Nodal Configurations and Voronoi Tessellations for Triangular Spectral Elements

by

Michael James Roth

B.A.Sc, University of Waterloo, 1994
M.A.Sc, University of Waterloo, 1997

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

in the School of Earth and Ocean Sciences

Supervisor: Dr. A. J. Weaver

Abstract

By combining the high-order accuracy of spectral expansions with the locality and geometric flexibility of finite elements, spectral elements are an attractive option for the next generation of numerical climate models. Crucial to their construction is the configuration of nodes in an element — casual placement leads to polynomial fits exhibiting Runge phenomena manifested by wild spatial oscillations. I provide high-order triangular elements suitable for incorporation into existing spectral element codes. Constructed from a variety of measures of optimality, these nodes possess the best interpolation error norms discovered to date.

Motivated by the need to accurately determine these error norms, I present an optimization method suitable for finding extrema in a triangle. It marries a branch and bound algorithm to a quadtree smoothing scheme. The resulting scheme is both robust and efficient, promising general applicability.

In order to qualitatively evaluate these nodal distributions, I introduce the concept of a Lagrangian Voronoi tessellation. This partitioning of the triangle illustrates the regions over which each node dominates. I argue that distant and disconnected regions are undesirable as they exhibit a non-physical influence.

Finally, I have discovered a link between point distributions in the simplex and on the hypersphere. Through a simple transformation, a distance metric is defined permitting the construction of Voronoi diagrams and the calculation of mesh norms.

Table of Contents

List of Tables

List of Figures

Acknowledgements

This work was supported by an Natural Sciences and Engineering Research Council of Canada Postgraduate Scholarship.

Chapter 1

Introduction

In this chapter, I review several aspects of spectral element methods, a class of algorithms particularly well-suited to the numerical solution of partial differential equations in geophysical flows; discuss the Runge phenomenon, a potential pitfall of polynomial interpolation; and outline subsequent chapters of this dissertation, my attempt to deal with this phenomenon.

1.1 Spectral Elements

Spectral or *hp* elements [38] are high-order finite elements. As per the finite element method, the computational domain is partitioned into a series of sub-domains or "elements" over which separate "basis" functions approximate the solution locally. These basis functions are then stitched together to approximate the global solution. Whereas classical finite elements are typically restricted to first or second order polynomial bases, spectral elements are generally of much higher order, with order often allowed to vary from element to element. Spectral elements exhibit "spectral convergence", are "unstructured" and "adaptive", and communicate "locally".

1.1.1 Spectral convergence

With *hp* finite elements, there are two pathways to convergence. If h is some element length scale and p the degree of the highest polynomial basis function, one can either decrease h by increasing the number of elements or increase p by adding more polynomial basis functions. For smooth functions, these two pathways do not result in equal convergence rates. The Taylor series truncation error in a spectral element method is of order $O((h/p)^p)$. Evidently, increasing p results in rapid or "spectral convergence" whereas decreasing h results in slower "algebraic convergence" [10]. However, order is meaningless in non-smooth or discontinuous regions where a Taylor series expansion is a poor representation. In short, in smooth regions, the pathway to convergence is through increasing polynomial degree whereas in discontinuous regions, the pathway

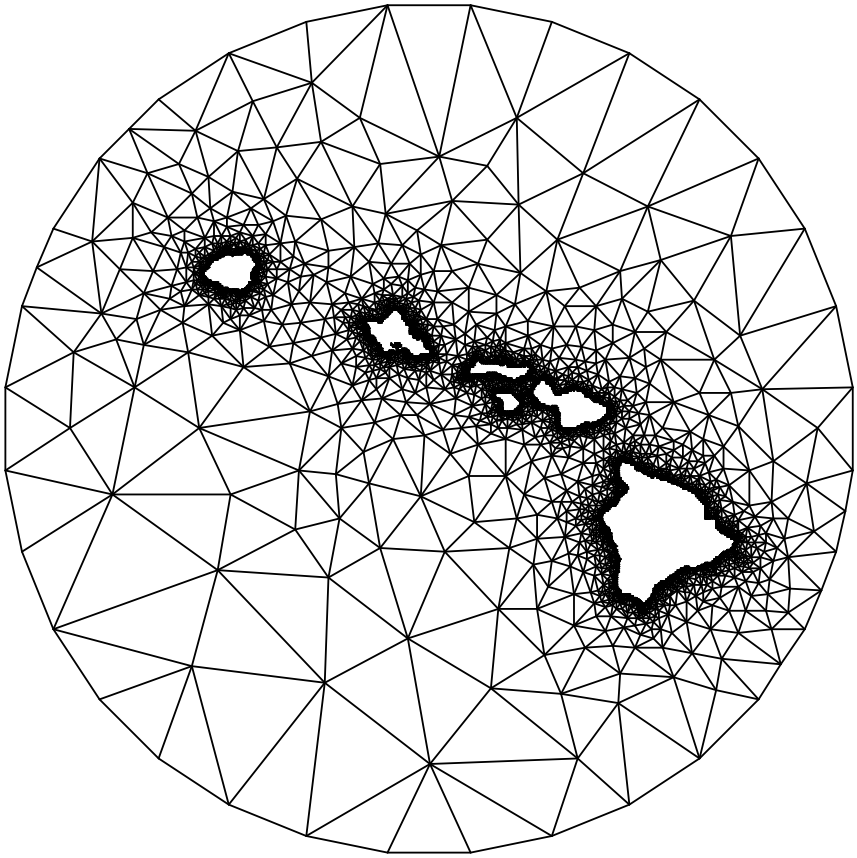


Figure 1.1: A finite element mesh of the waters around the Hawaiian Islands. The coastlines were extracted from the World Vector Shoreline database [59] and the mesh was generated using Triangle [55].

is through refinement of the mesh.

1.1.2 Unstructured, adaptive meshes

The argument for unstructured meshes is well-established and perhaps is more relevant to the study of geophysical flows than most other flows. While typical engineering flows are design-based and thus of relatively smooth geometries, geophysical flows are fractal in nature: each refinement of a coastline, seabed, or mountain-range results in a longer boundary. In addition, topographic features such as isolated sea mounts, ridges, mountains, islands, or straits can have profound impacts on weather and climate patterns with flow evolving over a large spectrum of length and time scales. In the current generation of general circulation models, many important local flow features which have global impacts are parameterized. Features such as deep convection sites are handled with a combination of vertical overturning mixing schemes and eddy parameterizations e.g. Gent–McWilliams [28]. Indeed, mesoscale eddies contain

a significant portion of the ocean's kinetic energy yet exist in only a small portion of the world's oceans. Mixing and dissipation is largely localized around sea mounts and mountain peaks [44] or near breaking gravity waves; a uniform mixing scheme parameterization is inadequate. In an unstructured, solution-adaptive finite element model, these features can be accommodated without resort to parameterization over much wider length scales.

It is well-known that the treatment of open boundary conditions is crucial to an accurate, realistic solution. If one is interested in examining a small region of the globe, say the North Atlantic, instead of treating this region in isolation, one can use a coarse global grid and refine in the region of interest.

1.1.3 Local communication

By their very nature, global spectral expansion methods are ill-suited to parallel computation. Interprocessor communication is in an "all-to-all" fashion. Spectral elements on the other hand benefit from dense local communication and sparse global communication; communication is simply between adjacent elements through shared nodes. Thus a trade-off exists between p and h — a decrease in h leads to greater "locality" while an increase in p leads to spectral convergence.

1.1.4 Element Shape

Hitherto, most spectral element implementations have relied on quadrilateral elements. The reasons for this are two-fold: quadrilaterals are currently computationally cheaper for a given accuracy and they typically fit man-made objects efficiently.

Quadrilaterals are built upon tensor products of one-dimensional polynomial expansions; therefore, typical calculations, such as determining derivatives, require fewer operations per node, generally meaning fewer "for" loops. However, Hesthaven and Teng [30] have demonstrated, by exploiting symmetries, that competitive algorithms in the triangle are possible. Together with the recent work of Pasquetti and Rapetti [48], which showed that comparable accuracy can be achieved with triangles and quadrilaterals, it seems premature to discount triangles on the basis of increased cost. It is entirely possible that a FFT-like algorithm, which utilizes the symmetry of a 2^p grid, can be found for the $\{1, 3, 6\}$-fold symmetries in the triangle.

As noted above, whereas engineering flows are typically design-based and thus easily meshed, geophysical flows have fractal geometries. Robust, efficient meshing algorithms exist for triangles [e.g. 55] whereas quadrilateral meshing can be difficult, if not impossible, especially for voids (islands). For example, a quadrilateral mesh is unattainable, without additional nodes, if the number of boundary edges is odd.

An additional undesirable property of the quadrilateral is the non-linearity of the transformation between the standard square element, in which space all computational work is done, and the general quadrilateral in (x, y) space. This non-affine transformation has consequences on the accuracy of integration and differentiation in highly-warped quadrilaterals as closed-form equations are unavailable.

1.1.5 Geophysical Applications

Early geophysical finite element models were typically diagnostic, barotropic implementations [e.g. 22], typically oceanic, for regional studies of tides and coastal currents. As there are no boundaries in the global atmosphere, the need for unstructured elements was not so acute and thus atmospheric models were typically based on global spectral expansions. More recently, there has been a progression of complexity in finite/spectral element models in the atmosphere and ocean. Starting with the shallow water equations, there have been implementations in the atmosphere [e.g. 26, 63] and the ocean [e.g. 18, 34]. The extension to the full three-dimensional primitive equations has been successful in both the atmosphere [25] and ocean [35].

In short, there is an increasing body of evidence showing spectral elements algorithms approaching the efficiency and robustness of classical finite difference schemes yet possessing all the attractiveness of unstructured meshes and spectral accuracy.

1.2 Runge Phenomenon

Runge [52] showed that polynomial fits on grids consisting of equally-spaced nodes may lead to unbounded and oscillatory interpolation of smooth, holomorphic functions. One such one-dimensional function exhibiting this "Runge phenomenon" is the Lorentzian or *averisera*[1] of Agnesi,

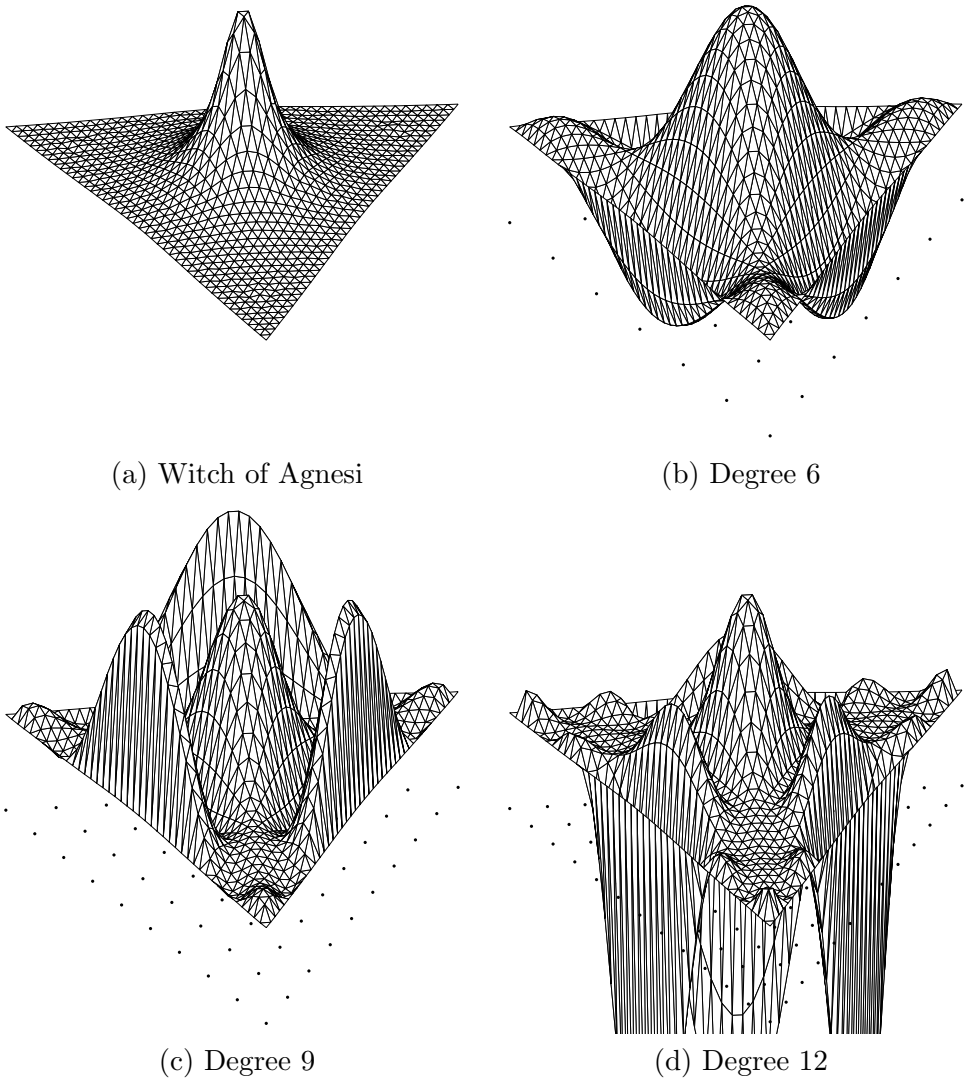
$$f(x) = \frac{1}{1 + 25x^2}. \tag{1.1}$$

We can approximate this function by spacing $p + 1$ nodes evenly across the interval $[-1, +1]$,

$$x_i = -1 + 2i/p, \qquad i = 0 \ldots p, \tag{1.2}$$

and fitting a pth degree polynomial through the exact values of $f(x)$ at these nodes. Unfortunately, as the degree p increases, we would see wild oscillations in this approximation near -1 and $+1$, suggesting that the regions near the ends are underresolved. Alternatively, if we evaluate our function at the Gauss–Lobatto–Chebyshev

[1] *Averisera* or "versed sine curve" is often mistranslated from the Latin as "wife of the devil" or *avversiera*, hence "witch" of Agnesi.



(a) Witch of Agnesi (b) Degree 6

(c) Degree 9 (d) Degree 12

Figure 1.2: Various degree isometric node approximations to the Runge function (1.3). The polynomial curve fits (b)–(d) do not converge to the witch of Agnesi (a).

nodes (§2.8), which are spaced $O(1/p^2)$ apart near the ends and $O(1/p)$ apart near the centre, our approximation will converge for any function which is analytic over the interval [10].

Unfortunately, there are no analogous analytical node distributions which solve the Runge phenomenon problem in the triangle. Consider a version of (1.1) suitable to the triangle and defined in terms of barycentric coordinates (§2.1),

$$f(x,y,z) = \frac{1}{1+100\left[\left(x-\frac{1}{3}\right)^2+\left(y-\frac{1}{3}\right)^2+\left(z-\frac{1}{3}\right)^2\right]}. \tag{1.3}$$

As with the one-dimensional case, if we fit a p-degree polynomial to (1.3) over a

uniformly-spaced, or "isometric" (§2.9), set of nodes, we again witness the Runge phenomenon in the form of wild oscillations near the triangle edges (see Figure 1.2).

1.3 Dissertation Outline

Throughout this dissertation I shall be interested in the optimal distribution of nodes in a d-dimensional simplex, with special focus on the triangle, in order to eliminate the Runge phenomenon. In doing so I have developed several new numerical algorithms and geometric constructs.

In Chapter 2, I review the basics of polynomial interpolation and integration in the triangle and establish notation for later chapters.

In Chapter 3, I present a new global optimization algorithm for determining the extrema in a triangle. Though developed specifically for the calculation of the Lebesgue norm of Lagrangian interpolation — the measure used throughout as an indicator of optimality — the method has general applicability.

In Chapters 4–6, I provide nodal point distributions based on optimizing the Fekete, Chen–Babuška, and Lebesgue measures, respectively. The resulting distributions represent the best configurations, based on Lebesgue norm, found to date.

In Chapter 7, I introduce the concept of a Lagrangian Voronoi tessellation to visually evaluate the spatial influence of nodes. In particular, I argue that the Lagrangians are suitable as a valid proxy for the Euclidean distance in ordinary Voronoi diagrams.

In Chapter 8, I introduce a transformation between the simplex and the hypersphere that allows the construction of a Voronoi tessellation and various mesh norms suitable for Lagrangian interpolation in simplicial spectral elements.

In Chapter 9, I summarize the main contributions of this dissertation and revisit the Runge phenomenon for the nodal distributions discovered.

Chapter 2

Interpolation Concepts

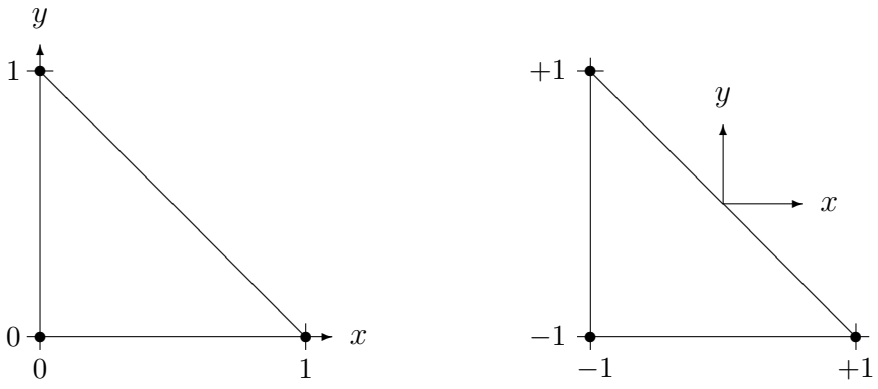
This chapter collects some well-known facts regarding polynomial interpolation in the triangle plus establishes some notation for later chapters.

2.1 Triangle Geometry

Each triangle in a mesh of triangles (e.g. Figure 1.1) may be transformed into a standard, reference triangle. All necessary derivatives and integrals may be computed in this standard triangle and then transformed back to the mesh. Here I consider the standard triangle (see Figure 2.1a) defined in terms of barycentric coordinates

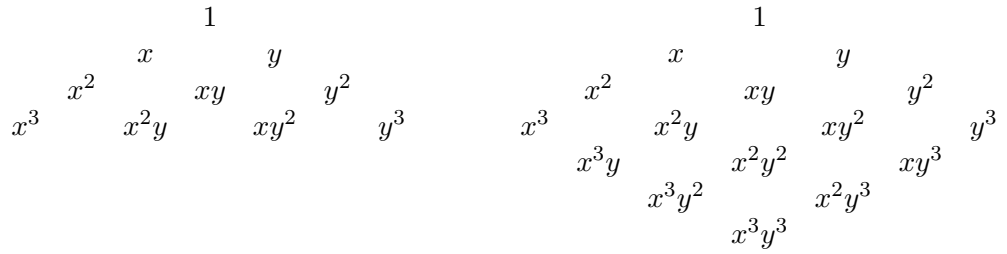
$$\{(x, y, z) \mid x, y, z \geq 0,\ x + y + z = 1\}. \tag{2.1}$$

The transformation is affine — points that are co-linear before transformation remain co-linear after transformation. In contrast, due to the non-linearity of its transforma-



(a) Barycentric (b) Koornwinder

Figure 2.1: I define two triangle geometries: (a) a unit standard triangle formed from the projection of the barycentric plane $x + y + z = 1$ onto the xy-plane and (b) a larger triangle for use with the Koornwinder basis.



(a) Triangle truncation (b) Diamond truncation

Figure 2.2: The monomials remaining from truncations of Pascal's triangle.

tion, the quadrilateral element does not share this property .

2.2 Koornwinder Basis

Interpolation requires the approximation of an unknown function f by a weighted sum of known functions,

$$f \approx \sum_{i=1}^{n} a_i b_i, \tag{2.2}$$

where a_i are the n spectral/expansion coefficients and $b_i(x, y)$ are the n corresponding basis functions. Physically, the spectral coefficients represent the relative importance of each basis function. Given a set of basis functions, the interpolation problem is to determine the coefficients which best approximate the unknown function. In non-periodic domains like the triangle, the usual set of bases are various truncations of monomials (see Figure 2.2).

In the triangle, the truncation is triangular, spanned by the monomials $x^i y^j$ with integers $\{i, j \geq 0, i + j \leq p\}$ where p is the maximum polynomial degree of the monomial. This truncation requires

$$n = \binom{n+2}{2} = \frac{1}{2}(p+1)(p+2) \tag{2.3}$$

monomials and is said to be complete. In contrast, quadrilateral elements typically use a diamond truncation of monomials $\{0 \leq i, j \leq p\}$ requiring $n = (p+1)^2$ monomials — extra monomials over and above what are required for a complete polynomial of degree p. Complete polynomials are preferred as they eliminate the impact of element orientation on the approximation [57], assuming that all triangle symmetries are respected as discussed below.

As is well-known, direct use of the monomials in Figure 2.2 as a basis is unwise. As

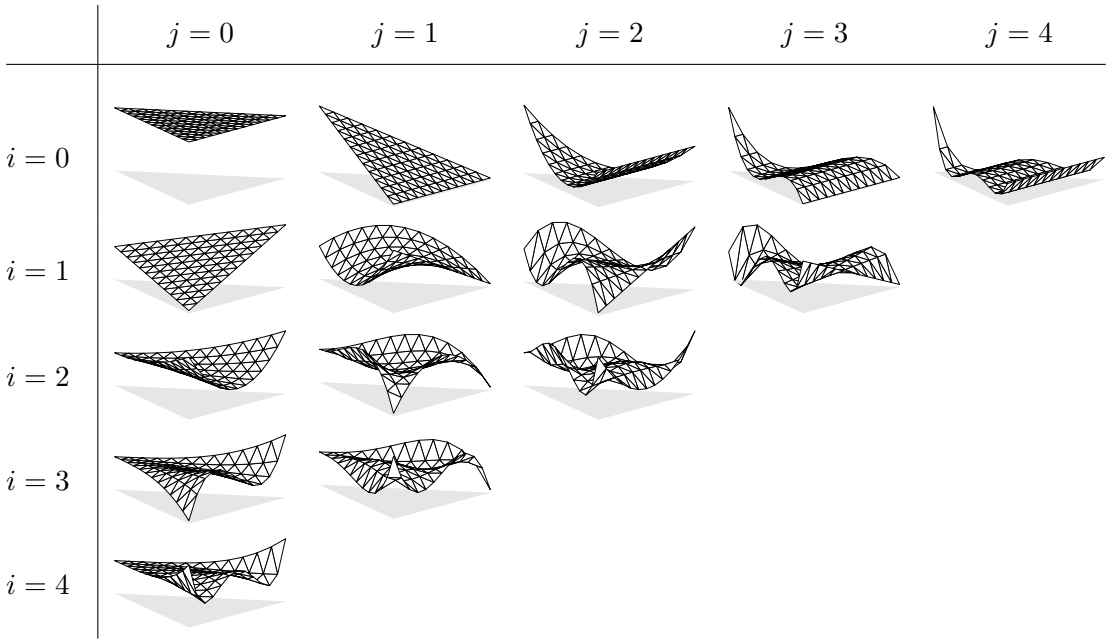


Figure 2.3: Koornwinder polynomial basis up to and including degree 4.

the degree p increases, each additional line of Pascal's triangle explores space already well-represented by previous bases — in other words, the monomial bases are non-orthogonal. However, the Koornwinder basis [17, 40, 47], a triangular analogue of the one-dimensional Legendre polynomials [32], provides an orthogonal combination of the above monomials. Defined over an extended right triangle (see Figure 2.1b), they are of the form

$$b_{i,j}(x, y) = \lambda_{i,j} P_i^{(0,0)}\left(\frac{2x + y + 1}{1 - y}\right)\left(\frac{1 - y}{2}\right)^i P_j^{(2i+1,0)}(y), \tag{2.4}$$

where $P_n^{(\alpha,\beta)}(x)$ represents the n-th order Jacobi polynomial [32] evaluated at x. The normalizing factor

$$\lambda_{i,j} = \sqrt{\frac{(2i + 1)(i + j + 1)}{2}} \tag{2.5}$$

yields an orthonormal set of polynomials such that

$$\int_{\triangle} b_i b_j \, \mathrm{d}A = \delta_{ij}. \tag{2.6}$$

In Figure 2.3, I sketch the $n = 15$ Koornwinder polynomials required by a $p = 4$ degree triangular truncation of the monomials.

2.3 Vandermonde Matrix

The evaluation of the n basis functions at n distinct nodes (x_j, y_j) yields the $n \times n$ Vandermonde matrix

$$V_{ij} = b_i(x_j, y_j) = \begin{bmatrix} b_1(x_1, y_1) & b_1(x_2, y_2) & \dots & b_1(x_n, y_n) \\ b_2(x_1, y_1) & b_2(x_2, y_2) & \dots & b_2(x_n, y_n) \\ \vdots & \vdots & \ddots & \vdots \\ b_n(x_1, y_1) & b_n(x_2, y_2) & \dots & b_n(x_n, y_n) \end{bmatrix}, \tag{2.7}$$

so-called in recognition of the fact that, in one dimension and with basis functions x^i, its determinant is the well-known Vandermonde determinant.

If the function values are known at the n nodes, $f_i = f(x_i, y_i)$, the spectral coefficients may be found through n applications of (2.2) and subsequent inversion of the resulting system

$$V_{ji} a_j = f_i. \tag{2.8}$$

This inversion is only possible if the determinant of the Vandermonde matrix is non-zero. Such a system is said to be fundamental. The determinant is a q-degree polynomial in the n nodes $\{x_j, y_j\}$, where

$$q = \frac{1}{3}(p+1)(p+2)(p+3). \tag{2.9}$$

2.4 Lagrangians

Alternatively, an unknown function can be approximated as a weighted sum of the nodal function values,

$$f \approx \sum_{i=1}^{n} \ell_i f_i, \tag{2.10}$$

where $\ell_i(x, y)$ are, interchangeably, the "Lagrange interpolating polynomials", "cardinal", or "shape" functions, hereafter simply the "Lagrangians", and are uniquely defined by the system

$$\ell_j(x_i, y_i) = \delta_{ij}. \tag{2.11}$$

A Lagrangian ℓ_i resembles an auto-correlation function, providing a measure of the spatial influence of node i throughout the triangle. Ideally, Lagrangians should act as discrete delta functions with influence decreasing with distance. Indeed, in the limit as $n \to \infty$, the Lagrangians should approach the Dirac limit.

