DOI:10.1111/cgf.14066
Eurographics Symposium on Geometry Processing 2020
Q. Huang and A. Jacobson
(Guest Editors)

Volume 39 (2020), Number 5

Approximating isosurfaces by guaranteed-quality triangular meshes

J. Hass[121] and M. Trnkova[341]

[1]Department of Mathematics, University of California, Davis, CA USA

Abstract

We describe a new method for approximating an implicit surface F by a piecewise-flat triangulated surface whose triangles are as close as possible to equilateral. The main advantage is improved mesh quality which is guaranteed for smooth surfaces. The GradNormal algorithm generates a triangular mesh that gives a piecewise-differentiable approximation of F, with angles between 35.2 and 101.5 degrees. As the mesh size approaches 0, the mesh converges to F through surfaces that are isotopic to F.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computational Geometry and Object Modeling]: Curve, surface, solid, and object representations—

1. Introduction

We study the problem of approximating a surface in R^3 by a mesh that has optimal angle properties. In computations based on mesh descriptions of a surface, it is often essential to avoid “slivers’, or triangles with angles close to zero. A random process for selecting vertices on a surface gives a triangulation with expected minimum angle approaching zero as the number of points increases [BEY91], implying that slivers are hard to avoid when creating meshes from points sampled on a surface.

Badly shaped triangles can cause mesh-based algorithms to break down for numerical reasons. Avoiding poor quality triangles is important for a wide variety of applications, including computer graphics, shape comparison, finite elements, finding numerical solutions of PDEs, and geometric modeling. In one important area of applications, a function $f : R^3 \to R$ measures the absorption at each point of an X-ray or imaging machine, or the density of a solid object, where a level set $F = f^{-1}(c)$ represents the surface of a scanned object, such as an organ, bone, brain cortex or protein. For purposes of visualization, geometric processing, surface comparison, surface classification, or modeling of properties of the surface, it is desirable to have a high quality mesh representing the surface.

We note that it is possible to construct a sequence of surfaces converging to F pointwise and consisting entirely of flat equilateral triangles, but with the tangent planes of the triangles not converging to those of the surface. The search for a regular mesh whose triangles approximate the tangent spaces of a given surface leads to two conflicting goals. One goal is to make the triangles as close to equilateral as possible, and the second is to have the mesh conform differentiably to the surface, so that its tangent planes approximate those of the surface.

In this paper we introduce the *GradNormal Algorithm*, which produces a mesh whose tangent planes converge to those of a differentiable implicit surface $F \subset R^3$, and whose triangles have angles in the interval $[35.2^\circ, 101.5^\circ]$. These angle bounds are the best rigorously established. We prove that the GradNormal meshes are 2-dimensional manifolds that converge to F as the mesh size approaches zero, and that the convergence is piecewise-smooth, as explained later.

Example meshes produced by GradNormal for implicit surfaces, defined as level sets of explicitly given mathematical functions, are shown in Figure 1 and Figure 2. These are obtained by tiling the

[1] Partially supported by NSF grants DMS1760485 and DMS1719582
[2] The authors thank Benjamin Hass for assistance with the software implementation.
[3] Partially supported by NSF grants DMS-1107452, 1107263, 1107367 “RNMS: Geometric structures and Representation Varieties" (the GEAR Network)

unit cube with 869,652 tetrahedra and displayed using MeshLab [CCC*08]. See Table 1 for data on how these angles improve as the tiling becomes finer.

GradNormal meshes can also be produced by inputting nonsmooth surfaces, described as a union of polygonal faces. This allows us to use GradNormal to improve a mesh with poor angle quality. A mesh produced from the Stanford Bunny [TL94, Bun] is shown in Figure 3, obtained using GradNormal with a tiling of the cube with 6,748,416 tetrahedra. A signed-distance-function can be obtained from the original Stanford Bunny Mesh, after filling in holes. This defines an implicit surface, computed using TriMesh [DH20]. The code used to produce these images is available on GitLab [HT19].

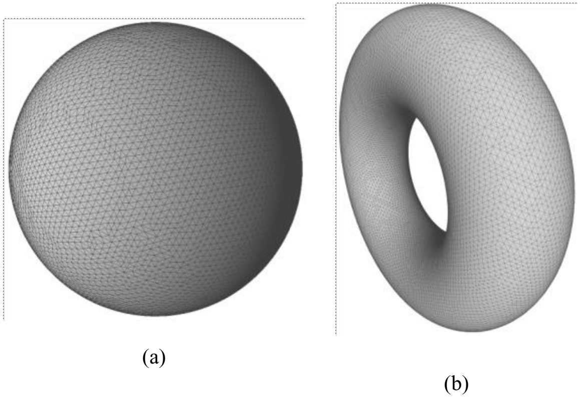


Figure 1: *The mesh of the sphere has 25,092 triangular faces with angles in the interval* [35.4°,102.7°]. *The torus has 35,838 triangular faces with angles in the interval* $[32.8^\circ, 104.7^\circ]$. *At sufficiently fine scales all angles lie in the interval* [35.2°,101.5°].

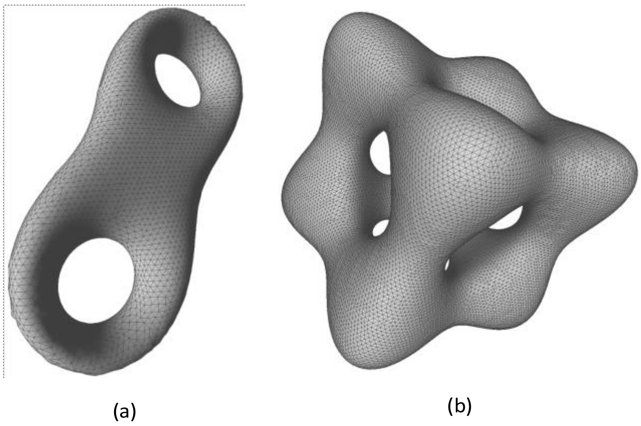
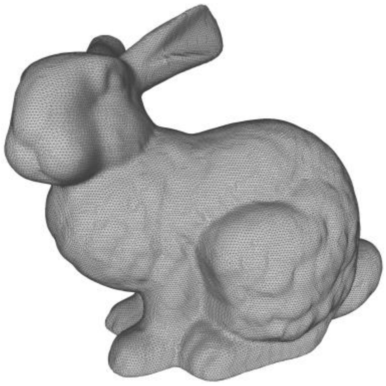


Figure 2: *These meshes have (a) 10,346 (genus 2) and (b) 55,122 (genus 5) triangular faces. They have angles in the intervals* [22.1°,129.0°] *(genus 2) and* $[29.2^\circ, 113.4^\circ]$ *(genus 5).*

In Figure 4 we see the range of mesh angles produced by the GradNormal algorithm at two resolutions and the angles in the original Stanford Bunny mesh. Note that the original mesh has holes and isolated vertices. The holes were filled in prior to remeshing with the GradNormal algorithm. When applied to a non-smooth surface that has sharp corners at vertices and folds along edges, the angle bounds that GradNormal guarantees for smooth surfaces do not apply, even in the limit. Nonetheless Figure 4 shows that the angles are more clustered around 60^o than in the original mesh. To compare the distribution of angles in one chart, the counts have been normalized by dividing by three times the number of triangular faces, or 348,162 for the GradNormal mesh with 6,748,416 tetrahedra and 1,400,310 for the GradNormal mesh with 52,931,340 tetrahedra. The two GradNormal meshes have very similar distributions, with most angles between 40^o and 80^o, while the original Stanford mesh had many angles near 30^o and 90^o.



Small angles may be created in the mesh produced by GradNormal when it is applied to a non-smooth or piecewise-flat surface, such as the Bunny mesh. These can occur near points where adjacent faces in the non-smooth surface meet with sharp dihedral angles.

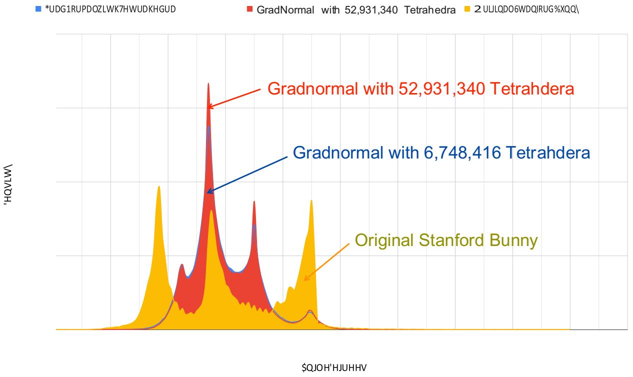


Figure 3: *The original Stanford Bunny Mesh (with several holes and isolated points) gave angles in the interval* [0.49°,177.6°]. *The mesh of the Stanford Bunny shown here, produced by GradNormal using a 6,748,416 tetrahedra tiling of the cube, gives angles in the interval* $[10.4^\circ, 150.4^\circ]$.

