A SPECTRAL ALGORITHM FOR ENVELOPE REDUCTION OF SPARSE MATRICES*

Dedicated to William Kahan and Beresford Parlett on the occasion of their 60th birthdays.

STEPHEN T. BARNARD†, ALEX POTHEN‡ AND HORST D. SIMON§

Abstract. The problem of reordering a sparse symmetric matrix to reduce its envelope size is considered. A new spectral algorithm for computing an envelope-reducing reordering is obtained by associating a Laplacian matrix with the given matrix and then sorting the components of a specified eigenvector of the Laplacian. This Laplacian eigenvector solves a continuous relaxation of a discrete problem related to envelope minimization called the minimum 2-sum problem. The permutation vector computed by the spectral algorithm is a closest permutation vector to the specified Laplacian eigenvector. Numerical results show that the new reordering algorithm usually computes smaller envelope sizes than those obtained from the current standards such as the Gibbs-Poole-Stockmeyer (GPS) algorithm or the reverse Cuthill-McKee (RCM) algorithm in SPARSPAK, in some cases reducing the envelope by more than a factor of two.

Keywords. envelope reduction, eigenvalues of graphs, Gibbs-King algorithm, Gibbs-Poole-Stockmeyer algorithm, Laplacian matrices, reordering algorithms, reverse Cuthill-McKee algorithm, sparse matrices.

AMS(MOS) subject classifications. 65F50, 65F10, 68R10.

---

* Written Oct. 1993. A shorter version has appeared in Supercomputing '93.

† This author is an employee of Cray Research Inc., NASA Ames Research Center, MS T045-1, Moffett Field, CA 94035, email: barnard@nas.nasa.gov, tel: (415) 604-0171, FAX: (415) 604-3957.

‡ Computer Science Department, University of Waterloo, Waterloo, Ontario, N2L 3G1 Canada, email: apothen@narnia.uwaterloo.ca, tel: (519) 885-1211 extn 2979, FAX: (519) 885-1208. This author was supported by U. S. National Science Foundation grant CCR-9024954 and by U. S. Department of Energy grant DE-FG02-91ER25095 at the Pennsylvania State University and by the Canadian Natural Sciences and Engineering Research Council under grant OGP0008111 at the University of Waterloo.

§ This author is an employee of Computer Sciences Corporation. This work was supported through NASA Contract NAS 2-12961, NASA Ames Research Center, MS T045-1, Moffett Field, CA 94035, email: simon@nas.nasa.gov, tel: (415) 604-4322, FAX: (415) 604-3957.

1. Introduction. Algorithms for reordering sparse matrices play a vital role in our ability to perform many large-scale matrix computations. Ordering algorithms such as minimum-degree and nested dissection have been developed for reducing fill in direct methods for solving sparse, symmetric positive definite systems of equations [7, 12, 27]. Various ordering algorithms for reducing the envelope (variable band or profile) of sparse matrices, such as the reverse Cuthill-McKee (RCM), Gibbs-Poole-Stockmeyer (GPS), and Gibbs-King (GK) algorithms, have also been designed [12, 15, 21]. Although envelope-reducing orderings were developed for use in envelope schemes for direct factorization, these orderings have been used in the past few years in several other applications. The RCM ordering has been found to be an effective preordering in computing incomplete factorization preconditioners for preconditioned conjugate gradients methods [6, 8]. Such orderings have also been used in parallel matrix-vector multiplication and tridiagonalization of sparse symmetric matrices.

The wider applicability of envelope-reducing orderings justifies a fresh look at the algorithms currently available and the development of new algorithms. In this paper we present a new spectral algorithm for computing an envelope-reducing ordering of sparse, symmetric matrices. The ordering algorithm uses an eigenvector corresponding to the smallest positive eigenvalue of the discrete Laplacian matrix associated with the given symmetric matrix. (If the matrix is irreducible, or equivalently if its adjacency graph is connected, then this eigenvector corresponds to the second smallest eigenvalue. Hence we call this a *second Laplacian eigenvector* or *Fiedler vector.*) The ordering is computed by permuting the components of a second Laplacian eigenvector in nonincreasing (or nondecreasing) order. For large matrices, the eigenvector computation is performed by a 'multilevel' approach described in [3].

Earlier, we had used a second eigenvector of the Laplacian matrix for computing a spectral nested dissection ordering, and for partitioning computations on finite element meshes on a distributed-memory multiprocessor [29, 30, 31]. The eigenvector of the adjacency matrix corresponding to the largest eigenvalue has been used to find a pseudoperipheral node by Grimes et al. [17].

A companion paper [13] provides theoretical justification for the spectral envelope-reduction algorithm by considering a closely related problem called the 2-sum problem. (This problem is defined in the next section.) It is shown there that this problem can be formulated as a quadratic assignment problem involving the Laplacian matrix. Lower bounds for the 2-sum are obtained in terms of the smallest positive Laplacian eigenvalue. These bounds appear to be reasonably tight, and hence indicate how close the computed orderings are to the optimal orderings. Further, permuting the matrix in nonincreasing (or nondecreasing) order of the components of a second Laplacian eigenvector is shown to yield a feasible solution to the 2-sum problem that is closest to an infeasible solution for which the lower bound is attained.

Fiedler [9, 11] studied the properties of the second Laplacian eigenvalue and a corresponding eigenvector and their relationship to the connectivity of a graph, and also observed [10] that the differences in the components of this eigenvector is an approximate measure of the distance between the vertices. Juvan and Mohar [19] have

advocated the use of this eigenvector to compute bandwidth and p-sum reducing orderings. Mohar and Poljak [25] have recently provided a comprehensive survey of the applications of Laplacian spectra to combinatorial problems.