The Lagrangians and the basis functions are linked through the Vandermonde

matrix,

$$V_{ij}\ell_j = b_i. \tag{2.12}$$

2.5 Integration of Products

Finite element methods are mainly concerned with the accurate integration of products of polynomials. Consider two functions f and g and their Lagrangian representation (2.10). If we integrate the product of these representations over the triangle we have

$$\int_\triangle fg \,\mathrm{d}A = \int_\triangle (\ell_i f_i)\,(\ell_j g_j)\,\mathrm{d}A = f_i W_{ij} g_j, \tag{2.13}$$

where we have defined a matrix W_{ij} consisting of the integrated products of Lagrangians,

$$W_{ij} = \int_\triangle \ell_i \ell_j \,\mathrm{d}A. \tag{2.14}$$

The determination of W_{ij} is facilitated by the use of an orthonormal basis such as the Koornwinder basis of §2.2. If we insert our relationships between the Lagrangians and the basis functions from (2.12), $\ell_i = V_{ik}^{-1} b_k$ and $\ell_j = V_{jl}^{-1} b_l$, into the above we have

$$W_{ij} = \int_\triangle \left(V_{ik}^{-1} b_k\right)\left(V_{jl}^{-1} b_l\right)\,\mathrm{d}A. \tag{2.15}$$

We may carefully rearrange the indexing such that

$$W_{ij} = \int_\triangle \left(V_{ik}^{-1} b_k\right)\left(b_l V_{lj}^{-1}\right)\,\mathrm{d}A, \tag{2.16}$$

which, pulling the Vandermonde matrices out of the integration, becomes

$$W_{ij} = V_{ik}^{-1}\left\{\int_\triangle b_k b_l \,\mathrm{d}A\right\} V_{lj}^{-1}. \tag{2.17}$$

By orthogonality (2.6), the quantity in curly braces reduces to the identity matrix so that

$$W_{ij} = V_{ik}^{-1}\delta_{kl} V_{lj}^{-1} = V_{ik}^{-1} V_{kj}^{-1} = (V_{kj} V_{ik})^{-1}. \tag{2.18}$$

Thus in matrix notation, the integration matrix is simply $W = (V^T V)^{-1}$. Note that this result is exact for polynomials in f and g each up to degree p.

We also may calculate the single integral,

$$\int_\triangle f \,\mathrm{d}A = f_i w_i, \tag{2.19}$$

where w_i is a vector of integration weights consisting of the row sums of W_{ij}. Analogous to (2.14), these weights represent the integrated Lagrangians,

$$w_i = \int_{\triangle} \ell_i \, \mathrm{d}A. \tag{2.20}$$

Negative weights would call into question the ability of the Lagrangians to accurately represent discrete delta functions. From a physical point of view, for a given node i, increasing f_i should increase the integral; however, a negative weight would decrease the integral.

It should also be noted that more efficient quadrature rules exist which require fewer nodal evaluations for a given degree [42]. However, these rules are limited to integration; corresponding interpolation or differentiation rules are unavailable.

2.6 Lebesgue Norm

The 1-norm of the vector of Lagrangians (see §2.4),

$$\lambda(x, y) = \sum_i |\ell_i(x, y)|, \tag{2.21}$$

defines the Lebesgue function (e.g. see Figure 3.1). This function is unity at the nodes and reaches maximal values where nodal coverage is poor at "holes" between nodes.

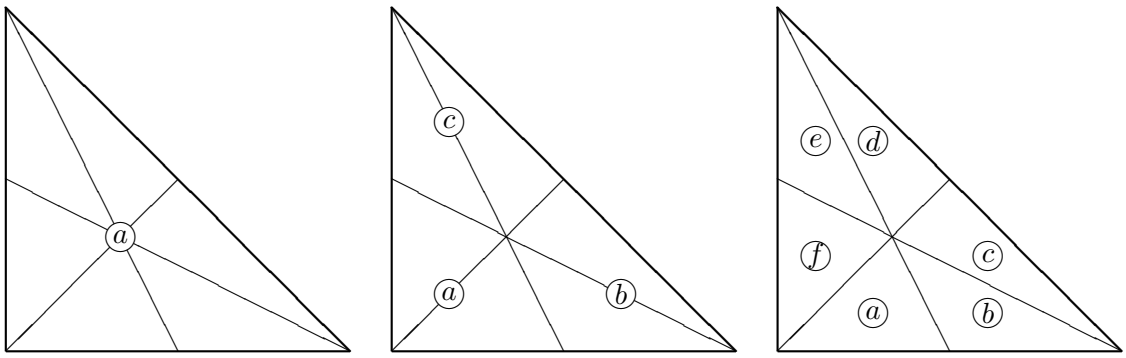
The infinity norm of λ over the triangle is termed the Lebesgue norm and here denoted Λ. This constant is useful as a measure of the quality of interpolation as it provides a conservative bound on the error [8],

$$\|f - g\|_\infty \leq (1 + \Lambda) \|f - h\|_\infty , \tag{2.22}$$

where f is a continuous function to be approximated, g is our unique approximation to f found by fitting a pth degree polynomial through n points such that $g(x_i, y_i) = f(x_i, y_i)$, and h is the best possible polynomial approximation to f in the sense that $\|f - h\|_\infty$ is minimal. Additionally, the Lebesgue norm bounds the polynomial approximation in terms of the nodal function values [8],

$$\|g(x, y)\|_\infty \leq \Lambda \max_i |f_i|. \tag{2.23}$$

While the Lebesgue norm bounds the error, perhaps more importantly, the growth of the Lebesgue norm with degree determines the convergence. By (2.22), if Λ grows faster than $\|f - h\|_\infty$ dies away, uniform convergence to f will be unattainable.



(a) Centre (b) Median (c) Sextant

Figure 2.4: There are three symmetries inherent to the triangle.

Strictly speaking, Bloom *et al.* [7] show that we require

$$\lim_{p\to\infty} \Lambda^{1/p} = 1. \tag{2.24}$$

In one-dimension, the Lebesgue function is convex in the interval between adjacent nodes [11]. To find the Lebesgue norm, search each interval for the maximum and take the largest maximum. In two-dimensions, however, the determination of the Lebesgue norm is a difficult global maximization problem. Thankfully, due to symmetry (§2.7), we need search only one sextant (1/6) of the triangle. I present an algorithm for finding Λ in Chapter 3.

The Lebesgue norm is defined solely in terms of the Lagrangians which in turn are a function solely of the nodal positions, regardless of the basis functions. If one wishes to limit Λ, one must optimize the placement of the nodes. I present algorithms and results for several attempts in Chapters 4–6.

2.7 Triangle Symmetry

Each triangular finite element consists of n nodes corresponding to the required n basis functions. Thus, if we wish to optimize the locations (x_i, y_i) of these nodes, there are $2n$ degrees of freedom. As the growth in n with p is quadratic by (2.3), the optimization space quickly becomes intractable. Fortunately, to be rotationally invariant, the nodal configuration should respect the symmetries inherent to the triangle (see Figure 2.4). Imposing symmetry reduces the number of degrees of freedom from $2n$ to $\lfloor n/3 \rfloor$.

Consider a node with barycentric coordinates (x, y, z). To preserve symmetry, there must be all possible *unique* permutations of these coordinates. There are three possibilities:

p	n	n_1	n_3	n_6	d.o.f.	
0	1	1	0	0	0	←
1	3	0	1	0	1	←
2	6	0	2	0	2	←
		0	0	1	2	
3	10	1	3	0	3	
		1	1	1	3	←
4	15	0	5	0	5	
		0	3	1	5	←
		0	1	2	5	
5	21	0	7	0	7	
		0	5	1	7	
		0	3	2	7	←
		0	1	3	7	
6	28	1	9	0	9	
		1	7	1	9	
		1	5	2	9	
		1	3	3	9	←
		1	1	4	9	
7	36	0	12	0	12	
		0	10	1	12	
		0	8	2	12	
		0	6	3	12	
		0	4	4	12	←
		0	2	5	12	
		0	0	6	12	

p	n	n_1	n_3	n_6	d.o.f.	
8	45	0	15	0	15	
		0	13	1	15	
		0	11	2	15	
		0	9	3	15	
		0	7	4	15	
		0	5	5	15	←
		0	3	6	15	
		0	1	7	15	
9	55	1	18	0	18	
		1	16	1	18	
		1	14	2	18	
		1	12	3	18	
		1	10	4	18	
		1	8	5	18	
		1	6	6	18	
		1	4	7	18	←
		1	2	8	18	
		1	0	9	18	

Table 2.1: Possible symmetry configurations and degrees of freedom (d.o.f.) in the triangle where n is the total number of nodes and n_1, n_3, and n_6 are the number of 1-fold (centre), 3-fold (meridian), and 6-fold (sextant) symmetry nodes, respectively. The arrows indicate the best configuration.

(a) **Centre** ($x = y = z = 1/3$): one node fixed in the centre ⇒ zero degrees of freedom;

(b) **Median** ($x = y \neq z$), ($y = z \neq x$), or ($z = x \neq y$): three nodes located on the medians joining a vertex and the midpoint of the opposite edge ⇒ one degree of freedom;

(c) **Sextant** ($x \neq y \neq z$): six nodes located in the region bounded by two medians and an edge ⇒ two degrees of freedom.

If n_1, n_3, and n_6 represent the number of each type in the triangle, the integer equation

$$n_1 + 3n_3 + 6n_6 = n \tag{2.25}$$

must hold, which, for $p \geq 1$, there are multiple solutions (see Table 2.1). Notice that a centre node exists if p is divisible by 3. Also note, in investigating the various configurations for a given p, it was quickly discovered that, regardless of quality measure, the best configuration corresponded to placing

$$n1 + n3 = \left\lceil \frac{p+1}{2} \right\rceil \tag{2.26}$$

nodes along each median.

The reduction in degrees of freedom may be further improved. For the system to remain fundamental (§2.3) the number of nodes along the each edge must be $p + 1$. This requirement ensures that an edge Lagrangian, restricted to the edge, reduces to a one-dimensional polynomial of degree p, ensuring continuity between elements. This fact further reduces the degrees of freedom: a median edge node (3-fold symmetry) may be either the midpoint of an edge or a vertex and thus has no degrees of freedom; a sextant edge node (6-fold symmetry) may freely move only along the edge between a vertex and a midpoint and thus has a single degree of freedom.

2.8 One-dimensional Nodes

In one-dimension there are two useful nodal sets based on orthogonal polynomials which eliminate the Runge phenomenon. Gauss–Lobatto–Chebyshev points are the abscissa of the extrema of the pth degree Chebyshev polynomial [32],

$$T_p(x) = T_p(\cos\theta) = \cos(p\theta), \qquad x \in [-1, 1], \tag{2.27}$$

together with the two endpoints $\{-1, 1\}$. They are defined explicitly as

$$x_i = -\cos\left(i\pi/p\right), \qquad i = 0 \ldots p. \tag{2.28}$$

Gauss–Lobatto–Legendre points are defined in terms of the extrema of a pth degree Legendre polynomial [32], together with the endpoints $\{-1, 1\}$, and can be calculated from the $p + 1$ zeros of

$$\frac{\mathrm{d}^{(p-1)}}{\mathrm{d}x^{(p-1)}}\left((1+x)^p(1-x)^p\right) = 0, \qquad x \in [-1, 1]. \tag{2.29}$$

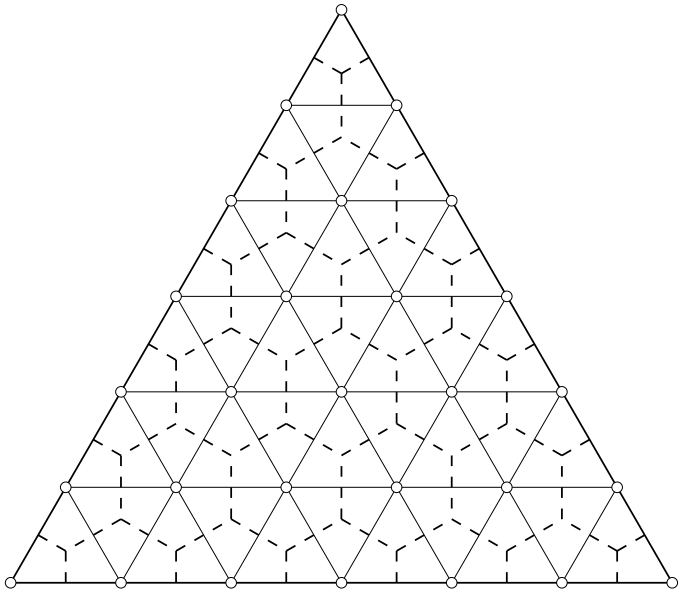


Figure 2.5: 6th degree isometric nodes. The solid lines indicate the Delaunay diagram while the dashed lines indicate the dual, the Voronoi diagram.

Both nodal sets exhibit a logarithmic increase of Lebesgue norm with degree thereby satisfying (2.24).

2.9 Isometric Nodes

Fejes Tóth [20] proved that the densest packing of circles in the plane has circle centres that form a triangular, uniform, isometric grid (Figure 2.5). The nodes in this grid may be described [57] by the barycentric triplet

$$(x, y, z) = (i, j, k)/p, \tag{2.30}$$

where the indices are the integers

$$\{(i, j, k) \mid i, j, k \geq 0, i + j + k = p\}. \tag{2.31}$$

As the Lagrangians may be defined explicitly, Bos [8] was able to derive an upper bound for the Lebesgue norm,

$$\Lambda \leq \binom{2p-1}{p}. \tag{2.32}$$

Bloom [6] determined the limiting behaviour

$$\lim_{p\to\infty} \Lambda^{1/p} = 2 \neq 1, \tag{2.33}$$

which, by (2.24), demonstrates the unsuitability of isometric nodes for interpolation.

Chapter 3

Dividing Triangles

Motivated by the need to accurately determine the Lebesgue norm of Lagrangian interpolation in a triangle, I present a new global optimization algorithm for finding the extrema of a continuous function in a triangle. Through definition of a new barycentric indexing scheme, the method subdivides a triangular quadtree based upon a Lipschitzian branch-and-bound heuristic coupled while requiring that the quadtree remain smooth.

3.1 Introduction

The Lebesgue norm Λ provides a bound on the error in Lagrangian interpolation (§2.6). However, the determination of this norm is a global maximization problem,

$$\Lambda = \max_{(x,y)} \lambda(x,y) = \max_{(x,y)} \sum_{i}^{n} \left| \ell_i(x,y) \right| , \tag{3.1}$$

where λ is the Lebesgue function (e.g. Figure 3.1) and ℓ_i are the set of Lagrangians (§2.4). Without exhaustive search, one cannot be certain of finding a global maximum.

Typically, the Lebesgue function is evaluated over a fine mesh of points and the largest result selected as the norm — a simple grid search. However, this method is undesirable as the cost for a given precision is prohibitive. Indeed, Lebesgue norms reported by other researchers are often understated, even for a relatively large number of function evaluations.

A better method is to initiate local Newton searches from a coarser grid and select the largest result as the norm. However, this method requires both a gradient and Hessian of the function value; here, both can be ill-behaved due to the many discontinuities in the Lebesgue function derivatives. Thus, while requiring fewer function evaluations than a fine mesh search, the result is not as robust.

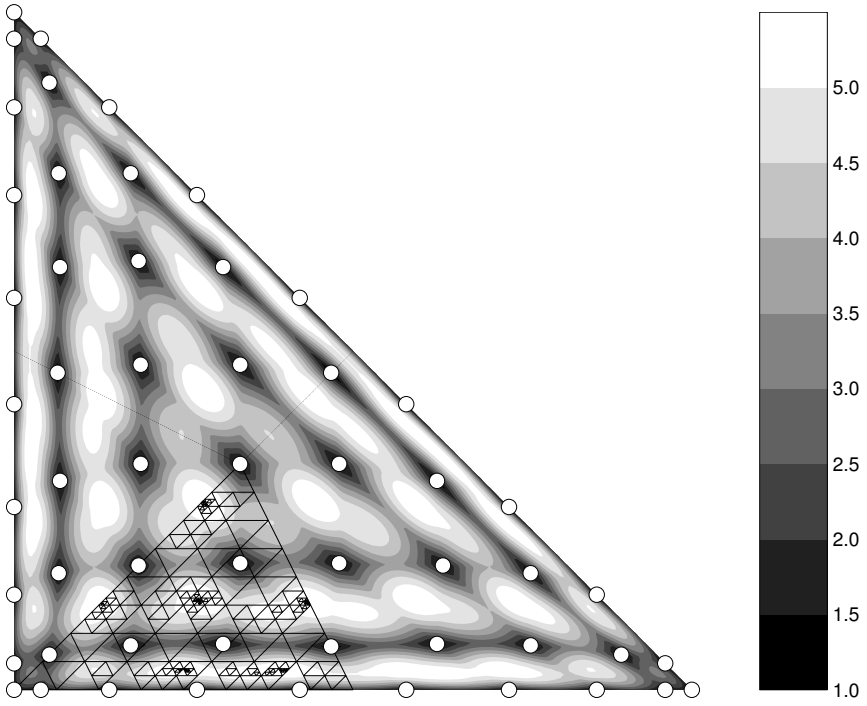


Figure 3.1: A contour plot of the Lebesgue function λ for a 9th degree polynomial. Nodes are indicated by the open circles. The Lebesgue norm is generally found at voids in the mesh. Note that, due to symmetry, we need only search 1/6 of the triangle.

I also require our result to be deterministic — for the same inputs, I seek the same output. This precludes such excellent, yet stochastic, algorithms as simulated annealing [33], particle swarm [39], or differential evolution [62].

Here, I modify a deterministic, Lipschitzian algorithm called DIRECT [36], short for DIviding RECTangles, which explores space by subdivision of an existing quadtree mesh based on a Pareto principle[1]. However, after a series of subdivisions, the quadtree mesh can become "rough" in the sense of an abrupt change in size between neighbouring triangles, preventing exploration of promising regions. Therefore, through a new indexing scheme, I insert an algorithm to find these rough triangles and add them to the subdivision queue. In the spirit of DIRECT, I call this new algorithm DITRI, for DIviding TRIangles.

3.2 Barycentric Indexing Scheme

The maximization algorithms described below explore space by splitting parent triangles into 4 identically-shaped child triangles — a standard triangular quadtree. We require an indexing scheme into this quadtree such that we may quickly determine

[1] Vital few, trivial many.

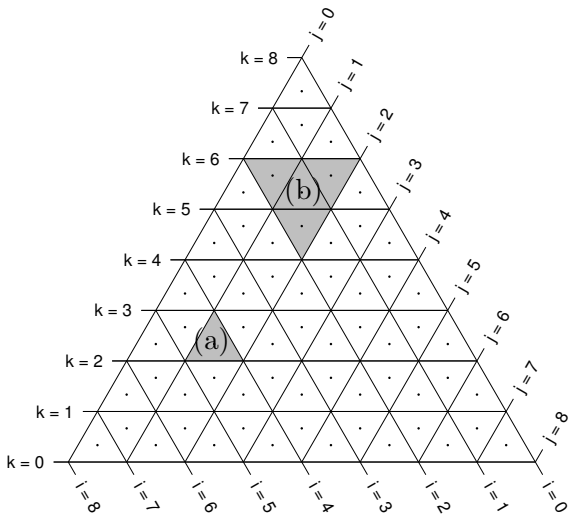


Figure 3.2: The (i, j, k) indexing notation for $p = 8$. The small triangle (a) is denoted $(4, 1, 2)$ with corresponding $\sigma = +1$ while the large triangle (b) is denoted $(2, 2, 6)$ with corresponding $\sigma = -2$. The magnitude of σ denotes the size of the cell while the sign indicates the rotation.

both the triangle centroid and its neighbours.

Consider isometric barycentric coordinates (§2.9). These coordinates describe the location of vertices of a triangular grid based on a triplet of indices (i, j, k). This representation, however, is vertex-based; we seek instead a triangle-based indexing scheme. We may switch to such a scheme by labelling each triangle with the triplet (i, j, k) with indices now indicating lines of constant i, j, and k which bound the triangle (Figure 3.2).

Now, however, $i + j + k \neq p$. Rather, the discrepancy between left and right hand side determines both the size and orientation of the triangle. I find it useful to define a quantity

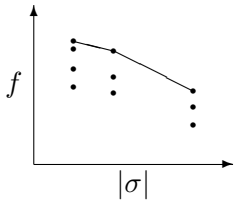
$$\sigma = p - (i + j + k). \tag{3.2}$$

The absolute value of σ indicates the size of the cell, with $|\sigma|/n$ the length of a triangle edge, while the sign of σ indicates the triangle orientation with $\triangle$ positive and $\triangledown$ negative. The depth of the triangle in the quadtree is simply

$$\delta = \log_2 \left(\frac{p}{|\sigma|} \right), \tag{3.3}$$

with $\delta = 0$ the root triangle.

1. Initialize root triangle and evaluate the function at the centroid.
2. For a specified number of iterations do:
   (a) Calculate the upper convex hull of the points $(|\sigma|, F)$ and queue corresponding triangles for subdivision.



   (b) Split each triangle in queue into four new child triangles and evaluate the function at the three new centroids.

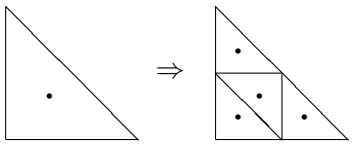


Figure 3.3: DITRI: A version of the DIRECT algorithm for determining the global maximum of a function in a triangle.

The three vertices in barycentric coordinates are

$$\begin{aligned}(x_1, y_1, z_1) &= (p-j-k, j, k)/p \\ (x_2, y_2, z_2) &= (i, p-i-k, k)/p \\ (x_3, y_3, z_3) &= (i, j, p-i-j)/p.\end{aligned} \tag{3.4}$$

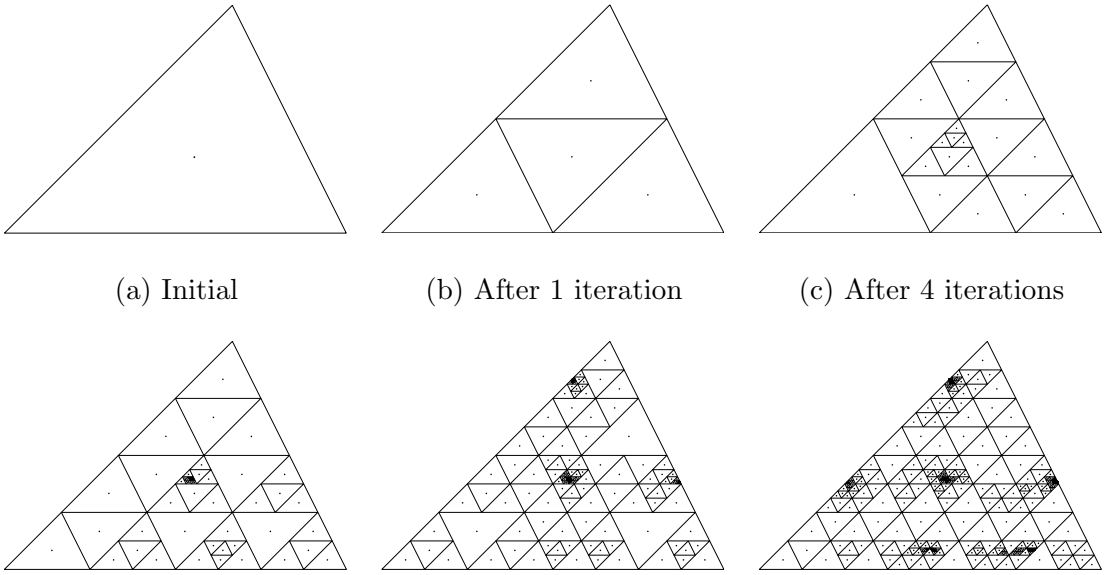
and thus the centroid of the triangle is simply the average of these vertices. Storage of the mesh degree p and an unordered list of (i, j, k) triplets are all that are required to describe the quadtree.

3.3 Lipschitzian Optimization

If we know the value of some maximum derivative in the triangle, the Lipschitz constant K, we may bound the change in some function f by

$$|\Delta f| \leq K \, |\Delta x| \, . \tag{3.5}$$

One could envision a maximization algorithm that, given an existing quadtree of triangles of varying sizes, evaluates the upper bound on the function expected in each triangle, identify the triangle(s) with the greatest upper bound, and split these



(a) Initial (b) After 1 iteration (c) After 4 iterations

(d) After 8 iterations (e) After 16 iterations (f) After 32 iterations

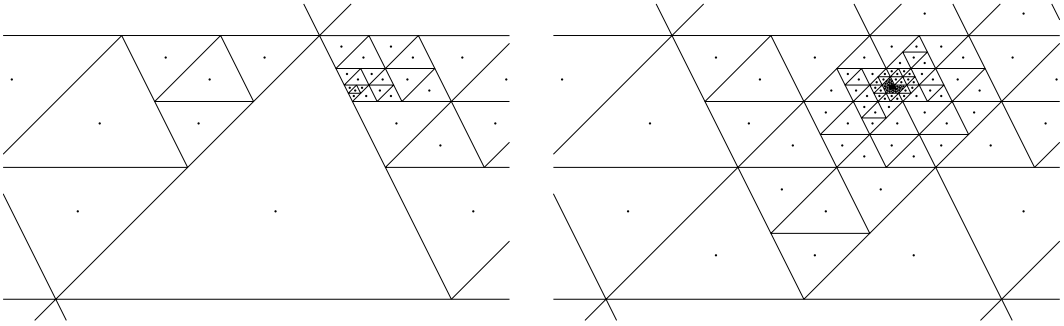
Figure 3.4: Several iterations of DITRI on the Lebesgue function of Figure 3.1. The function is evaluated at the locations indicated by the dots.

triangle(s). Through iteration, the algorithm would eventually converge to the global maximum. However, there are multiple problems with such an approach. First, the value of K is generally not known and is itself a global optimization problem. Secondly, even if K was known, it is generally so large than the largest triangles are generally chosen at each iteration making the method emphasize global search. Indeed, if $K \to \infty$, the method would reduce to exhaustive grid search [36].