Figure 4: *Angle distributions for the original Stanford Bunny mesh and for GradNormal meshes of the Stanford Bunny at two resolutions.*

The GradNormal algorithm proceeds in three steps. The first step produces a mesh with acute triangles that we call a *MidNormal* mesh. While having very good angle properties, and giving a 2dimensional manifold that lies close to the implicit surface, its normal vectors do not align with those of F. The second step involves a projection of the MidNormal mesh vertices to the implicit surface, so that normal vectors align with those of the surface. This is followed by a single remeshing operation involving vertices of valence four. A careful analysis of the distortion of angles under this step establishes the properties claimed for the GradNormal mesh.

The first step is similar to the Marching Tetrahedra algorithm, but with important differences. The underlying idea behind both algorithms appears in the theory of *normal surfaces*, a powerful tool used to study surfaces in 3-dimensional manifolds that goes back to work of Kneser in 1929 [Kne29]. The GradNormal algorithm begins by tiling space with tetrahedra of a fixed shape. It uses a tetrahedral tiling chosen to optimize the angles appearing in the final mesh. Intersecting a surface F with these tetrahedra tiles generates an approximation of F by triangles and quadrilaterals, as in Figure 5.

A normal surface with respect to a 3-dimensional triangulation τ is an embedded surface $S \subset M$ whose intersection with any tetrahedron in τ has the simplest possible form, cutting across each tetrahedron in the same way as a flat plane. The surface intersects a single tetrahedron in one of two types of *elementary disk*. An elementary disk is either a single flat triangle or two flat triangles meeting along a common edge and forming a quadrilateral. The vertices of each triangle of an elementary disk are located at the midpoints of different edges of the tetrahedron, as in Figure 5. The MidNormal mesh consists of such elementary disks.

Given an implicit surface F, the GradNormal algorithm first produces a MidNormal mesh. It then projects the mesh vertices to the closest point on the surface F, and finishes with a single remeshing step that removes valence four vertices.

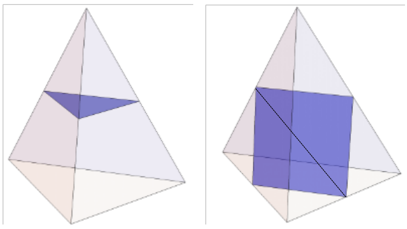


Figure 5: *Elementary disks forming part of a normal surface. A triangle separates one vertex from the other three. A quadrilateral that separates pairs of vertices is split into two triangles by adding a diagonal. There are four possible types of triangle and three possible types of quadrilaterals in each tetrahedron.*

The quadrilaterals of a normal surface are divided along a diagonal to produce a mesh. The mesh consists of flat triangles that separate the vertices of a tetrahedron in the same way as F. The MidNormal mesh locates the vertices of the triangles on the midpoints of the tetrahedron edges. In contrast Marching Tetrahedra interpolates these vertex positions along the edges of the tetrahedron. This interpolation results in angles that can be arbitrarily close to zero [NH91], so that Marching Tetrahedra is often combined with further algorithms that improve mesh quality [CDS12].

The tiling of R^3 that we use is obtained by optimizing angles among a family of tetrahedra discovered by Goldberg [Gol74]. A particular Goldberg tiling is determined, up to isometry, by a pair of positive constants a,e that determine a tetrahedron $\tau_{a,e}$, as in Figure 6. Isometric copies of $\tau_{a,e}$ fill R^3 with no gaps, with pairs of tetrahedra matching along faces. In our setting e is a scale parameter that determines the size of the tetrahedron, and we fix√ a to have the value $a = e$ 2/4. This choice gives the optimal angles for our method. It turns out that this choice of tetrahedral shape coincides with the tetrahedra in a tiling described by Sommerville [Som23]. A straightforward computation then shows that the MidNormal mesh has angles in the interval [45°,90°].

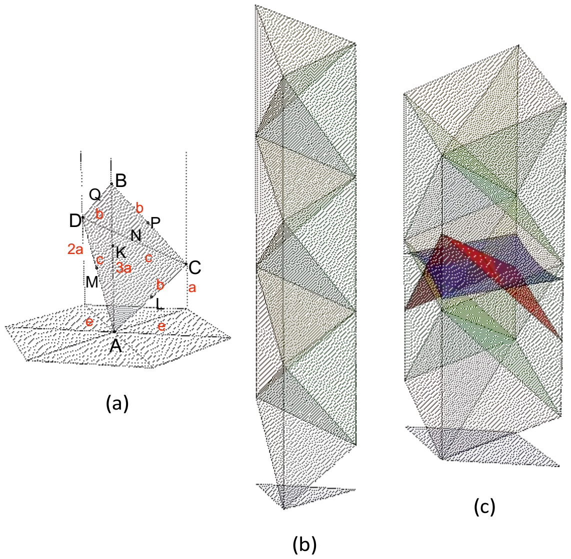


Figure 6: *(a) A tetrahedron $\tau_{a,e}$, one of a family that tiles R^3. The scale independent parameter $a \in (0,\infty)$ determines the shape. (b) These tetrahedra stack to tile a vertical column over an equilateral triangle of length e (center). (c) A surface in R^3 divides the vertices of these tetrahedra, leading to a triangular mesh. Part of a normal surface is also shown.*

The MidNormal mesh gives a continuous approximation to the surface F. A differentiable approximation is achieved in the second part of the GradNormal Algorithm. It starts with a MidNormal mesh and then projects each of the vertices to the closest point on the surface F. We will see that all angles are then as claimed, with the exception of a collection of valence-four vertices. These can have arbitrarily small angles, and are removed in a remeshing step which deletes four triangles adjacent to a common valence four vertex and triangulates the resulting quadrilateral by adding a diagonal. Properties of the resulting mesh $M^1(f,e)$ are given in the following theorem, proven in Section 4.

Theorem 1.1 Let $F = f^{-1}(0) \in R^3$ be a compact level surface of a smooth function f and let $M^1(f,e)$ be the mesh produced by the GradNormal algorithm with tetrahedra of scale e. Then as $e \to 0$,
(1) The triangular mesh $M^1(f,e)$ is a 2-dimensional manifold homeomorphic to F under the nearest point projection map.
(2) The surface $M^1(f,e)$ converges to F piecewise-differentiably.
(3) The mesh angles lie in the interval $[35.2^\circ, 101.5^\circ]$.

Piecewise-differentiable convergence means that (1) as $e \to 0$ the mesh converges pointwise to F and that (2) for any $\varepsilon > 0$, when e is sufficiently small the distance between a unit normal vector on the mesh at some point and the unit normal at the nearest point on F is

less than ε. Note that while normal vectors are not uniquely defined at a point which is a vertex or on an edge of a mesh, this property holds for any of the finite number of choices for a normal.

2. Related Work

A large number of algorithms has been written for representing surfaces given by an implicit function and almost all of them originate from Marching Cubes [LC87]. Marching cubes applied directly does not give good angle bounds, since a plane cutting close to an edge will intersect in a triangle containing an arbitrarily small angle. A relatively recent survey on implicit surface meshing techniques by Araujo [dALJ*15] gives an overview of different approaches to isosurface meshing. The article classifies and compares techniques for fast visualization of isosurfaces based on different features of meshes including quality of meshes but does not discuss provable bounds on angles. Recent results in this direction can be found at [Wen13, CPS19]. Labelle and Shewchuk developed an "Isosurface Stuffing" procedure that achieves dihedral angle bounds for tetrahedra filling a 3-dimensional region, along with angle bounds for the 2-dimensional mesh formed by the region's boundary [LS07]. It achieves angles in the interval $[16^\circ, 145^\circ]$ for the boundary surface mesh. Liang and Zhang used a related octree method to find meshes of regions bounded by smooth curves in the plane that are guaranteed to have angles in the interval
$[19.47^\circ, 141.06^\circ]$ [LZ14]. P. Chew gave a procedure based on point insertion and remeshing to achieve a Constrained Delaunay triangulation for a surface in the plane or in R^3 that gives angles between 30° and 120° [Che93]. A recent remeshing algorithm by Hu et. al. [HYB*17] gives experimental evidence for mesh regularity comparable to the GradNormal algorithm.

