Partitioning Meshes with Lines and Planes

Feng Cao* John R. Gilbert† Shang-Hua Teng‡

September 19, 1995
(revised January 9, 1996)

Abstract

We investigate several geometric methods for dividing an irregular mesh into pieces of roughly equal size with few interconnecting edges. All these methods are based on cutting a mesh with a line (in two dimensions) or a hyperplane (in any dimension).

Line cuts have often been used in practice, but their quality varies widely. Until now, no theory has existed to predict the effectiveness of any line-cut algorithm. We make two main contributions: First, we give rigorous (and tight) bounds on the quality of line cuts for meshes of bounded aspect ratio in terms of a parameter we call *grading* that measures the nonuniformity of the mesh. Our bound on line cut quality differs from the known bound on circle cut quality by a factor that grows only as the $1/d$ power of the logarithm of the mesh grading. Second, we give an upper bound on the quality of an exact 50:50 line cut. This is the first proof of a cut size guarantee for any exact bisection induced by a single geometric cut.

We use these bounds to design simple and efficient algorithms that guarantee to find good line cuts.

Keywords. finite element mesh, geometric methods, load balancing, partitioning, parallel scientific computing.

AMS subject classifications. 05C50, 65F50, 68R10

*Department of Computer Science, University of Minnesota, Minneapolis, MN 55455 (cao@cs.umn.edu).

†Xerox Corporation, Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304 (gilbert@parc.xerox.com).

‡Department of Computer Science, University of Minnesota, Minneapolis, MN 55455 (steng@cs.umn.edu). Supported in part by an NSF CAREER award (CCR-9502540) and an Intel research grant. Part of this work was done while the author was visiting Xerox Palo Alto Research Center.

1 Introduction

Graph partitioning is one of the basic problems in parallel scientific simulation. Solving a large problem on a parallel computer with distributed memory usually requires that the data for the problem be partitioned somehow among the processors. The quality of the partition affects the speed of solution; a good partition divides the work up evenly and requires as little communication as possible.

Graphs from large-scale problems in scientific computing are often defined geometrically [13]. They are *meshes* of elements in d-dimensional Euclidean space (typically $d = 2$ or 3). Several partitioning algorithms exploit the geometric structure of a mesh [6, 9, 10, 13, 11, 18, 20, 22].

This paper presents theoretical, algorithmic, and experimental results on a family of geometric partitioning methods. These methods partition a d-dimensional mesh by a suitable hyperplane in d-space that divides the vertices into those above and below the hyperplane, respectively.

Our work is motivated by a theoretical study of Miller, Teng, Thurston, and Vavasis [13, 14, 20], which provides the mathematical foundation for a method of partitioning a d-dimensional mesh by finding a suitable sphere in d-space, and then dividing the vertices into those inside and outside the sphere. Miller et al. showed that if the mesh elements are *well-shaped* in a suitable sense, then there is a strong guarantee on the quality of the partition generated by the sphere-cut algorithm. In particular, for each d-dimensional well-shaped mesh of n vertices, the algorithm finds a partition that cuts at most $O(n^{1-1/d})$ edges. Gilbert, Miller, and Teng [9] refined this algorithm and experimented with it; they showed that it gives good partitions in practice for a wide variety of meshes.

What about line cuts? A line is a degenerate circle, and both the theoretical and practical versions of the Miller et al. algorithm sometimes produce a cutting line. Line cuts are used in practice in coordinate bisection [11] and in inertial bisection [22]. However, lines are not always good enough: Teng [20] gave an example of a class of well-shaped meshes for which every line cut has at least $\Omega(n)$ edges. In practice, circle cuts seem to be often (but not always) better than line cuts.

In this paper, we relate the quality of a line cut to the degree of grading or nonuniformity of a well-shaped mesh. We define a measure of grading of a mesh that we call *grading.* If the longest mesh edge has length M and the shortest has length m, we define the grading as

$$\Delta = 1 + \log_2(M/m).$$

We can think of Δ as the number of significant digits (or bits) necessary to resolve both the longest and shortest mesh edges. Teng's counterexample has $\Delta = \Omega(n)$; the ratio of the longest mesh edge to the shortest edge is exponential in n. However, meshes used in practice do not need nearly so many bits for coordinates—a large mesh might have tens of thousands of points but $\Delta < 20$.

We show that if the grading of a well-shaped mesh of n vertices in d dimensions is Δ, then there is a hyperplane that cuts $O(\Delta^{1/d} n^{1-1/d})$ edges and divides the mesh into two components of sizes in a ratio no worse than 1:d. We give two simple algorithms for finding such a hyperplane. Moreover, we construct an example to demonstrate that this bound is the best possible for the class of well-shaped meshes with bounded grading.

We also prove that a two-dimensional mesh of grading Δ always has a line that cuts only $O(\Delta^{1/2} n^{1/2} (\log n)^{1/2})$ edges, and divides the mesh into two pieces of exactly equal size (within one vertex). In d dimensions, this result generalizes to give a hyperplane that splits a mesh exactly 50:50 and cuts $O(\Delta^{1/d} n^{1-1/d} (\log n)^{1/d})$ edges. This is the first geometric partitioning method that gives a guarantee on the size of an exact 50:50 cut.

The rest of the paper is organized as follows. In Section 2 we review well-shaped meshes and make some other definitions. In Section 3, we survey some basic line and hyperplane based partitioning methods. In Sections 4 and 5, we present two randomized algorithms and show that they give $O(\Delta^{1/d} n^{1-1/d})$ partitions of well-shaped meshes, thus proving our main theorem. We also give a lower bound example to show that this result is the best possible for meshes with bounded grading. In Section 6, we show that there is a 50:50 line based separator for two dimensions and generalize the result to higher dimensions. In Section 7 we give some experimental results. In Section 8 we sum up, and discuss the question of when line cuts should be used.

2 Well-Shaped Meshes

One important aspect that distinguishes a finite-element or a finite-difference mesh from a "combinatorial" graph is that it comes with two structures: the combinatorial structure and the geometric structure. We represent a mesh by an ordered pair (G, xyz), where G describes the combinatorial structure (say, as a sparse adjacency matrix), and xyz gives the geometric coordinates.