Instead, the DIRECT [36] algorithm proceeds without explicit knowledge of the Lipschitz constant. The functional value of each triangle is plotted against its size, here $(|\sigma|, f)$, the convex hull is computed, and the triangles representing the upper right hull are split (see Figure 3.3). By selecting the best triangles of different levels, a good balance is struck between global and local search. In essence, multiple Lipschitz constants are explored simultaneously.

At each iteration the triangle(s) corresponding to the upper right corner of the convex hull of $(|\sigma|, f)$ are split and, eventually, every triangle of that size is split. Therefore, at some time in the future *every* current triangle is split, thereby ensuring that the algorithm will visit every point in the triangle and thus guaranteeing convergence to the global maximum.

Figure 3.4 shows a progression of iterations of the algorithm on the Lebesgue function of Figure 3.1.



(a) Rough (b) Smooth

Figure 3.5: A portion of Figure 3.4(f) magnified contrasting (a) a rough quadtree and (b) a smooth quadtree.

3.4 Quadtree Smoothing

After many iterations of the DITRI algorithm, the quadtree may become extremely "unbalanced", especially in difficult global maximization problems with many local maximums close to the global maximum. In Figure 3.5, I zoom in on a portion of Figure 3.4(f). If we examine the large central triangle in (a) and compare it to its neighbours we see an abrupt jump in size.

> **Definition**: The larger of two neighbours is said to be *rough* if the two neighbours differ in depth by more than one. A quadtree is said to be *smooth* if there are no rough triangles.

We introduce the simple heuristic that very large triangles close to very small triangles potentially contain the global maximum and should be split. Thus at each iteration of DITRI, we add an extra step of finding and splitting these large, rough triangles. This heuristic will generally introduce more function evaluations in a given iteration yet leads to greater efficiency through fewer iterations and, possibly, fewer total function evaluations.

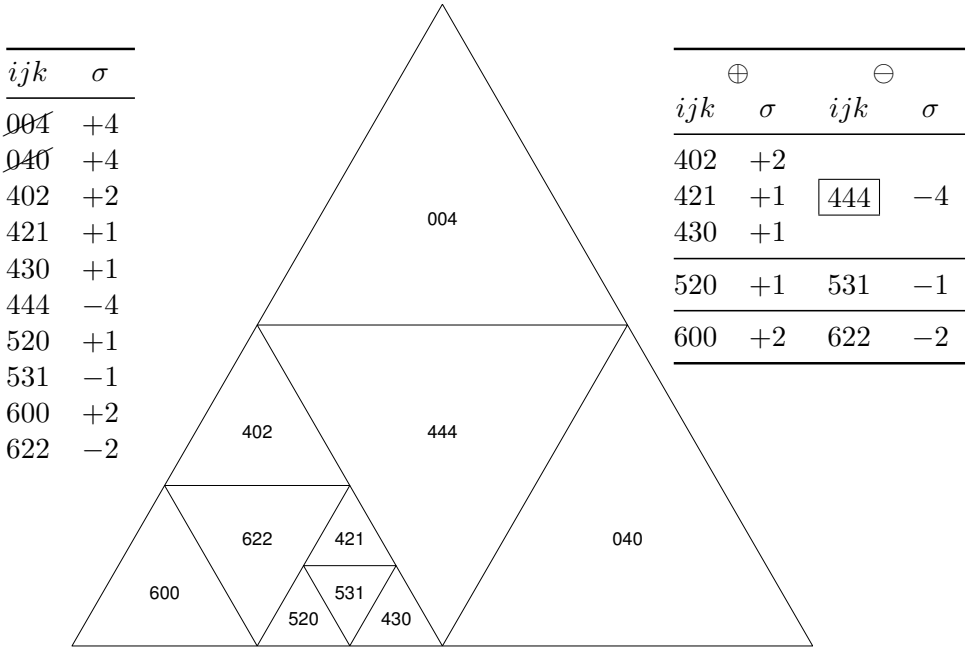
If we revisit Figure 3.5, we see the impact of the heuristic in the right panel. Now, the central triangle was split at a much earlier stage and we see the algorithm converging to a maximum just inside the original large triangle.

In Figure 3.6, I present pseudocode for the identification of rough triangles. This identification is made possible through the barycentric indexing scheme described earlier. In Figure 3.7, I provide a worked example.

The presented algorithm only finds the rough triangles along lines of constant i by passing the list of vertices $[i, j, k]$. To find the rough triangles along lines of constant

1. Sort rows of list of cells $[i, j, k]$.
2. Discard boundary cells with $i = 0$.
3. Calculate $\sigma = p - i - j - k$.
4. Split list into $\oplus$ or $\ominus$ based on the sign of σ.
5. Split $\oplus$ and $\ominus$ into matching subintervals by equating sums.
6. Determine the ratios between corresponding subintervals.
7. Label "rough" any subintervals with ratio greater than 2.

Figure 3.6: Rough triangle identification algorithm.



ijk	σ
~~004~~	+4
~~040~~	+4
402	+2
421	+1
430	+1
444	−4
520	+1
531	−1
600	+2
622	−2

$\oplus$		$\ominus$	
ijk	σ	ijk	σ
402	+2		
421	+1	444	−4
430	+1		
520	+1	531	−1
600	+2	622	−2

Figure 3.7: Example of the rough algorithm. The first matching subinterval $\{\oplus : 402, 421, 430\}$ and $\{\ominus : 444\}$ has a ratio of 3 : 1 and thus 444 is identified as "rough" and is queued for subdivision.

j and k, we simply pass lists of vertices as $[j, k, i]$ and $[k, i, j]$, respectively, to the algorithm.

3.5 Conclusion

The work presented in this chapter arose from a need to accurately and efficiently determine the maximum value of a function, in this case the Lebesgue norm, over a triangular domain. To remain focused on the goal of this thesis — the investigation

of polynomial interpolation in the simplex — no sustained effort was made to investigate the trade-off efficiency between the additional function evaluations required for smoothness at each iteration and the expected reduction in total number of function evaluations. Neither was this new algorithm exhaustively tested against other search algorithms. These investigations will be the root of future research.

If should be noted that the algorithm smoothing could conceivably be extended to determine the maximum of some continuous function over quite complex geometries, such as the complex coastlines and disconnected regions of Figure 1.1. Each triangle in the tessellation could act as the root of a quadtree. Triangles would be selected for subdivision based on a global convex hull of all root triangles and their children.

Chapter 4

Fekete Nodes

Interpolation on a uniformly-spaced, triangular grid of nodes leads to the Runge phenomenon, making this configuration unsuitable for spectral element methods. Fekete nodes, found by maximizing the determinant of the Vandermonde matrix of polynomial interpolation, provide an attractive alternative; by bounding the Lebesgue norm of interpolation, Fekete nodes ensure spectral convergence. Taylor *et al.* [64] presented an algorithm for determining Fekete node configurations in the triangle up to degree 19. Through a coupling of a modified version of their algorithm with a simple global perturbation method, I extend their results to degree 30. The resulting configurations are the best, measured by the Lebesgue norm, found to date.

4.1 Introduction

For a fixed set of basis functions Fekete [21] nodes are those which maximize the determinant of the Vandermonde matrix (§2.3),

$$\mathscr{F} = \arg\max_{\{x_i,y_i\}} \det V. \tag{4.1}$$

The determinant of a matrix is simply the hypervolume of the parallelepiped of the n vectors defined by the rows or columns. Thus, maximizing this determinant maximizes the volume spanned by the basis vectors, dispersing them maximally in space. Alternatively one speaks of this maximal volume as the extremal measure.

Analytically, the choice of basis is immaterial as long as it spans the full space of polynomials. Any polynomial bases may be written as a linear combination of any other polynomial bases; the Vandermonde matrices constructed from two such bases are said to be "similar" [27]. As such, a Vandermonde determinant of one basis will be a constant multiple of a Vandermonde determinant of another basis.

Numerically, however, a judicious choice of basis is crucial. A Cramer's rule

solution of (2.12) yields the Lagrangians

$$\ell_i(x,y) = \frac{\det\left[\boldsymbol{b}(x_1,y_1), \boldsymbol{b}(x_2,y_2), \ldots, \boldsymbol{b}(x,y), \ldots, \boldsymbol{b}(x_n,y_n)\right]}{\det\left[\boldsymbol{b}(x_1,y_1), \boldsymbol{b}(x_2,y_2), \ldots, \boldsymbol{b}(x_i,y_i), \ldots, \boldsymbol{b}(x_n,y_n)\right]}. \tag{4.2}$$

This representation shows that a well-conditioned Vandermonde matrix is essential to the success of the interpolation. Both the selection of an independent basis, preferably orthogonal, and the maximization of the determinant through optimizing nodal arrangements pushes the denominator far from singular leading to accurate inversion.

Additionally, an additional attractive property of Fekete nodes can be deduced from (4.2). If the Vandermonde determinant has truly been maximized, the numerator can never be larger than the denominator and thus we have a bound on our Lagrangians in $\triangle$,

$$|\ell_j(x,y)| \leq 1. \tag{4.3}$$

This bound on the Lagrangians also provides a bound on the Lebesgue constant. If all n Lagrangians are less than or equal to unity everywhere then their sum must be less than n,

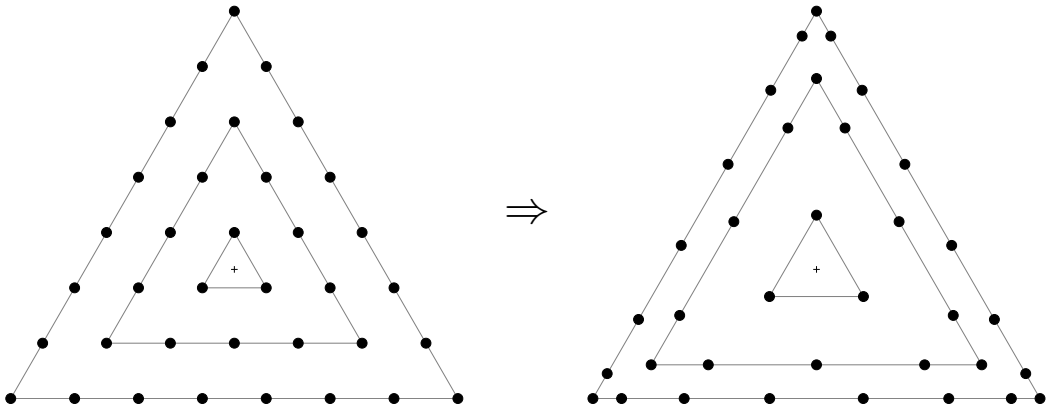
$$\Lambda = \sum_{i=1}^{n} |\ell_i| \leq n. \tag{4.4}$$

Even this however is a conservative bound as numerical simulations [64] indicate $\Lambda \leq p$, the polynomial degree.

Fejér [19] showed that, in one-dimension, the Fekete nodes are the Gauss–Lobatto–Legendre nodes (§2.8). Bos *et al.* [9] generalized this result by showing that the d-dimensional tensor-product of Gauss–Lobatto–Legendre nodes are Fekete nodes in the hypercube. Further, Reimer and Sündermann [49] and Sloan and Womersley [58] investigated numerically the equivalent Fekete problem on the sphere, the determination of extremal fundamental points. A fundamental result is the successful application of the Fekete framework in a variety of domains.

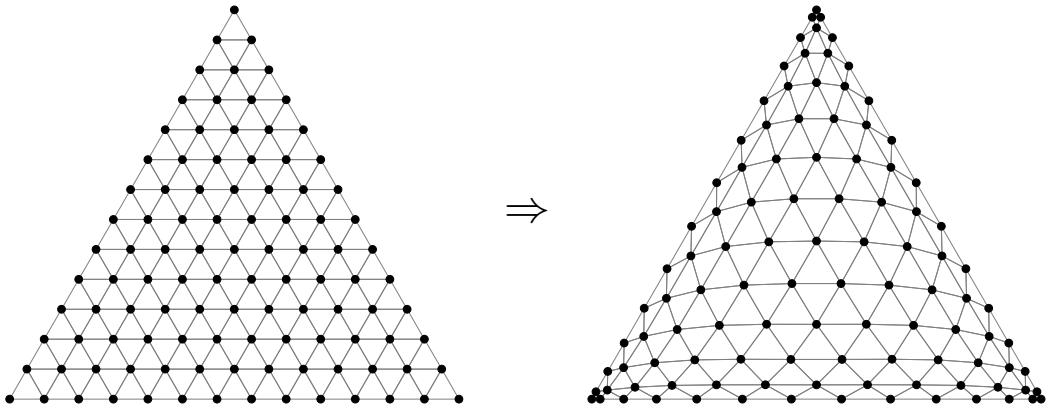
Bos [8] was the first to analyze the Fekete node problem in the triangle. He envisioned a series of concentric triangles (see Figure 4.1) of the same arrangement as a uniform grid yet with varying circumradii and then spaced nodes unevenly along each edge. By specifying an edge distribution of Gauss–Lobatto–Legendre nodes and optimizing the circumradii to maximize the determinant, Bos provided Fekete nodes which are exact for degrees up to 4 and approximate for degrees up to 7.

Chen and Babuška [13] presented results for $p \leq 13$ which improved upon the results of Bos. However, Fekete nodes with larger determinants and smaller Lebesgue



(a) Uniform (b) Bos

Figure 4.1: The 7th degree approximate Fekete nodes of Bos [8]. The circumradii of the concentric triangles are optimized to maximize the Vandermonde determinant. The nodes along each edge are distributed as one-dimensional Gauss–Lobatto–Legendre nodes.



(a) Uniform (b) Chen–Babuška

Figure 4.2: The 13th degree Fekete nodes of Chen and Babuška [13] found numerically from a uniform distribution.

constants have since been found. Though the details of their algorithm were not reported, based on the pattern, or connectivity, of nodes, their optimizations appear to have been initiated with uniformly-spaced nodes (see Figure 4.2). Subsequent results [64] have shown that this connectivity pattern does not extend for optimal Fekete nodes past degree 6.

Stieltjes [60, 61] discovered an interesting link between the logarithmic electrostatic energy E of n point charges and the Vandermonde determinant,

$$E = -\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}\log|x_i - x_j| = -\log\prod_{\substack{i,j\leq n\\ i\neq j}}|x_i - x_j| = -\log\det V, \tag{4.5}$$

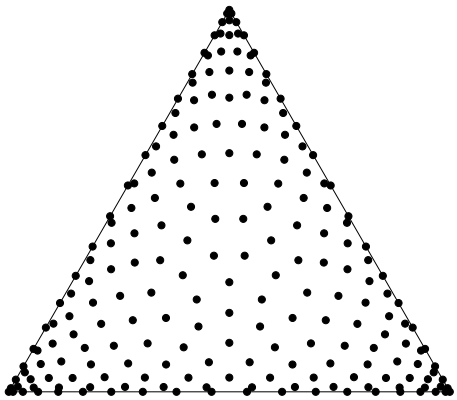


Figure 4.3: The degree 19 Fekete nodes of Taylor *et al.* [64].

where V is the classic one-dimensional Vandermonde matrix

$$V = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^n & x_2^n & \dots & x_n^n \end{bmatrix}. \tag{4.6}$$

Evidently, minimizing E is equivalent to maximizing the Vandermonde determinant; in one dimension, electrostatic nodes are equivalent to Fekete nodes. Hesthaven [29], inspired by this link, proposed the extension of the analogy to two dimensions, and higher, by introducing an electrostatic line charge along each edge of the triangle and minimizing an algebraic rather than logarithmic electrostatic potential,

$$\mathscr{E} = \arg \min_{\{x_j, y_j\}} E. \tag{4.7}$$

While not producing Fekete nodes in the triangle, for low degree ($p \leq 8$), this procedure produced the lowest Lebesgue constants at the time, albeit exhibiting an exponential growth rate in Λ. However, since his computed nodes share the same uniform connectivity pattern of Bos [8] and Chen and Babuška [13], better nodal sets might be found with alternative initial conditions.

Taylor *et al.* [64] have greatly extended our knowledge of the behaviour of Fekete nodes by computing Fekete nodes up to degree 19 (see Figure 4.3). Their elegant approach relies on the relationship between the Vandermonde determinant and the

Lagrangians (see Appendix C),

$$\frac{\partial}{\partial x_k} \ln \det V = \left. \frac{\partial \ell_k}{\partial x} \right|_k . \tag{4.8}$$

We thus see that, at a Fekete maximum, where the derivatives of the Vandermonde determinant with respect to the nodal positions are zero, the corresponding derivatives of the Lagrangians are zero — nodes sit atop the peaks of their respective Lagrangians. Their algorithm proceeds by moving the nodes toward the maximum of their respective Lagrangians along the directions of steepest ascent. As the Lagrangian topography changes with each movement, the algorithm is iterative in nature. In later iterations, higher derivatives were incorporated allowing a switch to Newton's method to speed convergence. By varying initial conditions, multiple local maximums were found and subsequently graded on their Lebesgue constants.

Baran [4] derived an extremal measure for the simplex, which for the triangle (2.1), yields

$$\rho = \frac{1}{\sqrt{xy(1-x-y)}} = \frac{1}{\sqrt{xyz}}. \tag{4.9}$$

I refer to this density as the Chebyshev density in recognition of the fact that, in one-dimension, it is equivalent, if shifted into the interval $[-1, 1]$, to the weighting function in the orthogonality integral of Chebyshev polynomials [32],

$$\int_{-1}^{+1} \frac{T_i(x) T_j(x)}{\sqrt{1-x}\sqrt{1+x}} \, \mathrm{d}x = \frac{1}{2} \pi \delta_{ij}. \tag{4.10}$$

It is conjectured [7] that points distributed to (4.9) will, in the limit $p \to \infty$, be the Fekete nodes. Unfortunately, distribution to a specified density in dimensions higher than one is non-trivial, if not ill-posed in the discrete sense. However, we may use this density to compute "random Chebyshev points" in the triangle (see Appendix A).

4.2 Newton–Raphson Perturbation Algorithm

Strictly speaking, the Fekete definition allows one unique nodal set: the global maximum of the Vandermonde determinant. However, locally maximal sets with smaller $\det V$ may have better Lebesgue norms or other interpolation qualities than the set representing the global maximum. For this reason, I search for nodal sets that are locally maximum in $\det V$ such that all points sit atop the peaks of their respective Lagrangians and also check that $\max_{(x,y)} |\ell_i| \leq 1$ throughout $\triangle$ for all nodes i. These local maximums will be called Fekete nodes.

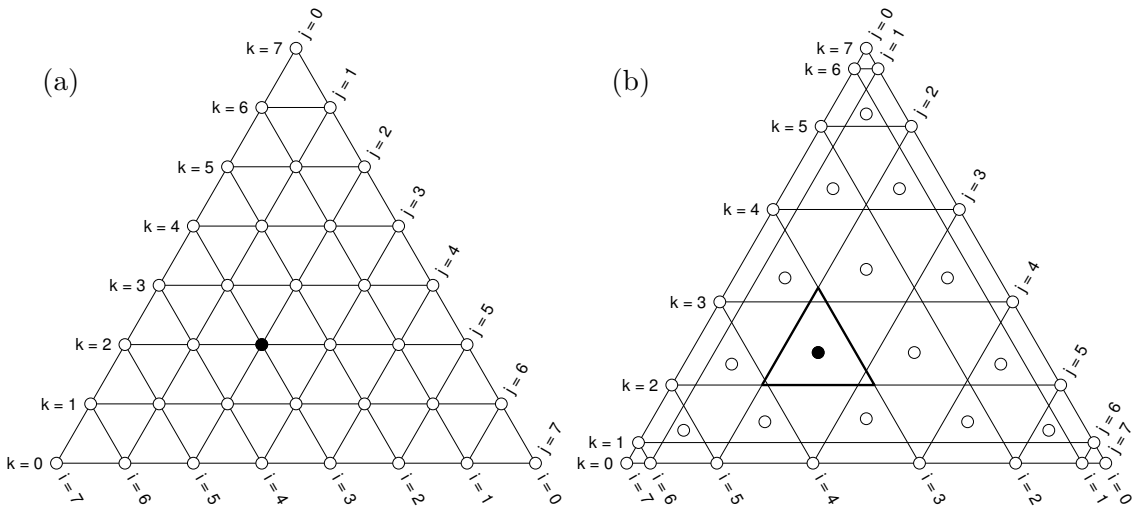


Figure 4.4: The uniformly-distributed points (a) in the triangle intersect along lines of constant i, j, and k. In (b), the lines joining the Gauss–Lobatto–Legendre points form the boundary of a triangle (shown in bold). We place a node at the centre of this triangle.

The Vandermonde determinant is a q-degree polynomial (§2.3), with $q \propto p^3$, in the n nodal coordinates $\{x_i, y_i\}$, with $n \propto p^2$. Thus the number of extrema grows extremely fast. That the maximization of such a polynomial is difficult is an understatement.

Inspired by progress in the minimization of the potential energy of Lennard–Jones clusters [e.g. 65], I combine a fast local search method *cum* Taylor *et al.* [64] with a simple global search perturbation method.

4.2.1 Initial Nodal Configuration

The algorithm requires an initial nodal configuration. While an isometric grid is possible, for higher degree simulations it became increasingly difficult to find an initial local maximum. This necessitated the development of an explicit nodal set. This set was derived from geometric considerations and is competitive with other configurations up to degree 10.

In Figure 4.4a, I plot the uniform, isometric points (§2.9). For this set of points, the lines of constant i, j, and k associated with a given node intersect. If, instead, the edge points are distributed in a Gauss–Lobatto–Legendre fashion (Figure 4.4b), the lines no longer intersect, but rather form the boundary of a triangle. If we now place a node at the centroid of the triangle thus formed, we are left with a distribution of points distributed with higher density near the edges and lower density near the centre. We call this distribution of points "pseudo–Legendre".

The points thus constructed can be explicitly defined. If $\{\xi_i\}$ is the set of one

p	4	5	6	7	8	9	10	11	12	13	14	15
(a)	2.7	3.6	4.2	4.9	5.9	6.8	7.9	7.9	8.5	9.3	10.0	10.0
(b)	2.7	3.1	3.9	4.7	5.9	7.4	9.8	12.9	17.8	24.5	34.7	49.6
(c)	3.5	5.5	8.7	14.3	24.0	40.9	70.9	124.5	221.4	397.7	720.7	1315.9

Table 4.1: Lebesgue norms of (a) Fekete, (b) pseudo-Legendre, and (c) Isometric nodes.

dimensional Gauss–Lobatto–Legendre (2.29) nodes both scaled and translated into the interval $[0, 1]$, the vertices of the triangle enclosing the node are

$$\begin{aligned}(x_1, y_1, z_1) &= \big(1 - \xi_j - \xi_k, \xi_j, \xi_k\big),\\ (x_2, y_2, z_2) &= \big(\xi_i, 1 - \xi_k - \xi_i, \xi_k\big),\\ (x_3, y_3, z_3) &= \big(\xi_i, \xi_j, 1 - \xi_i - \xi_j\big).\end{aligned} \tag{4.11}$$

The centroid of these points is then simply

$$(x, y, z) = \frac{1}{3}\Big[(x_1, y_1, z_1) + (x_2, y_2, z_2) + (x_3, y_3, z_3)\Big]. \tag{4.12}$$

The Lebesgue norm of pseudo–Legendre nodes grows at a faster rate than that of Fekete nodes and does not satisfy the Lebesgue limit of (2.24) but they are a marked improvement over isometric nodes (see Table 4.1); in fact, pseudo-Legendre nodes have lower Lebesgue norms than Fekete nodes for $p \leq 8$.

