UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Geometric Spectral Partitioning

Tony F. Chan
John R. Gilbert
Shang-Hua Teng

January 1995

CAM Report 95-5

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

Geometric Spectral Partitioning

Tony F. Chan* John R. Gilbert† Shang-Hua Teng‡

July 6, 1994
(Revised January 14, 1995)

Abstract

We investigate a new method for partitioning a graph into two equal-sized pieces with few connecting edges. We combine ideas from two recently suggested partitioning algorithms, spectral bisection (which uses an eigenvector of a matrix associated with the graph) and geometric bisection (which applies to graphs that are meshes in Euclidean space). The new method does not require geometric coordinates, and it produces partitions that are often better than either the spectral or geometric ones.

Keywords: graph and geometric algorithms, mesh partitioning, spectral partitioning, parallel processing, separators, sparse matrix computations.

AMS subject classifications: 65F50, 68Q20

Computing Reviews descriptors: G.1.3 [**Numerical Analysis**]: Numerical Linear Algebra — *Linear systems (direct and iterative methods), Sparse and very large systems*; G.2.2 [**Discrete Mathematics**]: Graph Theory — *Graph algorithms*;

1 Introduction

The graph bisection problem is to partition the vertices of a graph into two sets of equal size, such that only a small number of edges join the two sets. Graph bisection has long been studied from both theoretical and heuristic points of view, motivated by such applications as VLSI layout, divide-and-conquer algorithms, solution of sparse linear systems, and load balancing for parallel computation [9, 15, 20]. Theoretical work includes separator theorems for various special classes of graphs [1, 10, 12, 21, 23], and approximation algorithms that guarantee to produce bisections within a polylog factor [19] or, recently, a constant factor [6] of optimal. It remains to be seen whether these approximation algorithms will be useful in practice.

Heuristics for graph bisection have recently received intense study motivated by the problem of partitioning a computational mesh for parallel processing. Typically the graph represents a finite element mesh, the nonzero structure of a sparse matrix, an electrical network, or some other sparsely connected collection of sites, elements,

*Mathematics Department, University of California, Los Angeles, CA 90024 (chan@math.ucla.edu). The work of this author was partially supported by the National Science Foundation under contract ASC 92-01266, the Army Research Office under contracts DAAL03-91-G-0150 and DAAL03-91-C-0047 (Univ. Tenn. subcontract ORA4466.04 Amendment 1), and ONR under contract ONR-N00014-92-J-1890. He also acknowledges support from RIACS under contract number NAS 2-13721 for visits to RIACS/NASA Ames where part of this work was carried out.

†Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, California 94304-1314 (gilbert@parc.xerox.com).

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

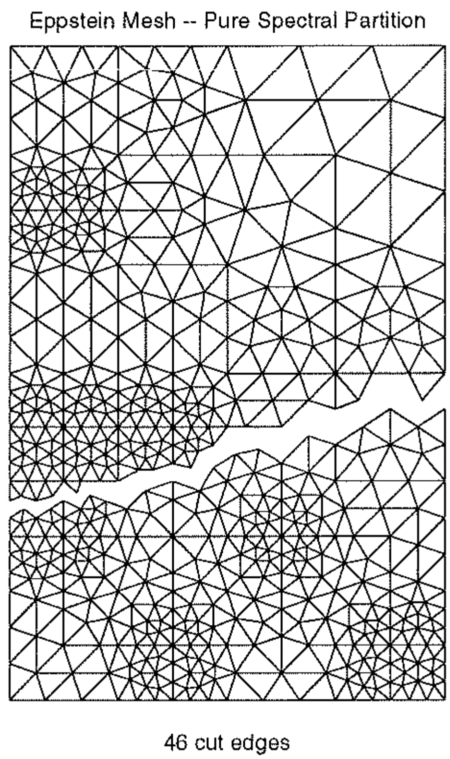

Figure 1: Spectral bisection of a finite element mesh.

or vertices. The computation to be performed consists of doing some work at each vertex, and interchanging data between adjacent vertices. For such a computation to be efficient on a distributed-memory parallel machine, the vertices must be partitioned among the memories in a way that satisfies two goals: first, for load balance, each partition should have the same size; second, to minimize communication, as few edges as possible should join vertices in different partitions. The most usual (though not the only) way to partition a graph among a number of processors that is a power of two is to bisect the graph recursively.