2.1 Finite Element Meshes

A finite element mesh is a subdivision of a domain in d dimensions into a *mesh* of polyhedral *elements*; see Figure 1 for an example. A common type of element is a d-dimensional *simplex*, which is the convex hull of $d+1$ affinely independent points in $\mathbb{R}^d$—a triangle in two

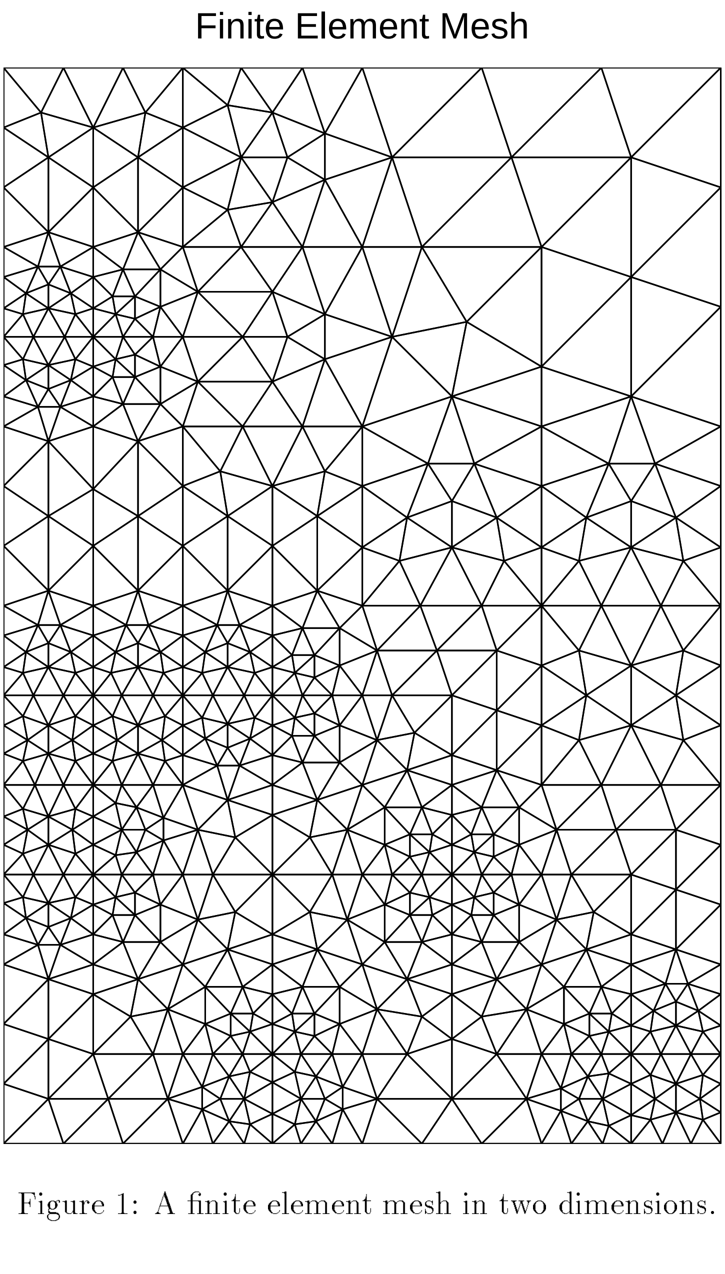

Figure 1: A finite element mesh in two dimensions.

dimensions or a tetrahedron in three dimensions. A d-dimensional *simplicial complex* is a collection of d-dimensional simplices that meet only at shared faces [2, 3]. For example, a 2-dimensional simplicial complex is a triangulation whose triangles intersect only at shared edges and vertices.

For most applications, a mesh is specified as a list of its elements, where each element is associated with a description of its lower dimensional structures such as faces, edges, and vertices. Each vertex is specified by its geometric coordinates. Associated with each simplicial complex is a natural graph called its *1-skeleton*. For example, the 1-skeleton of a two-dimensional simplicial complex is a planar graph.

2.2 Conditions on Element Shape

To ensure accuracy in the finite-element approximation, in addition to the conditions that a mesh must conform to the boundaries of the region and be fine enough, each individual element of the mesh must be *well shaped*. A common condition for finite element meshes is that the angles of each element should not be too small, or equivalently that the *aspect ratio* of each element should be bounded [2, 7]. Several other shape criteria exist. In some contexts, an upper bound on the largest angle is important [1]. Using Voronoi diagrams in the control volume method [15], the ratio of the radius of the circumsphere to the shortest edge of any element in the dual Delaunay triangulation must be bounded. (The latter condition is called *bounded radius aspect ratio.*)

For our purposes, we will consider a mesh well-shaped if it satisfies a *density condition* as defined by Miller and Vavasis [16]. Meshes with bounded element aspect ratio satisfy a density condition.

Let G be a graph and let xyz be an embedding of its vertices in $\mathbb{R}^d$. We say xyz is an embedding of *density* α if the following inequality holds for all vertices v in G. Let w be the farthest vertex from v that is connected to v by an edge. Let u be the vertex closest to v, whether connected by an edge or not. Then the α-density condition requires that the ratio of the distance between v and w to the distance between v and u be at most α, for all v. In general, G is an *α-density graph* in $\mathbb{R}^d$ if there exists an embedding of G in $\mathbb{R}^d$ with density α.

It is easy to show that a density condition implies a bound on vertex degree: there is a $C(\alpha, d)$ depending only on α and d such that the maximum degree of an α-density graph is bounded by $C(\alpha, d)$.

In this paper, if a graph G is an α-density graph under the embedding xyz, then we say the mesh (G, xyz) is an *α-well-shaped mesh*. Both finite element meshes with bounded aspect ratio, and Delaunay triangulations with bounded radius aspect ratio, are well shaped by this definition [14, 15].

A well-shaped mesh can contain both regions of high refinement (or closely spaced points) and low refinement (or widely spaced points); its grading Δ can be arbitrarily large. However, we can define a sort of "local feature size" at each mesh point that captures the level of refinement "nearby" that point. To be precise, suppose $M = (G, xyz)$ is an α-density mesh in $\mathbb{R}^d$. For each point $p \in xyz$, let the *feature* of p, denoted by $h(p)$, be the distance from p to its nearest Euclidean neighbor in xyz.

We will also make use of a special set of balls centered at the mesh vertices, as follows. Let B_p be a ball centered at p of radius $\alpha h(p)$. Let $\Gamma = \{B_p : p \in xyz\}$. We will call Γ the *α-nearest neighborhood system* (or α-NNS) of M.

2.3 Partitions, Bisections, and Separators

Now we define three related types of cuts in a graph or mesh.

A *bisection* of a graph with n vertices is a division of its vertices into two disjoint subsets V_L and V_R of exactly equal size. (If n is odd, we could allow the sizes to differ by one; however, for simplicity we will consider only bisections in graphs with an even number of vertices). The *cut size* of a bisection is the number of edges with one endpoint in V_L and the other in V_R.

For each constant δ such that $1/2 \leq \delta \leq 1$, a *δ-bisection* is a partition of the vertices of G into two disjoint subsets V_L and V_R such that both $|V_L| \leq \delta n$ and $|V_R| \leq \delta n$. The *cut size* of a δ-bisection is the number of edges between V_L and V_R. By definition, a bisection is a 1/2-bisection.

A bisection or a δ-bisection is often called an *edge separator*. A related concept is a *vertex separator*, which is a set of vertices whose removal disconnects the graph. In particular, a subset C of vertices of an n-vertex graph G is an *$f(n)$-separator* that *δ-splits* G if $|C| \leq f(n)$ and the vertices of $G - C$ can be partitioned into two sets V_L and V_R such that there are no edges from V_L to V_R and $|V_L|, |V_R| \leq \delta n$. For example, Lipton and Tarjan's planar separator theorem [12] says that every planar graph has an $O(\sqrt{n})$-separator that 2/3-splits it.