The spectral envelope-reduction algorithm has several features which set it apart from the earlier reordering algorithms such as the GPS, GK, or RCM algorithms [5, 12, 15, 21]. These algorithms employ local-search in the adjacency graph of the matrix. All of them try to find a pseudo-diameter in the graph by generating a long level-structure by breadth-first-search beginning from a suitable vertex. These types of algorithms generally do not vectorize, and there is no obvious way to implement them in parallel. In contrast the new algorithm proposed here is based on the computation of an eigenvector of a special matrix, and hence involves standard floating point operations, such as matrix vector multiplications, dot products etc. The algorithms for these operations not only vectorize easily, but also can be implemented in parallel with little effort. (Parallel implementation of the basic spectral method, which uses the Lanczos algorithm to find eigenvectors, is straightforward. Parallel implementation of the 'multilevel' enhancements described in Section 3 is more difficult, but possible in principle.) The algorithm is also iterative in nature, in the same sense that SOR or the Lanczos methods are iterative. It allows a user to terminate the reordering process depending on a stopping criterion, thus permitting the user to make trade-offs in ordering time versus storage efficiency.

Before we end this introduction, some comments are in order about the applicability of the results to envelope factorization schemes. Frontal methods related to envelope or profile schemes are still the method of choice for solving large-scale systems of linear equations in many structural engineering applications, for example in the computational structural mechanics testbed (CSM) at NASA Langley [20]. Implementations of these methods are also widely distributed in most of the finite element software packages such as MSC/NASTRAN or ANSYS. Parallel algorithms for the actual numerical factorization of a matrix in envelope format have been investigated [28, 33].

Efficient implementations of sparse matrix algorithms [1, 2, 22, 32] on supercomputers demonstrate that very high levels of performance are attainable with general sparse algorithms. Hence there are no good reasons to use envelope schemes for sparse matrix factorizations for the sake of performance alone. Furthermore, it has long been known that general sparse methods are considerably more efficient with respect to storage [12]. Ashcraft et al. [2] presented numerical evidence that general sparse methods outperform envelope methods in both respects. However, envelope methods and related methods such as frontal or skyline methods continue to be the standard solution option in many commercial structural analysis packages. Thus, demonstrating the efficiency of the new spectral algorithm offers potential performance improvements in these packages without making substantial changes to the underlying data structures. Further, Liu [23] has described a *generalized envelope* algorithm for computing the numerical factorization by rows, and his results show that such a scheme can compete with general sparse algorithms.

The following is an outline of the rest of this paper. In Section 2 we formulate the

problems associated with the minimization of envelope parameters and describe related problems called the 1-sum and 2-sum problems. We describe some theoretical results to justify the proposed new algorithm. The second Laplacian eigenvector solves a continuous relaxation of a discrete problem related to the envelope problem, the minimum 2-sum problem. Further, it is proved that the permutation vector computed by the spectral algorithm is a closest (in the 2-norm sense) permutation vector to a second Laplacian eigenvector. In Section 3 we discuss the spectral algorithm and its numerical implementation. The multilevel algorithm, which uses coarsening of the underlying graph combined with Rayleigh Quotient iteration [3], to compute the eigenvector is described. Numerical results and comparisons with GPS, GK, and RCM are presented in Section 4. These results indicate that the new algorithm is often considerably more efficient in reducing the storage requirements. The spectral algorithm does require greater execution time for computing the ordering, but the new ordering often yields greatly reduced factorization times for the spectrally reordered matrices.

2. The envelope reduction problem.

2.1. The envelope of a matrix. Let A be an $n \times n$ symmetric matrix with elements a_{ij}, whose diagonal elements are nonzero. We consider various parameters of the matrix A associated with its envelope.

We denote the column indices of the nonzeros in the lower triangular part of the i-th row by $row(i) = \{j : a_{ij} \neq 0, \text{ and } 1 \leq j \leq i\}$. For the i-th row of A we define

$$f_i(A) = \min\{j : j \in row(i)\}, \quad \text{and} \tag{2.1}$$

$$r_i(A) = i - f_i(A). \tag{2.2}$$

Here $f_i(A)$ is the column index of the first nonzero in the i-th row of A (by our assumption of nonzero diagonals, $1 \leq f_i \leq i$), and the parameter $r_i(A)$ is the *row-width* of the i-th row of A. The *bandwidth* of A is the maximum row-width

$$bw(A) = \max\{r_i(A) : i = 1, \ldots, n\}.$$

The *envelope* of A is the set of column indices that lie between the first nonzero column index and the diagonal in each row:

$$Env(A) = \{(i,j) : f_i(A) \leq j < i, \text{and } i = 1, \ldots, n\}.$$

We denote the size of the envelope by $Esize(A) = |Env(A)|$. The work in the Cholesky factorization of A that makes use of an envelope storage scheme can be bounded from above by $(1/2)\sum_{i=2}^{n} r_i(r_i + 3)$. Hence hereafter we will denote $Ework(A) = \sum_{i=1}^{n} r_i^2$ as a measure of the work in such a factorization. We stress that this estimate is an upper bound on the actual work in an envelope factorization scheme.

The values of these parameters strongly depend on the choice of an ordering of the rows and columns, and thus we consider how these parameters vary for a symmetrically permuted matrix $P^T AP$, where P is a permutation matrix. We define $Esize_{min}(A)$, the minimum envelope size of A, to be the minimum size among the envelopes of all

permuted matrices $P^T AP$. The quantities $Ework_{min}(A)$ and $bw_{min}(A)$ are defined in similar fashion. In general the minima for these three quantities will not be attained by the same permutation.