The problem of finding meshes with good angle properties has been extensively studied for subregions of the plane with fixed boundary. We refer to the survey articles by Bern and Eppstein [BE92] and Zamfirescu [Zam13]. Some approaches create Delaunay Triangulations for planar regions, which give various forms of optimal regularity for a given vertex set [CDS12]. However Delaunay Triangulations can produce triangles with small angles.

In some settings acute triangulations can be realized. Work of Burago and Zalgaller shows that any polyhedral surface has a subdivision that is acute [BZ60] (see also [Sar09], [HU07]). Colin de Verdiere and A. Marin showed that any smooth Riemannian surface admits a sequence of geodesic triangulations with vertices on the surface and angles that, in the limit, lie in the intervals $[3\pi/10, 2\pi/5]$ for the case of genus zero, $[\pi/3, \pi/3]$ for genus one, and $[2\pi/7, 5\pi/14]$ for the case of genus greater than one [dVM90]. By Gauss-Bonnet, these bounds are optimal for smooth surfaces. Their results use the Uniformization Theorem and constructing triangulations on an appropriate conformal model in the Moduli Space associated to the surface. We are not aware of algorithms based on these approaches.

3. Dimension two

In this section we explain the idea behind the GradNormal algorithm in the simpler setting of curves in the plane. The 3dimensional setting will be presented in the next section. The planar algorithm takes as input an implicit curve given by a function $f : R^2 \to R$ with domain containing the unit square $I^2 = [0,1]\times[0,1]$ and outputs a piece-wise linear curve that approximates the level curve $F = f^{-1}(0)$ within the unit square.
Step 1 (lower-dimensional analog of producing the MidNormal mesh): A neighborhood of the unit square is tiled by triangles. The algorithm evaluates the function f on a triangle's vertices and checks if all function values have the same sign. If not then it takes midpoints of two edges with endpoints of opposites signs and connect them by a straight line segment. This becomes an edge of the resulting polygonal curve α as in Figure 7.

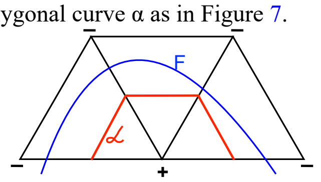


Figure 7: *In the planar version of the MidNormal mesh, a curve F passing through three triangles is approximated by a polygonal "normal" curve* α, *which passes through midpoints of triangle edges.*

Note that in contrast to Marching Tetrahedra in this dimension, the vertices are not interpolated along the edge, but are always taken at midpoints. See Figure 8.

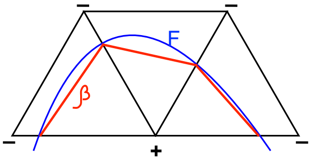


Figure 8: *In the planar version of the Marching Tetrahedra mesh, the curve F is approximated by a polygonal curve β that meets triangle edges where the curve does.*

Step 2 (lower-dimensional analog of the projection step of GradNormal): Each vertex $v \in \alpha$ is projected to the closest point on the isocurve F to give a polygonal curve approximating F that has vertices on F. For differentiable functions f, GradNormal computes the gradient ∇f of the function f at vertices of the polygonal curve γ and uses a first order method to estimate the closest point. This takes a vertex v to $v = v - f(v)\nabla f/||\nabla f||^2$, with v a vertex of the final mesh. Alternate methods to find the closest point are used when the gradient function is not available. See Figure 9.

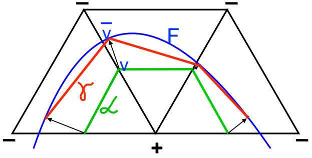


Figure 9: *In the second step of the GradNormal algorithm, each vertex of α is projected to the closest point on the level curve, resulting in the polygonal curve γ.*

Step 3: The 3-dimensional version of the GradNormal algorithm requires an additional operation to achieve the claimed angle bounds. After the projection step, small angles can appear in one particular configuration. The corrective operation involves removing four triangles meeting at a valence four vertex and adding a diagonal to the resulting quadrilateral, as in Figure 10.

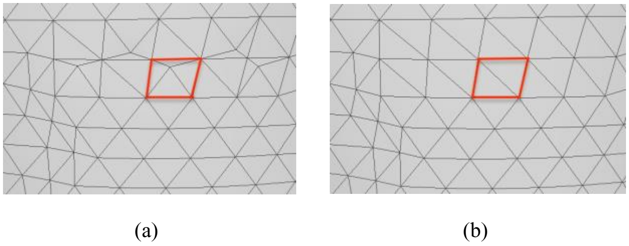

(a) (b)

Figure 10: *Poor angles are eliminated by removing the four edges meeting each valence four vertex (as in the outlined quadrilateral in (a)) and adding a diagonal to the resulting quadrilateral (see (b)).*

The GradNormal algorithm has a superficial resemblance to the Dual Contouring Algorithm, where every edge intersecting the isocurve gives an edge of the final mesh and every cell intersected by the curve gives a vertex, Figure 11 [AF05]. But there is little actual overlap between them.

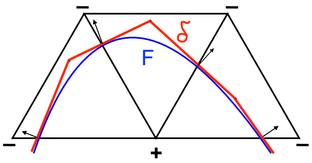


Figure 11: *In the Dual Contouring Algorithm, the curve F is approximated by a polygonal curve δ whose normals agree with those of F along intersections with triangle edges.*

4. The GradNormal Algorithm

In this section we describe in detail the two steps of the GradNormal Algorithm. We first describe the tiling by tetrahedra used to generate the intermediate mesh in Step 1 of the algorithm. We then describe the projection used to obtain the finial mesh.

4.1. Goldberg tetrahedra

The classification of tilings of R^3 by tetrahedra is still not completely understood. An interesting historical note is that Aristotle falsely claimed that regular tetrahedra can meet five-to-an-edge and fit together to tile space [Sen81]. In fact, the dihedral angle of a regular tetrahedron is somewhat less than $2\pi/5 = 72^\circ$, so they do not fit evenly around an edge. The search for tetrahedra that do fit together led Sommerville to find four tetrahedral shapes that tile R^3. Baumgartner found a further example and Goldberg discovered three infinite families. Eppstein, Sullivan and Ungor constructed tilings of space by acute tetrahedra, with all dihedral angles less than 90° [ESU04]. It might seem that tetrahedra that are acute, or as close to regular as possible, are preferable for producing regular triangulations, but that turns out not to be the case. A search through the infinite family of tilings discovered by Goldberg was carried out and led to the tiling that gives the best mesh angles for our method.

A tetrahedron in the Goldberg family, shown in Figure 6, is constructed by first tiling the xy-plane with equilateral triangles of unit length. Three edges of the tetrahedron are graphs over edges of one of these equilateral triangles, each rising by a distance of a from its initial to its final vertex. The other edges connect pairs of the resulting four vertices. The vertical edge AB has length $3a$. If we rescale by a factor of e then the equilateral triangle has edge length e and the edge AB has length $3ae$. We call this tetrahedron $\tau_{a,e}$. Every Goldberg tetrahedron has two edges with dihedral angle $\pi/2$ and one with dihedral angle $\pi/3$. In our application,√ e determines

the size of the tetrahedron and we take $a = e\ 2/4$. This choice of a optimizes the resulting mesh angles among those obtained by Goldberg tetrahedra, and can be proved to give a near optimal mesh when our method is applied to any tetrahedral shape. The resulting tetrahedron shape coincides with a tiling described by Sommerville [Som23] and has vertices located along the body-centered cubic lattice.

4.2. Step 1 of the GradNormal Algorithm

Algorithm 1 Step 1 of the GradNormal Algorithm - the MidNormal Procedure

```
procedure MidNormal(e)
```

```
Input a function $f : I^3 \rightarrow R$, a choice of scale $e = 1/N$.
for $i = 1$ to $6N^3$ do
    Compute the sign of $f$ at the four vertices $\tau_i 1,\tau_i^2,\tau_i^3,\tau_i^4$ of tetrahedron $\tau_i$ in a tiling of the unit cube $I^3$ by Goldberg tetrahedra isometric to $\tau_e$. If the value of $f$ at a vertex is exactly zero, take the sign to be positive.
    If the sign of $f$ is different at one vertex of $\tau_i$ from the sign at the remaining three vertices, add to a list of triangles T the elementary normal triangle in $\tau_i$ that separates that vertex from the remaining three. The vertices of this triangle are located at midpoints of edges of $\tau_i$.
    If the sign of $f$ is different at two vertices from the sign at the remaining two vertices of $\tau_i$, add to T two triangles formed by taking the normal quadrilateral in $\tau_i$ that separates the two pairs of vertices and adding a diagonal as follows: For quadrilateral $LMNQ$ add diagonal $MN$. For quadrilateral $KMPN$ add diagonal $MN$. For quadrilateral $KLPQ$ add diagonal $LQ$. Again position the vertices of these triangles at midpoints of edges of $\tau_i$. See Figure 6(a) for notation.
Output T .
```

It is straightforward to compute the edge lengths and angles in the triangular meshes obtained by applying the MidNormal procedure to the tetrahedral tilings of R^3 produced by the Goldberg tiling.

