# Optimal Point Placement for Mesh Smoothing

Nina Amenta [1] Marshall Bern [2] David Eppstein [3]

**Abstract**

We study the problem of moving a vertex in a finite element mesh to optimize the shapes of adjacent triangles. We show that many such problems can be solved in linear time using generalized linear programming. We also give efficient algorithms for some mesh smoothing problems that do not fit into the generalized linear programming paradigm.

## 1 Introduction

Unstructured mesh generation, a key step in the finite element method, can be divided into two stages. In *point placement*, the input domain is augmented by Steiner points and a preliminary mesh is formed, typically by Delaunay triangulation. In *mesh improvement*, local optimizations are performed, involving the movement of Steiner points and rearrangement of the triangulation.

Computational geometry has made some inroads into point placement, and methods including Delaunay refinement, quadtrees, and circle packing are now known to generate meshes with guaranteed quality; for surveys of these results, see [8, 9]. There has been less theoretical progress, however, in mesh improvement, which has remained largely the domain of practitioners.

Mesh improvement typically combines several kinds of local optimization:

- *Refinement and derefinement* split and merge triangles, changing the number of Steiner points.
- *Topological changes* such as *flipping* replace sets of elements by other such sets, while preserving the positions of the Steiner points.
- *Mesh smoothing* moves the Steiner points of the mesh while preserving its overall topology.

In this paper we study mesh smoothing algorithms. Our focus is not to determine the best smoothing method, which is more properly a subject for experiment or numerical analysis; rather we show that a wide variety of methods can be performed efficiently.

A commonly used technique, *Laplacian smoothing*, sweeps over the mesh, successively moving each point to the centroid of its neighbors. This technique lacks motivation because it is not directly connected to any specific mesh quality criterion; moreover, the result may not even remain a valid triangulation. But in practice Laplacian smoothing spaces points evenly and gives two-dimensional meshes of reasonable quality. In three dimensions, however, even spacing does not guarantee good element quality. A *sliver* tetrahedron has evenly spaced vertices, but very sharp angles. (See [7] for a classification of tetrahedra in terms of solid and dihedral angles.) Laplacian smoothing sometimes removes slivers, but in large meshes it often leaves clusters of slivers [20].

Freitag, Jones, and Plassmann [19] proposed an alternative to Laplacian smoothing. Rather than using the centroid, their optimization-based method computes for each Steiner point a new placement that maximizes the minimum angle in adjacent triangles. Freitag et al. use an iterative steepest-descent algorithm to solve this optimal placement problem. Empirically this algorithm finds the optimum location in an average of 2.5 steps, but Freitag et al. do not prove their algorithm correct.

The same optimal placement problem was independently considered by Matoušek et al. [28] as an instance of the paradigm called generalized linear programming.

[1] Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA, 94304; http://www.geom.umn.edu/∼nina/; amenta@parc.xerox.com.

[2] Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA, 94304; bern@parc.xerox.com. Work performed in part while visiting Carnegie-Mellon University.

[3] Department of Information and Computer Science, University of California, Irvine, CA 92697-3425; http://www.ics.uci.edu/∼eppstein/; eppstein@ics.uci.edu. Work supported in part by NSF grant CCR-9258355 and by matching funds from Xerox Corp, and performed in part while visiting Xerox PARC.

Matoušek et al. show how to solve this problem using an algorithm related to the dual simplex method. (In retrospect, the steepest-descent algorithm of Freitag et al. can be seen as a primal simplex method, but its correctness is not directly justified by the work of Matoušek et al.; correctness follows from our analysis below.)

Minimum angle, however, is not the only measure of mesh quality. Various papers have provided theoretical justification for other measures including maximum angle [4], maximum edge length [32], minimum height [23], minimum containing circle [12], and—most recently—ratio of area to sum of squared edge lengths [6]. Data-dependent criteria [6, 16, 31] may be used in adaptive meshing, which uses the finite element method's output to improve the mesh for another run.

In this paper, we study optimization-based smoothing using quality criteria such as those mentioned above. We show that, as in the case of minimum angle, many of these criteria give rise to *quasiconvex programs* and can be solved by linear-time dual simplex methods or steepest-descent primal simplex methods. Because of the generality of these methods, they can also solve mixed-criterion optimization problems.

We generalize the theory to three- and higher dimensions, where effective smoothing methods are a more critical need and asymptotic time complexity is more important. We show that again quasiconvex programming often arises; for instance it can maximize the minimum solid angle. We believe optimization-based three-dimensional mesh smoothing should outperform Laplacian smoothing in practice. Indeed, in very recent experimental work Freitag and Ollivier-Gooch [20] have shown that optimization-based smoothing for minimum dihedral angle outperforms Laplacian smoothing, both alone and in conjunction with flipping.

Finally, we show that although several other optimal point placement problems do not form quasiconvex programs, we can solve them efficiently by other means. This direction may also be relevant in practice; Freitag and Ollivier-Gooch recommend smoothing for the sine of the dihedral, a non-quasiconvex quality measure.

## 2 Generalized Linear Programming

Many problems in computational geometry, such as separating points by a hyperplane, can be modeled directly as low dimensional linear programs. Many other problems, such as the circumcircle of a point set, are not linear programs, but the same techniques often apply to them. To explain this phenomenon, various authors have formulated a theory of *generalized linear programming* [3, 22, 28].

