What is a Good Linear Element? Interpolation, Conditioning, and Quality Measures

Jonathan Richard Shewchuk

University of California at Berkeley, Berkeley, CA, U.S.A. jrs@cs.berkeley.edu

Abstract

When a mesh of simplicial elements (triangles or tetrahedra) is used to form a piecewise linear approximation of a function, the accuracy of the approximation depends on the sizes and shapes of the elements. In finite element methods, the conditioning of the stiffness matrices also depends on the sizes and shapes of the elements. This paper explains the mathematical connections between mesh geometry, interpolation errors, and stiffness matrix conditioning. These relationships are expressed by error bounds and element quality measures that determine the fitness of a triangle or tetrahedron for interpolation or for achieving low condition numbers. Unfortunately, the quality measures for these two purposes do not agree with each other; for instance, small angles are bad for matrix conditioning but not for interpolation. Several of the upper and lower bounds on interpolation errors and element stiffness matrix conditioning given here are tighter than those that have appeared in the literature before, so the quality measures are likely to be unusually precise indicators of element fitness.

Keywords: finite element mesh, multivariate interpolation error, condition number, quality measure

1 Introduction

Interpolation, contouring, and finite element methods rely on the availability of meshes whose elements have the right shapes and sizes. The accuracy or speed of some applications can be compromised by just a few bad elements. Algorithms for mesh generation and mesh improvement are expected to produce elements whose "quality" is as good as possible. However, forty-odd years after the invention of the finite element method, our understanding of the relationship between mesh geometry, numerical accuracy, and stiffness matrix conditioning remains incomplete, even in the simplest cases. Engineering experience and asymptotic mathematical results have taught us that equilateral elements are usually good, and skinny or skewed elements are usually bad. However, there has been insufficient mathematical guidance for, say, choosing the better of two elements of intermediate quality.

This paper examines triangular and tetrahedral meshes used for piecewise linear interpolation (including finite element methods with piecewise linear basis functions). The quality of a mesh depends on the application that uses it. Interpolation accuracy is important for most tasks. For finite element methods, discretization errors and the condition number of the global stiffness matrix are important too. Error bounds and quality measures are provided here that estimate the influence of element geometry on accuracy and conditioning, and can guide numerical analysts and mesh generation algorithms in creating and evaluating meshes.

Interpolation on a triangular or tetrahedral mesh constructs a function that attempts to approximate some "true" function, whose exact identity might or might not be known. For example, a surveyor may know the altitude of the land at each point in a large sample, and use interpolation over a triangulation to approximate the altitude at points where readings were not taken. If a triangulation's sole purpose is as a basis for interpolation, the primary criterion of its fitness is how much the interpolated function differs from the true function. There are two types of *interpolation error* that matter for most applications: the difference between the interpolated function and the true function, and the difference between the gradient of the interpolated function and the gradient of the true function. Errors in the gradient can be surprisingly

Supported in part by the National Science Foundation under Awards ACI-9875170, CMS-9980063, and EIA-9802069, and in part by a gift from the Okawa Foundation. The views and conclusions in this document are those of the author. They are not endorsed by, and do not necessarily reflect the position or policies of, the Okawa Foundation or the U. S. Government.

important, whether the application is rendering, mapmaking, or simulation, because they can compromise accuracy or create unwanted visual artifacts. In finite element methods they contribute to discretization errors.

If the true function is smooth, the error in the interpolated function can be reduced simply by making the triangles or tetrahedra smaller. However, the error in the gradient is strongly affected by the shape of the elements as well as their size, and this error is often the primary arbiter of element quality. The enemy is large angles: the error in the gradient can grow arbitrarily large as angles approach $180°$. Bounds on the errors associated with piecewise linear interpolation are discussed in Section 2.

If your application is the finite element method, then the condition number of the stiffness matrix associated with the method should be kept as small as possible. Poorly conditioned matrices affect linear equation solvers by slowing them down or introducing large roundoff errors into their results. Element shape has a strong influence on matrix conditioning, but unlike with interpolation errors, small angles can have as bad an effect as large ones. The relationship between element shape and matrix conditioning depends on the partial differential equation being solved and the basis functions and test functions used to discretize it. Bounds on condition number must be derived on a case-by-case basis. The stiffness matrices associated with Poisson's equation on linear elements are studied in Section 3.

The *discretization error* is the difference between the approximation computed by the finite element method and the true solution. Like stiffness matrix condition numbers, discretization error depends in part on the partial differential equation and the method of discretization. However, discretization error is closely related to the interpolation errors, and is mitigated by elements whose shapes and sizes are selected to control the interpolation errors.

Quality measures for evaluating and comparing elements are discussed in Section 4. These include measures of an element's fitness for interpolation and stiffness matrix formation. The quality measures discussed in this paper can be used in either an *a priori* or *a posteriori* fashion, and are designed to interact well with numerical optimization methods for mesh smoothing.

The results of this paper can be extended to *anisotropic* meshes, whose elements are elongated in response to properties of an interpolated function or a differential equation. Specifically, long, thin, correctly oriented elements can achieve the best tradeoff between interpolation error and the number of elements when the function being interpolated has a large curvature along one axis and very little curvature along an orthogonal axis. Elongated, correctly oriented elements can achieve the best matrix conditioning for partial differential equations whose coefficients create anisotropy. These extensions are omitted because of space, but they appear in the full-length version of this paper. All the derivations and proofs of the results may be found there too.

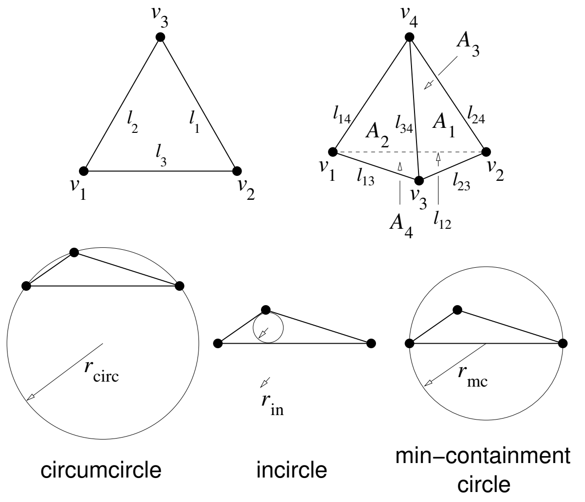

Figure 1: Quantities associated with triangles and tetrahedra.

Table 1 is a reference chart for the notation used in this paper. Some of the quantities are illustrated in Figure 1. Triangle vertices and edges are numbered from 1 to 3, with vertex v_i opposite edge i. Tetrahedron vertices and faces are numbered from 1 to 4, with vertex v_i opposite face i.

The value r_{circ} is the radius of the circumcircle or circumsphere of a triangular or tetrahedral element t, r_{in} is the radius of the incircle or insphere of t, and r_{mc} is the radius of the min-containment circle or sphere of t. The *circumcircle*, or circumscribed circle, of a triangle is the unique circle that passes through all three of its vertices, and the *circumsphere* of a tetrahedron passes through all four of its vertices. The *incircle*, or inscribed circle, of a triangle is the smallest circle that touches all three of its sides, and the *insphere* of a tetrahedron is the smallest sphere that touches all four of its triangular faces. The *min-containment circle* of a triangle is the smallest circle that encloses the triangle; its center is either the circumcenter of the triangle or a midpoint of one of its edges. The *min-containment sphere* of a tetrahedron is the smallest sphere that encloses it; its center is either the circumcenter of the tetrahedron, the circumcenter of one of its triangular faces, or a midpoint of one of its edges.