Lemma 4.1 The MidNormal mesh produced using the Goldberg tiling is a 2-dimensional manifold mesh that has angles in the interval $[45^\circ, 90^\circ]$.

Remark. When applied to a smooth compact implicit surface $F = f^{-1}(0)$ that has an embedded ε-tubular neighborhood $N_\varepsilon(F)$, or equivalently having reach ε, and given a positive constant $e < \varepsilon/2$, the MidNormal algorithm produces a mesh $M(f,e)$ satisfying

1. $M(f,e)$ is an embedded 2-dimensional manifold.
2. The triangles around a given vertex are graphs over a common plane.
3. The surface $M(f,e)$ converges to F in Hausdorff distance as $e \rightarrow 0$.
4. The surface $M(f,e)$ is isotopic to F in $N_\varepsilon(F)$.
5. The nearest neighbor projection from the mesh $M(f,e)$ to F is a homeomorphism for e sufficiently small.

The faces of the approximating meshes have normal vectors that lie in a fixed finite set of 18 normal directions, and therefore the approximation is continuous, but not piecewise-differentiable. However the geometry of the approximating surfaces is uniformly biLipschitz equivalent to the limiting surface F.

4.3. The GradNormal algorithm: Steps 2 and 3

The second step of the GradNormal algorithm moves the vertices of the MidNormal mesh so that they lie on the implicit surface F. A vertex of the MidNormal mesh is moved to the closest point on F. This point is unique when F is differentiable and the mesh lies sufficiently close to F. The current implementation uses the gradient of the function f defining F in this step. In the case of a linear function it exactly projects each vertex to the nearest point on F. In general it produces a first-order approximation of F, improving the zeroth-order approximation given by the MidNormal procedure. An alternative to using the gradient projection would be to call a function that returns for each point the coordinates of the nearest point on F.

Unfortunately the mesh resulting from the projection process can have sliver triangles with arbitrarily small angles, so a corrective step 3 is needed. An analysis of the badly behaving triangles shows that they all result from a particular phenomenon that can be easily corrected. We will show that small angles in the projected mesh can only arise from projecting angles that lie in one of four triangles that are adjacent in the MidNormal mesh to a vertex of valence four. The GradNormal algorithm corrects this by removing any set of four triangles that meet at a common valence-four vertex and adding a diagonal to the resulting quadrilateral, as in Figure 10.

This valence-four vertex move will be shown to eliminate all sliver triangles and to give a high quality mesh with the claimed angle bounds of $[35.2^\circ, 101.5^\circ]$. We now state the algorithm.

The proof of the resulting properties claimed in Theorem 1.1 is rather lengthy, and is given in full in the Appendix. In this section we sketch the general idea.

A smooth compact surface F has bounded curvature and as $e \rightarrow 0$, its intersection with a tetrahedron $\tau(e)$ is closely approximated by a plane Q. This plane can be chosen to be the plane that intersects the edges of the tetrahedron at the points where F intersects these edges. Thus the angles of the nearest point projection of an elementary normal disc in τ of diameter less than e onto F has angles that converge as $e \rightarrow 0$ to the angles determined by the nearest point projection onto the plane Q.

Algorithm 2 GradNormal Algorithm

```
procedure GRADNORMAL($e, f$)
    Input a differentiable function $f : I^3 \rightarrow R$ with level set $F = f^{-1}(0)$ and a choice of scale $e = 1/N$.
    Apply the MidNormal procedure with size parameter $e$ to obtain a mesh $M(f,e)$.
    Compute the gradient $\nabla f$ at the vertices of $M(f,e)$.
    Remove each vertex of valence 4 and its four adjacent triangles. Add a diagonal to the resulting quadrilateral, giving two new triangles in the mesh.
    Relocate each vertex $v$ to $v - f(v)\nabla f/||\nabla f||^2$.
    Output the resulting list of triangles $T^0$.
```

When the surface F separates vertices of a tetrahedron τ it defines elementary normal discs which are very close to flat

triangles or quadrilaterals. The MidNormal procedure produces a normal surface that intersects a tetrahedron τ along one of four triangles or one of three quadrilaterals. Four triangles 4*KLM,*4*KNQ,*4*LNP* and 4*MPQ* come from elementary triangles. The quadrilaterals are divided into two triangles, leading to six additional triangles,
4*KLQ* and 4*LPQ*, 4*KMN* and 4*MNP*, and 4*LMN* and 4*MNQ*.
We need to consider the angles obtained when these 10 triangles are projected onto *F*, which as noted can be assumed to be a plane.

To capture all possible projections we consider all possible planes that could represent *F*. We describe such a plane by its unit normal vector, a point on the unit sphere. The space of such planes is given by certain regions on the unit sphere. We prove that the smallest angles of a projected triangle must happen along the boundary of one of these spherical regions. Each boundary segment is parameterized and its minimal angle values are evaluated. This computation gives the desired angle bounds except for triangles that meet a valence-four vertex. It is essential for the achievement of the angle bounds to remove these triangles.

Altogether there are 12 triangles with 36 angles projecting to four edges each, or 144 angle functions in total, each defined on an interval of normal directions connecting two points on the sphere along a spherical arc. The union of all these angle functions is graphed in Figure 18. Computations carried out in Mathematica show that all angles are between 35.25° and 101.45°.

Vertices of valence-four come from an intersection with an edge of type *AD* in a Goldberg tetrahedron, as shown in Figure 6. This edge has a dihedral angle of 90° in each of the four adjacent tetrahedra, and the four adjacent tetrahedra combine to form an octahedron as in Figure 12. The removal of triangles meeting vertices of valence-four and their replacement by two triangles gotten by adding a diagonal improves the angle bounds of the resulting triangles to lie between 35.25° and 101.45°.

The nearest point projection from mesh $M(f,e)$ to F is a homeomorphism for e sufficiently small. When f is linear, the projected triangle is contained in F, and gives a C^1 approximation for e sufficiently small. In the argument above, the angle bounds established for F a plane also hold for e sufficiently small, since $F \cap \tau$ converges smoothly to the intersection of a plane with τ as $e \rightarrow 0$.

5. Remarks

5.1. Other Surface Descriptors

The GradNormal algorithm takes as input an implicit surface, given as a level set of a function on R^3, but is amenable to other forms of surface input. For example, if the input is a triangulated surface *F* having poor mesh quality, then there exist procedures to produce a function on R^3 that computes distance from the surface. Such signed distance functions have been extensively studied [Sym98,PT92]. If the input describing a surface is a point cloud, methods such as the Moving Least Squares and Adaptive Moving Least Squares produce a function giving a level set description of the surface [Dey06,SOS04]. This function can then be used as input to the GradNormal algorithm.

5.2. Convergence and curvature

When we have bounds on the principle curvatures of *F* we can get angle bounds on the mesh for a given value of *e*. We investigate these bounds here, as they are relevant to whether the GradNormal algorithm can be used effectively. The bounds of Theorem 1.1 are guaranteed to apply as the scale size $e \rightarrow 0$. To test them at a given size, we can fix $e = 1$ and consider how the angle bounds on the mesh are affected by curvature bounds on the surface *F*. Though this can be done rigorously, we present here some experimental results obtained as a preliminary step. These give some preliminary evidence that the GradNormal algorithm is fast in practice and converges at reasonable scale.

We set $e = 1$ and consider the angles attained by a mesh approximating a surface *F* whose principle curvatures are bounded above in absolute value by a constant k_0. We estimate these angles by modeling *F* with a sphere. Since spheres of the appropriate radius have maximal principal curvatures and since they realize all tangent directions, this gives a reasonable approach to modeling the worst case for an angle bound. We obtain in this way experimental bounds for the angles obtained in the GradNormal algorithm. In Table 1 the result of applying the GradNormal algorithm to surfaces of genus zero (sphere), genus two and genus five. These surfaces are scaled to exhibit the effect of varying curvature on the quality of the output mesh. Computations at four different scales were carried out for each surface on a 2014 MacBook Pro. The number of tetrahedra used to tile the cube in the four runs was 113,100, 869,652, 6,748,416 and 52,931,340. Each run took at most 1,141 seconds on a 2014 MacBook Pro with a 2.5 GHz Quad-Core Intel Core i7. After rescaling to fix the size of the tetrahedra, the principle curvatures of the implicit surfaces are bounded above by k_M, shown in the second column. The following columns show the minimum θ_m and maximum angles in the resulting mesh θ_M.