The envelope parameters can also be defined with respect to the adjacency graph $G = (V, E)$ of A. Denote $nbr(v) = \{v\} \cup adj(v)$. In terms of the graph G and an ordering α of its vertices, we can define

$$r(v, \alpha) \;=\; \max\{\alpha(v) - \alpha(w) : w \in nbr(v), \alpha(w) \leq \alpha(v)\}.$$

Hence we can write the envelope size and work associated with an ordering α as

$$\begin{aligned} Esize(G, \alpha) &= \sum_{v \in V} r(v) = \sum_{v \in V} \max\{\alpha(v) - \alpha(w) : w \in nbr(v), \alpha(w) \leq \alpha(v)\} \\ Ework(G, \alpha) &= \sum_{v \in V} r^2(v) = \sum_{v \in V} \max\{(\alpha(v) - \alpha(w))^2 : w \in nbr(v), \alpha(w) \leq \alpha(v)\}. \end{aligned}$$

The goal is to choose a vertex ordering $\alpha : V \mapsto \{1, \ldots, n\}$ to minimize one of the parameters described above. We denote by $Esize_{min}(G)$ ($Ework_{min}(G)$) the minimum value of $Esize(G, \alpha)$ ($Ework_{min}(G, \alpha)$) over all orderings α, where again (in general) the minima will not all be attained by the same α. We will use the definitions in terms of matrices throughout the rest of the paper.

It will be helpful to consider quantities related to the envelope size and envelope work: the 1-sum, $\sigma_1(A)$, and the 2-sum, $\sigma_2^2(A)$. We write the envelope size and 1-sum, and the envelope work and the 2-sum in a way that shows their relationships:

$$\begin{aligned} Esize(A) &= \sum_{i=1}^{n} \max_{j \in row(i)} (i - j), \\ \sigma_1(A) &= \sum_{i=1}^{n} \sum_{j \in row(i)} (i - j), \\ Ework(A) &= \sum_{i=1}^{n} \max_{j \in row(i)} (i - j)^2, \\ \sigma_2^2(A) &= \sum_{i=1}^{n} \sum_{j \in row(i)} (i - j)^2. \end{aligned}$$

The parameters $\sigma_{1,min}(A)$ and $\sigma_{2,min}^2(A)$ are the minimum values of these parameters over all permuted matrices $P^T AP$.

It is known that minimizing the bandwidth and the 1-sum are NP-complete problems, the former even for trees with degree bounded by three. Minimizing any of the other quantities considered here is likely to be intractable as well, so one has to settle for heuristic orderings to reduce the quantity.

Recently it has been shown that the envelope size problem is intimately related to the 1-sum problem, and that the envelope work problem is related to the 2-sum problem [13]. Let Δ denote the maximum number of off-diagonal nonzeros in a row of A. (This is the maximum vertex degree in the adjacency graph of A.)

THEOREM 2.1 ([13]). *Let A be a symmetric matrix. The minimum values of the envelope size, estimate of the envelope work in the Cholesky factorization, 1-sum, and 2-sum of a symmetric matrix A are related by the following inequalities:*

$$Esize_{min}(A) \leq \sigma_{1,min}(A) \leq \Delta Esize_{min}(A). \tag{2.3}$$

$$Ework_{min}(A) \leq \sigma^2_{2,min}(A) \leq \Delta Ework_{min}(A). \tag{2.4}$$

$$\sigma_{2,min}(A) \leq \sigma_{1,min}(A) \leq \sqrt{|E|}\sigma_{2,min}(A). \tag{2.5}$$

■

2.2. The Laplacian matrix and bounds on envelope parameters. The Laplacian matrix $Q(G)$ of an undirected graph G is the $n \times n$ matrix $D - B$, where D is the diagonal degree matrix and B is the adjacency matrix of G. If G is the adjacency graph of a symmetric matrix M, then we could define the Laplacian matrix Q directly:

$$q_{ij} = \begin{cases} -1 & \text{if } i \neq j \text{ and } m_{ij} \neq 0, \\ 0 & \text{if } i \neq j \text{ and } m_{ij} = 0, \\ -\sum_{\substack{j=1 \\ j \neq i}}^{n} q_{ij} & \text{if } i = j. \end{cases}$$

The eigenvalues of $Q(G)$ are the *Laplacian eigenvalues* of G, and we list them as $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. An eigenvector corresponding to λ_k will be denoted by $\underline{x}_k$, and will be called a kth eigenvector of Q. It is well-known that Q is a singular M-matrix, and hence its eigenvalues are nonnegative. Thus $\lambda_1 = 0$, and the corresponding eigenvector is any nonzero constant vector $\underline{c}$. If G is connected, then Q is irreducible, and $\lambda_2 > 0$. The smallest nonzero eigenvalues and the corresponding eigenvectors have important properties that make them useful in the solution of various partitioning and ordering problems. These properties were first investigated by Fiedler [9, 11]; more recently several authors have studied their application to such problems.

Juvan and Mohar [19] have obtained bounds for bandwidth and p-sums in terms of Laplacian eigenvalues. They have also suggested the use of a second eigenvector to compute orderings to reduce bandwidth, 1-sum, and 2-sum. Helmberg, Mohar, Poljak, and Rendl [18] have obtained additional lower bounds on the bandwidth. The 1- and 2-sum problems have been recently formulated as quadratic assignment problems and thus bounds have been obtained for the envelope size and work [13]. The following result describes two of the simpler bounds:

THEOREM 2.2 ([13]). *The envelope size of a symmetric matrix A can be bounded in terms of its second and largest Laplacian eigenvalues as*

$$\frac{\lambda_2(A)}{6\Delta}(n^2 - 1) \leq Esize_{min}(A) \leq \frac{\lambda_n(A)}{6}(n^2 - 1).$$

Our estimate of the envelope work in the Cholesky factorization of A can be bounded as

$$\frac{\lambda_2(A)}{12\Delta}n(n^2 - 1) \leq Ework_{min}(A) \leq \frac{\lambda_n(A)}{12}n(n^2 - 1).$$

■

2.3. Approximate minimization of envelope work. We now offer some justification for the spectral envelope-reduction algorithm, which computes an ordering by sorting the components of a second Laplacian eigenvector. The idea is to consider the related 2-sum problem, and then to show that a second Laplacian eigenvector $\underline{x}_2$ solves a continuous relaxation of the problem. We then prove that the permutation vector computed by the spectral algorithm is a closest vector (in the 2-norm sense) among the permutation vectors to the eigenvector $\underline{x}_2$.

