles to planar triangles. He interpolates integrals over simple smooth surfaces using a one-to-one mapping with $\mathbb{S}^2$ coupled with one of his two methods.

Fuselier et al. [25] developed quadrature methods that extend to manifolds $\mathbb{M}$ that are either homogeneous (including $\mathbb{S}^2$) or diffeomorphic to homogeneous spaces. Their methods result in a convergence rate of roughly $O(N^{-2})$ for sufficiently smooth integrands.

Additional types of quadrature for surfaces in $\mathbb{R}^3$ can be found at [26, 27, 28]. For example, [26] discusses numerical quadrature for piecewise smooth surfaces using polynomial interpolants, while [27] couples Thin-Plate Spline interpolation (see table 1) with Green's integral formula [29] for a meshless cubature in $\mathbb{R}^3$. Quadrature involving a Galerkin discretization of a boundary integral equation with a weakly singular kernel is adapted from electromagnetics to a general framework in [28].

In any case, a common theme throughout the development of each of these methods is that they are fixed to particular node sets featuring near-uniform density in order to achieve stability. Some of these can achieve high orders of accuracy, even spectral, on the near-uniform data sets, but at the expense of a lengthy, often intractable, process for constructing the node sets themselves. For further discussion of node sets on $\mathbb{S}^2$, see section 2.1.1.

1.5 Organization of Thesis

The aim of this thesis is to evolve the spherical quadrature method described in the literature [2] and apply it to approximate the surface integral of a scalar function over an arbitrary smooth closed surface. Chapter II walks through the steps involved

in applying the spherical quadrature method, from discretizing a sphere surface to computing the quadrature weights. This is followed in Chapter III by a similar procedure applied to an arbitrary smooth surface. Chapters IV and V include some results of the methods described in Chapter III and possible future considerations.

II. Spherical Quadrature

Much of the basis behind the method for numerical quadrature over a smooth closed surface comes from the techniques used for spherical quadrature discussed in [2]. The purpose of this chapter is to explain the construction of the set of weights $\mathcal{W}_N$ in (1) for S the surface of a sphere with radius ρ centered at the origin. The N nodes make up the set $\mathcal{S}_N$: $\{(x_i, y_i, z_i)\}_{i=1}^N$. This notation will be consistent throughout this chapter. The process to construct the quadrature weights can be summarized by

1. Discretize the sphere surface using a node set $\mathcal{S}_N$ and triangulation $\mathcal{T}$.

2. For each triangle $\tau_k \in \mathcal{T}$, $k = 1, ..., K$, project a local region S_{Ω_k} containing n surface nodes (from $\mathcal{S}_N$) and corresponding triangles to a plane tangent to the sphere.

3. Interpolate the transformed integrand over n projected points $\mathcal{N}_n \subset \mathcal{S}_N$ in the tangent plane using radial basis functions as a basis.

4. Integrate the interpolant over a triangle in the tangent plane region Ω_k to get planar quadrature weights

5. Convert the planar quadrature weights to spherical quadrature weights to be used in (1)

This process will set the foundation for understanding the surface quadrature presented in Chapter III.

2.1 Discretizing the Surface S

2.1.1 Types of Node Sets.

The sphere surface $\mathbb{S}^2$ can be easily parameterized, and hence many discretization processes exist to yield a desired set of surface nodes $\mathcal{S}_N \subset \mathbb{S}^2$. The approximation in (1) requires such a set of quadrature nodes $\mathcal{S}_N$ of N nodes in $\mathbb{R}^3$. Various types of node sets exist for $\mathbb{S}^2$, including the following examples illustrated in figure 1. Define quasi-uniform to mean that the spatial density of a set of points on $\mathbb{S}^2$ is near uniform (such node sets are necessary when considering the methods discussed in section 1.4). That is, each point on $\mathbb{S}^2$ accounts for approximately the same surface area. Some quasi-uniform node sets include Fibonacci, Icosahedral, minimum energy, and maximum determinant. The Fibonacci node set [30] creates a mesh such that each node on $\mathbb{S}^2$ represents roughly the same area (see section 2.3.2 on fill distance). Meanwhile, the Icosahedral node set discussed in [25] subdivides the triangular facets of the icosahedron. Quasi-minimum energy nodes in [25], on the other hand, treat the nodes as point charges and nearly minimize the Riesz energy over possible node configurations [31, 32]. Another minimum energy node set locally minimizes potential energy [5]. The last depicted quasi-uniform node set, the maximum determinant node set [5], locally maximizes the determinant of the interpolation matrix on the basis of spherical harmonics. These node distributions vary such that they should be chosen based on the application.

To illustrate the geometric flexibility of the present spherical quadrature method, this thesis considers the Halton node set that maps the first N points from the Halton sequence in $\mathbb{R}^2$ to the unit sphere $\mathbb{S}^2$ as discussed in [23]. Likewise, the random node set maps N pseudo-random (computer generated) points from the standard normal distribution in $\mathbb{R}^2$ to $\mathbb{S}^2$ also as discussed in [23].

Each of these node sets generates a different set of quadrature weights, with vary-

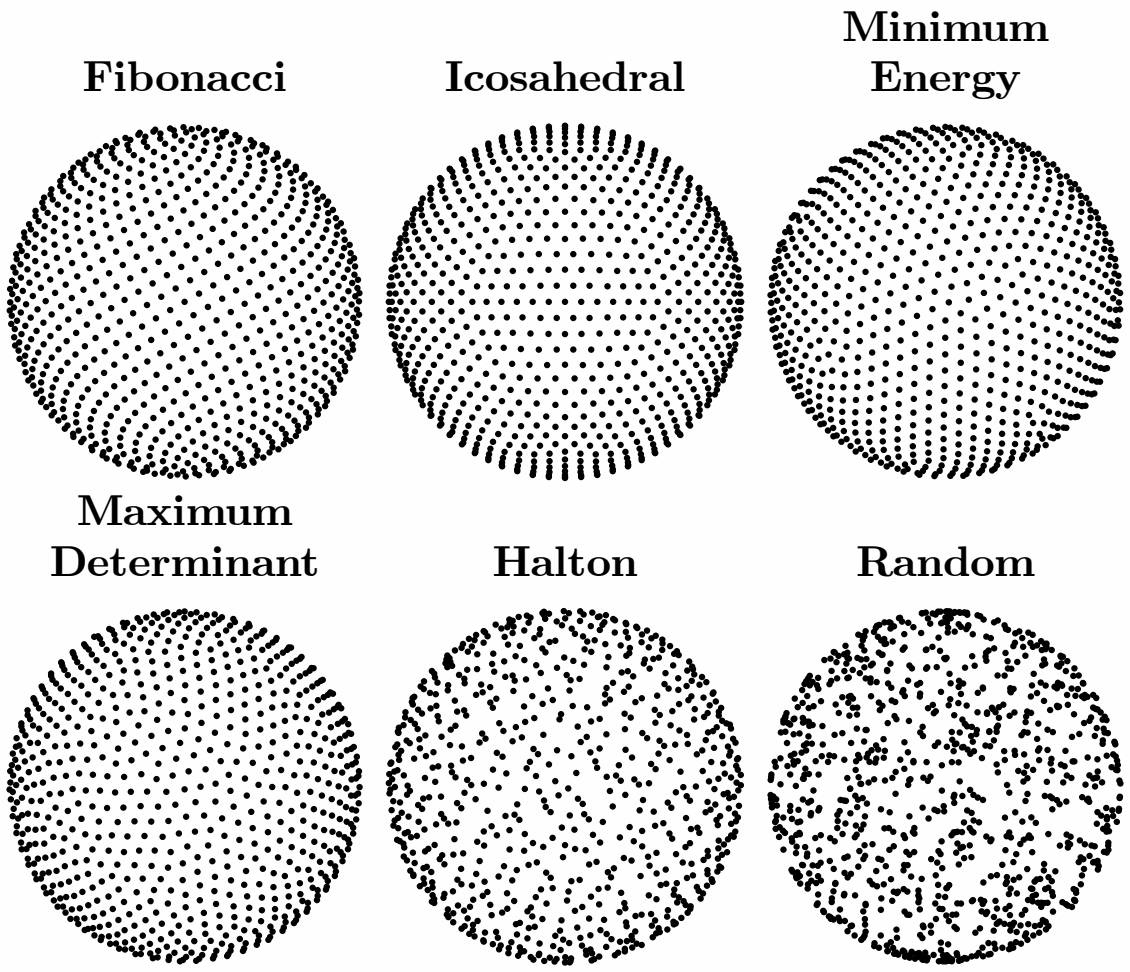


Figure 1. Example node distributions over $\mathbb{S}^2$. Illustrations include $N = 1443$ for the Fibonacci node set, $N = 1442$ for Icosahedral, and $N = 1444$ for the rest.

ing effects on convergence of the methods used to generate them [5, 23].

2.1.2 Delaunay Triangulation.

The set of quadrature nodes $\mathcal{S}_N$ can be constructed in many ways as seen in section 2.1.1. An example of such a set appears in figure 2a. From the set $\mathcal{S}_N$ a spherical triangulation $\mathcal{T} = \{\tau_k\}_{k=1}^{K}$ is constructed [33] such that

- the triangle vertices are the elements of $\mathcal{S}_N$,
- the triangle edges are geodesics between the vertices of the triangles (uniquely

defined as long as the two vertices on the edge are not separated, in angle, by exactly π),

- 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 T covers $\mathbb{S}^2$, and
- no circumcircle of a triangle τ_k contains an element of $\mathcal{S}_N$ on its interior.

An example Delaunay triangulation is illustrated in figure 2b. It is a generalization of the straight line dual of the Voronoi diagram, formally introduced by Delaunay in 1934 [34]. The particular triangulation listed here covers the convex hull of the sphere nodes, and is further discussed by Renka in [33]. Renka also includes an $O(N\log N)$ algorithm for its construction. The triangulation can also be generalized to higher dimensions according to [35].

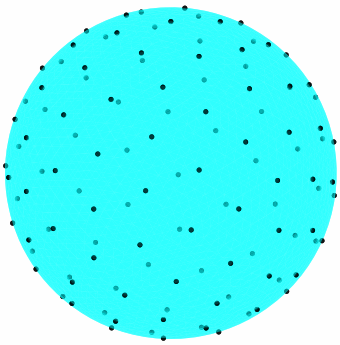
The triangulation $\mathcal{T}$ allows $\mathcal{I}_S(f)$ in (1) to be written as

$$\mathcal{I}_S(f) = \iint_S f(x,y,z)dS = \sum_{k=1}^{K} \iint_{\tau_k} f(x,y,z)dS$$

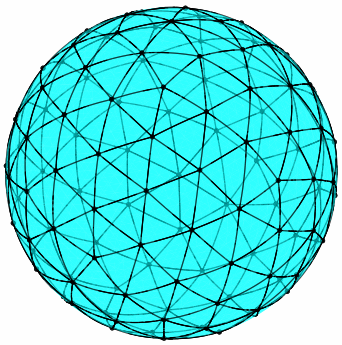
such that each $\tau_k \in \mathcal{T}$, for $k = 1, ..., K$, can be considered independently. That is, quadrature can be performed on each individual triangle before adding up all the quadrature weights to yield the desired weights $\mathcal{W}_N$ for the surface integral $\tilde{\mathcal{I}}_S(f)$.

2.2 Projection into a 2-D plane

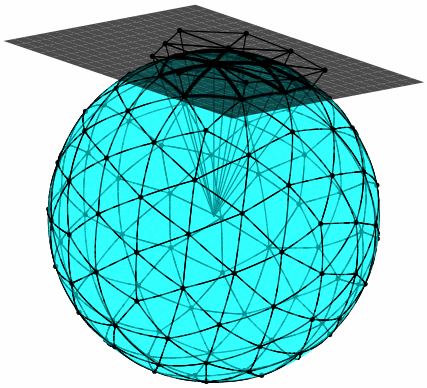
With $\mathcal{S}_N$ and $\mathcal{T}$ in hand, the following subsections will transform a three-dimensional region, S_{Ω_k}, on a sphere surface to a two-dimensional region, Ω_k, on a plane tangent to the sphere.



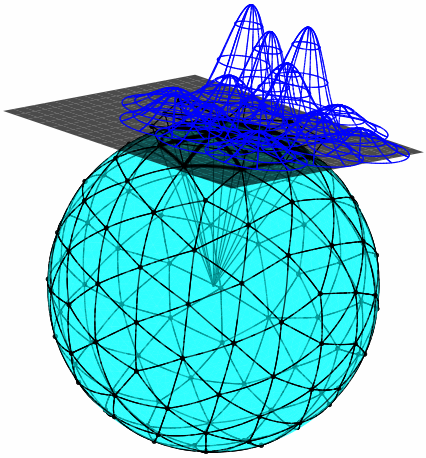
(a) Nodes $\mathcal{S}_N$ distributed over $\mathbb{S}^2$.



(b) Delaunay triangulation $\mathcal{T}$ from section 2.1.2.



(c) Nearest nodes $\mathcal{N}_n$ and surface region S_{Ω_k} projected to tangent plane.



(d) $s(\eta, \xi)$ interpolating $g(\eta, \xi)$ over planar region Ω_k

Figure 2. Progression of quadrature steps from sections 2.1.2 through 2.2.3

2.2.1 Projecting Neighboring Nodes to a Tangent Plane.

In order to reduce the dimensionality of the quadrature problem (1), consider parameterizing each triangle $\{\tau_k\}_{k=1}^K \in \mathcal{T}$ locally through a change of variables. The change of variables is performed via a gnomonic projection [36] into a plane that is tangent to the sphere at a point in each triangle τ_k, as seen in figure 2c. The hallmark of the chosen projection must be that geodesics on the sphere become lines in the tangent plane. This requirement enforces that the surface area of each triangle in $\mathcal{T}$ is not included more than once, and that the surface integral over a spherical triangle τ_k becomes an area integral over a planar triangle $\tilde{\tau}_k$.

To form the projection for some spherical triangle τ_k, let τ_{ABC} be the vertex representation of τ_k - that is, $\mathbf{x}_A, \mathbf{x}_B, \mathbf{x}_C \in \mathcal{S}_N$ are the vertices of τ_k - and define

$$\mathbf{x}_M = [x_M, y_M, z_M] = \frac{\mathbf{x}_A + \mathbf{x}_B + \mathbf{x}_C}{3}$$

As this point lies below the sphere surface, the spherical triangle midpoint is

$$\tilde{\mathbf{x}}_M := [\tilde{x}_M, \tilde{y}_M, \tilde{z}_M] = \frac{\mathbf{x}_M}{||\mathbf{x}_M||_2}\rho$$

(for a sphere of radius ρ) to project $\mathbf{x}_M$ to the sphere surface S. The equation for the tangent plane that passes through $\tilde{\mathbf{x}}_M$ and that has the same normal vector as that of the sphere surface at $\tilde{\mathbf{x}}_M$ is

$$\tilde{x}_m(x - \tilde{x}_m) + \tilde{y}_m(y - \tilde{y}_m) + \tilde{z}_m(z - \tilde{z}_m) = 0$$

since $\tilde{\mathbf{x}}_M - \mathbf{0}$ is the surface normal at $\tilde{\mathbf{x}}_M$.

Consider projecting n neighboring nodes (from $\mathcal{S}_N$ and nearest $\tilde{\mathbf{x}}_M$) into the tangent plane. Call this set of neighboring nodes $\mathcal{N}_n$. Projecting this set of nodes is a

necessity when approximating a surface integral over an entire spherical triangle τ_k via an area integral over a planar triangle $\tilde{\tau}_k$. Let S_{Ω_k} represent the region of the sphere surface S occupied by the neighboring nodes in $\mathcal{N}_n$. The projection occurs by finding the intersection between the plane tangent to the sphere at $\tilde{\mathbf{x}}_M$ and the line passing through a point $(x, y, z) \in S$ near $(\tilde{x}_M, \tilde{y}_M, \tilde{z}_M)$. That is, consider the parameter c such that

$$\tilde{x}_M(x - \tilde{x}_M) + \tilde{y}_M(y - \tilde{y}_M) + \tilde{z}_M(z - \tilde{z}_M) = 0$$

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = c \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

are satisfied simultaneously. Therefore,

$$c = \frac{\tilde{x}_M^2 + \tilde{y}_M^2 + \tilde{z}_M^2}{\tilde{x}_M x + \tilde{y}_M y + \tilde{z}_M z} = \frac{\rho^2}{\tilde{x}_M x + \tilde{y}_M y + \tilde{z}_M z} \tag{2}$$

Notice that a singularity in c occurs where $\tilde{\mathbf{x}}_M \cdot \mathbf{x} = 0$, for $\cdot$ the vector dot product. That is, when the vector in the direction of $\mathbf{x}$ is perpendicular to the vector in the direction of the triangle midpoint $\tilde{\mathbf{x}}_M$. This will become an important factor when approximating the integral of f in the tangent plane since a discrete set of points from $\mathcal{S}_n$ near τ_k, but not within τ_k, will need to be projected as well. As a result, only points $\mathcal{S}_n$ within a 90° angle of $\tilde{\mathbf{x}}_M$ are considered for projection.

2.2.2 Defining a Two-Dimensional System.

Now that the region S_{Ω_k} about a spherical triangle $\tau_k \in \mathcal{T}$ has been projected into a tangent plane, it is natural to define a two-dimensional coordinate system in the plane for computation. Let Ω_k represent the two-dimensional planar region projected

and transformed from S_{Ω_k}. To simplify converting the three-dimensional planar coordinates $\tilde{\mathbf{x}}$: $(\tilde{x}, \tilde{y}, \tilde{z})$ to two-dimensional planar coordinates $\boldsymbol{\omega}$: (η, ξ), consider rotating the coordinates $(\tilde{x}, \tilde{y}, \tilde{z})$ so that $\tilde{\mathbf{x}}_M$ is located along the z-axis at $(0, 0, \rho)$. Furthermore, this would place the tangent plane at $\tilde{\mathbf{x}}_M$ parallel to the (x, y) plane, making all $\tilde{z}$ set equal to ρ. This can be done through a series of rotation matrices

$$M_1 = \begin{bmatrix} \frac{\tilde{x}_M}{\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}} & \frac{\tilde{y}_M}{\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}} & 0 \\ -\frac{\tilde{y}_M}{\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}} & \frac{\tilde{x}_M}{\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } M_2 = \begin{bmatrix} \frac{\tilde{z}_M}{\rho} & 0 & -\frac{\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}}{\rho} \\ 0 & 1 & 0 \\ \frac{\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}}{\rho} & 0 & \frac{\tilde{z}_M}{\rho} \end{bmatrix}$$

where $\rho = \sqrt{\tilde{x}_M^2 + \tilde{y}_M^2 + \tilde{z}_M^2}$. So,

$$M_2 M_1 = \begin{bmatrix} \frac{\tilde{x}_M \tilde{z}_M}{\rho\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}} & \frac{\tilde{y}_M \tilde{z}_M}{\rho\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}} & -\frac{\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}}{\rho} \\ -\frac{\tilde{y}_M}{\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}} & \frac{\tilde{x}_M}{\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}} & 0 \\ \frac{\tilde{x}_M}{\rho} & \frac{\tilde{y}_M}{\rho} & \frac{\tilde{z}_M}{\rho} \end{bmatrix}$$

When performing the rotation on the triangle midpoint,

$$M_2 M_1 \begin{bmatrix} \tilde{x}_M \\ \tilde{y}_M \\ \tilde{z}_M \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \rho \end{bmatrix}$$

verifies the intended rotation. Performing the transformation from three-dimensional coordinates $(x, y, z) \in S_{\Omega_k}$ to two-dimensional coordinates $(\eta, \xi) \in \Omega_k$ (with c in (2)

paired with the rotation matrices) yields the transformation

$$\begin{bmatrix} \eta \\ \xi \\ \gamma \end{bmatrix} = cM_2M_1 \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{\rho[\tilde{z}_M(\tilde{x}_M x+\tilde{y}_M y)-z(\tilde{x}_M^2+\tilde{y}_M^2)]}{\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}(\tilde{x}_M x+\tilde{y}_M y+\tilde{z}_M z)} \\ \frac{\rho^2(\tilde{x}_M y-\tilde{y}_M x)}{\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}(\tilde{x}_M x+\tilde{y}_M y+\tilde{z}_M z)} \\ \rho \end{bmatrix}$$

Just as intended, the last element of the planar coordinates is the radius of the sphere and can therefore be dropped from the two-dimensional coordinate system. This yields the same results as [2]:

$$\begin{aligned} \boldsymbol{\omega} = \begin{bmatrix} \eta \\ \xi \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \xi \\ \gamma \end{bmatrix} \\ &= \begin{bmatrix} \frac{\rho[\tilde{z}_M(\tilde{x}_M x+\tilde{y}_M y)-z(\tilde{x}_M^2+\tilde{y}_M^2)]}{\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}(\tilde{x}_M x+\tilde{y}_M y+\tilde{z}_M z)} \\ \frac{\rho^2(\tilde{x}_M y-\tilde{y}_M x)}{\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}(\tilde{x}_M x+\tilde{y}_M y+\tilde{z}_M z)} \end{bmatrix} \end{aligned} \tag{3}$$

for $\tilde{x}_M^2 + \tilde{y}_M^2 \neq 0$.

That is, when $\tilde{\mathbf{x}}_M$ is already at $(0, 0, \rho)$ (or $(0, 0, -\rho)$) and before applying the rotation matrices M_1 and M_2, the parameter c becomes

$$c = \frac{\rho^2}{\tilde{x}_M x + \tilde{y}_M y + \tilde{z}_M z} = \frac{\rho^2}{\rho z} = \frac{\rho}{z}$$

This results in (3) becoming (for $\tilde{x}_M^2 + \tilde{y}_M^2 = 0$) [2]

$$\begin{aligned}\boldsymbol{\omega} = \begin{bmatrix}\eta \\ \xi\end{bmatrix} &= \begin{bmatrix}1 & 0 & 0 \\ 0 & 1 & 0\end{bmatrix} c \begin{bmatrix}x \\ y \\ z\end{bmatrix} \\ &= \begin{bmatrix}1 & 0 & 0 \\ 0 & 1 & 0\end{bmatrix} \begin{bmatrix}\frac{\rho x}{z} \\ \frac{\rho y}{z} \\ \frac{\rho z}{z}\end{bmatrix} \\ &= \begin{bmatrix}\frac{\rho x}{z} \\ \frac{\rho y}{z}\end{bmatrix}\end{aligned} \tag{4}$$

Hence, a two-dimensional coordinate system is defined for each tangent plane region Ω_k projected from S_{Ω_k} about $\tau_k \in \mathcal{T}$ for $k = 1, ..., K$.