5.3. Running time

The GradNormal algorithm as implemented runs in linear time in the number of tetrahedra used in the 3-dimensional tiling. This follows from the same arguments used to establish this bound for Marching Cubes [Wen13]. There are no point insertion or remeshing steps, other than a single round involving valence four vertex removal in Step 3. While the python code in the implementation

Genus	kM	θ_m	θ_M	vertices	faces
0	0.23	33.0°	102.8°	1,082	1,988
	0.09	34.2°	101.3°	6,782	12,564
	0.05	35.4°	102.7°	27,104	50,300
	0.03	35.2°	101.1°	433,208	866,412
2	0.57	10.9°	153.8°	1,336	2,540
	0.29	22.2°	129.0°	5,438	10,306
	0.15	27.8°	118.8°	21,880	41,600

	0.08	31.7°	108.9°	87,802	166,898
5	0.8	26.6°	122.0°	7,318	14,652
	0.4	29.2°	113.4°	29,348	55,122
	0.2	30.8°	109.4°	117,878	235,772
	0.1	33.4°	104.4°	471,696	943,408

Table 1: *A tetrahedral tiling with scale given by* $e = 1$ *is intersected with surfaces whose principle curvatures are bounded above by* k_M. *The* θ_m *and* θ_M *columns show the minimal and maximal mesh angles. The vertex and face columns indicate the number of vertices and faces in the mesh. As* $k_M \to 0$ *the surface becomes flatter, and the angles converge to lie within the predicted interval* $[35.2^\circ, 101.5^\circ]$

has not yet been optimized for speed, the code that produced the genus-two surface in Figure 2 ran in 15.8 seconds on a 2014 MacBook Pro.

5.4. Computational Methodology

The proof of Theorem 1.1 involves extensive angle computations with trigonometric functions. These were carried out with the software package Mathematica 12. The GradNormal Algorithm has been implemented in Python. Files are available at [HT19]. The values obtained for the angle bounds depend on the accuracy of Mathematica floating point functions. It is possible to calculate these bounds using interval arithmetic, but this has not yet been done.

6. Conclusion

We present a novel algorithm for meshing an isosurface. The main advantages include simplicity of the algorithm, fast running time, low space usage and guaranteed triangles of high quality when applied to a smooth surface. Experiments with surfaces defined by simple mathematical functions indicate that the resulting mesh has reasonable angle bounds even with a coarse tetrahedral tiling, and that the guaranteed limiting angle bounds of $[35.2^\circ, 101.5^\circ]$ for meshes approximating smooth surfaces are approached using practical sized tilings of space. A drawback of the GradNormal algorithm, as with other methods based on regular meshes, is that it is not adapted to local surface features. In particular, sharp features, angles and corners may not be captured by the procedure, and the guaranteed angle bounds do not apply when GradNormal is applied to non-smooth surfaces.

References

[AF05] Azernikov S., Fischer A.: Anisotropic meshing of implicit surfaces. In *Proceedings of the International Conference on Shape Modeling and Applications 2005* (2005), pp. 94–103. 5

[BE92] Bern M., Eppstein D.: Mesh generation and optimal triangulation. In *Computing in Euclidean Geometry* (1992), Du D.-Z., Hwang F. K.-M., (Eds.), vol. 1 of *Lecture Notes Series on Computing*, World Scientific, pp. 23–90. 4

[BEY91] Bern M., Eppstein D., Yao F.: The expected extremes in a delaunay triangulation. *International Journal of Computational Geometry & Applications 1* (1991), 79–91. 1

[Bun] Stanford university computer graphics laboratory, stanford bunny. URL: https://graphics.stanford.edu/data/3Dscanrep/ [cited 6/24/20]. 1

[BZ60] Burago Y., Zalgaller V.: Polyhedral embedding of a net. *Vestnik St. Petersburg Univ. Math.* (1960), 66–80. 4

[CCC*08] Cignoni P., Callier M., Corsini M., Dellepiane M., Ganovelli F., Ranzuglia G.: Meshlab: an open-source mesh processing tool. In *Sixth Eurographics Italian Chapter Conference* (2008), pp. 129–136. 1

[CDS12] Cheng S., Dey T., Shewchuk J.: *Delaunay Mesh Generation*. CRC, 2012. 3, 4

[Che93] Chew L. P.: Guaranteed quality triangular meshes. In *Proceedings of the Ninth annual Symposium on Computational geometry* (San Diego, 1993), pp. 274–280. 4

[CPS19] Custodio L., Pesco S., Silva C.: An extended triangulation to the marching cubes 33 algorithm. *Journal of the Brazilian Computer Society 25*, 1 (2019). 4

[dALJ*15] de Araujo B. R., Lopes D. S., Jepp P., Jorge J., Wyvill B.: A survey on implicit surface polygonization. *ACM Comput. Surv. 47*, 4 (2015). 4

[Dey06] Dey T.: Curve and surface reconstruction: Algorithms with mathematical analysis. In *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, 2006. 7

[DH20] Dawson-Haggerty M.: Trimesh [computer software], 2020. URL: https://github.com/mikedh/trimesh. 2

[dVM90] de Verdiere Y. C., Marin A.: Triangulations presque equilaterales des surfaces. *J. Differential Geom 32* (1990), 199–207. 4

[ESU04] Eppstein D., Sullivan J., Ungor A.: Tiling space and slabs with acute tetrahedra. *Comp. Geom. Theory & Applications 27*, 3 (2004), 237–255. 5

[Gol74] Goldberg M.: Three infinite families of tetrahedral spacefillers. *J. Comb. Theory 16* (1974), 348–354. 3

[HT19] Hass J., Trnkova M.: Gradnormal source files. gitlab.com/joelhass/midnormal, 12 2019. 2, 7

[HU07] H.Erten, Ungor A.: Computing acute and non-obtuse triangulations. In *CCCG* (Ottawa, Canada, 2007). 4

[HYB*17] Hu K., Yan D. M., Bommes D., Alliez P., Benes B.: Error-bounded and feature preserving surface remeshing with minimal angle improvement. *IEEE Transactions on Visualization and Computer Graphics 23*, 12 (Dec 2017), 2560–2573. 4

[Kne29] Kneser H.: Geschlossene flachen in dreidimensionalen mannigfaltigkeiten. *Jahresbericht Math. Verein 28* (1929), 248–260. 3

[LC87] Lorensen W., Cline H.: Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH Computer Graphics* (1987), vol. 21, ACM, pp. 163–169. 4

[LS07] Labelle F., Shewchuk J.: Isosurface stuffing: Fast tetrahedral meshes with good dihedral angles. In *Proceedings of ACM SIGGRAPH* (2007), Proceedings of ACM SIGGRAPH, ACM, ACM Press. 4

[LZ14] Liang X., Zhang Y.: An octree-based dual contouring method for triangular and tetrahedral mesh generation with guaranteed angle range. *Engineering with Computers 30* (2014). 4

[NH91] Nielson G. M., Hamann B.: The asymptotic decider: Resolving the ambiguity in marching cubes. In *Proceedings of the 2nd*

Conference on Visualization '91 (1991), pp. 83–91. 3

[PT92] Payne B., Toga A.: Distance field manipulation of surface models. In *Computer Graphics and Applications* (1992), 12, IEEE, pp. 65–71. 7

[Sar09] Saraf S.: Acute and non-obtuse triangulations of polyhedral surfaces. *European J. Combin 30* (2009), 833–840. 4

[Sen81] Senechal M.: Which tetrahedra fill space? *Mathematics Magazine 54* (1981), 227–243. 5

[Som23] Sommerville D.: Space-filling tetrahedra in euclidean space. *Proc. Edinburgh Mathematical Society 41* (1923), 49–57. 3, 5

[SOS04] Shen C., O'Brien J., Shewchuk J.: Interpolating and approximating implicit surfaces from polygon soup. In *Proceedings of ACM SIGGRAPH* (2004), Proceedings of ACM SIGGRAPH, ACM, ACM Press, pp. 227–243. 7

[Sym98] Symposium on Volume Visualization:. *Using distance maps for accurate surface representation in sampled volumes* (1998), IEEE. 7

[TL94] Turk G., Levoy M.: Zippered polygon meshes from range images. In *Proceedings of ACM SIGGRAPH* (1994), Proceedings of ACM SIGGRAPH, ACM, ACM Press, pp. 311–318. 1

[Wen13] Wenger R.: *Isosurfaces: Geometry, Topology, and Algorithms*. A K Peters/CRC Press, 2013. 4, 7

[Zam13] Zamfirescu C.: Survey of two-dimensional acute triangulations. *Discrete Mathematics 313* (2013), 35–49. 4

7. Appendix A.

In this Appendix we present the details of the proof of Theorem 1.1. We first state a result on the distortion of angles under projections to a rotated plane.