4.2.2 Local Search

Our local search algorithm is a recasting of the elegant algorithm of Taylor *et al.* [64]. Inspired by (4.8), we work with the Lagrangians without direct reference to the Vandermonde determinant. The Lagrangians are functions of both the nodal locations $\{x_j, y_j\}$ and position (x, y). We first perform a Taylor series of its gradient,

$$\begin{aligned}\frac{\partial \ell_i}{\partial x} &\approx \frac{\partial \ell_i}{\partial x}\bigg|_\circ + \frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial x}\bigg|_\circ \delta x + \frac{\partial}{\partial y}\frac{\partial \ell_i}{\partial x}\bigg|_\circ \delta y + \sum_j \frac{\partial}{\partial x_j}\frac{\partial \ell_i}{\partial x}\bigg|_\circ \delta x_j + \sum_j \frac{\partial}{\partial y_j}\frac{\partial \ell_i}{\partial x}\bigg|_\circ \delta y_j\\ \frac{\partial \ell_i}{\partial y} &\approx \frac{\partial \ell_i}{\partial y}\bigg|_\circ + \frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial y}\bigg|_\circ \delta x + \frac{\partial}{\partial y}\frac{\partial \ell_i}{\partial y}\bigg|_\circ \delta y + \sum_j \frac{\partial}{\partial x_j}\frac{\partial \ell_i}{\partial y}\bigg|_\circ \delta x_j + \sum_j \frac{\partial}{\partial y_j}\frac{\partial \ell_i}{\partial y}\bigg|_\circ \delta y_j\end{aligned} \tag{4.13}$$

where $|_\circ$ denotes quantities evaluated at the point of expansion, the current nodal locations. At each node we wish the gradient of its corresponding Lagrangian to be

zero, enforced by setting the left hand sides to zero,

$$\begin{aligned}
0 &\approx \left.\frac{\partial \ell_i}{\partial x}\right|_{\circ,i} + \left.\frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial x}\right|_{\circ,i} \delta x_i + \left.\frac{\partial}{\partial y}\frac{\partial \ell_i}{\partial x}\right|_{\circ,i} \delta y_i + \sum_j \left.\frac{\partial}{\partial x_j}\frac{\partial \ell_i}{\partial x}\right|_{\circ,i} \delta x_j + \sum_j \left.\frac{\partial}{\partial y_j}\frac{\partial \ell_i}{\partial x}\right|_{\circ,i} \delta y_j \\
0 &\approx \left.\frac{\partial \ell_i}{\partial y}\right|_{\circ,i} + \left.\frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial y}\right|_{\circ,i} \delta x_i + \left.\frac{\partial}{\partial y}\frac{\partial \ell_i}{\partial y}\right|_{\circ,i} \delta y_i + \sum_j \left.\frac{\partial}{\partial x_j}\frac{\partial \ell_i}{\partial y}\right|_{\circ,i} \delta x_j + \sum_j \left.\frac{\partial}{\partial y_j}\frac{\partial \ell_i}{\partial y}\right|_{\circ,i} \delta y_j
\end{aligned} \tag{4.14}$$

where $|_i$ denotes quantities evaluated at $(x, y) = (x_i, y_i)$. The second-derivative terms have a simple representation (see Appendix B)

$$\begin{aligned}
0 &\approx \left.\frac{\partial \ell_i}{\partial x}\right|_{i} + \left.\frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial x}\right|_{i} \delta x_i + \left.\frac{\partial}{\partial y}\frac{\partial \ell_i}{\partial x}\right|_{i} \delta y_i - \sum_j \left.\frac{\partial \ell_i}{\partial x}\right|_{j} \left.\frac{\partial \ell_j}{\partial x}\right|_{i} \delta x_j - \sum_j \left.\frac{\partial \ell_i}{\partial y}\right|_{j} \left.\frac{\partial \ell_j}{\partial x}\right|_{i} \delta y_j \\
0 &\approx \left.\frac{\partial \ell_i}{\partial y}\right|_{i} + \left.\frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial y}\right|_{i} \delta x_i + \left.\frac{\partial}{\partial y}\frac{\partial \ell_i}{\partial y}\right|_{i} \delta y_i - \sum_j \left.\frac{\partial \ell_i}{\partial x}\right|_{j} \left.\frac{\partial \ell_j}{\partial y}\right|_{i} \delta x_j - \sum_j \left.\frac{\partial \ell_i}{\partial y}\right|_{j} \left.\frac{\partial \ell_j}{\partial y}\right|_{i} \delta y_j
\end{aligned} \tag{4.15}$$

Physically, the first three terms on the right hand side represent a quadratic model of each Lagrangian, a hill which are we attempting to climb. The remaining terms reflect the fact that, as we climb, the topography changes.

We are left with a gradient–Hessian system of linear equations,

$$\mathbf{0} = \boldsymbol{g} + \mathsf{H}\,\boldsymbol{\delta x}. \tag{4.16}$$

With suitable boundary conditions, this system of equations may be solved for the nodal displacements $\{\delta x_j, \delta y_j\}$. By ensuring that boundary nodes remain on the boundary, the problem is well-posed.

However, it is useful to enforce symmetry on the system as discussed in §2.7. For nodes with $\{1, 3, 6\}$-fold symmetry, the corresponding equations above are replaced with relationships (Figure 4.5) either fixing the nodes in space or otherwise constraining their behaviour.

It is well-known that Newton's method will diverge when the quadratic model does not accurately represent the landscape or the current solution is far from a stationary point. We thus first attempt a full Newton step. If this step is greater than some threshold, we recalculate a step based on only the first three terms of (4.15), thereby freezing the Lagrangians in place over the step. As a threshold, we use the smallest distance between adjacent one-dimensional Gauss–Lobatto–Chebyshev (§2.8) nodes,

Centre (1-fold symmetry):

$$\begin{aligned}(a)\ \delta x_a &= 0\\ (b)\ \delta y_a &= 0\end{aligned} \tag{4.17}$$

Median (3-fold symmetry)

$$\begin{aligned}&(a)\ \delta x_a = \delta y_a \quad (c)\ \delta x_b = -\delta x_a - \delta y_a \quad (e)\ \delta x_c = \delta x_a\\ &(b)\ \delta x_a = 0 \qquad (d)\ \delta y_b = \delta x_a \qquad\quad (f)\ \delta y_c = -\delta x_a - \delta y_a\end{aligned} \tag{4.18}$$

Sextant (6-fold symmetry)

$$\begin{aligned}&(a)\ \delta y_a = 0 && (e)\ \delta x_c = -\delta x_a - \delta y_b && (i)\ \delta x_e = \delta y_a\\ &(b)\ \delta x_a = 0 && (f)\ \delta y_c = \delta x_a && (j)\ \delta y_e = -\delta x_a - \delta y_b\\ &(c)\ \delta x_b = -\delta x_a - \delta y_a && (g)\ \delta x_d = \delta x_a && (k)\ \delta x_f = \delta y_a\\ &(d)\ \delta y_b = \delta x_a && (h)\ \delta y_d = -\delta x_a - \delta y_b && (l)\ \delta y_f = \delta x_a\end{aligned} \tag{4.19}$$

Figure 4.5: Equations enforcing symmetry. (4.18b) is applied only to nodes at the vertices or edge midpoints. (4.19b) is applied only to nodes on the edges and further (4.19a) is applied only to edge nodes which are fixed, say to a Gauss–Lobatto–Legendre distribution.

$1 - \cos(\pi/p)$.

Note that this algorithm avoids all computation of the Vandermonde determinant, for which it is often difficult to maintain accuracy or avoid overflow or underflow [31], even with a good choice of basis.

The presented local algorithm is found to be largely independent of initial conditions and rapidly converges to machine epsilon within typically 5–8 iterations.

4.2.3 Edge Nodes

It was conjectured [8] and confirmed numerically [13, 64] that Fekete along triangle edges are distributed as one-dimensional Gauss–Lobatto–Legendre nodes (2.29). Here, we confirm this fact analytically.

Consider the set of nodes along the bottom edge of the triangle (Figure 2.1),

$$\mathcal{E} = \{(x, y) \mid 0 \le x \le 1, y = 0\}. \tag{4.20}$$

In (4.15), we replace the second equation with $\delta y_i = 0$, fixing the node vertically. Further, we recognize that the Lagrangian corresponding to nodes not in $\mathcal{E}$ are zero on the edge, resulting from the continuity requirement between adjacent elements —

the solution along an edge must be solely a function of the nodes on the edge. Thus,

$$\left.\frac{\partial \ell_j}{\partial x}\right|_i = 0, \quad i \in \mathcal{E} \cap j \notin \mathcal{E}. \tag{4.21}$$

We may thus rewrite the remaining equation in (4.15) for $i \in \mathcal{E}$,

$$\begin{aligned} 0 = \left.\frac{\partial \ell_i}{\partial x}\right|_i + \left.\frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial x}\right|_i \delta x_i + \cancelto{0}{\left.\frac{\partial}{\partial y}\frac{\partial \ell_i}{\partial x}\right|_i \delta y_i} \\ - \sum_{j \in \mathcal{E}} \left.\frac{\partial \ell_i}{\partial x}\right|_j \left.\frac{\partial \ell_j}{\partial x}\right|_i \delta x_j - \sum_{j \in \mathcal{E}} \left.\frac{\partial \ell_i}{\partial y}\right|_j \cancelto{0}{\left.\frac{\partial \ell_j}{\partial x}\right|_i \delta y_j} \\ - \sum_{j \notin \mathcal{E}} \left.\frac{\partial \ell_i}{\partial x}\right|_j \cancelto{0}{\left.\frac{\partial \ell_j}{\partial x}\right|_i} \delta x_j - \sum_{j \notin \mathcal{E}} \left.\frac{\partial \ell_i}{\partial y}\right|_j \cancelto{0}{\left.\frac{\partial \ell_j}{\partial x}\right|_i} \delta y_j, \end{aligned} \tag{4.22}$$

which is solely a function of the x location of nodes $\mathcal{E}$,

$$0 = \left.\frac{\partial \ell_i}{\partial x}\right|_i + \left.\frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial x}\right|_i \delta x_i - \sum_{j \in \mathcal{E}} \left.\frac{\partial \ell_i}{\partial x}\right|_j \left.\frac{\partial \ell_j}{\partial x}\right|_i \delta x_j. \tag{4.23}$$

We recognize this equation as a Newton–Raphson iteration for one-dimensional Fekete nodes, the solution of which is just the set of Gauss–Lobatto–Legendre points (2.29). We therefore may simplify our calculations by initializing the edge nodes as Gauss–Lobatto–Legendre nodes and setting both δx and δy to zero (see Figure 4.5).

4.2.4 Global Search

Armed with an elegant local search algorithm, I present an effective, yet rather brutish, global search procedure (Figure 4.6) motivated by the fact, discovered early on, that two nodal configurations may differ in as little as two nodes. That is, the impact of each node is often local, with perturbation of one node only affecting its immediate neighbours.

At each stage of the algorithm, a random node is selected from one of the maximums found so far. The position of this node is then perturbed by selecting a random point as per Appendix A.

The algorithm resembles a drunkard's walk in n-dimensional space with every step either snapping to a new maximum or to a previous maximum. While inelegant, this procedure has proved successful in finding all the local maximums of Taylor *et al.* [64] plus extending results to degree 30.

One of the strengths of the algorithm is the accumulation of local Fekete maxi-

1. Start with an initial configuration of points as per §4.2.1.
2. Determine the local Fekete maximum as per §4.2.2 and place in a stack.
3. While not futile do:
   (a) Choose a configuration of points from the stack.
   (b) Perturb a random node in the configuration.
   (c) Determine the local Fekete maximum as per §4.2.2.
   (d) If new maximum, place in stack and reset futility, else increase futility.
   (e) If stack too large, cull stack.

Figure 4.6: Global Fekete search algorithm

mums, maximums which may, while not having maximal Vandermonde determinants, have smaller Lebesgue norms. As the number of maximums grows quickly as the algorithm proceeds, I limit the number of the maximums by culling all but the top $2p$ maximums.

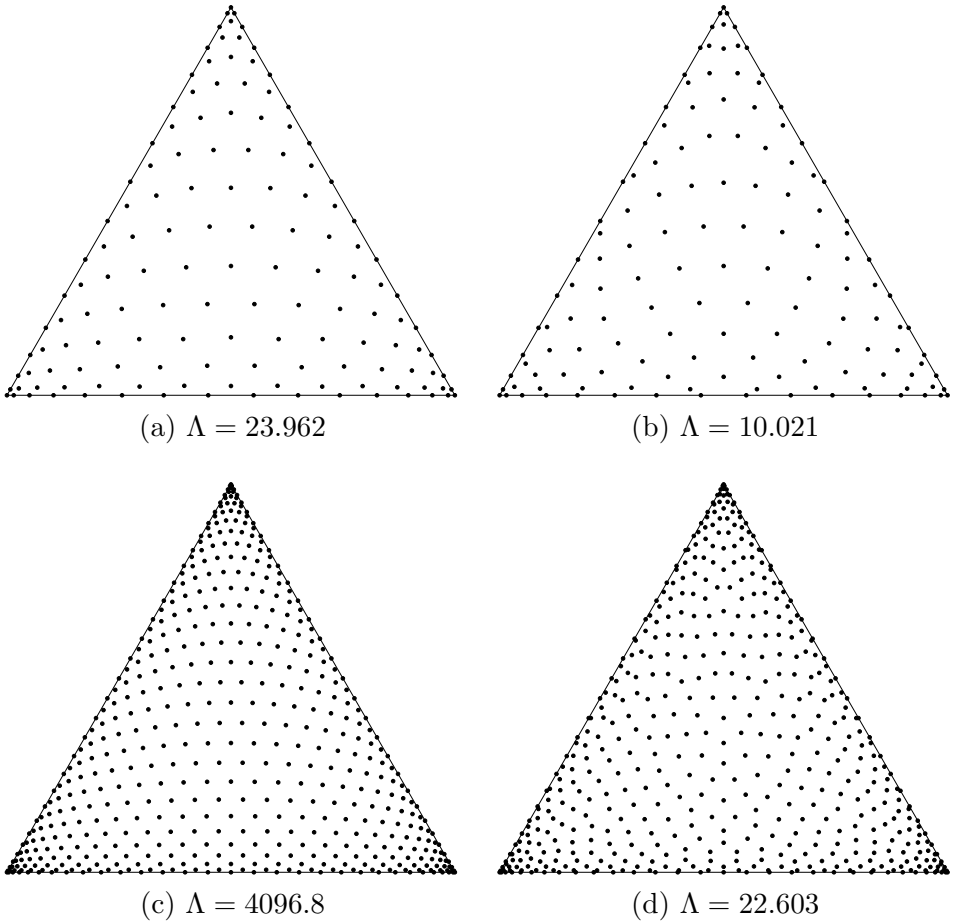
4.3 Numerical Simulations

Using the aforementioned algorithm, I have computed sets of Fekete nodes through degree 30 (Figure 4.7). For each degree I report the minimum Lebesgue norm found (Table 4.2) with previous results included for comparison. Note that, in many cases, these prior results underestimated Λ; I thus have verified each Lebesgue norm interactively. The new Lebesgue norms are the smallest reported to date.

In agreement with Taylor *et al.* [64], my extended Fekete results confirm the linear growth in degree of the Lebesgue norm. By (2.24), this fact ensures the spectral convergence of interpolation to any holomorphic function.

Note also that, in agreement with Taylor *et al.* [64], the connectivity is highly non-uniform for degrees higher than six. Indeed, the rank of the Fekete maximum corresponding to a uniform connectivity pattern decreases rapidly along with a corresponding rapid increase in Λ (Figure 4.7), not meeting the requirements of (2.24).

As the degree increases, the number of local Fekete maximums close to the global Fekete maximums increases. It would appear that a laborious search for a global Fekete maximum, perhaps unattainable for large degree, is unnecessary. Indeed, the global maximum was seldom the best configuration in terms of Lebesgue norm.



(a) $\Lambda = 23.962$ (b) $\Lambda = 10.021$

(c) $\Lambda = 4096.8$ (d) $\Lambda = 22.603$

Figure 4.7: The degree 15 and 30 Fekete node distributions. The distributions on the left were initialized with a uniform connectivity pattern while the distributions on the right have the smallest Lebesgue norms found.

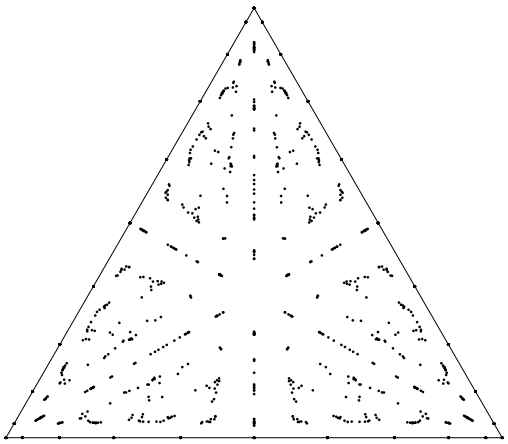


Figure 4.8: An overlay of 20 different degree 10 Fekete point solutions showing a tendency to clustering.

p	$\mathscr{F}$(a)	$\mathscr{F}$(b)	$\mathscr{E}$(c)	$\mathscr{E}$(d)	$\mathscr{B}$(e)	$\mathscr{F}$(f)	$\mathscr{F}$(g)
3	2.113	2.113	2.115	2.113	2.111	2.112	2.113
4	2.729	2.729	2.605	2.588	2.692	2.729	2.729
5	3.611	3.611	3.210	3.196	3.301	3.611	3.973
6	4.171	4.171	4.074	4.075	3.791	4.171	4.512
7	4.928	4.928	4.790	4.780	4.391	4.928	5.525
8	5.905	5.905	5.883	5.855	5.089	5.905	-
9	6.803	6.803	6.939	6.886	5.918	6.803	-
10	7.852	7.875	8.421	8.466	7.085	7.997	-
11	7.907	7.907	10.092	10.144	8.338	9.519	-
12	8.472	8.472	12.416	12.637	10.083	11.080	-
13	9.279	9.279	15.331	-	12.046	13.240	-
14	9.959	9.959	22.861	-	-	-	-
15	10.021	10.021	29.691	-	-	-	-
16	10.689	12.198	41.724	-	-	-	-
17	11.534	13.884	-	-	-	-	-
18	13.128	14.739	-	-	-	-	-
19	12.634	15.440	-	-	-	-	-
20	15.216	-	-	-	-	-	-
21	15.921	-	-	-	-	-	-
22	15.684	-	-	-	-	-	-
23	15.876	-	-	-	-	-	-
24	16.443	-	-	-	-	-	-
25	17.405	-	-	-	-	-	-
26	19.975	-	-	-	-	-	-
27	20.333	-	-	-	-	-	-
28	18.756	-	-	-	-	-	-
29	21.935	-	-	-	-	-	-
30	22.603	-	-	-	-	-	-

Table 4.2: Summary of the minimum Lebesgue norms: (a) Fekete nodes; (b) Fekete nodes of Taylor *et al.* [64]; (c,d) electrostatic nodes of Hesthaven [29] with Chebyshev and Legendre edge nodes, respectively; (e,f) Chen–Babuška and Fekete nodes, respectively, of Chen and Babuška [13]; and (g) Fekete nodes of Bos [8]. Columns are ordered chronologically with earliest results on the right.

The global optimization procedure depended upon the fact that two configurations often roughly shared many common nodes. This can be confirmed by overlaying all the Fekete maximums for a given degree (Figure 4.8). The distributions all tend to cluster in certain preferred areas while avoiding wide "channels" near the medians and edges. This fact could lead to an improved genetic global search scheme by which two parent distributions could combine through crossover to form a new, unique offspring. This individual would share all the common elements of the parents while incorporating some of the unique elements of both parents.

Symmetry was imposed on the system to eliminate any impact of triangle rotation on the solution. However, it is possible that asymmetrical solutions would yield Fekete maximums with even higher determinants, perhaps based on Fibonacci spirals.

Chapter 5

Chen–Babuška Nodes

The Lebesgue norm Λ bounds the error in polynomial interpolation — its minimization should yield an optimal set of nodes for a given degree p. However, its calculation is itself a difficult global optimization problem. Chen and Babuška [13] proposed an alternative, explicit proxy for the Lebesgue norm and proceeded to minimize this measure to degree 13, in the process discovering some of the best nodal sets of the time. Unfortunately their nodal sets do not exhibit the required asymptotic requirement $\lim_{p\to\infty} \Lambda^{1/p} = 1$. However their optimizations appear to have been initialized with uniformly-spaced, triangular grids of nodes and it is expected that more irregular-shaped initializations will yield better nodal sets. To this end, using the Fekete nodal sets of Chapter 4 as starting points, I minimize the above Chen–Babuška measure up to degree 30. I discover that these new nodal sets do indeed satisfy the required limit above yielding the best Lebesgue norms found to date, however with only minimal improvement in Λ.

5.1 Introduction

The calculation of the Lebesgue norm is a difficult global maximization problem (see Chapter 3). Chen and Babuška [13], motivated by this fact, introduced a "mean L^2" optimality measure,

$$\beta \equiv \int_{\triangle} \sum_{i=1}^{n} \ell_i^2 \, \mathrm{d}A, \tag{5.1}$$

and suggested that minimizing β is tantamount to minimizing Λ. We thus define the set of Chen–Babuška nodes such that

$$\mathscr{B} = \arg \min_{\{x_i, y_i\}} \beta. \tag{5.2}$$

The sets $\mathscr{B}$ reported by Chen and Babuška [13] do not approach the required

asymptotic limit on the Lebesgue norm, $\lim_{p\to\infty} \Lambda^{1/p} = 1$, a requirement for our polynomial approximation to converge to a holomorphic function as we increase the polynomial degree (see §2.6). However, though the details of their method were not made available, it appears that their optimization simulations were initialized with a uniform set of nodes. As discovered in Chapter 4, this initialization of Fekete nodes resulted in Lebesgue norms which did not approach the above limit. Here, we investigate if a non-uniform initialization of nodes would result in Chen–Babuška nodes which would return the proper asymptotic limit and thus establish their suitability for polynomial interpolation.

5.2 Newton–Raphson Algorithm

Using the integration of product result of §2.5, the determination of β is remarkably simple, providing an intriguing linkage with the trace of matrix of integration weights,

$$\beta = \int_{\triangle} \ell_i \ell_i \, \mathrm{d}A = \operatorname{trace} W_{ij} = W_{ii}. \tag{5.3}$$

As W_{ij} is a function solely of the Vandermonde determinant, β is a function solely of the nodal positions $\{x_j, y_j\}$. A Taylor series expansion of the gradient of β with respect to these nodal positions is

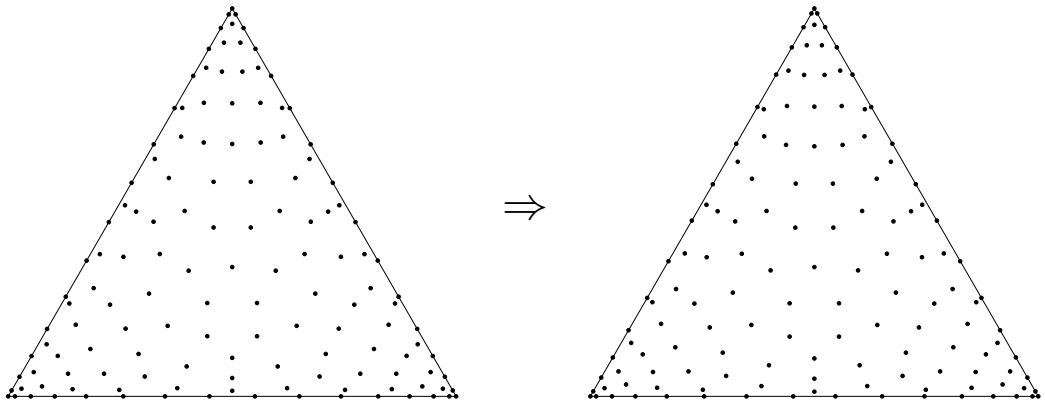
$$\begin{aligned}
\frac{\partial \beta}{\partial x_i} &\approx \left.\frac{\partial \beta}{\partial x_i}\right|_\circ + \sum_j \left.\frac{\partial}{\partial x_j}\frac{\partial \beta}{\partial x_i}\right|_\circ \delta x_j + \sum_j \left.\frac{\partial}{\partial y_j}\frac{\partial \beta}{\partial x_i}\right|_\circ \delta y_j + \text{H.O.T.} \\
\frac{\partial \beta}{\partial y_i} &\approx \left.\frac{\partial \beta}{\partial y_i}\right|_\circ + \sum_j \left.\frac{\partial}{\partial x_j}\frac{\partial \beta}{\partial y_i}\right|_\circ \delta x_j + \sum_j \left.\frac{\partial}{\partial y_j}\frac{\partial \beta}{\partial y_i}\right|_\circ \delta y_j + \text{H.O.T.}
\end{aligned} \tag{5.4}$$

where $|_\circ$ denotes quantities evaluated at the point of expansion. Ignoring higher-order terms in the Taylor series, setting the left-hand side to zero, results in a Newton–Raphson system

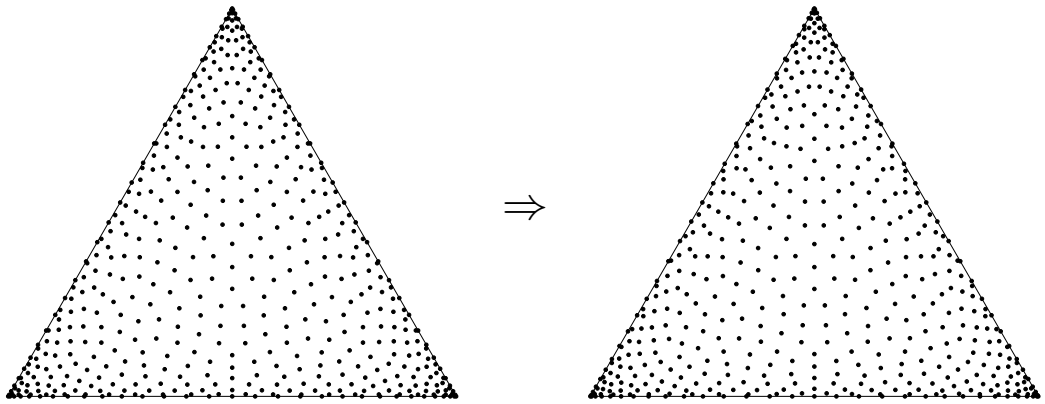
$$0 = \begin{Bmatrix} \frac{\partial \beta}{\partial x_i} \\ \frac{\partial \beta}{\partial y_i} \end{Bmatrix} + \begin{bmatrix} \frac{\partial}{\partial x_j}\frac{\partial \beta}{\partial x_i} & \frac{\partial}{\partial y_j}\frac{\partial \beta}{\partial x_i} \\ \frac{\partial}{\partial x_j}\frac{\partial \beta}{\partial y_i} & \frac{\partial}{\partial y_j}\frac{\partial \beta}{\partial y_i} \end{bmatrix} \begin{Bmatrix} \delta x_j \\ \delta y_j \end{Bmatrix}, \quad i, j = 1 \ldots n \tag{5.5}$$

where determination of $\{\delta x_i, \delta y_i\}$ is found through inversion given suitable boundary and initial conditions. Derivation of the gradient and Hessian are algebraically tedious but have a simple representation (see Appendix B).

