WELL-CENTERED TRIANGULATION[¶]

EVAN VANDERZEE[*], ANIL N. HIRANI[†], DAMRONG GUOY[‡], AND EDGAR RAMOS[§]

Abstract. Well-centered meshes (meshes composed of well-centered simplices) have the advantage of having nice orthogonal dual meshes (the dual Voronoi diagram), which is useful for certain numerical algorithms that require or prefer such primal-dual mesh pairs. We present a characterization of a well-centered n-simplex and introduce a cost function that quantifies well-centeredness of a simplicial mesh. We investigate some properties of the cost function and describe an iterative algorithm for optimizing the cost function. The algorithm can transform a given triangulation into a well-centered one by moving the interior vertices while keeping the mesh connectivity and boundary vertices fixed. We show the results of applying our algorithm to small, large, and graded two-dimensional meshes as well as one tiny three-dimensional mesh and a small tetrahedralization of the cube. Also, we prove for planar meshes that the optimal triangulation with respect to the cost function is the minmax angle triangulation.

Key words. well-centered, meshing, mesh optimization, acute, triangulation, discrete exterior calculus

AMS subject classifications. 65N50, 65M50, 65D18, 51M04

1. Introduction.

A completely *well-centered* mesh is a simplicial mesh in which each simplex contains its circumcenter. A 3-dimensional example is a tetrahedral mesh in which the circumcenter of each tetrahedron lies inside it, and the circumcenter of each triangle face lies inside it. In two dimensions, a well-centered triangulation is that same thing as an acute triangulation. In this paper we discuss well-centered triangulations, with particular application to triangle and tetrahedral meshes. Typical meshing algorithms do not guarantee well-centeredness. For example, a Delaunay triangulation is not necessarily well-centered. We present an iterative energy minimization approach in which a given mesh, after possible preprocessing, is made well-centered by moving the internal vertices while keeping the boundary vertices and connectivity fixed.

A well-centered (primal) mesh has a corresponding dual mesh assembled from a circumcentric subdivision [16]. For an n-dimensional primal mesh, a k-simplex in the primal corresponds to an $(n-k)$-cell in the dual. For example, in a well-centered planar triangle mesh, the dual of a primal interior vertex is a convex polygon with boundary edges that are orthogonal and dual to primal edges. This orthogonality makes it possible to discretize the Hodge star operator of exterior calculus [1] as a diagonal matrix which simplifies certain computational methods for solving partial differential equations. Some numerical methods that mention well-centered meshes in this context are the covolume method [21] and Discrete Exterior Calculus [8, 16].

[¶] Preliminary results for the 2-dimensional problem of well-centered planar triangulations appeared previously in the Proceedings of the 16th International Meshing Roundtable, Seattle, WA, October 14-17, 2007 [25].

[*] Department of Mathematics, 1409 W. Green Street, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (vanderze@uiuc.edu). Research supported by CSE Fellowship from the Computational Science and Engineering Program and Applied Mathematics Program, UIUC.

[†] Author for Correspondence, Department of Computer Science, 201. N. Goodwin Avenue, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (hirani@cs.uiuc.edu). Research supported by NSF CAREER Award, Grant No. DMS-0645604.

[‡] Computational Science and Engineering Program, Center for Simulation of Advanced Rockets, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (guoy@uiuc.edu)

[§] edgar.a.ramos@gmail.com

Well-centered meshes are not strictly required for these or other related methods. However, some computations may be easier if such meshes were available. For example, a stable mixed finite volume type method for Darcy flow has recently been derived using Discrete Exterior Calculus [17] and applied to well-centered meshes generated by our code. That numerical method passes patch tests in 2 and 3 dimensions, for homogeneous and heterogeneous problems. Figure 1.1 (taken from [17] by permission of authors) shows the velocities in a layered medium in the Darcy flow problem.

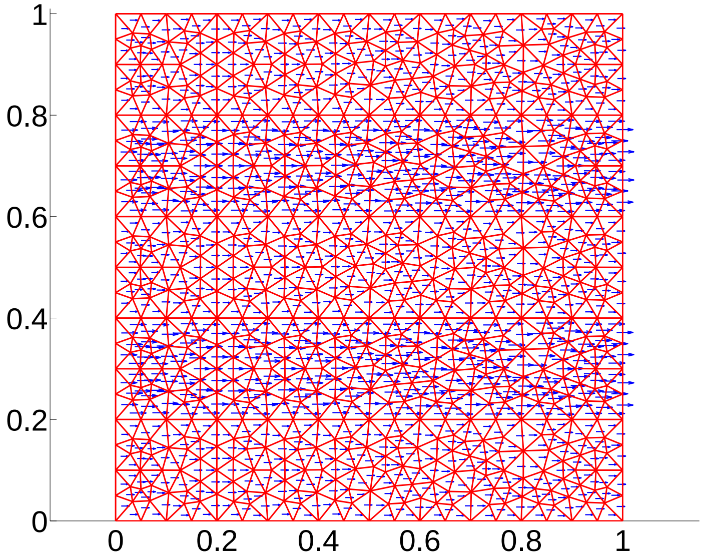

Fig. 1.1. *Darcy flow in a medium with 5 layers which is well-centered. The odd layers have a permeability of 5 and even layers have permeability of 10. The velocities in the odd and even layers should be different and should have no vertical component as shown. The mesh was generated using our WCT code. Figure taken from [17], used by permission from authors.*

Another example from scientific computing is space-time meshing. When tent-pitching methods for space-time meshing were first introduced, the initial spatial mesh was required to be acute, which for two-dimensional meshes, is the same thing as being well-centered [24]. More recently, this requirement has been avoided, although at the expense of some optimality in the construction [13].

2. Previous Results.

We are primarily concerned with triangulations for which the domain is specified by a polygonal or polyhedral boundary. In addition to the triangulations being well-centered, we are interested in *quality triangulations*. In two dimensions, a quality triangulation is one for which a lower bound on the triangle angles is achieved. Relevant work can be divided into constructive and iterative approaches.

Constructive approaches start with the specified input boundary/constraints and generate additional points, called *Steiner points*, and a corresponding triangulation. Normally a point is committed to a position and never moved afterwards. An algorithm for nonobtuse planar triangulations based on circle packings is described in [3], and more recent works describe improved constructions while also describing how to derive an acute triangulation from a nonobtuse one [19, 26]. In fact, these algorithms aim to achieve a triangulation of size linear in the input size, so the smallest angle can be arbitrarily close to zero. It is not straightforward to extend this class of algorithms to well-centered quality triangulations. As recently as 2007, a variant of the Delaunay refinement algorithm is proposed for generating acute triangulations of planar domains [14]. This algorithm, which relocates Steiner points after they are added, appears to work well. The authors do not, however, have a proof that the algorithm will terminate [14], and the experiments suggest that the maximum angle is often near $\pi/2$.

There are also constructive algorithms that achieve well-centered quality triangulations for point sets [4], or nonobtuse quality triangulations [20]. Also relevant is an algorithm that, given a constraint set of both points and segments in the plane, finds a triangulation that minimizes the maximum angle [11], without adding points. If an acute triangulation exists for the input constraints, the algorithm will find it, otherwise it fails. The most promising of the constructive algorithms is probably [14], but that algorithm as well as most other constructive approaches is specific to two-dimensional planar domains.

On the other hand, there are *iterative* or *optimization* approaches which allow an initial triangulation (possibly the canonical Delaunay) and then move the points while possibly changing the connectivity. This is the class of algorithms in which we are primarily interested: there are well-known algorithms to generate quality triangulations [10, 22] for which reliable implementations exist. Such quality triangulations may be good candidates as starting points for iterative approaches that seek well-centered meshes. In this class there are optimization approaches like centroidal Voronoi diagrams [9] and variational triangulations [2]. Each approach has a global cost function that it attempts to minimize through an iterative procedure that alternates between updating the location of the mesh vertices and the triangulation of those vertices. The cost functions optimized in these approaches are designed to optimize certain qualities of the mesh elements, but they do not explicitly seek well-centered simplices.

Though these methods appear to produce nice experimental results, there is no guarantee that they construct quality triangulations, much less well-centered ones. In fact, in Section 6 we show two-dimensional examples in which the converging triangulation is not acute. Also, only limited convergence results are known (there are indeed local minima that can be reached). In addition to the optimization approaches that work directly with a mesh, there are several algorithms that generate circle packings or circle patterns by optimizing the radii of the circles. In particular the algorithms for creating circle patterns that were proposed in [7] and [5] can be adapted to create triangulations. These algorithms produce circle patterns that have specified combinatorics but they do not permit a complete specification of the domain boundary, so they are not appropriate to our purpose.

Finally, concerning the 3-dimensional case, we point out that the problem of generating a well-centered tetrahedralization in $\mathbb{R}^3$ is considerably harder than the two-dimensional analogue. Similarly, the problem of generating a three-dimensional acute triangulation—a tetrahedralization in which all the dihedral angles are acute—is more difficult than generating a two-dimensional acute triangulation. For tetrahedra, it is no longer true that well-centeredness and acuteness are equivalent. In addition, acute tetrahedralizations are known only for very restricted domains (for example, whole space and slabs [12]). In fact, it is not even known whether the cube has an acute tetrahedralization. In Sec. 6 we show that the cube can be triangulated with

3-well-centered tetrahedra.

3. Characterization of Well-Centeredness. It is possible for a simplex σ^n of dimension $n \geq 3$ to contain its circumcenter in its interior $\mathrm{Int}(\sigma^n)$, while some of its proper faces $\sigma^p \prec \sigma^n$ do not contain their circumcenters. Similarly, it is possible for a simplex σ^n that all of its proper faces $\sigma^p \prec \sigma^n$ do contain their circumcenters, while the simplex σ^n itself does not contain its circumcenter. For this reason we introduce the terminology of an n-simplex being $(p_1, \dots, p_k)$-well-centered, meaning that for p_i, $i = 1, \dots, k$, all faces of σ^n of dimension $p_i \leq n$ contain their circumcenters. The parentheses may be suppressed when referring to only one dimension.