A *generalized linear program* (GLP, also known as an *LP-type problem*) consists of a finite set $S$ of *constraints* and an *objective function* $f$ mapping subsets of $S$ to some totally ordered space and satisfying the following properties:

1. For any $A \subset B$, $f(A) \leq f(B)$.
2. For any $A$, $p$, and $q$, if $f(A) = f(A \cup \{p\}) = f(A \cup \{q\})$, then $f(A) = f(A \cup \{p, q\})$.[4]

Typically the constraints are subsets of a space $X$ and $f(A)$ is the minimum over the intersection of the constraints in $A$ of an objective function defined over $X$. The problem is to compute $f(S)$ using only evaluations of $f$ on small subsets of $S$.

A *basis* of a GLP is a set $B$ such that for any $A \subsetneq B$, $f(A) < f(B)$. The *dimension* $d$ of a GLP is the maximum cardinality of a basis. A number of efficient GLP algorithms are known [1, 3, 10, 15, 22, 28], the best running time of which is $O(dnT + f(d)E \log n)$ where $n$ is the number of constraints, $T$ measures the time to test a proposed solution against a constraint (typically this is $O(d)$), $f$ is exponential or subexponential, and $E$ is the time to perform a single basis evaluation. Indeed, these algorithms are straightforward to implement and have small constant factors, so they should be practical even for the modest values of $n$ relevant in our problems. (The number of constraints should range roughly from 10 to 100 in the planar problems, depending on how complicated a criterion one chooses to optimize and on the degree of the initial mesh, and may possibly reach several hundred in the three-dimensional problems.)

Our GLPs have the following form, which we call "quasiconvex programming". We wish to optimize some quantity that is the pointwise maximum (or minimum) of a finite set of functions. Such a problem will be

[4] Property 2 is often expressed in the more complicated form that, if $A \subset B$ and $f(A) = f(B)$, then, for any $p$, $f(A) = f(A \cup \{p\})$ iff $f(B) = f(B \cup \{p\})$. A simple induction shows this to be equivalent to our formulation.

a low-dimensional GLP so long as the level sets of the functions are all convex. Note that this does not necessarily imply that the functions themselves are convex; in convex analysis, such functions are called *quasiconvex*.

More formally, define a *nested convex family* to be a map $\kappa(t)$ from $\mathbb{R}$ to convex sets in $\mathbb{R}^d$ such that if $a < b$ then $\kappa(a) \subset \kappa(b)$. Any nested convex family $\kappa$ determines a function $f_\kappa(x) = \inf\{\, t \mid x \in \kappa(t)\,\}$ the level sets of which are the closures of $\kappa(t)$. If $f_\kappa$ is continuous, and does not take a constant value on any open set, we say that $\kappa$ is continuous.

If $S = \{\kappa_1, \kappa_2, \ldots \kappa_n\}$ is a set of nested convex families, and $A \subset S$, let

$$f(A) = \inf\left\{ (t, x) \;\Big|\; x \in \bigcap_{\kappa_i \in A} \kappa_i(t) \right\}$$

where the infimum is taken in the lexicographic ordering, first by $t$ and then by the coordinates of $x$.[5] We use this same lexicographic ordering to compare the values of $f$ on different subsets of $S$.

LEMMA 2.1. *The set $S$ and objective function $f$ described above form a GLP of dimension at most $2d+1$. If each $\kappa_i$ in $S$ is either constant or continuous, the dimension is at most $d+1$.*

*Proof.* . Property 1 of GLPs is obvious. Property 2 follows from the observation that $(t, x) = f(A) = f(A \cup \{\kappa_j\})$ iff $x \in \kappa_j(t')$ for every $t' > t$.

For any $t' < t$, $\bigcap \kappa_i(t') = \emptyset$ so by Helly's theorem some $(d+1)$-tuple of sets has empty intersection. Since there are only finitely many $(d+1)$-tuples, we can choose a tuple $B^-$ that has an empty intersection for all $t' < t$. Then $f(B^-) = (t, x')$ for some $x'$, so the presence of $B^-$ forces the GLP solution to have the correct value of $t$.

If $\bigcap \kappa_i(t) \neq \emptyset$, then $x$ is the minimal point in that intersection, and is determined by some $d$-tuple $B^+$ of the $\kappa_i$. Otherwise, for any $t' > t$ there is some $d$-tuple determining the minimal point in $\bigcap \kappa_i(t')$; let $B^+$ be such a set determining the minimal point for values of $t'$ arbitrarily close to $t$. Then $f(B^- \cup B^+) = f(S)$, so some basis of $S$ is a subset of $B^- \cup B^+$ and has cardinality at most $2d+1$.