For odd n, let $\mathcal{P}$ denote the set of n-vectors $\underline{p}$ whose components are permutations of $\{-(n-1)/2, \ldots, -1, 0, 1, \ldots, (n-1)/2\}$. For even n, let $\mathcal{P}$ denote vectors that are permutations of $\{-n/2, \ldots, -1, +1, \ldots, n/2\}$. We denote the i-th component of a vector $\underline{x}$ by x_i. We consider the 2-sum of a symmetric matrix A, defined with respect to vectors in $\mathcal{P}$:

$$\min_{\underline{x}\in\mathcal{P}} \sum_{i=1}^{n} \sum_{j\in row(i)} (x_i - x_j)^2 = \frac{1}{2} \min_{\underline{x}\in\mathcal{P}} \sum_{a_{ij}\neq 0} (x_i - x_j)^2.$$

A strategy to approach this hard discrete problem is to relax the condition that $\underline{x}$ must belong to the set of permutation vectors and instead to minimize the objective function over a suitable class of n-vectors. This yields an easier continuous problem; we can then find the permutation vector closest to the solution vector of the continuous problem, and consider the former as an approximate solution of the combinatorial problem.

Note that any $\underline{p} \in \mathcal{P}$ satisfies $\underline{p}^T\underline{u} = 0$, and $\ell \equiv \underline{p}^T\underline{p} = (n/12)(n^2-1)$ for odd n, and $\ell = (n/12)(n+1)(n+2)$ for even n, where $\underline{u} = (1, 1, \ldots 1)^T$. Given a vector $\underline{x} \in \Re^n$, we can define a permutation vector $\underline{p}$ induced by $\underline{x}$ by the rule $p_i \leq p_j$ if and only if $x_i \leq x_j$. Note that the ordering of the columns and rows is unique except when two or more components have the same value x_i. Hence to obtain a continuous relaxation of the discrete problem, we consider the set $\mathcal{X}$ of vectors $\underline{x} \in \Re^n$ satisfying $\underline{x} \neq \underline{0}$, $\underline{x}^T\underline{u} = 0$, and $\underline{x}^T\underline{x} = \ell$. This is now a continuous optimization problem:

$$\begin{aligned}
&\frac{1}{2} \min_{\underline{x}\in\mathcal{X}} \sum_{a_{ij}\neq 0} (x_i - x_j)^2 \\
&= \min_{\underline{x}\in\mathcal{X}} \left(\sum_{i=1}^{n} d_i x_i^2 - 2 \sum_{\substack{j<i \\ a_{ij}\neq 0}} x_i x_j) \right) \\
&= \min_{\underline{x}\in\mathcal{X}} \underline{x}^T D \underline{x} - \underline{x}^T B \underline{x} \;=\; \min_{\underline{x}\in\mathcal{X}} \underline{x}^T Q \underline{x} \\
&= \lambda_2 \underline{x}_2{}^T \underline{x}_2 \;=\; \lambda_2 \ell.
\end{aligned}$$

Hence a second Laplacian eigenvector $\underline{x}_2$ solves the continuous approximation to the 2-sum problem. Now we prove that a permutation vector $\underline{p}_m$ induced by $\underline{x}_2$ is a closest vector in $\mathcal{P}$ to $\underline{x}_2$. Earlier a similar result was obtained by Chan and Szeto [4] for the graph bisection problem.

THEOREM 2.3. *The vector $\underline{p}_m$ induced by a second Laplacian eigenvector $\underline{x}_2$ is a closest (in the 2-norm) permutation vector to $\underline{x}_2$. In other words,*

$$\underline{p}_m = \arg\min_{\underline{p}\in\mathcal{P}} \|\underline{p} - \underline{x}_2\|_2.$$

We require the following lemma to prove the theorem.

LEMMA 2.4. *If $a_1 < a_2$, $b_1 < b_2$ are real numbers,*

$$r = (a_1 - b_2)^2 + (a_2 - b_1)^2, \quad \textit{and} \quad s = (a_1 - b_1)^2 + (a_2 - b_2)^2,$$

then $r > s$.

Proof. Suppose that $r \leq s$. Then

$$\begin{aligned} (a_1 - b_2)^2 + (a_2 - b_1)^2 &\leq (a_1 - b_1)^2 + (a_2 - b_2)^2 \\ \Rightarrow \quad a_2(b_2 - b_1) &\leq a_1(b_2 - b_1). \end{aligned}$$

Since $a_1 < a_2$, it follows that $b_2 \leq b_1$, which is a contradiction. ■

Proof of Theorem 2.3: For convenience of notation, let $\underline{x} \equiv \underline{x}_2$ in this proof. Let $\underline{y} \neq \underline{p}_m$ be a permutation vector such that there exists a pair of vertices u,v satisfying $x(u) < x(v)$ and $y(u) > y(v)$. Let z be the ordering such that $z(u) = y(v)$, $z(v) = y(u)$, and $z(w) = y(w)$ for all other vertices. Then

$$\begin{aligned} &\|\underline{y} - \underline{x}\|_2{}^2 - \|\underline{z} - \underline{x}\|_2{}^2 \\ &= (y(u) - x(u))^2 + (y(v) - x(v))^2 - (y(v) - x(u))^2 - (y(u) - x(v))^2 \\ &> 0, \end{aligned}$$

where the last inequality follows from the previous lemma. By the swapping of components, we have obtained a vector $\underline{z}$ that is closer than $\underline{y}$ to the eigenvector $\underline{x}$. By repeating this swapping procedure, we find that $\underline{p}_m$ is a closest vector in $\mathcal{P}$ to the vector $\underline{x}$. ■