2.2.3 Radial Basis Function Interpolation.

With a two-dimensional coordinate system defined, it is imperative to understand radial basis function (RBF) interpolation in order to interpolate the integrand g in

$$\mathcal{I}_{S_{\Omega_k}}(f) = \iint\limits_{S_{\Omega_k}} f(x, y, z) dS = \iint\limits_{\Omega_k} g(\eta, \xi) d\eta d\xi = I_{\Omega_k}(g)$$

for I_{Ω_k} the area integral over Ω_k (see figure 2d). As mentioned in section 1.3, RBFs are a type of basis function that can be used in interpolation in more than one dimension. Given a set of points (nodes) $\mathcal{N}_n$: $\{\mathbf{x}_j\}_{j=1}^n$ in a domain Ω_k and corresponding measurements or samplings of an unknown scalar function g, $\{g(\mathbf{x}_i)\}_{i=1}^n$, interpolation attempts to approximate g at other locations (besides those from $\mathcal{N}_n$). In practice, $\mathcal{N}_n$ will consist of the n nodes in $\mathcal{S}_N$ nearest $\tilde{\mathbf{x}}_M$ projected to the plane tangent to the sphere at $\tilde{\mathbf{x}}_M$ and transformed to the two-dimensional coordinate system (η, ξ). The

interpolation is done by forming a linear combination of basis functions such that the interpolation formula (interpolant) has the form

$$s(\mathbf{x}) = \sum_{j=1}^{n} c_j \phi \left(||\mathbf{x} - \mathbf{x}_j|| \right), \, \mathbf{x} \in \mathbb{R}^d \tag{5}$$

Here, d is the dimension, $\{c_j\}_{j=1}^n$ the coefficients that must be selected to satisfy interpolation conditions or constraints, and ϕ the set of basis functions. The interpolation constraints

$$s(\mathbf{x}_j) = g(\mathbf{x}_j), \; s, g \in \mathbb{R}, \; j = 1, ..., n \tag{6}$$

are enforced on the linear combination (5) to ensure that the approximation matches exactly at locations where information is known. The radial basis functions $\phi(r_j)$ depend on the radial parameter $r_j = ||\mathbf{x} - \mathbf{x}_j||$, $j = 1, ..., n$, which implies that the value of the radial function depends only on its distance from the center point $\mathbf{x}_j$ - and is thus rotationally invariant. This invariance property plays a key role in the derivation of the quadrature weights in section 2.3.2. The choice of the norm $|| \cdot ||$ can be made based on the application, but is most often the Euclidean two-norm. The definitions of the radial functions differ only on the center point, so the set $\{\phi(||\mathbf{x} - \mathbf{x}_j||)\}_{j=1}^n$ is a collection of translations of the same function. Using the Euclidean two-norm, the interpolant takes the form

$$s(\mathbf{x}) = \sum_{j=1}^{n} c_j \phi \left(\sqrt{(x^{(1)} - x_j^{(1)})^2 + (x^{(2)} - x_j^{(2)})^2 + \cdots + (x^{(n)} - x_j^{(n)})^2} \right) \tag{7}$$

Examples of RBFs that can be used in (7) are found in table 1. In determining the coefficients $\{c_j\}_{j=1}^n$, the interpolation constraints (6) form the linear system

$$AC = G \tag{8}$$

where

$$A = \begin{bmatrix} \phi(||\mathbf{x}_1 - \mathbf{x}_1||) & \phi(||\mathbf{x}_1 - \mathbf{x}_2||) & \cdots & \phi(||\mathbf{x}_1 - \mathbf{x}_n||) \\ \phi(||\mathbf{x}_2 - \mathbf{x}_1||) & \phi(||\mathbf{x}_2 - \mathbf{x}_2||) & \cdots & \phi(||\mathbf{x}_2 - \mathbf{x}_n||) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(||\mathbf{x}_n - \mathbf{x}_1||) & \phi(||\mathbf{x}_n - \mathbf{x}_2||) & \cdots & \phi(||\mathbf{x}_n - \mathbf{x}_n||) \end{bmatrix}, C = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix},$$

$$\text{and } G = \begin{bmatrix} g(\mathbf{x}_1) \\ g(\mathbf{x}_2) \\ \vdots \\ g(\mathbf{x}_N) \end{bmatrix}$$

Thus, A is an $n \times n$ matrix, while C and G are column vectors with n elements.

For many choices of $\phi(\mathbf{x})$, the matrix A can be shown to be invertible [7]; however, in those cases where A is singular, (7) can be regularized with polynomial constraints:

$$s(\mathbf{x}) = \sum_{j=1}^{n} c_j^{RBF} \phi\left(||\mathbf{x} - \mathbf{x}_j||_2\right) + \sum_{l=1}^{M} c_l^p \pi_l(\mathbf{x}), \tag{9}$$

$$\sum_{j=1}^{n} c_j^{RBF} \pi_l(\mathbf{x}_j) = 0, \text{ for } l = 1, ..., M \tag{10}$$

to guarantee nonsingularity [7]. That is, the polynomial terms are orthogonal to the

RBF coefficients. As an example, for $\mathbf{x} \in \mathbb{R}^2$, $M = (m+1)(m+2)/2$ and

$$\begin{aligned}
&\underline{\pi}(\eta,\xi) = \{1\} \text{ for } m = 0 \\
&\underline{\pi}(\eta,\xi) = \{1,\eta,\xi\} \text{ for } m = 1 \\
&\underline{\pi}(\eta,\xi) = \{1,\eta,\xi,\eta^2,\eta\xi,\xi^2\} \text{ for } m = 2 \\
&\underline{\pi}(\eta,\xi) = \{1,\eta,\xi,\eta^2,\eta\xi,\xi^2,\eta^3,\eta^2\xi,\eta\xi^2,\xi^3\} \text{ for } m = 3 \\
&\qquad \vdots \\
&\underline{\pi}(\eta,\xi) = \{1,\eta,\xi,\eta^2,\eta\xi,\xi^2,\cdots,\eta^m,\eta^{m-1}\xi,\cdots,\eta\xi^{m-1},\xi^m\}
\end{aligned}$$

The inclusion of polynomial terms in (9) and the largest degree of the polynomial terms included (m) can have impacts on the accuracy of the approximation (see for example [10]) even when A is nonsingular. Furthermore, M is limited by, for instance, $\frac{(m+1)(m+2)}{2} < n$ since invertibility is no longer possible for $M > n$. As a result of (9) and (10), the linear system now becomes

$$\tilde{A}\tilde{C} = \tilde{G} \tag{11}$$

where

$$\tilde{A} = \begin{bmatrix} A & P \\ P^T & \mathbf{0}_{(M\times M)} \end{bmatrix}, \tilde{C} = \begin{bmatrix} C^{RBF} \\ C^p \end{bmatrix}, \text{ and } \tilde{G} = \begin{bmatrix} G \\ \mathbf{0}_{(M\times 1)} \end{bmatrix}$$

Here, A and F are the same matrix and vector, respectively, from (8), C^{RBF} and C^p are column vectors defined by $C_j^{RBF} = c_j^{RBF}$, $j = 1, ..., N$, and $C_l^p = c_l^p$, $l = 1, ..., M$,

as from (9), while

$$P = \begin{bmatrix} \pi_1(\mathbf{x}_1) & \pi_2(\mathbf{x}_1) & \cdots & \pi_M(\mathbf{x}_1) \\ \pi_1(\mathbf{x}_2) & \pi_2(\mathbf{x}_2) & \cdots & \pi_M(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \pi_1(\mathbf{x}_n) & \pi_2(\mathbf{x}_n) & \cdots & \pi_M(\mathbf{x}_n) \end{bmatrix}$$

Thus, P is an $n \times M$ matrix, $\tilde{A}$ is an $(n+M) \times (n+M)$ matrix, and $\tilde{C}$ and $\tilde{G}$ are column vectors with $(n+M)$ elements.

Referring back to table 1, when using the infinitely smooth basis functions, the interpolation matrix A is positive-definite (nonsingular). Whereas, the monomial RBFs (piecewise smooth functions) applied in this thesis do not always have positive definiteness [7]. As a result, (9) with nonsingular interpolation matrix $\tilde{A}$ is used to interpolate $g(\eta, \xi)$ on the two-dimensional coordinate system (η, ξ) of section 2.2.2.

Notice

$$I_{\Omega_k}(g) \approx I_{\Omega_k}(s) = \sum_{j=1}^{n} c_j^{RBF} I_{\Omega_k}\left(\phi\left(\sqrt{(\eta-\eta_j)^2 + (\xi-\xi_j)^2}\right)\right) + \sum_{l=1}^{\frac{(m+1)(m+2)}{2}} c_l^p I_{\Omega_k}(\pi_l(\eta,\xi)) \tag{12}$$

Define

$$\tilde{I} = \begin{bmatrix} I^{RBF} \\ \\ I^p \end{bmatrix}$$

where $I_j^{RBF} = I_{\Omega_k}\left(\phi\left(\sqrt{(\eta-\eta_j)^2 + (\xi-\xi_j)^2}\right)\right)$, $j = 1, 2, ..., n$, and $I_l^p = I(\pi_l(\eta, \xi))$,

$l = 1, 2, ..., M$. Since $\tilde{A}\tilde{C} = \tilde{G}$ and $\tilde{A}$ is invertible,

$$\begin{aligned} I_{\Omega_k}(g) \approx \tilde{C}^T \tilde{I} &= \left(\tilde{A}^{-1}\tilde{G}\right)^T \tilde{I} \\ &= \tilde{G}^T \left(\left(\tilde{A}^T\right)^{-1} \tilde{I}\right) \\ &= \tilde{G}^T W \end{aligned}$$

That is, $W = \left(\tilde{A}^T\right)^{-1} \tilde{I}$ is the solution to $\tilde{A}^T W = \tilde{I}$. Enumerate the elements of W as

$$W = \begin{bmatrix} w_1^{RBF} & w_2^{RBF} & \cdots w_n^{RBF} & w_1^p & w_2^p & \cdots & w^p_{\frac{(m+1)(m+2)}{2}} \end{bmatrix}^T$$

Note that by equation (11) only the first n elements in $\tilde{G}$ have nonzero values, so

$$I_{\Omega_k}(g) \approx \tilde{G}^T W = \sum_{j=1}^{n} w_j^{RBF} g(\eta_j, \xi_j) = \tilde{I}_{\Omega_k}(g) \tag{13}$$

2.3 Weight calculations

To find the weights $\{w_j^{RBF}\}_{j=1}^n$ for approximating $I_{\Omega_k}(g)$, the linear system of (11) must be constructed. The matrix $\tilde{A}$ is readily populated by evaluating each of the basis functions (RBFs and polynomials) at each of the n interpolation points. The construction of $\tilde{I}_{\Omega_k}(g)$, on the other hand, requires integrating each of the basis functions (preferably analytically) over the planar triangle $\tilde{\tau}_{ABC}$.

2.3.1 Planar triangles.

In order to compute the weights for (13), the values of I_j^{RBF}, $j = 1, 2, ..., n$, and I_l^p, $l = 1, 2, ..., M$, from (12) are required. While integrating polynomial terms in I_l^p is more common, this section discusses a way to obtain the values I_j^{RBF} exactly.

2.3.1.1 Integrating Rotationally Invariant Functions over Right Triangles.

Let O: $\boldsymbol{\omega}_O = (\eta_O, \xi_O)$ be an arbitrary projection point in the (η, ξ) plane defined in section 2.2.2. As adapted from [2], a radial basis function $\phi\left(\sqrt{(\eta-\eta_O)^2+(\xi-\xi_O)^2}\right)$ can be integrated over the projected triangle $\tilde{\tau}_{ABC} \in \Omega_k$ with vertices A: $\boldsymbol{\omega}_A = (\eta_A, \xi_A)$, B: $\boldsymbol{\omega}_B = (\eta_B, \xi_B)$, and C: $\boldsymbol{\omega}_C = (\eta_C, \xi_C)$ (A, B, C being projected nodes) by integrating over a combination of six different right triangles. How these six triangles relate to $\tilde{\tau}_{ABC}$ is different for each individual choice of O. Now consider D: $\boldsymbol{\omega}_D = (\eta_D, \xi_D)$, E: $\boldsymbol{\omega}_E = (\eta_E, \xi_E)$, and F: $\boldsymbol{\omega}_F = (\eta_F, \xi_F)$ to be the orthogonal projection of O onto the lines through sides AB, BC, and AC of $\tilde{\tau}_{ABC}$, respectively (see figure 3). Then it will be shown that the integral $I(\phi)$ over $\tilde{\tau}_{ABC}$ can be written as

$$\begin{aligned} I_{\tilde{\tau}_{ABC}}(\phi) := \iint_{\tilde{\tau}_{ABC}} \phi\left(\sqrt{(\eta-\eta_O)^2+(\xi-\xi_O)^2}\right) d\xi d\eta = s_{OAD} I_{t_{OAD}}(\phi) + s_{ODB} I_{t_{ODB}}(\phi) \\ + s_{OBE} I_{t_{OBE}}(\phi) + s_{OEC} I_{t_{OEC}}(\phi) \\ + s_{OFA} I_{t_{OFA}}(\phi) + s_{OCF} I_{t_{OCF}}(\phi) \end{aligned}$$

That is, it can be computed as a linear combination of the integrals over the right triangles t_{OAD}, t_{ODB}, t_{OBE}, t_{OEC}, t_{OFA}, and t_{OCF} with corresponding weights s_{OAD}, s_{ODB}, s_{OBE}, s_{OEC}, s_{OFA}, and s_{OCF}, respectively, equal to $+1$ or -1.

Consider the integral

$$I_{t_{OAD}}(\phi) = \iint_{t_{OAD}} \phi\left(\sqrt{(\eta-\eta_O)^2+(\xi-\xi_O)^2}\right) d\xi d\eta \tag{14}$$

since the integrals for the other five right triangles are analogous. Since O could be located anywhere in Ω_k, it will generally not be at the origin, and the side DO will

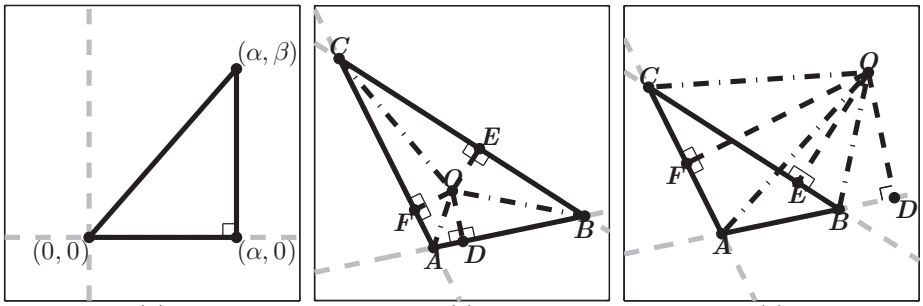


Figure 3. Example divisions of a planar triangle into six right triangles. (a) Final orientation of an individual right triangle used for RBF integration. (b) Example division of $\tilde{\tau}_{ABC}$ where the signs s_{OAD}, s_{ODB}, s_{OBE}, s_{OEC}, s_{OFA}, and s_{OCF} in equation (19) are all positive. (c) Example division where the signs s_{OAD}, s_{OFA}, and s_{OCF} are positive while signs s_{ODB}, s_{OBE}, and s_{OEC} are negative.

not align with the η-coordinate axis. As discussed in [37], consider moving O to the (η, ξ) origin by the change of variables (depcited in figure 4b)

$$\boldsymbol{\omega}' = \boldsymbol{\omega} - \boldsymbol{\omega}_O$$

Next consider the rotation matrix

$$R = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \text{ and } R^{-1} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

where θ is the counterclockwise angle between the positive η-axis and the vector in the direction of the side DO. Thus, define the change of variables

$$\boldsymbol{\omega}'' = R\boldsymbol{\omega}' = R(\boldsymbol{\omega} - \boldsymbol{\omega}_O) = \begin{bmatrix} (\eta - \eta_O)\cos\theta + (\xi - \xi_O)\sin\theta \\ -(\eta - \eta_O)\sin\theta + (\xi - \xi_O)\cos\theta \end{bmatrix}$$

that corresponds to translating $\boldsymbol{\omega}_O$ to $(0, 0)$ and rotating the edge DO to lie on the positive η-axis with the edge AD either above or below, but perpendicular to, the

η-axis (see figure 4c). Note that

$$(\eta'')^2 + (\xi'')^2 = (\eta - \eta_O)^2 + (\xi - \xi_O)^2 \tag{15}$$

and

$$\left| \begin{bmatrix} \frac{\partial \eta''}{\partial \eta} & \frac{\partial \xi''}{\partial \eta} \\ \frac{\partial \eta''}{\partial \xi} & \frac{\partial \xi''}{\partial \xi} \end{bmatrix} \right| = \left| \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \right|$$
$$= \cos^2\theta + \sin^2\theta = 1$$

So, $d\eta d\xi = d\eta'' d\xi''$. Therefore, (14) is equivalently

$$\iint\limits_{t_{OAD}} \phi\left(\sqrt{(\eta - \eta_O)^2 + (\xi - \xi_O)^2}\right) d\xi d\eta = \iint\limits_{\tilde{t}_{OAD}} \phi\left(\sqrt{(\eta'')^2 + (\xi'')^2}\right) d\eta'' d\xi''$$

for $\tilde{t}_{OAD}$ the triangle t_{OAD} translated and rotated. Now with O at the origin and DO lying on the positive η-axis with AD perpendicular to it, $\eta''_D = \eta''_A$ and $\xi''_D = 0$. Letting $\alpha_D = |\eta''_D|$, (15) shows that

$$\alpha_D = |\eta''_D| = \sqrt{(\eta''_D)^2 + (\xi''_D)^2} = \sqrt{(\eta_D - \eta_O)^2 + (\xi_D - \xi_O)^2}$$

is the length of edge DO. Similarly letting $\beta_{OAD} = |\xi''_A|$, from (15),

$$\beta_{OAD} = |\xi''_A| = \sqrt{(\eta_D - \eta_A)^2 + (\xi_A - \xi_D)^2}$$

is the length of edge AD. As a result, the integral over the triangle $\tilde{t}_{OAD}$ is now

$$\iint\limits_{\tilde{t}_{OAD}} \phi\left(\sqrt{(\eta'')^2 + (\xi'')^2}\right) d\xi'' d\eta'' = \int_0^{\alpha_D} \int_0^{\frac{\beta_{OAD}}{\alpha_D}\eta''} \phi\left(\sqrt{(\eta'')^2 + (\xi'')^2}\right) d\xi'' d\eta''$$

for AD above the η-axis (see figure 4d). For AD below the η-axis (figure 4e), let $\eta''' = \eta''$ and $\xi''' = -\xi''$ where

$$\int_0^{\alpha_D} \int_{-\frac{\beta_{OAD}}{\alpha_D}\eta''}^{0} \phi\left(\sqrt{(\eta'')^2 + (\xi'')^2}\right) d\xi'' d\eta'' = \int_0^{\alpha_D} \int_{-\frac{\beta_{OAD}}{\alpha_D}\eta'''}^{0} \phi\left(\sqrt{(\eta''')^2 + (-\xi''')^2}\right) (-1) d\xi''' d\eta'''$$
$$= \int_0^{\alpha_D} \int_0^{\frac{\beta_{OAD}}{\alpha_D}\eta'''} \phi\left(\sqrt{(\eta''')^2 + (\xi''')^2}\right) d\xi''' d\eta'''$$

In any case,

$$\iint_{t_{OAD}} \phi\left(\sqrt{(\eta - \eta_O)^2 + (\xi - \xi_O)^2}\right) d\xi d\eta = \int_0^{\alpha_D} \int_0^{\frac{\beta_{OAD}}{\alpha_D}\eta} \phi(\sqrt{\eta^2 + \xi^2}) d\xi d\eta \qquad (16)$$

Here, α_D is the length of side DO. So, α_E and α_F would be the lengths of sides EO and FO, respectively. Likewise, β_{ODB}, β_{OEC}, β_{OBE}, β_{OFA}, and β_{OCF} could be defined in a similar way as β_{OAD}.

2.3.1.2 Some Closed Form Integrals of RBFs.

Note that many commonly considered RBFs can be integrated exactly over right triangles in $\mathbb{R}^2$. For example, given a right triangle with base α and height β, then

$$\int_0^{\alpha} \int_0^{\frac{\beta}{\alpha}\eta} (\eta^2 + \xi^2)^{\frac{3}{2}} d\xi d\eta = \frac{1}{40}\alpha \left(3\alpha^4 \sinh^{-1}\left(\frac{\beta}{\alpha}\right) + \beta\sqrt{\alpha^2 + \beta^2}(5\alpha^2 + 2\beta^2)\right),$$

$$\int_0^{\alpha} \int_0^{\frac{\beta}{\alpha}\eta} (\eta^2 + \xi^2)^{\frac{5}{2}} d\xi d\eta$$
$$= \frac{1}{336}\alpha \left(15\alpha^6 \sinh^{-1}\left(\frac{\beta}{\alpha}\right) + \beta\sqrt{\alpha^2 + \beta^2}(33\alpha^4 + 26\alpha^2\beta^2 + 8\beta^4)\right),$$

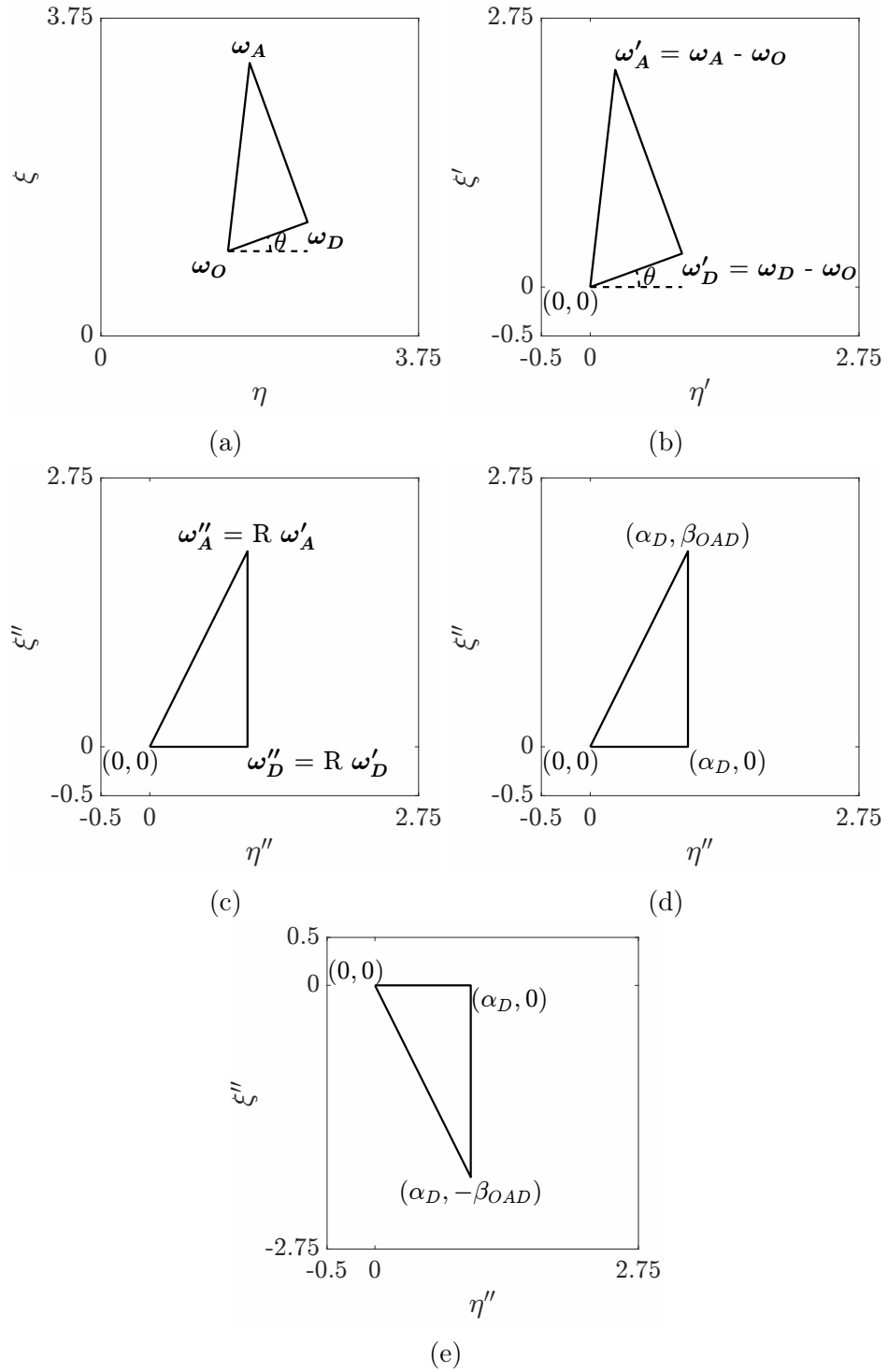


Figure 4. (a)-(d) is the progression of shifting and rotating a right triangle for use in RBF integration as outlined in section 2.3.1.1. Note that (a) has different axes for scaling purposes. In the case of $\boldsymbol{\omega}_D < \boldsymbol{\omega}_O$ in (a), (b)-(d) would look upside-down, resulting in (e).

and

$$\int_0^{\alpha}\int_0^{\frac{\beta}{\alpha}\eta}(\eta^2+\xi^2)^{\frac{7}{2}}d\xi d\eta$$
$$=\frac{\alpha\left(105\alpha^8\sinh^{-1}\left(\frac{\beta}{\alpha}\right)+\beta\sqrt{\alpha^2+\beta^2}(279\alpha^6+326\alpha^4\beta^2+200\alpha^2\beta^4+48\beta^6)\right)}{3456}$$

Each of these are defined as long as $\alpha \neq 0$ (that is, when the vertices of the right triangle are not all three collinear) [2]. Hence, these closed form integrals of monomial RBFs can be used to evaluate such integrals as (16).