As in Chapter 4, we impose symmetry on the system and ensure that $p + 1$ nodes remain on each triangle edge. To this end, we replace the gradient equations



(a) Fekete $\Lambda = 10.894$ (b) Chen–Babuška $\Lambda = 10.306$



(c) Fekete $\Lambda = 23.128$ (d) Chen–Babuška $\Lambda = 18.724$

Figure 5.1: On the right are the degree 15 and 30 Chen–Babuška node distributions with the smallest Lebesgue norms found and on the left are the corresponding Fekete node distributions.

corresponding to each node with the boundary conditions of Figure 4.5.

Newton's method is an eigenvector-following technique in that it will maximize along Hessian eigenvectors corresponding to negative eigenvalues and minimize along positive eigenvalues. In short, it seeks stationary points; thus, in order to minimize β, one must ensure a positive-definite matrix. Starting from a good configuration such as a Fekete solution typically results in such a matrix. If and while the Hessian is non-positive-definite, I rely on a steepest-descent, gradient-following algorithm.

For each degree p, we have $2p$ Fekete solutions from Chapter 4. We use each of these nodal sets as initial conditions to our minimization routine. We then select the resulting set with the smallest Λ.

p	$\mathscr{B}$(a)	$\mathscr{F}$(b)	$\mathscr{F}$(c)	$\mathscr{E}$(d)	$\mathscr{E}$(e)	$\mathscr{B}$(f)	$\mathscr{F}$(g)	$\mathscr{F}$(h)
3	2.111	2.113	2.113	2.115	2.113	2.111	2.112	2.113
4	2.692	2.729	2.729	2.605	2.588	2.692	2.729	2.729
5	3.301	3.611	3.611	3.210	3.196	3.301	3.611	3.973
6	3.791	4.171	4.171	4.074	4.075	3.791	4.171	4.512
7	4.391	4.928	4.928	4.790	4.780	4.391	4.928	5.525
8	5.089	5.905	5.905	5.883	5.855	5.089	5.905	-
9	5.918	6.803	6.803	6.939	6.886	5.918	6.803	-
10	7.085	7.852	7.875	8.421	8.466	7.085	7.997	-
11	7.266	7.907	7.907	10.092	10.144	8.338	9.519	-
12	8.669	8.472	8.472	12.416	12.637	10.083	11.080	-
13	9.291	9.279	9.279	15.331	-	12.046	13.240	-
14	8.988	9.959	9.959	22.861	-	-	-	-
15	10.306	10.021	10.021	29.691	-	-	-	-
16	10.413	10.689	12.198	41.724	-	-	-	-
17	10.254	11.534	13.884	-	-	-	-	-
18	11.165	13.128	14.739	-	-	-	-	-
19	12.025	12.634	15.440	-	-	-	-	-
20	12.861	15.216	-	-	-	-	-	-
21	13.630	15.921	-	-	-	-	-	-
22	14.059	15.684	-	-	-	-	-	-
23	14.725	15.876	-	-	-	-	-	-
24	15.618	16.443	-	-	-	-	-	-
25	15.319	17.405	-	-	-	-	-	-
26	16.991	19.975	-	-	-	-	-	-
27	17.670	20.333	-	-	-	-	-	-
28	17.221	18.756	-	-	-	-	-	-
29	18.031	21.935	-	-	-	-	-	-
30	18.724	22.603	-	-	-	-	-	-

Table 5.1: Summary of the minimum Lebesgue norms: (a) Chen–Babuška nodes; (b) Fekete nodes of Chapter 4; (c) Fekete nodes of Taylor *et al.* [64]; (d,e) electrostatic nodes of Hesthaven [29] with Chebyshev and Legendre edge nodes, respectively; (f,g) Chen–Babuška and Fekete nodes, respectively, of Chen and Babuška [13]; and (h) Fekete nodes of Bos [8]. Columns are ordered chronologically with earliest results on the right.

5.3 Numerical Simulations

I have computed Chen–Babuška nodal sets through to degree 30 (see Figure 5.1). For each degree I report the smallest Lebesgue norm found (see Table 5.1), with previous results included for comparison. Note that, in some cases, prior results were underestimated and have thus been corrected. I have verified each Lebesgue norm interactively.

In contrast to the work of Chen and Babuška [13], these results indicate a Lebesgue norm which grows linearly with degree thereby satisfying the asymptotic limit (2.24).

With only a few exceptions, $p = \{12, 13, 15\}$, the Chen–Babuška Lebesgue norms are smaller than the corresponding degree Fekete Lebesgue norms and grow at a slower rate.

In agreement with the results of Chapter 4, the optimal triangulation of nodes is non-isometric (§2.9) for degrees higher than six. The rank of the Chen–Babuška minimum corresponding to an isometric-like connectivity decreases rapidly with degree.

The Λ of Chen–Babuška nodes is sometimes larger than the Λ of their corresponding initial Fekete nodes. For example, of 30 Chen–Babuška degree 15 solutions, 7 had larger Λ than their Fekete node counterparts. This fact is not surprising. Near the edges where the points are loosely packed, λ is a high frequency function. While λ may have large peaks near the edges and thus a large Λ, the area of integration is small and thus β small. β is dominated by the large λ of the vast open spaces in the middle of the triangle. The minimization β therefore preferentially reduces λ in the middle, perhaps at the expense of λ near the edges.

Chapter 6

Lebesgue Nodes

The Lebesgue norm Λ bounds the error in polynomial interpolation. In Chapter 5, I minimized a proxy for Λ and found nodal sets with the best Lebesgue norms discovered to date. In this chapter, I take the next logical step and minimize Λ directly in an attempt to find the optimal nodal configurations. These minimizations are made possible through the dividing triangles algorithm of Chapter 3 coupled with a differential evolution algorithm. The discovered nodal sets are putatively optimal.

6.1 Introduction

If the Lebesgue norm (see §2.6) is the best indicator of the quality of interpolation, direct minimization should yield optimal nodal configurations. We define the set of Lebesgue nodes found from the continuous min-max problem

$$\mathscr{L} = \arg \min_{\{x_i,y_i\}} \max_{(x,y)} \lambda. \tag{6.1}$$

In one dimension, the Lebesgue function λ achieves a single maximum in the interval between each node, here denoted λ_i. Not only does this facilitate the calculation of the Lebesgue norm, $\Lambda = \max_i \lambda_i$, but de Boor and Pinkus [14] proved that the nodal set $\mathscr{L}$ requires that the Lebesgue function "equioscillates" such that

$$\lambda_1 = \lambda_2 = \cdots = \lambda_p. \tag{6.2}$$

Benefiting from this relationship, Angelos *et al.* [1] constructed an algorithm which determined the one-dimensional Lebesgue nodes up to degree 100.

In two-dimensions, no such relationship as (6.2) exists. Both the calculation of Λ and its subsequent minimization are difficult global optimization problems. Indeed, in the literature only results for the equivalent problem on the sphere exist [66].

Here, we use a dividing triangles algorithm developed in Chapter 3 to determine

Λ coupled with a differential evolution algorithm [62] to optimize the nodes based on this Λ.

6.2 Differential Evolution Algorithm

Minimizing the Lebesgue norm is like trying to compress a balloon — reducing Λ at one point in the triangle may cause a bulge in Λ elsewhere. This is due to the many local maximums of the Lebesgue function. As the algorithm proceeds toward convergence, it is expected these multiple local maximums will equioscillate somewhat analogously to (6.2).

In this work, I will calculate the Lebesgue norm as per Chapter 3 but expect that the answer will not be exact and thus discontinuous, precluding the use of any gradient-based algorithm. Several deterministic algorithms such as Nelder–Mead [45] and minimax smoothing [67] and several stochastic algorithms such as simulated annealing [33], particle swarm optimization [39], and differential evolution [62] were implemented. Of these algorithms only differential evolution proved adequate.

Genetic algorithms mimic the Darwinian principle of survival of the fittest: a population of members mutate and combine to form the next generation. This next generation then competes with the previous generation by comparing some "fitness" function. Differential evolution [62] is a particularly compact and robust genetic algorithm. Consider an individual X described with "genes" x_i, here containing the nodal locations. From a population of many such individuals, three distinct members are chosen randomly. To the first member, some fraction of the differential between the other two members is added

$$Y \leftarrow X_1 + F\left(X_2 - X_3\right). \tag{6.3}$$

In the next stage, this perturbed individual is combined with an additional randomly-chosen, distinct vector X using binomial crossover. That is, a new vector Z is formed with crossover probability C with genes from X and with probability $1 - C$ with genes from Y

$$z_i \leftarrow \begin{cases} x_i, & \text{if } u_i \leq C \\ y_i, & \text{if } u_i > C \end{cases} \tag{6.4}$$

where u_i is a uniform random variate in $[0, 1]$. This process is repeated for each member in the population to form a new generation. These new members are evaluated for fitness, here the Lebesgue norm Λ, and compared with their corresponding member in the previous generation. The member with the better fitness survives, creating

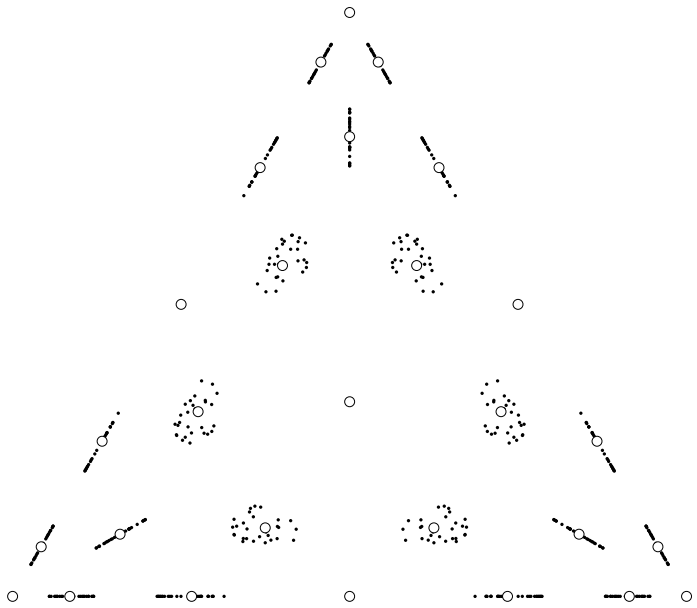


Figure 6.1: An example initial population for differential evolution distributed about a Fekete nodal set.

a new generation as good or better than the previous generation. By repeating the process, the population should evolve to a higher state.

There are few parameters to differential evolution and the algorithm is robustly indifferent to them. Indeed, we used $F = 1/2$ and $C = 9/10$ for every simulation.

In an attempt to find the configuration corresponding to the global Lebesgue norm minimum, we apply the differential evolution algorithm to each Fekete point configuration or, in the spirit of differential evolution, "tribe" found in Chapter 4. The use of Fekete nodes is based on the assumption that $\mathscr{F} \approx \mathscr{L}$.

A population is initialized by placing points randomly in a small neighbourhood about a tribe benefiting from the transformation introduced in Chapter 8. That is, for a given configuration of Fekete nodes, we first calculate the packing radius (see §8.3.2). Then, for each node we project the barycentric coordinates onto the sphere (see §8.1), distribute points randomly in a spherical cap of radius equal to the packing radius and centred around the node, and transform the points back to barycentric space (see Figure 6.1). Points on the triangle edges and medians are kept there and symmetry is maintained (see Figure 4.5). This initialization procedure ensures that the distribution of points is scaled, with points packed loosely in the middle of the triangle and tightly near the edges.

However, like evolution itself, genetic algorithms are extremely slow and not guaranteed to achieve a global maximum. For each tribe we run differential evolution

p	$\mathscr{L}$(a)	$\mathscr{B}$(b)	$\mathscr{F}$(c)	$\mathscr{F}$(d)	$\mathscr{E}$(e)	$\mathscr{E}$(f)	$\mathscr{B}$(g)	$\mathscr{F}$(h)	$\mathscr{F}$(i)
3	2.108	2.111	2.113	2.113	2.115	2.113	2.111	2.112	2.113
4	2.587	2.692	2.729	2.729	2.605	2.588	2.692	2.729	2.729
5	3.081	3.301	3.611	3.611	3.210	3.196	3.301	3.611	3.973
6	3.595	3.791	4.171	4.171	4.074	4.075	3.791	4.171	4.512
7	4.143	4.391	4.928	4.928	4.790	4.780	4.391	4.928	5.525
8	4.766	5.089	5.905	5.905	5.883	5.855	5.089	5.905	-
9	5.486	5.918	6.803	6.803	6.939	6.886	5.918	6.803	-
10	5.921	7.085	7.852	7.875	8.421	8.466	7.085	7.997	-
11	6.724	7.266	7.907	7.907	10.092	10.144	8.338	9.519	-
12	7.187	8.669	8.472	8.472	12.416	12.637	10.083	11.080	-
13	7.253	9.291	9.279	9.279	15.331	-	12.046	13.240	-
14	7.706	8.988	9.959	9.959	22.861	-	-	-	-
15	8.243	10.306	10.021	10.021	29.691	-	-	-	-

Table 6.1: Summary of the minimum Lebesgue constants: (a) Lebesgue nodes; (b) Chen–Babuška nodes of Chapter 5; (c) Fekete nodes of Chapter 4; (d) Fekete nodes of Taylor *et al.* [64]; (e,f) electrostatic nodes of Hesthaven [29] with Chebyshev and Legendre edge nodes, respectively; (g,h) Chen–Babuška and Fekete nodes, respectively, of Chen and Babuška [13]; and (i) Fekete nodes of Bos [8]. Columns are ordered chronologically with earliest results on the right.

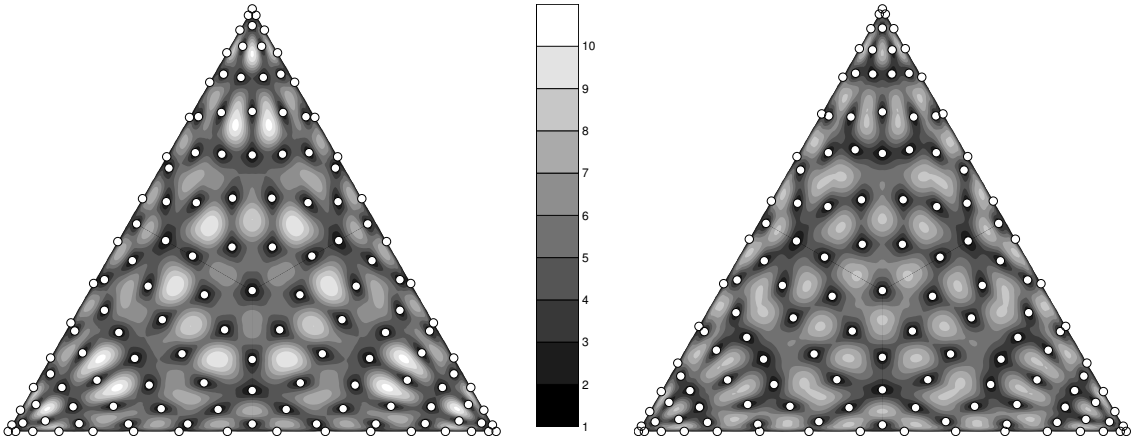
until an initial convergence criterion is met: considering the best and worst members of the tribe, we simulate until the difference of their Lebesgue constants and their nodal positions does not exceed 1×10^{-4}. We then concentrate our efforts on the best tribe found until convergence to 1×10^{-8}. This best tribe is our best estimate of the Lebesgue set $\mathscr{L}$.

6.3 Numerical Simulations

We have computed Lebesgue nodes through degree 15. For each degree we report the smallest Lebesgue norm found (see Table 6.1) with previous results included for comparison. Note that, in many cases, these prior results were underestimated. We have verified and corrected each Lebesgue norm interactively. Our Lebesgue norms are the smallest reported to date and thus putatively optimal.

As with Fekete and Chen-Babuška nodes, the Lebesgue norm grows linearly with degree, ensuring their suitability for polynomial interpolation by (2.24). Additionally, of the three classes of nodes, the Lebesgue node norms grow at the slowest rate with increasing degree.

The presented algorithm attempts to both equalize and reduce the Lebesgue function peaks (Figure 6.2). This behaviour is the two-dimensional analogue of the one-dimensional result of (6.2).



(a) Fekete $\Lambda = 10.386$ (b) Lebesgue $\Lambda = 8.243$

Figure 6.2: In (a) we plot the Lebesgue function of the initial distribution of 15th degree Fekete nodes while in (b) we show the corresponding Lebesgue nodes. Both contour plots are coloured to the same scale.

Fekete configurations with minimum Λ, except for low degree cases, do not typically lead to Lebesgue configurations with minimum Λ. Rather the best Lebesgue configurations tend to derive from the Fekete node configurations with the largest Vandermonde determinant regardless of initial Lebesgue norm.

As the Lebesgue function maximum always occurs in the interior of the triangles, the Lebesgue edge nodes do not attempt to equioscillate the Lebesgue function along the edges, rather they move to affect a reduction in the interior. Thus, Lebesgue exterior nodes are not distributed as one-dimensional Lebesgue nodes.

However, though not shown, if the Lebesgue norm is minimized while holding the edge nodes fixed and distributed as Gauss–Lobatto–Legendre points (§2.8), the Lebesgue constant is only slightly larger in comparison to free edge nodes. Thus in hybrid triangle–quadrilateral meshes, the triangle edges may be made to conform with standard tensor-product Legendre quadrilaterals with little cost in Λ.

Chapter 7

Lagrangian Voronoi Tessellations

I introduce a new class of Voronoi diagram pertinent to interpolation. In contrast to the classic Euclidean Voronoi diagram, the Lagrangian Voronoi diagram uses the magnitude of a node's Lagrangian interpolating polynomial as a measure of its influence on the domain. Though applicable to any domain, I present an algorithm suitable for their construction in the triangle. Through several illustrative examples, I show that these diagrams provide an additional qualitative measure for evaluating nodal distributions. Lagrangians should influence the solution only in the immediate neighbourhood of their respective nodes. Lagrangian Voronoi diagrams violating this principle typically exhibit "orphan" Voronoi regions, regions isolated from their parent nodes.

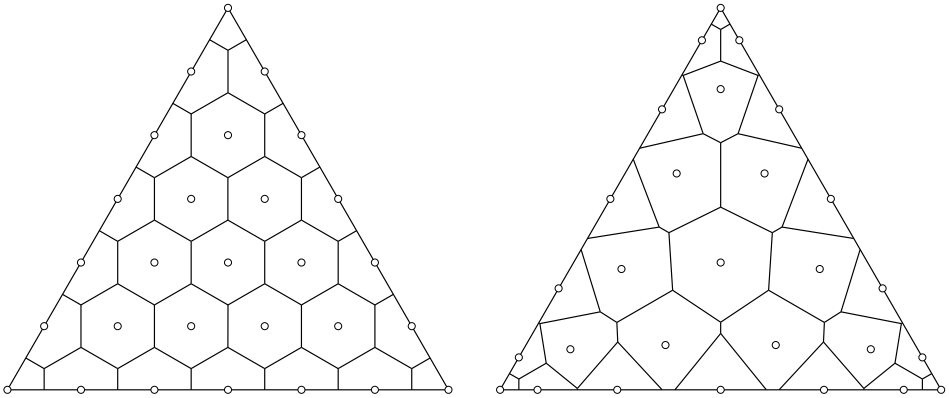
7.1 Introduction

The Voronoi diagram is a fundamental concept in computational geometry with applications in biology, computer science, scheduling, facility placement, to name but a few [3, 46]. The concept is simple: given a set of points in some continuous space, associate all locations in that space with the closest member of the point set. More precisely, the Voronoi diagram is the union of all Voronoi regions R_i defined by

$$R_i = \left\{ x \in \Omega^d \mid \delta(x, z_i) < \delta(x, z_j), \forall j \neq i \right\}, \tag{7.1}$$

where $\delta(x, y)$ is typically the geodesic, or shortest, distance between vectors x and y. In other words, the Voronoi region R_i associated with node z_i is the continuum of points x closer to z_i than any other node. But what distance is pertinent to interpolation in the triangle?

We define the Euclidean Voronoi tessellation as the Voronoi tessellation with distance metric $\delta(x, y) = \|x - y\|$. Consider a set of isometric nodes (§2.9) and its corresponding Euclidean Voronoi diagram (Figure 7.1a). The hexagonal pattern of



(a) Isometric (b) Fekete

Figure 7.1: Euclidean Voronoi tessellations for isometric and Fekete nodal sets.

cells demonstrates the uniform spacing of the nodes. However, these nodes are a poor choice for interpolation; we require a higher density of nodes near the edges and a lower density in the centre. Instead, we select an equivalent degree set of Fekete nodes. These nodes possess our required density distribution and are an excellent choice for interpolation (Chapter 4). Consider the Euclidean Voronoi diagram of this nodal set (Figure 7.1b). Interior Voronoi regions now extend to the triangle edges; in fact, some Voronoi vertices extend outside the triangle. It is a feature and requirement of Lagrangian interpolation that the interior nodes have zero influence on the approximation on the edges. Otherwise, continuity between adjacent elements is not ensured. The fact that interior Voronoi regions extend to the boundaries implies that our distance metric is wrong.

Instead of talking about distance, it is perhaps more useful to speak of dominance or influence. For example, a Voronoi region can also be described as the area over which a given node is dominant. In Euclidean terms, this dominance is inversely proportional to distance. One might ask what is a analogous measure of dominance or influence of a node for interpolation?

In this framework, there is only one answer — the Lagrangians (§2.4). The Lagrangians describe the spatial influence of each node on the domain. If we substitute them as a proxy for the actual distance, we may define a "Lagrangian" Voronoi tessellation as the union of all regions

$$R_i = \left\{ x \in \Omega^d \mid |\ell_i(x)| > |\ell_j(x)|, \forall j \neq i \right\}. \tag{7.2}$$

Well-behaved Lagrangians should behave in a Gaussian or sinc-like manner exhibiting

1. Find an initial Voronoi vertex and place in stack.
2. While stack not empty do:
   (a) Pop top vertex from stack.
   (b) Let $i = \arg\max_i |\ell_i|$.
   (c) Find all surfaces j such that $|\ell_j| = |\ell_i|$.
   (d) Order surfaces j $\circlearrowleft$ about vertex.
   (e) For each pair of adjacent surfaces in j do:
      i. Step along intersection curve until near a Voronoi vertex.
      ii. Refine location of Voronoi vertex using Newton iteration.
      iii. If new, push vertex to stack.

Figure 7.2: Algorithm to determine the Voronoi vertices of a Lagrangian Voronoi diagram.

a decaying magnitude with distance from the node. Indeed, in the limit as $n \to \infty$, the Lagrangians should resemble discrete Dirac delta functions.

7.2 Construction

The construction of a Euclidean Voronoi diagram is well advanced. Voronoi vertices are located at circumcentres of the Delaunay triangulation, with many efficient schemes available [24]. Unfortunately, Lagrangian Voronoi tessellations are rather more difficult. Here, Voronoi cells are delineated by the intersections of the absolute value of the Lagrangians. While Euclidean distance surfaces are conical, intersecting in straight lines, the Lagrangian surfaces are high-order, sinc-like polynomials, with complicated intersection curves.

Here, I present a simple algorithm (Figure 7.2) to construct the Lagrangian diagram by finding the complete list of Voronoi vertices through travelling from vertex to vertex along the curve of intersection between two Lagrangian surfaces. Of course, alternative algorithms are possible; however, our current algorithm has proved robust and adequate for our needs.

Key to this algorithm is the journey from vertex to vertex. In the spirit of fluid dynamics, I define a stream function

$$\psi = |\ell_1| - |\ell_2|. \tag{7.3}$$

The intersection between the two surfaces corresponds to $\psi = 0$. The tangent to this

line is then just the "velocity"

$$u = -\frac{\partial \psi}{\partial y} = \operatorname{sgn}(\ell_2)\frac{\partial \ell_2}{\partial y} - \operatorname{sgn}(\ell_1)\frac{\partial \ell_1}{\partial y}, \tag{7.4}$$

$$v = +\frac{\partial \psi}{\partial x} = \operatorname{sgn}(\ell_1)\frac{\partial \ell_1}{\partial x} - \operatorname{sgn}(\ell_2)\frac{\partial \ell_2}{\partial x}. \tag{7.5}$$

Given a fairly arbitrary step size, I make a tentative step along this velocity. If this step no longer has the two specified surfaces to the right and left, I bisect the step; otherwise, I accept the step. In this manner I approach the next vertex. When the size of step is reduced sufficiently, we are in the vicinity of a Voronoi vertex, a point at which at least three surfaces meet. Choosing any three of these surfaces, they are well represented by

$$|\ell_i| \approx |\tilde{\ell}_i| + \frac{\partial |\tilde{\ell}_i|}{\partial x}\delta x + \frac{\partial |\tilde{\ell}_i|}{\partial y}\delta y, \quad i = 1, 2, 3, \tag{7.6}$$

in the vicinity of the intersection point. Their intersection, $|\ell_1| = |\ell_2| = |\ell_3|$, creates the system

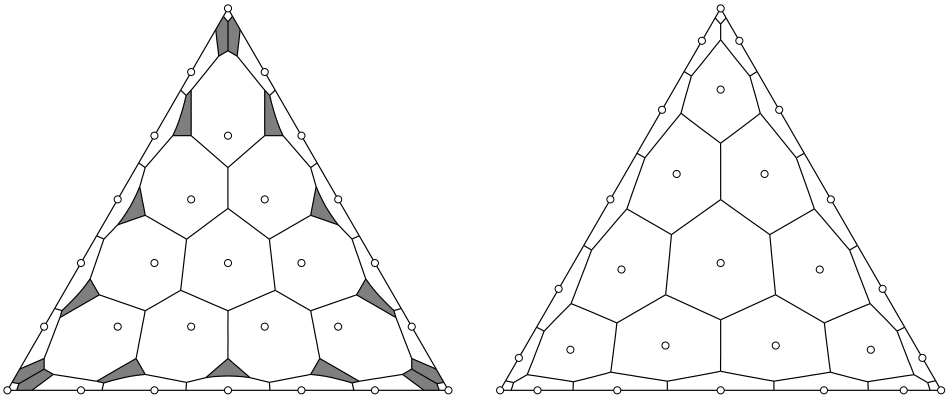
$$\begin{bmatrix} 1 & -\frac{\partial |\tilde{\ell}_1|}{\partial x} & -\frac{\partial |\tilde{\ell}_1|}{\partial y} \\ 1 & -\frac{\partial |\tilde{\ell}_2|}{\partial x} & -\frac{\partial |\ell_2|}{\partial y} \\ 1 & -\frac{\partial |\tilde{\ell}_3|}{\partial x} & -\frac{\partial |\ell_3|}{\partial y} \end{bmatrix} \begin{Bmatrix} |\ell| \\ \delta x \\ \delta y \end{Bmatrix} = \begin{Bmatrix} |\tilde{\ell}_1| \\ |\tilde{\ell}_2| \\ |\tilde{\ell}_3| \end{Bmatrix}, \tag{7.7}$$

where the tilde indicates quantities evaluated at the current estimate of the intersection point. Iterating this system rapidly yields the location of the Voronoi vertex.