Earlier Juvan and Mohar [19] had shown that $\underline{p}_m$ maximized the value of the following inner product over all permutation vectors $\underline{p}$:

$$|(\underline{x}_2, \underline{p}_m)| \geq |(\underline{x}_2, \underline{p})|.$$

Stronger justification of the spectral algorithm for reducing the 2-sum is obtained in the companion paper [13] by considering a quadratic assignment formulation of the problem. This formulation leads to a lower bound for the 2-sum in terms of the second Laplacian eigenvalue, and the orthogonal matrix attaining this lower bound can be characterized. It can be shown that a closest permutation matrix (defined in a suitable sense) to this orthogonal matrix is obtained by sorting the components of a second Laplacian eigenvector in nondecreasing (nonincreasing) order.

2.4. Adjacency orderings. We now consider the concept of an *adjacency ordering* of a graph G. Let G be the adjacency graph of a matrix A, and suppose that the vertices of G are ordered in some ordering as $\{v_1, \ldots, v_n\}$ (i.e., $\alpha(v_j) = j$), and let $V_j = \{v_1, \ldots, v_j\}$. For $Y \subset V$, define $adj(Y)$ to be the set of vertices in $V \setminus Y$ that are adjacent to some vertex in Y. We will say that an ordering is an *adjacency ordering* if $v_{j+1} \in adj(V_j)$, for $j = 1, \ldots, n-1$.

The size $|adj(V_j)|$ has been called the jth *frontwidth* [24], and corresponds to the size of the j-th column of the envelope of A. Hence an alternative expression for the the envelope size is

$$Esize(A) = \sum_{j=1}^{n} |adj(V_j)|.$$

This expression for the envelope size shows the rationale for considering adjacency orderings for envelope-reduction. The idea is to locally reduce the jth frontwidth by choosing v_j to be a vertex of low degree belonging to $adj(V_{j-1})$. The Cuthill-McKee ordering is an adjacency ordering, but RCM is not an adjacency ordering. The GPS and GK algorithms attempt to number vertices in the level structures to obtain an adjacency ordering, as far as is possible.

The ordering induced by a second Laplacian eigenvector is not an adjacency ordering, but comes close in the sense described below. The following theorem, proved by Fiedler [11], provides the necessary insight.

THEOREM 2.5. *Let G be a connected graph, and $\underline{x} = (x_1, x_2, \ldots x_n)$ be a second Laplacian eigenvector of G. For any real $\rho \leq 0$, define $S(\rho) = \{v_j \in V : x_j \geq \rho\}$. Then the subgraph induced on $S(\rho)$ is connected. Similarly, if $\rho \geq 0$, then $S'(\rho) = \{v_j \in V : x_j \leq \rho\}$ induces a connected subgraph.*

In the notation of the theorem, let the vertices $v_j \in V$ be ordered such that $j \leq k$ if and only if $x_j \leq x_k$. Consider three subsets of vertices corresponding to positive, zero, and negative entries in the second eigenvector; i.e., define $P = \{v_j : x_j > 0\}$, $Z = \{v_j : x_j = 0\}$, and $N = \{v_j : x_j < 0\}$. Let the vertices in N be numbered by $j = 1, \ldots, k$, the vertices in Z by $j = k+1, \ldots, p-1$, and the vertices in P by $j = p, \ldots, n$. We have $k < p$. Then Theorem 2.5 implies that for $j = p-1, \ldots, n$, we have $v_{j+1} \in adj(V_j)$. A similar statement holds if we add vertices with negative entries in the eigenvector in decreasing order to the set $P \cup Z$. Thus the order implied by a second Laplacian eigenvector has the property of an adjacency ordering if vertices with positive components are added in increasing order to $N \cup Z$, or if vertices with negative components are added in decreasing order to $P \cup Z$. However, there exist simple examples, even trees, for which the spectral ordering is not an adjacency ordering.

3. The Spectral algorithm for envelope reduction. Based on the theorems in Section 2 the following new algorithm for reducing the envelope of a sparse matrix can be formulated. Since the algorithm is based on properties of the spectrum of the Laplacian matrix L, it will be called the *spectral algorithm.* We assume throughout this section that the adjacency graph of the given matrix is connected, or that the matrix is irreducible.

ALGORITHM 1. *Spectral Algorithm*

1. *Given the sparsity structure of a matrix M, form the Laplacian matrix L.*
2. *Compute a second eigenvector $\underline{x}_2$ of L.*
3. *Sort the components of the eigenvector in nondecreasing order, and reorder the matrix M using the corresponding permutation vector. Also sort the compo-*

nents in nonincreasing order, and compute the corresponding reordering of the matrix M. Choose the permutation that leads to the smaller envelope size.

The implementation of steps 1 and 3 are relatively straightforward. The formation of the Laplacian matrix requires the computation of the degree of the nodes x_i. Step 3 is a simple sort of the entries of $\underline{x}_2$, and recording the resulting permutation of indices. This can be done quickly by any efficient sorting algorithm such as quicksort. Computationally the difficult part is step 2.