In this section we give an alternate characterization for an n-simplex σ^n that is n-well-centered; we give a way of determining whether the simplex σ^n contains its circumcenter without explicitly computing its circumcenter and testing its orientation with respect to the facets, i.e., the faces $\sigma^{n-1} \prec \sigma^n$. We use the fact that in Euclidean space the intersection of two distinct n-spheres S^n_1, S^n_2 is contained in a hyperplane of dimension n, and that this hyperplane cuts the sphere S^n_1 into two pieces, one piece inside S^n_2, and the other piece outside S^n_2. We also use the concept of an equatorial ball. Given that the $(n-1)$-dimensional simplex $\sigma^{n-1} = v_0 v_1 \dots v_{n-1}$ is nondegenerate, the *equatorial ball* $B(\sigma^{n-1}) = B(v_0 v_1 \dots v_{n-1})$ is defined as follows. Since σ^{n-1} is nondegenerate, it has a unique $(n-2)$-dimensional circumsphere (embedded in a copy of $\mathbb{R}^{n-1}$), with center $c = c(\sigma^{n-1})$ and radius $R = R(\sigma^{n-1})$. We take $B(\sigma^{n-1})$ to be the closed n-dimensional ball having center $c(\sigma^{n-1})$ and radius $R(\sigma^{n-1})$, where the n-dimensional space containing $B(\sigma^{n-1})$ is the plane of a specified simplex σ^n such that $\sigma^{n-1} \prec \sigma^n$.

Theorem 3.1. *Let σ^n be an n-dimensional simplex with vertices $v_0, v_1, \dots, v_n$. The simplex σ^n is n-well-centered if and only if σ^n is nondegenerate and for each $i = 0, 1, \dots, n$, vertex v_i lies outside $B_i = B(v_0 v_1 \dots \hat{v}_i, \dots, v_n)$, with B_i taken in the plane of σ^n.*

Proof. First we suppose that σ^n is n-well-centered. Since σ^n is n-well-centered, its circumcenter is contained in its interior, and since a degenerate simplex has no interior, σ^n must be nondegenerate. This implies that all the facets of σ^n are also nondegenerate, so B_i is well-defined for each i.

Consider some vertex v_i of σ^n. Let $S^n = S^n(\sigma^n)$ be the circumsphere of σ^n. Thinking of σ^n in its plane, and identifying that plane with $\mathbb{R}^n$, σ^n is an intersection of half-spaces, and one of the bounding hyperplanes of σ^n is the hyperplane that contains σ^{n-1}_i, the simplex $v_0 v_1 \dots \hat{v}_i \dots v_n$. This hyperplane partitions $\mathbb{R}^n$ into two half-spaces — an open half-space H_1 that contains v_i and a closed half-space H_2 that contains all the other vertices of σ^n (on its boundary).

Because σ^n is well-centered, $c(\sigma^n)$ lies in its interior. It follows that $c(\sigma^n)$ lies in the same open half-space as v_i. Consider, then, the line through $c(\sigma^n)$ and $c(\sigma^{n-1}_i)$, the center of B_i. In the open half-space that contains v_i, this line intersects S^n at a point p with $d(c(\sigma^n), p) = R(\sigma^n)$. Moreover, $d(c(\sigma^{n-1}_i), p) > R(\sigma^n) > R(\sigma^{n-1}_i)$. We see that p lies outside B_i and conclude that $S^n \cap H_1$ lies outside B_i. In particular, since $v_i \in S^n \cap H_1$, we see that v_i lies outside B_i. Since v_i was chosen arbitrarily, we conclude that v_i lies outside B_i for each $i = 0, 1, \dots, n$, and one direction of the theorem is proved.

For the other direction we suppose that σ^n is a nondegenerate simplex such that v_i lies outside B_i for each $i = 0, 1, \dots, n$. We will show that the circumcenter $c(\sigma^n)$ lies in the interior of σ^n by demonstrating that for each vertex v_i, $c(\sigma^n)$ lies in the same open half-space as v_i relative to the plane of σ^{n-1}_i. We know that this hyperplane

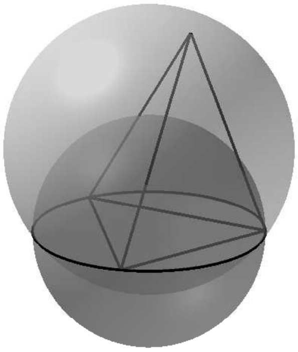

FIG. 3.1. *The circumsphere of the tetrahedron and the equatorial ball of one of its faces intersect in the circumcircle of the face. For a well-centered tetrahedron, the fourth vertex lies on the circumsphere in the region outside the equatorial ball.*

divides S^n into a part inside B_i and a part outside B_i, and we have just established that whichever (open) halfspace contains $c(\sigma^n)$ is the halfspace where S^n lies outside B_i. But since we are given that v_i lies outside B_i, we know that it lies in the same open halfspace that $c(\sigma^n)$ lies in. This holds for every v_i, so $c(\sigma^n)$ is in the interior of σ^n, and σ^n is, by definition, well-centered. □

Figure 3.1 is an illustration of Thm 3.1 for the case of one vertex of a tetrahedron. The fourth vertex lies outside the equatorial ball because the tetrahedron is 3-well-centered.

When we say that a mesh is $(p_1, \ldots, p_k)$-well-centered, we mean that the mesh is a simplicial complex and for p_i, $i = 1, \ldots, k$, every simplex of dimension p_i that appears in the simplicial complex contains its circumcenter.

4. Iterative Energy Minimization. Given a simplicial mesh, we iteratively modify the mesh guided by minimizing a cost function defined over the mesh. We'll refer to the cost function as *energy.* Our method is somewhat similar to the methods of [9] and [2] in that it uses an iterative procedure to minimize an energy defined on the mesh, but it differs in that the mesh topology and boundary vertices remain fixed as the energy is minimized. More importantly, we minimize a different energy, one that is designed to achieve the aim of well-centeredness. We next describe this energy, which is the main component of our method. At times the mesh connectivity or boundary vertices are defined in such a way that no well-centered mesh exists. For such cases one can apply a preprocessing algorithm to update the mesh connectivity. Section 5 outlines an approach to preprocessing that can change the mesh connectivity throughout the entire mesh. The algorithm we actually use is an algorithm that makes changes to the mesh locally in areas where problems exist. That algorithm is discussed in [25].

In the proof of Theorem 3.1 we see the importance of the position of the circumcenter of a simplex with respect to its facets. When the circumcenter is well inside of the simplex, it is far away from every facet, and when the circumcenter is near the boundary of the simplex, it is close to at least one of the facets of the simplex. Given a simplex σ^n, this concept is quantitatively measured for the facet of σ^n opposite vertex v by $h(v, \sigma^n)$, the height of the circumcenter above the plane of that facet. To be more precise, $h(v, \sigma^n)$ is the signed distance from the circumcenter of σ^n to the circumcenter of the facet opposite vertex v, with positive distance when the circumcenter lies on the same side of the plane of the facet as v and negative distance when the plane of the facet lies between v and the circumcenter of σ^n. Observe that a mesh is n-well-centered if and only if $h(v, \sigma^n) > 0$ for every vertex v of every n-simplex σ^n of the mesh.

Generally speaking one likes to preserve properties of a given mesh even while making whatever modifications are needed to achieve well-centeredness. In particular, it is good to preserve the relative size of elements of the given mesh, i.e., the grading of the initial mesh, where that is possible. For this reason we scale the measure $h(v, \sigma^n)$ by the circumradius of the simplex, denoted $R(\sigma^n)$. We see that a mesh is n-well-centered if and only if $h(v, \sigma^n)/R(\sigma^n) > 0$ for every vertex v of every n-simplex σ^n of the mesh. Sazonov et. al. have also noticed that the quantity h/R may be useful for measuring well-centeredness [23].

Note that $-1 < h(v, \sigma^n)/R(\sigma^n) < 1$ for finite nondegenerate σ^n, because $R(\sigma^n)^2 = h(v_i, \sigma^n)^2 + R(\sigma_i^{n-1})^2$. Instead of using the quantity h/R directly, we consider the measure

$$f_n(\sigma^n) = \max_{\text{vertices } v \in \sigma^n} \left| \frac{h(v, \sigma^n)}{R(\sigma^n)} - k_n \right| ,$$

where $0 < k_n \leq 1$ is a constant that may depend on the dimension of the simplex. The advantage of minimizing this measure as opposed to maximizing h/R is that if k_n is chosen properly, the measure penalizes simplex vertices where h/R approaches 1 (such as small angles in triangles or needle tetrahedra) as well as vertices where $h/R \leq 0$. It seems wise to choose k_n so that the expression is minimized by a regular simplex. Initially, taking $k_n = 1/n$ may seem like a good choice because it is clear that the regular simplex minimizes f_n. ($f_n = 0$ for the regular simplex if $k_n = 1/n$). We show in Lemma 4.1, however, that the regular simplex minimizes f_n for any $1 \geq k_n \geq 1/n$. Thus taking $k_n = 1/2$ is probably the best choice, since $k_n = 1/2$ is the minimal k_n such that every well-centered simplex is measured as better quality than every simplex that is not well-centered. We used $k_n = 1/2$ for all of the experiments we show in Sec. 6.

Lemma 4.1. *For $k_n \geq 1/n$, the measure $f_n(\sigma^n)$ is minimized when σ^n is a regular simplex.*

Proof. It suffices to show that for any simplex σ^n there exists a vertex v such that $h(v, \sigma^n) \leq R(\sigma^n)/n$, since at such a vertex we have

$$\left| \frac{h(v, \sigma^n)}{R(\sigma^n)} - k_n \right| = k_n - \frac{h(v, \sigma^n)}{R(\sigma^n)} \geq k_n - \frac{1}{n}$$

for any $k_n \geq 1/n$. We have seen that for a simplex that is not n-well-centered, there exists a vertex v with $h(v, \sigma^n) \leq 0$, so it remains to prove this for simplices that are n-well-centered.

Suppose σ^n is n-well-centered. Let $h = \min_v h(v, \sigma^n) > 0$. Consider a sphere S^{n-1} with center $c(\sigma^n)$ and radius h. We claim that σ^n contains the sphere S^{n-1}. Indeed, for each facet σ_i^{n-1} of σ^n, since the radius of S^{n-1} is $h \leq h(v_i, \sigma^n)$ we have that the sphere S^{n-1} is contained in the same half space as $c(\sigma^n)$. Thus the sphere is contained in the intersection of half spaces that defines the simplex, i.e., is contained in the simplex.