Some of the quantities are *signed*, which means that they are negative for inverted elements. To say an element is *inverted* is to presuppose that it has a fixed topological orientation, defined by an ordering of its vertices. For instance, a triangle is inverted if its vertices are supposed to occur in counterclockwise order, but upon inspection occur in clockwise order. The topology of a mesh determines the orientation of each element relative to the orientations of all the others.

2 Element Size, Element Shape, and Interpolation Error

This section describes the mathematical relationship between the size and shape of an element and the errors in a

Table 1: Notation used in this paper. Signed quantities are negative for inverted elements. Edge lengths are always nonnegative.

A	The signed area of a triangle.
V	The signed volume of a tetrahedron.
A_1, A_2, A_3, A_4	The unsigned areas of the faces of a tetrahedron.
$A_{\rm rms}$	The root-mean-square face area of a tetrahedron, $\sqrt{\frac{1}{4}\sum_{i=1}^{4} A_i^2}$.
ℓ_1, ℓ_2, ℓ_3	The edge lengths of a triangle.
$\ell_{\rm rms}$	The root-mean-square edge length of a triangle, $\sqrt{\frac{1}{3}\sum_{i=1}^{3} \ell_i^2}$.
$\ell_{\rm min}$, $\ell_{\rm med}$, $\ell_{\rm max}$	The minimum, median, and maximum edge lengths of an element. ($\ell_{\rm med}$ is defined for triangles only.)
ℓ_{ij}	The length of the edge connecting vertices v_i and v_j.
$a_{\rm med}$, $a_{\rm max}$	The median- and maximum-magnitude signed altitudes of a triangle: $a_{\rm med} = 2A/\ell_{\rm med}$ and $a_{\rm max} = 2A/\ell_{\rm min}$.
$r_{\rm circ}$	The signed *circumradius* of an element (the radius of its circumscribing circle or sphere). To avoid possible division by zero, calculate $1/r_{\rm circ}$ instead of $r_{\rm circ}$.
$r_{\rm in}$	The signed *inradius* of an element (the radius of its inscribed circle or sphere). For a triangle, $r_{\rm in} = 2A/(\ell_1 + \ell_2 + \ell_3)$. For a tetrahedron, $r_{\rm in} = 3V/\sum_{i=1}^{4} A_i$.
$r_{\rm mc}$	The unsigned radius of the *min-containment circle or sphere* of an element (the smallest circle or sphere that encloses the element).
θ_i	The angle at vertex v_i of a triangle.
$\theta_{\rm min}, \theta_{\rm max}$	The signed minimum and maximum angles of a triangle.
θ_{ij}	In a tetrahedron, the dihedral angle at the edge connecting vertices v_i and v_j.

piecewise linear approximation of a function.

The celebrated paper of Babuška and Aziz [2] demonstrates that the accuracy of finite element solutions on triangular meshes degrades seriously as angles are allowed to approach 180°, but the same is not true as angles are allowed to approach 0°, so long as the largest angles are not too large. In other words, small angles are not deleterious to the interpolation accuracy or the discretization error (although they may be deleterious to the stiffness matrix). Previously, researchers had believed that small angles must be prohibited.

The Babuška–Aziz paper is more often cited than understood, as it is cast in the language of functional analysis. Its results are asymptotic and offer little guidance in, say, how to compare two differently-shaped elements of intermediate quality. This section presents error bounds (and Section 4 presents related quality measures) that can be used to accurately judge the shape and size of an element. These bounds are stronger than the classical bounds of approximation theory—asymptotically stronger in some cases. The bounds for triangles are tight to within a small constant factor. Unfortunately, all proofs are omitted because of constraints on space.

Let T be a triangular or tetrahedral mesh, and let $f(p)$ be a continuous scalar function defined over the mesh. Let $g(p)$ be a piecewise linear approximation to $f(p)$, where $g(v) = f(v)$ at each vertex v of T, and $g(p)$ is linear over any single element of T. Table 2 gives bounds on two types of interpolation error associated with g. The norm $\|f - g\|_\infty$ is defined to be the maximum pointwise interpolation error over the element t, $\max_{p \in t} |f(p) - g(p)|$. The norm $\|\nabla f - \nabla g\|_\infty$ is the maximum magnitude of the pointwise error in the interpolated gradient, $\max_{p \in t} |\nabla f(p) - \nabla g(p)|$.

If $f(p)$ is arbitrary, $g(p)$ can be an arbitrarily bad approximation of $f(p)$. The error can be bounded only if $f(p)$ is constrained in some way. A reasonable constraint, which yields the error bounds in Table 2, is to assume that $f(p)$ is smooth and the absolute curvature of $f(p)$ is bounded in each triangle t by some constant $2c_t$ (where c_t may differ for each t). The *curvature* $f''_{\mathbf{d}}(p)$ of the function f at the point p along an arbitrary direction vector $\mathbf{d}$ is its directional second derivative along $\mathbf{d}$. Specifically, let the point p have

Table 2: Bounds on interpolation error for a single element t. The function g is a linear approximation of f over t. All bounds assume that the magnitude of the directional second derivative of f does not exceed $2c_t$ anywhere in the element t. The "weaker but simpler upper bounds" are not asymptotically weaker; they are weaker than the stronger upper bounds by a factor of no more than three. Each lower bound implies that there exists some function f for which the error is at least as large as the lower bound.

	$\|f-g\|_\infty$	$\|\nabla f - \nabla g\|_\infty$
Upper bound, triangles	$c_t r_{\mathrm{mc}}^2$	$c_t \frac{\ell_{\max}\ell_{\mathrm{med}}\left(\ell_{\min}+4r_{\mathrm{in}}\right)}{2A}$
Weaker but simpler upper bound, triangles	$c_t \frac{\ell_{\max}^2}{3}$	$c_t \frac{3\ell_{\max}\ell_{\mathrm{med}}\ell_{\min}}{2A}$
Lower bound, triangles	$c_t r_{\mathrm{mc}}^2$	$2c_t \max\left\{r_{\mathrm{circ}}, a_{\max}, \sqrt{\ell_{\max}^2 - a_{\mathrm{med}}^2}\right\}$
Note: for triangles, $c_t \frac{\ell_{\max}\ell_{\mathrm{med}}\ell_{\min}}{2A} = c_t \frac{\ell_{\max}}{\sin\theta_{\max}} = 2c_t r_{\mathrm{circ}}$		
Upper bound, tetrahedra	$c_t r_{\mathrm{mc}}^2$	$c_t \frac{\frac{1}{3V}\sum_{1\le i<j\le 4} A_i A_j \ell_{ij}^2 + 2\max_i \sum_{j\ne i} A_j \ell_{ij}}{\sum_{m=1}^4 A_m}$
Weaker but simpler upper bound, tetrahedra	$c_t \frac{3\ell_{\max}^2}{8}$	$c_t \frac{\sum_{1\le i<j\le 4} A_i A_j \ell_{ij}^2}{V\sum_{m=1}^4 A_m}$
Lower bound, tetrahedra	$c_t r_{\mathrm{mc}}^2$	$2c_t r_{\mathrm{circ}}$
Note: for tetrahedra, $c_t \frac{\sum_{1\le i<j\le 4} A_i A_j \ell_{ij}^2}{3V\sum_{m=1}^4 A_m} = c_t \frac{\sum_{1\le i<j\le 4} \ell_{ij}^2 \ell_{kl}/\sin\theta_{kl}}{2\sum_{m=1}^4 A_m}$, where i, j, k, and l are distinct in each term of the summation		

coordinates (x, y, z), and consider the Hessian matrix

$$H(p) = \left[\begin{array}{ccc} \frac{\partial^2}{\partial x^2} f(p) & \frac{\partial^2}{\partial x \partial y} f(p) & \frac{\partial^2}{\partial x \partial z} f(p) \\ \frac{\partial^2}{\partial x \partial y} f(p) & \frac{\partial^2}{\partial y^2} f(p) & \frac{\partial^2}{\partial y \partial z} f(p) \\ \frac{\partial^2}{\partial x \partial z} f(p) & \frac{\partial^2}{\partial y \partial z} f(p) & \frac{\partial^2}{\partial z^2} f(p) \end{array}\right].$$