2.3.1.3 Integration over Arbitrary Triangles via Integration over Right Triangles.

To integrate radial basis functions over planar triangles, first consider a right triangle t_{XYZ} in the (η, ξ) plane made up of the line segments from X: (η_x, ξ_x) to Y, Y: (η_y, ξ_y) to Z, and Z: (η_x, ξ_z) to X. Suppose that $\phi(\eta, \xi)$ is integrable with respect to both η and ξ and define

$$\phi_1(\eta,\xi)=+\frac{1}{2}\int^{\eta}\phi(\eta,\xi)d\chi \text{ and } \phi_2(\eta,\xi)=-\frac{1}{2}\int^{\xi}\phi(\eta,\xi)d\Psi$$

(this holds for all the radial basis functions used this thesis, as well as multivariate polynomials). Then,

$$\phi(\eta,\xi)=\frac{\partial}{\partial\eta}\phi_1(\eta,\xi)-\frac{\partial}{\partial\xi}\phi_2(\eta,\xi)$$

Define

$$\Phi\left(\begin{bmatrix}\eta\\ \xi\end{bmatrix}\right)=\begin{bmatrix}\phi_1(\eta,\xi)\\ \phi_2(\eta,\xi)\end{bmatrix}$$

so that the line integral of Φ along the line from X to Y can be written as

$$\begin{aligned} LI_{XY}(\Phi) &:= \int_0^1 \Phi\left(\begin{bmatrix}\eta_X\\ \xi_X\end{bmatrix} + t\begin{bmatrix}\eta_Y-\eta_X\\ \xi_Y-\xi_X\end{bmatrix}\right)\cdot\begin{bmatrix}\eta_Y-\eta_X\\ \xi_Y-\xi_X\end{bmatrix} dt \\ &= -\int_0^1 \Phi\left(\begin{bmatrix}\eta_Y\\ \xi_Y\end{bmatrix} + t\begin{bmatrix}\eta_X-\eta_Y\\ \xi_X-\xi_Y\end{bmatrix}\right)\cdot\begin{bmatrix}\eta_X-\eta_Y\\ \xi_X-\xi_Y\end{bmatrix} dt = -LI_{YX}(\Phi) \end{aligned}$$

Since the boundary of t_{XYZ} is a closed and piecewise smooth contour, and so long as X, Y, and Z are oriented counterclockwise in the plane, Green's theorem [29] allows the integral of t_{XYZ} to be written as

$$I_{t_{XYZ}}(\phi) = LI_{XY}(\Phi) + LI_{YZ}(\Phi) + LI_{ZX}(\Phi)$$

Note that if $\phi(\eta,\xi) = 1$, then

$$\text{Area}(t_{XYZ}) = I_{t_{XYZ}}(\phi) = I_{t_{XYZ}}(1) > 0$$

Also note that $\frac{\partial}{\partial\eta}(\frac{1}{2}\eta) - \frac{\partial}{\partial\xi}(-\frac{1}{2}\xi) = 1$. Continuing with a counterclockwise orientation,

$$\begin{aligned} I_{t_{XYZ}}(1) &= LI_{XY}\left(\begin{bmatrix}-\frac{1}{2}\xi\\ \frac{1}{2}\eta\end{bmatrix}\right) + LI_{YZ}\left(\begin{bmatrix}-\frac{1}{2}\xi\\ \frac{1}{2}\eta\end{bmatrix}\right) + LI_{ZX}\left(\begin{bmatrix}-\frac{1}{2}\xi\\ \frac{1}{2}\eta\end{bmatrix}\right) \\ &= \frac{1}{2}\begin{bmatrix}\xi_X-\xi_Y\\ \eta_Y-\eta_X\end{bmatrix}\cdot\begin{bmatrix}\eta_Z-\eta_X\\ \xi_Z-\xi_X\end{bmatrix} \end{aligned}$$

With a clockwise orientation, the direction (and therefore sign) of the path changes:

$$\begin{aligned} I_{t_{XYZ}}(1) &= LI_{XZ}\left(\begin{bmatrix} -\frac{1}{2}\xi \\ \\ \frac{1}{2}\eta \end{bmatrix}\right) + LI_{ZY}\left(\begin{bmatrix} -\frac{1}{2}\xi \\ \\ \frac{1}{2}\eta \end{bmatrix}\right) + LI_{YX}\left(\begin{bmatrix} -\frac{1}{2}\xi \\ \\ \frac{1}{2}\eta \end{bmatrix}\right) \\ &= -\frac{1}{2}\begin{bmatrix} \xi_X - \xi_Y \\ \\ \eta_Y - \eta_X \end{bmatrix} \cdot \begin{bmatrix} \eta_Z - \eta_X \\ \\ \xi_Z - \xi_X \end{bmatrix} \end{aligned}$$

Therefore, let

$$s_{XYZ} := \text{sign}\left(\begin{bmatrix} \xi_X - \xi_Y \\ \\ \eta_Y - \eta_X \end{bmatrix} \cdot \begin{bmatrix} \eta_Z - \eta_X \\ \\ \xi_Z - \xi_X \end{bmatrix}\right) = \pm 1 \tag{17}$$

where $s_{XYZ} = 1$ if the vertices of t_{XYZ} are oriented counterclockwise, and $s_{XYZ} = -1$ if they are oriented clockwise. It is apparent that if the vertices are oriented counterclockwise, then $s_{XYZ} = 1$ and

$$s_{XYZ} I_{t_{XYZ}}(\phi) = I_{t_{XYZ}}(\phi) = LI_{XY}(\Phi) + LI_{YZ}(\Phi) + LI_{ZX}(\Phi)$$

Whereas vertices oriented clockwise yields $s_{XYZ} = -1$ and

$$\begin{aligned} s_{XYZ} I_{t_{XYZ}}(\phi) = (-1) I_{t_{XYZ}}(\phi) &= (-1)(LI_{XZ}(\Phi) + LI_{ZY}(\Phi) + LI_{YX}(\Phi)) \\ &= (-1)(-LI_{ZX}(\Phi) - LI_{YZ}(\Phi) - LI_{XY}(\Phi)) \\ &= LI_{ZX}(\Phi) + LI_{YZ}(\Phi) + LI_{XY}(\Phi) \end{aligned}$$

In either case,

$$s_{XYZ} I_{t_{XYZ}}(\phi) = LI_{XY}(\Phi) + LI_{YZ}(\Phi) + LI_{ZX}(\Phi) \tag{18}$$

Recall that the task at hand is to show that the integral over an arbitrary right triangle can be evaluated by computing integrals over six right triangles with common vertex O. Parts of these right triangles may reside within $\tilde{\tau}_{ABC}$, while other parts may reside outside. A combination of the six right triangles should create the area for the ABC triangle. By adding or subtracting the integrals of the six right triangles in question, it is claimed that the integral value $I_{\tilde{\tau}_{ABC}}(\phi)$ over $\tilde{\tau}_{ABC}$ can be found by evaluating

$$\begin{aligned} I_{\tilde{\tau}_{ABC}}(\phi) =& s_{ABC}(s_{OAD}I_{t_{OAD}}(\phi) + s_{ODB}I_{t_{ODB}}(\phi) + s_{OBE}I_{t_{OBE}}(\phi) + s_{OEC}I_{t_{OEC}}(\phi) + \\ & s_{OCF}I_{t_{OCF}}(\phi) + s_{OFA}I_{t_{OFA}}(\phi)) \end{aligned} \tag{19}$$

for signs s_{OAD}, s_{ODB}, s_{OBE}, s_{OEC}, s_{OFA}, and s_{OCF} defined as in (17) by replacing XYZ and preserving vertex order. Depending on the orientation of the six right triangles and $\tilde{\tau}_{ABC}$, the signs may all be 1, or only some may be equal to 1. As seen in figure 3, (b) depicts a $\tilde{\tau}_{ABC}$ such that all the signs are positive, while (c) depicts a $\tilde{\tau}_{ABC}$ such that the signs s_{OAD}, s_{OFA}, and s_{OCF} are positive while signs s_{ODB}, s_{OBE}, and s_{OEC} are negative. Regardless, the orientation of the vertices A, B, C, D, E, F, and O is the deciding factor on delegating the values for the signs.

Recall that Green's theorem [29] is based on a boundary (path) integral and that $LI_{OA} = -LI_{AO}$. With all this information, the following works backwards to achieve

(19) [38]. Similar to figure 3(c),

$$
\begin{aligned}
s_{ABC}(s_{OAD}I_{t_{OAD}}(\phi) + s_{ODB}I_{t_{ODB}}(\phi) + s_{OBE}I_{t_{OBE}}(\phi) + \\
s_{OEC}I_{t_{OEC}}(\phi) + s_{OCF}I_{t_{OCF}}(\phi) + s_{OFA}I_{t_{OFA}}(\phi)) \\
= s_{ABC}(LI_{OA}(\Phi) + LI_{AD}(\Phi) + LI_{DO}(\Phi) + \\
LI_{OD}(\Phi) + LI_{DB}(\Phi) + LI_{BO}(\Phi) + \\
LI_{OB}(\Phi) + LI_{BE}(\Phi) + LI_{EO}(\Phi) + \\
LI_{OE}(\Phi) + LI_{EC}(\Phi) + LI_{CO}(\Phi) + \\
LI_{OC}(\Phi) + LI_{CF}(\Phi) + LI_{FO}(\Phi) + \\
LI_{OF}(\Phi) + LI_{FA}(\Phi) + LI_{AO}(\Phi)) \\
= s_{ABC}(LI_{AD}(\Phi) + LI_{DB}(\Phi) + LI_{BE}(\Phi) + \\
LI_{EC}(\Phi) + LI_{CF}(\Phi) + LI_{FA}(\Phi)) \\
= s_{ABC}(LI_{AB}(\Phi) + LI_{BC}(\Phi) + LI_{CA}(\Phi)) = I_{\tilde{\tau}_{ABC}}(\phi)
\end{aligned}
$$

Now that the integrals of RBFs for individual triangles can be evaluated exactly, the following puts all the previous Chapter II information together to yield the desired quadrature weights for $\mathcal{I}_S(f)$.

2.3.2 Converting Weights from Planar to Spherical.

Recall from sections 2.2.1 and 2.2.2 that the nearest neighboring nodes for each triangle $\tau_k \in \mathcal{T}$ (that is, the points in $\mathcal{N}_n$) are projected onto the (η, ξ) plane, and then a transformation allows the projected nodes to be treated as though they exist in two-dimensional space. With intermediate quadrature weights from (13) in hand, the task of relating the weights computed for each tangent plane to the ones required for computing surface integrals on the sphere remains.

Considering the change of variables (3) and (4) from section 2.2.2, inverse transformations mapping points in Ω_k back to S_{Ω_k} can be realized by

$$\begin{aligned} x &= \frac{\eta\rho}{\sqrt{\rho^2+\eta^2+\xi^2}} \\ y &= \frac{\xi\rho}{\sqrt{\rho^2+\eta^2+\xi^2}} \\ z &= \frac{\rho^2}{\sqrt{\rho^2+\eta^2+\xi^2}} \end{aligned} \tag{20}$$

in the case of $\tilde{x}_M^2 + \tilde{y}_M^2 = 0$. Similarly, when the triangle midpoint $\tilde{\mathbf{x}}_M$ is not located on the z-axis ($\tilde{x}_M^2 + \tilde{y}_M^2 \neq 0$), the inverse transformation is

$$\begin{aligned} x &= \frac{\tilde{x}_M\left(\eta\tilde{z}_M + \rho\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}\right) - \rho\xi\tilde{y}_M}{\sqrt{\rho^2+\eta^2+\xi^2}\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}} \\ y &= \frac{\tilde{y}_M\left(\eta\tilde{z}_M + \rho\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}\right) + \rho\xi\tilde{x}_M}{\sqrt{\rho^2+\eta^2+\xi^2}\sqrt{\tilde{x}_M^2+\tilde{y}_M^2}} \\ z &= \frac{-\eta\sqrt{\tilde{x}_M^2+\tilde{y}_M^2} + \rho\tilde{z}_M}{\sqrt{\rho^2+\eta^2+\xi^2}} \end{aligned} \tag{21}$$

That is, $\tau_k \in \mathcal{T}$ is parameterized locally by the parameters η and ξ. Given a surface parameterization $x(\eta,\xi), y(\eta,\xi), z(\eta,\xi)$, it is true by definition [39] that

$$\iint\limits_{\tau_k} f(x,y,z)dS = \iint\limits_{\tilde{\tau}_k} f(x(\eta,\xi),y(\eta,\xi),z(\eta,\xi)) \left|\left| \frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi) \times \frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi) \right|\right|_2 d\eta d\xi$$

In either case of (20) or (21), and after much simplification,

$$\left|\left| \frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi) \times \frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi) \right|\right|_2 = \frac{\rho^3}{(\rho^2+\eta^2+\xi^2)^{\frac{3}{2}}}$$

So,

$$\iint\limits_{\tau_k} f(x,y,z)dS = \iint\limits_{\tilde{\tau}_k} f(x(\eta,\xi),y(\eta,\xi),z(\eta,\xi))\frac{\rho^3}{(\rho^2+\eta^2+\xi^2)^{\frac{3}{2}}}d\eta d\xi$$

Defining

$$g(\eta,\xi) = f(x(\eta,\xi),y(\eta,\xi),z(\eta,\xi))\frac{\rho^3}{(\rho^2+\eta^2+\xi^2)^{\frac{3}{2}}},$$

the integral over τ_k can be approximated as discussed in the preceding sections to give

$$\iint\limits_{\tau_k} f(x,y,z)dS \approx \sum_{j=1}^{n} w_j^{RBF} f(x(\eta,\xi),y(\eta,\xi),z(\eta,\xi))\frac{\rho^3}{(\rho^2+\eta^2+\xi^2)^{\frac{3}{2}}} \tag{22}$$

Notice that the parameterization for each triangle τ_k is different, so the respective parameterization will now be indexed as η_k and ξ_k. Considering the surface integral over all of the sphere S,

$$\begin{aligned}\mathcal{I}_S(f) &= \sum_{k=1}^{K}\iint\limits_{T_k} f(\eta_k,\xi_k)\frac{\rho^3}{(\rho^2+\eta_k^2+\xi_k^2)^{\frac{3}{2}}}d\eta_k d\xi_k \\ &\approx \sum_{k=1}^{K}\sum_{j=1}^{n}(w_k^{RBF})_j f((\eta_k)_j,(\xi_k)_j)\frac{\rho}{(\rho^2+(\eta_k)_j^2+(\xi_k)_j^2)^{\frac{3}{2}}}\end{aligned}$$

Here, $\{(w_k^{RBF})_j\}_{j=1}^n$ is the set of weights in (22) for triangle τ_k and $\{(\eta_k)_j\}_{j=1}^n$, $\{(\xi_k)_j\}_{j=1}^n$ represent the n nearest neighbors in $\mathcal{S}_N$ to $\tilde{\mathbf{x}}_M$ after the parameterization is applied.

Let $\mathcal{K}_i$, $i = 1,2,...,N$, be the set of all pairs (k,j) such that $((\eta_k)_j,(\xi_k)_j) \mapsto (x_i,y_i,z_i)$. That is, each node (x_i,y_i,z_i) maps to at least one respective (η_k,ξ_k) plane. Indeed, this easily occurs more than once as each triangle τ_k projects n nearest neighbors. The larger n is, the more times each node will be projected into separate tangent planes. So each node (x_i,y_i,z_i) has quadrature weights from a number of

triangles, and the sum of those weights from all the planar regions it was projected into is called W_i. Then the integral $\mathcal{I}_S(f)$ can be written as

$$\begin{aligned}\mathcal{I}_S(f) &\approx \sum_{i=1}^{N} \left(\sum_{(k,j)\in\mathcal{K}_i} (w_k^{RBF})_j \frac{\rho^3}{(\rho^2 + (\eta_k)_j^2 + (\xi_k)_j^2)^{\frac{3}{2}}} \right) f(x_i, y_i, z_i) \\ &= \sum_{i=1}^{N} W_i f(x_i, y_i, z_i) = \tilde{\mathcal{I}}_S(f) \end{aligned} \tag{23}$$

As with previous parts, the above calculations are referenced from [2]. Equation (23) is the cumulation of the works of Chapter II, and is the numerical approximation of the integral (i.e. quadrature) of some function $f(x, y, z)$ over the surface of a sphere.

The accuracy of (23) will depend on the accuracy of the RBF interpolation scheme employed for each planar region Ω_k [2]. Over scattered data sets, the accuracy is often reported relative to the fill distance [40] (a.k.a. mesh-size or mesh norm [25]). The fill distance is defined based on the situation. In some cases, the mesh size is the (largest) distance between any two points. In the case of the sphere used in this chapter, the fill distance can be described as the diameter (in arc-length) of the largest empty spherical cap between two points on the sphere surface. Whereas on the tangent planes, the fill distance is the diameter of the largest empty planar circle in the convex hull of the point set. In other words,

$$h_{\mathcal{N},\Omega_k} := 2\text{sup}_{x\in\Omega_k}\text{dist}(x, \mathcal{N})$$

where $\mathcal{N}$ is the set of points in a bounded domain Ω_k and $h_{\mathcal{N},\Omega_k}$ is the diameter (not radius) of the largest empty circle between any two points in Ω_k (adapted from [2, 25]).

When projecting the points on the sphere to an (η_k, ξ_k) plane, the sphere fill distance may become distorted as per (20) and (3). As this affects the accuracy of

the approximation integral from (23), verifying that the distortion will not decrease the accuracy significantly for a small change in sphere fill distance is important [2].

2.3.3 Some Test Cases Over a Sphere Used in [2].

A total of six test cases are presented in [2], of which two in particular are re-displayed here for the purpose of comparison in section 4.2. For the unit sphere $\mathbb{S}^2$, these are

$$\tilde{f}_1(x,y,z) = 1 + x + y^2 + x^2y + x^4 + y^5 + x^2y^2z^2$$
$$\tilde{f}_2(x,y,z) = \frac{1 + \text{sign}(-9x - 9y + 9z)}{9}$$

whose exact surface integrals are

$$\mathcal{I}_{\mathbb{S}^2}(\tilde{f}_1) = \frac{216\pi}{35}$$
$$\mathcal{I}_{\mathbb{S}^2}(\tilde{f}_2) = \frac{4\pi}{9}$$

Consider the application of the method discussed in this chapter with $\phi(r) = r^7$, $m = 7$ (degree of polynomial), and $n = 80$ (nearest neighboring nodes). Note that $\tilde{f}_1(x,y,z) \in C^\infty(\mathbb{S}^2)$ with convergence rates from the RBF quadrature method displayed in figure 5. Meanwhile, $\tilde{f}_2(x,y,z)$ is a discontinuous yet bounded function whose convergence rates from the RBF quadrature method are displayed in figure 6. The relative errors reported in figures 5 and 6 are the maximum error over 1000 random rotations of each node set. The minimum energy (H) nodes were generated by [2] using the methods from [32], which appear in [25]. The minimum energy (W) nodes were generated from [5]. It was shown in [2] that various test integrands in

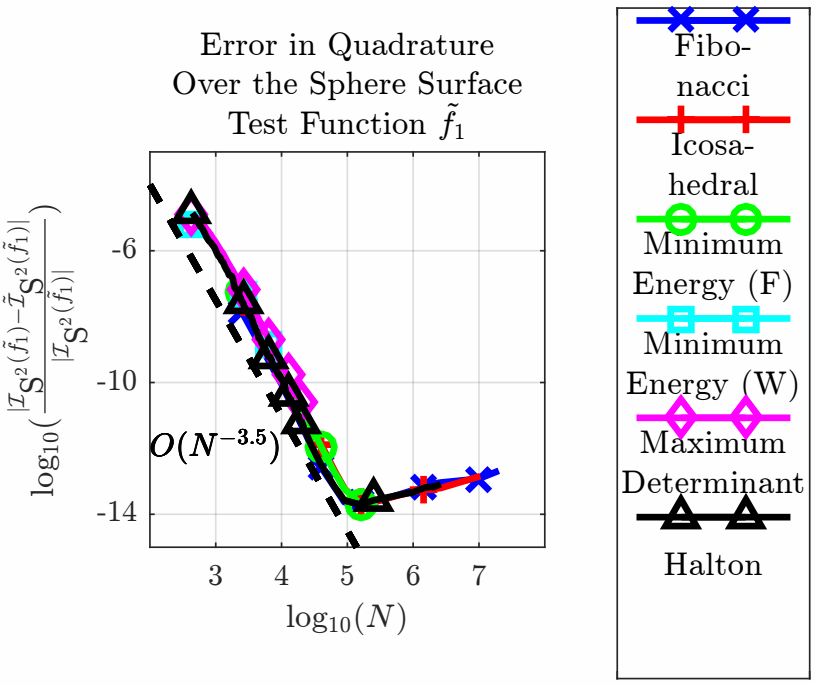


Figure 5. Relative error in RBF quadrature for test function $\tilde{f}_1(x, y, z)$ over $\mathbb{R}^2$ as applied in [2].