It follows, then, that $h \leq r$ where r is the inradius of the simplex. We know that $h/R \leq r/R \leq 1/n$ and that equality is achieved for only the regular simplex. (The inequality $r/R \leq 1/n$ is proved in [18], among others.) □

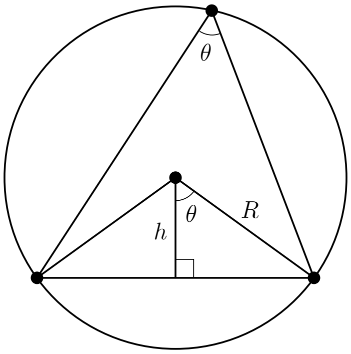

FIG. 4.1. *For a triangle,* $h/R = \cos(\theta)$.

For $k_n > 0$ the objective of n-well-centeredness is achieved when $|h/R - k_n| < k_n$ at every vertex of every simplex σ^n. Our goal, then, is to minimize $|h/R - k_n|$, driving it below k_n at every vertex. It could be effective to work directly with the measure $|h/R - k_n|$, but we choose instead to minimize the approximation

$$E_p\left(\mathcal{M}\right) = E_p\left(\mathcal{V}, \mathcal{T}\right) = \sum_{\substack{\text{simplices } \sigma^n \in \mathcal{T} \\ \text{vertices } v \in \sigma^n \cap \mathcal{V}}} \left| \frac{h(v, \sigma^n)}{R(\sigma^n)} - k_n \right|^p , \tag{4.1}$$

where p is a parameter. This is indeed an approximation, because

$$\lim_{p \to \infty} \left(E_p\left(\mathcal{M}\right)\right)^{1/p} = E_\infty\left(\mathcal{M}\right) = \max_{\substack{\sigma^n \in \mathcal{T} \\ v \in \sigma^n \cap \mathcal{V}}} \left| \frac{h(v, \sigma^n)}{R(\sigma^n)} - k_n \right| .$$

The parameter p influences the relative importance of the worst vertex-simplex pair compared to the other vertex-simplex pairs in computing the quality of the mesh as a whole. It is convenient to choose p as a positive even integer, since the absolute value need not be taken explicitly in those cases. $\mathcal{M}$ here stands for a mesh consisting of vertices with particular coordinates $\mathcal{V}$ and a connectivity table $\mathcal{T}$ that describes which groups of vertices form simplices.

As stated, the measure $E_p(\mathcal{M})$ leaves some ambiguity in the case of degenerate simplices. For several reasons, including a desire to maintain upper semicontinuity of the cost function, we say that any degenerate simplex, even one with coincident vertices, has its circumcenter at infinity and $h/R = -1$.

To help build intuition for the measure $|h/R - k_n|$, Fig. 4.1 displays the quantities h and R in a sample triangle. Elementary Euclidean geometry establishes that the two angles marked θ in Fig. 4.1 are congruent, and from triangle trigonometry we know that $\cos(\theta) = h/R$. This shows that the energy defined in equation (4.1) is a generalization of the energy

$$E_p(\mathcal{M}) = E_p\left(\mathcal{V}, \mathcal{T}\right) = \sum_{\theta \in \mathcal{M}} \left|\cos(\theta) - 1/2\right|^p , \tag{4.2}$$

which the authors proposed earlier for achieving well-centeredness of planar meshes [25]. In three dimensions the quantity h/R is related to the cosine of the tetrahedron vertex angle, as discussed in [23], and the relationship given there can be extended to higher dimensions.

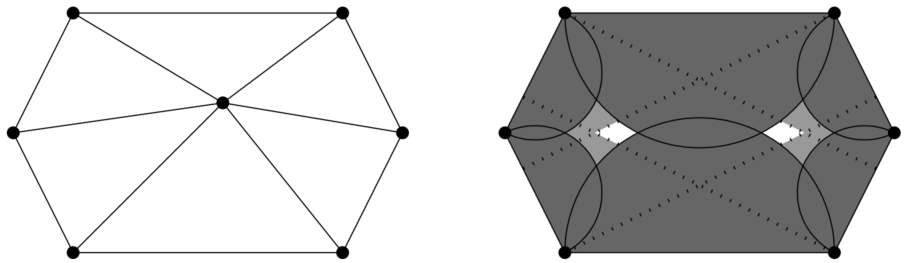

Fig. 4.2. *A cost function that accurately reflects the goal of well-centeredness cannot have a unique minimum, because the set of points that make the mesh well-centered may be a symmetric disconnected set.*

The cost functions E_p and E_∞ are not convex. When designing a cost function for mesh optimization, one might hope to develop a function that is convex, or, if not convex, at least one that has a unique minimum. It is, however, not possible to define an energy that accurately reflects the goals of well-centered meshing and also has a unique minimum. Consider the mesh shown on the left in Fig. 4.2. We suppose that the boundary vertices are fixed, but the interior vertex is free to move. We want to decide where to move the interior vertex in order to obtain a well-centered mesh. The right side of Fig. 4.2 shows the constraints on where the vertex can be placed to produce a well-centered mesh. The light gray regions are not allowed because placing the interior vertex in those regions would make some boundary angle nonacute. (The dotted lines indicate how the four most important boundary angles influence the definition of this region.) The darker gray regions, shown overlaying the light gray region, are not permitted because placing the interior vertex in those regions would make some angle at the interior vertex nonacute.

If the interior vertex is placed in either of the two small white regions that remain, the mesh will be well-centered. We see that the points permitted for well-centeredness form a disconnected set in $\mathbb{R}^2$. Moreover, the mesh is radially symmetric, so there is no way to create an energy that prefers one white region over the other unless we violate the desired property that the energy be insensitive to a rotation of the entire mesh. Any symmetric energy that has minima in only the white regions must have at least two distinct global minima.

In most planar meshes there is an interior vertex v that has exactly six neighbors, all of which are interior vertices. If all interior vertices are free to move, as we assume in the method we propose, then the six neighbors could be moved into the arrangement that the boundary vertices have in the mesh in Fig. 4.2. Moving v around when its neighbors have such a configuration should exhibit nonconvexity in whatever cost function we might define.

5. The Optimal Triangulation. A variety of our experimental results appears in Section 6 below. The results suggest that, in the plane at least, where we have performed most of our experiments, E_p is an appropriate cost function for quantifying the well-centeredness of a mesh. In some cases, though, the mesh connectivity, the fixed boundary vertices, or a combination of the two are specified in such a way that no well-centered mesh exists with the given topology and boundary vertices. The simplest example of this is a planar mesh with an interior vertex v that has fewer than five neighbors. Since the angles around v sum to 2π, v has some adjacent angle of at least $\pi/2$. The triangle containing that angle is not well-centered. Similarly, a boundary vertex with a boundary angle measuring at least $\pi/2$ must have enough interior neighbors to divide the boundary angle into pieces strictly smaller than $\pi/2$. We will refer to a vertex that does not have enough neighbors as a *lonely* vertex. (In three dimensions, a vertex must have at least 7 neighboring edges to permit a 3-well-centered mesh, though having 7 neighbors is not sufficient to guarantee that a 3-well-centered neighborhood exists.)

One way to approach problems with mesh connectivity, such as the problem of lonely vertices, is a global mesh connectivity update, i.e., to changing the mesh connectivity over the entire mesh. The methods that use Voronoi diagrams [9] and variational triangulations [2] both employ this approach, updating to a Delaunay mesh each time the vertices are relocated. We discuss here what constitutes the optimal triangulation with respect to E_∞ and explain why we have chosen to update the mesh connectivity as a preprocessing step, keeping the mesh connectivity fixed during energy minimization. In this section we consider the vertices $\mathcal{V}$ to be fixed at their initial locations, and given the fixed vertex locations we seek the mesh connectivity $\mathcal{T}$ that minimizes E_∞.

The discussion that follows is limited to planar meshes, so we use planar angles θ and the cosine-based cost function defined in 4.2 in the discussion that follows. In particular we consider the cost functions

$$E_{cos}\left(\mathcal{V},\mathcal{T}\right) = \lim_{p\to\infty}\left[\sum_{\theta\in\mathcal{M}}\left|\cos\left(\theta\right)-\frac{1}{2}\right|^{p}\right]^{1/p} = \max_{\theta\in\mathcal{M}}\left\{\left|\cos(\theta)-\frac{1}{2}\right|\right\}$$

$$E_{min}\left(\mathcal{V},\mathcal{T}\right) = \min_{\theta\in\mathcal{M}}\left\{\theta\right\}$$

$$E_{max}\left(\mathcal{V},\mathcal{T}\right) = \max_{\theta\in\mathcal{M}}\left\{\theta\right\},$$

where in the latter two cases we take $\theta \in [0,\pi]$. It is known that the Delaunay triangulation, which can be computed efficiently, maximizes the minimum angle of a mesh. In other words, $\mathcal{T} = \arg\max E_{min}$ is the Delaunay triangulation. There is also an $O(n^2 \log n)$ time algorithm for computing the minmax angle triangulation of a fixed set of points [11], so we have a reasonable way to compute $\mathcal{T} = \arg\min E_{max}$.

We claim that when all triangulations of a point set have a maximum angle that is at least $\pi/2$, a triangulation minimizing E_{max} is also a triangulation that minimizes E_{cos}. This claim is readily proved as a corollary of the following theorem.

Theorem 5.1. *Let f be a strictly increasing function of θ and g a nondecreasing function of θ for $\theta \in [0,\pi]$. If $E_f(\mathcal{T}) = \max\{f(\theta_i)\}$ and $E_g(\mathcal{T}) = \max\{g(\theta_i)\}$, then* $\arg\min E_f \subseteq \arg\min E_g$.

Proof. For each triangulation $\mathcal{T}$, there exists some angle $\theta_{\mathcal{T}}$ such that $E_f(\mathcal{T}) = \max\{f(\theta_i)\} = f(\theta_{\mathcal{T}})$. Thus for all other angles θ appearing in triangulation $\mathcal{T}$, we have that $f(\theta_{\mathcal{T}}) \geq f(\theta)$.