Edge separators are useful for partitioning meshes and sparse matrices among the processors of a parallel computer, particularly for explicit time-stepping methods and for iterative linear system solvers like conjugate gradients. Vertex separators have been used in direct methods such as nested dissection for solving sparse linear systems [8, 12].

Miller, Teng, Thurston, and Vavasis [14] showed that well-shaped meshes in d dimensions have $O(n^{1-1/d})$-separators. In a sense, this gives a natural extension of the planar separator theorem to higher dimensions.

Theorem 2.1 (Sphere Separator Theorem) *Let M be an α-well-shaped mesh in d dimensions. Then M has an*

$$O\left(\alpha \cdot n^{1-1/d}\right)\text{-vertex separator}$$

that $(d+1)/(d+2)$-splits.

Because well-shaped meshes have bounded vertex degree, any $f(n)$-vertex separator that δ-splits an α-well-shaped mesh implies a δ-bisection of cut size $O\left(\alpha^d \cdot f(n)\right)$.

The separator in this result in induced by a circle cut in the graph. Our concern in this paper is to show exactly when a similar result holds for line cuts.

3 Centerpoint-Based Linear Partition

The simplest and perhaps fastest geometric approach to mesh partitioning is *coordinate bisection.* Given a mesh $M = (G, xyz)$ in d dimensions, it examines *d median hyperplanes*, one orthogonal to each coordinate axis. Each median hyperplane divides the mesh into two parts of equal size. The algorithm simply returns the median hyperplane that cuts the smallest number of edges. We can recursively apply coordinate bisection to divide the mesh into more than two pieces.

As a variant, we can relax the 50:50 balance condition. Instead of choosing a median hyperplane, we can choose the best hyperplane normal to the axis that at least δ-splits the mesh, for some choice of δ. We will call this variant *balanced coordinate partition.* This is essentially the partitioning algorithm used by Heath and Raghavan [11].

Coordinate bisection or balanced coordinate partition does not come with a guarantee on the quality of the partition it computes. We now discuss a family of more robust relatives of coordinate bisection, which we will call *linear partitions.*

3.1 Random Balanced Linear Partition

A simple improvement of coordinate bisection is to use hyperplanes that are not axis aligned. The first variant randomly chooses a direction, projects the mesh onto this direction, and returns the hyperplane that passes through the median of the projected coordinates. We can choose a few random median hyperplanes and return the best one. We will call this method *random linear bisection.*

By relaxing the 50:50 balance condition in the random linear bisection scheme, we obtain a scheme that will be referred as *random balanced linear partition*. This partitioning method selects a random direction, projects the mesh onto that direction, and returns the best δ-balanced hyperplane that is normal to the chosen direction. We will analyze this scheme in Section 4 below; first, however, we must describe and analyze a more complicated linear partition scheme.

3.2 Linear Partition Around a Centerpoint

Our first linear partitioning scheme is motivated by the sphere separator construction of Miller et al. [13]. A *centerpoint*, which is a sort of generalized median, plays a key role in that construction; we now review its definition.

Suppose P is a finite set of points in $\mathbb{R}^d$. A hyperplane H in $\mathbb{R}^d$ divides P into three subsets: $P^+ = H^+ \cap P$, $P^- = H^- \cap P$, and $P \cap H$. The *splitting ratio* of H over P, denoted by $\phi_H(P)$, is defined as

$$\phi_H(P) = \max\left(\frac{|P^+|}{|P|}, \frac{|P^-|}{|P|}\right)$$

For each $0 < \delta < 1$, a point $c \in \mathbb{R}^d$ is a δ-*centerpoint* of P if every hyperplane containing c δ-splits P. A $d/(d+1)$-centerpoint is called a *centerpoint* of P. Every finite point set in $\mathbb{R}^d$ has a centerpoint, which can be found by linear programming [5, Section 4]. An efficient and practical constant-time algorithm to approximate a centerpoint can be found in [4]; see [9] for an implementation.

We now describe a linear partitioning scheme that uses a centerpoint.

Algorithm *Centerpoint-Based Linear Partition*
Input A well-shaped mesh $M = (G, xyz)$
Output A δ-balanced hyperplane H

1. Find a δ-centerpoint c of xyz;

2. Return a random hyperplane H that contains c.

In practice, we can examine several random hyperplanes through the centerpoint and return the best one.

3.3 From Planes to Separators

For consistency and to be able to compare our results with the sphere separator theorem of Miller et al., we will state all the separator theorems in this paper in the vertex-separator form. We now show how to derive a vertex separator from a hyperplane.

Let $M = (G, xyz)$ be an α-well-shaped mesh, and H be a hyperplane that does not contain any mesh vertex. There is a natural "edge partition" induced by H as follows. The hyperplane H divides the mesh vertices into two subsets, $H^+ \cap M$ and $H^- \cap M$, one for each side of the hyperplane. If H δ-splits xyz, then we obtain a δ-bisection. The cut size is then equal to the number of edges of M that H intersects.

We can also obtain a vertex separator from a hyperplane: Let $\Gamma = \{B_1, .., B_n\}$ be the α-NNS of M. Let V be the set of vertices of M. Let C be the set of vertices whose α-ball in the NNS intersects H. Throughout this paper, we will refer to this set of vertices as the vertex-separator of H. Let $V_L = H^+ \cap V$ and $V_R = H^- \cap V$ be the partition induced by H after removing C.

Lemma 3.1 *If H δ-splits M then C is a δ-splitting vertex separator that divides the vertices of M into V_L and V_R.*

Proof: To see why C is a vertex separator, we need to show for each pair $j \in V_L$ and $k \in V_R$ that (j, k) is not an edge in M. For the sake of contradiction, assume that (i, j) is an edge of M. Then B_j contains vertex k and hence B_j must intersect H. This contradiction completes the proof. $\square$

Another way to get a vertex separator from an edge separator, which is often used in practice [9], is as follows. Given an edge separator, consider the bipartite subgraph of G consisting of the separator edges and their endpoints. Any vertex cover of this bipartite graph is a vertex separator; the smallest vertex cover can be found by bipartite matching.

4 A Separator Theorem

In this section, we analyze the average size of the vertex separator from by the centerpoint-based linear partitioning scheme. This analysis will yield the following separator theorem.

Theorem 4.1 (Linear Separator Theorem) *Let M be an α-well-shaped mesh in d dimensions. Let Δ be the grading of M. Then the centerpoint-based linear partition finds a $d/(d+1)$-balanced hyperplane that induces a vertex separator of size $O\left(\alpha \Delta^{1/d} n^{1-1/d}\right)$.*

For meshes with bounded Δ, this is the same order of magnitude as the Miller et al. sphere separator result. In general, this says that line cuts can be worse than circle cuts by a factor that grows only as the $1/d$ power of the logarithm of the mesh grading.