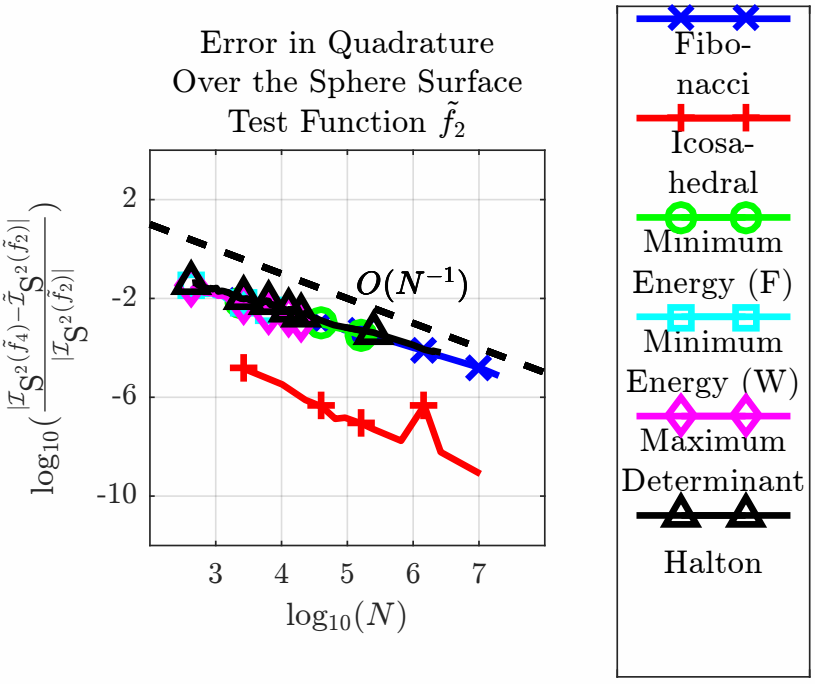


Figure 6. Relative error in RBF quadrature for test function $\tilde{f}_2(x, y, z)$ over $\mathbb{R}^2$ as applied in [2].

$C^\infty(\mathbb{S}^2)$ resulted in a convergence rate of $O(N^{-3.5})$, as illustrated in figure 5. Since $\tilde{f}_2(x, y, z)$ is a discontinuous function, its expected convergence rate should be worse than that of a smooth function.

It was noted in section 2.2.3 that as the degree of interpolating polynomial m increases, then the accuracy of the interpolation should conceptually increase as well. An application of varying m is depicted in figure 7 verifying this concept. However, just as the accuracy increases with increasing m and n, so too does the CPU time for computing the quadrature weights increase. As a result, balancing between accuracy and computation cost is dependent on the application.

For fixed $m = 7$, $n = 80$, and $\phi(r) = r^7$, the total cost in computing the RBF quadrature weights over $\mathbb{S}^2$ is illustrated in figure 8 [41]. Not only is the spherical quadrature method in [2] accurate at $O(N^{-3.5})$ (for smooth integrands f), the method is computationally inexpensive at $O(N\log N)$ operations and $O(N)$ memory usage. When expanding the computation of weights beyond one core, parallel scalability with number of cores was also observed (see figure 8c).

With the conclusion of Chapter II, the following chapter adapts the numerical quadrature method for a sphere surface to a quadrature method for a smooth closed surface. Because much of the methodology remains the same in approximating a smooth versus sphere surface, a separate chapter was dedicated to spherical quadrature as a build-up to the more recent research on surface quadrature.

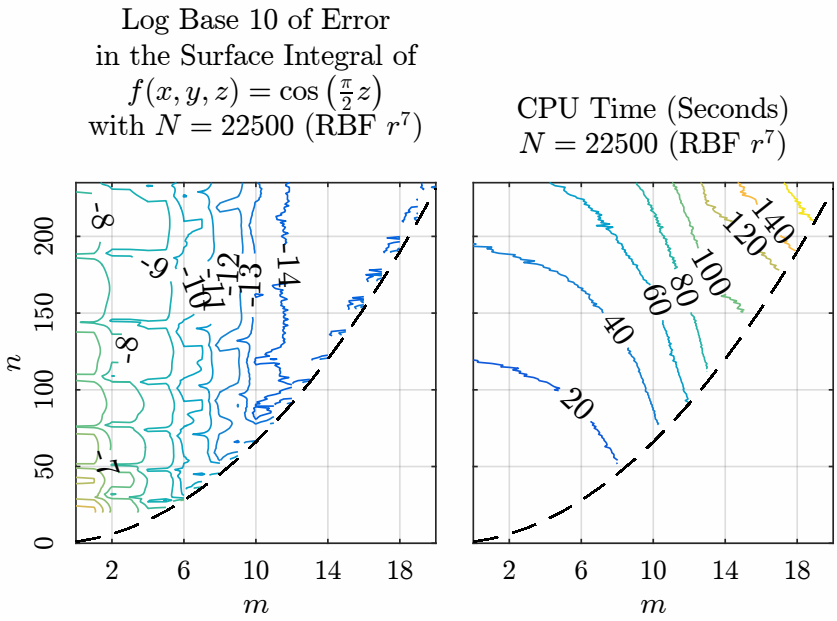
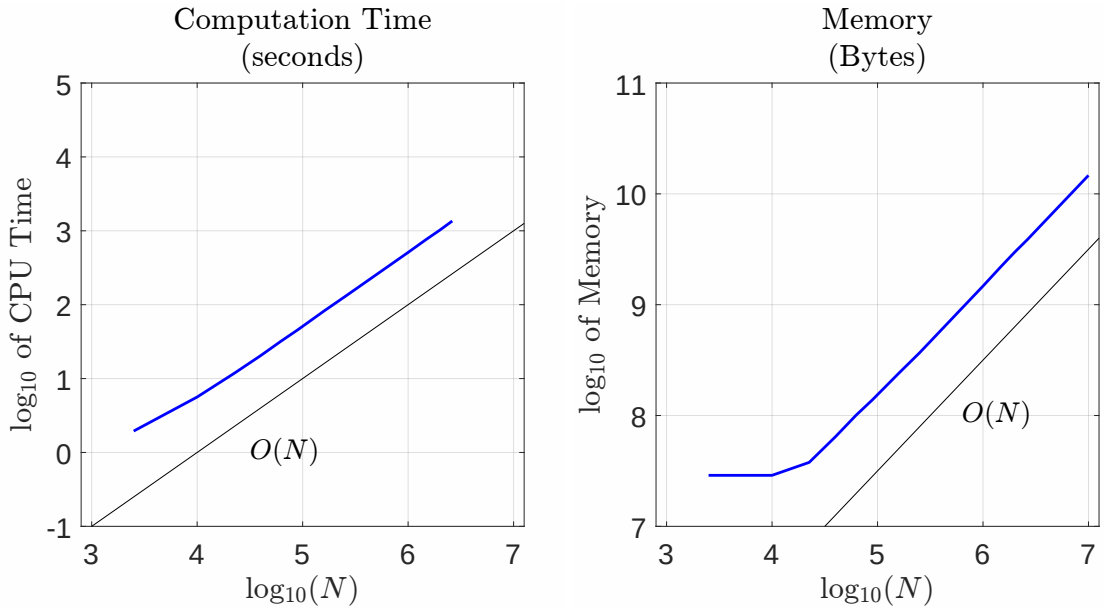
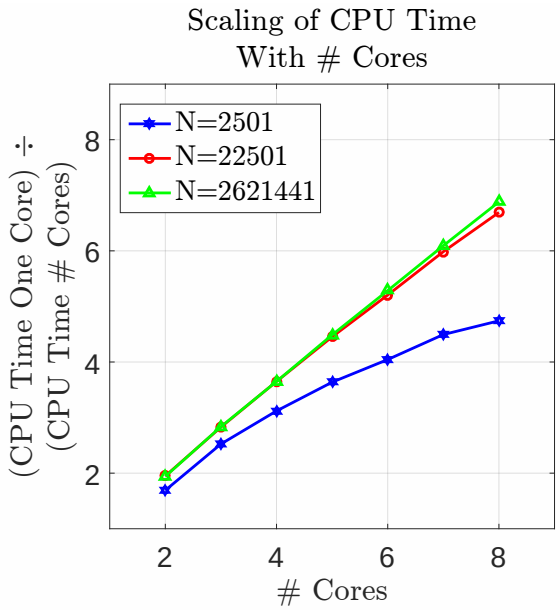


Figure 7. (Adapted from [2]) For some example function $f(x,y,z) = \cos(\frac{\pi}{2}z)$ with a given number of nodes and radial basis function $\phi(r) = r^7$, (left) indicates the quadrature error with respect to the number of nearest neighbors n and degree of polynomial m used in equation (9), (right) indicates the CPU time (in seconds) for computing the quadrature weights with respect to n and m. The black dashed line indicates the constraint $\frac{(m+1)(m+2)}{2} < n$, such that the interpolation problem becomes singular when the constraint is not met. These plots were generated in machines with dual Intel Xeon E5-2687W 3.1 GHz, 8-core processors.



(a) (b)



(c)

Figure 8. (Adapted from [2]) Timing results for computation of the RBF quadrature weights for the surface area of $\mathbb{R}^2$ as applied in [41]. The spherical quadrature method has rate $O(N)$ for computation cost and memory usage for at least millions of nodes, while a cost of $O(N\log N)$ operations is expected for when the Delaunay triangulation and nearest neighbor search dominate the computation [41]. Parallel scalability with number of cores was also observed. These plots were generated in machines with dual Intel Xeon E5-2687W 3.1 GHz, 8-core processors.

III. Quadrature for Smooth Closed Surfaces

As the previous chapter focused on approximating the integral of some scalar function $f(x, y, z)$ over the surface of a sphere, this chapter too approximates the integral of some scalar function $f(x, y, z)$, but over a generalized closed smooth surface. The surface S can either be predefined implicitly by $h(x, y, z) = 0$, predefined explicitly by a parameterization of coordinates $x(u, v)$, $y(u, v)$, and $z(u, v)$, or simply be expressed through a set $\mathcal{S}_N$ of N points in $\mathbb{R}^3$. However, if the surface is not predefined by an equation and is only described by a set of quadrature nodes $\mathcal{S}_N$, then the surface shape appears as a clustering of N points. This being the case, the remainder of this paper assumes that in the case of only $\mathcal{S}_N$ being given (no surface equations), then a triangulation $\mathcal{T}$ with triangles $\{\tau_k\}_{k=1}^{K}$ is also given whereby the nodes $\mathcal{S}_N$ make up the vertices of the triangles - just as in Chapter II.

A process for discretizing a surface described by $h(x, y, z) = 0$ using N nodes and an associated triangulation is described in the section 3.1. While a surface described explicitly by a parameterization of coordinates $x(u, v)$, $y(u, v)$, and $z(u, v)$ can be expressed by discrete nodes and a triangulation, this thesis does not go into details about how this is done. This is a topic left for future consideration.

Just as with Chapter II, this chapter seeks to approximate the integral of $f(x, y, z)$, but over some smooth closed surface S as in (1):

$$\mathcal{I}_S(f) := \iint_S f(x, y, z) dS \approx \sum_{i=1}^{N} W_i f(x_i, y_i, z_i)$$

with quadrature weights $\mathcal{W}_N$: $\{W_i\}_{i=1}^{N}$ for N nodes. The organization of this chapter is similar to that of the previous chapter, in that it will walk through the steps for determining the quadrature weights in (1). As various steps are similar, many references to Chapter II will be made while the differences are expanded upon. The

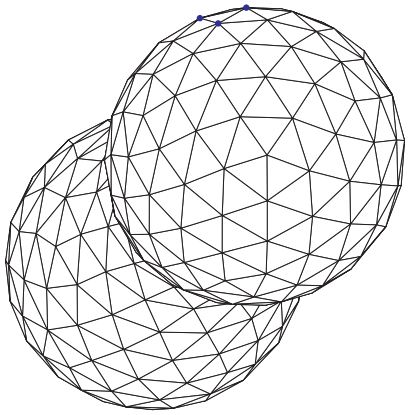


Figure 9. Example surface of revolution of a Cassini oval which has been discretized using **distmeshsurface**. The quadrature method of Chapter III focuses on approximating the integral over each individual triangle.

material for this chapter was derived from [42].

3.1 Triangulation

Consider a smooth closed surface S defined by $h(x, y, z) = 0$. While the surface is defined, the quadrature method for this chapter requires quadrature nodes $\{(x_i, y_i, z_i)\}_{i=1}^{N} \in \mathcal{S}_N$ and corresponding triangulation $\mathcal{T}$. Suppose, however, an application requires both $\mathcal{S}_N$ and $\mathcal{T}$ to be composed (only given $h(x, y, z) = 0$). This section focuses on a MATLAB program called **distmeshsurface** by Persson and Strang [43], which does just that.

Distmeshsurface takes a surface function $h(x, y, z) = 0$ and outputs equilibrium nodes based on its algorithm, as well as the necessary triangulation $\mathcal{T}$. Recall the earlier description for fill distance in section 2.3.2. The mesh size for this algorithm can be described as the closest distance between any two points (Euclidean two norm). While the full algorithm can be found at [43], a summary of its mechanism can be described by the following:

1. Fill a user-defined domain with points separated by a given mesh size, and connect the uniformly distributed points with edges to create a tessellation.

2. Points whose edges are completely outside the surface S (edges not crossing through S) are discarded, and the new grid is re-triangulated.

3. Following the re-triangulation, the points are moved around (while ensuring they remain approximately on S) to make the edge lengths as uniform as possible through a repelling force dependent on the current edge lengths.

4. Steps 2 and 3 repeat until the edge lengths are uniform within a set tolerance.

The algorithm then outputs quadrature nodes $\mathcal{S}_N \in \mathbb{R}^3$ and triangulation $\mathcal{T}$, as in figure 9.

3.2 Locating Projection Origin(s)

Now with quadrature nodes $\mathcal{S}_N$ and triangulation $\mathcal{T}$, the next step in this thesis' quadrature method is to reduce the problem dimensionality from three to two. As in Chapter II, the integral over each surface triangle is considered separately. However, the flat triangles in $\mathcal{T}$ are only approximations to the surface, not actually part of the surface whose edges are geodesics. As a result, the projection and change of variables for each triangle $\{\tau_k\}_{k=1}^K \in \mathcal{T}$ must be made so that the sum of the projected areas do not exceed the surface area of S. This is done by first locating a projection "origin" $\mathbf{x}_O$, or point through which a portion S_{Ω_k} of the surface is projected onto the plane through the triangles vertices (see figure 12).

To start, consider the three edges of triangle $\tau_{ABC} \in \mathcal{T}$: AB, AC, and BC with vertices $\mathbf{x}_A$: (x_A, y_A, z_A), $\mathbf{x}_B$: (x_B, y_B, z_B), and $\mathbf{x}_C$: (x_C, y_C, z_C). Now consider an adjacent triangle τ_{ABE} with third vertex $\mathbf{x}_E$: (x_E, y_E, z_E) (see figure 10). Triangles τ_{ABC} and τ_{ABE} share the edge AB. Also note that the planes containing these triangles each have their own respective normals $\mathbf{n}_{ABC}$ and $\mathbf{n}_{ABE}$ of τ_{ABC} and τ_{ABE}, respectively.

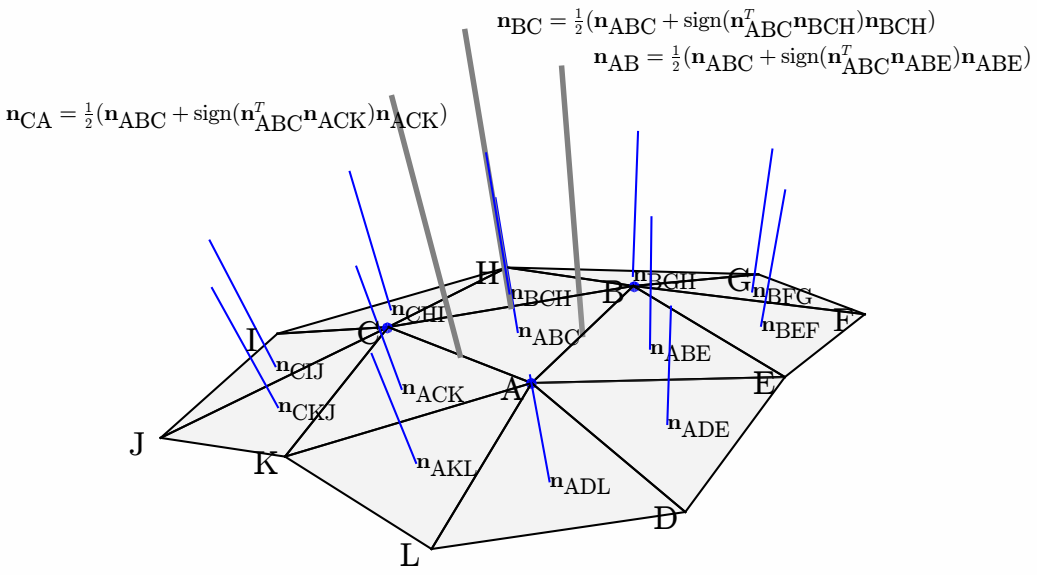


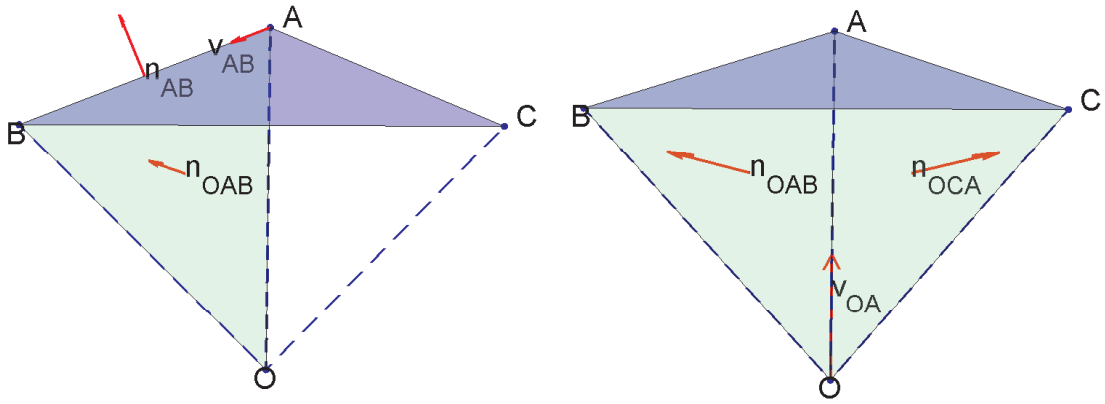
Figure 10. Taking a region S_{Ω_k} about τ_{ABC}, this illustrates the computation of the edge normals for edges AB, CA, and BC.

Let

$$\mathbf{n}_{AB} = \frac{1}{2}(\mathbf{n}_{ABC} + \mathrm{sign}(\mathbf{n}_{ABC}^T \mathbf{n}_{ABE})\mathbf{n}_{ABE})$$

describe the average of the two normals that lies on the edge AB. The same can be done similarly to define $\mathbf{n}_{AC}$ and $\mathbf{n}_{BC}$, as in figure 10.

Define the "cutting plane" through the edge AB to be the plane containing the edge AB such that the plane is parallel to $\mathbf{n}_{AB}$. Consider the cutting planes passing through the edges AB and AC of triangle τ_{ABC}. Also consider the vectors $\mathbf{v}_{AB}$ (which points from $\mathbf{x}_A$ to $\mathbf{x}_B$) and $\mathbf{v}_{CA}$ (which points from $\mathbf{x}_C$ to $\mathbf{x}_A$). Note that the vectors $\mathbf{n}_{AB}$ and $\mathbf{v}_{AB}$ are both parallel to the cutting plane through the edge AB. Likewise,



(a) (b)

Figure 11. (a) Taking the cross product of $\mathbf{n}_{AB}$ and $\mathbf{v}_{AB}$ yields the perpendicular vector $\mathbf{n}_{OAB}$. (b) Taking the cross product of $\mathbf{n}_{OAB}$ and $\mathbf{n}_{OCA}$ yields the perpendicular vector $\mathbf{v}_{OA}$.

the vectors $\mathbf{n}_{CA}$ and $\mathbf{v}_{CA}$ are both parallel to the cutting plane through the edge AC. In order to mathematically define the cutting plane through the edge AB, a normal vector to the plane is required. From the vector cross product,

$$\mathbf{n}_{OAB} = \mathbf{n}_{AB} \times \mathbf{v}_{AB}.$$

where the subscript O indicates that the projection origin $\mathbf{x}_O$ also falls in the AB cutting plane (see figure 11a). Similarly,

$$\mathbf{n}_{OCA} = \mathbf{n}_{CA} \times \mathbf{v}_{CA}$$

$$\mathbf{n}_{OBC} = \mathbf{n}_{BC} \times \mathbf{v}_{BC}$$

From $\mathbf{n}_{OAB}$ and $\mathbf{n}_{OCA}$, a vector $\mathbf{v}_{OA}$ (pointing in the same direction as the vector from $\mathbf{x}_O$ to $\mathbf{x}_A$) can be computed via

$$\mathbf{v}_{OA} = \mathbf{n}_{OAB} \times \mathbf{n}_{OCA}$$

as shown in figure 11b. The line in the direction of $\mathbf{v}_{OA}$ and passing through $\mathbf{x}_A$ can be parameterized by

$$\mathbf{x}_O = \mathbf{x}_A + t\mathbf{v}_{OA} \tag{24}$$

such that $\mathbf{x}_O$ is the point along this line that intersects the cutting plane for the edge BC. This cutting plane can be represented by

$$\mathbf{n}_{OBC} \cdot (\mathbf{x} - \mathbf{x}_B) = 0 \tag{25}$$

where $\cdot$ is the vector dot product. Solving for t from (24) and (25) yields

$$t = \frac{\mathbf{n}_{OBC} \cdot (\mathbf{x}_B - \mathbf{x}_A)}{\mathbf{n}_{OBC} \cdot \mathbf{v}_{OA}}$$

Hence, the projection origin used for triangle τ_{ABC} is

$$\mathbf{x}_O = \mathbf{x}_A + \frac{\mathbf{n}_{OBC} \cdot (\mathbf{x}_B - \mathbf{x}_A)}{\mathbf{n}_{OBC} \cdot \mathbf{v}_{OA}}\mathbf{v}_{OA} \tag{26}$$

While the above can certainly be used as the projection origin for projecting the triangle area into a plane, notice that the node $\mathbf{x}_C$ was not included. If a more symmetric version of the projection origin (with regard to all three vertices of τ_{ABC}) is preferred, it can be written as

$$\mathbf{x}_O = \mathbf{x}_M + \frac{1}{3V}\left[(\mathbf{n}_{OBC} \cdot (\mathbf{x}_B - \mathbf{x}_A))\mathbf{v}_{OA} + (\mathbf{n}_{OCA} \cdot (\mathbf{x}_C - \mathbf{x}_B))\mathbf{v}_{OB} + (\mathbf{n}_{OAB} \cdot (\mathbf{x}_A - \mathbf{x}_C))\mathbf{v}_{OC}\right] \tag{27}$$

where

$$
\begin{aligned}
V &= ((\mathbf{x}_C - \mathbf{x}_B) \times \mathbf{n}_{BC}) \cdot [((\mathbf{x}_B - \mathbf{x}_A) \times \mathbf{n}_{AB}) \times ((\mathbf{x}_A - \mathbf{x}_C) \times \mathbf{n}_{CA})] \\
&= ((\mathbf{x}_A - \mathbf{x}_C) \times \mathbf{n}_{CA}) \cdot [((\mathbf{x}_C - \mathbf{x}_B) \times \mathbf{n}_{BC}) \times ((\mathbf{x}_B - \mathbf{x}_A) \times \mathbf{n}_{AB})] \\
&= ((\mathbf{x}_B - \mathbf{x}_A) \times \mathbf{n}_{AB}) \cdot [((\mathbf{x}_A - \mathbf{x}_C) \times \mathbf{n}_{CA}) \times ((\mathbf{x}_C - \mathbf{x}_B) \times \mathbf{n}_{BC})]
\end{aligned}
$$

3.3 Projecting the Nearest Neighboring Nodes

Just as with section 2.2.1, the n nearest nodes $\mathcal{N}_n$: $\{\mathbf{x}_j\}_{j=1}^n \in \mathcal{S}_N$ to each triangle are needed for numerical approximation via RBF interpolation. When projecting these neighbors, care must be taken to avoid projecting points for which the vector from the projection origin $\mathbf{x}_O$ to the point being projected ($\mathbf{x} \in \mathcal{N}_n$) is not an angle more than 90° from the normal $\mathbf{n}_{ABC}$. Similar to section 2.2.1, points at this angle result in singularities in the following mathematics. Therefore, points having these conditions are not utilized for projection. That is, consider the projection of a point $\mathbf{x} \in S_{\Omega_k}$ to the plane

$$\mathbf{n}_{ABC} \cdot (\mathbf{x} - \mathbf{x}_M) = 0$$

containing triangle τ_{ABC}, for $\mathbf{x}_M$ the midpoint of τ_{ABC}. The projection illustrated in figure 12 can be performed by finding the intersection of the line

$$\tilde{\mathbf{x}} = \mathbf{x}_O + t(\mathbf{x} - \mathbf{x}_O)$$

where $\mathbf{x}_O$ is taken from (27) and $\tilde{\mathbf{x}}$ is the projection of $\mathbf{x} \in S_n$ onto the τ_{ABC} plane. Plugging the $\tilde{\mathbf{x}}$ equation into the equation for the plane yields

$$t = \frac{\mathbf{n}_{ABC} \cdot (\mathbf{x}_M - \mathbf{x}_O)}{\mathbf{n}_{ABC} \cdot (\mathbf{x} - \mathbf{x}_O)}$$

so that

$$\begin{aligned} \tilde{\mathbf{x}} &= \mathbf{x}_O + \frac{\mathbf{n}_{ABC} \cdot (\mathbf{x}_M - \mathbf{x}_O)}{\mathbf{n}_{ABC} \cdot (\mathbf{x} - \mathbf{x}_O)} (\mathbf{x} - \mathbf{x}_O) \\ &= \mathbf{x}_O + \frac{n^{(x)}_{ABC}(x_M - x_O) + n^{(y)}_{ABC}(y_M - y_O) + n^{(z)}_{ABC}(z_M - z_O)}{n^{(x)}_{ABC}(x - x_O) + n^{(y)}_{ABC}(y - y_O) + n^{(z)}_{ABC}(z - z_O)} (\mathbf{x} - \mathbf{x}_O) \end{aligned} \tag{28}$$

where $\mathbf{n}_{ABC}$: $(n^{(x)}_{ABC}, n^{(y)}_{ABC}, n^{(z)}_{ABC})$, $\mathbf{x}_M$: (x_M, y_M, z_M), $\mathbf{x}$: (x, y, z), and $\mathbf{x}_O$: (x_O, y_O, z_O). Another example of this process is illustrated by the points in figure 13a being projected as in figure 13b. Note that (28) differs from the spherical case in section 2.2.1 only in that $\mathbf{n}_{ABC}$ is no longer the vector $\tilde{\mathbf{x}}_M$ through the origin, while $\mathbf{x}_O$ is no longer at the origin. Following the same procedure as with Chapter II, the projected nodes $\{\tilde{\mathbf{x}}_j\}_{j=1}^n$ will then be used to define a two-dimensional coordinate system on the τ_{ABC} plane.