Now consider a specific $\mathcal{T}_0 \in \arg\min E_f$. We have $E_f(\mathcal{T}_0) \leq E_f(\mathcal{T})$ for all admissible triangulations $\mathcal{T}$. It follows that $f(\theta_{\mathcal{T}_0}) \leq f(\theta_{\mathcal{T}})$ for all admissible triangulations $\mathcal{T}$, since $E_f(\mathcal{T}) = f(\theta_{\mathcal{T}})$. Moreover, since f is a strictly increasing function of θ, we can conclude that $\theta_{\mathcal{T}_0} \leq \theta_{\mathcal{T}}$ for all admissible triangulations $\mathcal{T}$. Then since g is a nondecreasing function of θ, we have $g(\theta_{\mathcal{T}_0}) \leq g(\theta_{\mathcal{T}})$ for all admissible $\mathcal{T}$.

Now we claim that for arbitrary triangulation $\mathcal{T}$ we have that $g(\theta_{\mathcal{T}}) \geq g(\theta)$ for all angles θ appearing in triangulation $\mathcal{T}$. If this were not the case, then there would exist some angle $\hat{\theta}$ in $\mathcal{T}$ with $g(\hat{\theta}) > g(\theta_{\mathcal{T}})$. Since g is nondecreasing, it would follow that $\hat{\theta} > \theta_{\mathcal{T}}$, and since f is strictly increasing, we would have $f(\hat{\theta}) > f(\theta_{\mathcal{T}})$. This, however, contradicts our definition of $\theta_{\mathcal{T}}$, which states that $f(\theta_{\mathcal{T}}) = \max\{f(\theta_i)\} \geq f(\hat{\theta})$. We conclude that the claim is correct.

It follows, then, that $g(\theta_{\mathcal{T}}) = \max\{g(\theta_i)\} = E_g(\mathcal{T})$ for each triangulation $\mathcal{T}$. In particular, the inequality $g(\theta_{\mathcal{T}_0}) \leq g(\theta_{\mathcal{T}})$ implies that $E_g(\mathcal{T}_0) \leq E_g(\mathcal{T})$ for all admissible triangulations $\mathcal{T}$. By definition, $\mathcal{T}_0$ is a member of the set $\arg\min E_g$. □

Corollary 5.2. *If f is a strictly increasing function of θ for $\theta \in [0, \pi]$, then* $\arg\min E_f = \arg\min E_{max}$.

Proof. The function E_{max} is of the form E_g where g is the identity function on $[0, \pi]$. Since g is a strictly increasing function, we may apply the theorem in both directions to show that $\arg\min E_f \subseteq \arg\min E_{max}$ and that $\arg\min E_{max} \subseteq \arg\min E_f$. We conclude that $\arg\min E_{max} = \arg\min E_f$. □

Corollary 5.3. *If all triangulations of a set V of vertices have maximum angle at least $\pi/2$, then a triangulation minimizing E_{max} also minimizes E_{cos} and vice versa.*

Proof. We can restate the corollary as follows. If $E_{max} \geq \pi/2$ for all admissible triangulations $\mathcal{T}$, then $\arg\min E_{cos} = \arg\min E_{max}$. This follows because E_{cos} is of the form E_f where $f = |\cos(\theta) - 1/2|$ is a strictly increasing function on the interval $[\pi/2, \pi]$, and $f(\theta) < f(\pi/2)$ for $0 < \theta < \pi/2$. For all practical purposes, we could redefine f on $[0, \pi/2]$ and make it a strictly increasing function on $[0, \pi]$ because we know that for all admissible $\mathcal{T}$, the maximal $f(\theta_i)$ occurs for some $\theta_i \geq \pi/2$.

Some care must be taken for meshes that have an angle $\theta = 0$, but we know that a triangle with an angle of 0 has some angle measuring at least $\pi/2$, even if two of the triangle vertices coincide. Since $f(\pi/2) = f(0)$, we may say that on the triangle f is maximized at the largest angle $\theta \geq \pi/2$. □

It should be clear that if if a triangulation exists with $E_{max} < \pi/2$, this kind of reasoning no longer applies. In that case, E_{cos} may be maximized at some angle $\theta \approx 0$ rather than at the largest angle of the mesh. In the next theorem we establish that there is an important relationship between $\arg\min E_{max}$ and $\arg\min E_{cos}$ even when a well-centered triangulation exists.

Theorem 5.4. *If a well-centered triangulation of a point set exists, then that well-centered triangulation is both the Delaunay triangulation of the point set and the minmax triangulation of the point set.*

Proof. When the Delaunay triangulation is not unique (i.e., the unique Delaunay complex is not a triangulation), at least four points are cocircular. When the Delaunay complex has a cell with four or more vertices, any triangulation of the cell must contain an angle of at least $\pi/2$. This can be argued from considering triangulations of the cell formed by inserting diagonals. An *e*ar of the triangulation of the cell is a triangle for which two of the triangle edges are also edges of the Delaunay cell. We know that for any triangulation of the Delaunay cell, at least two of the resulting triangles will be ears. Moreover, we can cover the Delaunay cell with a pair of closed semidiscs in such a way that at least one semidisc completely contains an ear. For that ear, the angle along the boundary of the Delaunay cell is at least $\pi/2$. We conclude that if there is no unique Delaunay triangulation of a point set, then no Delaunay triangulation of the point set is well-centered.

Suppose, then, that a point set permits a well-centered triangulation $\mathcal{T}_0$. $\mathcal{T}_0$ is a Delaunay triangulation, because the characterization of well-centeredness in Thm. 3.1 is a stronger condition than the empty circle condition that defines a Delaunay trian-

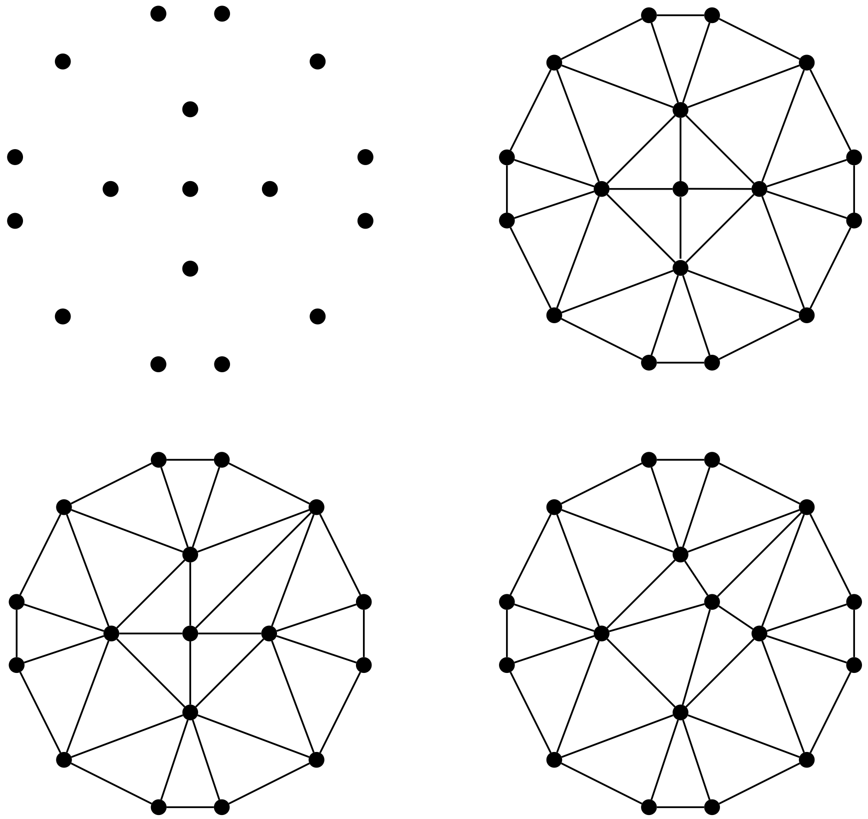

FIG. 5.1. *The minmax triangulation may produce a triangulation in which interior vertices have insufficient valence, even when there are triangulations that permit sufficient valence for all interior vertices. The sequence of figures shows a point set, the minmax triangulation of the point set, and alternate triangulation of the point set (in which all vertices have sufficient valence), and a well-centered triangulation that is obtained by moving only one vertex starting from the alternate triangulation of the point set.*

gulation; the n-dimensional equatorial balls of Thm. 3.1 are a cover for the circumcircle of an n-well-centered simplex, and in an n-well-centered mesh, the equatorial balls are empty. The Delaunay triangulation is unique in this case (by the argument of the preceding paragraph). Moreover, any other triangulation $\mathcal{T}$ of the point set has a maximum angle that is at least as large as $\pi/2$; otherwise $\mathcal{T}$ would be well-centered, and, therefore, Delaunay, contradicting the uniqueness of the Delaunay triangulation. We conclude that the minmax triangulation in this case is $\mathcal{T}_0$ and is unique. □

Combining Thm. 5.4 with Cor. 5.3 we see that $\arg\min E_{cos} = \arg\min E_{max}$ in all cases. This is a nice result, and it suggests that using the minmax triangulation is always a good idea. Unfortunately, the minmax triangulation and the Delaunay triangulation both have the undesirable property that they may have interior vertices with only four neighbors. Figure 5.1 shows a small point set for which the minmax triangulation contains an interior vertex with only four neighbors. As we have already noted, as long as we maintain the mesh connectivity, we cannot make this mesh well-centered, regardless of what function we optimize.

This is one of the main reasons we keep the mesh connectivity the same after preprocessing the mesh. We want to avoid the possibility of a lonely vertex appearing in the mesh during optimization. A reasonable alternative to the minmax triangulation might be the minmax triangulation over the set of triangulations that have no lonely vertices, but we do not know how to compute that triangulation efficiently.

There is also a question of the objective in obtaining a well-centered triangulation. If we want the result mesh to be as similar to the input as possible, a global mesh connectivity update might not be appropriate. The update could potentially change the mesh significantly. It is probably better in that case to update the mesh connectivity locally, leaving the mesh as similar as possible to the input mesh. This is the approach we have taken in our preprocessing algorithm, which, for the two-dimensional case, is not described in this paper, but is outlined in [25].

6. Experimental Results. In this section we give some experimental results of applying our energy minimization to a variety of meshes. All of the initial meshes shown here permit well-centered triangulations. In some cases the "initial mesh" shown here is actually the output of our preprocessing algorithm. For the energy minimization for planar triangulations we initially implemented the conjugate gradient method in MATLAB, using the Polak-Ribiere formula for modifying the search direction [15]. For a problem with n free variables, we reset the search direction to the negative gradient after every n iterations. We also implemented our own line search in order to have greater control than MATLAB's `fminbnd` function provides. Most of the results that follow are from the MATLAB implementation, and where the phrase *number of iterations* appears, it refers to the number of iterations of the conjugate gradient method.