Two currently popular graph bisection heuristics are *spectral bisection*, which computes an eigenvector of a matrix associated with the graph, and *geometric bisection*, which applies to a graph whose vertices have specific locations in Euclidean space. In this note we begin by trying to improve upon spectral bisection, and we are led to a method we call *geometric spectral bisection* that combines elements of both approaches. The next three sections motivate and describe geometric spectral bisection. The following section presents some preliminary experiments comparing the new method to spectral bisection, geometric bisection, and to a recently suggested multilevel vertex-swapping method. The last section discusses directions for further research.

2 Spectral bisection

Spectral bisection is based on ideas of Fiedler and Mohar [8, 24]; its application to parallel scientific computing was first suggested by Pothen, Simon, and Liou [26]. Barnard and Simon [2] and Hendrickson and Leland [17] have experimented with efficient implementations of the method. We describe the method in some detail in order to motivate our new algorithm as a heuristic improvement. Figure 1 shows a spectral bisection of one of our test graphs.

Consider a graph G with an even number of vertices, numbered arbitrarily from 1 to n. The *Laplacian matrix* of G is the n by n symmetric matrix $A = (a_{ij})$ whose diagonal element a_{ii} is the degree of vertex i, and whose off-diagonal element $a_{ij} = a_{ji}$ is equal to -1 if vertices i and j are joined by an edge and to 0 otherwise. The Laplacian matrix is singular, since its rows all sum to zero. In fact it can be shown that A is positive semidefinite, and that the multiplicity of the zero eigenvalue is equal to the number of connected components of G. In spectral bisection it is conventional to add dummy edges if necessary to make G connected, so that A has rank exactly $n-1$. Then the eigenvalues of A are $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_n$, with corresponding eigenvectors $u_1, u_2, \ldots, u_n$; and the components of u_1 are all equal to $1/\sqrt{n}$.

A two-way partition of the vertices of G can be described by a *cut vector* x, whose component x_i is equal to 1 if vertex i is in one part or to -1 if vertex i is in the other part. If x is a cut vector, then it is easy to show that the scalar $x^T Ax$ is 4 times the number of edges that cross the cut. (For a proof, note that $x^T Ax = \sum_{(i,j)\in E}(x_i - x_j)^2$.) Therefore the optimal graph bisection problem can be stated as:

$$\min_x x^T Ax$$

$$x_i = \pm 1$$

$$\sum_i x_i = 0,$$

where the last condition says that the two parts must have equal size. This is a discrete optimization problem, since the admissible values of x are some of the vertices of an n-dimensional hypercube. The idea of spectral bisection is to solve a continuous version of the problem, by replacing the hypercube with a sphere, and then to round the continuous optimum to a nearby allowable discrete point. The continuous problem is

$$\min_x x^T Ax$$

$$||x|| = 1$$

$$\sum_i x_i = 0,$$

where the second condition now specifies the whole sphere instead of just the corners of the hypercube. (Actually $||x|| = \sqrt{n}$ would specify the sphere containing the corners of the hypercube, but this doesn't make any difference.)

The continuous problem asks to minimize the Rayleigh quotient $x^T Ax/x^T x$ over vectors x orthogonal to the vector of all ones, or equivalently, over vectors orthogonal to the first eigenvector u_1. Thus the solution is the second eigenvector $x = u_2$. Spectral bisection computes this eigenvector, then rounds it to a discrete point by changing to $+1$ those $n/2$ components closest to $+\infty$, and changing to -1 those $n/2$ components closest to $-\infty$.