3.4 Defining a Two-Dimensional Coordinate System

Recall from section 2.3.2 that in order to simplify the transformation from three-dimensional coordinates to two-dimensional, points in the three-dimensional coordinate system were multiplied by a rotation matrix

$$M = \begin{bmatrix} \frac{\tilde{x}_M \tilde{z}_M}{\rho\gamma} & \frac{\tilde{y}_M \tilde{z}_M}{\rho\gamma} & -\frac{\gamma}{\rho} \\ -\frac{\tilde{y}_M}{\gamma} & \frac{\tilde{x}_M}{\gamma} & 0 \\ \frac{\tilde{x}_M}{\rho} & \frac{\tilde{y}_M}{\rho} & \frac{\tilde{z}_M}{\rho} \end{bmatrix}$$

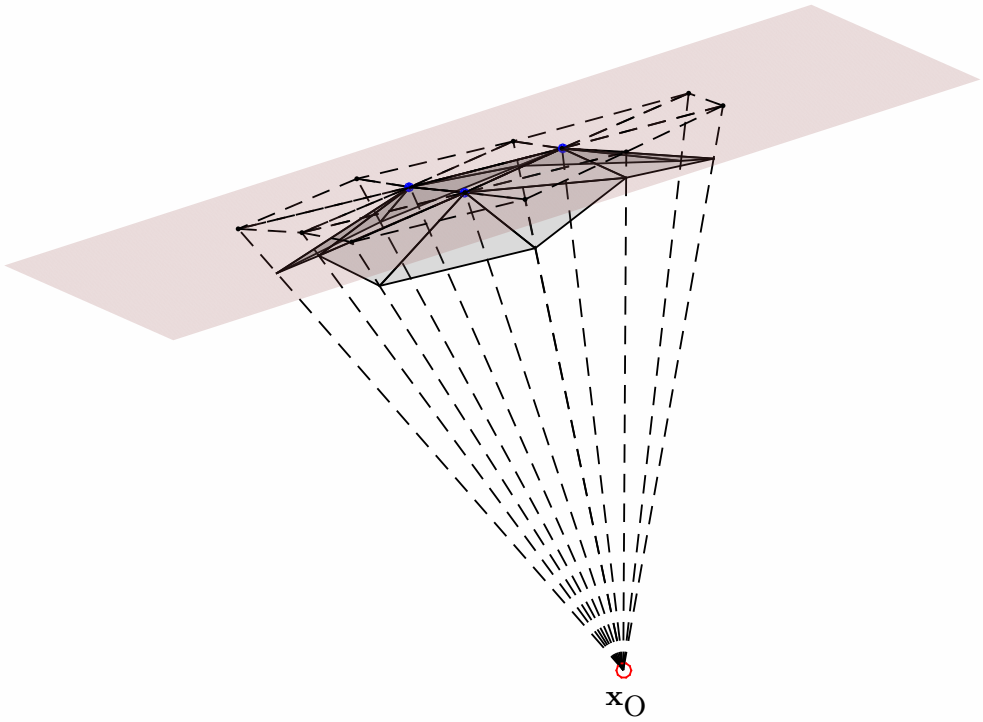


Figure 12. Projection of region S_{Ω_k} onto planar region Ω_k by projection origin $\mathbf{x}_O$.

where $\tilde{\mathbf{x}}_M$: $(\tilde{x}_M, \tilde{y}_M, \tilde{z}_M)$ was the spherical triangle midpoint, $\rho = \sqrt{\tilde{x}_M^2 + \tilde{y}_M^2 + \tilde{z}_M^2}$ and $\gamma = \sqrt{\tilde{x}_M^2 + \tilde{y}_M^2}$. This rotation matrix rotated any triangle midpoint $\tilde{\mathbf{x}}_M$ to the z-axis at $(0, 0, \rho)$ while also preserving length and area. For an arbitrary smooth closed surface S, the transformation is more complex as the projection origin $\mathbf{x}_O$ for each τ_k is not necessarily located at the origin $(0, 0, 0)$.

In order to achieve a similar result as was done with the spherical quadrature in section 2.3.2, the three-dimensional coordinate system is translated such that $\mathbf{x}_O$ is moved to the origin. Thus, define the coordinates

$$\mathbf{x}''' = \tilde{\mathbf{x}} - \mathbf{x}_O.$$

for $\tilde{\mathbf{x}}$ defined in (28). While the surface S may not be symmetric about the origin as with the sphere, the points $\mathbf{x}'''$ can still be rotated about the origin to align $\mathbf{n}_{ABC}$ with the vertical axis. As a result, the rotation matrix becomes

$$M_{12} = \begin{bmatrix} \frac{n_{ABC}^{(x)} n_{ABC}^{(z)}}{\gamma_n \rho_n} & \frac{n_{ABC}^{(y)} n_{ABC}^{(z)}}{\gamma_n \rho_n} & \frac{-\gamma_n}{\rho_n} \\ \frac{-n_{ABC}^{(y)}}{\gamma_n} & \frac{n_{ABC}^{(x)}}{\gamma_n} & 0 \\ \frac{n_{ABC}^{(x)}}{\rho_n} & \frac{n_{ABC}^{(y)}}{\rho_n} & \frac{n_{ABC}^{(z)}}{\rho_n} \end{bmatrix}$$

where

$$\gamma_n = \sqrt{(n_{ABC}^{(x)})^2 + (n_{ABC}^{(y)})^2}$$
$$\rho_n = \sqrt{(n_{ABC}^{(x)})^2 + (n_{ABC}^{(y)})^2 + (n_{ABC}^{(z)})^2}$$

for $\mathbf{n}_{ABC}$: $(n_{ABC}^{(x)}, n_{ABC}^{(y)}, n_{ABC}^{(z)})$. Notice that M_{12} is of the same form as rotation matrix M, only differing by the same reasons that the projection into the plane differs, as in

section 3.3. Rotating the translated coordinate system,

$$\begin{aligned}
\mathbf{x}'' &= M_{12}\mathbf{x}''' \\
&= M_{12}(\tilde{\mathbf{x}} - \mathbf{x}_O) \\
&= \frac{\mathbf{n}_{ABC} \cdot (\mathbf{x}_M - \mathbf{x}_O)}{\mathbf{n}_{ABC} \cdot (\mathbf{x} - \mathbf{x}_O)} M_{12}(\mathbf{x} - \mathbf{x}_O) \\
&= \frac{\mathbf{n}_{ABC} \cdot (\mathbf{x}_M - \mathbf{x}_O)}{\mathbf{n}_{ABC} \cdot (\mathbf{x} - \mathbf{x}_O)} \begin{bmatrix} \frac{n^{(x)}_{ABC} n^{(z)}_{ABC}(x-x_O) + n^{(y)}_{ABC} n^{(z)}_{ABC}(y-y_O) - ((n^{(x)}_{ABC})^2 + (n^{(y)}_{ABC})^2)(z-z_O)}{\sqrt{(n^{(x)}_{ABC})^2 + (n^{(y)}_{ABC})^2}\sqrt{(n^{(x)}_{ABC})^2 + (n^{(y)}_{ABC})^2 + (n^{(z)}_{ABC})^2}} \\ \frac{-n^{(y)}_{ABC}(x-x_O) + n^{(x)}_{ABC}(y-y_O)}{\sqrt{(n^{(x)}_{ABC})^2 + (n^{(y)}_{ABC})^2}} \\ \frac{n^{(x)}_{ABC}(x-x_O) + n^{(y)}_{ABC}(y-y_O) + n^{(z)}_{ABC}(z-z_O)}{\sqrt{(n^{(x)}_{ABC})^2 + (n^{(y)}_{ABC})^2 + (n^{(z)}_{ABC})^2}} \end{bmatrix}
\end{aligned}$$

constructs a new three-dimensional coordinate system such that $\mathbf{x}_M$, $\mathbf{x}$, and $\mathbf{x}_O$ are the same as in section 3.3 (see for example figure 13c). When simplified, the x and y terms (first and second terms of the vector, respectively) remain dependent on $\mathbf{x}$. However, the z term (third term in the vector) simplifies to

$$z'' = \frac{n^{(x)}_{ABC}(x_M - x_O) + n^{(y)}_{ABC}(y_M - y_O) + n^{(z)}_{ABC}(z_M - z_O)}{\sqrt{(n^{(x)}_{ABC})^2 + (n^{(y)}_{ABC})^2 + (n^{(z)}_{ABC})^2}}$$

which is not dependent on the points $\mathbf{x} \in \mathcal{S}_N$. As a result, the plane containing the points $\mathbf{x}''$ is indeed parallel with the (x, y) plane. Further, the two-dimensional coordinate system depends on the x and y terms of $\mathbf{x}''$ by

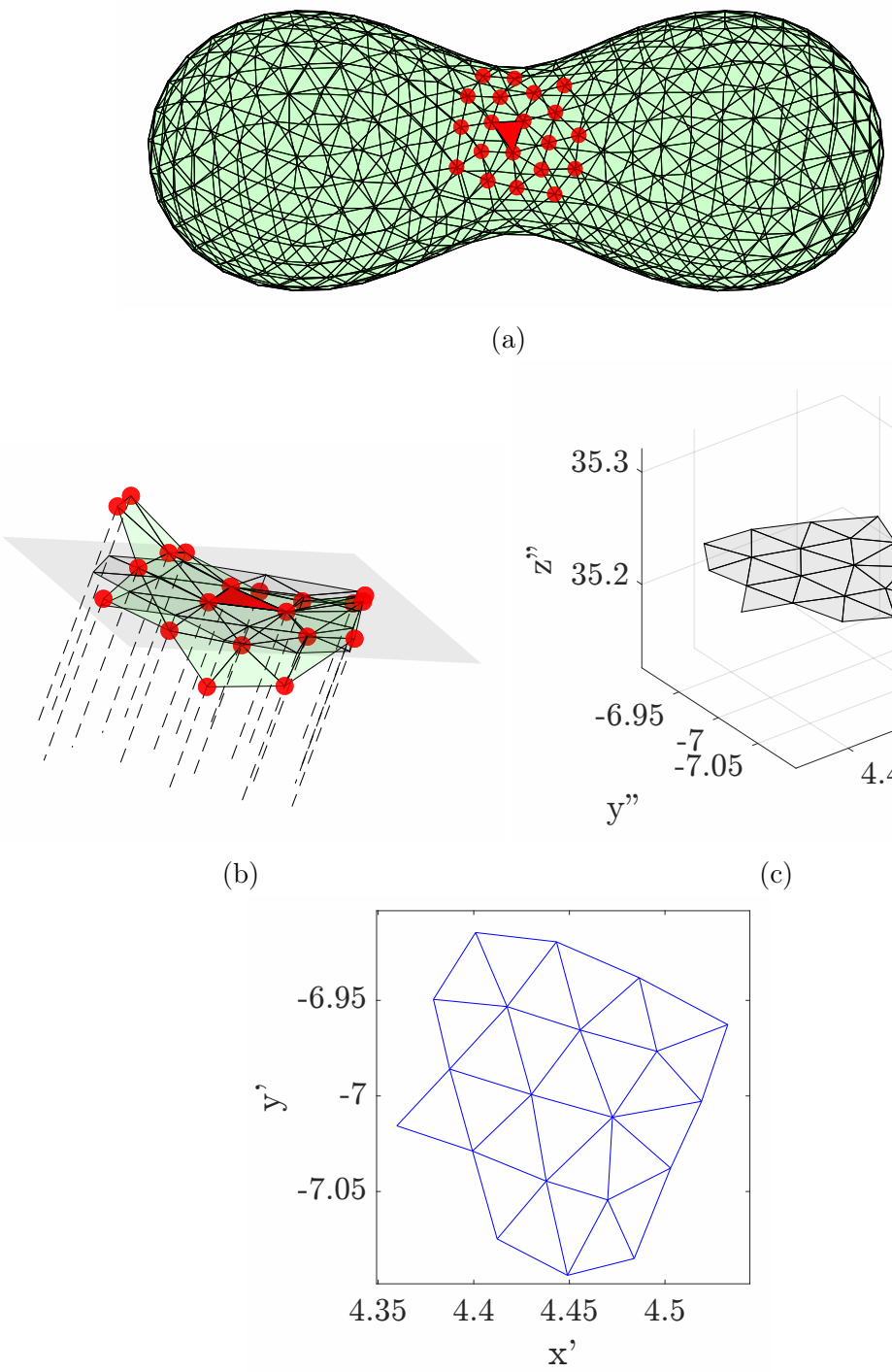
$$\begin{aligned}
\mathbf{x}' &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{x}'' \\
&= \frac{\mathbf{n}_{ABC} \cdot (\mathbf{x}_M - \mathbf{x}_O)}{\mathbf{n}_{ABC} \cdot (\mathbf{x} - \mathbf{x}_O)} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} M_{12}(\mathbf{x} - \mathbf{x}_O)
\end{aligned}$$

where $\mathbf{x}' \in \mathbb{R}^2$, as in figure 13d.

Recall that in the two-dimensional coordinate system, the RBF interpolant is defined based on both $\phi(\eta, \xi)$ and polynomial terms $\{\pi_l(\eta, \xi)\}_{l=1}^{M}$ in (9). Because some $\mathbf{x}'$ points may be located far from the $(0, 0)$ (large angle between $\mathbf{n}_{ABC}$ and $\mathbf{x} - \mathbf{x}_O$) the polynomial function values $\pi_l(\eta, \xi)$ can be large related to the size of ϕ. As a result, the interpolation matrix $\tilde{A}$ for $s(\eta, \xi)$ may be ill-conditioned. To remedy large polynomial terms, the points $\mathbf{x}'$ are shifted one last time by $\mathbf{x}'_M$, the projected midpoint of triangle τ_k. This point is chosen since the projected neighboring nodes are nearest $\mathbf{x}_M$. That is, the final two-dimensional coordinate system is

$$\begin{aligned}
\boldsymbol{\omega} = \begin{bmatrix} \eta \\ \xi \end{bmatrix} &= \mathbf{x}' - \mathbf{x}'_M \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} M_{12} \left(\frac{\mathbf{n}_{ABC} \cdot (\mathbf{x}_M - \mathbf{x}_O)}{\mathbf{n}_{ABC} \cdot (\mathbf{x} - \mathbf{x}_O)} (\mathbf{x} - \mathbf{x}_O) - (\mathbf{x}_M - \mathbf{x}_O) \right) \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} M_{12} \frac{1}{\mathbf{n}_{ABC} \cdot (\mathbf{x} - \mathbf{x}_O)} (\mathbf{n}_{ABC} \times ((\mathbf{x} - \mathbf{x}_O) \times (\mathbf{x}_M - \mathbf{x}_O)))
\end{aligned}$$

where $\mathbf{x}_M$ is moved to the origin. Note that all the operations performed to get $\tilde{\mathbf{x}}$ to $\boldsymbol{\omega}$ are transformations for rotation and/or translation. As a result, the planar region Ω_k maintains the same lengths and area as it did in the triangulation $\mathcal{T}$. In this process, it can be thought of that the surface S is rotated and translated (along with the integrand) before any approximations for the integral (13) are carried out. With this, the planar quadrature weights $\{w_j^{RBF}\}_{j=1}^{n}$ in (13) are computed identically as in section 2.2.3 with calculations of RBF integrals evaluated as in section 2.3.1.



(a)

(b)

(c)

(d)

Figure 13. An example of the progression of coordinates on S from $\mathbb{R}^3$ in S_{Ω_k} to $\mathbb{R}^2$ in Ω_k. (a) A triangulation of the surface $h(x, y, z) = (x^2 + y^2 + z^2)^2 - 2a^2(x^2 - y^2 - z^2) + a^4 - b^4 = 0$ with parameters $a \approx 0.331$, $b \approx 0.348$. (b) The projection of $\mathcal{N}_n$ neighboring nodes and the corresponding S_{Ω_k} surface region from (a). (c) The $\mathcal{N}_n$ projected points after having been shifted and rotated. (d) The two-dimensional coordinate system of the planar region Ω_k before the final shift.

3.5 Converting Planar Weights to Surface Weights

Just as in section 2.3.2, this section will take the surface integral

$$\iint_{S_{\Omega_k}} f(x,y,z)dS$$

for S_{Ω_k} the region of interest (i.e. S_{Ω_k} includes the neighboring nodes $\mathcal{N}_n$ and the corresponding triangulation) and rewrite it as an area integral to be approximated with the quadrature weights determined in the section 3.4. The conversion from a surface to area integral used in section 2.3.2 for the sphere was relatively simple as the distance between the projection point (origin) and the projected triangle mid-point $\tilde{\mathbf{x}}_M$ was always ρ, and the nodes on $\mathcal{S}_N$ were projected onto the tangent plane through a scalar c from section 2.2.1. For an arbitrary smooth closed surface S, the derivation for the transformation ratio for $g(\eta,\xi)$ to $f(x,y,z)$ focuses on the use of a local parameterization of S described by $\mathbf{x}(\eta,\xi)$: $(x(\eta,\xi), y(\eta,\xi), z(\eta,\xi))$. While this chapter acquires much of its information from [42], this particular section is based on the details in [44].