4.1 Concave Functions and the Volume Argument

In several proofs in the subsequent sections, we will use the following *concave function optimization argument*. Fix α with $0 < \alpha < 1$. Consider the *concave sum* $\sum_{i=1}^{n} x_i^{\alpha}$, where $x_1, x_2, \ldots, x_n$ are positive reals that add up to 1. The sum is maximized when $x_1 = x_2 = \cdots = x_n = 1/n$. Therefore, the maximum value of $\sum_{i=1}^{n} x_i^{\alpha}$, subject to $\sum_{i=1}^{n} x_i = 1$, is $n^{1-\alpha}$.

For example, let $y_1, y_2, \cdots, y_n$ be positive reals with $\sum_{i=1}^{n} y_i^d = a$. Then $\sum_{i=1}^{n} y$ is maximized when $y_1 = y_2 = \cdots = y_n = (a/n)^{1/d}$. Therefore, the maximum value of $\sum_{i=1}^{n} y_i$, subject to $\sum_{i=1}^{n} x_i^d = 1$, is $a^{1/d} n^{1-1/d}$.

We will also use the following *volume argument*. If P is a set of points in d dimensions whose pairwise distance is at least 1, then any ball in $\mathbb{R}^d$ of volume V contains at most $O(V)$ points of P. (This is also true if we replace "ball" with any cell of bounded aspect ratio.)

4.2 Local Graph Decomposition

In this section, we give a high-level discussion of our proof strategy for Theorem 4.1. The basic idea is to decompose an α-well-shaped mesh M with n vertices into Δ sub-meshes $M_1, ..., M_\Delta$ such that

1. Every edge of M is an edge of at least one M_i;
2. $\sum_{i=1}^{\Delta} n_i = 2n$, where n_i is the number of vertices of M_i; and
3. For each $1 \leq i \leq \Delta$, the average separator size for M_i induced by a random hyperplane passing through the centerpoint c is $O(\alpha n_i^{1-1/d})$.

Such a decomposition ensures that the average separator size for M by a random hyperplane passing through c is bounded by $O(\sum_{i=1}^{\Delta} \alpha n_i^{1-1/d})$, subject to the condition that $\sum_{i=1}^{\Delta} n_i = 2n$. Because $\sum_{i=1}^{\Delta} n_i^{1-1/d}$ is a concave sum, it is maximized when $n_1 = \cdots = n_\Delta = 2n/\Delta$. Therefore, the average separator size by a random hyperplane containing c is bounded by

$$\begin{aligned}\sum_{i=1}^{\Delta} \alpha n_i^{1-1/d} &= O\left(\Delta\alpha(n/\Delta)^{1-1/d}\right) \\ &= O\left(\Delta^{1/d}\alpha n^{1-1/d}\right).\end{aligned}$$

Our decomposition is based on *local graphs*, first introduced by Vavasis [21]. Define the *smallest feature* of a mesh M to be the shortest pairwise Euclidean distance between its vertices. A mesh M is an *α-local graph* if the ratio of the longest edge to the smallest feature of M is bounded by α. Vavasis showed that any α-local graph has a balanced vertex separator of size $O(\alpha n^{1-1/d})$.

Now let $M = (G, xyz)$ be an α-well-shaped mesh of grading Δ. For each vertex p in xyz, define the *feature* of p, denoted $h(p)$, as the distance between p and its nearest Euclidean neighbor in xyz. Because M is α-well-shaped, the length of each edge incident on p is at least $h(p)$ but no more than $\alpha h(p)$.

Let h be the smallest feature of any vertex of M. We partition the points xyz into Δ subsets $P_1, ..., P_\Delta$ according to their features. Let $\beta = \max(\alpha, 2)$. Let P_i be the set of points in xyz whose features are at least $\beta^{i-1}h$ but no more than $\beta^i h$. Because the grading of M is Δ, we have $\cup P_i = xyz$. Note that any individual P_i may be an empty set.

For each $1 \leq i \leq \Delta$, let M_i be the mesh whose vertices are $P_{i-1} \cup P_i$ and whose edges are those edges in M that have length at least $\beta^{i-1}h$ but at most $\beta^i h$. Let n_i be the number of vertices in M_i. Clearly

- $\cup_{i=1}^{\Delta} M_i = M$, because each edge of M is in one of the submeshes;
- $\sum n_i \leq 2n$, because each vertex appears in at most two consecutive submeshes; and
- M_i is an α-local graph for all $i : 1 \leq i \leq \Delta$.

To establish Theorem 4.1, it suffices to prove the following lemma.

Lemma 4.2 *Let $M = (G, xyz)$ be an α-local graph in $\mathbb{R}^d$, let Γ be its α-NNS, and let c be a point in $\mathbb{R}^d$. Then the average number of balls from Γ that intersect a random hyperplane through c is bounded by $O(\alpha n^{1-1/d})$.*

The next subsection contains the proof.

4.3 A Continuous Proof

We will use the following geometric characterization of an α-local graph $M = (G, xyz)$. Let h be the smallest feature of M. Without loss of generality, we may scale M so that $h = 1$. Let $\Gamma = \{B_p : p \in xyz\}$ be the α-NNS of M. The following lemma is immediate from the definition of an NNS.

Lemma 4.3 *If (p, q) is an edge of M, then ball B_p contains q and ball B_q contains p. Therefore, if a hyperplane H cuts edge (p, q), then H intersects both B_p and B_q.*