In other words, spectral bisection partitions the vertices are about the median of the entries in the second eigenvector of the Laplacian matrix. Using the median guarantees that the partition is balanced. Chan, Ciarlet, and Szeto [4] prove that the result is in fact the closest balanced discrete point to u_2, in any l_p norm. Chan and Szeto [5] suggest a variation in which the vertices are divided according to the sign of the elements of u_2 (possibly giving an unbalanced partition), and then a discrete vertex-moving rule is used to walk along the edges of the hypercube until a balanced partition is reached.

Relaxing the discrete problem to a continuous one is intutitively attractive but not provably good. There is no guarantee that the the rounded continuous optimum closely approximates the true discrete optimum. Chung and Yau [6] have recently shown that a related but more complicated continuous relaxation algorithm can approximate the discrete optimum within a constant factor.

3 Geometric bisection

A *mesh* is a graph whose vertices are located at specific points in Euclidean space. Several groups [7, 13, 16, 27, 28] have suggested partitioning a d-dimensional mesh by a $(d-1)$-dimensional hyperplane (a line to partition a 2D mesh, a plane to partition a 3D mesh). *Coordinate bisection* is a simple heuristic that chooses a partitioning plane perpendicular to one of the coordinate axes; *inertial bisection* tries to do better by choosing a plane perpendicular to some version of a moment of inertia of the points.

Geometric bisection is an algorithm of Miller, Teng, Thurston, and Vavasis [22] that partitions a d-dimensional mesh by a $(d-1)$-sphere (a circle to partition a 2D mesh, a sphere to partition a 3D mesh). The algorithm provides theoretical guarantees for the quality of the partition it produces for certain special types of mesh; for example, a 2D mesh of triangles with a fixed lower bound on their smallest angle is partitioned by cutting at most $O(\sqrt{n})$ edges. The algorithm produces good partitions in practice for a wide variety of 2D and 3D meshes [11].

The circle separator for the mesh is actually constructed as a plane separator for an embedding of the mesh in one higher dimension. The geometric algorithm projects the points stereographically from d-space onto the unit sphere in $d+1$ dimensions. It uses combinatorial techniques to find a generalized median called a *centerpoint* for the projected points; any plane through the centerpoint will partition the mesh approximately evenly, though it may cut many edges. The algorithm then performs a conformal mapping on the surface of the sphere that corresponds to moving the centerpoint to the origin. The mapping spreads the points on the surface of the sphere so that the average number of edges cut by a random plane through the origin is small. The algorithm the chooses the best of a small (inertially biased) random sample of such cutting planes. Undoing the conformal map and stereographic projection yields a circle in the original d-space.

See Miller, Teng, Thurston, and Vavasis [22] for an overview of the theory of the geometric bisection algorithm, and Gilbert, Miller, and Teng [11] for an implementation.

4 Geometric spectral bisection

We motivate the combination of geometric and spectral bisection in two steps.

First step. Spectral bisection exactly minimizes the objective function $x^T Ax$ on the $(n-2)$-dimensional sphere of unit vectors orthogonal to u_1. The rounding step, however, does not take the shape of the objective function into account; rounding to a nearby discrete point in a direction of rapid increase of the objective function may not be as good as rounding to a more distant discrete point in a direction in which the objective function increases more slowly. On the $(n-2)$-sphere, the direction of slowest increase of $x^T Ax$ from its minimum is toward $x = \pm u_3$, which is the minimum of $x^T Ax$ among points on the sphere orthogonal to u_2. This suggests using u_2 as a starting point and performing a one-dimensional search along the great circle $x(\alpha) = ((1-\alpha)u_2 + \alpha u_3)/\|(1-\alpha)u_2 + \alpha u_3\|$, for $-1 \leq \alpha \leq 1$.

We experimented with this idea, and it does indeed frequently give better bisections than the "pure spectral" bisection $\alpha = 0$. The line search is expensive. We need to minimize the function $c(\alpha)$ that describes how many edges cross the cut obtained by partitioning the vertices around the median element of $x(\alpha)$. This function is not particularly well-behaved; in fact, $c(\alpha)$ is piecewise constant. Figure 2 shows cut size as a function of α for the sample graph in Figure 1. The best cut in this case is near $\alpha = 0.13$, and has 42 edges.