To begin, consider the points $\mathbf{x}(\eta,\xi)$, $\mathbf{x}(\eta+\Delta\eta,\xi)$, and $\mathbf{x}(\eta,\xi+\Delta\xi)$ on S, and the corresponding vectors $\mathbf{x}(\eta+\Delta\eta,\xi)-\mathbf{x}(\eta,\xi)$ and $\mathbf{x}(\eta,\xi+\Delta\xi)-\mathbf{x}(\eta,\xi)$. Let $\Delta\eta, \Delta\xi > 0$. Recall that when two vectors $\mathbf{a}$ and $\mathbf{b}$ compose the sides of a parallelogram, then the area of the parallelogram is $A = ||\mathbf{a} \times \mathbf{b}||$. Thus, the area of the parallelogram with sides $\mathbf{x}(\eta+\Delta\eta,\xi) - \mathbf{x}(\eta,\xi)$ and $\mathbf{x}(\eta,\xi+\Delta\xi) - \mathbf{x}(\eta,\xi)$ is

$$A_{S \text{ patch}} = ||(\mathbf{x}(\eta+\Delta\eta,\xi) - \mathbf{x}(\eta,\xi)) \times (\mathbf{x}(\eta,\xi+\Delta\xi) - \mathbf{x}(\eta,\xi))||_2$$

which can be rewritten as

$$\begin{aligned} A_{S \text{ patch}} &= \left|\left| \frac{\Delta\eta\Delta\xi}{\Delta\eta\Delta\xi} \left[(\mathbf{x}(\eta + \Delta\eta, \xi) - \mathbf{x}(\eta, \xi)) \times (\mathbf{x}(\eta, \xi + \Delta\xi) - \mathbf{x}(\eta, \xi)) \right] \right|\right|_2 \\ &= \left|\left| \frac{\mathbf{x}(\eta + \Delta\eta, \xi) - \mathbf{x}(\eta, \xi)}{\Delta\eta} \times \frac{\mathbf{x}(\eta, \xi + \Delta\xi) - \mathbf{x}(\eta, \xi)}{\Delta\xi} \right|\right|_2 \Delta\eta\Delta\xi \end{aligned}$$

since $\Delta\eta, \Delta\xi > 0$. As $\Delta\eta, \Delta\xi \to 0$,

$$dS = \left|\left| \frac{\partial}{\partial\eta}\mathbf{x}(\eta, \xi) \times \frac{\partial}{\partial\xi}\mathbf{x}(\eta, \xi) \right|\right|_2 d\eta d\xi$$

where dS is the same used for

$$\iint\limits_{S_{\Omega_k}} f(x, y, z) dS$$

In planar coordinates, considering the same parallelogram, the area element for integration in the plane is given as

$$dA = \left|\left| \frac{\partial}{\partial\eta}\tilde{\mathbf{x}}(\eta, \xi) \times \frac{\partial}{\partial\xi}\tilde{\mathbf{x}}(\eta, \xi) \right|\right|_2 d\eta d\xi$$

That is,

$$I(g) = \iint\limits_{\Omega_k} g(\eta, \xi) dA$$

such that $g(\eta, \xi) = f(\eta, \xi) dS/dA$ since $dS = (dS/dA)dA$. Note that $\tilde{\mathbf{x}}$ from section 3.3 is used instead of the points in the (η, ξ) plane. As described before, the operations performed in section 3.4 preserve the same area between planar region Ω_k and the

triangle in $\mathcal{T}$. Expanding the terms in dA,

$$\begin{aligned}\frac{\partial}{\partial\eta}\tilde{\mathbf{x}}(\eta,\xi) &= \frac{\partial}{\partial\eta}\left(\mathbf{x}_O + \frac{\mathbf{n}_{ABC}\cdot(\mathbf{x}_M-\mathbf{x}_O)}{\mathbf{n}_{ABC}\cdot(\mathbf{x}(\eta,\xi)-\mathbf{x}_O)}(\mathbf{x}(\eta,\xi)-\mathbf{x}_O)\right)\\ &= \frac{\mathbf{n}_{ABC}\cdot(\mathbf{x}_M-\mathbf{x}_O)}{(\mathbf{n}_{ABC}\cdot(\mathbf{x}(\eta,\xi)-\mathbf{x}_O))^2}\left[(\mathbf{n}_{ABC}\cdot(\mathbf{x}(\eta,\xi)-\mathbf{x}_O))\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\right.\\ &\quad\left.-\left(\mathbf{n}_{ABC}\cdot\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\right)(\mathbf{x}(\eta,\xi)-\mathbf{x}_O)\right]\end{aligned}$$

and similarly for $\frac{\partial}{\partial\xi}\tilde{\mathbf{x}}(\eta,\xi)$. Therefore,

$$\begin{aligned}&\frac{\partial}{\partial\eta}\tilde{\mathbf{x}}(\eta,\xi)\times\frac{\partial}{\partial\xi}\tilde{\mathbf{x}}(\eta,\xi)\\ &\quad= \frac{(\mathbf{n}_{ABC}\cdot(\mathbf{x}_M-\mathbf{x}_O))^2}{(\mathbf{n}_{ABC}\cdot(\mathbf{x}(\eta,\xi)-\mathbf{x}_O))^3}\left((\mathbf{x}(\eta,\xi)-\mathbf{x}_O)\cdot\left(\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)\right)\right)\mathbf{n}_{ABC}\end{aligned}$$

Substituting this back into the equation for dA,

$$\begin{aligned}dA &= \left|\left|\frac{\partial}{\partial\eta}\tilde{\mathbf{x}}(\eta,\xi)\times\frac{\partial}{\partial\xi}\tilde{\mathbf{x}}(\eta,\xi)\right|\right|_2 d\eta d\xi\\ &= \frac{|\mathbf{n}_{ABC}\cdot(\mathbf{x}_M-\mathbf{x}_O)|^2}{|\mathbf{n}_{ABC}\cdot(\mathbf{x}(\eta,\xi)-\mathbf{x}_O)|^3}\left|(\mathbf{x}(\eta,\xi)-\mathbf{x}_O)\cdot\left(\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)\right)\right|\,||\mathbf{n}_{ABC}||_2 d\eta d\xi\end{aligned}$$

It was noted earlier that $g(\eta,\xi) = f(\eta,\xi)dS/dA$ since $dS = (dS/dA)dA$. Hence,

$$\begin{aligned}dS = \frac{dS}{dA}dA &= \frac{\left|\left|\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)\right|\right|_2 d\eta d\xi}{\frac{|\mathbf{n}_{ABC}\cdot(\mathbf{x}_M-\mathbf{x}_O)|^2}{|\mathbf{n}_{ABC}\cdot(\mathbf{x}(\eta,\xi)-\mathbf{x}_O)|^3}\left|(\mathbf{x}(\eta,\xi)-\mathbf{x}_O)\cdot\left(\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)\right)\right|\,||\mathbf{n}_{ABC}||_2 d\eta d\xi}dA\\ &= \frac{\left|\frac{\mathbf{n}_{ABC}}{||\mathbf{n}_{ABC}||_2}\cdot(\mathbf{x}-\mathbf{x}_O)\right|}{\left|\frac{\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)}{||\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)||_2}\cdot(\mathbf{x}-\mathbf{x}_O)\right|}\left(\frac{\mathbf{n}_{ABC}\cdot(\mathbf{x}-\mathbf{x}_O)}{\mathbf{n}_{ABC}\cdot(\mathbf{x}_M-\mathbf{x}_O)}\right)^2 dA\end{aligned}$$

Therefore,

$$\mathcal{I}_{S_{\Omega_k}}(f) = \iint\limits_{S_{\Omega_k}} f(x,y,z)dS =$$

$$\iint\limits_{\Omega_k} f(x,y,z)\frac{\left|\frac{\mathbf{n}_{ABC}}{||\mathbf{n}_{ABC}||_2}\cdot(\mathbf{x}-\mathbf{x}_O)\right|}{\left|\frac{\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)}{||\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)||_2}\cdot(\mathbf{x}-\mathbf{x}_O)\right|}\left(\frac{\mathbf{n}_{ABC}\cdot(\mathbf{x}-\mathbf{x}_O)}{\mathbf{n}_{ABC}\cdot(\mathbf{x}_M-\mathbf{x}_O)}\right)^2 dA \quad (29)$$

Equation (29) is then used for a parameterized surface S defined by $\mathbf{x}(\eta,\xi)$. From section 3.4,

$$I(g) = \iint\limits_{\Omega_k} g(\eta,\xi)d\eta d\xi \approx \sum_{j=1}^{n} w_j^{RBF} g(\eta_j,\xi_j)$$

where this section determined

$$g(x,y,z) = f(x,y,z)\frac{\left|\frac{\mathbf{n}_{ABC}}{||\mathbf{n}_{ABC}||_2}\cdot(\mathbf{x}-\mathbf{x}_O)\right|}{\left|\frac{\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)}{||\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)||_2}\cdot(\mathbf{x}-\mathbf{x}_O)\right|}\left(\frac{\mathbf{n}_{ABC}\cdot(\mathbf{x}-\mathbf{x}_O)}{\mathbf{n}_{ABC}\cdot(\mathbf{x}_M-\mathbf{x}_O)}\right)^2$$

As a result, a similar procedure as in section 2.3.2 applies when determining the overall weights $\mathcal{W}_N$. Recall that each quadrature weight w_j^{RBF} is actually computed as $(w_k^{RBF})_j$ for each triangle $k = 1,...,K$, and each projected node (x_j,y_j,z_j) for $j = 1,...,n$. As with the spherical quadrature, let $\mathcal{K}_i$, for $i = 1,...,N$ quadrature nodes, be the set of all pairs (k,j) such that $((\eta_k)_j,(\xi_k)_j) \mapsto (x_i,y_i,z_i)$. Then the

integral of f over the whole surface S is

$$\begin{aligned}
\mathcal{I}_S(f) &= \sum_{k=1}^{K} \iint_{\tau_k} f(x,y,z)dS \\
&\approx \sum_{i=1}^{N} \left(\sum_{(k,j)\in\mathcal{K}_i} (w_k^{RBF})_j \frac{\left| \frac{\mathbf{n}_{ABC}}{||\mathbf{n}_{ABC}||_2} \cdot ((\mathbf{x}_k)_j - (\mathbf{x}_O)_k) \right|}{\left| \frac{\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)}{||\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)||_2} \cdot ((\mathbf{x}_k)_j - (\mathbf{x}_O)_k) \right|} \left(\frac{\mathbf{n}_{ABC}\cdot((\mathbf{x}_k)_j - (\mathbf{x}_O)_k)}{\mathbf{n}_{ABC}\cdot((\mathbf{x}_M)_k - (\mathbf{x}_O)_k)} \right)^2 \right) \\
&\quad f(x_i,y_i,z_i) \\
&= \sum_{i=1}^{N} W_i f(x_i,y_i,z_i) = \tilde{\mathcal{I}}_S(f)
\end{aligned} \tag{30}$$

Equation (30) is the numerical approximation for the integral of f over an explicitly parameterized surface. Suppose instead that the surface S is defined implicitly by $h(x,y,z) = 0$. Note that

$$\frac{\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)}{||\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)||_2}$$

is a unit normal to the surface S. This can be related to the gradient $\nabla h(x,y,z)$ (also a surface normal) by

$$\begin{aligned}
&\frac{\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)}{||\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)||_2} \\
&\qquad = \text{sign}\left(\left(\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi)\times\frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi) \right) \cdot \nabla h(x,y,z) \right) \frac{\nabla h(x,y,z)}{||\nabla h(x,y,z)||_2}
\end{aligned}$$

which when placed in (30) gives

$$\begin{aligned}
\mathcal{I}_S(f) &= \sum_{k=1}^{K} \iint_{\tau_k} f(x,y,z)dS \\
&\approx \sum_{i=1}^{N} \left(\sum_{(k,j)\in\mathcal{K}_i} (w_k^{RBF})_j \frac{\left| \frac{\mathbf{n}_{ABC}}{||\mathbf{n}_{ABC}||_2} \cdot ((\mathbf{x}_k)_j - (\mathbf{x}_O)_k) \right|}{\left| \frac{\nabla h(x,y,z)}{||\nabla h(x,y,z)||_2} \cdot ((\mathbf{x}_k)_j - (\mathbf{x}_O)_k) \right|} \left(\frac{\mathbf{n}_{ABC} \cdot ((\mathbf{x}_k)_j - (\mathbf{x}_O)_k)}{\mathbf{n}_{ABC} \cdot ((\mathbf{x}_M)_k - (\mathbf{x}_O)_k)} \right)^2 \right) \\
&\quad \cdot f(x_i, y_i, z_i) \\
&= \sum_{i=1}^{N} W_i f(x_i, y_i, z_i) = \tilde{\mathcal{I}}_S(f)
\end{aligned} \tag{31}$$

Equation (31) is hence the numerical approximation for the integral of f over an implicitly defined surface. Consider, however, that S may not be defined explicitly through parameterization nor implicitly by $h(x,y,z) = 0$. Just as the intermediate quadrature weights in section 3.3 were determined with only quadrature nodes $\mathcal{S}_N$ and a triangulation $\mathcal{T}$, $\mathcal{I}_S(f)$ must also be computable under the same conditions. Section 3.6 covers this case.

3.6 Approximating a Normal to the Surface

Suppose that the surface S is defined neither explicitly through a parameterization nor implicitly by $h(x,y,z) = 0$. Instead, only a node set $\mathcal{S}_N$ and triangulation $\mathcal{T}$ are provided. The terms

$$\frac{\nabla h(\mathbf{x})}{||\nabla h(\mathbf{x})||_2} \text{ and } \frac{\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi) \times \frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)}{||\frac{\partial}{\partial\eta}\mathbf{x}(\eta,\xi) \times \frac{\partial}{\partial\xi}\mathbf{x}(\eta,\xi)||_2}$$

were the only terms in (30) and (31) which did not use the information provided by $\mathcal{S}_N$ and $\mathcal{T}$. Notice that just as points in the (η,ξ) plane are dependent on the nodes in $\mathcal{S}_N$, so too can the nodes $\{(x_j,y_j,z_j)\}_{j=1}^n$ for n projected nodes be expressed as

a function of η and ξ. That is, the projection and transformations carried out on $\{(x_j, y_j, z_j)\}_{j=1}^n$ provide a local "explicit" parameterization for each projected point in $\mathcal{S}_N$:

$$(x(\eta_j, \xi_j), y(\eta_j, \xi_j), z(\eta_j, \xi_j)) = (x_j, y_j, z_j) \text{ for } j = 1, ..., n$$

where (η_j, ξ_j) is the two-dimensional planar version of (x_j, y_j, z_j). While the true parameterization $(x(\eta, \xi), y(\eta, \xi), z(\eta, \xi))$ is unknown (since S is not equation-based), $(x(\eta, \xi), y(\eta, \xi), z(\eta, \xi))$ can be approximated through a radial basis function interpolation

$$\begin{aligned}
x(\eta, \xi) \approx s_x(\eta, \xi) &:= \sum_{j=1}^{n} c_{xj}^{RBF} \phi\left(\sqrt{(\eta - \eta_j)^2 + (\xi - \xi_j)^2}\right) + \sum_{l=1}^{M} c_{xl}^{p} \pi_l(\eta, \xi) \\
y(\eta, \xi) \approx s_y(\eta, \xi) &:= \sum_{j=1}^{n} c_{yj}^{RBF} \phi\left(\sqrt{(\eta - \eta_j)^2 + (\xi - \xi_j)^2}\right) + \sum_{l=1}^{M} c_{yl}^{p} \pi_l(\eta, \xi) \\
z(\eta, \xi) \approx s_z(\eta, \xi) &:= \sum_{j=1}^{n} c_{zj}^{RBF} \phi\left(\sqrt{(\eta - \eta_j)^2 + (\xi - \xi_j)^2}\right) + \sum_{l=1}^{M} c_{zl}^{p} \pi_l(\eta, \xi)
\end{aligned}$$

Without loss of generality, $c_{x1}^{RBF}, ..., c_{xn}^{RBF}$, $c_{x1}^{p}, ..., c_{xM}^{p} \in \mathbb{R}$ are chosen such that $s_x(\eta_j, \xi_j) = x(\eta_j, \xi_j)$, $j = 1, 2, ..., n$, along with constraints $\sum_{j=1}^{n} c_{xj}^{RBF} \pi_l(\eta_j, \xi_j) = 0$, for $l = 1, 2, ..., M$. The coefficients for $s_x(\eta, \xi)$, $s_y(\eta, \xi)$, and $s_z(\eta, \xi)$ are evaluated by the matrix multiplication elaborated in section 2.2.3. Once the interpolants $s_x(\eta, \xi)$, $s_y(\eta, \xi)$, and $s_z(\eta, \xi)$ are constructed,

$$\frac{\partial}{\partial \eta}\mathbf{x}(\eta, \xi) \times \frac{\partial}{\partial \xi}\mathbf{x}(\eta, \xi)$$

can be approximated by

$$\frac{\partial}{\partial \eta}s_{\mathbf{x}}(\eta, \xi) \times \frac{\partial}{\partial \xi}s_{\mathbf{x}}(\eta, \xi)$$

where

$$s_{\mathbf{x}}(\eta,\xi) = \begin{bmatrix} s_x(\eta,\xi) \\ s_y(\eta,\xi) \\ s_z(\eta,\xi) \end{bmatrix} \approx \begin{bmatrix} x(\eta,\xi) \\ y(\eta,\xi) \\ z(\eta,\xi) \end{bmatrix} = \mathbf{x}(\eta,\xi)$$

The partial derivatives of $s_x(\eta,\xi)$ (and therefore $s_{\mathbf{x}}(\eta,\xi)$) can be determined by, for instance,

$$\frac{\partial}{\partial\eta}s_x(\eta,\xi) = \sum_{j=1}^{n} c_{xj}^{RBF}\frac{\partial}{\partial\eta}\phi\left(\sqrt{(\eta-\eta_j)^2+(\xi-\xi_j)^2}\right) + \sum_{l=1}^{M} c_{xl}^{p}\frac{\partial}{\partial\eta}\pi_l(\eta,\xi)$$
$$\frac{\partial}{\partial\xi}s_x(\eta,\xi) = \sum_{j=1}^{n} c_{xj}^{RBF}\frac{\partial}{\partial\xi}\phi\left(\sqrt{(\eta-\eta_j)^2+(\xi-\xi_j)^2}\right) + \sum_{l=1}^{M} c_{xl}^{p}\frac{\partial}{\partial\xi}\pi_l(\eta,\xi)$$

where the partial derivatives of the RBF ϕ will depend on the chosen basis function. Hence, (30) becomes

$$\begin{aligned}
\mathcal{I}_S(f) &= \sum_{k=1}^{K}\iint_{\tau_k} f(x,y,z)dS \\
&\approx \sum_{i=1}^{N}\left(\sum_{(k,j)\in\mathcal{K}_i}(w_k^{RBF})_j \frac{\left|\frac{\mathbf{n}_{ABC}}{||\mathbf{n}_{ABC}||_2}\cdot((\mathbf{x}_k)_j-(\mathbf{x}_O)_k)\right|}{\left|\frac{\frac{\partial}{\partial\eta}s_{\mathbf{x}}(\eta,\xi)\times\frac{\partial}{\partial\xi}s_{\mathbf{x}}(\eta,\xi)}{||\frac{\partial}{\partial\eta}s_{\mathbf{x}}(\eta,\xi)\times\frac{\partial}{\partial\xi}s_{\mathbf{x}}(\eta,\xi)||_2}\cdot((\mathbf{x}_k)_j-(\mathbf{x}_O)_k)\right|}\left(\frac{\mathbf{n}_{ABC}\cdot((\mathbf{x}_k)_j-(\mathbf{x}_O)_k)}{\mathbf{n}_{ABC}\cdot((\mathbf{x}_M)_k-(\mathbf{x}_O)_k)}\right)^2\right) \\
&\quad f(x_i,y_i,z_i) \\
&= \sum_{i=1}^{N} W_i f(x_i,y_i,z_i) = \tilde{\mathcal{I}}_S(f)
\end{aligned} \tag{32}$$

which is the numerical approximation of f over a surface S expressed by a set of nodes $\mathcal{S}_N$ and triangulation $\mathcal{T}$.

IV. Results

4.1 Test Surface

This chapter applies the material presented in Chapter III to a surface S implicitly defined by

$$h(x,y,z) = (x^2+y^2+z^2)^2 - 2a^2(x^2-y^2-z^2) + a^4 - b^4 = 0 \tag{33}$$

and several test integrands. S is a surface of revolution induced by the Cassini oval [45] in the (x,y) plane (rotated about the x-axis) with parameters a and b. Examples can be seen in figure 15. Parameters a and b are chosen to satisfy $a = \lambda b$ for values of $\lambda = \{0.8, 0.85, 0.9, 0.95\}$ and b such that S has surface area equal to 1 (see figure 14 for slices of S in the (x,y) plane). This was done to normalize the surface area's contribution to any error curves shown. The surface area of S can be approximated using MATLAB's **quad2d** where, in polar coordinates [39],

$$A(S) = \int_0^{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \left|\left| \frac{\partial}{\partial\theta}\mathbf{x}(\theta,\phi) \times \frac{\partial}{\partial\phi}\mathbf{x}(\theta,\phi) \right|\right|_2 d\theta d\phi$$

with

$$\mathbf{x}(\theta,\phi) = r(\phi) \begin{bmatrix} \cos(\phi) \\ \sin(\phi)\sin(\theta) \\ sin(\phi)cos(\theta) \end{bmatrix}$$

and [45]

$$r(\phi) = \sqrt{\sqrt{b^4 - a^4 + a^4\cos^2(2\phi)} + a^2\cos(2\phi)}$$

The quadrature nodes $\mathcal{S}_N$ and triangulation $\mathcal{T}$ were generated using **distmeshsurface** [43], where the fill distance (as described in section 2.3.2) varied from 0.005 to

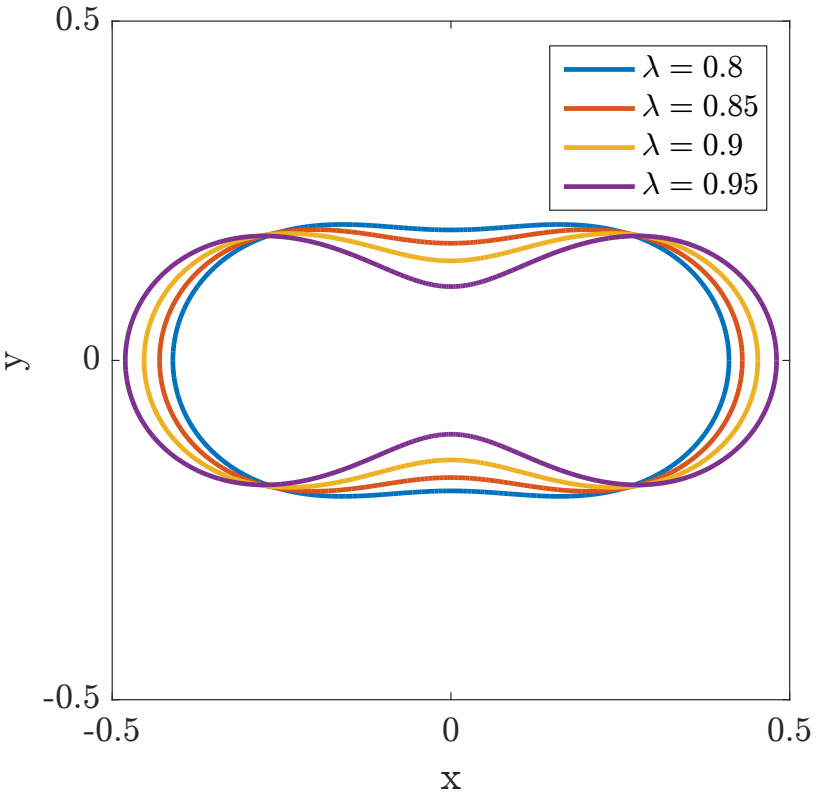


Figure 14. Slices of Cassini ovals used for application, where $a = \lambda b$ for λ values shown.

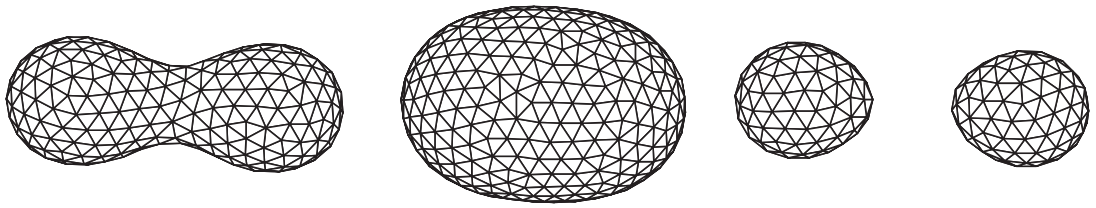


Figure 15. Example Cassini ovals rotated about the x-axis for given parameter values a and b. For $a < b$, S appears as an increasingly pinched oval as $a \to b$. For $a = b$, S is completely pinched (but inherently not smooth), while $a > b$ depicts S as a piecewise smooth closed surface with two separated ovals.

0.10. Mesh sizes much smaller than 0.005 were not considered because of the computational time required to generate them with **distmeshsurface**, while mesh sizes much larger than 0.10 generate too few nodes for a reasonable approximation.

4.2 Applying the Chapter III Quadrature Method

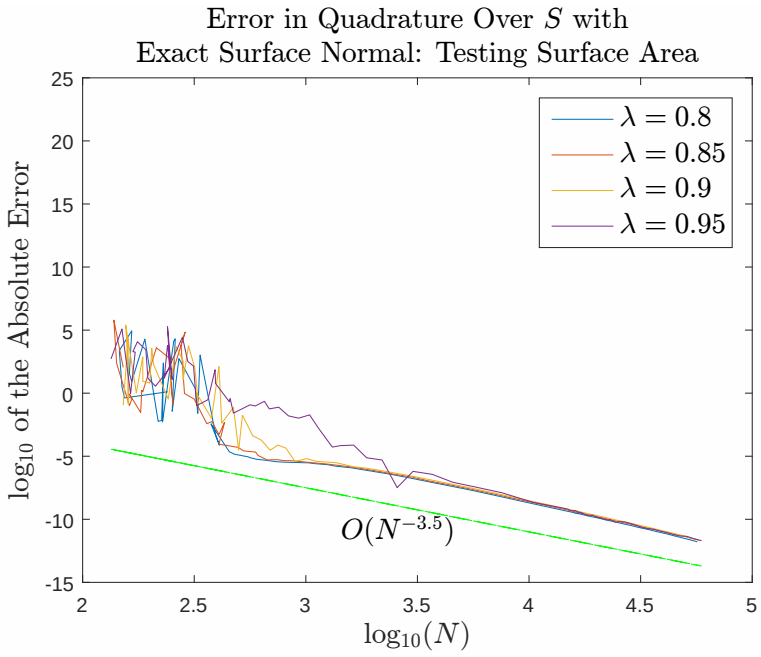
Following section 3.4, the approximation (12) here uses the RBF $\phi(r) = r^7$, $n = 80$ (number of nearest neighbors), and $m = 7$ (degree of polynomial terms). From the implicitly defined surface in (33), the quadrature weights for (31) were determined.