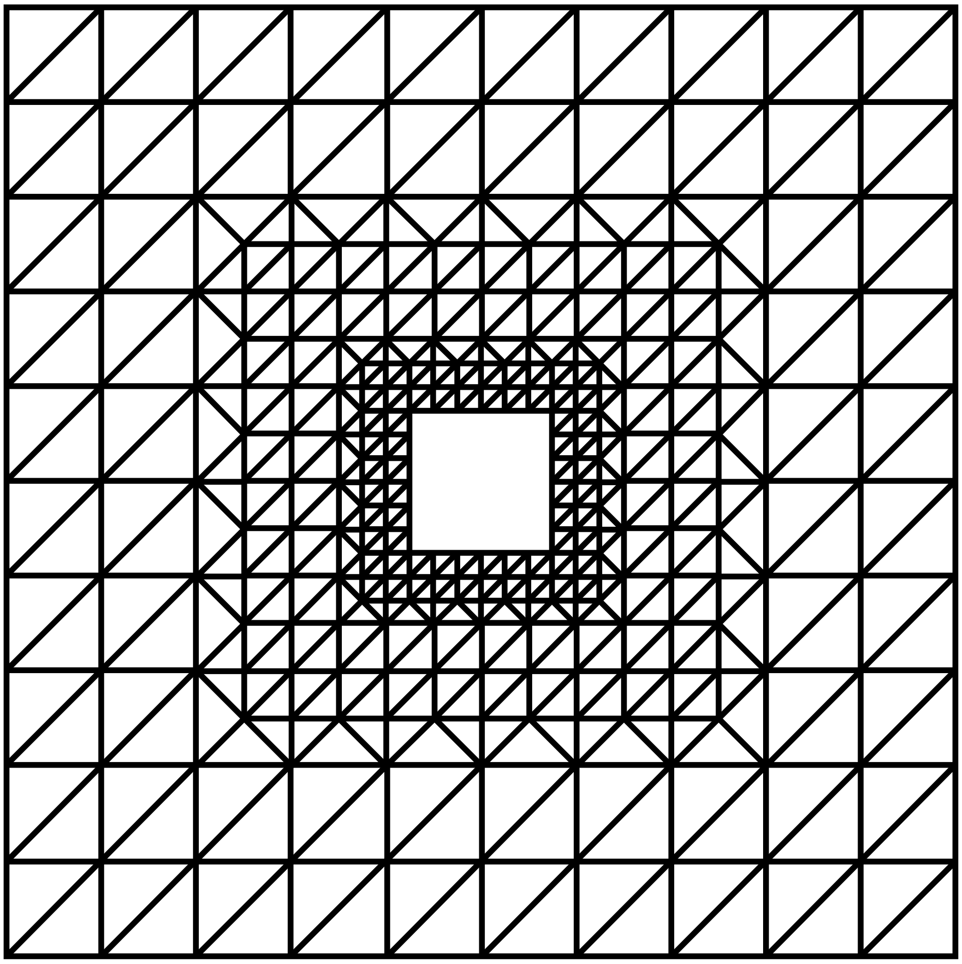

Figure 2: The lower bound example in two dimensions, with $n = 288$, $\Delta = 3.5$, $s = 3$, $k = 11$.

To prove of Lemma 4.2, we will estimate the average number of balls that intersect a random hyperplane through a point $c \in \mathbb{R}^d$.

Proof: (**Lemma 4.2**): Let S_i be the sphere of radius i centered at c. Let $S = S_1$, the sphere of unit radius. It follows from the volume argument that the number of balls from Γ that intersect S is bounded by $O(\alpha^d)$. Moreover, the centers of these balls are inside $S_{\lceil \alpha \rceil}$.

It remains to count the balls that do not intersect S. Abusing notation slightly, we will now refer to Γ as the set of such balls.

For each ball $B_p \in \Gamma$, let C_p be the cap obtained by projecting B_p onto S through the center of S. (A *cap* on a sphere is the intersection of a halfspace with the sphere.) Let $\Phi = \{C_p : p \in xyz\}$ be the set of caps. Let

$$Area(\Phi) = \sum_{p \in xyz} Area(C_p).$$

We now show that $Area(\Phi) = O(\alpha^{d-1} n^{1/d})$. Let Γ_i be the set of balls whose centers lie between S_i and S_{i+1}. Let Φ_i be the set of caps from balls in Γ_i. Let n_i be the number of balls in Γ_i. By the volume argument, there is a constant a_1 depending only on the dimension such that $n_i \leq a_1 i^{d-1}$. Furthermore, there is a constant a_2 such that, for every ball $B_p \in \Gamma_i$, the area of its cap C_p is bounded by $a_2(\alpha/i)^{d-1}$. Therefore,

$$\begin{aligned} Area(\Phi) &= \sum_{p \in xyz} Area(C_p) \\ &= \sum_{i=1}^{+\infty} n_i a_2 (\alpha/i)^{d-1}, \end{aligned}$$

where $\sum_{i=1}^{\infty} n_i = n$ and $n_i \geq 0$.

Letting k be the largest integer such that $\sum_{j=1}^{k} a_1 j^{d-1} \leq n$, we maximize $Area(\Phi)$ as a function of n_i when

$$n_i = \begin{cases} a_1 i^{d-1} & i \leq k \\ n - \sum_{j=1}^{i-1} a_1 j^{d-1} & i = k+1 \\ 0 & i > k. \end{cases}$$

It is not hard to see that $k = O(n^{1/d})$. Therefore $Area(\Phi) = O(\alpha^{d-1} n^{1/d})$.

Lemma 4.2 is now a direct consequence of the following theorem of Miller et al. [14]. We will be more specific after stating the theorem.

Theorem 4.4 *Let $\Phi = \{C_1, \ldots, C_n\}$ be a set of caps on a unit sphere S in $\mathbb{R}^{d+1}$. Let A_d be the surface area of the unit sphere. If $Area(\Phi) \leq kA_d$, then the number of caps that intersect a random great circle of S is bounded by $O(k^{1/d} n^{1-1/d})$.*

We now complete the proof of Lemma 4.2. Note that a hyperplane H intersects a ball B_p iff the great circle defined by the intersection of H and S intersects C_p. It follows from Theorem 4.4 that the average number of balls in Γ that intersect a random hyperplane through c is bounded by

$$O\left((\alpha^{d-1} n^{1/d})^{1/(d-1)} n^{1-1/(d-1)}\right) = O(\alpha n^{1-1/d}).$$

This also completes the proof of Theorem 4.1. $\square$

4.4 A Lower Bound

We now construct a class of well-shaped meshes to show that Theorem 4.1 is the strongest possible result for all n and Δ in all dimensions d.

Let d, n, and Δ be given. Adjust n and Δ by constant factors if necessary so that both $s = \Delta - \log_2 d/2$ and $k = (n/s)^{1/d} \cdot 11/(11^d - 5^d)^{1/d}$ are integers.

Construct a mesh M_k as follows: Begin with a regular cubic mesh, k vertices on each side. Remove the centermost $(5k/11)^d$ vertices, leaving a hollow cubic shell with sides $3k/11$ vertices thick. Now divide each unit cube in the shell into $d!$ right simplices.

Finally, let mesh M be s copies of M_k nested within each other, each with its unit cell of dimension half that of its immediately containing M_k, with the space between copies of M_k triangulated with right simplices.

The resulting mesh has n vertices (by a straightforward computation). Its grading is $\Delta = s + \log_2 d/2$ because the ratio of the longest and shortest edges of a single right simplex is $d^{1/2}$, and the ratio of the linear dimensions of the largest and smallest simplices is 2^{s-1}. The mesh is well-shaped because its elements are all right simplices.

Any 50:50 line or plane cut for M must cut from the outer to the inner face of at least $s/2$ copies of M_k, at a cost of $\Omega(k^{d-1})$ vertices each. Thus the entire separator must have size at least $\Omega(sk^{d-1}) = \Omega(\Delta(n/\Delta)^{1-1/d}) = \Omega(\Delta^{1/d} n^{1-1/d})$.

Figure 2 shows the mesh for $d = 2$, $\Delta = 3.5$, and $n = 288$ (so that $s = 3$ and $k = 11$).

5 Analysis of Random Balanced Linear Partition

Our upper bound on the centerpoint-based linear scheme also implies a bound on the (simpler) random balanced linear partition scheme, provided the balance ratio is at least $d/(d+1)$. We first review the random balanced linear scheme.