(For the two-dimensional case, delete the last row and column of $H(p)$.) To support matrix notation, each point or vector p is treated as a $d \times 1$ vector whose transpose p^T is a $1 \times d$ vector. The notation $\mathbf{d}^T H(p)\mathbf{d}$ denotes the scalar result of the matrix multiplication

$$\mathbf{d}^T H(p)\mathbf{d} = \left[\begin{array}{ccc} d_x & d_y & d_z \end{array}\right] H(p) \left[\begin{array}{c} d_x \\ d_y \\ d_z \end{array}\right].$$

For any unit direction vector $\mathbf{d}$, the directional curvature is $f''_{\mathbf{d}}(p) = \mathbf{d}^T H(p)\mathbf{d}$. If $\mathbf{d}$ is not a unit vector, it is easy to show that $f''_{\mathbf{d}}(p) = \mathbf{d}^T H(p)\mathbf{d}/|\mathbf{d}|^2$. Assume that f is known to satisfy the following curvature constraint:[1] for any direction $\mathbf{d}$,

$$|f''_{\mathbf{d}}(p)| \le 2c_t.$$

[1] For those familiar with matrix norms, note that $\|H\|_2 = \max_{|\mathbf{d}|=1} |\mathbf{d}^T H(p)\mathbf{d}|$, so the constraint can be written $\|H\|_2 \le 2c_t$. An equivalent statement is that the eigenvalues of H are all in $[-2c_t, 2c_t]$.

How does one obtain bounds on curvature to use in generating, evaluating, or improving a mesh? The per-element curvature bounds $2c_t$ sometimes come from *a priori* error estimators, based on knowledge of the function to be interpolated. Sometimes they are provided by *a posteriori* error estimators, which are estimated from a finite element solution over another mesh of the same domain. If bounds on curvature are not available, it might not be possible to bound the interpolation error, but the formulae in Table 2 may still be used to compare elements, by dropping c_t from each formula. This is equivalent to assuming that there is some unknown bound on curvature that holds everywhere.

Let's examine the bounds. The upper bound on $\|f-g\|_\infty$, the maximum interpolation error over t, is $c_t r_{\mathrm{mc}}^2$. This bound is tight: for any triangle or tetrahedron t with min-containment radius r_{mc}, there is a function f such that $\|f-g\|_\infty = c_t r_{\mathrm{mc}}^2$. This bound (and its tightness) was first derived by Waldron [12], and it applies to higher-dimensional simplicial elements as well.

Interestingly, Rajan [10] shows that for any set of vertices in any dimensionality, the Delaunay triangulation of those vertices minimizes the maximum min-containment radius (as compared with all other triangulations of the vertices).

It is interesting to compare this bound to the bounds usually given for interpolation, which implicate the maximum edge

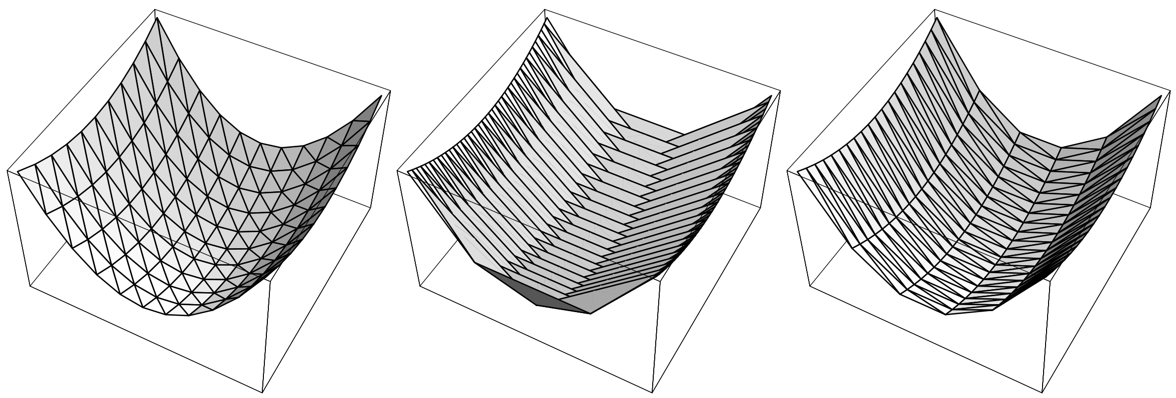

Figure 2: A visual illustration of how large angles, but not small angles, can cause the error $\nabla f - \nabla g$ to explode. In each triangulation, 200 triangles are used to render a paraboloid.

length of each element. To obtain a specified level of accuracy, a mesh is refined until no edge is larger than a specified length. However, the min-containment radius of an element gives a tighter bound on $\|f - g\|_\infty$ than the maximum edge length.

Unfortunately, the min-containment radius r_{mc} is expensive to compute. The maximum edge length $\ell_{\max}$ is a much faster alternative. For a triangle, $r_{\mathrm{mc}} \leq \ell_{\max}/\sqrt{3}$, and for a tetrahedron, $r_{\mathrm{mc}} \leq \sqrt{3/8}\ell_{\max}$. Substitution yields the faster-to-compute but slightly looser bounds $\|f - g\|_\infty \leq c_t \ell_{\max}^2/3$ (for triangles), $\|f - g\|_\infty \leq 3c_t \ell_{\max}^2/8$ (for tetrahedra).

The error $f - g$ is not the only concern. In many applications, g is expected to accurately represent the gradients of f, and the error $\nabla f - \nabla g$ is just as important or more important than $f - g$. Consider using the finite element method to find a piecewise linear approximation h to the true solution f of a self-adjoint second-order partial differential equation. Although g and h are both piecewise linear functions, they differ because h does not usually equal f at the mesh vertices. Nevertheless, the discretization error $f - h$ normally can be bounded only if both $f - g$ and $\nabla f - \nabla g$ can be bounded. Simulations of mechanical deformation provide a second example, where the accuracy of ∇g is particularly important because ∇f (the strains) is of more interest than f (the displacements). Visualization of height fields provides a third example, as we shall see shortly.

Newly derived bounds on $\|\nabla f - \nabla g\|_\infty$ appear in Table 2. The bounds reveal that $\|\nabla f - \nabla g\|_\infty$ can grow arbitrarily large as elements become arbitrarily badly shaped, unlike $\|f - g\|_\infty$. Observe that the area or volume appears in the denominator of these bounds. Imagine distorting a triangle or tetrahedron so that its area or volume approaches zero. Then ∇g may or may not approach infinity, depending on whether the numerator of the error bound also approaches zero.

First imagine an isosceles triangle with one angle near 180° and two tiny angles. As the large angle approaches 180°, A approaches zero and the edge lengths do not change much, so the error bounds grow arbitrarily large. Now imagine an

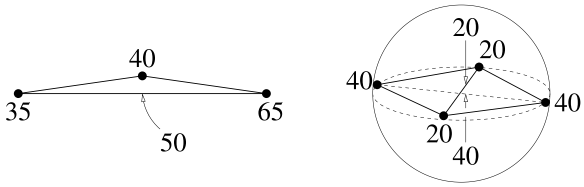

Figure 3: As the large angle of the triangle approaches 180°, or the sliver tetrahedron becomes arbitrarily flat, the magnitude of the vertical component of ∇g becomes arbitrarily large.

isosceles triangle with one tiny angle and two angles near 90°. As the tiny angle approaches zero, A approaches zero, but $\ell_{\min}$ and r_{in} approach zero at the same rate, so the error bounds change little. Hence, angles near 180° are harmful, whereas angles near zero are, by themselves, benign. The same can be said of the dihedral angles of tetrahedra.