If $f(x, y, z) = 1$, then (31) becomes the surface area of S:

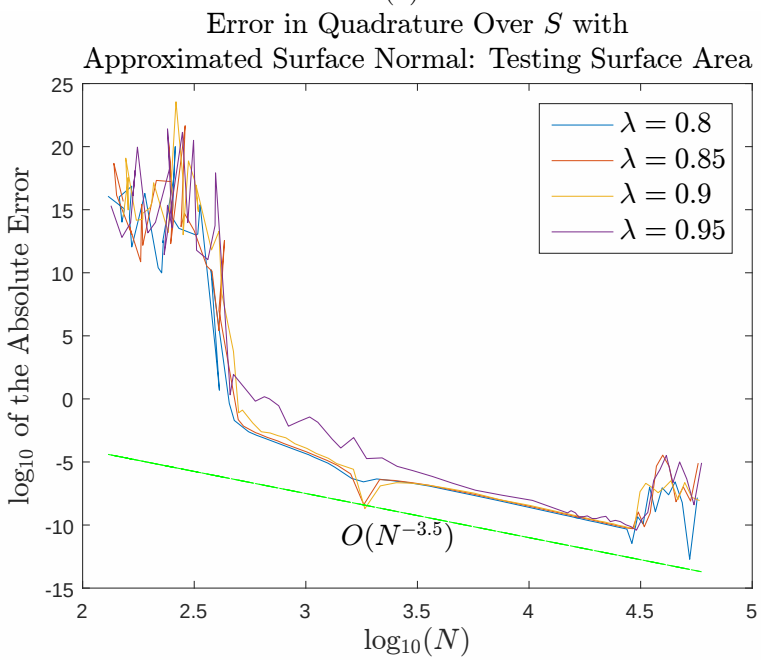
$$A(S) = 1 = \iint_S 1 dS \approx \sum_{i=1}^{N} W_i$$

As described in section 4.1, the surface area of S was set up to equal 1. With this, the sum of the quadrature weights in (33) should be approximately equal to 1. Suppose, however, that (32) was used instead such that an approximation to the true normal $(\nabla h(x, y, z))$ was utilized:

$$\frac{\partial}{\partial \eta} s_{\mathbf{x}}(\eta, \xi) \times \frac{\partial}{\partial \xi} s_{\mathbf{x}}(\eta, \xi)$$



(a)



(b)

Figure 16. Relative error in RBF quadrature for the surface area over surfaces $h(x, y, z) = 0$ for various λ on sets of quasi-uniformly spaced nodes with: (a) surface normal computed via ∇h; (b) surface normal approximated. For the computations, $\phi(r) = r^7$, $n = 80$, and $m = 7$.

Estimates for computing surface areas of S defined by (33) are given in figure 16. In either case, a convergence rate of $O(N^{-3.5})$ is achieved. The large errors for smaller N in both cases, and larger N in the case of the approximate surface normal, will be discussed after the results are presented.

Now consider

$$f_i(x, y, z) = \mathbf{F}_i \cdot \mathbf{n}, i = 1, 2, 3$$

where

$$\mathbf{n} = \frac{\nabla h(x, y, z)}{||\nabla h(x, y, z)||_2}$$

$$\mathbf{F}_1 = \frac{1}{3} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\mathbf{F}_2 = \frac{1}{(x^2 + y^2 + z^2)^{3/2}} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\mathbf{F}_3 = \nabla \times \mathbf{F}_{3b} = \nabla \times \begin{bmatrix} -\frac{1}{2}y \\ \frac{1}{2}x \\ 0 \end{bmatrix}$$

Note that $f_1, f_3 \in C^\infty(S)$ while f_2 is also continuous except for when the denominator is zero. That is, $f_2 \in C^\infty(S)$ since $x^2 + y^2 + z^2 = 0$ does not exist on S. Also consider

$$f_4(x, y, z) = \frac{2}{\pi}\tan^{-1}(100z) \text{ and } f_5(x, y, z) = \text{sign}(z) = \begin{cases} +1, & z > 0 \\ 0, & z = 0 \\ -1, & z < 0 \end{cases}$$

Mathematically, $\mathcal{I}_S(f_1)$ is (by the Divergence Theorem) [39]

$$\begin{aligned}\mathcal{I}_S(f_1) = \iint\limits_S (\mathbf{F}_1 \cdot \mathbf{n})\, dS &= \iiint\limits_V (\nabla \cdot \mathbf{F}_1)\, dV \\ &= \iiint\limits_V 1 dV\end{aligned}$$

That is, it is the volume of the region enclosed by the surface S. Recall that S is a surface of revolution about the x-axis, such that

$$h(x, y, 0) = (x^2 + y^2)^2 - 2a^2(x^2 - y^2) + a^4 - b^4 = 0 \tag{34}$$

Then,

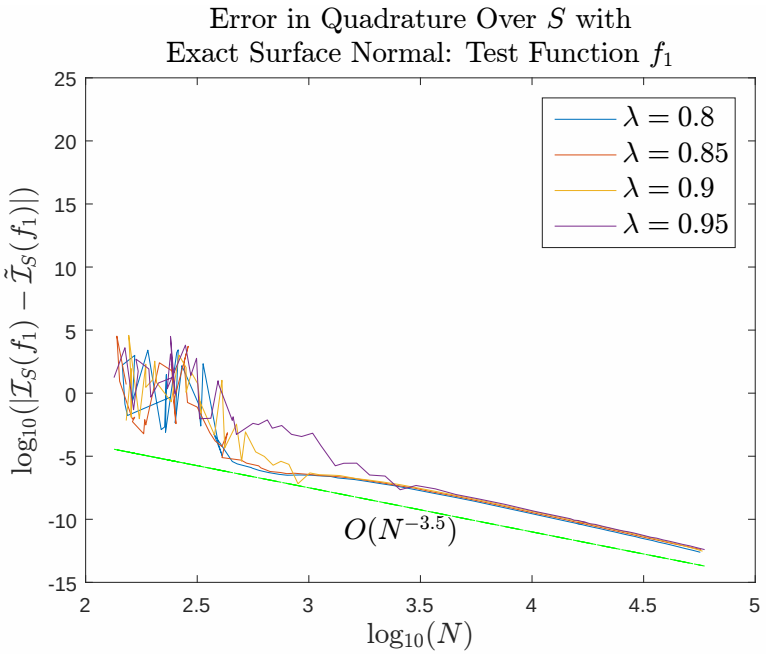
$$r(x) = y(x) = \sqrt{-a^2 - x^2 + \sqrt{b^4 - 4a^2x^2}}$$

where $y(x)$ is one of four roots of (34) and, from this, $x = \sqrt{a^2 + b^2}$ when $y = 0$. Therefore,

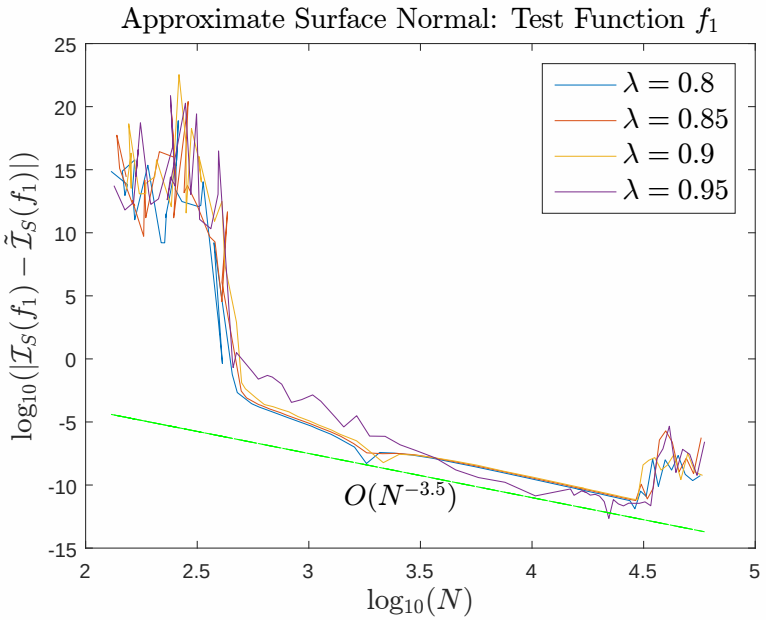
$$\begin{aligned}\mathcal{I}_S(f_1) &= \int\limits_x \left(\iint\limits_A 1 dy dz \right) dx \\ &= \int\limits_0^{\sqrt{a^2+b^2}} \pi r^2(x) dx \\ &= 2\pi \int\limits_0^{\sqrt{a^2+b^2}} \left(-a^2 - x^2 + \sqrt{b^4 - 4a^2x^2} \right) dx \\ &= \frac{\pi}{6a} \left[2a(b^2 - 2a^2)\sqrt{a^2 + b^2} + 3b^4 \sinh^{-1}\left(\frac{2a\sqrt{a^2 + b^2}}{b^2} \right) \right]\end{aligned}$$

The error plot for $\mathcal{I}_S(f_1)$ can be found in figure 17.

The integral of $f_2(x, y, z)$, however, uses the Divergence Theorem [39] and a prop-



(a)



(b)

Figure 17. Relative error in RBF quadrature for function $f_1(x, y, z)$ over surfaces $h(x, y, z) = 0$ for various λ on sets of quasi-uniformly spaced nodes with: (a) surface normal computed via ∇h; (b) surface normal approximated. For the computations, $\phi(r) = r^7$, $n = 80$, and $m = 7$.

erty concerning $\mathbf{F}_2(x, y, z)$ [46]:

$$\begin{aligned}\mathcal{I}_S(f_2) = \iint_S (\mathbf{F}_2 \cdot \mathbf{n})\, dS &= \iiint_V (\nabla \cdot \mathbf{F}_2)\, dV \\ &= \iiint_V 4\pi\delta^3\left(x^2 + y^2 + z^2\right) dV \\ &= 4\pi\end{aligned}$$

The error plot for $\mathcal{I}_S(f_2)$ can be found in figure 18.

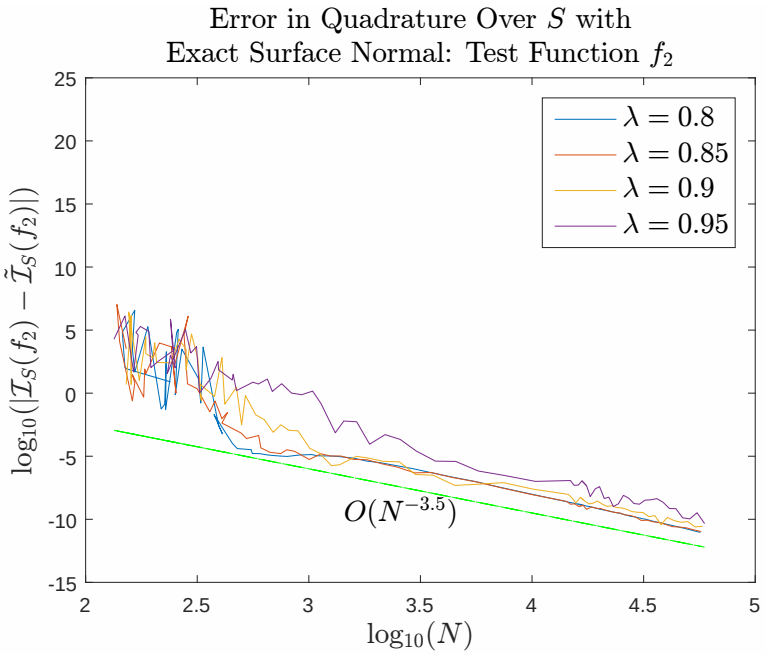
The integral of $f_3(x, y, z)$ uses Stokes' Theorem [39],

$$\begin{aligned}\mathcal{I}_S(f_3) = \iint_S ((\nabla \times \mathbf{F}_{3b}) \cdot \mathbf{n})\, dS &= \oint_{\partial S} \mathbf{F}_{3b} \cdot d\mathbf{r} \\ &= \int_{t_0}^{t_f} \mathbf{F}_{3b}(\mathbf{r}(t)) \cdot \mathbf{r}'(t) dt \\ &= \int_{t_0}^{t_f} \left(F_{3b}^{(1)}(x(t), y(t), z(t))x'(t) + F_{3b}^{(2)}(x(t), y(t), z(t))y'(t)\right) dt \\ &= \int_{\partial S} \left(F_{3b}^{(1)}(x, y, z)dx + F_{3b}^{(2)}(x, y, z)dy\right)\end{aligned}$$

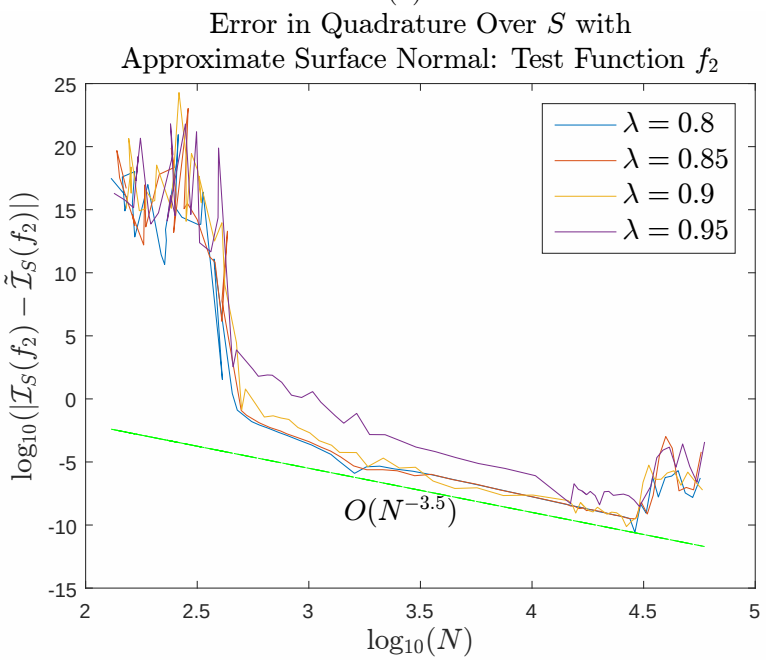
which Green's Theorem [39] relates the last line to

$$\begin{aligned}\mathcal{I}_S(f_3) &= \iint_A \left(\frac{\partial}{\partial x}F_{3b}^{(2)}(x, y, z) - \frac{\partial}{\partial y}F_{3b}^{(1)}(x, y, z)\right) dxdy \\ &= \iint_A 1 dxdy \\ &= 0\end{aligned}$$

The last step follows from the assumptions of Stokes' Theorem, where the the surface contour (boundary) traverses one direction for the upper half of S but the opposite

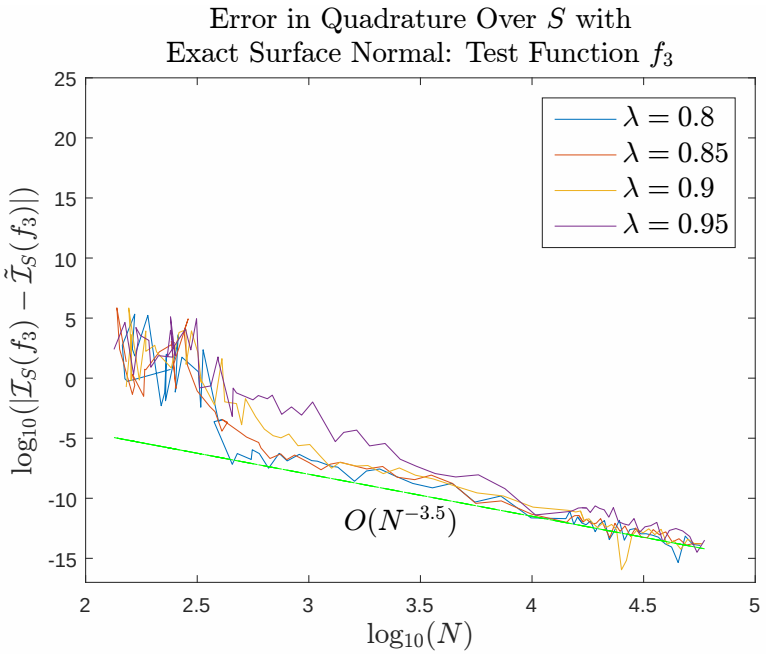


(a)

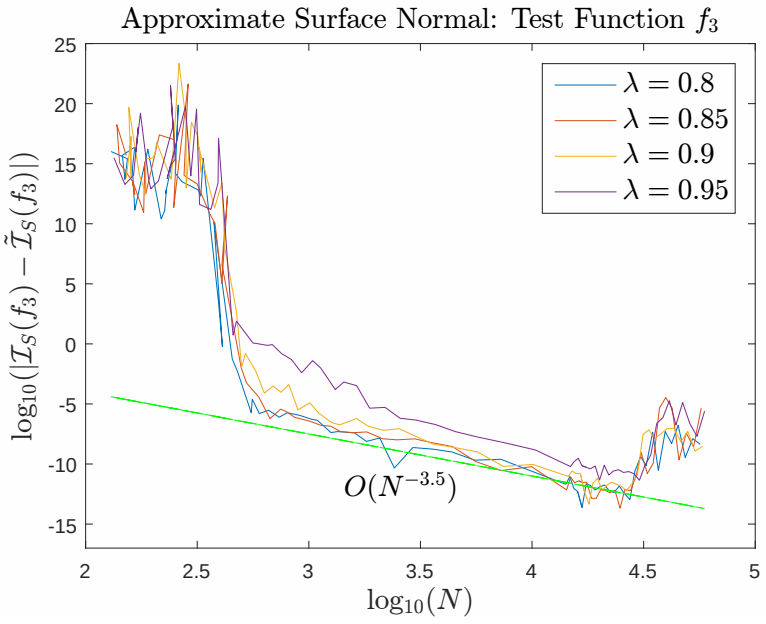


(b)

Figure 18. Relative error in RBF quadrature for function $f_2(x, y, z)$ over surfaces $h(x, y, z) = 0$ for various λ on sets of quasi-uniformly spaced nodes with: (a) surface normal computed via ∇h; (b) surface normal approximated. For the computations, $\phi(r) = r^7$, $n = 80$, and $m = 7$.



(a)



(b)

Figure 19. Relative error in RBF quadrature for function $f_3(x, y, z)$ over surfaces $h(x, y, z) = 0$ for various λ on sets of quasi-uniformly spaced nodes with: (a) surface normal computed via ∇h; (b) surface normal approximated. For the computations, $\phi(r) = r^7$, $n = 80$, and $m = 7$.

direction for the bottom half of S, so that the surface normals are consistent from one to the other [39]. This causes the integrals for the upper and lower halves of S to be equal in magnitude but opposite in sign. The resultant error plot for $\mathcal{I}_S(f_3)$ can be found in figure 19.

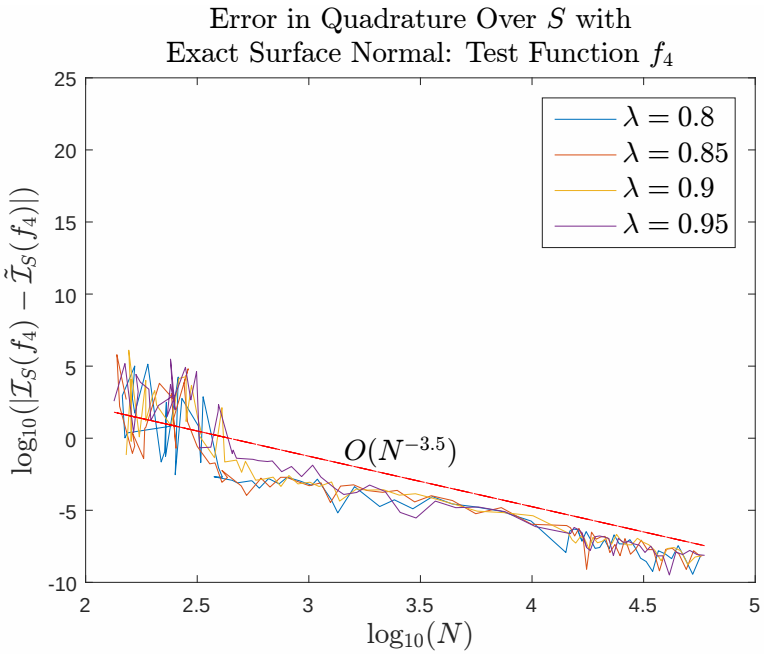
Now, consider $f_4(x, y, z) = \frac{2}{\pi}\tan^{-1}(100z)$ (where the details concerning f_4 are from [47]). Since S is a surface of revolution induced by the Cassini oval in the (x, y) plane, then $S \in \mathbb{R}^3$ is also symmetric about the (x, y) plane. Therefore, define the portion of S for which $z \geq 0$ to be S^+. Similarly, let S^- be the portion of S for which $z < 0$. So,

$$\begin{aligned}
\mathcal{I}_S(f_4) &= \iint_S \frac{2}{\pi}\tan^{-1}(100z)dS \\
&= \iint_{S^+} \frac{2}{\pi}\tan^{-1}(100z)dS + \iint_{S^-} \frac{2}{\pi}\tan^{-1}(100z)dS \\
&= \iint_{S^+} \frac{2}{\pi}\tan^{-1}(100z)dS + \iint_{S^+} \frac{2}{\pi}\tan^{-1}(100(-z))dS
\end{aligned}$$

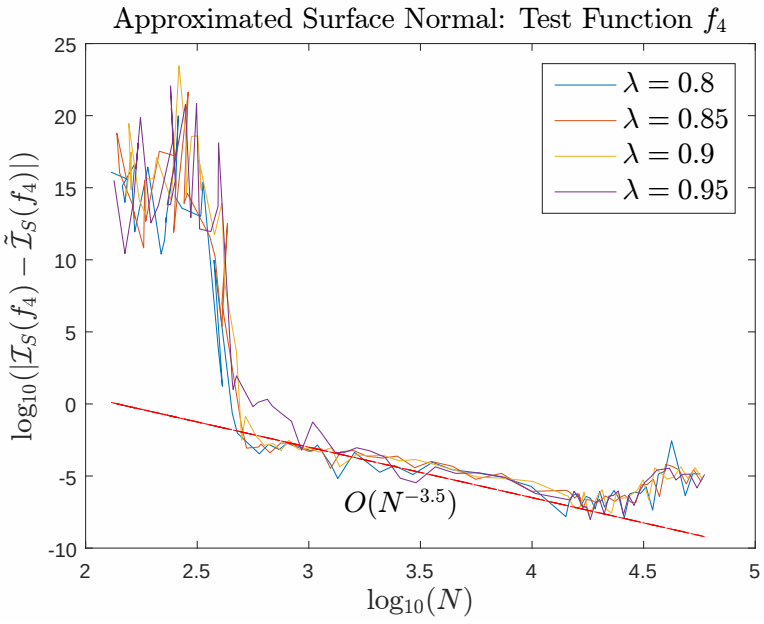
Since arctangent is an odd function (i.e. $\tan^{-1}(-\mathbf{x}) = -\tan^{-1}(\mathbf{x})$),

$$\mathcal{I}_S(f_4) = \iint_{S^+} \frac{2}{\pi}\tan^{-1}(100z)dS - \iint_{S^+} \frac{2}{\pi}\tan^{-1}(100z)dS = 0$$

The resultant error plot for $f_4(x, y, z)$ is illustrated in figure 20.