Finally, suppose each $\kappa_i$ in $S$ is constant or continuous. Let $F = \bigcap \mathrm{cl}(\kappa_i(t))$ (where cl denotes the topological closure); then $F$ is nonempty because it contains $x$, and $\mathrm{int}(F) = \emptyset$ by continuity. If some $\kappa_i(t)$ contains a point of the subspace spanned by $F$ in its interior, we say that $\kappa_i$ is "bad"; its boundary intersects $F$ in a subset of measure zero, so we can find a value $x'$ in $F$ that is not on the boundary of any bad $\kappa_i$. The shape of the neighborhood of $x'$ in $F$ is determined only by the boundaries of good $\kappa_i$, and the intersection of the closures of these good $\kappa_i$ must span the same subspace as $F$. Form the projection $\pi : \mathbb{R}^d \mapsto \mathbb{R}^{d-\dim F}$ perpendicular to $F$. Then $\bigcap \pi(\mathrm{cl}(\kappa_i(t)))$ (for good $\kappa_i$) is the single point $\pi(F)$. At least one good $\kappa_j$ is continuous (rather than constant) and the intersection of $\pi(\mathrm{int}(\kappa_j(t)))$ with $\bigcap \pi(\mathrm{cl}(\kappa_i(t)))$ is empty. By Helly's theorem, we can find a $(d - \dim F + 1)$-tuple $B^-$ of these convex sets having empty intersection. By continuity, for each each continous $\kappa_i$, and each $t' < t$, $\pi(\kappa_i(t')) \subset \pi(\mathrm{int}(\kappa_i(t)))$, so the intersection of the $\kappa_i(t')$ in $B^-$ is empty, even in projection, and the presence of $B^-$ forces the GLP solution to have the correct value of $t$. Similarly, we can reduce the size of the set $B^+$ determining the correct value of $x$ from $d$ to $\dim F$, so the total size of a basis is at most $(d - \dim F + 1) + \dim F = d + 1$. □

The first part of this lemma is similar to [3, Theorem 8.1]. Note that we only used the assumption of convexity to prove the dimension bound; similar nested families of non-convex sets still produce GLP problems, but could have arbitrarily large dimension.

By Lemma 2.1 we can solve quasiconvex programs using GLP algorithms. We can also perform a more direct local optimization procedure to find $(t, x)$. For any $x \in \mathbb{R}^d$, define

$$F(x) = \inf\left\{ t \;\Big|\; x \in \bigcap_{\kappa_i \in S} \kappa_i(t) \right\}.$$

Then the level sets of $F$ themselves form a nested convex family, so we can find $f(S)$ by applying steepest descent, nested binary search, or other local optimization techniques to find the point minimizing $F$. Thus we can justify the correctness of the local optimization mesh smoothing procedure used by Freitag et al. In practice, it may be appropriate to combine this idea with the dual simplex methods coming from GLP theory by

[5] In all our applications, the intersection above is bounded, and empty for sufficiently small $t$, so this infimum is well defined.

using steepest descent to perform the basis exchange operations needed in GLP algorithms.

## 3 Quasiconvex Mesh Smoothing in $\mathbb{R}^2$

Let $q(\Delta)$ measure the quality of a triangulation element $\Delta$ (a triangle, or perhaps a portion of a triangle such as one of its angles). We are given a triangulation, and wish to move one of its Steiner points in such a way as to minimize $\max q(\Delta_i)$, where the maximization occurs over elements incident to the moving point.

In this section we describe ways of formulating such problems as quasiconvex programs. The basic idea is to construct for each $\Delta_i$ a nested convex family $\kappa_i(t) = \{ x \mid q(\Delta_i(x)) \leq t \}$, where $\Delta_i(x)$ indicates the triangle formed by moving the Steiner point to position $x$. In other words, if we are given a bound $t$ on the triangulation quality, $\kappa_i(t)$ is the *feasible region* in which placement of the Steiner point will allow $\Delta_i$ to meet the quality bound. Finding the optimal Steiner point placement is equivalent to finding the optimal quality bound that allows a feasible placement.

The families $\kappa_i(t)$ are clearly nested, but convexity will need to be proven using the detailed properties of the quality measure $q$. One can then find the optimal placement $x$ by solving the quasiconvex program associated with this collection of nested convex families. To make sure that the result is a valid triangulation, we add additional halfspace constraints to our collection, forming constant nested families, to force $x$ into the kernel of the star-shaped polygon formed by removing the Steiner point from the triangulation (Figure 1(a)).

It remains to show convexity of the feasible regions $\kappa_i(t)$ for various quality measures. In the remainder of this section, we describe these measures and their corresponding feasible regions. As shown in Figure 2, many different criteria have identical feasible regions; however they do not necessarily lead to the same Steiner point placement as the parametrization of the nested families differs.

**Area.** The feasible regions for maximizing minimum triangle area are strips parallel to the fixed (external) sides of the triangles. In the presence of the halfspace constraints forcing the Steiner point into the kernel of its polygon, we can simplify these strips to halfspaces. The intersection of one such halfspace and the corresponding kernel constraint is shown in Figure 2(a). One can also balance neighboring element sizes by maximizing minimum area, using a halfspace with the same boundary but opposite orientation (Figure 2(b)).

**Altitude.** The *external altitude* of $\Delta_i$ (the altitude having the fixed side of $\Delta_i$ as its base) can be minimized or maximized using halfspace feasible regions identical to those for area (Figure 2(a,b)). The feasible regions in which the other two altitudes are at least $h$ are the intersections of pairs of halfspaces through one fixed point, passing at distance $h$ from the other point; one such halfspace is shown in Figure 2(d) and the other is its vertical reflection. The feasible regions for minimizing the maximum internal altitude are not convex.

**Angle.** As noted by Matoušek et al. [28], one can maximize the minimum angle by using constraints of two types. For the internal angles at the Steiner points, the region in which the angle is at least $\theta$ forms either the union or intersection of two congruent circles (as $\theta$ is acute or obtuse respectively) having the fixed side of $\Delta_i$ as a chord. In the former case this may not be convex, but in the presence of the kernel constraints we can simplify the feasible region to circles (Figure 2(e)). The regions in which the external angles are at least $\theta$ form wedges bounded by rays through a fixed vertex of $\Delta_i$, which can again be simplified in the presence of the kernel constraints to halfspaces (Figure 2(d)). It is also natural to minimize the maximum angle; unfortunately the feasible regions for the internal angles are non-convex (complements of circles). However one can still minimize the maximum angle at external vertices, using halfspace regions (Figures 2(c)).