Figure 2 visually illustrates these effects. Three triangulations, each having 200 triangles, are used to render a paraboloid. The mesh of long thin triangles at right has no angle greater than 90°, and visually performs only slightly worse than the isotropic triangulation at left. (The slightly worse performance is because of the longer edge lengths.) However, the middle paraboloid looks like a washboard, because the triangles with large angles have very inaccurate gradients.

Figure 3 shows why this problem occurs. The triangle illustrated has values associated with its vertices that represent heights (or, say, an approximation of some physical quantity). The values of g at the endpoints of the bottom edge are 35 and 65, so the linearly interpolated value of g at the center of the edge is 50. This value is independent of the value associated with the top vertex. As the angle at the upper vertex approaches 180°, the interpolated point (with value 50) becomes arbitrarily close to the upper vertex (with value 40). Hence, ∇g may become arbitrarily large (in its vertical component), and is clearly specious as an approximation of ∇f, even though $g = f$ at the vertices.

The same effect is seen between two edges of a sliver tetrahedron that pass near each other, also illustrated in Figure 3. A *sliver* is a tetrahedron that is nearly flat even though none of its edges is much shorter than the others. A typical sliver is formed by uniformly spacing its four vertices around the equator of a sphere, then perturbing one of the vertices just off the equator so that the sliver has some (but not much) volume.

Because of this sensitivity, mesh generators usually choose the shapes of elements to control $\|\nabla f - \nabla g\|_\infty$, and not $\|f - g\|_\infty$, which can be reduced simply by using smaller elements. Section 4 presents quality measures that judge the shape of elements based on their fitness for interpolation.

Table 2 gives two upper bounds on $\|\nabla f - \nabla g\|_\infty$ over a triangle. The "weaker but simpler upper bound" is not as good an indicator as the stronger upper bound, but it has the advantages of being smooth almost everywhere (and therefore more amenable to numerical optimization) and faster to compute. Both upper bounds are almost tight, to within a factor of three: for any triangle t, there is a function f such that $\|\nabla f - \nabla g\|_\infty = 2c_t r_{\mathrm{circ}}$, and the weaker upper bound is $6c_t r_{\mathrm{circ}}$.

These bounds are interesting because the two-dimensional Delaunay triangulation minimizes the maximum circumradius, just as it minimizes the maximum min-containment radius. (Unfortunately, this property does not hold in three or more dimensions, unlike Rajan's min-containment result.)

The upper bound of $6c_t r_{\mathrm{circ}}$ and the lower bound of $2c_t r_{\mathrm{circ}}$ can be expressed in three different forms (see the first note in Table 2), one of which implicates the largest angle of the triangle. The upper bound $3c_t \ell_{\max}/\sin\theta_{\max}$ can be loosely decomposed into a size contribution $3c_t\ell_{\max}$ and a shape contribution $1/\sin\theta_{\max}$. This seems to suggest that a triangular mesh generator should seek to minimize the maximum angle, but Section 4 discusses slightly better shape measures. For triangles with no large angle, the error decreases proportionately to the length of the longest edge. (Incidentally, Bramble and Zlámal [5] derived a well-known upper bound proportional to $c_t\ell_{\max}/\sin\theta_{\min}$. Replacing $\theta_{\min}$ with $\theta_{\max}$, as done here, obviously leads to different conclusions.)

The bounds for tetrahedra are more difficult to interpret than the bounds for triangles. The alternative form for the error bound (bottom of Table 2) suggests that the error may approach infinity as the sine of a dihedral angle θ_{kl} approaches zero. However, the error does not approach infinity if the length ℓ_{ij} of the opposite edge approaches zero at the same rate as the angle. If an angle θ_{kl} is small but the opposite edge length ℓ_{ij} is not, the tetrahedron must have a large dihedral angle as well. Small dihedral or planar angles are not problematic for interpolation unless a large angle is present too.

Figure 4 provides some insight into which tetrahedron shapes are good or bad for accurate interpolation. The "good" tetrahedra are of two types: those that are not flat, and those that can grow arbitrarily flat without having a large planar or dihedral angle. The "bad" tetrahedra have error bounds that explode, and a dihedral angle or a planar angle that approaches $180°$, as they are flattened. (Unfortunately, most of the "good" tetrahedra in the figure are bad for stiffness matrix conditioning because of the small angles.)

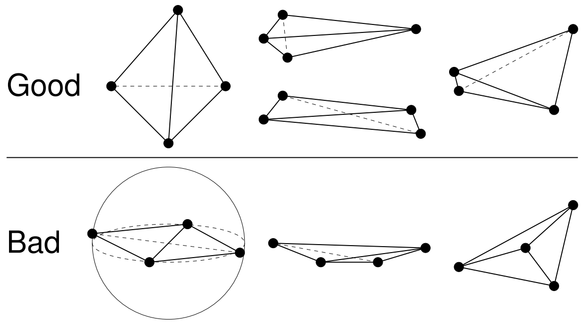

Figure 4: The top four tetrahedron shapes incur little interpolation error. The bottom three tetrahedron shapes can cause $\|\nabla f - \nabla g\|_\infty$ to be unnecessarily large, and to grow without bound if the tetrahedra are flattened.

The upper bounds for tetrahedra are not known to be asymptotically tight, but I conjecture that they are. Unfortunately, it is difficult to develop a strong lower bound that covers all tetrahedron shapes. However, the no-large-angle condition is necessary. For any tetrahedron with a dihedral angle or planar angle approaching $180°$, there is a function f for which $\|\nabla f - \nabla g\|_\infty$ approaches infinity.

3 Element Size, Element Shape, and Stiffness Matrix Conditioning

This section describes the mathematical relationship between the shapes of elements and the condition numbers of the stiffness matrices used in the finite element method. It assumes that the reader is familiar with the finite element method; no introduction is given here, but many good texts are available, including Johnson [8], Strang [11], and Becker, Carey, and Oden [3]. The main points are that both small and large angles can cause poor conditioning, and that the relationship can be quantified in a way useful for comparing differently shaped elements.

The finite element method is different for every partial differential equation, and unfortunately, so is the relationship between element shape and matrix conditioning. As a concrete example, I will study the Poisson equation,

$$-\nabla^2 f(p) = \eta(p),$$

where $\eta(p)$ is a known function of p, and the goal is to find an approximation $h(p)$ of the unknown function $f(p)$ for some specified boundary conditions.

In the Galerkin formulation of the finite element method, the piecewise approximation h is composed from local piecewise basis functions, which are in turn composed from *shape functions*. Each shape function is defined on just one element. If h is piecewise linear, then the shape functions are the barycentric coordinates of p.

For each element t, the finite element method constructs a $(d+1) \times (d+1)$ *element stiffness matrix* K_t, where d is the dimension. The element stiffness matrices are *assembled* into an $n \times n$ *global stiffness matrix* K, where n is (for Poisson's equation on linear elements) the number of mesh vertices. The assembly process sums the entries of each element stiffness matrix into the entries of the global stiffness matrix.

The difficulty of solving the linear system of equations associated with the finite element method typically grows with the condition number $\kappa = \lambda_{\max}^K / \lambda_{\min}^K$ of K, where $\lambda_{\max}^K$ and $\lambda_{\min}^K$ are the largest and smallest eigenvalues of K. A large condition number means that iterative solvers will run slowly, and direct methods may incur excessive roundoff error.

$\lambda_{\min}^K$ is closely tied to properties of the physical system being modeled, and to the sizes of the elements. Fried [7] offers a lower bound for $\lambda_{\min}^K$ that is proportional to the area or volume of the smallest element, and an upper bound proportional to the largest element. Fortunately, $\lambda_{\min}^K$ is not strongly influenced by element shape.