Recently we implemented the energy minimization using the Mesquite library developed at Sandia National Laboratories [6]. We implemented the cost function E_p by developing a new element-based QualityMetric with a constructor accepting the argument p and summing the QualityMetric values of each element with the standard LPtoPTemplate objective function (with power 1). We continued to use the conjugate gradient method for the optimization, but we did not provide an analytical gradient to Mesquite, leaving Mesquite to numerically estimate the gradients. With the Mesquite library it was much simpler to implement E_p for tetrahedral meshes, and the results we show for three-dimensional meshes were obtained from the Mesquite-based implementation of the energy minimization. The results for our largest two-dimensional mesh were also generated using Mesquite. The optimization was terminated with a TerminationCriterion based on the number of iterations, so the phrase *number of iterations* gives some indication of processing time for the Mesquite-based implementation as well as the implementation in MATLAB.

Shading scheme: For all the meshes shown hereafter the shading indicates triangle quality with regard to well-centeredness. The shade of a triangle is based on the cosine of the largest angle of the triangle. Darker shade indicates greater largest angle and there is a noticeable jump at 90° so that well-centered triangles can be distinguished from those that are not. For example, the ten triangles that are not well-centered in the initial mesh on the left in Fig. 6.1 should be easily identifiable.

6.1. Small Meshes. The top row of Fig. 6.1 shows a test involving a small mesh of a regular pentagon and the well-centered mesh we obtained. Fourteen iterations using the energy E_4, results in the well-centered mesh shown. The final mesh on the right in top row of Fig. 6.1 has some long, thin isosceles triangles and a rather abrupt change from small triangles in the center to large triangles along the boundary. These features may be unusual compared to an intuitive idea of a nice mesh, but they are

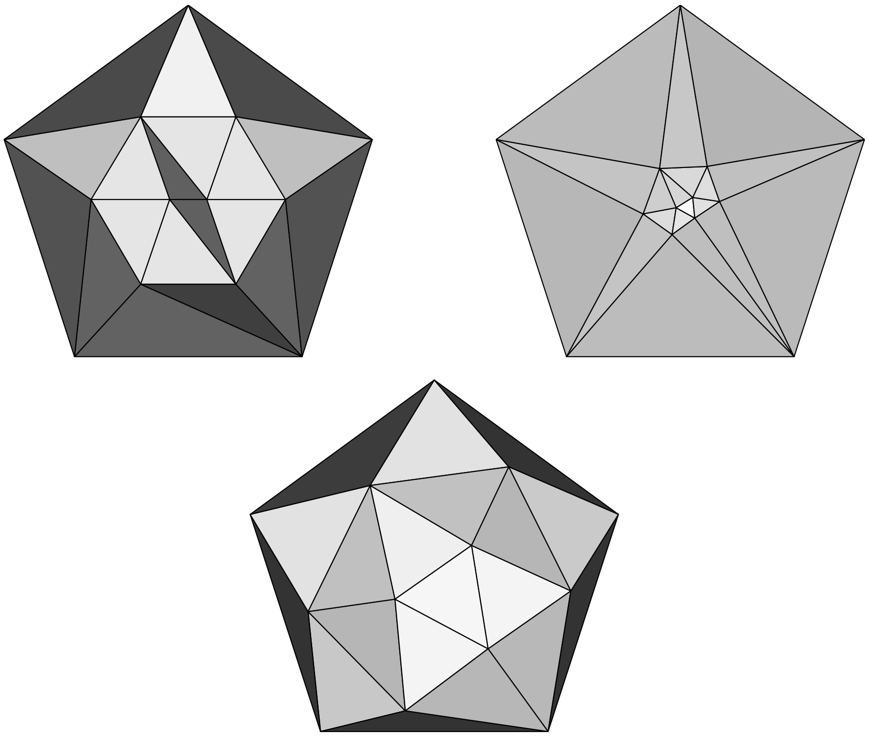

Fig. 6.1. *Regular pentagon on top left is not well-centered. On top right is the mesh obtained by our method by applying 14 steps of conjugate gradient minimization of E_4 and it is well-centered. The second row shows the result of applying variational triangle meshing to the initial mesh. We used a boundary-fixing variant of the 2D version of the variational tetrahedral meshing algorithm [2]. The resulting mesh shown is not well-centered.*

permitted in a well-centered planar triangulation, and are, in this case, essential to getting a well-centered triangulation with the given boundary vertices and topology.

In Section 2 we mentioned variational triangulations. These are based on an iterative energy minimization algorithm introduced in [2] for tetrahedral meshing. We adapted it for comparison with our method. Our adaptation keeps boundary vertices fixed, but is otherwise analogous to the algorithm given in [2]. The bottom row of Fig. 6.1 shows the result of applying variational triangle meshing to the initial mesh on top left. The result is shown after 10 iterations, which is quite near convergence. The vertices are spread out, and the triangles in the middle of the mesh are nice, but the boundary triangles are all obtuse. The energy used for variational triangle meshing does not detect a benefit of clustering the interior vertices near the center of the pentagon. We tried variational triangle meshing for several of our meshes, sometimes obtaining good results and sometimes not. We did not see any clear pattern to the cases for which it worked versus those for which it did not.

We also tried the method of Centroidal Voronoi Tessellation (CVT) [9] for several of our meshes. For our implementation of CVT, we kept boundary vertices fixed, along with any other vertices that had unbounded Voronoi cells. Vertices in the interior of the mesh with bounded Voronoi cells were moved to the centroid of their full Voronoi cells (not clipped by the domain). The algorithm worked in some cases but not in others, and we could see no good explanation for why it worked or did not work. The mesh shown in top left of Fig. 6.2 is one of the cases for which centroidal Voronoi tessellation did not work. The actual initial mesh is shown on the left in middle row of Fig. 6.2, but CVT depends on only the vertex positions and uses the Delaunay triangulation of the point set, so we show the Delaunay triangulation and the bounded Voronoi cells in Fig. 6.2. We see that this mesh is, in fact, a fixed point of our implementation of CVT while being far from well-centered. Note also that the pattern of the mesh can be extended, and it will remain a fixed point of CVT.

There are other extensible patterns that are not well-centered but are fixed points of the CVT variant that clips Voronoi cells to the domain and allows boundary vertices to move within the boundary. Our method yields a well-centered mesh with 30 iterations of energy minimization using E_4 (Fig. 6.2). That figure also shows how the energy and the maximum angle evolve. The graphs show that the method is nearing convergence at 30 iterations and that decreases in the energy E_4 do not always correspond to decreases in the maximum angle of the mesh.

6.2. Larger Meshes. These first two examples shown in Figures 6.1 and 6.2 are for small toy meshes. We have also done experiments with larger meshes. An experiment with the midsize mesh of the disc shown in Fig. 6.3 helps demonstrate that the method applies to large meshes while the mesh is small enough that its details can be visually inspected. The initial mesh is shown in the top row of Fig. 6.3. In the bottom row, the well-centered mesh obtained by 30 iterations minimizing E_4 appears. A histogram showing the distribution of the maximum angles of the triangles is included beside each mesh. We see that the energy minimization removes all the large angles of the mesh, finding a mesh with maximum angle less than 83° (actually about 82.383°), and having a large majority of triangles with maximum angle in the range [62, 76].

In Fig. 6.4 we show results for a much larger mesh, a mesh of a two-dimensional slice of the Titan IV rocket. This mesh, which is based on a mesh that the third author produced from his work for the Center for Simulation of Advanced Rockets, has 8966 triangles. The mesh is so large that it is impossible to see the details of the whole mesh at the same time, so we deviate from the usual shading scheme for this mesh. At the top of Fig. 6.4 we show an overview of the entire mesh, with the initial mesh at the very top and the result (after optimizing E_{10} for 1000 iterations) just below it. These meshes are drawn without showing element edges, because even the thinnest possible edges would entirely obscure the regions where the mesh is most refined. The interior of each element is displayed using a red color for nonacute triangles and gray for acute triangles. We follow the usual convention of making the triangles darker as the maximum angle of the triangle increases. The background color helps define the boundary of the mesh by providing contrast with the gray elements.

Below the mesh overview is a zoomed view of the top center portion of the mesh, which represents a portion of a joint slot of the titan IV rocket. The zoom for the initial mesh is shown on the left, and the zoom for the result mesh is shown on the right. In the initial mesh there are 1188 nonacute triangles ($\approx 13.25\%$ of the triangles), with a maximum angle around 155.88°. The result mesh has a maximum angle of 89.89°, and all but 141 triangles ($\approx 1.57\%$) have maximum angle below 85°. Of the 141 triangles that have angles above 85°, 14 have all three vertices on the boundary and are thus completely specified by the boundary. One example of this is in the upper left corner of the zoomed view, where there is a triangle that looks much like an isosceles right triangle. Another 60 triangles are forced to have triangles larger than 85°because they