**Edge length.** The feasible region for minimizing the length of the internal edges of $\Delta_i$ is an intersection of two circles of the given radius, centered on the fixed vertices of $\Delta_i$ (Figure 2(h)). We can use the same two-circle constraints (with larger radii than depicted in the figure) to minimize the maximum element diameter.

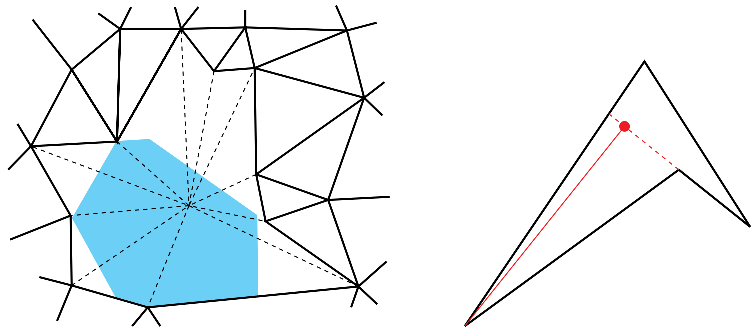

Figure 1. Steiner point may move within kernel of star-shaped region formed by its removal; for size-based criteria such as length the optimal placement may be on the kernel boundary.

**Aspect ratio.** The *aspect ratio* of a triangle is the ratio of its longest side length to its shortest altitude. We consider separately the ratios of the three sides to their corresponding altitudes; the maximum of these three will give the overall aspect ratio. The ratio of external sides to altitude has a feasible region (after taking into account the kernel constraints) forming a halfspace parallel to the external side, like that in Figure 2(b). To determine the aspect ratio on one of the other two sides of a triangle $\Delta_i$, normalize the triangle coordinates so that the moving point has coordinates $(x, y)$ and the other two have coordinates $(0, 0)$ and $(1, 0)$. The side length is then $\sqrt{x^2 + y^2}$, and the altitude is $y/\sqrt{x^2 + y^2}$, so the overall aspect ratio has the simple formula $(x^2+y^2)/y$. The locus of points for which this is a constant $b$ is given by $x^2+y^2 = by$, or equivalently $x^2 + (y - (b/2))^2 = (b/2)^2$. Thus the feasible region is a circle tangent to the fixed side of $\Delta_i$ at one of its two endpoints (Figure 2(f)).

**Perimeter.** The feasible region for minimizing the maximum perimeter is an ellipse (Figure 2(g)).

**Circumradius and containing circle.** The feasible regions for circumradius are nonconvex lunes bounded by pairs of circles. However, minimizing the maximum *containing circle* (the smallest circle containing the given triangle, without necessarily having the vertices on its boundary) produces convex feasible regions, formed by using the same region as the circumcircle within a vertical slab perpendicular to the fixed segment of the triangle, and a lune similar to that for edge length or diameter outside the slab. These regions' boundaries are three circular arcs, meeting at common tangents, with the radius of the middle arc equal to half that of the arcs on either side (Figure 2(i)).

**Inradius.** The feasible region for maximizing the minimum inradius of any triangle can be shown by some algebraic manipulation (with the two fixed points normalized to $(0, 0)$ and $(1, 0)$) to be given by the inequality

$$\begin{aligned}&-8r^3x + 8r^3x^2 + 4r^2y - 4r^4y + 4r^2xy\\&- 4r^2x^2y - 4ry^2 + 8r^3y^2 + y^3 - 4r^2y^3\\&\quad \geq 0.\end{aligned}$$

Since this is quadratic in $x$, its solutions form an interval in any horizontal line. Some manipulation (performed by *Mathematica* and omitted here) shows that the lower and higher branches of $x$, computed as functions of $y$ and the parameter $r$, have positive and negative second derivatives respectively for values of $y$ and $r$ for which inradius $r$ is possible; therefore this region is convex.

**Area over squared edge length.** Bank and Smith [6] define yet another measure of the quality of a triangle, computed by dividing the triangle's area by the sum of the squares of its edge lengths. This gives a dimensionless quantity which Bank and Smith normalize to be one for the equilateral triangle (and less than one for any other triangle). They then use this quality measure as the basis for a local improvement method for mesh smoothing. However,