In contrast, $\lambda_{\max}^K$ can be made arbitrarily large by a single badly-shaped element. $\lambda_{\max}^K$ is related to the largest eigenvalues of the element stiffness matrices as follows. For each element t, let $\lambda_{\max}^t$ be the largest eigenvalue of its element stiffness matrix. Let m be the maximum number of elements meeting at a single vertex. Fried shows that

$$\max_t \lambda_{\max}^t \leq \lambda_{\max}^K \leq m \max_t \lambda_{\max}^t,$$

so κ is roughly proportional to the largest eigenvalue among the element stiffness matrices.

Let's examine some element stiffness matrices and their eigenvalues. The element stiffness matrix for a triangle is

$$K_t = \frac{1}{8A} \begin{bmatrix} 2\ell_1^2 & \ell_3^2 - \ell_1^2 - \ell_2^2 & \ell_2^2 - \ell_1^2 - \ell_3^2 \\ \ell_3^2 - \ell_1^2 - \ell_2^2 & 2\ell_2^2 & \ell_1^2 - \ell_2^2 - \ell_3^2 \\ \ell_2^2 - \ell_1^2 - \ell_3^2 & \ell_1^2 - \ell_2^2 - \ell_3^2 & 2\ell_3^2 \end{bmatrix}$$

$$= \frac{1}{2} \begin{bmatrix} \cot\theta_2 + \cot\theta_3 & -\cot\theta_3 & -\cot\theta_2 \\ -\cot\theta_3 & \cot\theta_1 + \cot\theta_3 & -\cot\theta_1 \\ -\cot\theta_2 & -\cot\theta_1 & \cot\theta_1 + \cot\theta_2 \end{bmatrix}.$$

If one of the angles approaches $0°$ or $180°$, its cotangent approaches infinity, and so does $\lambda_{\max}^t$. Therefore, both small and large angles can ruin matrix conditioning.

The eigenvalues of K_t are the roots of its characteristic polynomial $p(\lambda)$. For triangles,

$$p(\lambda) = \lambda^3 - \frac{\ell_1^2 + \ell_2^2 + \ell_3^2}{4A}\lambda^2 + \frac{3}{4}\lambda.$$

The roots of this polynomial are $\lambda = 0$ and

$$\lambda = \frac{\ell_1^2 + \ell_2^2 + \ell_3^2 \pm \sqrt{(\ell_1^2 + \ell_2^2 + \ell_3^2)^2 - 48A^2}}{8A}.$$

The largest root $\lambda_{\max}^t$ is a scale-invariant indicator of the quality of the triangle's shape. (*Scale-invariant* means that if t is scaled uniformly without any change to its shape, $\lambda_{\max}^t$ does not change.) This eigenvalue is used as a quality measure in Section 4. If a simpler or smoother indicator is desired, the radical can be dropped, but the simplified bound is only tight to within a factor of two.

$$\frac{\ell_1^2 + \ell_2^2 + \ell_3^2}{8A} \leq \lambda_{\max}^t \leq \frac{\ell_1^2 + \ell_2^2 + \ell_3^2}{4A}.$$

Suppose the mesh has no badly shaped triangles—for every triangle t, $\lambda_{\max}^t$ is bounded below some small constant. In this case, $\lambda_{\max}^K$ is also bounded below a small constant. Because the lower bound on the smallest global eigenvalue $\lambda_{\min}^K$ is proportional to the area $A_{\min}$ of the smallest triangle, $\kappa \in \mathcal{O}(1/A_{\min})$. If the triangles are of uniform size, $\kappa \propto 1/\ell^2$ where ℓ is the typical edge length. Since the area of the domain is fixed, $\kappa \propto n$ where n is the number of mesh vertices. (The number of vertices and elements is typically dictated by the need to limit the discretization error, and therefore the interpolation error.) Highly nonuniform meshes suffer from worse conditioning. This serves as a reminder that the urge to refine meshes to reduce interpolation and discretization errors can lead to other sorts of trouble.

If t is a linear tetrahedron, the element stiffness matrix is

$$K_t = \frac{1}{6} \begin{bmatrix} \sum_{1\neq i<j} \ell_{ij}\cot\theta_{ij} & -\ell_{34}\cot\theta_{34} & -\ell_{24}\cot\theta_{24} & -\ell_{23}\cot\theta_{23} \\ -\ell_{34}\cot\theta_{34} & \sum_{2\neq i<j\neq 2} \ell_{ij}\cot\theta_{ij} & -\ell_{14}\cot\theta_{14} & -\ell_{13}\cot\theta_{13} \\ -\ell_{24}\cot\theta_{24} & -\ell_{14}\cot\theta_{14} & \sum_{3\neq i<j\neq 3} \ell_{ij}\cot\theta_{ij} & -\ell_{12}\cot\theta_{12} \\ -\ell_{23}\cot\theta_{23} & -\ell_{13}\cot\theta_{13} & -\ell_{12}\cot\theta_{12} & \sum_{i<j\neq 4} \ell_{ij}\cot\theta_{ij} \end{bmatrix}.$$

Tedious manipulation reveals that the characteristic polynomial of K_t is

$$p(\lambda) = \lambda^4 - \frac{\sum_{i=1}^4 A_i^2}{9V}\lambda^3 + \frac{\sum_{1\leq j<k\leq 4} \ell_{jk}^2}{36}\lambda^2 - \frac{V}{9}\lambda.$$

There does not seem to be a simple expression for the roots of this polynomial (except the smallest root $\lambda = 0$), but they can be found numerically or by the cubic equation. However, these are expensive computations. Furthermore, if we wish to use numerical optimization methods to move the vertices and improve the element quality, it is helpful to be able to compute the gradient of $\lambda_{\max}^t$ with respect to the vertex positions. This gradient is expensive to compute without a closed-form expression, and it is singular for an equilateral tetrahedron. For these reasons, a simpler and smoother measure of the conditioning of K_t is useful.

An estimate of $\lambda_{\max}^t$ follows from the fact that each element stiffness matrix is known to be positive indefinite, so all the eigenvalues are nonnegative. The second coefficient of the characteristic polynomial is the (negated) sum of the eigenvalues, one of which is known to be zero, so

$$\frac{\sum_{i=1}^4 A_i^2}{27V} \leq \lambda_{\max}^t \leq \frac{\sum_{i=1}^4 A_i^2}{9V},$$

giving upper and lower bounds tight to within a factor of three.

$\lambda_{\max}^t$ and $\lambda_{\max}^K$ are not scale-invariant (as they are for triangles). If t is scaled uniformly, $\lambda_{\max}^t$ grows linearly with $\ell_{\max}$. Thus, the largest tetrahedron in a mesh may determine the largest eigenvalue of the global stiffness matrix, and the shapes of the largest tetrahedra may be more important than the shapes of the smaller ones. However, this must not be misinterpreted to imply that refining tetrahedra is always a good way to improve the condition number, because $\lambda_{\min}^K$ is proportional to the volumes of the tetrahedra. A better recommendation is to use tetrahedra that have good shapes and are as uniform as they can be without compromising speed or interpolation accuracy. To judge tetrahedron shapes, Section 4 discusses how to define scale-invariant quality measures related to $\lambda_{\max}^t$.

If the mesh has no badly-shaped tetrahedra, the largest global eigenvalue $\lambda_{\max}^K$ is proportional to the length $\ell_{\max}$ of the longest edge in the entire mesh. The lower bound on $\lambda_{\min}^K$ is proportional to the volume $V_{\min}$ of the smallest tetrahedron, so $\kappa \in \mathcal{O}(\ell_{\max}/V_{\min})$. If the tetrahedra are of uniform size, $\kappa \propto 1/\ell^2$, just like in the two-dimensional case. Hence, $\kappa \propto n^{2/3}$. However, nonuniform meshes and meshes with poorly-shaped tetrahedra can have much worse conditioning.