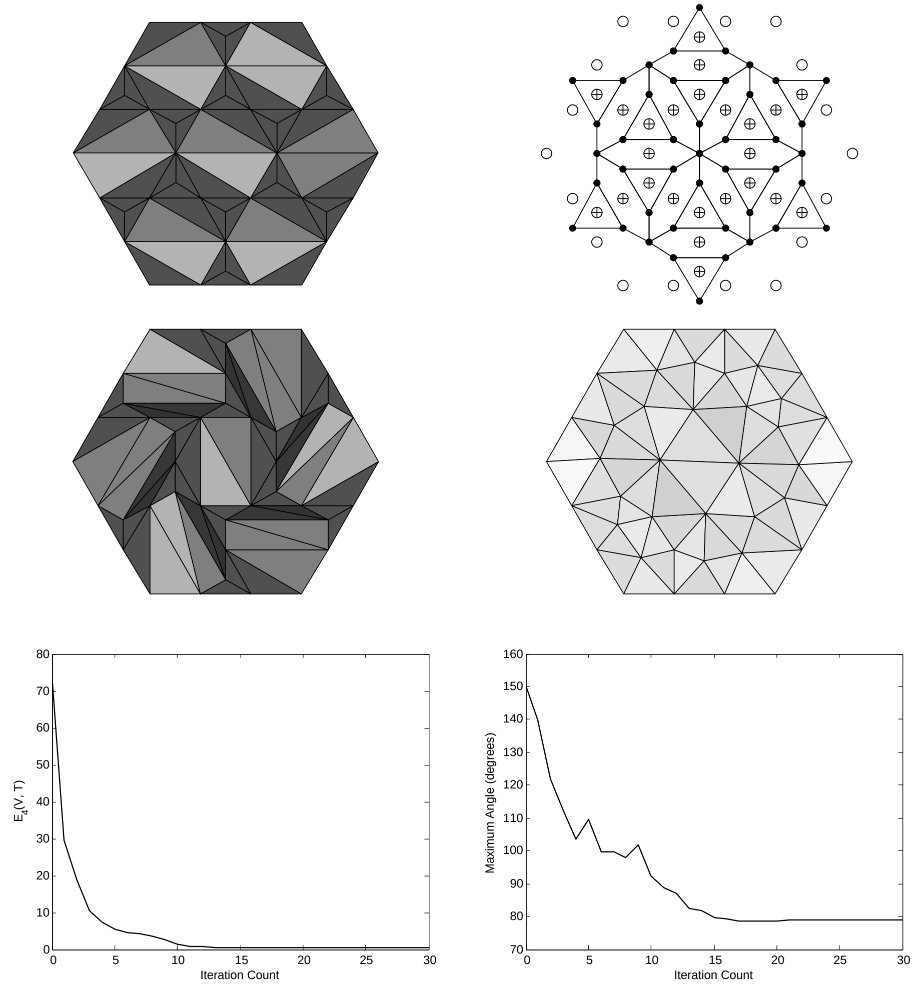

FIG. 6.2. *Mesh shown on top left is a fixed point of the Centroidal Voronoi Tessellation algorithm [9] but it is far from being well-centered. Bounded Voronoi cells are shown on right, with vertices denoted by empty circles and centroids of Voronoi cells by plus symbols. The middle row shows results of our algorithm. Starting with initial mesh in middle left, 30 iterations of our energy minimization using E_4 yields the well-centered mesh in middle right. The evolution of energy and maximum angle observed during energy minimization is shown in the bottom row.*

are part of a pair of triangles along a low curvature curved boundary. There are four such pairs along each curved boundary in the zoomed view in Fig. 6.4. In fact, all but 6 of the 141 "worst" triangles have at least one boundary vertex, and the remaining 6 triangles each have a vertex that is distance one from the boundary.

6.3. Meshes Requiring Retriangulation. Next, we show a mesh for which our energy could not find a well-centered configuration. However, when we applied our method after a retriangulation of the same set of vertices, we did obtain a well-

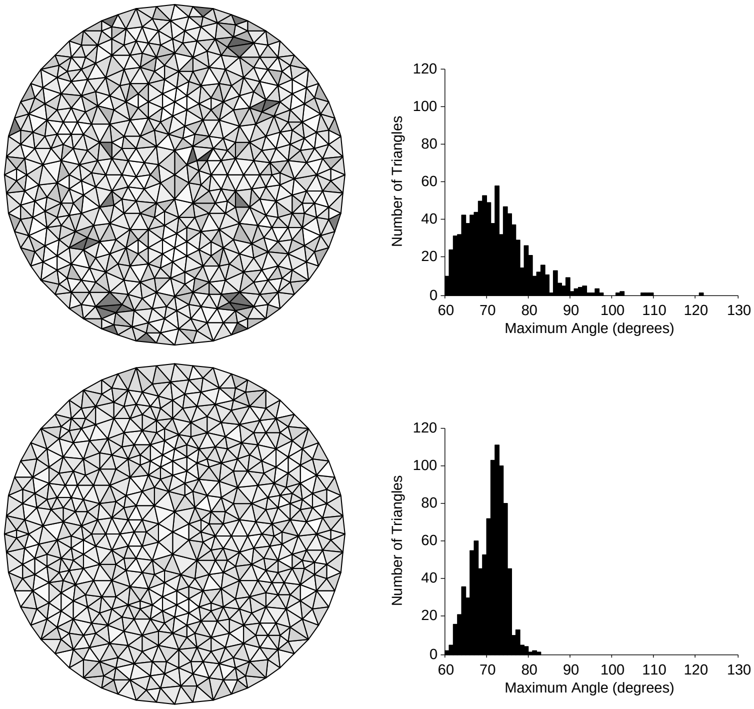

Fig. 6.3. *The initial mesh is shown in the top row. The well-centered mesh resulting from 30 iterations of E_4 minimization is shown in the bottom row. A histogram of maximum angles of the triangles is shown next to each mesh.*

centered mesh. This experiment involved a mesh called the two holes mesh, which is shown in its initial state on the top left in Fig. 6.5. The mesh has no lonely vertices, so we apply the energy minimization directly. After 500 iterations using the energy E_4, we obtain the mesh shown on top right of Fig. 6.5. We can see from the shading that the general quality is much improved, but there is a problem. In the top right corner of the result mesh there are some inverted triangles. The region where triangles have become inverted can be readily identified in middle row of Fig. 6.5, which is a zoom on the top right portion of the mesh. Inversion of triangles is rare, since it requires some angle of the mesh to reach 180°, but for the same reason, when inversion does occur, the inverted triangles tend to stay inverted.

It is possible that by using a higher power p or some linear combination of different powers p one could obtain a well-centered configuration directly from the initial two holes mesh. There are other ways to get a well-centered mesh, however, as we shall see. One way is to try a completely different mesh topology for the same vertex set. The bottom row of Fig. 6.5 shows the Delaunay triangulation of the two holes mesh (after preprocessing was applied, so the mesh is no longer Delaunay). After 500 iterations with energy E_4, the result looks quite good, but there are still some obtuse triangles, so we follow that with 500 iterations using the energy E_8, which focuses more on reducing the largest angles of the mesh and less on improving the general

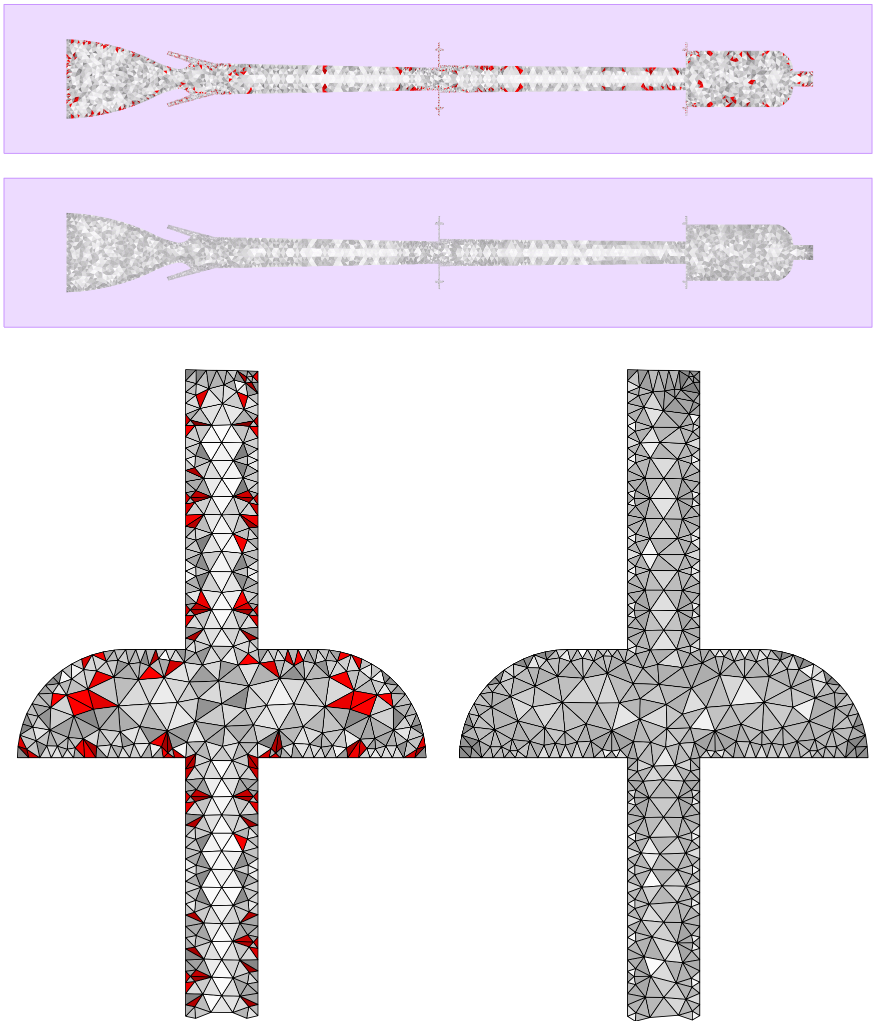

Fig. 6.4. *Results of an experiment with a mesh of a 2-dimensional slice of the Titan IV rocket. The initial mesh is displayed at the top. Below it is the result mesh, which was obtained by 1000 iterations minimizing E_{10} on the mesh. A zoomed in view on the joint slot (from the top center of the original view) shows the level of mesh refinement in the regions of higher detail. For the zoomed view, the original mesh is on the left, and the result mesh on the right. In all views, red is used for nonacute triangles, and gray indicates acute triangles. Colors get darker as the maximum angle increases.*

quality of the triangles. The well-centered result appears on bottom right in Fig. 6.5.