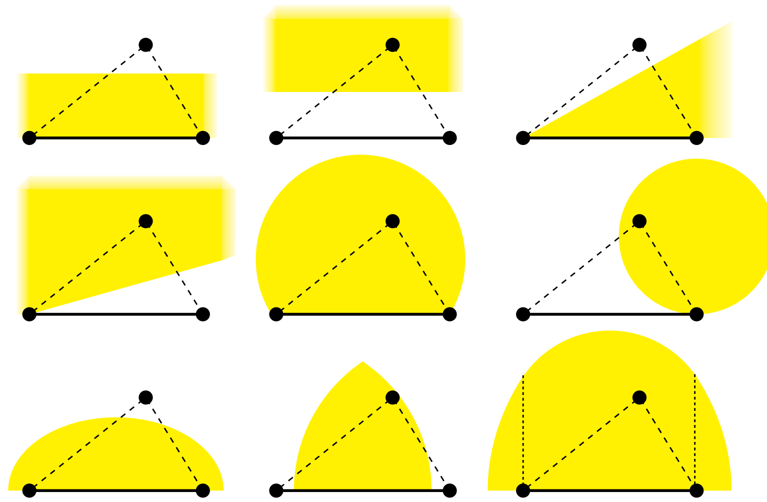

Figure 2. Feasible regions for planar mesh smoothing quality criteria: (a) minimizing maximum area or external altitude; (b) maximizing minimum area, external altitude, or external aspect ratio; (c) minimizing maximum external angle; (d) maximizing minimum external angle, or maximizing minimum internal altitude; (e) maximizing minimum internal angle; (f) maximizing internal aspect ratio; (g) minimizing maximum perimeter; (h) minimizing maximum edge length (a similar but larger lune occurs when minimizing diameter); (i) minimizing containing circle.

as Bank and Smith show, the feasible region for this measure forms a circle centered on the perpendicular bisector of the two fixed points, so our quasiconvex programming methods can be applied to find the optimum point placement.

**Mixtures of criteria.** We have described the various optimization criteria above as if only one is to be used in the actual mesh smoothing algorithm. But clearly, the same formulation applies to problems in which we combine various criteria, for instance some measuring element shape and others measuring element size, with the overall quality of an element equal to the weighted maximum of these criteria. Indeed, this idea can alleviate a problem with size-based criteria such as edge length, perimeter, area, etc.: if one optimizes such a criterion on its own, the optimal point placement may lie on the boundary of the kernel, giving rise to a degenerate triangulation (Figure 1(b)). If one combines these criteria with shape-based criteria such as angles or aspect ratio, this complication cannot occur. To solve such a mixed problem, we simply include constraints for each different criterion in the combination.

THEOREM 3.1. *The Steiner point placement optimizing the criteria described above, or a weighted combination of criteria, can be computed in linear time by quasiconvex programming.*

## 4 Mesh Smoothing in Higher Dimensions

Many of the two-dimensional quality criteria discussed above have higher-dimensional generalizations that also have convex feasible regions.

**Volume and altitude.** Just as the area of a triangle with a fixed base is proportional to its height, the volume of a simplex with a fixed base is proportional to its altitude. The triangulation minimizing the maximum volume, or maximizing the minimum volume, can be found using feasible regions in the form of halfspaces parallel to the fixed face of the simplex. The same type of feasible region can be used to optimize the altitude at the moving Steiner point. The feasible regions for maximizing the minimum of the other altitudes are the intersections of pairs of halfspaces through $d-1$ of the fixed points.

**Boundary measure.** The measure of any boundary face of a simplex is proportional to the distance of the moving Steiner point from the affine hull of the remaining fixed points on the facet, so one can minimize the maximum face measure using "generalized cylinders" formed by taking a cartesian product of a sphere with this affine hull. In particular the Steiner point placement minimizing the maximum edge length can be found by using spherical feasible regions centered on each fixed point, and in $\mathbb{R}^3$ the placement minimizing the maximum triangle area can be found using cylindrical feasible regions centered on each fixed edge. These face measures are convex functions, so their sums are also convex, implying that the level sets for total surface area of all triangles in a tetrahedron, or total length of all edges in a tetrahedron, again form convex feasible regions.

**Containing sphere.** As in $\mathbb{R}^2$, the feasible regions for the minimum containing sphere are bounded by $2^d - 1$ algebraic patches, in which the containing sphere has some fixed set of vertices on its boundary. These patches meet the plane of the fixed vertices perpendicularly, and are locally convex (they are figures of rotation of lower dimensional feasible regions, except for the one corresponding to the region in which the containing sphere equals the circumsphere, which is a portion of that sphere). In $\mathbb{R}^3$, these patches are portions of spheres and tori. Further, they meet at a continuous boundary (since the containing sphere radius is a continuous function of the moving point's location) and are continuously differentiable where they meet (at each point where two patches meet, they share tangent planes with the containing sphere itself). Thus these patches combine to form a convex region.

**Dihedrals.** The dihedral angles of a simplex are formed where two faces meet along an *axis* determined by some $d - 1$ points. If these axis points are all fixed, one of the two faces is itself fixed, and the feasible region is a halfspace forming the given angle with this fixed face. However, if the axis includes the moving point, the feasible regions are in general non-convex.

**Solid Angles.** As we show in the next section, the feasible regions for maximizing the minimum solid angle (measured at the fixed points of each tetrahedron, for three-dimensional problems, or at the moving point in any dimension) are convex.

THEOREM 4.1. *In any constant dimension, the Steiner point placement optimizing each of the criteria described above except exterior solid angle, or a weighted combination of criteria, can be computed in linear time by quasiconvex programming. The exterior solid angles as well can be optimized in three dimensions.*

## 5 Feasible Regions for Solid Angles

We now prove that the feasible regions for maximizing the minimum solid angles of the mesh elements are convex, for the angles at the moving point, in any dimension, and for the angles at fixed points of tetrahedra in $\mathbb{R}^3$ only. Convexity of the feasible regions for solid angles at fixed points in higher dimensions remains open.