Note that the algorithm presented here is for triangles; however, the principles and methods should extend directly to other geometries and dimensions such as simplices (e.g. tetrahedron) or hyperspheres. Note also that this algorithm is computationally expensive. While there is a lower bound of n Voronoi regions for n nodes there is no upper bound, due to the possibility of orphan regions. Conceivably, faster methods, perhaps of a level set nature [54], could be developed.

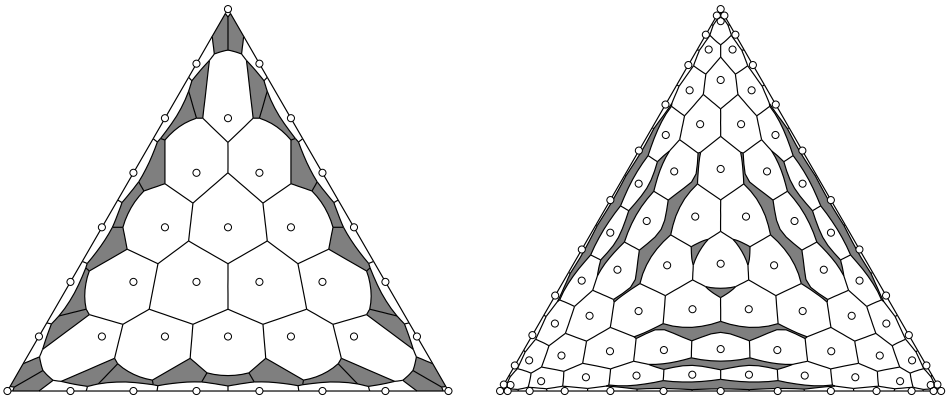
7.3 Example Tessellations

Applying our algorithm to the nodal sets of Figure 7.1 yields the Lagrangian Voronoi tessellations of Figure 7.3. We note two important consequences of our change in distance metric. Firstly, Voronoi region boundaries are not necessarily straight. Consequently, Voronoi regions are not necessarily convex polygons. Secondly, Voronoi regions associated with a given node are not necessarily contiguous, creating "orphan" regions.

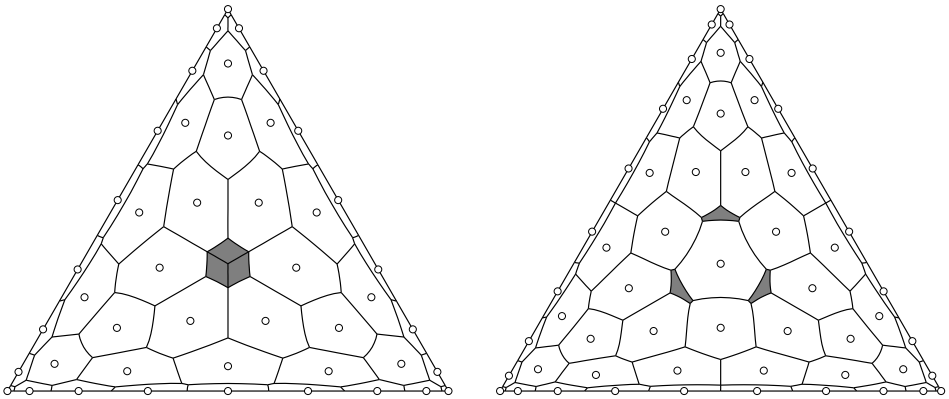


(a) Isometric (b) Fekete

Figure 7.3: Corresponding Lagrangian Voronoi tessellations for the isometric and Fekete nodal sets of Figure 7.1. Grey regions indicate "orphan cells".



(a) 7th degree Isometric nodes (b) 12th degree Electrostatic nodes



(a) 8th degree Fekete nodes (b) 9th degree Fekete nodes

Figure 7.4: Some additional, interesting Lagrangian Voronoi tessellations.

> **Definition:** An "orphan" region is a Voronoi region which does not contain its corresponding parent node.

Orphan regions are indicators of a non-physical influence. The node associated with the orphan region is exerting an influence at a distance — in other words, the node is overreaching itself. Consider the isometric nodes of Figures 7.3a and 7.4a. The orphan regions near the edges are manifestations of the low density of nodes near the edges, allowing nodes far away to "dominate" these regions. The situation rapidly deteriorates as the polynomial degree increases with an ever-increasing number of complex orphan regions near the edges, as evidence of the Runge phenomenon (§1.2).

However, even fairly good nodal sets may have orphan regions. The Lagrangian Voronoi diagrams in Figures 7.4(c,d) represent the degree 8 and 9 Fekete nodes which have achieved the global maximum of their corresponding Vandermonde matrix determinant (Chapter 4). While achieving this global maximum, they are not the optimal nodal set in terms of the Lebesgue norm Λ, the maximum of the Lebesgue function (§2.6). In fact, the location of the orphan regions roughly coincide with the location of the maximum of this Lebesgue function.

Interestingly, the area of a Lagrangian Voronoi region R_i is roughly equal to its corresponding integration weight w_i (§2.5), except in the presence of orphan regions.

7.4 Centroidal Voronoi Tessellations

A feature of Lagrangian Voronoi tessellations of nodes with poor interpolatory qualities (e.g. Figures 7.3a, 7.4a, 7.4b) is the off-centre location of the nodes within their respective Voronoi regions. These Lagrangian Voronoi tessellations typically exhibit undesirable orphan regions. For example, the fringe-like orphan regions of Figure 7.4b, corresponding to a degree 12 set of electrostatic nodes [29], are all associated with the three interior nodes closest to the three corner vertices. These three interior nodes are not centred in their Voronoi regions, in contrast to the rest of the nodes. Adjusting these nodes toward the centres of their respective Voronoi regions eliminates the orphan regions.

This result suggests an algorithm for improving a nodal set: move each node to the centre of their respective Voronoi cells until the nodes equilibrate. In fact, this is Lloyd's method [41] for the construction of "centroidal" Voronoi tessellations [15].

> **Definition:** A Voronoi tessellation is *centroidal* if the nodes of a Voronoi diagram are located at the centroids, or centres of mass, of their respective Voronoi regions.

Efficient algorithms exist [15, 37] to compute centroidal Voronoi tessellations. However, I suggest there is really no need to compute them. Consider the computation of Fekete nodes (Chapter 4). One simple method is to move the nodes to the tops of their respective Lagrangians, an analogous algorithm to Lloyd's method. In a sense, Lagrangian Voronoi tessellations with Fekete nodes *are* centroidal, at least qualitatively.

Chapter 8

Simplex–Hypersphere Transform

I introduce a transformation between the d-dimensional analogue of the triangle, the simplex, and the $(d+1)$-dimensional analogue of the sphere, the hypersphere. I show that points distributed uniformly on the surface of this hypersphere are distributed upon transformation to the simplex to a density equal to its extremal measure [4]. This extremal measure has been conjectured [64] to represent a limiting density of nodes required for the uniform convergence of interpolation in the simplex as the degree of polynomial approximation increases to infinity.

The proposed transformation permits us to define a geodesic distance measure suitable for interpolation in the simplex. This definition allows the construction of geodesic Voronoi tessellations [46] and the calculation of interpolation mesh norms. I show that, in one-dimension, points evenly spaced on the quarter circle are, upon transformation, the Gauss–Lobatto–Chebyshev points [32]. I thus conjecture that points distributed uniformly on the surface of the hypersphere are their multidimensional analogue in the simplex. As evidence of this conjecture, I show that points with excellent interpolation qualities, the two-dimensional Fekete nodes (Chapter 4) in the triangle, are distributed uniformly on the sphere. I also introduce a new suite of nodes based on weighted spherical averaging [12] which also exhibit good interpolatory qualities. This conjecture provides a linkage between the more advanced body of research devoted to distributing points on the sphere with the equivalent problem in the triangle.

8.1 Introduction

Any point in a d-dimensional simplex may be defined in terms of $d+1$ barycentric coordinates (c.f. §2.1) such that

$$\sum_{i=1}^{d+1} x_i = 1. \tag{8.1}$$

This equation traces out a hyperplane in dimension $d+1$. If we transform these barycentric coordinates, coordinate by coordinate, such that

$$X_i = \sqrt{x_i}, \tag{8.2}$$

the new coordinates obviously map to the surface of a hypersphere defined by

$$\sum_{i=1}^{d+1} X_i^2 = 1. \tag{8.3}$$

As the barycentric coordinates x_i are all positive, the transformed coordinates X_i are also positive, thus restricting points to a positive subset of the hypersphere surface.

Theorem 1. *Points distributed uniformly on the surface of the hypersphere* (8.3) *upon transformation to the hyperplane* (8.1) *are distributed with density*

$$\rho_x = \prod_{i=1}^{d+1} \frac{1}{2\sqrt{x_i}}. \tag{8.4}$$

Proof. We first must introduce the concept of a multidimensional surface "area" measure. In two and three dimensions, this measure represents the familiar arc length and surface area, respectively,

$$\mathrm{d}S_X = \left(\left(\frac{\partial Y}{\partial X} \right)^2 + 1 \right)^{1/2} \mathrm{d}X, \tag{8.5}$$

$$\mathrm{d}S_X = \left(\left(\frac{\partial Z}{\partial X} \right)^2 + \left(\frac{\partial Z}{\partial Y} \right)^2 + 1 \right)^{1/2} \mathrm{d}X\,\mathrm{d}Y. \tag{8.6}$$

Extending this result to $(d+1)$ dimensions yields the less familiar hypersurface measure

$$\mathrm{d}S_X = \left(\sum_{i=1}^{d+1} \left(\frac{\partial X_{d+1}}{\partial X_i} \right)^2 \right)^{1/2} \prod_{i=1}^{d} \mathrm{d}X_i. \tag{8.7}$$

In the particular case of the hypersphere, we may differentiate (8.3) to determine the surface gradient,

$$\frac{\partial X_{d+1}}{\partial X_i} = -\frac{X_i}{X_{d+1}}. \tag{8.8}$$

Our hypersurface measure then reduces to

$$\mathrm{d}S_X = \left(\frac{1}{X_{d+1}^2} \sum_{i=1}^{d+1} X_i^2 \right)^{1/2} \prod_{i=1}^{d} \mathrm{d}X_i. \tag{8.9}$$

which, recognizing (8.3), further simplifies to

$$\mathrm{d}S_X = \frac{1}{X_{d+1}} \prod_{i=1}^{d} \mathrm{d}X_i. \tag{8.10}$$

If we apply our transformation,

$$X_i = \sqrt{x_i} \quad \Rightarrow \quad \mathrm{d}X_i = \frac{\mathrm{d}x_i}{2\sqrt{x_i}} \tag{8.11}$$

our elemental surface area becomes

$$\mathrm{d}S_X = \prod_{i=1}^{d+1} \frac{1}{2\sqrt{x_i}} \prod_{i=1}^{d} \mathrm{d}x_i = \rho_x \, \mathrm{d}S_x. \tag{8.12}$$

So, points in X-space distributed uniformly over S_X upon transformation to x-space will be distributed to density (8.4) over S_x. □

This theorem would not be particularly useful but for the fact the resulting density (8.4) is the extremal measure for the simplex [4], hereafter the Chebyshev density. This extremal measure is important as it is conjectured [7] that it represents the asymptotic density of a certain class of excellent interpolatory points — the Fekete nodes — as the polynomial degree approaches infinity (see Chapter 4). Certainly, this conjecture is valid in one-dimension as both the Gauss–Lobatto–Chebyshev and Gauss–Lobatto–Legendre nodes (see §2.8) approach this limit. We provide visual and numerical evidence of this conjecture in §8.4.

8.2 Random Chebyshev Points

Corollary 1. *Uniform random points distributed on the surface of the hypersphere* (8.3) *will be randomly distributed on the hyperplane* (8.1) *with probability density function* (8.4).

In Appendix A, I determined a method to distribute random vectors in the triangle with density (8.4). Here I derive the same result by distributing points uniformly on the sphere and transforming them to the triangle.

The differential area on a unit-radius sphere in terms of latitude θ and longitude ϕ is

$$\mathrm{d}A = \cos\theta \,\mathrm{d}\theta \,\mathrm{d}\phi. \tag{8.13}$$

In Cartesian space, $Z = \sin\theta$ such that $\mathrm{d}Z = \cos\theta \,\mathrm{d}\,\theta$. Our differential area becomes

$$\mathrm{d}A = \mathrm{d}Z \,\mathrm{d}\phi. \tag{8.14}$$

Thus in determining a random point on the sphere, we merely pick a random point (Z, ϕ) uniformly in the square $[-1, 1] \times [0, 2\pi)$. The other Cartesian coordinates are then

$$X = \sqrt{1 - Z^2}\cos(\phi) \tag{8.15}$$

$$Y = \sqrt{1 - Z^2}\sin(\phi) \tag{8.16}$$

We limit points to the positive octant, $X, Y, Z \geq 0$, by only generating uniform random points (Z, ϕ) in $[0, 1] \times [0, \pi/2]$. Applying the transformation yields the barycentric coordinates

$$x = X^2 = (1 - Z^2)\cos^2(\phi) \tag{8.17}$$

$$y = Y^2 = (1 - Z^2)\sin^2(\phi) \tag{8.18}$$

$$z = Z^2 \tag{8.19}$$

which correspond directly with (A.14).

An alternative method [43], suitable for *all* dimensions, starts with the generation of $d + 1$ Gaussian/normal random variates G_i with mean zero and variance one. Normalizing, the variates project uniformly onto the hypersphere. Transforming, barycentric coordinates distributed to (8.4) in the d-simplex are then

$$x_i = \frac{G_i^2}{\sum_{j=1}^{d+1} G_j^2}. \tag{8.20}$$

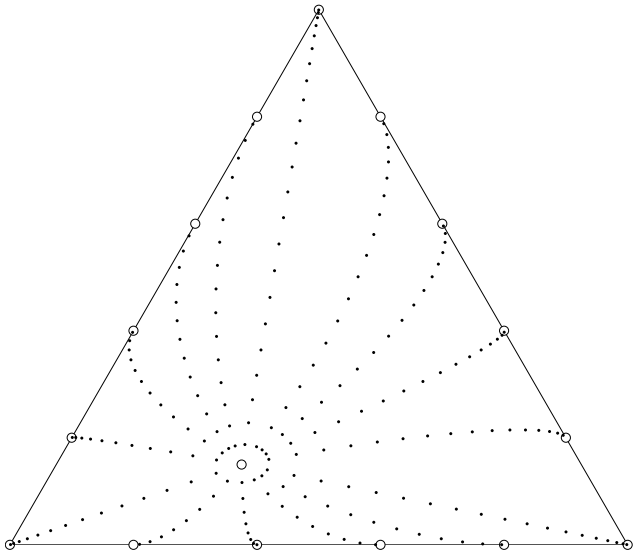


Figure 8.1: Geodesic paths between points in the triangle. The geodesic distance (angle on the sphere) between dots is constant along each path.

8.3 Geodesic Distance Metric

Given any two vectors A and B describing points on the unit hypersphere, we may calculate the angle δ between them as

$$\cos\delta = \sum_{i}^{d} A_i B_i. \tag{8.21}$$

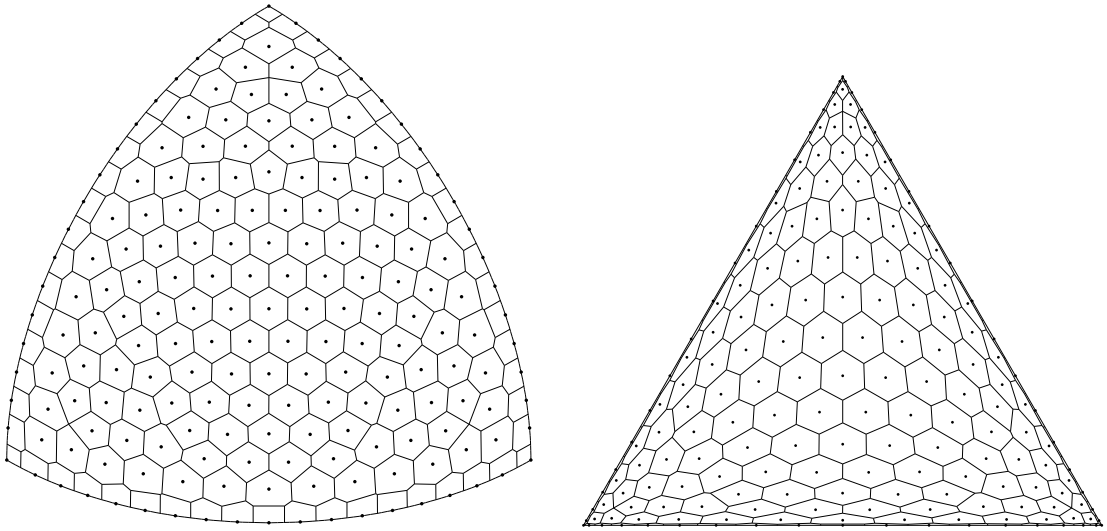
This angle is also the shortest distance, or geodesic, between A and B. In two dimensions this geodesic is the arc length while in three dimensions this geodesic is the great circle distance. By applying our transformation, we may also determine a geodesic distance for the simplex,

$$\cos\delta = \sum_{i}^{d} \sqrt{a_i}\sqrt{b_i}. \tag{8.22}$$

In Figure 8.1, I plot the geodesic rays between a point in the interior of a triangle and selected points on its exterior. The distance intervals between dots are equal on the sphere but vary when transformed to the triangle, shrinking and deflecting as the geodesic rays travel through the denser medium near the edges. Interestingly, the geodesic joining any point in the triangle and any of the triangle vertices is straight.

1. Transform barycentric coordinates x_i to the sphere X_i as per §8.1.
2. Compute their convex hull [e.g. 5].
3. Calculate the Voronoi vertices by calculating the facet normals.
4. Join adjacent Voronoi vertices to form the Voronoi diagram.
5. Transform diagram back to barycentric space.

Figure 8.2: Chebyshev Voronoi tessellation algorithm



(a) Sphere (b) Triangle

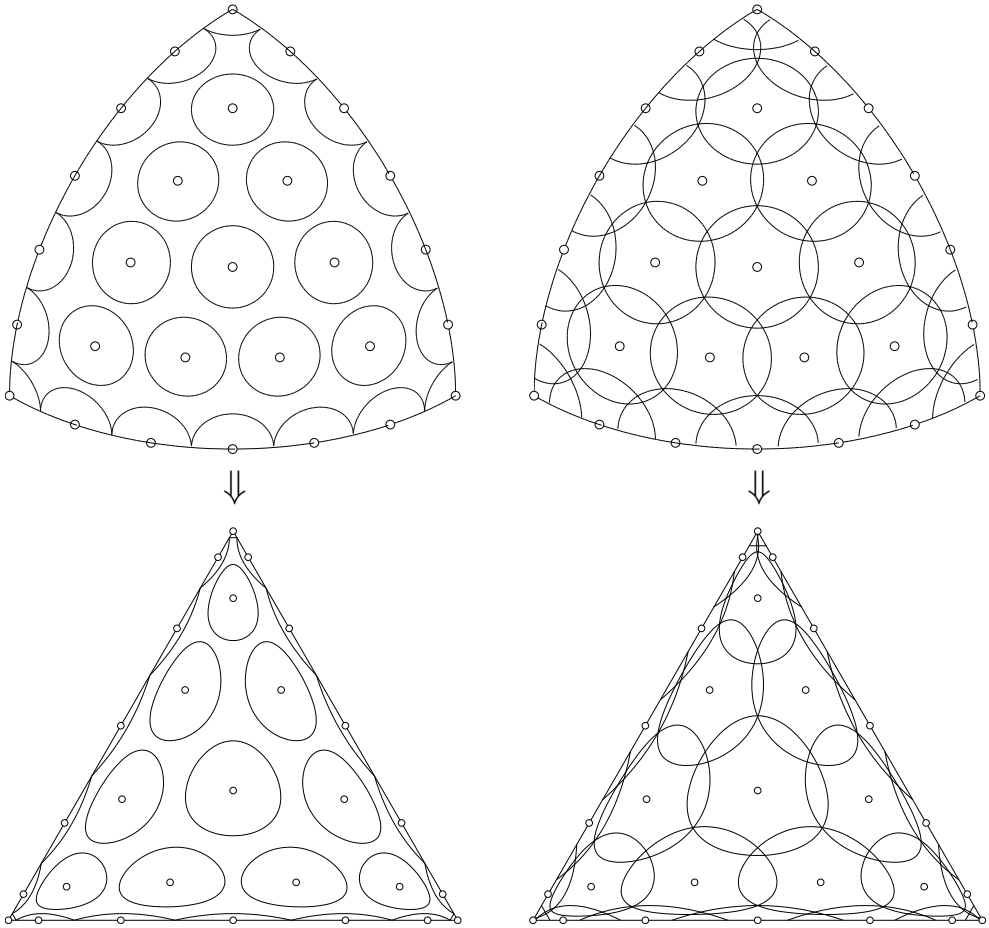
Figure 8.3: Chebyshev Voronoi tessellation of a set of degree 18 Fekete nodes.

8.3.1 Chebyshev Voronoi Tessellations

Armed with our geodesic metric we may construct Voronoi tessellations pertinent to the simplex. As they are based on a transformation inspired by the Chebyshev density (8.4), we call them Chebyshev Voronoi tessellations. Here we concentrate on the triangle but equivalent tessellations can be constructed for higher dimensions.

In §7.1, I defined the Voronoi tessellation. If we replace the usual Euclidean distance measure with our geodesic measure, we complete the definition of a Chebyshev Voronoi diagram. It remains to describe a simple algorithm for its construction which I summarize in Figure 8.2.

Simply put, given points in the triangle, I transform them to the sphere, form the Voronoi diagram, and transform the diagram back to the triangle. The construction of the Voronoi diagram on the sphere is well-known. The convex hull [e.g. 5] yields



(a) Packing (b) Covering

Figure 8.4: Covering and packing of spherical caps on the sphere and subsequent transformation to the triangle.

a tessellation of the sphere surface into triangular facets — the graph dual is then the Voronoi diagram with the Voronoi vertices located at the circumcentres of each triangle facet. If P, Q, and R are three vertices of this facet, the circumcentre is [50]

$$C = \frac{(Q-P) \wedge (R-P)}{|(Q-P) \wedge (R-P)|}, \tag{8.23}$$

which is simply the facet normal projected to the sphere surface. The Voronoi graph is then formed by joining neighbouring Voronoi vertices. In Figure 8.3, I plot the Chebyshev Voronoi diagram both on the sphere and in the triangle for a degree 18 Fekete node distribution of points.

8.3.2 Interpolation Mesh Norms

A spherical cap is the region within a given geodesic distance or radius from a point on the sphere. We are interested in two spherical cap radii:

> **Packing radius:** The minimum radius of n identical and non-overlapping spherical caps centred about n nodes. In a triangulation of the sphere, this radius is half the length of the shortest edge. This radius is also known as the kissing radius. See Figure 8.4a.
>
> **Covering radius:** The maximum radius of n identical and overlapping spherical caps centred about n nodes which completely cover the sphere. In a triangulation of the sphere, this radius is the largest triangle circumradius. This radius is also known as the mesh norm. See Figure 8.4b.

Either maximizing the packing radius or minimizing the covering radius lead to node distributions which attempt to equidistribute over the sphere. These radii are thus indicators of mesh uniformity.

8.4 Chebyshev Conjecture

The Chebyshev Voronoi diagram of Figure 8.3 exhibits Voronoi regions which, at least visually, are of uniform size. As this underlying nodal distribution has excellent interpolatory qualities, I make the following loosely-worded conjecture:

Conjecture 1. *Points with good interpolatory qualities in the d-dimensional simplex are distributed uniformly over the surface of the positive octant of the* $(d+1)$*-dimensional hypersphere.*

I provide two pieces of evidence for this conjecture: first, in one dimension, points distributed evenly over the quarter-circle are the Gauss–Lobatto–Chebyshev (§2.8) points, and second, in two dimensions, Fekete node distributions have excellent mesh norms. I also provide evidence that the converse statement is not necessarily valid.