Another way to get a well-centered mesh of the two holes domain is to change the location of the boundary vertices. The mesh on the left in Fig. 6.6 has the same topology as the initial two holes mesh from Fig. 6.5, but the vertices along the boundary have moved. Instead of being equally spaced, the vertices on the outer boundary are more concentrated at the north and south and more spread out along the east and west. The vertices along the inner boundary have also moved slightly. For this mesh we use the energy E_6, reaching a well-centered configuration by 100 iterations. This mesh appears on the right in Fig. 6.6. The converged result with

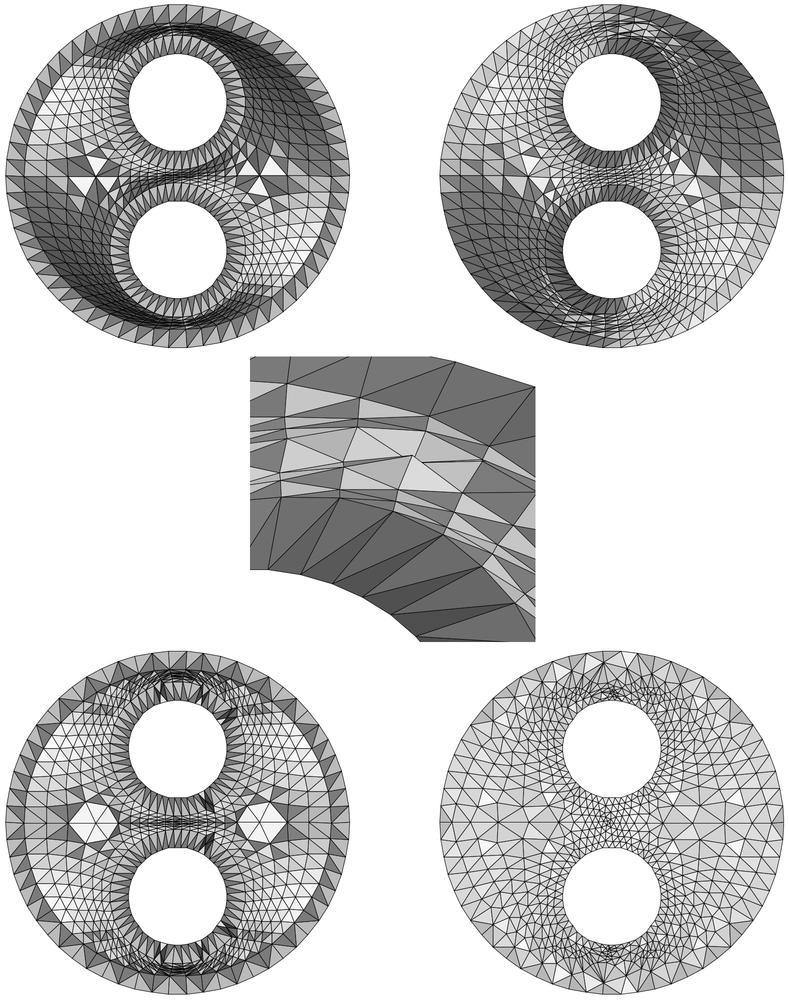

Fig. 6.5. *Energy minimization applied to the two holes mesh on top left does not yield a well-centered mesh. Result after 500 iterations of E_4 minimization is shown on top right. This resulting mesh has some inverted triangles which are shown in the close-up in middle row. With a different connectivity for the same vertex set, our minimization does produce a well-centered mesh. This is shown in bottom row. Bottom left shows a Delaunay triangulation of the original vertex set that has been preprocessed. Using this mesh as the initial mesh and applying 500 iterations of E_4 followed by 500 iterations of E_8 minimization yields the well-centered mesh shown in bottom right.*

E_6 actually has one slightly obtuse triangle (90.27°), but there are many iterations during the minimization for which the mesh is well-centered.

Working from the well-centered mesh on the right in Fig. 6.6 one can obtain a well-centered mesh that conforms to both the topology and the boundary vertices of the initial mesh in Fig. 6.5. The boundary vertices are slowly adjusted back to their original locations, taking about five steps of linear interpolation to move them back. At each step several hundred iterations with power p as high as 12 are necessary. We do not explain precise steps here, but a well-centered mesh obtained in this general way is shown in Fig. 6.7

6.4. Graded Meshes. The two holes mesh of Fig. 6.5 is graded. However, the gradation was controlled partly by the presence of the internal boundaries (of holes) and the geometry of the mesh. As a final result we show a mesh obtained by applying

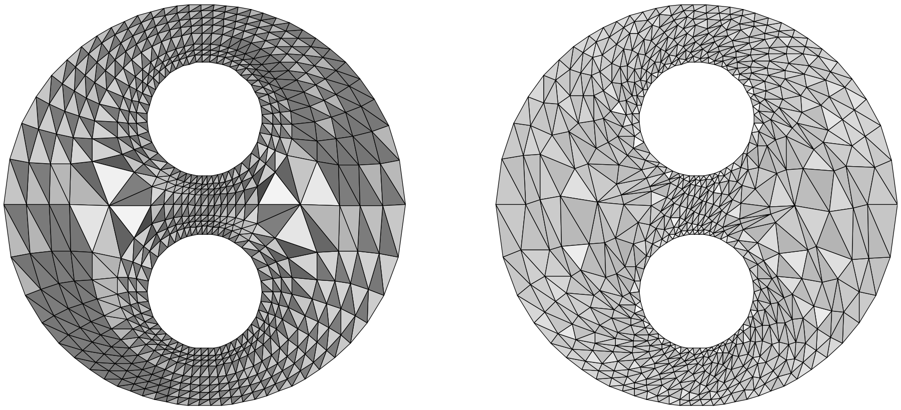

FIG. 6.6. *Two holes mesh, with a different boundary. The mesh on left has the same connectivity as the initial mesh in top left of Fig. 6.5, but the vertices along the boundary have been moved. The well-centered mesh on right was obtained with 100 iterations of E_6 minimization.*

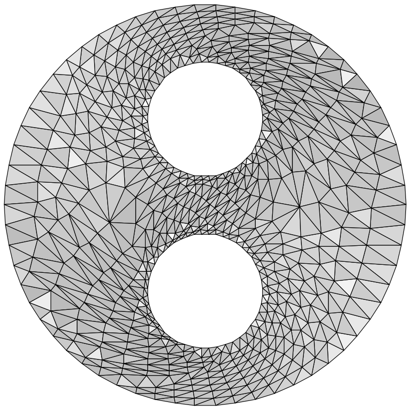

FIG. 6.7. *Well-Centered mesh of the two holes domain conforming to the mesh topology and boundary vertices of the original two holes mesh. The mesh was obtained from the mesh on the right of Fig. 6.6 by several steps of adjusting the boundary vertices back towards their original locations and finding a well-centered mesh matching the adjusted boundary vertices.*

energy minimization to a square mesh with an artificially induced gradation. The initial mesh appears at left in Fig. 6.8. The nearly converged result of 50 iterations minimizing E_4 is displayed to its right. From the other experimental results that we have shown, it is clear that the initial size of the triangles of a mesh is not always preserved well. We expect, however, that the energy will generally preserve the grading of an input mesh if the initial mesh is relatively high quality. This hypothesis stems from the observation that the energy is independent of triangle size, the idea that the topology of the mesh combined with the property of well-centeredness somehow controls the triangle size, and the supportive evidence of this particular experiment.

6.5. 3D Meshes. For tetrahedral meshes, the question of when the mesh connectivity permits a 3-well-centered mesh is more difficult than its two-dimensional analogue. In part because we do not yet have an effective preprocessing algorithm for tetrahedral meshes, most of our experiments in three dimensions have been limited

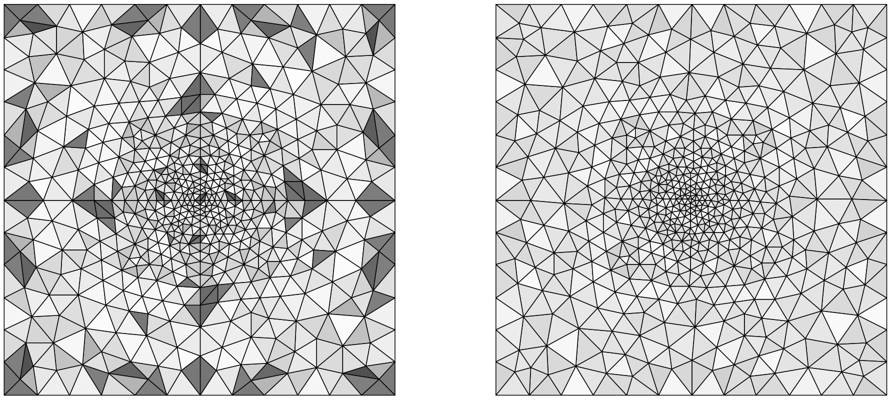

Fig. 6.8. *A Graded Mesh of the Square*

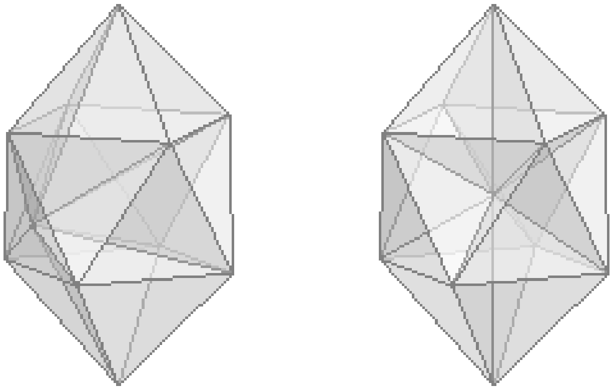

Fig. 6.9. *A simple 3-dimensional mesh with one free vertex. The initial mesh appears on the left. Optimizing E_4 produces the mesh on the right in 10 iterations.*

to very simple meshes such as the one shown in Fig. 6.9. Note that the shading in that mesh and also in the mesh shown in Fig. 6.10 has nothing to do with the quality of the tetrahedral elements of the mesh; it merely represents the shadows that would result from viewing the object under a light source. The mesh shown in Fig. 6.9 has only one free vertex, the vertex in the interior of the polyhedron. The initial mesh, which has several poor quality tetrahedra, is shown on the left of Fig. 6.9. Using the Mesquite software to optimize E_4 on the mesh produces the completely well-centered result (displayed on the right) in ten iterations.

In addition to running tests on small meshes for which we knew that a 3-well-centered mesh existed with the same mesh connectivity as the initial mesh, we did some experiments with a mesh of the cube. These experiments were partly motivated by the desire to confirm that there is a 3-well-centered mesh of the cube, and the initial mesh was carefully designed to have a nice surface mesh and a mesh connectivity for which each vertex had at least 10 adjacent edges (16 adjacent tetrahedra). The initial mesh, though nice, was not 3-well-centered. Optimizing E_{16} with the Mesquite implementation found a 3-well-centered mesh of the cube in 20 iterations. Figure 6.10 shows a cutout view of the interior of that 3-well-centered mesh of the cube.