We start with the simpler case, in which we are interested in the solid angle at one of the fixed vertices of a tetrahedron in $\mathbb{R}^3$. This angle can be measured by projecting the other three vertices onto a unit sphere centered on the fixed vertex, and measuring the area of the spherical triangle formed by these three projected points. If the three projected points are represented by three-dimensional unit vectors $a$, $b$, and $c$ (with $a$ representing the moving point and $b$, $c$, and the origin representing the three fixed points) then the solid angle $E$ at the origin satisfies the equation

$$\tan(E/2) = \frac{a \cdot (b \times c)}{1 + b \cdot c + c \cdot a + a \cdot b}$$

[18]. Therefore, the boundary of the feasible region (on the unit sphere) is given by an equation of the form

$$a \cdot (b \times c) = k(1 + b \cdot c + c \cdot a + a \cdot b),$$

which is linear in $a$ and therefore forms a circle on the unit sphere. In terms of the original unprojected points, the feasible region is therefore a convex circular cone.

To prove that the feasible regions for the interior solid angles are also convex, we use some facts from convex analysis [13]. A function $f(v)$ from some convex subset of a vector space $V$ to $\mathbb{R}$ is said to be *convex* if, for any $x, y \in V$, and any $0 \le t \le 1$,

$$f(t \cdot x + (1 - t) \cdot y) \le t \cdot f(x) + (1 - t) \cdot f(y).$$

A function $f(v)$ is said to be *quasiconcave* if its level sets $\{v \mid f(v) \geq k\}$ are convex. A function is $s$-concave if $f(v)^s$ is convex; in the cases of interest to us $s$ will always be negative. If $f$ is quasiconcave we also say that it is $(-\infty)$-concave (and if $f$ is logconcave, i.e. if $\log f$ is convex, we also say that it is 0-concave). We allow functions to take infinite values; these can be interpreted by the squeamish as shorthand for appropriate limits.

LEMMA 5.1. *Let $f(u)$ be $s$-concave for $s \leq 0$, and let $g(v)$ be the characteristic function of a convex set $\kappa$. Then the cartesian product $h(u, v) = f(u)g(v)$ is $s$-concave.*

We omit the straightforward proof. The next result appears as [13, Theorem 3.21].

LEMMA 5.2. *Let $f$ be $s$-concave on an open convex set $C$ in $\mathbb{R}^{m+n}$. Let $C^*$ be the projection of $C$ on $\mathbb{R}^m$ and for $x \in C^*$, let $C(x)$ be the $x$-section of $C$. Define*

$$f^*(x) = \int_{C(x)} f(x, y)\, dy, \quad x \in C^*.$$

*If $-1/n \leq s \leq \infty$, then $f^*$ is $s^*$-concave on $C^*$, where $s^* = s/(1+ns)$ with the usual conventions when $s = -1/n$ or $s = \infty$.*

COROLLARY 5.1. *Let $f : U \mapsto \mathbb{R}$ be $-1/k$-concave, and let $g : V \mapsto \{0, 1\}$ be the characteristic function of a convex set $\kappa$ in a $k$-dimensional subspace $V$ of $U$. Then the convolution of $f$ and $g$ is quasiconcave.*

*Proof.* Let $h(u, v)$ be the cartesian product of $f(u)$ and $g(v)$ as in Lemma 5.1. Then the convolution can be computed as $h^*(u - v)$. The result follows from Lemmas 5.1 and 5.2. □

A special case of Corollary 5.1, in which $k$ equals the dimension $d$ of the domain of $f$, appears (with a different proof) as [13, Theorem 3.24]. For our application, we are interested in a different case, in which $k = d - 1$. The solid angle of a $d$-simplex in $d$-dimensional space, measured at the moving point, can be interpreted as the fraction of the field of view at that moving point taken up by the convex hull $\kappa$ of the remaining fixed points. This fraction can be computed as the convolution of the characteristic function of $\kappa$ with a function $f(v)$ measuring the fraction of field of view taken by an infinitesimally small surface patch of $\kappa$. This function $f(v)$ is inversely proportional to the square ($d-1$ power, for general $d$) of the distance from $v$ to the patch, and directly proportional to the sine of the incidence angle of $v$ onto the patch. If we translate this patch to the origin, $f$ has the simple form $(v \cdot e)/|v|^d$ where $e$ is a vector normal to the patch.

LEMMA 5.3. *The function $f(v) = (v \cdot e)/|v|^d$, defined on the open halfspace $v \cdot e > 0$, is $-1/(d-1)$-concave.*

*Proof.* Because of the rotational symmetry of $f$, we need only prove this for the two-dimensional function $f(x, y) = y/(x^2 + y^2)^{d/2}$ in the halfplane $y > 0$. We used *Mathematica* to compute the principal determinants of the Hessian of $f^s$. These are

$$\frac{\partial^2}{\partial y^2} f(x, y)^{-1/(d-1)} = \frac{d\, x^2 y^{1/(d-1)} (x^2 + y^2)^{d/(2d-2)} (x^2 + (d-1)y^2)}{(d-1)^2 y^2 (x^2 + y^2)^2}$$

which is always positive (for $y > 0$, $d > 1$), and

$$\left( \frac{\partial^2}{\partial x^2} \frac{\partial^2}{\partial y^2} - \frac{\partial^2}{\partial x \partial y} \frac{\partial^2}{\partial y \partial x} \right) f(x, y)^{-1/(d-1)} = 0.$$

Since both principal determinants are non-negative, the function is convex. □

THEOREM 5.1. *The feasible region for the solid angle at the moving point of a simplex is convex.*

*Proof.* As described above, we can express the solid angle as the convolution of $f(v)$ with the characteristic function of the convex hull of the fixed points. By Lemma 5.3, $f$ is $-1/(d-1)$-concave within a halfspace defined by the kernel constraints. Therefore we can use Corollary 5.1 to show that the solid angle is quasiconcave and therefore has convex level sets. □

Our proof for the interior solid angles generalizes to any dimension, but that for the exterior solid angles does not. There seems to be some correspondence between the feasible regions of interior solid angles in dimension $d$, and the feasible regions of exterior solid angles in dimension $d + 1$; perhaps this correspondence can be exploited to show that the exterior solid angle feasible regions are convex in higher dimensions as well.