4 Quality Measures

Ideally, an algorithm for mesh generation or mesh improvement would optimize the fidelity of the interpolated surface over the mesh, or the accuracy of the approximate solution of a system of partial differential equations. However, these criteria are difficult and expensive to measure. At any rate, a single mesh is typically used to interpolate several different surfaces, or to solve several different numerical problems.

Instead, mesh generation and improvement algorithms usually select a single, easily-computed quality measure to evaluate the individual elements they create. For instance, a program might try to maximize the minimum angle.

Table 3 tabulates several quality measures for evaluating triangular and tetrahedral elements. These measures are related to the fitness of the elements for interpolation and stiffness matrix formation. For each measure, the higher the value of the quality measure, the better the element. All the measures are positive for properly oriented elements, zero (or undefined) for degenerate elements, and negative for inverted elements. Some of the measures depend on both size and shape, and some (the scale-invariant measures) depend on shape only.

In an ideal mesh, the sizes of the elements are controlled by the need to bound $\|f - g\|$ and $\|\nabla f - \nabla g\|$, for which purpose the error bounds are intended. Each element should be small enough that both these errors are below some application-determined bound—but no smaller, because an application's running time is tied to the number of elements.

The shapes of elements are usually controlled by the need to bound $\|\nabla f - \nabla g\|$ and the largest eigenvalue of the element stiffness matrix. For these purposes either the size-and-shape quality measures or the scale-invariant quality measures might be best, depending on the circumstances. For applications that have no stiffness matrix and do not care about accurate gradients (the latter being unusual), the shapes of elements are controlled by the need to bound $\|f - g\|$. (For most applications, the scale-invariant measures related to $\|f - g\|_\infty$ should rarely be used.)

The size-and-shape quality measures for interpolation are just the reciprocals of the error bounds. (Constants have been dropped because the measures are used only to compare elements.) Maximizing an element's measure is equivalent to minimizing its error. The reciprocals of the error bounds, rather than the error bounds themselves, are preferable for several reasons: the reciprocals are not infinite for degenerate elements; they vary continuously from negative for inverted elements to positive for correctly oriented elements; and they have better behaved derivatives (with respect to the position of each vertex), a helpful property for optimization-based mesh smoothing methods. Some of the quality measures vary smoothly with the vertex positions, and some do not. The smooth measures simplify optimization-based smoothing, but they are based on weaker bounds, so they are less accurate indicators than the nonsmooth measures.

For some purposes, scale-invariant measures of quality are more appropriate. Size-and-shape measures give no consideration to the number of elements needed to solve a problem, but the number should be controlled, because the computation time for applications is at least linearly proportional to the number of elements. The number of elements needed to cover a domain is inversely proportional to their average area or volume. How can we measure an element's ability to offer low error and high volume? The first impulse might be to express the ratio of error to area or volume, but the error $\|\nabla f - \nabla g\|$ varies according to the square root of area or the cube root of volume, so a ratio is not appropriate.

A better idea is to use a measure that compares an element's error bound with other elements of the same area or volume. The method I advocate here is to scale an element t uniformly until its area or volume is one, then evaluate its quality using the size-and-shape quality measure. This two-step procedure can be replaced by a single formula that yields exactly the same result. To find this formula, begin with the size-and-shape quality measure. Multiply every length by a scaling factor s, every area by s^2, and every volume by s^3. Then set $s = A^{-1/2}$ if t is a triangle, or $s = V^{-1/3}$ if t is a tetrahedron, thereby scaling the element so its area or volume is one.

Table 3: Quality measures related to interpolation error or stiffness matrix conditioning for a single element. $\lambda_{\max}$ is the largest eigenvalue of an element stiffness matrix (see Section 3), and is computed numerically from the characteristic polynomial or by the cubic equation. See Section 1 for explanations of other notation.

	Triangles	Tetrahedra
Interpolation quality measures, based on $\|f-g\|_\infty$		
Size and shape (mostly size)	$\frac{1}{c_t r_{\mathrm{mc}}^2}$	$\frac{1}{c_t r_{\mathrm{mc}}^2}$
Scale-invariant (rarely useful)	$\frac{A}{r_{\mathrm{mc}}^2}$	$\frac{V}{r_{\mathrm{mc}}^3}$
Interpolation quality measures, based on $\|\nabla f-\nabla g\|_\infty$		
Size and shape	$\frac{A}{c_t \ell_{\max}\ell_{\mathrm{med}}(\ell_{\min}+4\|r_{\mathrm{in}}\|)}$	$\frac{V\sum_{m=1}^4 A_m}{c_t(\sum_{1\le i<j\le 4} A_iA_j\ell_{ij}^2+6\|V\|\max_i\sum_{j\ne i}A_j\ell_{ij})}$
Size and shape (smooth)	$\frac{A}{c_t\ell_1\ell_2\ell_3}$	$\frac{V\sum_{m=1}^4 A_m}{c_t\sum_{1\le i<j\le 4} A_iA_j\ell_{ij}^2}$
Scale-invariant	$\frac{A}{(\ell_{\max}\ell_{\mathrm{med}}(\ell_{\min}+4\|r_{\mathrm{in}}\|))^{2/3}}$	$V\left(\frac{\sum_{m=1}^4 A_m}{\sum_{1\le i<j\le 4} A_iA_j\ell_{ij}^2+6\|V\|\max_i\sum_{j\ne i}A_j\ell_{ij}}\right)^{3/4}$
Scale-invariant (smooth)	$\frac{A}{(\ell_1\ell_2\ell_3)^{2/3}}$	$V\left(\frac{\sum_{m=1}^4 A_m}{\sum_{1\le i<j\le 4} A_iA_j\ell_{ij}^2}\right)^{3/4}$
Conditioning quality measures		
Scale-invariant	$\frac{A}{3\ell_{\mathrm{rms}}^2+\sqrt{(3\ell_{\mathrm{rms}}^2)^2-48A^2}}$	$\frac{V^{1/4}}{\lambda_{\max}^{3/4}}$
Scale-invariant (smooth)	$\frac{A}{\ell_{\mathrm{rms}}^2}$	$\frac{V}{A_{\mathrm{rms}}^{3/2}}$

This procedure converts the measure $A/(c_t\ell_1\ell_2\ell_3)$ to $A^{3/2}/(\ell_1\ell_2\ell_3)$. (The constant c_t is dropped because it is irrelevant for shape comparisons. The curvature bound determines the ideal size, but not the ideal shape.) This measure is flawed for two reasons: it is undefined if A is negative, and its gradient (with respect to the position of any vertex) is zero for degenerate elements, which can be crippling if the measure is used as an objective function for optimization-based smoothing. This problem is fixed by raising the quality measure to a power of $2/3$ to ensure that the numerator is A, yielding the quality measure $A/(\ell_1\ell_2\ell_3)^{2/3}$. (For the tetrahedral measure, use a power of $3/4$ to ensure that the numerator is V.) The justification for doing this is that raising the quality measure to a power does not change which element is preferred in any comparison. The same treatment generates all the scale-invariant measures in Table 3.

For matrix conditioning, only scale-invariant measures are offered here. Unlike with interpolation error, the effect of an element's size on conditioning cannot be judged in isolation. To minimize the condition number, the element sizes should be kept as uniform as possible. A quality measure for a single element cannot measure this uniformity.