(a)



(b)

Figure 20. Relative error in RBF quadrature for function $f_4(x, y, z)$ over surfaces $h(x, y, z) = 0$ for various λ on sets of quasi-uniformly spaced nodes with: (a) surface normal computed via ∇h; (b) surface normal approximated. For the computations, $\phi(r) = r^7$, $n = 80$, and $m = 7$.

Similar to $\mathcal{I}_S(f_4)$, the integral of $f_5(x,y,z) = \text{sign}(z)$ is

$$\begin{aligned}\mathcal{I}_S(f_5) &= \iint_S \text{sign}(z) dS \\ &= \iint_{S^+} (+1) dS + \iint_{S^-} (-1) dS \\ &= \iint_{S^+} dS - \iint_{S^-} dS\end{aligned}$$

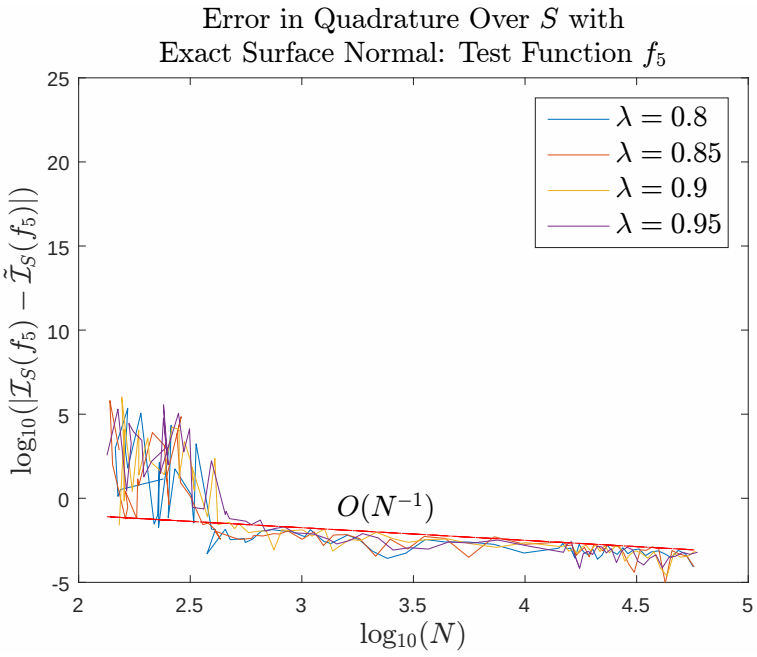
Since the surface integral over S (with an integrand of 1) represents the surface area of S, the surface integrals over S^+ and S^- each represent half the surface area of S. That is,

$$\mathcal{I}_S(f_5) = \left(\frac{1}{2}\iint_S dS\right) - \left(\frac{1}{2}\iint_S dS\right) = 0$$

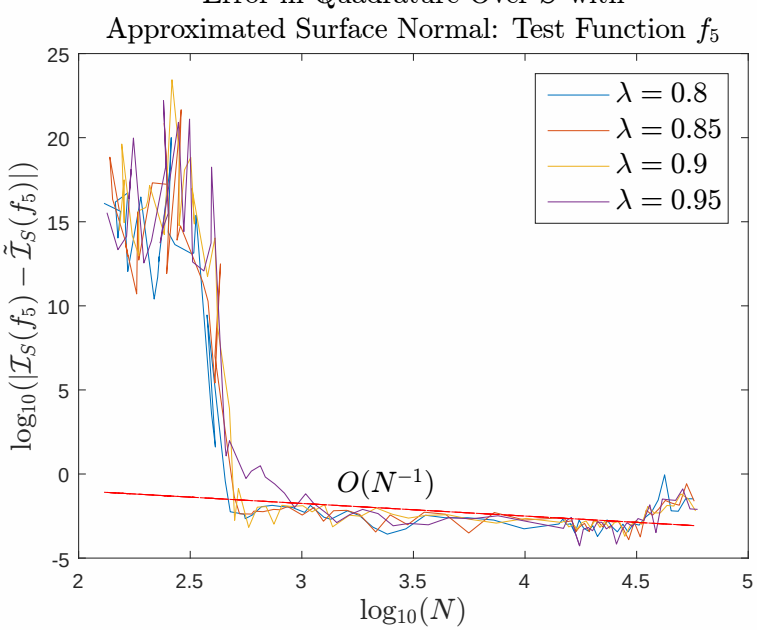
The resultant error plot for $f_5(x,y,z)$ is illustrated in figure 21.

Notice that with almost all the error plots, the convergence rate is $O(N^{-3.5})$ or better, which is that attained with the sphere for continuous functions. The convergence rate for $f_4(x,y,z)$ is closer to $O(N^{-2.5})$. A lower convergence rate is to be expected due to the steep gradient that occurs along the (x,y) plane. The convergence rate for $f_5(x,y,z)$ was approximately $O(N^{-0.75})$, which is similar to the convergence rate illustrated by $\tilde{f}_2(x,y,z)$ in section 2.3.3. These rates are not surprising, since as long as the surface gradient ∇h is nonzero, the surface S is as smooth as the sphere.

Recall the computational costs of $O(N\text{log}N)$ operations and $O(N)$ memory storage for the sphere quadrature. These computational costs will remain nearly the same for the smooth closed surface S since the operations performed for the intermediate quadrature weights in (13) and final quadrature weights in (1) are the same for both the sphere and smooth surface S. The cases of the smooth surface and the sphere differ only in that a projection origin must be computed for each triangle. This adds



(a)



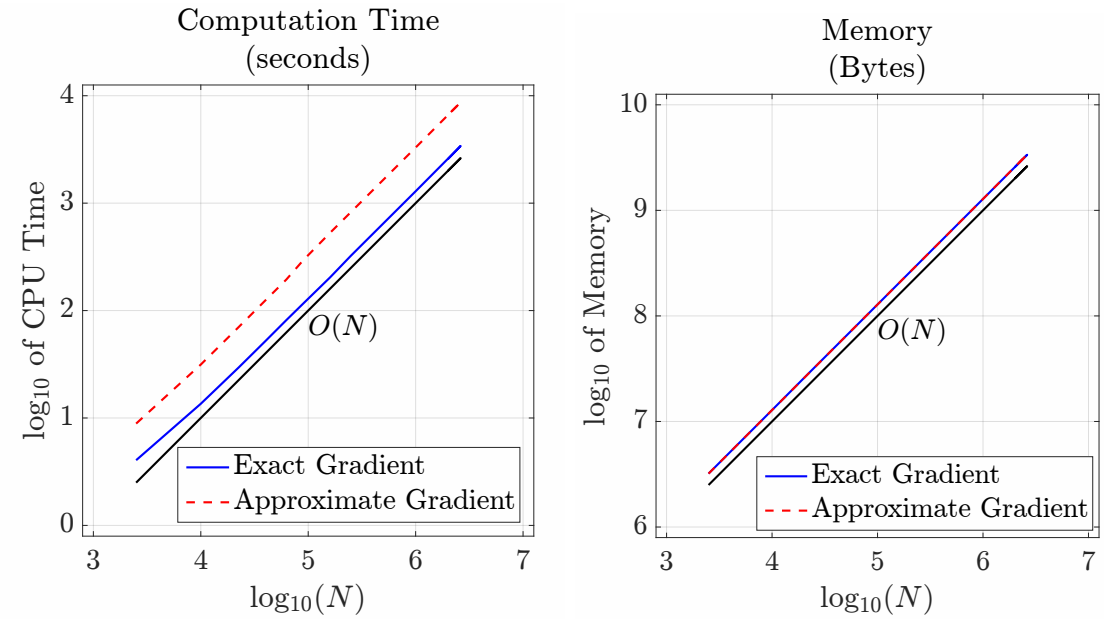
(b)

Figure 21. Relative error in RBF quadrature for function $f_5(x, y, z)$ over surfaces $h(x, y, z) = 0$ for various λ on sets of quasi-uniformly spaced nodes with: (a) surface normal computed via ∇h; (b) surface normal approximated. For the computations, $\phi(r) = r^7$, $n = 80$, and $m = 7$.

only a few scalar operations and an initial $O(N\log N)$ sorting process (to determine the vectors $\mathbf{n}_{AB}$, $\mathbf{n}_{BC}$, and $\mathbf{n}_{CA}$ in section 3.2).

For fixed $m = 7$, $n = 80$, and $\phi(r) = r^7$, the total cost in computing the RBF quadrature weights over S implicitly defined by (33) is illustrated in figure 22. Since the spherical quadrature method and the Chapter III quadrature method differ only by the computation of the projection origin, it is not surprising that they exhibit similar computational costs. Whether using the exact gradient or approximate to it, the Chapter III quadrature method is computationally inexpensive at $O(N)$ operations and $O(N)$ memory usage. The memory usage plot exhibits no apparent difference between the two, while the computation time plot exhibits a constant displacement. This displacement comes from computing the gradient approximation. While parallel scaling was not tested for the method in Chapter III, it is expected parallel scalability with number of cores will be observed similar to the spherical quadrature plot in figure 8c.

In each of the error plots, the error is large for N less than roughly $10^{2.6}$. This is because mesh sizes that are too large generate too few points ($N \approx 10^{2.6} \approx 400$) such that the requirement for Gnomonomic projection ($\mathbf{x}-\mathbf{x}_O$ within 90° of $\mathbf{n}_{ABC}$) is often violated for $n = 80$. Meanwhile, increases in error past $N \approx 10^{4.5} \approx 31,500$ (in the case of the approximate surface normal) are indicative of numerical approximation errors when computing derivatives for (32). The deliberation for the increase in error is left for future consideration.



(a) (b)

Figure 22. Timing results for computation of the RBF quadrature weights for the surface area of S implicitly defined by (33) as applied in section 4.2. The quadrature method has rate $O(N)$ for computation cost and memory usage for at least millions of nodes. An additional computational cost is observed when computing the approximation to the gradient. These plots were generated in machines with dual Intel Xeon E5-2687W 3.1 GHz, 8-core processors.

V. Conclusion

5.1 Future Considerations

With such rich areas of applicability, this thesis infers many avenues for future consideration. The work here uses only monomial radial basis functions, whereas infinitely smooth RBFs have more robust convergence properties [7]. Therefore, it is important to consider the stable use of these RBFs in the future, as was done for interpolation in [13, 14, 48, 49, 50].These infinitely smooth RBFs include the shape parameter ε that has been discussed throughout the literature [49, 51, 52]. Using different RBFs, and analyzing how ε affects the results would provide intriguing future work.

Decreasing computation time when generating either $\mathcal{S}_N$ or $\mathcal{T}$ is a separate but important issue. Using a less costly version of **distmeshsurface** would help with computing these. Furthermore, **distmeshsurface** does not generate $\mathcal{S}_N$ and $\mathcal{T}$ for a surface defined by parameterized coordinates $x(u, v)$, $y(u, v)$, and $z(u, v)$. Developing an algorithm that does so could be a significant contribution.

When analyzing the numerical approximation for errors for an approximated surface normal (see section 4.2), there is an increase in error past $N \approx 10^{4.5} \approx 31,500$. Deliberating the source of this is an important avenue for future research.

Other future considerations include non-smooth (jagged) or non-closed surfaces. Non-smooth surfaces pose an issue when approximating the derivative in use with (32). Furthermore, the quadrature method from this thesis requires the projection of nearest neighbors, whereby cusps in the surface may cause difficulty. Non-closed surfaces present a similar issue when using points near the surface boundary. Approximating the integral of a scalar function f over a non-smooth or non-closed surface would indeed be a challenging but significant endeavor.

5.2 Concluding Remarks

In the case of approximating surface integrals on the sphere, the convergence rate of the method in Chapter II was determined to be on the order of $O(N^{-3.5})$ for quasi-uniform node sets. Therefore, the spherical quadrature method from Chapter II yields accurate results with fast convergence. The results of Chapter IV showed simlar convergence for smooth, closed surfaces at still low computational cost. However, the computation for generating the quadrature nodes $\mathcal{S}_N$ and triangulation $\mathcal{T}$ via **distmeshsurface** took an intensive amount of time. It is only after these nodes and triangulation are generated that the results in Chapter IV can be interpreted.

The method for using an approximation to the surface normal is very robust in the sense that the integral of a scalar function f should be able to be approximated over any smooth closed surface S regardless of a (un)known parameterization. As the accuracy of the approximation attains levels of approximately 10^{-10}, this can be considered an excellent approximation. Furthermore, the convergence rate remained the same at $O(N^{-3.5})$. As the method discussed in Chapter III allows the surface to be on scattered grids, it can be used for a wide range of applications. Depending on the surface, however, the user may need to acquire a large number of samples to achieve a desired accuracy.

Bibliography

1. R. L. Hardy, "Theory and Applications of the Multiquadric-Biharmonic Method," *Comput. Math. Appl.*, vol. 19, no. 8-9, pp. 163–208, 1990.

2. J. A. Reeger and B. Fornberg, "Numerical Quadrature over the Surface of a Sphere," *Stud. Appl. Math.*, 2016. DOI: 10.1111/sapm.12106.

3. Z. P. Bažant and B. H. Oh, "Efficient Numerical Integration on the Surface of a Sphere," *Zamm-Z. Angew. Math. ME.*, vol. 66, pp. 37–49, 1986.

4. A. Sommariva and R. Womersley, "Integration by RBF over the Sphere," *Appl. Math. Report AMR05/17*, 2005.

5. R. S. Womersley and I. H. Sloan, "Interpolation and Cubature on the Sphere." http://web.maths.unsw.edu.au/ rsw/Sphere/, 2003/2007.

6. J. P. Bardhan, *Efficient Numerical Algorithms for Surface Formulations of Mathematical Models for Biomolecule Analysis and Design*. PhD thesis, Massachusetts Institute of Technology, 2006.

7. M. D. Buhmann, "Radial Basis Functions," *Acta Numerica*, vol. 9, pp. 1–38, 2000.

8. J. C. Mairhuber, "On Haar's Theorem Concerning Chebychev Approximation Problems Having Unique Solutions," *P. Am. Math. Soc.*, vol. 7, no. 4, pp. 609–615, 1956.

9. B. Fornberg, E. Larsson, and N. Flyer, "Stable Computations with Gaussian Radial Basis Functions," *SIAM J. Sci. Comput.*, vol. 33, pp. 869–892, 2011.

10. B. Fornberg and J. Zuev, "The Runge Phenomenon and Spatially Variable Shape Parameters in RBF Interpolation," *Comput. Math. Appl.*, vol. 54, pp. 379–398, 2007.

11. S. Bochner, "Monotone Funktionen, Stieltjessche Integrale und Harmonische Analyse," *Math. Ann.*, vol. 108, pp. 378–410, 1933.

12. I. J. Schoenberg, "Metric Spaces and Positive Definite Functions," *T. Am. Math. Soc.*, vol. 44, pp. 522–536, Nov. 1938.

13. R. L. Hardy, "Multiquadric Equations of Topography and Other Irregular Surfaces," *J. Geophys. Res.*, vol. 76, pp. 1905–1915, March 1971.

14. E. J. Kansa, "Multiquadrics - A Scattered Data Approximation Scheme with Applications to Computational Fluid-Dynamics - I: Surface Appoximations and Partial Derivative Estimates," *Comput. Math. Appl.*, vol. 19, no. 8-9, pp. 127–145, 1990.

15. A. I. Tolstykh, "On Using RBF-Based Differencing Formulas for Unstructured and Mixed Structured-Unstructured Grid Calculations," in *Proceedings of the 16th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation*, p. 6, 2000. Lausanne, Switzerland.

16. J. G. Wang and G. R. Liu, "A Point Interpolation Meshless Method Based on Radial Basis Functions," *Int. J. Numer. Meth. Eng.*, vol. 54, no. 11, pp. 1623–1648, 2002.

17. G. B. Wright, *Radial Basis Function Interpolation: Numerical and Analytical Developments*. PhD thesis, University of Colorado at Boulder, 2003.

18. C. Shu, H. Ding, and K. S. Yeo, "Local Radial Basis Function-Based Differential Quadrature Method and its Application to Solve Two-Dimensional Incompressible Navier–Stokes Equations," *Comput. Method Appl. M.*, vol. 192, no. 7, pp. 941–954, 2003.

19. N. Flyer, G. B. Wright, and B. Fornberg, "Radial Basis Function-Generated Finite Differences: A Mesh-Free Method for Computational Geosciences," *Handbook of Geomathematics*, 2014.

20. P. Keast and J. Diaz, "Fully Symmetric Integration Formulas for the Surface of the Sphere in S Dimensions," *SIAM J. Numer. Anal.*, vol. 20, no. 2, pp. 406–419, 1983.

21. V. Lebedev, "Quadratures on a Sphere," *USSR Comp. Math. Math.+*, vol. 16, pp. 1–23, 1976.

22. A. Stroud, *Approximate Calculation of Multiple Integrals*. Prentice-Hall, 1971. New Jersey, USA.

23. B. Fornberg and J. Martel, "On Spherical Harmonics Based Numerical Quadrature over the Surface of a Sphere," *Adv. Comput. Math.*, vol. 40, no. 5-6, pp. 1169–1184, 2014.

24. K. E. Atkinson, "Numerical Integration on the Sphere," *J. Aust. Math. Soc. B*, vol. 23, pp. 332–347, 1982.

25. 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.*, vol. 127, pp. 57–92, 2014.

26. D. Chien, "Numerical Evaluation of Surface Integrals in Three Dimensions," *Math. Comput.*, vol. 64, no. 210, pp. 727–743, 1995.

27. A. Sommariva and M. Vianello, "Meshless Cubature by Green's Formula," *Appl. Math. Comput.*, vol. 183, pp. 1098–1107, 2006.

28. J. D'Elía, L. Battaglia, A. Cardona, and M. Storti, "Full Numerical Quadrature of Weakly Singular Double Surface Integrals in Galerkin Boundary Element Methods," *Int. J. Numer. Meth. Eng.*, vol. 27, no. 2, pp. 314–334, 2011.

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

30. A. González, "Measurement of Areas on a Sphere using Fibonacci and Latitude-Longitude Lattices," *Math. Geosci.*, vol. 42, no. 1, pp. 49–64, 2010.

31. D. P. Hardin and E. B. Saff, "Discretizing Manifolds via Minimum Energy Points," *Notices of the AMS*, vol. 51, no. 10, pp. 1186–1194, 2004.

32. S. V. Borodachov, D. P. Hardin, and E. B. Saff, "Low Complexity Methods for Discretizing Manifolds via Riesz Energy Minimization," *Found. Comput. Math.*, vol. 14, no. 6, pp. 1173–1208, 2014.

33. R. J. Renka, "Algorithm 772: STRIPACK: Delaunay Triangulation and Voronoi Diagram on the Surface of a Sphere," *ACM T. Math. Software*, vol. 23, pp. 416–434, Sept. 1997.

34. B. Delaunay, "Sur la Sphére Vide," *B. Acad. Sci. USSR*, vol. 7, pp. 793–800, 1934.

35. T. M. Liebling and L. Pournin, "Voronoi Diagrams and Delaunay Triangulations: Ubiquitous Siamese Twins," *Doc. Math. Extra Volume: Optimization Stores*, pp. 419–431, 2012.

36. J. P. Snyder, *Map Projections-A Working Manual*, vol. 1395. US Government Printing Office, 1987. Washington, D.C., USA.

37. J. A. Reeger, "Handout: 19 December 2013." Private communication, Dec. 2013.

38. J. A. Reeger and B. Fornberg, "Numerical Quadrature Over the Surface of a Sphere." Private communication, Nov. 2014.

39. J. Stewart, *Calculus: Early Transcendentals.* Brooks/Cole-Thomson Learning, 5 ed., 2003. Belmont, CA, USA.

40. H. Wendland, *Scattered Data Approximation*, vol. 17. Cambridge University Press, 2004. United Kingdom.

41. J. A. Reeger and B. Fornberg, "Numerical Quadrature Over the Surface of a Sphere." Unpublished Powerpoint, 2015.

42. J. A. Reeger, "Handout: 4 November 2015." Private communication, Nov. 2015.

43. P. Persson and G. Strang, "A Simple Mesh Generator in MATLAB," *SIAM Rev.*, vol. 46, pp. 329–345, June 2004.

44. J. A. Reeger, "Handout: 29 June 2015." Private communication, June 2015.

45. R. C. Yates, *Cassinian Curves*, pp. 8–11. J. W. Edwards, 1952. Ann Arbor, MI, USA.

46. D. J. Griffiths and R. College, *The Three-Dimensional Delta Function*, vol. 3, p. 50. Prentice Hall Upper, 1999. Saddle River, NJ, USA.

47. J. A. Reeger, "Handout: 28 January 2016." Private communication, Jan. 2016.

48. B. Fornberg, E. Lehto, and C. Powell, "Stable Calculation of Gaussian-Based RBF-FD Stencils," *Comput. Math. Appl.*, vol. 65, pp. 627–637, 2013.

49. B. Fornberg and G. Wright, "Stable Computation of Multiquadratic Interpolants for all Values of the Shape Parameter," *Comput. Math. Appl.*, vol. 48, pp. 853–867, 2004.

50. B. Fornberg and C. Piret, "A Stable Algorithm for Flat Radial Basis Functions on a Sphere," *SIAM J. Sci. Comput.*, vol. 30, pp. 60–80, 2007.

51. R. Schaback, *Comparisons of Radial Basis Function Interpolants*, ch. 18, pp. 293–305. World Scientific, 1993. Singapore.

52. T. A. Driscoll and B. Fornberg, "Interpolation in the Limit of Increasingly Flat Radial Basis Functions," *Comput. Math. Appl.*, vol. 43, pp. 413–422, 2002.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
24–03–2016	Master's Thesis	Sept 2014 — Mar 2016

4. TITLE AND SUBTITLE

Radial Basis Function Based Quadrature over Smooth Surfaces

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Watts, Maloupu L., Second Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENS)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENC-MS-16-M-003

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Statement A.
Approved for Public Release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The numerical approximation of definite integrals, or quadrature, often involves the construction of an interpolant of the integrand and subsequent integration of the interpolant. It is natural to rely on polynomial interpolants in the case of one dimension; however, extension of integration of polynomial interpolants to two or more dimensions can be costly and unstable. A method for computing surface integrals on the sphere is detailed in the literature (Reeger and Fornberg, *Studies in Applied Mathematics*, 2016). The method uses local radial basis function (RBF) interpolation to reduce computational complexity when generating quadrature weights for the particular node set. This thesis expands upon the same spherical quadrature method and applies it to an arbitrary smooth closed surface defined by a set of quadrature nodes and triangulation.

15. SUBJECT TERMS

Quadrature, Radial Basis Functions, RBF, Numerical Integration, Smooth Surface, Interpolation, Closed Surface

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Capt. Jonah A. Reeger, AFIT/ENC
U	U	U	UU	95	19b. TELEPHONE NUMBER *(include area code)* (937)255-3636x3320; jonah.reeger@afit.edu