Algorithm *Random Balanced Linear Partition*
Input A well-shaped mesh $M = (G, xyz)$
Output A δ-balanced hyperplane H

1. Find a random vector v in $\mathbb{R}^d$.

2. Return the hyperplane normal to v that has the smallest separator size among all δ-balanced hyperplanes normal to v.

Corollary 5.1 *Fix a balance factor $\delta \geq d/(d+1)$. Let $M = (G, xyz)$ be an α-well-shaped mesh in d dimensions, and let Δ be the grading of M. Then the random balanced linear scheme finds a δ-balanced hyperplane that induces a vertex separator of size $O\left(\alpha \Delta^{1/d} n^{1-1/d}\right)$.*

Proof: Let c be a δ-centerpoint of xyz. For each direction v, let H_0 be the hyperplane normal to v that contains c. Let H be the hyperplane returned by the random balanced linear scheme. By the definition of the random balanced linear scheme, the separator induced by H is no larger than that induced by H_0. In other words, for each direction, the random balanced linear scheme finds a hyperplane that is as good as the hyperplane found by the centerpoint based linear scheme for the same direction. Thus Corollary 5.1 follows directly from Theorem 4.1. $\square$

6 Bisections and 50:50 Separators

Most separator theorems guarantee a split that is balanced within some constant δ. Most separator heuristics provide an exact 50:50 bisection. Can we prove anything about 50:50 bisections?

If an $O(n^\lambda)$-separator theorem holds for a graph and its subgraphs, then a 50:50 separator of size $O(n^\lambda)$ (with a larger constant in the O) can be found by separating repeatedly. However, this is unattractive in the geometric setting because the final separator is induced by $O(\log n)$ different cuts.

The sphere-based geometric partitioner of Miller et al. [14] guarantees that the partition is at least $1 : (d+1)$ balanced, that is, the larger of the two submeshes is no more than $((d+1)/(d+2))n$. Gilbert, Miller, and Teng [9] describe a heuristic modification to that algorithm that produces a 50:50 balanced sphere partition; however, although its performance is good in practice, the guarantee on cut size is lost. It remains open whether every well-shaped mesh has a 50:50 sphere-based bisection that has a cut size of $O(n^{1-1/d})$.

In this section, we show that a 50:50 line-based partition exists for all well-shaped meshes with a bounded grading. We will prove the following theorem.

Theorem 6.1 *Let M be an α-well-shaped mesh in d dimensions. Let Δ be the grading of M. Then there is a hyperplane that 50:50 splits M and induces a $1/2$-vertex separator of size*

$$O(\alpha \Delta^{1/d} n^{1-1/d} (\log n)^{1/d}).$$

As in Theorem 4.1, the separator size grows only as the $1/d$ power of the logarithm of the mesh grading. However, our bound is larger than that in Theorem 4.1 by a factor of $(\log n)^{1/d}$.

6.1 The 50:50 Algorithm

The algorithm that we will analyze is the random linear bisection scheme. We will show that it finds a hyperplane that satisfies the condition of Theorem 6.1. We first review this scheme.

Algorithm *Random Linear Bisection*
Input: a well-shaped mesh $M = (G, xyz)$ with grading Δ.

1. Choose a random direction v;

2. Return the hyperplane normal to v that evenly splits xyz.

Again, in practice, one could examine several random directions and return the best normal hyperplane.

6.2 An Analysis for Two Dimensions

The two-dimensional version of Theorem 6.1 is the following.

Theorem 6.2 *Let M be an α-well-shaped mesh in 2 dimensions. Let Δ be the grading of M. Then there is a line that 50:50 splits M and induces a $1/2$-vertex separator of size*

$$O(\alpha \Delta^{1/2} n^{1/2} (\log n)^{1/2}).$$

Just as in Section 4.2, we can reduce Theorem 6.2 to the case of constant Δ. Thus it suffices to prove the following lemma.

Lemma 6.3 *Let M be an α-local mesh in 2 dimensions, and let Γ be its α-NNG. The expected number of balls of Γ that intersect a random median line is $O(\alpha n^{1/2} (\log n)^{1/2})$.*

Henceforth, we take $M = (G, xyz)$ to be an α-local graph in $\mathbb{R}^2$ with n vertices, and we take $\Gamma = \{B_1, ..., B_n\}$ to be its α-NNS.

We will start by analyzing, not a random line, but a collection of $\sqrt{n}$ lines whose angles with the x axis are evenly spaced. Let $k = \lceil \sqrt{n} \rceil$, and let θ be the angle π/k. For each $0 \le i < k$, let L_i be some line that forms an angle of $i\theta$ with the x-axis.

To simplify the subsequent discussion, we define the following bipartite graph BL between the balls of Γ and the lines $\{L_0, ..., L_{k-1}\}$. Edge (B_i, L_j) is present in BL iff L_j intersects B_i. Let ℓ_j be the number of balls that L_j intersects (the degree of L_j in BL). Let b_i be the number of lines that B_i intersects (the degree of B_j in BL). Clearly, $\sum_{i=1}^{n} b_i = \sum_{j=0}^{k-1} \ell_j$.

We will use the following lemma for 2D α-local graphs.

Lemma 6.4 *Let BL, b_j, and ℓ_j be as defined above. Then*

$$\sum_{j=0}^{k-1} \ell_j = \sum_{i=1}^{n} b_i = O(\alpha n (\log n)^{1/2}).$$

Proof: We scale M so that its smallest feature is 1. Let $c_{i,j}$ be the number of balls that intersect both L_i and L_j, for $0 \le i < j < k$. We now give an upper bound on $c_{i,j}$. By symmetry, we only need to bound $c_{0,j}$ for all $1 \le j < k$.

Let $p_{0,j}$ be the intersection of L_0 and L_j. Let B be a ball in Γ that intersects both L_0 and L_j. Let $\delta_j = \min((\pi - j\theta)/2, j\theta/2)$. Because the radius of B is at least α, the distance from the center of B to $p_{0,j}$ is at most $\alpha/\sin(\delta_j)$.

Since $0 < \delta_j \le \pi/4$, we have $\sin(\delta_j) \ge 2\delta_j/\pi$. Thus $\alpha/\sin(\delta_j) \le (\alpha\pi)/(2\delta_j)$.

It follows from the volume argument and the fact that M is an α-local graph that $c_{0,j} = O(\alpha^2/\delta_j)$, which implies that $\sum_{j=1}^{k-1} c_{0,j} = O(\alpha^2 k \log k)$. Therefore

$$\sum_{0 \le i < j < k} c_{i,j} = \sum_{i=0}^{k-1} \sum_{j \ne i} c_{i,j} = O(\alpha^2 k^2 \log k) = O(\alpha^2 n \log n).$$

For each i, if $b_i \ge 2$, then b_i contributes $b_i(b_i - 1)/2$ to the sum $\sum_{0 \le i < j < k} c_{i,j}$. By the definition of $c_{i,j}$ and b_i, we have $\sum_{i=1}^{n} b_i(b_i - 1)/2 = (\sum_{0 \le i < j < k} c_{i,j}) + c_1$, where c_1 is the number of balls that intersect exactly one line. Since $c_1 \le n$, this implies $\sum_{i=1}^{n} b_i^2 = O(n + \alpha^2 n \log n)$. By the concave function optimization argument, we therefore have

$$\sum_{i=1}^{n} b_i = O(n + \alpha n \sqrt{\log n}).$$

$\square$

We now prove Lemma 6.3 using Lemma 6.4. Let L_i be the median line that makes an angle of $i\theta$ with the x-axis, for $0 \le i < k = \lceil \sqrt{n} \rceil$. By Lemma 6.4 the average number of balls of Γ that intersect a random line from $\{L_0, \ldots, L_{k-1}\}$ is $O(\alpha\sqrt{n}\sqrt{\log n})$. This implies that the average number of balls of Γ that intersect a random median line is also $O(\alpha\sqrt{n}\sqrt{\log n})$, which completes the proof of Lemma 6.3 and also Theorem 6.2.