Figure 5 illustrates the two nonsmooth quality measures related to $\|\nabla f-\nabla g\|_\infty$ over a triangle. In these contour plots, two vertices of a triangle are fixed at the coordinates $(0,0)$ and $(1,0)$, and the third vertex varies freely. The contours illustrate the quality of the triangle as a function of the position of the third vertex: the lighter the region, the higher the quality. Observe that the scale-invariant measure penalizes small angles more strongly than the size-and-shape measure, because triangles with small angles consume computation time without covering much area.

Figure 6 illustrates the two quality measures related to $\|\nabla f-\nabla g\|_\infty$ over a tetrahedron. Three vertices of a tetrahedron are fixed at the coordinates $(0,0,0)$, $(\sqrt{3}/2,1/2,0)$, and $(\sqrt{3}/2,-1/2,0)$, and the fourth vertex varies freely along the x- and z-axes. Each plot depicts a cross-section of space, $y=0$, as Figure 7 shows.

Figure 8 illustrates the two quality measures (one for triangles, one for tetrahedra) associated with the maximum eigenvalue of the element stiffness matrix.

Should you use an error bound, a size-and-shape measure, or a scale-invariant measure? The answer depends on where and how the bound or measure is used. The rest of this section gives suggested answers for mesh refinement, mesh smoothing, topological transformations, and point placement. For mesh refinement and topological transformations, the error bounds are most useful when an application can

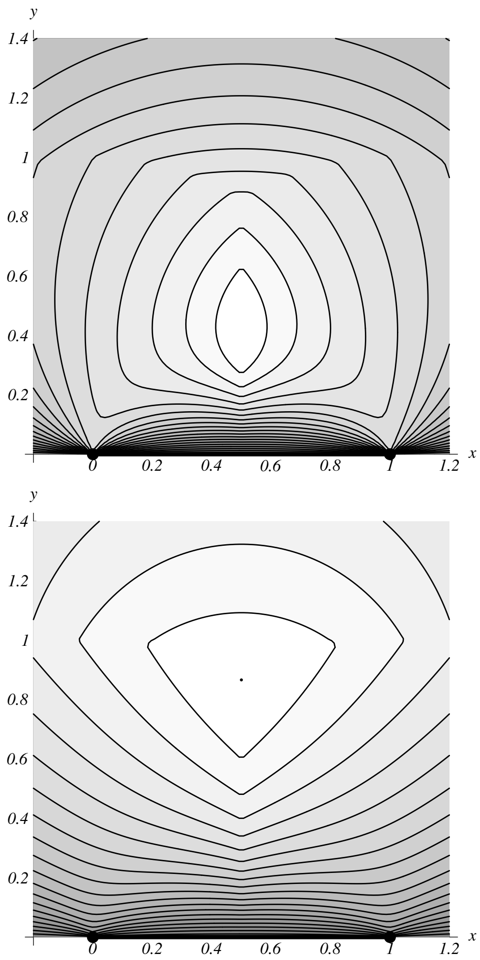

Figure 5: Two nonsmooth interpolation-based quality measures for a triangle with vertices $(0, 0)$, $(1, 0)$, and (x, y). The top measure is a size-and-shape measure; the bottom measure is scale-invariant.

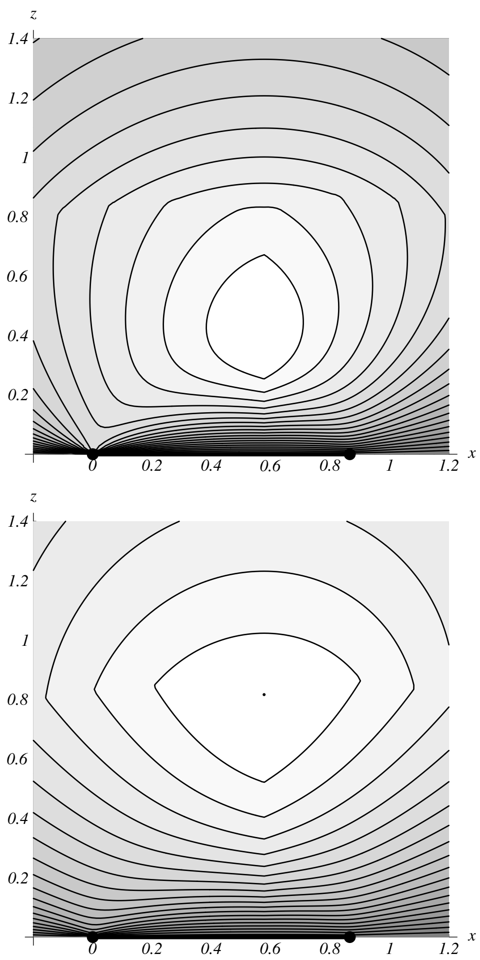

Figure 6: Two nonsmooth interpolation-based quality measures for a tetrahedron with vertices $(0, 0, 0)$, $(\sqrt{3}/2, 1/2, 0)$, $(\sqrt{3}/2, -1/2, 0)$, and $(x, 0, z)$. The top measure is a size-and-shape measure; the bottom measure is scale-invariant.

establish pointwise upper bounds on the sizes of the interpolation errors it is willing to accept. (For mesh smoothing, such bounds are unnecessary.)

Mesh refinement. In mesh refinement algorithms, including Delaunay refinement, an element is refined if it is too large or badly shaped. There is no need to try to wrap up element quality into a single measure; instead, an element can be required to pass separate tests for interpolation error and stiffness matrix conditioning.

To control interpolation accuracy, the error bounds are most appropriate. There are two such bounds—one related to the absolute interpolation error, and one related to the error in the gradient. The application that uses the mesh should set a pointwise maximum for one or both of these errors. A mesh refinement program can compare each element against both error bounds, and refine any element that fails either test.

If bounds on the maximum allowable errors are not available, or if estimates for c_t are not available, another approach is to choose an upper limit on the number of elements, and repeatedly refine the element with the smallest size-and-shape measure. This has the effect of keeping interpolation error bounds as uniform across elements as possible. (If c_t is unknown, simply drop it from the measure.)

An advantage of error bounds and size-and-shape measures over scale-invariant measures for mesh refinement is that the restrictions on shape are gradually relaxed as the element sizes decrease, so overrefinement is less likely.

Only scale-invariant measures are available for matrix conditioning, and these can be dangerous in the context of refinement, because the creation of smaller elements can occasionally decrease $\lambda_{\min}^K$ and worsen the conditioning of the stiffness matrix. Refining an element to achieve a slight improvement in its smallest angle can be a false economy. One option is to use a measure for matrix conditioning to compare

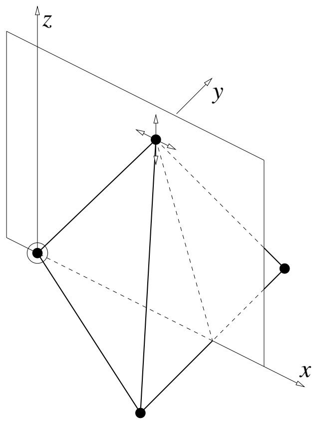

Figure 7: The tetrahedron configuration and cross-section of space used to plot the tetrahedron quality measures in Figures 6 and 8.

an element with the elements that will appear if the original element is refined. If the worst new element is better than the original by a margin large enough to justify the smaller elements, go ahead and refine the original element. If it is not possible to determine the new elements in advance, then the scale-invariant measure can be used with a weak bound on the minimum acceptable quality, so that an element is refined if it is likely to be replaced by much better shaped elements.

Optimization-based mesh smoothing. As I have mentioned, the error bounds are poorly behaved objective functions for numerical optimization, and the quality measures behave much better. For applications in which interpolation error is important but matrix conditioning is not, the size-and-shape measures are suitable for smoothing because they make appropriate tradeoffs between the size and shape of an element. Note, however, that because the size-and-shape measures do not penalize small angles harshly, an optimization-based smoother must take extra care not to create degenerate or inverted elements.