(The maximum number of constant pieces $c(\alpha)$ can have is actually equivalent to a well-known open problem of computational geometry, namely the complexity of the levels of an arrangement of lines. Given n non-vertical lines in the xy plane, consider the piecewise linear path from $x = -\infty$ to $x = +\infty$ that follows the line whose y value is the $n/2$'th largest at each point. How many linear segments can this piecewise linear path have? An easy upper bound is $O(n^2)$, since segments begin and end only at intersections of lines, and there are at most $O(n^2)$ intersections. In fact the maximum is known to lie between $O(n^{3/2})/\log^* n$ and $\Omega(n \log n)$ [25].)

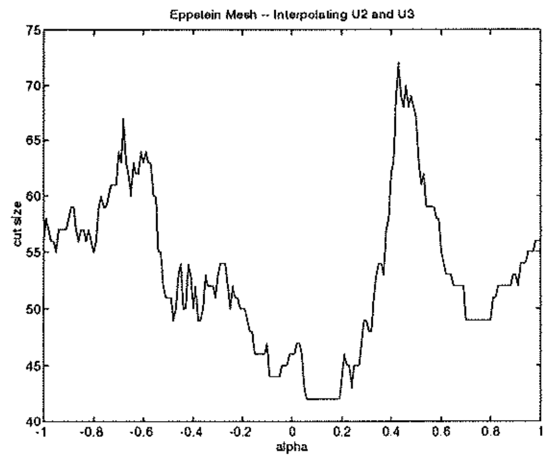

Figure 2: Cut size as a function of α for the sample mesh.

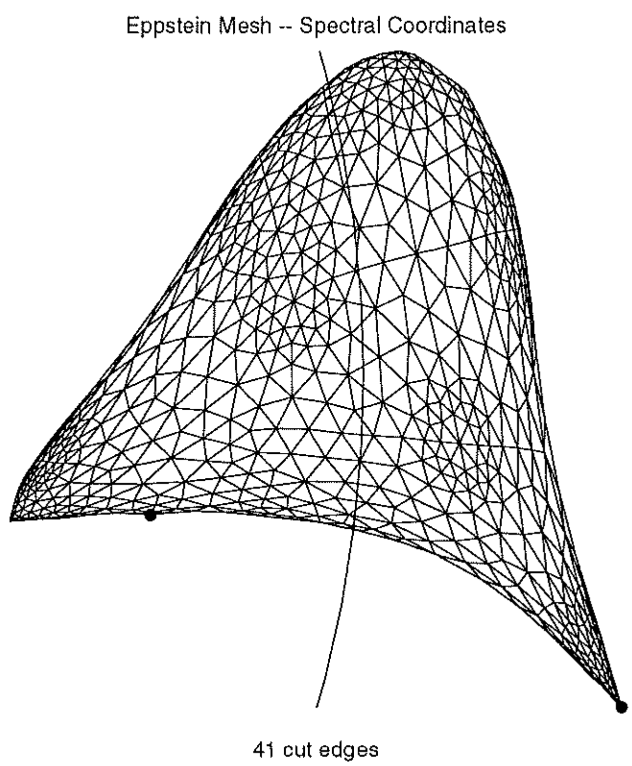

Figure 3: Sample mesh in spectral coordinates, with geometric spectral bisection.

Hendrickson and Leland [17] have also developed a partitioning method that uses more than one eigenvector of the Laplacian matrix. Their context is different than ours: they use multiple eigenvectors to divide the graph into more than two pieces at once.