## 6 Non-quasiconvex Mesh Smoothing

We have seen that many mesh smoothing criteria give rise to quasiconvex programming problems; however, other criteria, including minmax angle, minmax circumradius, and maxmin perimeter, do not have convex feasible regions.

Perhaps this can be seen as evidence that these measures are less appropriate for mesh smoothing applications, since it means among other things that there may be many local optima instead of one global optimum. Indeed, it seems likely that the height and perimeter criteria mentioned above do not lead to good element shapes. However there is evidence that the maximum angle is an appropriate quality measure for finite element meshes [4], so we now discuss methods for optimizing this measure. Our results should be seen as preliminary and unready for practical implementation.

THEOREM 6.1. *We can find the placement of a Steiner point in a star-shaped polygon, minimizing the maximum angle, in time $O(n \log^c n)$ for some constant $c$.*

*Proof.* Each feasible region in which some particular angle is at most $\theta$ forms a halfplane or circle. The lifting transformation $(x, y) \mapsto (x, y, x^2 + y^2)$ maps these regions to halfspaces in $\mathbb{R}^3$; $\theta$ is feasible if the intersection of all these halfspaces meets the paraboloid $z = x^2 + y^2$. The result follows by applying parametric search [29] to a parallel algorithm that constructs the intersection [2, 24] and tests whether any of its features crosses the paraboloid. □

We can of course combine the maximum angle with the many other criteria, including circumradius, for which the feasible regions are bounded by lines and circles.

An alternate approach suggests itself, which may have a better chance of leading to a practical algorithm. Define a generalized Voronoi diagram the cells of which determine which mesh angle would be worst if the Steiner point were placed in the cell. Are the cells of this diagram connected? If so it seems likely that generalized Voronoi diagram algorithms [25, 26, 30] can construct this diagram in time $O(n \log n)$ or perhaps even $O(n)$. We could then find the optimal placement by examining the features of this diagram.

Finally, we consider one last criterion, minimum total edge length. This does not fit into our quasiconvex programming framework, since the overall quality is a sum of terms from each element rather than a minimum or maximum of such terms; however the corresponding optimal triangulation problem remains a topic of considerable theoretical interest [14, 27]. A mesh improvement phase might also help reduce the (large) constant factors in known approximate minimum weight Steiner triangulation algorithms [17]. Without the kernel constraints enforcing that the result is a valid triangulation, the problem of placing one Steiner point to minimize the total distance to all other points is a facility location problem known as the *Weber* or *Fermat-Weber problem*. Although it has no good exact solution (the solution point is a high degree polynomial in the inputs [5, 11]) one can easily solve it approximately by steepest descent [33]. The kernel constraints do not change the overall nature of this solution. Thus this version of the mesh smoothing problem can again be solved efficiently.

## 7 Conclusions

We have described a general framework for theoretical analysis of mesh smoothing problems, and have shown how to perform optimal Steiner point placement efficiently for many important quality measures. Although there remain some open problems (for instance, to what extent our results generalize to quadrilateral and hexahedral meshes) the most important directions for future research are empirical: which of the criteria we have described leads to the best quality meshes, and to what extent can theoretical generalized linear programming algorithms serve as practical methods for the solution of these problems?

## References

[1] I. Adler and R. Shamir. A randomization scheme for speeding up algorithms for linear and convex quadratic programming problems with a high constraints-to-variables ratio. *Math. Prog.* 61, 1993, pp. pp. 39–52.

[2] N. Amato, M. T. Goodrich, and E. A. Ramos. Parallel algorithms for higher dimensional convex hulls. *35th IEEE Symp. Foundations of Comp. Sci.*, 1994, pp. 683–694; http://www.cs.tamu.edu/research/robotics/Amato/Papers/focs94.300.ps.gz.

[3] N. Amenta. Helly-type theorems and generalized linear programming. *Disc. Comp. Geom.* 12, 1994, pp. 241–261; http://www.geom.umn.edu/∼nina/papers/dcg.ps.

[4] I. Babuška and A. Aziz. On the angle condition in the finite element method. *SIAM J. Num. Anal.* 13, 1976, pp. 214–227.

[5] C. Bajaj. The algebraic degree of geometric optimization problems. *Disc. Comp. Geom.* 3, 1988, pp. 177–191.

[6] R. E. Bank and R. K. Smith. Mesh smoothing using a posteriori error estimates. SIAM J. Num. Anal., to appear; ftp://math.ucsd.edu/pub/scicomp/reb/ftpfiles/a67.ps.Z.