Lemma 7.1 Suppose that~v = (v_1,1),v_1 > 0 is a vector in the first quadrant of the xy-plane and that ~w = (w_1,w_2)6=~0 and subtends an angle α < π with~v. Rotate the xy-plane around the x-axis through an angle of θ, 0 ≤ θ ≤ π/2 and denote the orthogonal projections of the rotated vectors~v,~w back to the xy-plane by~v(θ)),~w(θ). Then as θ increases from 0 to π/2 the angle α(θ) between~v(θ) and ~w(θ) satisfies:

(1) If ~w is parallel to the positive x-axis or to the negative y-axis then α(θ) is monotonically decreasing.

(2) If ~w is parallel to the negative x-axis or to the positive y-axis then α(θ) is monotonically increasing.

(3) If ~w lies in the interior of the second quadrant then α(θ) is monotonically increasing.

(4) If ~w lies in the interior of the fourth quadrant then α(θ) is monotonically decreasing.

(5) If ~w lies in the interior of the first quadrant then α(θ) achieves its minimum at an endpoint of the interval [0,π/2].

(6) If ~w lies in the interior of the third quadrant then α(θ) achieves its maximum at an endpoint of the interval [0,π/2].

Proof Rotation about the x-axis through an angle of θ takes the point (x,y,0) to (x,$y$cosθ,$y$sinθ). Thus ~v(θ) = (v_1,cosθ) and ~w(θ) = (w_1,$w_2$cosθ). The angle between each vector and the $x$axis is decreasing with θ, implying the claims in Cases (1) – (4). The last two cases needs a more detailed investigation. In Case (5) each of w_1,w_2 is positive, and we can assume that w_2 = 1 by scaling. The angle α(θ) between~v(θ) and ~w(θ) satisfies

cosα(θ) =

v_1w_1 +cos2θ . p($v_1$2

+cos2θ)p($w_1$2 +cos2θ)

For given vectors~v and ~w the cosine of α(θ) has first derivative

$$(\cos\alpha(\theta))' = \frac{\sin\theta\cos\theta(v_1-w_1)^2(v_1w_1-\cos^2\theta)}{(v_1^2+\cos^2\theta)^{3/2}(w_1^2+\cos^2\theta)^{3/2}}.$$

A computation shows that the critical points of cosα(θ) lie either at the boundary of the interval [0,π/2], or in the case where v_1w_1 < 1, at an interior point where θ= arccos $\sqrt{v_1w_1}$. A further computation shows that the second derivative at the interior critical point is positive, so there is no interior local maximum. Thus the cosine of α is maximized at the endpoints of θ ∈ [0,π/2], implying that the angle α(θ) is minimized at one of these two endpoints. For Case (6), where the angle between~v and ~w is greater than π/2, we note that this angle is complementary to that between ~v and −~w, which was studied in Case (5). Thus a maximum in this case coincides with a minimum in Case (5), and this again occurs at an endpoint of the interval as claimed. □

Corollary 7.1 Suppose two vectors in R^3 are orthogonally projected to a family of rotated planes that begins with the plane containing them and contains planes rotated about a line through an angle of at most π/2. If the vectors subtend an angle smaller or equal to π/2 then the minimum angle between the projected edges occurs at either the initial or final projection. If they subtend an angle greater than π/2 then the maximum angle between the projected edges occurs at either the initial or final projection.

Let $M(f,e)$ be the mesh produced by the MidNormal procedure and $M^1(f,e)$ a projection of $M(f,e)$ along gradient vectors of f towards the surface F as in the GradNormal algorithm. Since F is smooth and compact it has bounded curvature and as e → 0, its intersection with a tetrahedron τ is increasingly closely approximated by the plane that intersects the edges of the tetrahedron at points where F intersects these edges. When the surface F separates the vertices of τ so as to define an elementary normal disk E, then the plane Q separating the same vertices and intersection the edges of τ at points where F intersects these edges smoothly converges to F on a neighborhood of τ of radius e. Thus the angles of the nearest point projection of an elementary normal triangle in τ of diameter less than e onto F gives angles that converge as e → 0 to the angles determined by the nearest point projection onto the plane Q.

We note that in the GradNormal projection we do not project vertices onto the surface F, but rather onto the plane where F would be if f was a linear function. This plane smoothly converges to F in a neighborhood of τ_e as e → 0. We conclude that in computing the angles of a projection of an elementary normal triangle in τ_e whose three points have been projected to F, we can assume, with arbitrarily small error as e → 0, that F is a plane that separates the vertices of τ in the same way as the normal surface F.

We now classify the various cases of how a plane F can intersect a tetrahedron τ. There are four cases where $F \cap\tau$ is a triangle and three where it is a quadrilateral that is divided into two triangles along a diagonal. A special case occurs when four adjacent tetrahedra meet along an edge of valence four and produce a rhombus which is then divided into two triangles. Counting cases, we see that there are altogether 12 triangles and 36 angles that can be projected onto some plane. Due to symmetry, some of these are equivalent. The case of a valence-four vertex in $M(f,e)$ includes two triangle shapes up to isometry, and requires special treatment and we consider it first because it affects projections of the remaining cases. Valence-four come from an intersection with an edge of type AD in a Goldberg tetrahedron, as in Figure 6.

We now consider the valence-four vertices. A vertex of valencefour appears in the mesh $M(f,e)$ when four elementary normal triangles meet the edge AD at its midpoint M. This edge has a dihedral angle of 90° in each of the four adjacent tetrahedra, and the four adjacent tetrahedra combine to form an octahedron as in Figure 12.

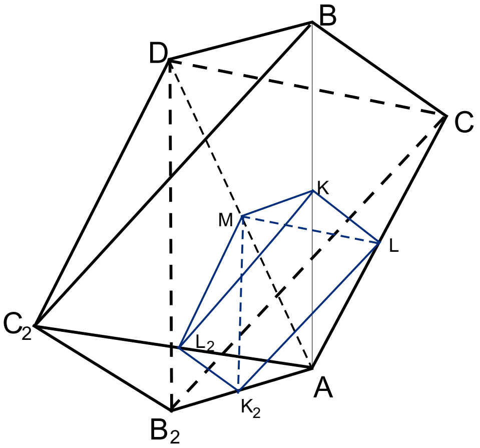


Figure 12: *Four adjacent tetrahedra meet along AD, forming an octahedron. The mesh surface meets this octahedron in four triangles, with a common valence-four vertex at M.*

We consider first the case where F is a plane that intersects the octahedron separating vertex A from vertices B,C,D. We denote by X the closure of the set of unit vectors perpendicular to such planes, oriented to point towards A. We denote by Y the subset of X consisting of normals to planes separating vertex A from vertices B,C,D,B_2,C_2. We want to study angle bounds of projected triangles described by the set Y after a valence-four vertex has been removed and a new edge have been inserted in the remaining quadrilateral.

For a plane separating vertex A from vertices B,C,D,B_2,C_2, the induced mesh has a valence-four vertex where it intersects edge AD. The GradNormal algorithm removes the four triangles adjacent to the edge AD: 4KLM, 4KL_2M, 4K_2LM and 4K_2L_2M.

Note that the four vertices B,C,B_2,C_2 are coplanar, since there is a reflection through M preserving the octahedron and interchanging A and D, B and B_2, and C and C_2. These four triangles form a pyramid $MKLK_2L_2$ whose base is a flat rhombus parallel to rhombus BCB_2C_2. For $a \underset{\circ}{=} e\sqrt{2}/$4, the rhombus is a square that realizes dihedral angles of 45 with the faces ABC, ABC_2, AB_2C and AB_2C_2 of the octahedron, as indicated in Figure 12. We now analyze the location of the set Y in the unit sphere.

Claim 7.1 Suppose F is a plane separating vertex A from vertices B,C,D,B_2,C_2. Then the unit normal vector of the plane F lies in the interior of a spherical quadrilateral $Y \subset X$. The vertices of Y are normal to the faces ABC, ABC_2, AB_2C and AB_2C_2.