Second step. Seeking an efficient way to choose a good α actually leads to an idea that finds even better bisections, as follows. The *d-dimensional spectral embedding* of a graph is the mesh obtained by assigning to vertex i the coordinates $(u_{i2}, u_{i3}, \ldots, u_{i,d+1})$, where u_{ij} is the j-th element of the eigenvector u_i of the Laplacian. Thus, in two dimensions, the spectral embedding labels each vertex with the corresponding element of u_2 as an x coordinate and the corresponding element of u_3 as a y coordinate. Figure 3 shows the sample mesh drawn with these coordinates. In this coordinate system, pure spectral bisection is the same as coordinate bisection. The family of cuts parameterized by α described above is the family of balanced cuts obtained by straight lines at arbitrary angles to the coordinate axes. Only a small leap of faith is needed to suggest using the geometric bisection algorithm to find a circle partition in this coordinate system. This is the idea that we call "geometric spectral bisection".

A straight line cut is a special case of a circle cut, so this idea searches for a cut among a family that includes the one defined above. In addition, geometric bisection gives an efficient way to look for a good cut in this family, since the geometric bisection algorithm itself is quite fast.

We know very little about the geometry imposed on a graph by its eigenvalues u_2 and u_3. This geometry has been used heuristically as a starting point for VLSI layout [3]. Geometric bisection of a mesh with coordinates in the "real world" seems to be effective because spatial locality reflects locality in the graph in some way; it is not clear that this should be the case for the artificial spectral coordinates. Horst Simon (personal communication) experimented a few years ago with drawings of meshes based on spectral coordinates; he observed, as we have, that these coordinates do seem to preserve locality in some sense. It seems that this spectral geometry does contain some information about a graph, but for the present we leave its investigation as an open problem and simply report on its use as a bisection heuristic.

5 Experiments

We experimented with geometric spectral bisection by using Gilbert, Miller, and Teng's Matlab mesh partitioning toolkit [11], which implements both geometric and spectral bisection. We used several standard test problems from finite element methods and other sources, as described in Table 1.

To illustrate the effect of the new method, we bisected each graph with a pure spectral algorithm, with a pure geometric algorithm (if the graph was originally given with coordinates), with geometric spectral bisection in both two and three dimensions, and with Hendrickson and Leland's multilevel Kernighan–Lin algorithm as implemented in their Chaco package [18]. Geometric spectral partitioning seems to work surprisingly well, both on structural problems and on the the BCSPWR problems that do not have an underlying mesh.

6 Remarks

Our new partitioning method can be viewed either as a generalization of spectral bisection to higher dimensions, or as a way to use geometric bisection on a graph whose coordinates are unknown. The heuristic is rather loosely motivated, but it appears to be quite effective. The most interesting question about the method is why it works so well.

The spectral embedding's role is, in a sense, to reduce the size of the search space. An n-vertex graph has 2^n partitions; an n-vertex mesh in d-space has only about n^d different partitions induced by planes or n^{d+1} induced by circles. The surprising thing is that the exponentially smaller space still tends to contain good partitions.

The geometric partitioner's role is to search that space. Our geometric partitioner performs a random search, with a carefully constructed probability distribution. The distribution was constructed to guarantee good results for meshes satisfying geometric conditions that our spectrally embedded graphs do not necessarily meet. Thus it may be that a different searching strategy would give better results with the same amount of effort. It would be

Name	Description	Mesh Type	Vertices	Edges
BCSPWR05	Western US power network	not a mesh	443	590
EPPSTEIN	Lightly graded mesh	2-D triangles	547	1566
TAPIR	Cartoon animal	2-D triangles, sharp angles	1024	2846
PARC	Mesh with holes	2-D triangles	1240	3355
BCSPWR09	Western US power network	not a mesh	1723	2394
BCSSTK14	Roof of Omni coliseum, Atlanta	unknown	1806	30824
BCSSTK15	Module of an offshore platform	unknown	3948	56934
AIRFOIL1	Three-element airfoil	2-D triangles	4253	12289
BCSPWR10	Eastern US power network	not a mesh	5300	8271
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.