The standard algorithm for computing a few eigenvalues and eigenvectors of large sparse symmetric matrices is the Lanczos algorithm. Since the Lanczos algorithm is discussed extensively in the textbook literature [16, 26], we do not include a detailed description of the standard algorithm here. Recently, we have developed a much more efficient multilevel method for finding a second eigenvector [3]. The multilevel method requires three elements in addition to the Lanczos algorithm:

- **Contraction**: Construct a series of smaller graphs that in some sense retain the global structure of the original large graph.
- **Interpolation**: Given a second eigenvector of a contracted graph, interpolate this vector to the next larger graph in a way that provides a good approximation to an eigenvector of the larger graph.
- **Refinement**: Given an approximate eigenvector for a graph, compute a more accurate vector efficiently.

Graph contraction is accomplished by first finding a maximal independent set of vertices, which are to be the vertices of the contracted graph. The edges of the contracted graph are determined by growing domains from the selected vertices in a breadth-first manner, adding an edge to the contracted graph when two domains intersect. A series of smaller contracted graphs is constructed until the size of the vertex set is less than some number (typically 100). The Lanczos algorithm can then be used to find the eigenvector of the smallest graph very quickly. This eigenvector is then interpolated to a vector corresponding to the next larger graph. This interpolated vector yields a very good approximation to the eigenvector of the larger graph. The approximation is then refined using the Rayleigh Quotient Iteration algorithm, which, because of its cubic convergence, usually requires only one or perhaps two iterations to obtain an acceptable result. This process of interpolation and refinement is continued until the eigenvector of the original graph is determined.

4. Numerical results. This section shows numerical results for the envelope sizes and bandwidths obtained from the spectral, RCM, GPS, and GK algorithms for three sets of matrices. The first set, shown in Table 4.1, includes matrices for structural analysis applications from the Boeing-Harwell data set. The next set, shown in Table 4.2, consists of miscellaneous matrices from the Boeing-Harwell collection. Finally, the third set, shown in Table 4.3, is a selection of matrices from structural analysis used at NASA. The computations were performed on a Silicon Graphics workstation with a 33 MHZ IP7 processor.

The spectral algorithm finds the reordering with the smallest envelope in 14 out of 18 cases (as shown in the "Rank" column of the tables). In those cases in which the re-

TABLE 4.1
Results (Boeing-Harwell — Structural Analysis)

Title (equations) (nonzeros)	Envelope	Bandwidth	Run time (sec.)	Algorithm	Rank
BCSSTK13	64,486	455	3.92	SPECTRAL	4
(2,003)	58,542	223	.64	GK	3
(11,973)	57,501	145	.57	GPS	2
	56,299	198	.08	RCM	1
BCSSTK29	3,067,004	882	31.95	SPECTRAL	1
(13,992)	6,948,091	1,505	9.53	GK	2
(316,740)	7,040,998	869	5.29	GPS	3
	7,374,140	914	2.37	RCM	4
BCSSTK30	9,135,742	4,769	78.18	SPECTRAL	1
(28,924)	15,686,968	16,947	78.10	GK	2
(1,036,208)	23,242,990	2,515	61.65	GPS	3
	23,242,990	2,512	6.32	RCM	4
BCSSTK31	19,574,992	4,763	55.06	SPECTRAL	1
(35,588)	22,330,987	1,880	22.05	GK	2
(608,502)	23,416,579	1,104	9.12	GPS	3
	23,641,124	1,176	4.69	RCM	4
BCSSTK32	27,614,531	13,792	92.09	SPECTRAL	1
(44,609)	49,457,764	3,761	102.44	GK	2
(1,029,655)	50,067,390	2,339	79.48	GPS	3
	52,170,122	2,390	7.83	RCM	4
BCSSTK33	3,788,702	1,199	31.01	SPECTRAL	3
(8,738)	3,571,395	932	5.20	GK	1
(300,321)	3,717,032	519	3.22	GPS	2
	3,799,285	749	1.82	RCM	4

sult of the spectral algorithm is not the best (i.e., BCSSTK13, BKSSTK33, SHUTTLE, and CAN1072), it is still fairly close to the best result. In several cases, however, the spectral algorithm finds a reordering with an envelope substantially smaller than any of the other algorithms, sometimes by a factor of more than two. Note also that the spectral algorithm clearly outperforms the others on the larger problems in the Tables. The run time of the spectral algorithm is usually, but not always, greater than that of the other algorithms. We expect the differences in runtimes between the ordering algorithms to be smaller on computers with vector-processing capabilities, such as the Crays.

The GPS, GK, and RCM algorithms, which are all closely related, use local search (breadth-first search) from a pseudo-peripheral vertex to generate a long rooted level structure. The RCM algorithm then numbers the vertices by increasing level values,

TABLE 4.2
Results (Boeing-Harwell — Miscellaneous)

Title (equations) (nonzeros)	Envelope	Bandwidth	Run time (sec.)	Algorithm	Rank
CAN1072	55,228	301	.51	SPECTRAL	2
(1,072)	48,538	234	.20	GK	1
(6,758)	74,067	159	.13	GPS	4
	56,361	175	.05	RCM	3
POW9	29,149	264	.45	SPECTRAL	1
(1,723)	64,788	201	.14	GK	2
(4,117)	69,446	116	.10	GPS	3
	79,260	133	.05	RCM	4
BLKHOLE	120,767	426	.56	SPECTRAL	1
(2,132)	169,219	134	.17	GK	2
(8,502)	173,243	106	.12	GPS	4
	171,437	105	.07	RCM	3
DWT2680	93,907	142	.78	SPECTRAL	1
(2,680)	96,591	92	.28	GK	2
(13,853)	101,769	65	.19	GPS	3
	102,983	69	.11	RCM	4
SSTMODEL	86,635	228	2.21	SPECTRAL	1
(3,345)	104,562	125	.28	GK	2
(13,047)	110,936	83	.17	GPS	4
	105,421	88	.10	RCM	3

where the vertices in each level are numbered in nondecreasing order of their degrees. The final RCM ordering is obtained by reversing the ordering thus obtained. The GPS and GK algorithms use more sophisticated techniques to create a more general level structure by combining the information from two rooted level structures obtained from the endpoints of a pseudo-diameter in the RCM algorithm. They also use more refined numbering techniques to reduce the size of the envelope and the bandwidth. This is the reason why the latter two algorithms require more time than the RCM algorithm.