Note that this analysis leads to both a (constant-time) randomized algorithm, and a deterministic algorithm that simply examines all $\lceil \sqrt{n} \rceil$ evenly spaced median lines and returns the one with the smallest cut size. Since at least one of those lines must have a cut no larger than average, the algorithm guarantees to find a separator of size $O(\alpha\Delta^{1/2}n^{1/2}(\log n)^{1/2})$.

6.3 Extension to Higher Dimensions

Finally, we sketch the proof of Theorem 6.1 for general dimension d.

Suppose $M = (G, xyz)$ is an α-local mesh in $\mathbb{R}^d$ with n vertices. Let $\Gamma = \{B_1, \ldots, B_n\}$ be the α-NNS of M.

Let $k = \lceil n^{1/d} \rceil$. Let $\{\nu_0, ..., \nu_{k-1}\}$ be k "uniformly spaced" points on a unit $(d-1)$-sphere with center o in $\mathbb{R}^d$. By "uniformly spaced", we mean that (1) the nearest neighbor of each point is at distance on the order of $k^{-1/(d-1)}$, and (2) for any d points $\nu_{i_1}, \ldots, \nu_{i_d}$, the solid angle formed by the radii $o\nu_{i_1}, \ldots, o\nu_{i_d}$ measures at least $\Omega(k^{-1/(d-1)})$. The second condition is a stronger version of the condition that no d points are co-hyperplanar with the center o. Let H_i be any hyperplane that is normal to ν_i.

Again, we define a bipartite graph BH between balls Γ and hyperplanes $\{H_0, ..., H_{k-1}\}$. Edge (B_i, H_j) is present in BH iff H_j intersects B_i. Let h_j be the number of balls that H_j intersects. Let b_i be the number of hyperplanes that B_i intersects. Clearly, $\sum_{i=1}^{n} b_i = \sum_{j=0}^{k-1} h_j$.

Lemma 6.5 *Let BH, b_j, and h_j be as defined above.*

$$\sum_{j=0}^{k-1} h_j = \sum_{i=1}^{n} b_i = O(\alpha n(\log n)^{1/d}).$$

Proof: (Sketch) We scale M so that its smallest feature is 1. Let $c_{i_1,i_2,\ldots,i_d}$ be the number of balls that simultaneously intersect $H_{i_1}, \ldots, H_{i_d}$, for $0 \le i_1 < \cdots < i_d < k$.

Let us fix $i_1, \ldots, i_{d-1}$ such that $0 \le i_1 < \cdots < i_{d-1} < k$, and vary i_d over the remaining points on the sphere (that is, $0 \le i_d < k$ and $i_d \ne i_j$ for any $1 \le j \le d-1$). The intersection of $H_{i_1}, \ldots, H_{i_{d-1}}$ is some line L. As i_d changes, the angle of L with H_{i_d} changes accordingly. If the angle between ν_{i_d} and L is $\pi/2 - \theta$, then the angle between L and H_{i_d} is θ. Hence the volume argument (as in the proof of Lemma 6.4) shows that $c_{i_1,i_2,\ldots,i_d} \le$

$\alpha^d/\sin(\theta/2) = O(\alpha^d/\theta)$. Because ν_{i_d} is roughly uniformly distributed on the unit sphere, for each i, $(0 \le i \le k^{1/(d-1)})$, there is a constant a such that there are at most $ak^{1-1/(d-1)}$ such hyperplanes that form an angle with L between $i\pi/(2k^{1/(d-1)})$ and $(i+1)\pi/(2k^{1/(d-1)})$. Therefore

$$\begin{aligned}\sum_{i_d \ne i_j, 1\le j\le d-1} c_{i_1,i_2,\ldots,i_d} &= \sum_{i=1}^{k^{1/(d-1)}} O(a\alpha^d k^{1-1/(d-1)} k^{1/(d-1)}/i) \\ &= \alpha^d k \log k.\end{aligned}$$

Therefore

$$\sum_{0\le i_1<\ldots<i_d<k} c_{i_1,\ldots,i_d} = O(\alpha^d k^d \log k) = O(\alpha^d n \log n).$$

For each i, if $b_i \ge d$, then b_i contributes $\binom{b_i}{d}$ to the sum $\sum_{0\le i_1<\ldots<i_d<k} c_{i_1,\ldots,i_d}$. By the definition of $c_{i_1,\ldots,i_d}$ and b_i, we have

$$\sum_{i=1}^{n} \binom{b_i}{d} \le \left(\sum_{0\le i,j<k} c_{i_1,\ldots,i_d}\right) + (d-1)n,$$

where $(d-1)n$ is an upper bound on the number of intersections of balls that intersect at most $d-1$ hyperplanes. Thus $\sum_{i=1}^{n} b_i^d = O(\alpha^d n \log n)$. We would like to have an upper bound on $\sum_{j=0}^{k-1} h_j = \sum_{i=1}^{n} b_i$. By the concave function optimization argument, we have

$$\sum_{i=1}^{n} b_i = O(\alpha n (\log n)^{1/d}).$$

□

Theorem 6.1 then follows from Lemma 6.5.

7 Experiments

As an illustration of the quality of various sorts of line and circle cuts, we present results from some preliminary experiments. Table 1 describes the meshes we used; see Gilbert et al. [9] for more details. Table 2 gives the value of the grading for each mesh, and the size of the cut found by each of five methods:

- Random 50:50 balanced line cut, using $n^{1/d}$ randomly chosen lines or planes.
- Deterministic 50:50 balanced line cut, using $n^{1/2}$ evenly spaced lines (for 2-D meshes only).

Mesh	Description	Mesh Type	Vertices	Edges
EPPSTEIN	Rectangular region	2-D triangles, lightly graded	547	1566
TAPIR	Cartoon animal	2-D triangles, sharp angles	1024	2846
PARC	Square with letters cut out	2-D triangles	1240	3355
AIRFOIL2	Three-element airfoil	2-D triangles	4720	13722
TRIANGLE	Equilateral triangle	2-D triangles, all same size	5050	14850
AIRFOIL3	Four-element airfoil	2-D triangles	15606	45878
PWT	Pressurized wind tunnel	Thin shell in 3-space	36519	144794
BODY	Automobile body	3-D volumes and surfaces	45087	163734

Table 1: Test problems.

Mesh	Grading Δ	Random Linear Bisection	Deterministic Bisection	Inertial Bisection	Coordinate Bisection	Circle Bisection
EPPSTEIN	3.8	44	42	45	42	41
TAPIR	17.4	37	34	49	55	32
PARC	9.4	35	36	37	36	35
AIRFOIL2	18.0	165	165	209	172	93
TRIANGLE	1.0	142	142	142	142	142
AIRFOIL3	15.9	210	211	245	230	148
PWT	8.1	533		553	562	499
BODY	10.9	802		938	953	768

Table 2: Cut size for two-way partitions.