Problem	Pure Spectral	Pure Geometric	Multilevel K-L	2D Geometric Spectral	3D Geometric Spectral
BCSPWR05	24	*	14	10	10
EPPSTEIN	46	42	40	41	41
TAPIR	58	35	23	23	23
PARC	24	36	21	21	21
BCSPWR09	34	*	14	9	9
BCSSTK14	1394	*	788	786	781
BCSSTK15	1496	*	1474	1477	1492
AIRFOIL1	132	93	82	86	81
BCSPWR10	44	*	30	30	32
AIRFOIL3	194	148	171	156	150
PWT	362	442	364	360	360
BODY	936	808	505	362	343

Table 2: Bisections. (* means no coordinates were supplied.)

interesting to study (either analytically or experimentally) the quality of the best circle- or plane-induced partitions of spectrally embedded graphs.

The circle separator for the spectrally embedded points is a plane separator for an embedding one dimension higher, where the extra dimension comes from stereographic projection. There are other ways to get another dimension—for example, one could use another eigenvector of the Laplacian. There is no obvious reason to expect that a circle separator for a 2D spectral embedding should be better than a plane separator for a 3D spectral embedding. However, it remains unclear how to search efficiently for a good separating plane.

How many eigenvectors of the Laplacian are useful in partitioning a given graph? Using one eigenvector gives ordinary spectral bisection; we have seen that using more can give better results. At the extreme, using all the eigenvectors gives an embedding with no information at all: the $(n-1)$-dimensional spectral embedding places the graph's nodes at the vertices of a regular simplex, all equidistant from each other. (Proof: Prepend $1/\sqrt{n}$ to each node's $(n-1)$-dimensional spectral coordinates. The resulting points are equidistant in n dimensions because they are the rows of an orthogonal matrix, the matrix whose columns are eigenvectors of the Laplacian.) Intuitively, one might expect a graph that began as a finite element mesh in d dimensions to have d "useful" Laplacian eigenvectors, but we do not see evidence of that in Table 2.

What theorems might one try to prove about the geometry of spectral embeddings? One striking thing about the pictures is that many of the 2D embeddings have no crossing edges: do u_2 and u_3 actually give planar embeddings of planar graphs? The answer is no; for example, the spectral embedding of the AIRFOIL1 mesh is almost, but not quite, planar. A weaker result might be true; for example,

> **Conjecture:** The following holds for some $\alpha, c > 0$: If a graph has a planar embedding in which every face except the exterior is a triangle with no angle smaller than α, and in which the exterior face has at most $c\sqrt{n}$ vertices, then the graph's 2D spectral embedding is planar.

One could also ask what conditions on a finite element mesh in d dimensions would imply that its spectral embedding in $d' > d$ dimensions, considered as a d-manifold, is non-self-intersecting.

Guattery and Miller [14] have constructed an interesting class of graphs for which no geometric spectral algorithm can guarantee to find good partitions. These are bounded-degree graphs that correspond to 3D meshes with good separators, but for which no bounded-dimension spectral embedding contains enough information to find the best separators. It is not clear whether such an example corresponds to any realistic finite element mesh in three dimensions. It would be interesting to identify a realistic class of graphs for which geometric spectral bisection (or some other geometric algorithm applied to a spectral embedding) guarantees good partitions. This would probably require both new results about properties of spectral embeddings, and stronger versions of geometric separator theorems.

References

[1] Noga Alon, Paul Seymour, and Robin Thomas. A separator theorem for graphs with an excluded minor and applications. In *Proceedings of the 22th Annual Symposium on Theory of Computing*. ACM, 1990.

[2] Stephen T. Barnard and Horst D. Simon. A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. Technical Report RNR-92-033, NASA Ames Research Center, 1992. A short version is in *Proc. 6th SIAM Conf. Parallel Processing for Scientific Computing*, SIAM, 1993, pp. 711–718.

[3] John P. Blanks. Near-optimal quadratic-based placement for a class of IC layout problems. *IEEE Circuits and Devices Magazine*, pages 31–37, 1985.

[4] T. F. Chan, P. Ciarlet Jr., and W. K. Szeto. On the optimality of the median cut spectral bisection graph partitioning method. Technical Report CAM 93-14, University of California at Los Angeles, 1993.