Proof The set of planes with these separation properties is a subset of the 3-dimensional set of planes in R^3, and their unit normal vectors Y form a 2-dimensional subset of the unit sphere. If a plane with normal vector in Y does not meet a vertex of the octahedron then it is in the interior of an open disk contained in Y, since it can be rotated in any direction while remaining in Y. The same is true for planes that meet only one vertex of the octahedron, since they too can be rotated in all directions while still passing through only this vertex. Planes in Y meeting two vertices of the octahedron can be rotated only in one circular direction, and lie along a geodesic arc on the 2-sphere that forms part of ∂Y. Planes that meet three or more vertices of the octahedron cannot be rotated while maintaining their intersection with these points, and thus form vertices of ∂Y. To understand Y we consider which planes separating vertex A from vertices B,C,D,B_2,C_2 meet three or more vertices, giving a vertex of ∂Y on the unit sphere, or meet two vertices, giving an edge of ∂Y.

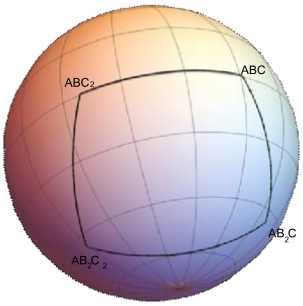


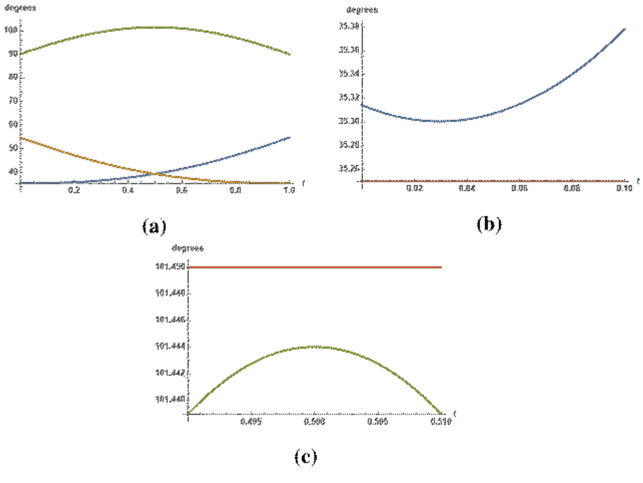
Figure 13: *The spherical quadrilateral Y indicates normal directions to planes that separate vertex A from vertices B,C,D,B2,C2.*

Moreover any plane separating A from B,C,D,B_2,C_2 can be displaced through parallel planes towards A till it contains A. It follows that the vertices of Y are determined by triples of vertices that include A and are limits of planes with the right separation property. These are given by normals $\sim n_{ABC}, \sim n_{ABC_2}$ $vecn_{AB_2C}, \sim n_{AB_2C_2}$ to the faces ABC, ABC_2, AB_2C and AB_2C_2, each of which gives a vertex of ∂Y. These four points on the unit sphere are vertices of a spherical quadrilateral forming Y. All planes that separate vertex A from the other vertices of the octahedron with normal pointing towards A have unit normal vectors lying inside Y. See Figure 13.

□

In the GradNormal algorithm we replace the four triangles adjacent to edge AD with the rhombus BCB_2C_2, divided into two triangles along a diagonal. We need to estimate the angles of these two triangles after they are projected onto a plane F with normal in the spherical quadrilateral Y. Lemma 7.1 implies that the largest and smallest angles among projections of the rhombus KLK_2L_2 onto F occur either in the rhombus KLK_2L_2 itself or at a plane whose normal lies in ∂Y. This rhombus is a square and a diagonal divides it into a pair of $45^{\circ}, 45^{\circ}, 90^{\circ}$ triangles.

We project these two triangles onto planes with normals on ∂Y. The rhombus KLK_2L_2 projects to a parallelogram, so the two triangles project to congruent triangles, and it suffices to consider the angles of one, say KLK_2. We investigate what angles result from projecting triangle KLK_2 onto a plane normal to ∂Y. Each point in an arc of ∂Y is normal to a plane obtained by rotating one face of the octahedron to another through an edge containing A. One set of angles results from projecting each of the three angles of triangle KLK_2 to planes determined by the spherical arc from~n_{ABC_2} to ~$n_{AB_2C_2}$. We parameterize an arc of normal vectors ~$v(t)$ passing from~$v(0)$ =~n_{ABC_2} to~$v(1)$ =~$n_{AB_2C_2}$ and compute the angles resulting from projecting triangle KLK_2 to planes normal to~$v(t)$. These angles are then given by a collection of functions of a parameter $t \in [0,1]$. The three angle functions from triangle KLK_2 are plotted in Figure 14. The absolute minimum of the three angle functions on this arc of ∂Y is $\approx 35.3004^{\circ} > 35.25^{\circ}$, and the absolute maximum is $\approx 101.445^{\circ} < 101.45^{\circ}$. We then do a similar computation for each of the other arcs on ∂Y. Figure 15 shows the angles resulting from projecting $4KLK_2$ onto the boundary arc of Y running between~n_{ABC} and~n_{ABC_2}. Again each curve lies above 35.25° and below 101.45°, showing that all projected angles are between these two bounds. The remaining two boundary arcs give the same angle functions, due to a symmetry of the octahedron.



(a) (b)

(c)

Figure 14: *(a) Angles of* $4KLK_2$ *after projection onto the boundary arc from* ~n_{ABC} *to* ~n_{AB_2C} *of* ∂Y*, parametrized by* $t \in [0,1]$. *Detailed views of these graphs near (b)* $t = 0$ *and (c)* $t = 0.5$ *indicate that each curve lies above* 35.25° *and below* 101.45°.

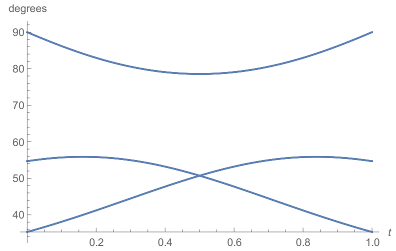


Figure 15: *Angles of* $4KLK_2$ *after projection onto the boundary arc from*~n_{ABC} *to*~n_{ABC_2} *of* ∂Y*. Again each curve lies above* 35.25° *and below* 101.45°.

We conclude that all projections of the triangles obtained from the diagonally divided rhombus in the GradNormal algorithm have angles between 35.25° and 101.45°.

There is a symmetric case involving a rhombus where F is a plane that separates vertex D from A,B,C. A symmetry interchanges A and D, and it follows that this case gives the same angle bounds.

There are 10 remaining cases to consider for projecting along a gradient besides the valence-four case considered above. Four of them involve angles obtained by projecting triangles $4KLM$ and $4KNQ$ with edge lengths $(b/2, b/2, c/2)$, and $4LNP$ and $4MPQ$ with edge lengths $(3a/2, b/2, c/2)$. Six remaining cases involve quadrilaterals divided into pairs of triangles: $KLPQ$ is divided into triangles KLQ and LPQ, $KMPN$ is divided triangles KMN and MNP, and $LMNQ$ is divided into triangles LMN and MNQ. Projections of each of these follows the same procedure therefore we describe it here only for one of the triangles, $4KLM$.

We compute the smallest angle that can occur from a projection of $4KLM$ onto a plane F that cuts off vertex A from the other vertices of the tetrahedron, and for which $4KLM$ is an elementary normal disk. The closure of the set of possible unit normal vectors for the plane F, oriented to point towards A, belongs to a spherical triangle T. Vertices of T are unit normal vectors~n_{ABC},~n_{ABD},~n_{ACD} to the faces ABC, ACD and ABD.

We compute the minimal and the maximal angles that can occur from a projection of $4KLM$ onto a plane F that cuts off vertex A from the other vertices of the tetrahedron, and for which $4KLM$ is an elementary normal disk. The closure of the set of possible unit normal vectors for the plane F, oriented to point towards A, belongs to a spherical triangle T. Vertices of T are unit normal vectors~n_{ABC},~n_{ABD},~n_{ACD} to the faces ABC, ACD and ABD.

The dihedral angles between $4KLM$ and its three adjacent faces are either 60° or 90°, and F can be nearly parallel to one of these faces. A projection of $4KLM$ to a nearly perpendicular plane can return a triangle with angles close to 0 or π, giving very poor angle bounds. Fortunately, the elimination of valence-four vertices in the GradNormal algorithm resolves this problem.