Generally the GPS algorithm yields a lower bandwidth while the GK algorithm yields a lower envelope size [14, 21]. Our results are in agreement with this conclusion. It should be pointed out that $n = 2680$ was the largest order of the problems considered in earlier work, and that the results reported here are for much larger problems.

In contrast to the above algorithms, the spectral algorithm relies on the global information in the components of a second Laplacian eigenvector. The results show that the bandwidths of the spectral reorderings are often much greater than those of the other reorderings, even when the spectral envelopes are much smaller. This can be seen in Figures 4.1 through 4.5, which show the sparse matrix structure of the original

TABLE 4.3
Results (NASA)

Title (equations) (nonzeros)	Envelope	Bandwidth	Run time (sec)	Algorithm	Rank
BARTH4	345,623	593	1.60	SPECTRAL	1
(6,019)	658,181	280	.54	GK	2
(23,492)	669,239	213	.33	GPS	3
	725,950	215	.21	RCM	4
SHUTTLE	566,496	631	2.59	SPECTRAL	3
(9,205)	531,420	92	1.12	GK	1
(45,966)	531,422	92	.93	GPS	2
	567,887	150	.32	RCM	4
SKIRT	688,924	1,021	5.14	SPECTRAL	1
(12,598)	1,013,423	425	3.20	GK	2
(104,559)	1,039,544	309	2.46	GPS	3
	1,068,993	314	.82	RCM	4
PWT	5,101,527	1,627	13.62	SPECTRAL	1
(36,519)	5,520,603	450	29.65	GK	2
(181,313)	5,638,855	340	28.27	GPS	4
	5,652,184	340	1.67	RCM	3
BODY	6,706,747	2,496	26.60	SPECTRAL	1
(45,087)	10,526,446	1,081	13.60	GK	2
(208,821)	10,658,164	667	8.42	GPS	3
	11,470,411	756	2.23	RCM	4
FLAP	10,471,456	1,784	45.90	SPECTRAL	1
(51,537)	12,367,171	1,019	24.96	GK	3
(531,157)	12,339,642	743	19.08	GPS	2
	12,598,705	874	4.19	RCM	4
IN3C	425,232,466	9,504	117.83	SPECTRAL	1
(262,620)	519,316,395	3,780	56.97	GK	2
(1,026,888)	526,302,263	2,473	26.28	GPS	3
	581,700,745	2,746	12.88	RCM	4

BARTH4 matrix and of the four reorderings considered here. A black dot indicates a nonzero element. The GK, GPS, and RCM reorderings all look very similar, whereas the SPECTRAL reordering has a quite different appearance.

Juvan and Mohar [19] had suggested the use of the spectral ordering for reducing the bandwidth (and p-sums), but our results show that the GPS algorithm is much more effective than the spectral algorithm in reducing the bandwidth. A possibility is to make limited use of a local reordering strategy based on the adjacency structure to improve the envelope parameters obtained from the spectral method. Such reordering

TABLE 4.4
Factorization times

Title	Envelope	Factor time (sec)	Algorithm
BCSSTK29	3,067,004	257	SPECTRAL
	7,374,140	1,677	RCM
BCSSTK33	3,788,702	670	SPECTRAL
	3,799,285	685	RCM
BARTH4	345,623	8.19	SPECTRAL
	725,950	35.17	RCM

strategies will be considered elsewhere since the evaluation of the various possibilities will require much effort.

Finally we list in Table 4.4 the factorization times for a few matrices, reordered with both the spectral algorithm and with RCM. These times are for the envelope factorization routine from SPARSPAK, and are measured again on a SGI workstation. We selected one example where the spectral algorithm is comparable in storage requirements to RCM (BCSSTK33), and two examples where the spectral algorithm yields considerably lower storage memory requirements. The results demonstrate the quadratic behavior of the factorization time as a function of the envelope size. Therefore we conclude that spectral reordering not only reduces the memory requirements, but also improves execution times.

REFERENCES

[1] P. R. AMESTOY AND I. S. DUFF, *Vectorization of a multiprocessor multifrontal code*, Int. J. Supercomputer Applications, 3 (1989), pp. 41 – 59.

[2] C. C. ASHCRAFT, R. G. GRIMES, J. G. LEWIS, B. W. PEYTON, AND H. D. SIMON, *Recent progress in sparse matrix methods for large linear systems*, International Journal on Supercomputer Applications, 1 (1987), pp. 10 – 30.

[3] S. T. BARNARD AND H. D. SIMON, *A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems*, Tech. Report RNR-092-033, NASA Ames Research Center, Moffett Field, CA 94035, Nov. 1992.

[4] T. F. CHAN AND W. K. SZETO, *On the near optimality of the recursive spectral bisection method for graph partitioning.* manuscript, Feb. 1993.

[5] E. H. CUTHILL AND J. MCKEE, *Reducing the bandwidth of sparse symmetric matrices*, in Proceed. 24th Nat. Conf. Assoc. Comp. Mach., ACM Publications, 1969, pp. 157–172.

[6] E. F. D'AZEVEDO, P. A. FORSYTH, AND W. P. TANG, *Ordering methods for preconditioned conjugate gradients methods applied to unstructured grid problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 944–961.

[7] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, 1986.

[8] I. S. DUFF AND G. A. MEURANT, *The effect of ordering on preconditioned conjugate gradients*, BIT, 29 (1989), pp. 635–657.