[5] Tony F. Chan and W. K. Szeto. A sign cut version of the recursive spectral bisection graph partitioning algorithm. In *Proceedings of the Fifth SIAM Conference on Applied Linear Algebra*, pages 562–566, Snowbird, UT, June 1994.

[6] F. R. K. Chung and S.-T. Yau. A near optimal algorithm for edge separators. In *Proceedings of the 26th Annual Symposium on Theory of Computing*. ACM, 1994.

[7] 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.

[8] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovakian Mathematics Journal*, 23:298–305, 1973.

[9] Alan George and Joseph W. H. Liu. An automatic nested dissection algorithm for irregular finite element problems. *SIAM Journal on Numerical Analysis*, 15:1053–1069, 1978.

[10] John R. Gilbert, Joan P. Hutchinson, and Robert Endre Tarjan. A separator theorem for graphs of bounded genus. *Journal of Algorithms*, 5:391–407, 1984.

[11] John R. Gilbert, Gary L. Miller, and Shang-Hua Teng. Geometric mesh partitioning: Implementation and experiments. Technical Report CSL–94–13, Xerox Palo Alto Research Center, 1994.

[12] John R. Gilbert, Donald J. Rose, and Anders Edenbrandt. A separator theorem for chordal graphs. *SIAM Journal on Algebraic and Discrete Methods*, 5:306–313, 1984.

[13] Keith D. Gremban, Gary L. Miller, and Shang-Hua Teng. Moments of inertia and graph separators. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 452–461. SIAM, 1994.

[14] Stephen Guattery and Gary L. Miller. On the performance of spectral graph partitioning methods. In *Proceedings of the Sixth ACM-SIAM Symposium on Discrete Algorithms*, 1995.

[15] Steven Hammond and Robert Schreiber. Solving unstructured grid problems on massively parallel computers. Technical Report TR 90.22, Research Institute for Advanced Computer Science, 1990.

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

[17] Bruce Hendrickson and Robert Leland. An improved spectral graph partitioning algorithm for mapping parallel computations. Technical Report SAND92-1460, Sandia National Laboratories, Albuquerque, NM, 1992.

[18] Bruce Hendrickson and Robert Leland. The Chaco user's guide, Version 1.0. Technical Report SAND93-2339, Sandia National Laboratories, Albuquerque, NM, 1993.

[19] Tom Leighton and Satish Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pages 422–431. IEEE, 1988.

[20] Charles E. Leiserson. Area-efficient graph layouts (for VLSI). In *Proceedings of the 21st Annual Symposium on Foundations of Computer Science*, pages 270–281, 1980.

[21] Richard J. Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36:177–189, 1979.

[22] Gary L. Miller, Shang-Hua Teng, William Thurston, and Stephen A. Vavasis. Automatic mesh partitioning. In Alan George, John R. Gilbert, and Joseph W. H. Liu, editors, *Graph Theory and Sparse Matrix Computation*. Springer-Verlag, 1993.

[23] Gary L. Miller, Shanghua Teng, and Steven A. Vavasis. A unified geometric approach to graph separators. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science*, pages 538–547. IEEE, 1991.

[24] Bojan Mohar. The Laplacian spectrum of graphs. In Y. Alavi et al., editor, *Graph Theory, Combinatorics, and Applications*, pages 871–898. J. Wiley, 1991.

[25] János Pach, William Steiger, and Endre Szemerédi. An upper bound on the number of planar k-sets. *Discrete and Computational Geometry*, 7:109–123, 1992.

[26] Alex Pothen, Horst D. Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11:430–452, 1990.

[27] H. D. Simon. Partitioning of unstructured problems for parallel processing. *Computing Systems in Engineering*, 2:135–148, 1991.

[28] R. D. Williams. Performance of dynamic load balancing algorithms for unstructured mesh calculations. *Concurrency: Practice and Experience*, 3:457–481, 1991.