8.4.1 Gauss–Lobatto–Chebyshev Nodes

Consider $n = p + 1$ points distributed evenly in polar angle θ over the unit quarter circle (see Figure 8.5),

$$\{X_1, X_2\} = \{\cos\theta, \sin\theta\}, \quad \theta = (i\pi)/(2p), \quad i = 0 \ldots p. \tag{8.24}$$

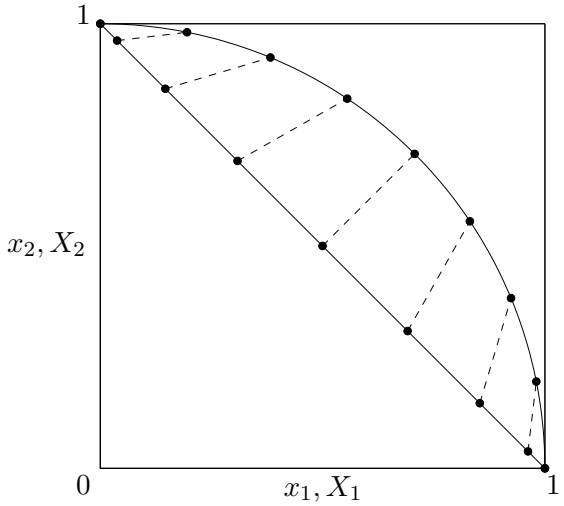


Figure 8.5: Points distributed evenly on the quarter circle and transformed are distributed as Gauss–Lobatto–Chebyshev points.

If we transform to barycentric coordinates, the nodal set is

$$\{x_1, x_2\} = \{\cos^2\theta, \sin^2\theta\}. \tag{8.25}$$

If translate x_2 from the interval $[0, 1]$ to $[-1, 1]$, we have

$$x = 2x_2 - 1 = 2\sin^2\theta - 1 = -\cos(2\theta) = -\cos\left(i\pi/p\right). \tag{8.26}$$

Thus we have the Gauss–Lobatto–Chebyshev nodes (c.f. §2.8).

8.4.2 Fekete Nodes

For the triangle, we seek to distribute points evenly on the sphere, or at least the positive octant. However, it is topologically impossible to tessellate the surface of a sphere uniformly [53], except in the special instance of the Platonic solids (tetrahedron, octahedron, cube, icosahedron, and dodecahedron). Further, the concept of "uniformity" is vague — we must first define some measure of equidistribution and then optimize.

As discussed previously, either the minimization of the packing radius (Tammes problem) or maximization of the covering radius would provide some sense of uniformity. Alternative methods which have proved successful include electrostatic potential energy minimization [23] (Thomson problem), Vandermonde determinant maximization [49, 58] (extremal fundamental systems or Fekete points), Lebesgue norm minimization [66], or the generation of centroidal Voronoi diagrams [16]. Any of the above methods could be applied to our problem by either restricting attention to the

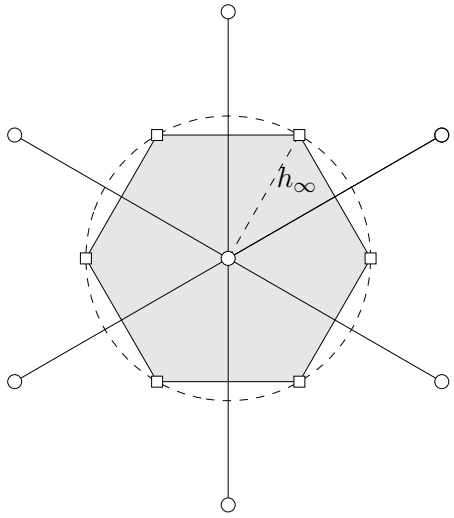


Figure 8.6: A stencil corresponding to an optimal hexagonal packing of nodes. The open circles represent nodes while the open squares represent Voronoi vertices. The shaded hexagon is the Voronoi region of the centre node.

positive sphere octant or by applying symmetry across the 8 octants.

However, we already possess a suite of nodes which meet our requirements; the Fekete nodes (see Chapter 4) have excellent interpolation qualities in that they satisfy $\lim_{p\to\infty} \Lambda^{1/p} = 1$ (see §2.6). Given a measure of uniformity, we may compare the Fekete nodal set to the isometric (see §2.9) nodal set to test the consistency of my conjecture.

In the plane, the optimal arrangement of nodes is a hexagonal-shaped configuration, where each node has six neighbours (see Figure 8.6) and the Voronoi vertices form hexagonal Voronoi regions [20]. Unfortunately, the Euler constraint relating edges, faces, and vertices in a graph precludes a spherical tessellation of all hexagons [53]. However, if we assume a uniform tiling of the sphere with hexagons we may get a lower limit on our mesh norm h_∞ as the number of nodes becomes large.

Consider Figure 8.6. Our mesh norm is the distance between the centre node and any Voronoi vertex and represents the radius of a circle circumscribing the shaded Voronoi region. While this region is actually a spherical polygon, in the limit as the number of nodes becomes large, it is well approximated by a planar hexagon with area

$$A = \frac{3\sqrt{3}}{2} h_\infty^2. \tag{8.27}$$

We require $n = (p+1)(p+2)/2$ nodes $\{X_i, Y_i, Z_i\}$ for a degree p polynomial approximation in the triangle. However, these nodes, when transformed, only cover one-eighth of the sphere. To cover the entire sphere we require nodes $\{\pm X_i, \pm Y_i, \pm Z_i\}$ for a total

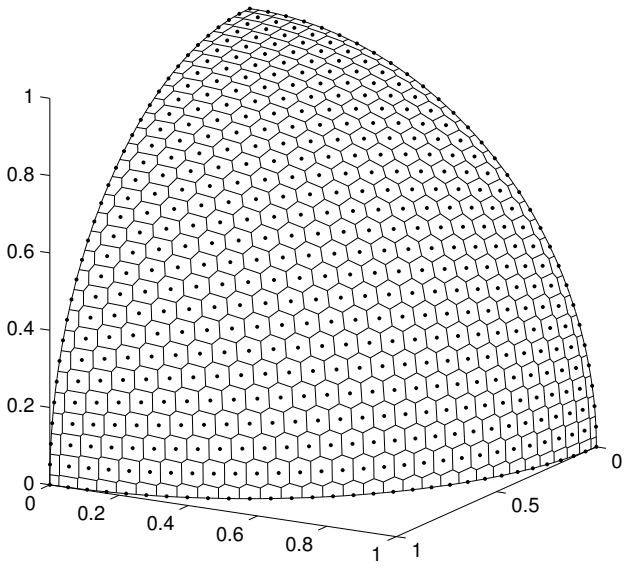


Figure 8.7: Buss–Fillmore nodes: 3602 points, corresponding to a degree 30 polynomial, uniformly distributed via spherical averages.

of $m = 4p^2 + 2$ nodes. Equating m copies of A with the total surface area of the sphere 4π leads to an optimal bound on the mesh norm,

$$h_\infty = \sqrt{\frac{8\pi}{3\sqrt{3}m}}. \tag{8.28}$$

Therefore, if mesh norm is our measure of uniformity, we seek $h/h_\infty \to 1$ as $p \to \infty$. In Table 8.1, I list this quantity alongside $\Lambda^{1/p}$. The Fekete nodes see a superior Lebesgue and mesh norm trend in comparison to the isometric nodes, consistent with the conjecture. This trend in mesh norm also provides numerical evidence for the conjecture of Bloom *et al.* [7] that Fekete nodes are distributed as (8.4).

8.4.3 Buss–Fillmore Nodes

I now examine the converse of the conjecture: do nodes distributed uniformly on the sphere necessarily have good interpolatory qualities? I thus seek a nodal set with superior mesh norms to Fekete nodes. Instead of implementing one of the methods discussed in the previous section, I introduce a new nodal set based on spherical averages [12].

The spherical average of two points on the sphere X_1 and X_2 with weights w_1 and w_2 such that $w_1 + w_2 = 1$ is found through the "slerp" [56] operator,

$$X = \frac{\sin(w_2\delta)}{\sin\delta} X_1 + \frac{\sin(w_1\delta)}{\sin\delta} X_2, \tag{8.29}$$

			Isometric		Fekete		Buss–Fillmore	
p	n	m	$\Lambda^{1/p}$	h/h_∞	$\Lambda^{1/p}$	h/h_∞	$\Lambda^{1/p}$	h/h_∞
1	3	6	1.000	1.064	1.000	1.064	1.000	1.064
2	6	18	1.291	1.187	1.291	1.187	1.291	1.187
3	10	38	1.314	1.192	1.283	1.133	1.284	1.115
4	15	66	1.365	1.341	1.285	1.153	1.318	1.094
5	21	102	1.404	1.481	1.293	1.167	1.369	1.079
6	28	146	1.435	1.611	1.269	1.176	1.333	1.074
7	36	198	1.463	1.733	1.256	1.183	1.321	1.067
8	45	258	1.488	1.847	1.249	1.188	1.322	1.065
9	55	326	1.510	1.954	1.237	1.192	1.307	1.062
10	66	402	1.531	2.057	1.229	1.409	1.302	1.061
11	78	486	1.551	2.154	1.207	1.236	1.304	1.059
12	91	578	1.568	2.248	1.195	1.455	1.298	1.059
13	105	678	1.585	2.338	1.187	1.469	1.297	1.057
14	120	786	1.600	2.424	1.178	1.226	1.300	1.057
15	136	902	1.614	2.508	1.166	1.265	1.298	1.056
16	153	1026	1.627	2.589	1.160	1.265	1.300	1.056
17	171	1158	1.640	2.667	1.155	1.469	1.302	1.055
18	190	1298	1.651	2.743	1.154	1.251	1.303	1.055
19	210	1446	1.662	2.818	1.143	1.476	1.306	1.054
20	231	1602	1.672	2.890	1.146	1.275	1.309	1.054
21	253	1766	1.681	2.961	1.141	1.288	1.315	1.054
22	276	1938	1.690	3.029	1.133	1.274	1.321	1.054
23	300	2118	1.698	3.097	1.128	1.284	1.328	1.053
24	325	2306	1.706	3.163	1.124	1.247	1.334	1.053
25	351	2502	1.713	3.227	1.121	1.280	1.340	1.053
26	378	2706	1.720	3.291	1.122	1.281	1.347	1.053
27	406	2918	1.727	3.353	1.118	1.277	1.353	1.053
28	435	3138	1.733	3.414	1.110	1.250	1.358	1.053
29	465	3366	1.739	3.474	1.112	1.294	1.364	1.053
30	496	3602	1.745	3.533	1.109	1.280	1.369	1.052

Table 8.1: For the isometric (§2.9), Fekete (Chapter 4), and Buss–Fillmore (§8.4.3) nodal sets, I list the Lebesgue norm Λ and mesh norm h as a function of polynomial degree p, corresponding number of nodes in the triangle n and sphere m. We seek $\Lambda^{1/p} \to 1$ for convergence and $h/h_\infty \to 1$ for uniformity.

where δ is the geodesic distance (angle) between X_1 and X_2. However, spherical interpolation between points on the sphere is non-trivial for more than two points.

Buss and Fillmore [12] introduced a heuristic which generalizes the slerp operator to many points. Here, the centroid of a set of points with varying weights on the sphere is calculated by transforming the points to a plane tangent to the sphere. The transformation, an exponential map, both preserves geodesic distance and an-

gle. Thus, a simple weighted average of points in this tangent plane represents the centroid of the points in this plane. If this plane centroid and the point of tangency coincide, this point is also the spherical average. This suggests an obvious algorithm which, from a guessed location, forms the tangent plane, maps points onto this plane, calculates the planar centroid, and reverse maps this centroid back to the sphere. With iteration, this algorithm converges on the spherical average. A faster algorithm [12] based on Newton–Raphson iteration returns a quadratically convergent spherical average.

In our case, we wish to distribute nodes evenly over the sphere octant $X_i \geq 0$. Thus, we fix three points on each axis at $(1,0,0)$, $(0,1,0)$, and $(0,0,1)$ and vary the weights according to the barycentric coordinates of an underlying isometric grid. That is,

$$(w_1, w_2, w_3) = (i, j, k)/p, \tag{8.30}$$

where the indices are integers

$$\{(i,j,k) \mid i,j,k \geq 0, i+j+k = p\}. \tag{8.31}$$

We then apply the Buss–Fillmore heuristic to find the spherical average for each set of weights. The result is a degree p set of nodes which are uniformly distributed on the octant (see e.g. Figure 8.7).

Revisiting Table 8.1, we see a significantly improved mesh norm for the Buss–Fillmore nodes relative to the Fekete nodes; however, not only do we see an inferior Lebesgue norm, the trend in $\Lambda^{1/p}$ is not unity. That is, equidistribution of points on the sphere is not sufficient to ensure nodes with good interpolatory qualities.

However, perhaps our conclusion is hasty. It has been shown that an isometric-like triangulation of nodes does not lead to optimal Lebesgue norms (see Chapter 4). Indeed, the Lebesgue norm of set of degree 30 Fekete nodes with isometric-like triangulation (see Figure 4.7c) is of the same order of magnitude as that of the Buss–Fillmore nodes. It is entirely possible that a set of nodes uniformly distributed but with an irregular triangulation could be found with an acceptable Lebesgue norm trend.

It should also be noted that the Buss–Fillmore framework extends to the hyper-sphere allowing the quick generation of interpolation points in the simplex.

Chapter 9

Conclusion

In this dissertation, I investigated the optimal nodal distribution of nodes in the triangle with a view to improving the accuracy of polynomial interpolation. While focused on integration with the spectral element method, much of the research in the form of new optimization algorithms and new geometric concepts has more general applicability. In this chapter I summarize my contributions and finish back where we started — the Runge phenomenon

9.1 Dissertation Contributions

The following are the main contributions of this dissertation:

1. **Dividing Triangles Algorithm** (Chapter 3): I have discovered a new global optimization algorithm for finding the extrema of a continuous function in a single triangle or an assemblage of triangles.

2. **Optimal Nodal Distributions**: I have discovered through the development of various new numerical algorithms and transformations many new high-order nodal distributions for interpolation in the triangle and suitable for inclusion in existing spectral element methods. Each class of nodes are the best discovered to date in terms of Lebesgue norm. While the focus was the triangle, most of the algorithms have applicability to higher dimensions and other geometries:

   (a) **Pseudo-Legendre** (§4.2.1): *Explicit* nodal configurations with excellent Lebesgue norms for low degrees.

   (b) **Fekete** (Chapter 4): Nodal configurations through to degree 30 such that the Vandermonde matrix determinant is maximized.

   (c) **Chen–Babuška** (Chapter 5): Nodal configurations through to degree 30 such that the integrated sum of Lagrangian squares is minimized.

(d) **Lebesgue** (Chapter 6): Putatively optimal nodal configurations up to degree 15 found through directly optimizing the Lebesgue norm.

(e) **Buss–Fillmore** (§8.4.3): Nodal configurations with excellent mesh norms found through distributing points uniformly on the sphere.

3. **Simplex to Hyperphere Transformation** (Chapter 8): I have discovered a transformation between the simplex and the hypersphere which links the problem of distributing points uniformly on the sphere to the problem of distributing points to the Chebyshev density in the simplex. This transformation allowed the definition of a distance metric and, consequently, a mesh norm.

4. **Random Chebyshev Points in the Simplex** (§8.2, Appendix A): Through the above transformation, I have derived a method to generate random variates with a Chebyshev probability density function in the simplex.

5. **Voronoi Tessellations**: I have defined and constructed Voronoi tessellations pertinent to polynomial interpolation:

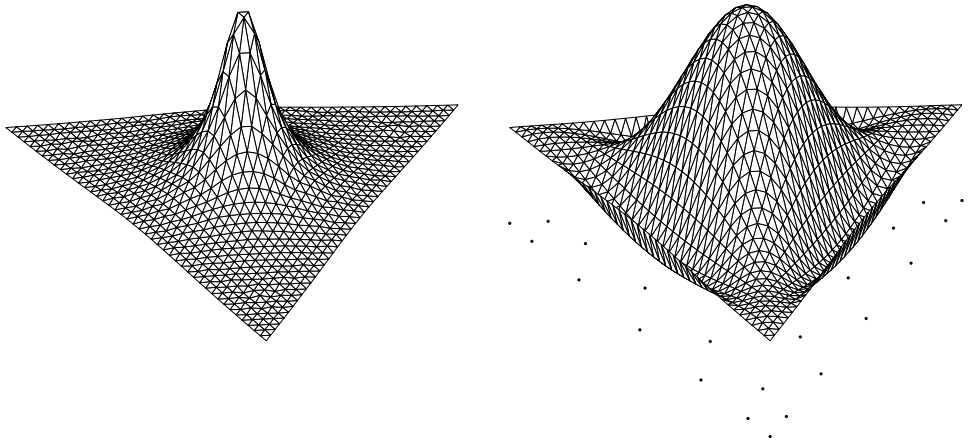
   (a) **Lagrangian** (Chapter 7): A distance metric based on the magnitude of the Lagrangians with applicability to higher dimensions and other geometries.

   (b) **Chebyshev** (§8.3.1): A distance metric based on geodesics (great circles) on the sphere with applicability to higher dimensional simplices.

9.2 Runge Phenomenon Revisited

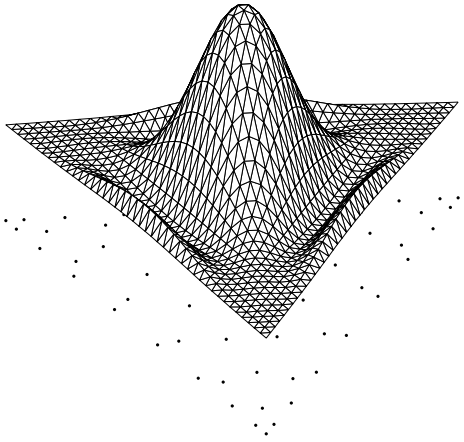
In §1.2 we saw the dangers of an isometric distribution of nodes for polynomial interpolation in the triangle. Here we revisit our triangular analogue of the Witch of Agnesi (1.3) but interpolate over the Fekete nodes of Chapter 4.

In Figure 9.1, I plot the resulting approximations through to degree 30. While initially there are oscillations, as the degree increases, they die out. The degree 30 approximation is indistinguishable from the exact function.

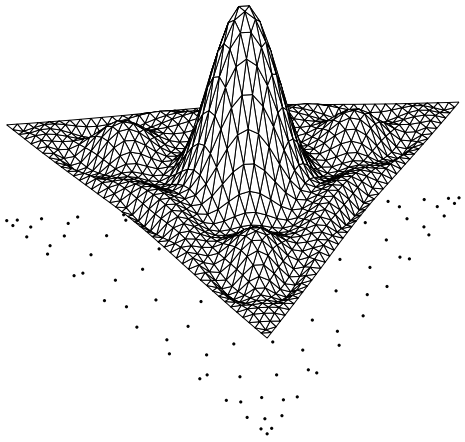


(a) Witch of Agnesi

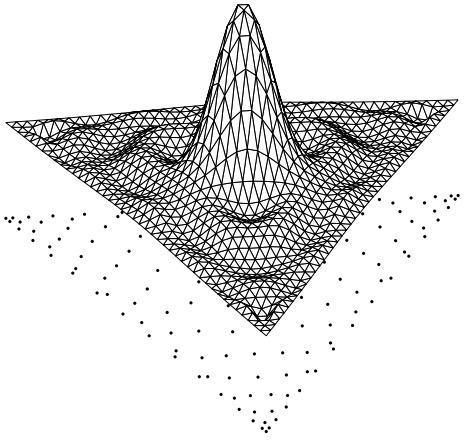
(b) Degree 6



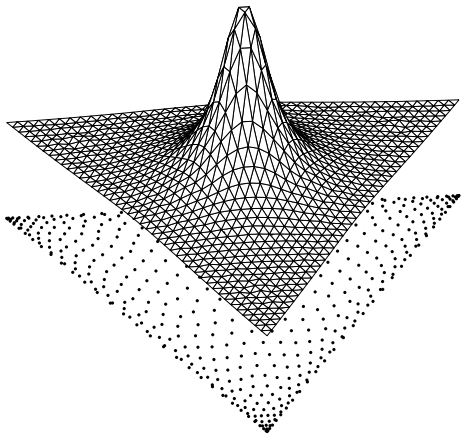
(c) Degree 9



(d) Degree 12



(e) Degree 15



(f) Degree 30

Figure 9.1: Various degree Fekete node approximations to the Runge function (1.3). The polynomial curve fits (b)–(f) now converge to the "Witch of Agnesi" (a).

List of References

[1] Angelos, J. R., J. E. H. Kaufman, M. S. Henry, and T. D. Lenker. Optimal nodes for polynomial interpolation. In C. K. Chui, L. L. Schumaker, and J. D. Ward, editors, *Approximation Theory VI*, volume 1, pages 17–20. Academic Press, 1989.

[2] Aris, R. *Vectors, Tensors, and the Basic Equations of Fluid Mechanics.* Dover, 1989.

[3] Aurenhammer, F. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3), 345–405, 1991.

[4] Baran, M. Complex equilibrium measure and Bernstein type theorems for compact sets in R^n. *Proceedings of the American Mathematical Society*, 123(2), 485–494, 1995.

[5] Barber, C. B., D. P. Dobkin, and H. Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4), 469–483, 1996.

[6] Bloom, T. The Lebesgue constant for Lagrange interpolation in the simplex. *Journal of the Atmospheric Sciences*, 54, 338–353, 1988.

[7] Bloom, T., L. Bos, C. Christensen, and N. Levenberg. Polynomial interpolation of holomorphic functions in C and C^n. *Rocky Mountain Journal of Mathematics*, 22(2), 441–469, 1992.

[8] Bos, L. Bounding the Lebesgue function for Lagrange interpolation in the simplex. *Journal of Approximation Theory*, 38, 43–59, 1983.

[9] Bos, L., M. A. Taylor, and B. A. Wingate. Tensor product Gauss-Lobatto points are Fekete points for the cube. *Mathematics of Computation*, 70(236), 1543–1547, 2001.

[10] Boyd, J. P. *Chebyshev and Fourier Spectral Methods.* Dover, 2nd edition, 2001.

[11] Brutman, L. On the Lebesgue function for polynomial interpolation. *SIAM Journal on Numerical Analysis*, 15(4), 1978.

[12] Buss, S. R. and J. P. Fillmore. Spherical averages and applications to spherical splines and interpolation. *ACM Transactions on Graphics*, 20(2), 95–126, 2001.

[13] Chen, Q. and I. Babuška. Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle. *Computer Methods in Applied Mechanics and Engineering*, 128, 405–417, 1995.

[14] de Boor, C. and A. Pinkus. Proof the conjectures of Bertstein and Erdös concerning the optimal nodes for polynomial interpolation. *Journal of Approximation Theory*, 24, 289–303, 1978.

[15] Du, Q., V. Faber, and M. D. Gunzburger. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4), 637–676, 1999.

[16] Du, Q., M. D. Gunzburger, and L. Ju. Constrained centroidal Voronoi tessellations for surfaces. *SIAM Journal on Scientific Computing*, 24(5), 1488–1506, 2003.

[17] Dubiner, M. Spectral methods on triangles and other domains. *Journal of Scientific Computing*, 6(4), 345–390, 1991.

[18] Dupont, F. and C. A. Lin. The adaptive spectral element method and comparisons with more traditional formulations for ocean modeling. *Journal of Atmospheric and Oceanic Technology*, 21(1), 135–147, 2004.

[19] Fejér, L. Bestimmung derjenigen Abszissen eines Intervalles für welche die Quadratsumme der Grundfunktionen der Lagrangeschen Interpolation im Intervalle $[-1, 1]$ ein möglicht kleines Maximum besitzt. *Annali della Scuola Normale Superiore di Pisa. Class di Scienze. Serie II*, 1, 263–276, 1932.

[20] Fejes Tóth, L. *Regular Figures*. Pergamon, 1964.

[21] Fekete, M. Über die Verteilung der Wurzeln bei gewissen algebraischen Gleichungen mit ganzzahligen Koeffizienten. *Mathematische Zeitschrift*, 17, 228–249, 1923.

[22] Fix, G. J. Finite element models for ocean circulation problems. *SIAM Journal on Applied Mathematics*, 29(3), 371–387, 1975.

[23] Fliege, J. and U. Maier. The distribution of points on the sphere and corresponding cubature formula. *IMA Journal of Numerical Analysis*, 19(2), 317–334, 1999.

[24] Fortune, S. Sweepline algorithms for Voronoi diagrams. *Algorithmica*, 2, 153–174, 1987.

[25] Fournier, A., M. A. Taylor, and J. J. Tribbia. The spectral element atmosphere model (SEAM): High-resolution parallel computation and localized resolution of regional dynamics. *Monthly Weather Review*, 132, 726–748, 2004.

[26] Giraldo, F. X., J. S. Hesthaven, and T. Warburton. Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations. *Journal of Computational Physics*, 181, 499–525, 2003.

[27] Golub, G. H. and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University, 3rd edition, 1996.

[28] Griffies, S. M. The Gent-McWilliams skew flux. *Journal of Physical Oceanography*, 28, 831–841, 1998.

[29] Hesthaven, J. S. From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex. *SIAM Journal on Numerical Analysis*, 35(2), 655–676, 1998.

[30] Hesthaven, J. S. and C. H. Teng. Stable spectral methods on tetrahedral elements. *SIAM Journal on Scientific Computing*, 21(6), 2352–2380, 2000.

[31] Higham, N. J. *Accuracy and Stability of Numerical Algorithms*. SIAM, 1996.

[32] Hochstrasser, U. W. Orthogonal polynomials. In M. Abramowitz and I. A. Stegun, editors, *Handbook of Mathematical Functions*, pages 771–802. Dover, 1965.

[33] Ingber, L. Simulated annealing: practice versus theory. *Mathematical Computer Modelling*, 18(11), 29–57, 1993.

[34] Iskandarani, M., D. B. Haidvogel, and J. P. Boyd. A staggered spectral element model with application to the oceanic shallow water equations. *International Journal for Numerical Methods in Fluids*, 20, 393–414, 1995.

[35] Iskandarani, M., D. B. Haidvogel, and J. C. Levin. A three-dimensional spectral element model for the solution of the hydrostatic primitive equations. *Journal of Computational Physics*, 186, 397–425, 2003.

[36] Jones, D. R., C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Application*, 79(1), 157–181, 1993.