- Inertial bisection, using a single line or plane normal to the moment of inertia of the point set.
- Coordinate bisection, using the best of the d cuts normal to the coordinate axes.
- Circle cuts, using the more general (and more expensive) algorithm from Gilbert et al. [9].

All the cuts are exact 50:50 bisections (within a single vertex).

We see that the random and deterministic line cuts behave very similarly, given the same number of samples. Both of these cuts are often significantly better than the single cut normal to the moment of inertia, or to coordinate bisection. As expected, the best circle cut is always at least as good as the best line cut, and sometimes significantly better. On such a small sample it is difficult to observe the expected correlation between grading and the relative quality of line and circle cuts; we are currently doing more extensive experiments.

8 Final Remarks

The geometric structure of a finite element mesh can be useful in finding a partition of the mesh. Up to now, the only provably good geometric partitioning algorithms used cuts based on spheres or circles. Several heuristics have used cuts based on planes or lines, but their effectiveness varies considerably.

Here we have shown that mesh grading is the key issue in the difference between the quality of line and circle cuts. Some surprisingly simple variants of the plane and line cut methods are provably good, if the mesh grading is taken into account in the bound.

All the line- and plane-based partitioning methods covered in this paper have "embarrassingly parallel" formulations. (The earliest parallel implementation of a line cut method that we know of is due to Heath and Raghavan [11].) Some, like coordinate bisection, have been incorporated in standard scientific application packages; all are very easy to implement. For example, the Matlab geometric partitioning toolbox of Gilbert and Teng [9] uses a hybrid of the sphere-cut method and an inertially biased random balanced linear method. Gilbert et al. showed experimentally that the hybrid procedure gives better performance than the sphere-based method alone.

Gilbert et al. observed that, when computing a multi-way partition recursively, most of the latter-stage partitions used line cuts, while in the early stages the more expensive circle cuts were better. Our new results suggest an explanation for this: Circle cuts outperform line cuts when the grading of the mesh is large, which is often the case for the original mesh.

However, since grading often changes fairly slowly across the mesh, after a few levels of partitioning the remaining fragments have much smaller grading and the line cuts become more competitive.

Our two-way partitioning bound can be extended to multi-way partitioning when the line and plane schemes are called recursively [19]. The total separator size or cut size for a p-way partition of a well-shaped mesh with grading Δ is $O(p^{1/d}\Delta^{1/d}n^{1-1/d})$.

We also showed that a line cut produces a provably good 50:50 split for a well-shaped mesh, where again the cut size depends on the grading. To the best of our knowledge, this is the first rigorous analysis of any exact 50:50 partitioning method that uses only a single cut. Our bound on cut size for the 50:50 line partitioner is $O(\Delta^{1/d}n^{1-1/d}(\log n)^{1/d})$ (for fixed α). This is weaker by a factor of $(\log n)^{1/d}$ than our bound for a cut with 1:d balance. Based on work in progress, we believe that the extra factor of $(\log n)^{1/d}$ may be required in dimension $d = 2$ but not in higher dimensions; we conjecture that for $d \geq 3$ an exact 50:50 line cut of size $O(\Delta^{1/d}n^{1-1/d})$ always exists.

Acknowledgments We thank Nina Amenta, Marshall Bern, Michael Paterson, and Frances Yao for helpful discussions of Lemma 6.4, which sharpened our insight into both the proof and the possibility of lower bounds.

References

[1] Babuška, I., and Aziz, A. On the angle condition in the finite element method. *SIAM J. Numer. Analysis 13*, 2 (1976), 214–227.

[2] M. Bern, D. Eppstein and J. R. Gilbert. Provably good mesh generation. In *31st Annual Symposium on Foundations of Computer Science, IEEE*, 231–241, 1990, (to appear JCSS).

[3] M. Bern and D. Eppstein. Mesh generation and optimal triangulation. In *Computing in Euclidean Geometry*, F. K. Hwang and D.-Z. Du editors, World Scientific, 1992.

[4] K. Clarkson, D. Eppstein, G. L. Miller, C. Sturtivant, and S.-H. Teng. Approximating center points with and without linear programming. In *Proceedings of 9th ACM Symposium on Computational Geometry*, pages 91–98, 1993.

[5] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, NY, 1987.

[6] C. Farhat and M. Lesoinne. Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics. Int. J. Num. Meth. Eng. 36:745-764 (1993).

[7] I. Fried. Condition of finite element matrices generated from nonuniform meshes. *AIAA J.* 10, pp 219–221, 1972.

[8] J. A. George and J. W. H. Liu. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, 1981.

[9] J. R. Gilbert, G. L. Miller, and S.-H. Teng. Geometric mesh partitioning: Implementation and experiments. *SIAM J. Sci. Comp.*, to appear 1995.

[10] K. Gremban, G. L. Miller, and S.-H. Teng. Moment of inertia and graph separators. In *Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp452–461, January 1994.

[11] M. Heath and P. Raghavan. A cartesian parallel nested dissection algorithm. To appear in *SIAM Journal on Matrix Analysis and Applications*, 1994.

[12] R. J. Lipton, D. J. Rose, and R. E. Tarjan. Generalized nested dissection. *SIAM J. on Numerical Analysis*, 16:346–358, 1979.

[13] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis. Automatic mesh partitioning. In A. George, J. Gilbert, and J. Liu, editors, *Graph Theory and Sparse Matrix Computation*, volume 56 of *IMA Volumes in Mathematics and its Applications*. Springer-Verlag, 1993.

[14] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis. Finite element meshes and geometric separators. *SIAM J. Scientific Computing*, to appear, 1995.

[15] G. L. Miller, D. Talmor, S.-H. Teng, and N. Walkington. Delaunay based numerical method for three dimensions: generation, formulation, and partition. In the *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, ACM, 1995

[16] G. L. Miller and S. A. Vavasis. Density graphs and separators. In *Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 331–336, San Francisco, January 1991. ACM-SIAM.

[17] S. A. Mitchell and S. A. Vavasis. Quality mesh generation in three dimensions. Proc. ACM Symposium on Computational Geometry, pp 212-221, 1992.

[18] H. D. Simon. Partitioning of unstructured problems for parallel processing. *Computing Systems in Engineering* 2(2/3):135-148, 1991.

[19] H. D. Simon and S.-H. Teng. How good is recursive bisection? *SIAM J. Scientific Computing*, to appear, 1995.

[20] S.-H. Teng. *Points, Spheres, and Separators: a unified geometric approach to graph partitioning*. Ph.D. Thesis, Carnegie Mellon University, CMU-CS-91-184, 1991

[21] S. A. Vavasis. Automatic domain partitioning in three dimensions. *SIAM J. Sci. Stat. Comp.*, 12:950–970, 1991.

[22] R. D. Williams. Performance of dynamic load balancing algorithms for unstructured mesh calculations *Concurrency: Practice and Experience*, 3(5): 457-481, 1991.