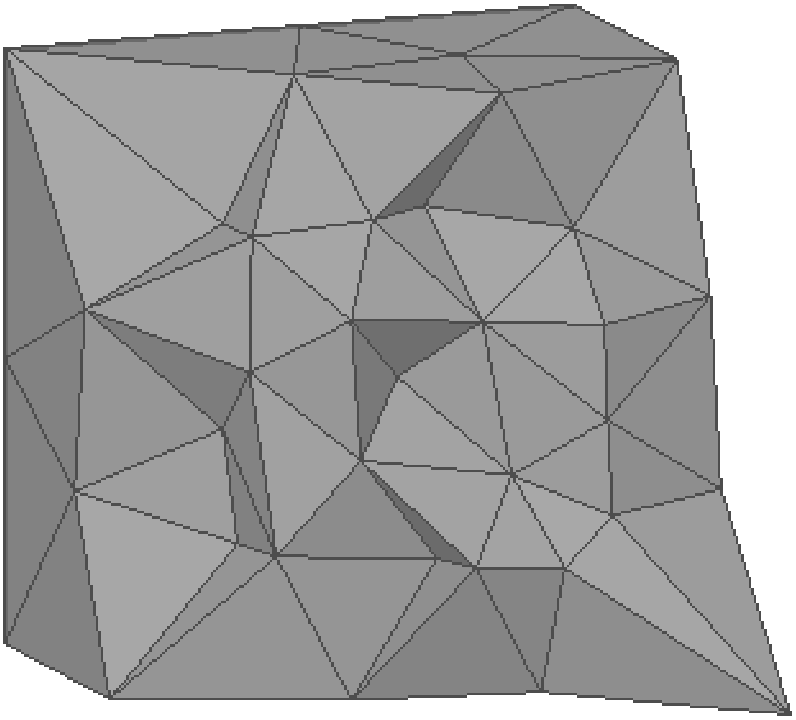

Fig. 6.10. *A cutout view of some of the tetrahedra in the interior of a 3-well-centered mesh of the cube.*

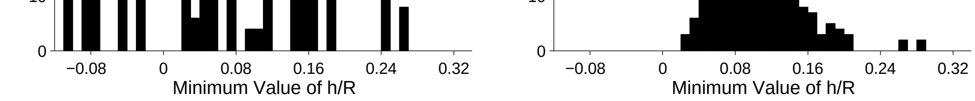

Fig. 6.11. *Histogram of minimum h/R for tetrahedra in the mesh of the cube, before (left) and after (right) minimizing E_{16} with 20 iterations of the conjugate gradient method.*

It is difficult to visually compare tetrahedral meshes, so we choose instead to compare them with a histogram of the minimal h/R values. This histogram appears in Fig. 6.11. As you study Fig. 6.11, keep in mind that having $h/R > 0$ at every vertex of every tetrahedron is the condition for 3-well-centeredness. Also recall that for the regular simplex, $h/R = 1/3$ for every vertex, and for any other tetrahedron, there is some vertex for which $h/R < 1/3$ (by Lemma 4.1).

7. Conclusions and Future Work. This paper characterizes the well-centeredness of a simplex in arbitrary dimension and introduces a related cost function that quantifies the well-centeredness of triangulations in any dimension, extending the function introduced in [25]. Some properties of the cost function are discussed, and it is shown that a cost function quantifying well-centeredness must be nonconvex. The authors intend to do some additional theoretical analysis and proofs of when and how quickly the optimization converges to a well-centered configuration. The nonconvexity of the function makes the analysis difficult, and initial investigation suggests that it will be hard to perform the analysis even in the context of a one-ring. Investigating other options for performing the optimization would be worthwhile. In particular, we hope to find good ways to localize the optimization so that it can be applied effectively in the specific locations where it is needed, and also so that the algorithm will be easy to parallelize. Collecting meaningful data about efficiency remains future work as well.

After introducing the cost function, the paper shows that the minmax angle triangulation is the optimal triangulation with respect to the cost function, but discards the minmax angle triangulation in favor of a local preprocessing algorithm introduced in [25]. It would be nice to develop a similarly local preprocessing algorithm that works in higher dimensions. Any characterization of the necessary conditions for the mesh connectivity of a tetrahedral mesh in $\mathbb{R}^3$ to permit 3-well-centeredness would be helpful in that regard. In addition, a complete characterization of the necessary and sufficient conditions on mesh connectivity in two dimensions is still lacking.

The experiments of Sec. 6 show that the proposed cost function can be effective in finding a well-centered triangulation for meshes that permit such triangulations. It is possible, however, that the cost function could be improved. For example, the authors are in the process of implementing an algorithm that minimizes E_p with a constraint that prevents element inversions. Something as simple as taking a linear combination of E_p for different powers of p also might improve the cost function. A systematic study of the effects of this algorithm on the aspect ratio of elements would be worthwhile as well. Although the finite element method is not a primary motivation for the work on well-centered meshes, it is possible that the method could be used effectively to improve finite element meshes in some cases.

Acknowledgment. We would like to thank Vadim Zharnitsky for useful discussions.

References.

[1] R. Abraham, J. E. Marsden, and T. Ratiu, *Manifolds, Tensor Analysis, and Applications*, Springer–Verlag, New York, second ed., 1988.

[2] Pierre Alliez, David Cohen-Steiner, Mariette Yvinec, and Mathieu Desbrun, *Variational tetrahedral meshing*, ACM Transactions on Graphics, 24 (2005), pp. 617–625.

[3] M. Bern, S. Mitchell, and J. Ruppert, *Linear-size nonobtuse triangulation of polygons*, in Proceedings of the tenth annual ACM Symposium on Computational Geometry, New York, 1994, ACM Press, pp. 221–230.

[4] M. W. Bern, D. Eppstein, and J. Gilbert, *Provably good mesh generation*, J. Computer and Systems Sciences, 48 (1994), pp. 384–409. Special issue for 31st FOCS.

[5] Alexander I. Bobenko and Boris A. Springborn, *Variational principles for circle patterns and Koebe's theorem*, Trans. Amer. Math. Soc., 356 (2004), pp. 659–689.

[6] Michael Brewer, Lori Freitag Diachin, Patrick M. Knupp, Thomas Leurent, and Darryl J. Melander, *The mesquite mesh quality improvement toolkit*, in Proceedings of the 12th International Meshing Roundtable, Santa Fe, New Mexico, 2003, Sandia National Laboratories, pp. 239–250.

[7] Charles R. Collins and Kenneth Stephenson, *A circle packing algorithm*, Computational Geometry: Theory and Applications, 25 (2003), pp. 233–256.

[8] Mathieu Desbrun, Anil N. Hirani, Melvin Leok, and Jerrold E. Marsden, *Discrete exterior calculus*, arXiv:math.DG/0508341, (2005).

[9] Q. Du, V. Faber, and M. Gunzburger, *Centroidal Voronoi tessellations: applications and algorithms*, SIAM Review, 41 (1999), pp. 637–676.

[10] H. Edelsbrunner, *Geometry and Topology for Mesh Generation*, Cambridge University Press, 2001.

[11] H. Edelsbrunner, T.S. Tan, and R. Waupotitsch, *An $O(n^2 \log n)$ time algorithm for the minmax angle triangulation.*, SIAM Journal on Scientific and Statistical Computing, 13 (1992), pp. 994–1008.

[12] D. Eppstein, J. M. Sullivan, and A. Üngör, *Tiling space and slabs with acute tetrahedra*, Comput. Geom. Theory Appl., 27 (2004), pp. 237–255.

[13] Jeff Erickson, Damrong Guoy, John Sullivan, and Alper Üngör, *Building spacetime meshes over arbitrary spatial domains*, in Proceedings of the 11th International Meshing Roundtable, Sandia National Laboratories, 2002, pp. 391–402.

[14] Hale Erten and Alper Üngör, *Computing acute and non-obtuse triangulations*, in Proceedings of the 19th Annual Canadian Conference on Computational Geometry, CCCG 2007, 2007, pp. 205–208.

[15] Michael T. Heath, *Scientific Computing: An Introductory Survey*, McGraw-Hill, second ed., 2002.

[16] Anil N. Hirani, *Discrete Exterior Calculus*, PhD thesis, California Institute of Technology, May 2003.

[17] Anil N. Hirani, Kalyana B. Nakshatrala, and Jehanzeb H. Chaudhry, *Numerical method for Darcy flow derived using Discrete Exterior Calculus*, Tech. Report UIUCDCS-R-2008-2937, Department of Computer Science, University of Illinois at Urbana-Champaign, February 2008.

[18] Murray S. Klamkin and George A. Tsintsifas, *The circumradius-inradius inequality for a simplex*, Mathematics Magazine, 52 (1979), pp. 20–22.

[19] H. Maehara, *Acute triangulations of polygons*, European Journal of Combinatorics, 23 (2002), pp. 45–55.

[20] Elefterios A. Melissaratos and Diane L. Souvaine, *Coping with inconsistencies: A new approach to produce quality triangulations of polygonal domains with holes*, in SCG '92: Proceedings of the Eighth Annual Symposium on Computational Geometry, New York, NY, USA, 1992, ACM Press, pp. 202–211.

[21] R. A. Nicolaides, *Direct discretization of planar div-curl problems*, SIAM J. Numer. Anal., 29 (1992), pp. 32–56.

[22] Jim Ruppert, *A Delaunay refinement algorithm for quality 2-dimensional mesh generation*, J. Algorithms, 18 (1995), pp. 548–585.

[23] Igor Sazonov, Oubay Hassan, Kenneth Morgan, and Nigel P. Weatherill, *Smooth Delaunay–Voronoi dual meshes for co-volume integration schemes*, in Proceedings of the 15th International Meshing Roundtable, Birmingham, Alabama, September 17–20 2006, Sandia National Laboratories.

[24] Alper Üngör and Alla Sheffer, *Pitching tents in space-time: Mesh gen-*

eration for discontinuous Galerkin method, International Journal of Foundations of Computer Science, 13 (2002), pp. 201–221.

[25] Evan Vanderzee, Anil N. Hirani, Damrong Guoy, and Edgar Ramos, *Well-centered planar triangulation – an iterative approach*, in Proceedings of the 16th International Meshing Roundtable, Seattle, Washington, October 14–17 2007, Sandia National Laboratories.

[26] Liping Yuan, *Acute triangulations of polygons*, Discrete and Computational Geometry, 34 (2005), pp. 697–706.