If the plane F is almost parallel to the face ABC and thus nearly perpendicular to $4KLM$, then F cuts off the vertex A from the other vertices of octahedron $ABCDB_2C_2$. This case results in a valencefour vertex in the MidNormal mesh. The GradNormal algorithm removes the vertex M in this case and thus avoids projecting 4KLM to a near perpendicular plane. The same will apply for planes with normals in a neighborhood of the vertex $\sim n_{ABC}$ of T. We now investigate exactly how T is truncated in the unit sphere when we eliminate planes for which MidNormal leads to valence-four vertices at M

Call a plane *allowable* if it separates vertex A from vertices B,C,D. Denote by X the closure of the set of unit normal vectors to allowable planes, oriented to point towards A. Then X forms a spherical triangle in the unit sphere with vertices$\sim n_{ABC}$,$\sim n_{ABD}$,$\sim n_{ACD}$. Inside X is a subset $Y \subset X$ corresponding to normals of allowable planes that separate A from the vertices B_2,C_2 of the octahedron. All normals to planes for which MidNormal gives valence-four vertices at M are in Y, but some of these are also normal to planes that lead to higher valence vertices at M. This leads us to define another subset $Z \subset Y$ whose points are in the closure of normals$\sim v$ with the property that if the normal to an allowable plane is in Z, then any parallel allowable plane separates A from vertices B,C,D,B_2,C_2. It can be seen from Figure 12 that a neighborhood of$\sim n_{ABC}$ in X lies in Z, so this set is non-empty. We now determine the precise shapes of Y and $Z \subset Y$ on the sphere, determining the configuration shown in Figure 16.

We first consider what points lie in$Y$. Planes normal to vectors in Y can be moved to a parallel allowable plane that separates A from vertices B_2,C_2,B,C,D. Any such plane can be pushed through parallel planes in Y towards A, until it hits A, since it separates A from the other five vertices. The boundary of the set of such planes containing A is a spherical quadrilateral with vertices corresponding to the normals to the four faces of the octahedron meeting A, namely

$\sim n_{ABC_2}$,$\sim n_{AB_2C_2}$ =$\sim n_{BCD}$,$\sim n_{ACB_2}$,$\sim n_{ABC}$. Then$Y$ consists of points inside the spherical quadrilateral with these four vertices, a subset of the spherical triangle X.

Next we consider what points lie in Z. An allowable plane normal to a vector in Z must separate A from B_2,C_2,B,C,D. This plane can be pushed away from A through parallel planes until it first hits one or more of the other five vertices. It cannot first hit D, as no allowable plane through D separates A from B_2,C_2,B,C.

This set of vertices that it hits must include some subset of B,C since if it hits only one or both of B_2,C_2 then a parallel plane in Xwould not separate A from vertices B_2,C_2,B,C,D and thus its normal would not lie in Z. We consider which sets of three or more vertices may be reached by planes in Z when these planes are translated away from A through parallel planes. These form some of the vertices of the spherical polygon Z. Note that the four vertices B_2,C_2,B,C are coplanar, and form one plane defining a vertex of Z. Thus this is the only vertex hit by pushing a plane in Z away from A. Other vertices are found by planes in Z that contain A and two or more additional vertices, giving vertices of Z at$\sim n_{ABC}$,$\sim n_{ABC_2}$,$\sim n_{AB_2C}$ (but not $\sim n_{AB_2C_2}$, a neighborhood of which lies in $Y - Z$). The resulting region $Z \subset Y$ is shown in Figure 16. It is the interior of the spherical quadrilateral formed by spherical geodesic arcs joining the four vertices$\sim n_{ABC}$,$\sim n_{ABC_2}$, $\vec{n}_{AB_2C}$, $\vec{n}_{BCB_2C_2}$.

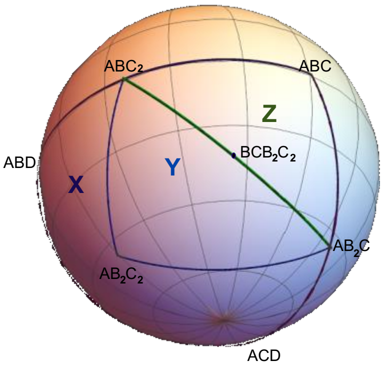


Figure 16: *The spherical triangle X consists of normals to planes separating vertex A from vertices B,C,D. The regionY ⊂X consists of directions for which at least one normal plane gives a vertex of valence-four at M. The region Z ⊂ Y consists of directions where* all *normal planes in X give a a vertex of valence-four at M.*

The region $X - Z$ is a spherical quadrilateral, since the vertices $\sim n_{ABC_2}$, $\sim n_{BCB_2C_2}$ and $\sim n_{AB_2C}$ lie on a single spherical geodesic. This holds for all a and follows from the fact that lines BC_2 and B_2C are parallel to a line of intersection of planes ABC_2 and AB_2C. Therefore unit normal vectors for planes BCB_2C_2, ABC_2 and AB_2C are coplanar. Moreover $X - Z$ is contained in a hemisphere, since all vectors in X have positive inner product with A.

Each vertex of the spherical quadrilateral $X - Z$ has distance at most $\pi/2$ from$\sim n_{KLM}$, as seen by computing dihedral angles of the faces of the tetrahedron τ. The maximum distance of a boundary point from $\sim n_{BCD}$ occurs at a vertex of $X - Z$, since $X - Z$ is a spherical polyhedron contained in a hemisphere. It follows that each boundary point of $X - Z$ has distance at most $\pi/2$ from

$\sim n_{KLM}$ =$\sim n_{KLM}$. Corollary 7.1 implies that extreme angles for the projection of 4KLM in the GradNormal algorithm are realized either by the triangle itself or by a projection to a plane with normal vector lying on one of the boundary edges of $X - Z$. There are three angles for 4KLM and four boundary edges of $X - Z$ determining planes onto which they can project. The three angle functions given by 4KLM when projected onto the arc from $\sim n_{ACD}$ to $\sim n_{AB_2C}$ are shown in Figure 17, as are angles along each of the other three arcs of $\partial(X - Z)$.

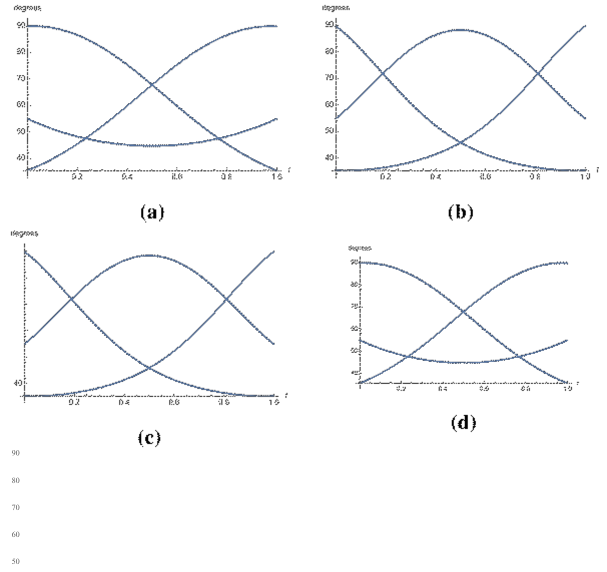


Figure 17: *Angles of* 4*KLM after projection onto an arc of* $\partial(X-Z)$ *running from (a) ~*n_{ACD} *to ~*$n_{AB2}C$*, (b) ~*n_{BCD} *to ~*$n_{AB2}C$*, (c) ~*n_{ABD} *to~*n_{ACD}*, and (d)~*n_{ABD} *to~*n_{ABC2}*. Graphs repeat due to symmetries. Again all angles are in* [35.25°, 101.45°]

We now consider projections of triangles after the removal of valence-four vertex.

Triangles 4*KNQ*,4*LNP* and 4*MPQ*, as well as the triangles coming from dividing elementary quadrilaterals along a diagonal, all give rise to similar angle functions for each edge of a corresponding quadrilateral spherical region. Altogether there are 12 triangles with 36 angles projecting to four edges each, or 144 angle functions in total, each defined on an interval of normal directions connecting two points on the sphere along a spherical arc. The union of all these angle functions is graphed in Figure 18.

This completes the proof of Theorem 1.1.

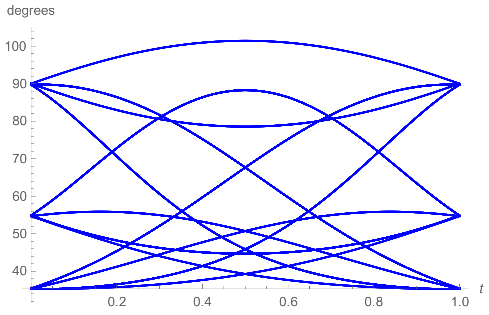


Figure 18: *Angles of all triangles in the GradNormal mesh are bounded above and below by the maximum and minimum values obtained in these graphs. A total of 144 angles are graphed over the boundary of spherical regions to produce these functions. Because of symmetries and coinciding functions, there are only 12 distinct graphs resulting from these 144 angles. All curves lie above* 35.25° *and below* 101.45°.