There is at least one common circumstance in which the scale-invariant measures might do better. Suppose the input to the smoother is a graded mesh generated by some other program that had access to information about the ideal sizes of elements, but the mesh smoother does not have that information. (This information might include the value of c_t for each triangle and a function that specifies the maximum allowable interpolation error at each point in the domain.) In this circumstance, the size-and-shape measures will try to make the mesh more uniform, whereas the scale-invariant measures will better preserve the original sizes of the elements.

Unfortunately, optimization-based smoothing can only optimize one objective function. For applications in which only interpolation error is important, the quality measure associated with $\|\nabla f - \nabla g\|_\infty$ is recommended over the measure associated with $\|f - g\|_\infty$, because the error in the gradients

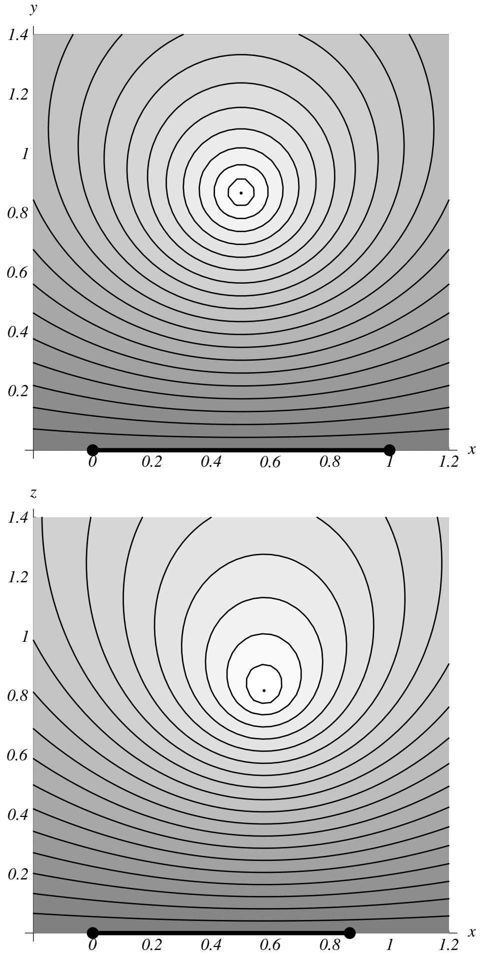

Figure 8: Two matrix conditioning-based scale-invariant quality measures. At top are level sets of $A/(\sum \ell_i^2 + \sqrt{(\sum \ell_i^2)^2 - 48A^2})$ for a triangle with vertices $(0,0)$, $(1,0)$, and (x,y). Below that are level sets of $V^{1/4}/\lambda_{\max}^{3/4}$ for a tetrahedron with vertices $(0,0,0)$, $(\sqrt{3}/2, 1/2, 0)$, $(\sqrt{3}/2, -1/2, 0)$, and $(x, 0, z)$.

can approach infinity if the element shape is poor. For applications in which matrix conditioning is important, there is a natural tension between the needs of interpolation and matrix conditioning. This tension can be resolved by using a conditioning measure alone, or by using a "combined" measure that is a weighted harmonic mean of two scale-invariant measures. For example, given an interpolation-based measure Q_1 and a conditioning-based measure Q_2, define a "combined" measure Q where

$$\frac{1}{Q} = \frac{w}{Q_1} + \frac{1-w}{Q_2}$$

and the constant weights w and $1 - w$ are chosen according to how much influence each measure should have. An advantage of the harmonic mean is that if one of the original measures assigns a low score (near zero) to an element, its opinion dominates.

Topological transformations. Although topological transformations and smoothing are both mesh improvement methods, they differ in that transformations can change the number of elements. The size-and-shape measures related to interpolation tend to prefer transformations that increase the number of elements, so they run the risk of overrefining the mesh. This pitfall is avoided by the use of an error threshold, probably the same threshold used for mesh refinement. Specifically, a topological transformation that increases the number of elements is performed only if it eliminates an element whose bound on interpolation error exceeds the threshold, and all the new elements are better. If bounds on the maximum allowable errors are not available, an alternative is to use scale-invariant measures.

For applications in which matrix conditioning is important, the comments on smoothing apply to topological transformations as well.

Vertex placement in advancing front methods. An advancing front mesh generator should try to place the largest possible element whose bounds on interpolation error meet the prescribed thresholds, and which perhaps meets a threshold on a conditioning measure as well. (The conditioning measure is scale-invariant; the error bounds are not.)

5 Related Work

Error estimates and quality measures for finite elements have been the subject of much research. Only a tiny fraction can be mentioned here.

Much of the work on error estimates (including the aforementioned paper of Babuška and Aziz [2]) is built on functional analysis and embedding theorems. Apel's habilitation [1, especially Section 10] includes an excellent summary. These results are asymptotic in nature, and ignore the constants associated with the error bounds. The premise of this paper is that small constants and nearly-tight bounds are valuable, because quality measures based on precise bounds are better able to choose among differently-shaped elements of similar quality, or to trade off element size against element shape.

Some *a posteriori* error indicators are not asymptotic. Notable examples include indicators proposed by Berzins [4], which estimate the interpolation error by approximating the true local solution by a quadratic function. The two main distinctions between the present work and Berzins' are that Berzin's indicators are approximations (not true upper and lower bounds as here), and they are for *a posteriori* use (whereas the bounds given here can be used as either *a priori* or *a posteriori* error estimates).

Many quality measures have appeared in the meshing literature; see Field [6] for a survey. However, none of these quality measures appears to have been derived from error estimates, nor from the study of stiffness matrix conditioning. (Knupp [9] presents quality measures based on the condition number of a transformation matrix, but it is not an element stiffness matrix.) Section 4 demonstrates that quality measures can be derived from these numerical considerations. Mesh generators that optimize these measures are likely to come closer to the goal of minimizing the interpolation and discretization errors (for a fixed number of elements), or minimizing the condition number of the global stiffness matrix.

References

[1] Thomas Apel. *Anisotropic Finite Elements: Local Estimates and Applications*. Technical Report SFB393/99-03, Technische Universität Chemnitz, January 1999. Habilitation.

[2] Ivo Babuška and A. K. Aziz. *On the Angle Condition in the Finite Element Method*. SIAM Journal on Numerical Analysis **13**(2):214–226, April 1976.

[3] Eric B. Becker, Graham F. Carey, and John Tinsley Oden. *Finite Elements: An Introduction*. Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[4] M. Berzins. *A Solution-Based Triangular and Tetrahedral Mesh Quality Indicator*. SIAM Journal on Scientific Computing **19**(6):2051–2060, November 1998.

[5] J. Bramble and M. Zlámal. *Triangular Elements in the Finite Element Method*. Mathematics of Computation **24**:809–820, 1970.

[6] David A. Field. *Qualitative Measures for Initial Meshes*. International Journal for Numerical Methods in Engineering **47**:887–906, 2000.

[7] Isaac Fried. *Condition of Finite Element Matrices Generated from Nonuniform Meshes*. AIAA Journal **10**(2):219–221, February 1972.

[8] Claes Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, New York, 1987.

[9] Patrick Knupp. *Matrix Norms & the Condition Number: A General Framework to Improve Mesh Quality via Node-Movement*. Eighth International Meshing Roundtable (Lake Tahoe, California), pages 13–22, October 1999.

[10] V. T. Rajan. *Optimality of the Delaunay Triangulation in* $\mathbb{R}^d$. Proceedings of the Seventh Annual Symposium on Computational Geometry, pages 357–363, 1991.

[11] Gilbert Strang and George J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[12] Shayne Waldron. *The Error in Linear Interpolation at the Vertices of a Simplex*. SIAM Journal on Numerical Analysis **35**(3):1191–1200, 1998.