[7] M. Bern, L. P. Chew, D. Eppstein, and J. Ruppert. Dihedral bounds for mesh generation in high dimensions. *6th ACM/SIAM Symp. Discrete Algorithms*, 1995, pp. 189–196; http://www.ics.uci.edu/∼eppstein/pubs/p-dihedral.ps.Z. See also D. Eppstein, Tetrahedra classified by bad angles, http://www.ics.uci.edu/∼eppstein/junkyard/tetraqual.html.

[8] M. Bern and D. Eppstein. Mesh generation and optimal triangulation. In *Computing in Euclidean Geometry*, 2nd Ed., World Scientific, 1995, pp. 47–123.

[9] M. Bern and P. E. Plassmann. Mesh generation. Manuscript, 1996.

[10] K. Clarkson. A Las Vegas algorithm for linear programming when the dimension is small. *29th IEEE Symp. Foundations of Computer Science*, 1988, pp. 452–456; *J. ACM* 42, 1995, pp. 488–499; http://cm.bell-labs.com/who/clarkson/lp2.html.

[11] E. J. Cockayne and Z. A. Melzak. Euclidean constructability in graph minimization problems. *Math. Mag.* 42, 1969, pp. 206–208.

[12] E. F. D'Azevedo and R. B. Simpson. On optimal interpolation triangle incidences. *SIAM J. Sci. Stat. Comput.* 10, 1989, pp. 1063–1075.

[13] S. Dharmadhikari and K. Joag-Dev. Unimodality, Convexity and Applications. Academic Press, 1988.

[14] M. T. Dickerson and M. H. Montague. A (usually?) connected subgraph of the minimum weight triangulation. *12th ACM Symp. Comp. Geom.*, 1996, pp. 204–213; http://www.middlebury.edu/∼dickerso/mwtskel.html.

[15] M. E. Dyer and A. M. Frieze. A randomized algorithm for fixed-dimensional linear programming. *Math. Prog.* 44, 1989, pp. 203–212.

[16] N. Dyn, D. Levin, and S. Rippa. Data dependent triangulations for piecewise linear interpolation. *IMA J. Num. Anal.* 10, 1990, pp. 137–154.

[17] D. Eppstein. Approximating the minimum weight Steiner triangulation. *Disc. Comp. Geom.* 11, 1994, pp. 163–191; http://www.ics.uci.edu/∼eppstein/pubs/p-mwst.html.

[18] F. Eriksson. On the measure of solid angles. *Math. Mag.* 63, no. 3, 1990, pp. 184–187.

[19] L. A. Freitag, M. T. Jones, and P. E. Plassmann. An efficient parallel algorithm for mesh smoothing. *4th Int. Meshing Roundtable*, Sandia Labs., 1995, pp. 47–58; ftp://fea1.ansys.com/pub/sowen/freitag.epsi.gz.

[20] L. A. Freitag and C. Ollivier-Gooch. A comparison of tetrahedral mesh improvement techniques. Manuscript, 1996.

[21] L. A. Freitag, C. Ollivier-Gooch, M. T. Jones, and P. E. Plassmann. Scalable unstructured mesh computation, http://www.mcs.anl.gov/home/freitag/SC94demo/.

[22] B. Gärtner. A subexponential algorithm for abstract optimization problems. *SIAM J. Comput.* 24, 1995, pp. 1018–1035; http://www.inf.fu-berlin.de/inst/pubs/tr-b-93-05.abstract.html.

[23] C. Gold, T. Charters, and J. Ramsden. Automated contour mapping using triangular element data structures and an interpolant over each irregular triangular domain. *Proc. SIGGRAPH*, 1977, pp. 170–175.

[24] M. T. Goodrich. Geometric partitioning made easier, even in parallel. *9th ACM Symp. Comp. Geom.*, 1993, pp. 73–82.

[25] R. Klein and A. Lingas. Hamiltonian abstract Voronoi diagrams in linear time. *5th Int. Symp. Algorithms and Computation*, Springer LNCS 834, 1995, pp. 11–19; *10th Eur. Worksh. Comp. Geom.*, 1994, pp. 1–4; http://www.dna.lth.se/Research/Algorithms/Papers/andrzej4.ps.

[26] R. Klein, K. Mehlhorn, and S. Meiser. Randomized incremental construction of abstract Voronoi diagrams. *Comp. Geom. Th. & Appl.* 3, 1993, pp. 157–184.

[27] C. Levcopoulos and D. Krznaric. Quasi-greedy triangulations approximating the minimum weight triangulation. *7th ACM/SIAM Symp. Discrete Algorithms*, 1996, pp. 392–401.

[28] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. Tech. Report B 92-17, Freie Univ. Berlin, Fachb. Mathematik, Aug. 1992.

[29] N. Megiddo. Applying parallel computation algorithms in the design of sequential algorithms. *J. ACM* 30, 1983, pp. 852–865.

[30] K. Mehlhorn, S. Meiser, and C. O'Dunlaing. On the construction of abstract Voronoi diagrams. *Disc. Comp. Geom.* 6, 1991, pp. 211–224.

[31] S. Rippa and B. Schiff. Minimum energy triangulations for elliptic problems. *Comp. Meth. in Appl. Mech. and Eng.* 84, 1990, pp. 257–274.

[32] G. Strang and G. J. Fix. An analysis of the finite element method. Prentice-Hall, 1973.

[33] E. Weiszfeld. Sur le point pour lequel la somme des distances de $n$ points donnes est minimum. *Tohoku Math. J.* 43, 1937, pp. 355–386.