[37] Ju, L., Q. Du, and M. D. Gunzburger. Probabilistic methods for centroidal Voronoi tessellations and their parallel implementations. *Parallel Computing*, 28, 1477–1500, 2002.

[38] Karniadakis, G. E. and S. J. Sherwin. *Spectral/hp Element Methods for CFD*. Oxford University Press, 1999.

[39] Kennedy, J. and R. Eberhart. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[40] Koornwinder, T. Two-variable analogues of the classical orthogonal polynomials. In *Theory and Application of Special Functions*, pages 435–495. Academic Press, 1975.

[41] Lloyd, S. P. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, IT-28(2), 129–137, 1982.

[42] Lyness, J. N. and R. Cools. A survey of numerical cubature over triangles. In *Proceedings of Symposia in Applied Mathematics*, volume 48, pages 127–150. AMS, 1994.

[43] Muller, M. E. A note on a method for generating points uniformly on N-dimensional spheres. *Communications of the ACM*, 2(4), 19–20, 1959.

[44] Munk, W. H. and C. Wunsch. Abyssal recipes II: energetics of tidal and wind mixing. *Deep Sea Research I*, 45, 1977–2010, 1998.

[45] Nelder, J. A. and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4), 308–313, 1965.

[46] Okabe, A., B. Boots, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, 1992.

[47] Owens, R. G. Spectral approximations on the triangle. *Proceedings of the Royal Society: Mathematical, Physical and Engineering Sciences*, 454(1971), 857–872, 1998.

[48] Pasquetti, R. and F. Rapetti. Spectral element methods on triangles and quadrilaterals: comparisons and applications. *Journal of Computational Physics*, 198, 349–362, 2004.

[49] Reimer, M. and B. Sündermann. A Remez-type algorithm for the calculation of extremal fundamental systems for polynomial spaces on the sphere. *Computing*, 37, 43–58, 1986.

[50] Renka, R. J. Algorithm 772: STRIPACK: Delaunay triangulation and Voronoi diagram on the surface of a sphere. *ACM Transactions on Mathematical Software*, 23(3), 416–434, 1997.

[51] Rubinstein, R. Y. *Simulation and the Monte-Carlo Method*. Wiley, 1981.

[52] Runge, C. Überempirische Functionen und die Interpolation zwischen äquidistanten Ordinaten. *Zeitschrift für Mathematik und Physik*, 46, 224–243, 1901.

[53] Saff, E. B. and A. B. J. Kuijlaars. Distributing many points on a sphere. *The Mathematical Intelligencer*, 19(1), 5–11, 1997.

[54] Sethian, J. A. Fast marching methods. *SIAM Review*, 41(2), 199–235, 1999.

[55] Shewchuk, J. R. Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In M. Lin and D. Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, pages 203–222. Springer-Verlag, 1996.

[56] Shoemake, K. Animating rotation with quaternion curves. *ACM SIGGRAPH Computer Graphics*, 19(3), 245–254, 1985.

[57] Silvester, P. High-order polynomial triangular finite elements for potential problems. *International Journal of Engineering Science*, 7(8), 849–861, 1969.

[58] Sloan, I. H. and R. S. Womersley. Extremal systems of points and numerical integration on the sphere. *Advances in Computational Mathematics*, 21(1), 107–125, 2004.

[59] Soluri, E. A. and V. A. Woodson. World vector shoreline. *International Hydrographic Review*, LXVII(1), 27–36, 1990.

[60] Stieltjes, T. J. Sur les polynômes de Jacobi. *Comptes Rendus de l'Acadèmie des Sciences*, 100, 620–622, 1885.

[61] Stieltjes, T. J. Sur quelques théorèmes d'algèbre. *Comptes Rendus de l'Acadèmie des Sciences*, 100, 439–440, 1885.

[62] Storn, R. and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341–359, 1997.

[63] Taylor, M. A., J. J. Tribbia, and M. Iskandarani. The spectral element method for the shallow water equations on the sphere. *Journal of Computational Physics*, 130, 92–108, 1997.

[64] Taylor, M. A., B. A. Wingate, and R. E. Vincent. An algorithm for computing the Fekete points in the triangle. *SIAM Journal on Numerical Analysis*, 35(5), 1707–1720, 2000.

[65] Wales, D. J. and J. P. K. Doye. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *Journal of Physical Chemistry A*, 101, 5111–5116, 1997.

[66] Womersley, R. S. A continuous minimax problem for calculating minimum norm polynomial interpolation points on the sphere. *Australian & New Zealand Industrial and Applied Mathematics Journal*, 42(E), C1536–C1557, 2000.

[67] Xu, S. Smoothing method for minimax problems. *Computational Optimization and Applications*, 20, 267–279, 2001.

Appendix A

Random Chebyshev Points in the Triangle

In order to generate a series of random Fekete points (e.g. Figure A.1) in the triangle, I present an inverse transform solution that follows directly from a worked example in Rubinstein [51, pg. 60].

The joint probability density function (pdf) of two dependant variables x_1 and x_2 is

$$f_{1,2}(x_1, x_2) = f_1(x_1) f_2(x_2|x_1), \tag{A.1}$$

where X_1 and X_2 are the desired random variates, $f_1(x_1)$ is the marginal pdf of X_1, and $f_2(x_2|x_1)$ is the conditional pdf of X_2 given $X_1 = x_1$. If the U_1 and U_2 are independent uniformly distributed variates on the interval $[0, 1]$, then the vector

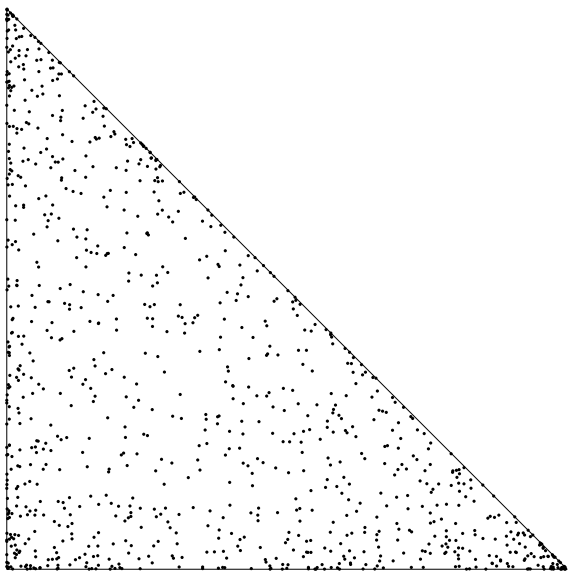


Figure A.1: 1000 random Chebyshev points in the triangle.

$X = (X_1, X_2)$ may be generated from the solution of

$$F_1(X_1) = U_1 \tag{A.2}$$

$$F_2(X_2|X_1) = U_2 \tag{A.3}$$

where $F_1(X_1)$ is the cumulative probability distribution function (cdf) of X_1 and $F_2(X_1|X_2)$ is the cdf of X_2 given X_1. The joint probability density pertinent to the triangle is

$$f_{1,2}(x_1, x_2) = \begin{cases} \rho, & \text{if}\,\{(x_1, x_2) \mid x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \leq 1\}, \\ 0, & \text{else} \end{cases}, \tag{A.4}$$

where the extremal density [4] is

$$\rho = \frac{1}{2\pi\sqrt{x_1 x_2(1 - x_1 - x_2)}}. \tag{A.5}$$

The marginal pdf of X_1 is then

$$f_1(x_1) = \int_0^{1-x_1} f_{1,2}(x_1, x_2)dx_2 = \frac{1}{2\sqrt{x_1}}, \tag{A.6}$$

with a corresponding cdf

$$F_1(x_1) = \int_0^{x_1} f_1(x_1)dx_1 = \sqrt{x_1}. \tag{A.7}$$

The conditional pdf of X_2 is

$$f_2(x_2|x_1) = \frac{f_{1,2}(x_1, x_2)}{f_1(x_1)} = \frac{1}{\pi\sqrt{x_2(1 - x_1 - x_2)}}, \tag{A.8}$$

and a corresponding cdf

$$F_2(x_2|x_1) = \int_0^{x_2} f_2(x_2|x_1)dx_2 = \frac{1}{2} - \frac{1}{\pi}\sin^{-1}\left(\frac{2x_2 + x_1 - 1}{x_1 - 1}\right). \tag{A.9}$$

Our resulting system of equations is

$$\sqrt{X_1} = U_1, \tag{A.10}$$

$$\frac{1}{2} - \frac{1}{\pi}\sin^{-1}\left(\frac{2X_2 + X_1 - 1}{X_1 - 1}\right) = U_2, \tag{A.11}$$

which has solution

$$X_1 = U_1^2, \tag{A.12}$$

$$X_2 = \frac{1}{2}\big(1 - U_1^2\big)\big(1 - \cos(\pi U_2)\big), \tag{A.13}$$

$$= \big(1 - U_1^2\big)\sin^2\left(\frac{\pi}{2}U_2\right). \tag{A.14}$$

Appendix B

Gradients and Hessians

In this Appendix, I determine the required gradients and Hessians of both the Lagrangians ℓ_i (§2.4) and the Chen and Babuška [13] measure β (Chapter 5) with respect to changes in nodal positions $\{x_j, y_j\}$. I will use tensor notation [e.g. 2] throughout.

B.1 Lagrangian (ℓ_i) Derivatives

The Lagrangians are uniquely determined from the system of equations, c.f. (2.12),

$$V_{ij}\ell_j = b_i \tag{B.1}$$

where V_{ij} is the Vandermonde matrix, solely a function of the nodal positions, and $\{b_i\}$ are a set of basis functions, solely a function of (x, y),

$$V_{ij} = V_{ij}(x_j, y_j) = b_i(x_j, y_j) \tag{B.2a}$$

$$\ell_j = \ell_j(x, y, x_j, y_j) \tag{B.2b}$$

$$b_i = b_i(x, y) \tag{B.2c}$$

B.1.1 Terms like $\dfrac{\partial \ell_i}{\partial y}$

Differentiating (B.1) with respect to y, say, yields

$$V_{ij}\frac{\partial \ell_j}{\partial y} = \frac{\partial b_i}{\partial y}, \tag{B.3}$$

whose inversion furnishes the desired result. An analogous result for $\frac{\partial \ell_i}{\partial x}$ completes the calculation of the Lagrangian gradient.

B.1.2 Terms like $\frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial y}$

Differentiating (B.3) with respect to x, say, yields

$$V_{ij}\frac{\partial}{\partial x}\frac{\partial \ell_j}{\partial y} = \frac{\partial}{\partial x}\frac{\partial b_i}{\partial y}, \tag{B.4}$$

with analogous results for $\frac{\partial^2 \ell_i}{\partial x^2}$ and $\frac{\partial^2 \ell_i}{\partial y^2}$ completes the derivation of the Hessian.

B.1.3 Terms like $\frac{\partial \ell_i}{\partial y_p}$

Differentiating (B.1) with respect to y_p, say, yields

$$\frac{\partial V_{ij}}{\partial y_p}\ell_j + V_{ij}\frac{\partial \ell_j}{\partial y_p} = \cancelto{0}{\frac{\partial b_i}{\partial y_p}}. \tag{B.5}$$

But the derivative of (B.2a),

$$\frac{\partial V_{ij}}{\partial y_p} = \left[\{0\}, \{0\}, \ldots, \left\{\left.\frac{\partial b_i}{\partial y}\right|_p\right\}, \ldots, \{0\}\right], \tag{B.6}$$

is clearly non-zero only along the column $j = p$, simplifying (B.5),

$$\left.\frac{\partial b_i}{\partial y}\right|_p \ell_p + V_{ij}\frac{\partial \ell_j}{\partial y_p} = 0. \tag{B.7}$$

Cleverly, we select a *Lagrangian* basis, $b_i = \ell_i$. This choice reduces the Vandermonde matrix to the identity matrix, $V_{ij} = \delta_{ij}$, and gives

$$\left.\frac{\partial \ell_i}{\partial y}\right|_p \ell_p + \delta_{ij}\frac{\partial \ell_j}{\partial y_p} = 0, \tag{B.8}$$

or, simply,

$$\frac{\partial \ell_i}{\partial y_p} = -\left.\frac{\partial \ell_i}{\partial y}\right|_p \ell_p. \tag{B.9}$$

An analogous result for $\frac{\partial \ell_i}{\partial x_p}$ is found by inspection.

B.1.4 Terms like $\frac{\partial}{\partial x_p}\frac{\partial \ell_i}{\partial y}$

Differentiating (B.3) with respect to x_p, say, yields

$$\frac{\partial V_{ij}}{\partial x_p}\frac{\partial \ell_j}{\partial y} + V_{ij}\frac{\partial}{\partial x_p}\frac{\partial \ell_j}{\partial y} = \cancelto{0}{\frac{\partial}{\partial x_p}\frac{\partial b_i}{\partial y}}. \tag{B.10}$$

Applying the results from the derivation above, the Vandermonde determinant derivative simplifies such that

$$\left.\frac{\partial b_i}{\partial x}\right|_p \frac{\partial \ell_p}{\partial y} + V_{ij}\frac{\partial}{\partial x_p}\frac{\partial \ell_j}{\partial y} = 0, \tag{B.11}$$

and again selecting a Lagrangian basis,

$$\left.\frac{\partial \ell_i}{\partial x}\right|_p \frac{\partial \ell_p}{\partial y} + \delta_{ij}\frac{\partial}{\partial x_p}\frac{\partial \ell_j}{\partial y} = 0, \tag{B.12}$$

or simply,

$$\frac{\partial}{\partial x_p}\frac{\partial \ell_i}{\partial y} = -\left.\frac{\partial \ell_i}{\partial x}\right|_p \frac{\partial \ell_p}{\partial y}, \tag{B.13}$$

where the derivatives may be evaluated by (B.3).

Again, analagous terms for $\frac{\partial}{\partial y_p}\frac{\partial \ell_i}{\partial y}$, $\frac{\partial}{\partial x_p}\frac{\partial \ell_i}{\partial x}$, and $\frac{\partial}{\partial y_p}\frac{\partial \ell_i}{\partial x}$, are found by inspection.

B.1.5 Terms like $\frac{\partial}{\partial x_q}\frac{\partial \ell_i}{\partial y_p}$

Differentiating (B.5) results in four terms,

$$\frac{\partial}{\partial x_q}\frac{\partial V_{ij}}{\partial y_p}\ell_j + \frac{\partial V_{ij}}{\partial y_p}\frac{\partial \ell_j}{\partial x_q} + \frac{\partial V_{ij}}{\partial x_q}\frac{\partial \ell_j}{\partial y_p} + V_{ij}\frac{\partial}{\partial x_q}\frac{\partial \ell_j}{\partial y_p} = 0. \tag{B.14}$$

As before, derivatives of V_{ij} are only non-zero along a single column of corresponding to $j = p$ or $j = q$, reducing the first three terms to

$$\frac{\partial}{\partial x_q}\frac{\partial V_{ij}}{\partial y_p}\ell_j = \begin{cases} \left.\frac{\partial}{\partial x}\frac{\partial b_i}{\partial y}\right|_p \ell_p, & p = q, \\ 0, & p \neq q, \end{cases} \tag{B.15a}$$

$$\frac{\partial V_{ij}}{\partial y_p}\frac{\partial \ell_j}{\partial x_q} = \left.\frac{\partial b_i}{\partial y}\right|_p \frac{\partial \ell_p}{\partial x_q}, \tag{B.15b}$$

$$\frac{\partial V_{ij}}{\partial x_q}\frac{\partial \ell_j}{\partial y_p} = \left.\frac{\partial b_i}{\partial x}\right|_q \frac{\partial \ell_q}{\partial y_p}. \tag{B.15c}$$

(B.14) becomes

$$\underline{\frac{\partial}{\partial x}\frac{\partial b_i}{\partial y}\bigg|_p}\,\ell_p + \frac{\partial b_i}{\partial y}\bigg|_p \frac{\partial \ell_p}{\partial x_q} + \frac{\partial b_i}{\partial x}\bigg|_q \frac{\partial \ell_q}{\partial y_p} + V_{ij}\frac{\partial}{\partial x_q}\frac{\partial \ell_j}{\partial y_p} = 0, \tag{B.16}$$

where the underlined term is non-zero only for $p = q$. We again use a Lagrangian basis plus our previous results for $\frac{\partial \ell_p}{\partial x_q}$ and $\frac{\partial \ell_q}{\partial y_p}$,

$$\underline{\frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial y}\bigg|_p}\,\ell_p - \frac{\partial \ell_i}{\partial y}\bigg|_p \frac{\partial \ell_p}{\partial x}\bigg|_q \ell_q - \frac{\partial \ell_i}{\partial x}\bigg|_q \frac{\partial \ell_q}{\partial y}\bigg|_p \ell_p + \delta_{ij}\frac{\partial}{\partial x_q}\frac{\partial \ell_j}{\partial y_p} = 0, \tag{B.17}$$

or just,

$$\frac{\partial}{\partial x_q}\frac{\partial \ell_i}{\partial y_p} = \frac{\partial \ell_i}{\partial y}\bigg|_p \frac{\partial \ell_p}{\partial x}\bigg|_q \ell_q + \frac{\partial \ell_i}{\partial x}\bigg|_q \frac{\partial \ell_q}{\partial y}\bigg|_p \ell_p - \underline{\frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial y}\bigg|_p}\,\ell_p. \tag{B.18}$$

B.2 Chen–Babuška (β) Derivatives

The Chen and Babuška [13] measure β is simply the trace of the integration matrix W_{ij} (§2.5),

$$\beta = \int_\triangle \ell_i \ell_i \,\mathrm{d}A = W_{ii}. \tag{B.19}$$

In implementing Newton's method for optimizing points based on this measure we require its gradient and Hessian.

B.2.1 Terms like $\dfrac{\partial \beta}{\partial y_p}$

Differentiating (B.19) with respect to y_p, say, gives

$$\frac{\partial \beta}{\partial y_p} = \frac{\partial}{\partial y_p}\int_\triangle \ell_i \ell_i \,\mathrm{d}A = \int_\triangle 2\ell_i \frac{\partial \ell_i}{\partial y_p}\,\mathrm{d}A. \tag{B.20}$$

Substituting our relationship for $\frac{\partial \ell_i}{\partial y_p}$ (B.9), this becomes

$$\frac{1}{2}\frac{\partial \beta}{\partial y_p} = \int_\triangle \ell_i \left\{ -\ell_p \frac{\partial \ell_i}{\partial y}\bigg|_p \right\} \mathrm{d}A = -\frac{\partial \ell_i}{\partial y}\bigg|_p \int_\triangle \ell_i \ell_p \,\mathrm{d}A, \tag{B.21}$$

or simply,

$$\frac{1}{2}\frac{\partial \beta}{\partial y_p} = -\frac{\partial \ell_i}{\partial y}\bigg|_p W_{ip}. \tag{B.22}$$

B.2.2 Terms like $\dfrac{\partial}{\partial x_q}\dfrac{\partial \beta}{\partial y_p}$

Differentiating (B.20) with respect to x_q, say, gives

$$\frac{\partial}{\partial x_q}\frac{\partial \beta}{\partial y_p} = \int_{\triangle} \frac{\partial}{\partial x_q}\left\{2\ell_i \frac{\partial \ell_i}{\partial y_p}\right\} \mathrm{d}A = 2\int_{\triangle}\left\{\frac{\partial \ell_i}{\partial x_q}\frac{\partial \ell_i}{\partial y_p} + \ell_i \frac{\partial}{\partial x_q}\frac{\partial \ell_i}{\partial y_p}\right\}\mathrm{d}A. \tag{B.23}$$

Substituting previous results (B.9) and (B.18) leads to

$$\begin{aligned}\frac{1}{2}\frac{\partial}{\partial x_q}\frac{\partial \beta}{\partial y_p} = \int_{\triangle}\Bigg\{&\left(-\ell_q \left.\frac{\partial \ell_i}{\partial x}\right|_q\right)\left(-\ell_p \left.\frac{\partial \ell_i}{\partial y}\right|_p\right) \\ &+ \ell_i\left(\left.\frac{\partial \ell_i}{\partial y}\right|_p \left.\frac{\partial \ell_p}{\partial x}\right|_q \ell_q + \left.\frac{\partial \ell_i}{\partial x}\right|_q \left.\frac{\partial \ell_q}{\partial y}\right|_p \ell_p - \underline{\left.\frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial y}\right|_p \ell_p}\right)\Bigg\}\mathrm{d}A.\end{aligned} \tag{B.24}$$

Rearranging,

$$\begin{aligned}\frac{1}{2}\frac{\partial}{\partial x_q}\frac{\partial \beta}{\partial y_p} = &\left.\frac{\partial \ell_i}{\partial x}\right|_q \left.\frac{\partial \ell_i}{\partial y}\right|_p \int_{\triangle} \ell_q \ell_p \,\mathrm{d}A + \left.\frac{\partial \ell_i}{\partial y}\right|_p \left.\frac{\partial \ell_p}{\partial x}\right|_q \int_{\triangle} \ell_i \ell_q \,\mathrm{d}A \\ &+ \left.\frac{\partial \ell_i}{\partial x}\right|_q \left.\frac{\partial \ell_q}{\partial y}\right|_p \int_{\triangle} \ell_i \ell_p \,\mathrm{d}A - \underline{\left.\frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial y}\right|_p} \int_{\triangle} \ell_i \ell_p \,\mathrm{d}A,\end{aligned} \tag{B.25}$$

or simply,

$$\begin{aligned}\frac{1}{2}\frac{\partial}{\partial x_q}\frac{\partial \beta}{\partial y_p} = &\left.\frac{\partial \ell_i}{\partial x}\right|_q \left.\frac{\partial \ell_i}{\partial y}\right|_p W_{qp} + \left.\frac{\partial \ell_i}{\partial y}\right|_p \left.\frac{\partial \ell_p}{\partial x}\right|_q W_{iq} \\ &+ \left.\frac{\partial \ell_i}{\partial x}\right|_q \left.\frac{\partial \ell_q}{\partial y}\right|_p W_{ip} - \underline{\left.\frac{\partial}{\partial x}\frac{\partial \ell_i}{\partial y}\right|_p} W_{ip}.\end{aligned} \tag{B.26}$$

Appendix C

Vandermonde Determinant Derivatives

By exploting the multilinear nature of determinants, we may rewrite derivatives of the Vandermonde determinant as derivatives of the Lagrangians. If the individual elements of a matrix are a function of some variable, say s, then the derivative of said matrix is the sum of the n determinants formed by replacing one column by the derivatives of its elements [2]:

$$\begin{aligned}\frac{\partial}{\partial x_k}\det V = &\det\left[\cancelto{0}{\frac{\partial}{\partial x_k}\boldsymbol{b}(x_1,y_1)},\boldsymbol{b}(x_2,y_2),\ldots,\boldsymbol{b}(x_k,y_k),\ldots,\boldsymbol{b}(x_n,y_n)\right]\\ &+\det\left[\boldsymbol{b}(x_1,y_1),\cancelto{0}{\frac{\partial}{\partial x_k}\boldsymbol{b}(x_2,y_2)},\ldots,\boldsymbol{b}(x_k,y_k),\ldots,\boldsymbol{b}(x_n,y_n)\right]\\ &+\ldots\\ &+\det\left[\boldsymbol{b}(x_1,y_1),\boldsymbol{b}(x_2,y_2),\ldots,\frac{\partial}{\partial x_k}\boldsymbol{b}(x_k,y_k),\ldots,\boldsymbol{b}(x_n,y_n)\right]\\ &+\ldots\\ &+\det\left[\boldsymbol{b}(x_1,y_1),\boldsymbol{b}(x_2,y_2),\ldots,\boldsymbol{b}(x_k,y_k),\ldots,\cancelto{0}{\frac{\partial}{\partial x_k}\boldsymbol{b}(x_n,y_n)}\right].\end{aligned} \tag{C.1}$$

Since all determinants but one will be zero, we have simply

$$\frac{\partial}{\partial x_k}\det V = \det\left[\boldsymbol{b}(x_1,y_1),\boldsymbol{b}(x_2,y_2),\ldots,\frac{\partial}{\partial x_k}\boldsymbol{b}(x_k,y_k),\ldots,\boldsymbol{b}(x_n,y_n)\right]. \tag{C.2}$$

If we differentiate (4.2),

$$\frac{\partial \ell_i}{\partial x} = \frac{\det\left[\boldsymbol{b}(x_1,y_1),\boldsymbol{b}(x_2,y_2),\ldots,\frac{\partial}{\partial x}\boldsymbol{b}(x,y),\ldots,\boldsymbol{b}(x_n,y_n)\right]}{\det\left[\boldsymbol{b}(x_1,y_1),\boldsymbol{b}(x_2,y_2),\ldots,\boldsymbol{b}(x_i,y_i),\ldots,\boldsymbol{b}(x_n,y_n)\right]}, \tag{C.3}$$

and evaluate the result at $(x, y) = (x_k, y_k)$,

$$\left.\frac{\partial \ell_i}{\partial x}\right|_k = \frac{\det\left[\boldsymbol{b}(x_1, y_1), \boldsymbol{b}(x_2, y_2), \ldots, \frac{\partial}{\partial x_k}\boldsymbol{b}(x_k, y_k), \ldots, \boldsymbol{b}(x_n, y_n)\right]}{\det\left[\boldsymbol{b}(x_1, y_1), \boldsymbol{b}(x_2, y_2), \ldots, \boldsymbol{b}(x_i, y_i), \ldots, \boldsymbol{b}(x_n, y_n)\right]}, \tag{C.4}$$

we are left with (C.2) divided by $\det V$. For the case $i = k$, we may combine the results of (C.2) and (C.4) such that,

$$\left.\frac{\partial \ell_k}{\partial x}\right|_k = \frac{\frac{\partial}{\partial x_k}\det V}{\det V} = \frac{\partial}{\partial x_k}\ln\det V. \tag{C.5}$$