[9] M. FIEDLER, *Algebraic connectivity of graphs*, Czech. Math. J., 23 (1973), pp. 298–305.

[10] M. FIEDLER, *Algebraische zusammenhangszahl der graphen und ihre numerische bedeutung*, in Numerische Methoden bei graphentheoretischen und kombinatorischen Problemen, Oberwol-

fach 1974, ISNM, L. Collatz, H. Werner, and G. Meinardus, eds., vol. 29, Birkhauser Verlag, 1975, pp. 69–85. In German.

[11] M. Fiedler, *A property of eigenvectors of non-negative symmetric matrices and its application to graph theory*, Czech. Math. J., 25 (1975), pp. 619–633.

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

[13] J. A. George and A. Pothen, *An analysis of the spectral approach to envelope reduction via quadratic assignment problems.* In preparation, 1993.

[14] N. E. Gibbs, *Algorithm 509: A hybrid profile reduction algorithm*, ACM Trans. on Math. Software, 2 (1976), pp. 378–387.

[15] N. E. Gibbs, W. G. Poole, Jr., and P. K. Stockmeyer, *An algorithm for reducing the bandwidth and profile of a sparse matrix*, SIAM J. Num. Anal., 13 (1976), pp. 236 – 249.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1989.

[17] R. G. Grimes, D. J. Pierce, and H. D. Simon, *A new algorithm for finding a pseudoperipheral node in a graph*, SIAM J. Mat. Anal. Appl., 11 (1990), pp. 323 – 334.

[18] C. Helmberg, B. Mohar, S. Poljak, and F. Rendl, *A spectral approach to bandwidth and separator problems in graphs.* manuscript, Feb. 1993.

[19] M. Juvan and B. Mohar, *Optimal linear labelings and eigenvalues of graphs*, Discr. Appl. Math., 36 (1992), pp. 153–168.

[20] N. Knight, R. Gillian, S. McCleary, C. Lotts, E. Poole, A. Overman, and S. Macy, *CSM testbed development and large-scale structural applications*, in Science and Engineering on Cray Supercomputers, E. J. Pitcher, ed., Minneapolis, MN, 1988, Cray Research, pp. 359 – 387.

[21] J. G. Lewis, *Implementations of the Gibbs-Poole-Stockmeyer and Gibbs-King algorithms*, ACM Trans. on Math. Soft., 8 (1982), pp. 180 – 189.

[22] J. G. Lewis and H. D. Simon, *The impact of hardware gather/scatter on sparse Gaussian elimination*, SIAM J. Sci. Stat. Comp., 9 (1988), pp. 304 – 311.

[23] J. W. H. Liu, *A generalized envelope method for sparse factorization by rows*, Tech. Report CS-88-09, Department of Computer Science, York University, 1988.

[24] J. W. H. Liu and A. H. Sherman, *Comparative analysis of the Cuthill-Mckee and the reverse Cuthill-Mckee ordering algorithms for sparse matrices*, SIAM J. Num. Anal., 13, pp. 198–213, 1976.

[25] B. Mohar and S. Poljak, *Eigenvalues in combinatorial optimization.* Preprint, 1992.

[26] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, New Jersey, 1980.

[27] S. Pissanetzky, *Sparse Matrix Technology*, Academic Press, New York, 1984.

[28] E. L. Poole and A. L. Overman, *The solution of linear systems of equations with a structural analysis code on the NAS CRAY-2*, Contractor Report 4159, NASA Langley Research Center, Hampton, Virginia, December 1988.

[29] A. Pothen, H. D. Simon, and K. P. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430–452.

[30] A. Pothen, H. D. Simon, and L. Wang, *Spectral nested dissection*, Tech. Report CS-92-01, Computer Science, Pennsylvania State University, University Park, PA, 1992. Also NASA Ames Research Center Report RNR-092-003.

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

[32] H. D. Simon, P. Vu, and C. W. Yang, *Performance of a supernodal general sparse solver on the CRAY Y-MP: 1.68 GFLOPS with autotasking*, Tech. Report RNR-89/04, NASA Ames Research Center, Moffett Field, CA 94035, 1989.

[33] O. Storaasli, D. Nguyen, and T. Agarwal, *Parallel-vector solution of large-scale structural analysis problems on supercomputers*, in Proceedings of the AIAA/ASME/ASCE/AHS/ASC 30th Structures, Structural Dynamics and Materials Conference, Vol. II, Mobile, Alabama, 1989, pp. 859 – 867.

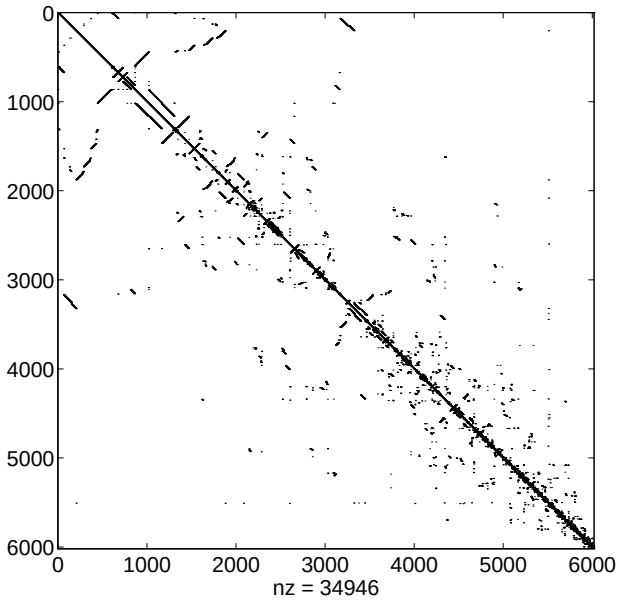

FIG. 4.1. *Structure of the original ordering of the matrix BARTH4.*

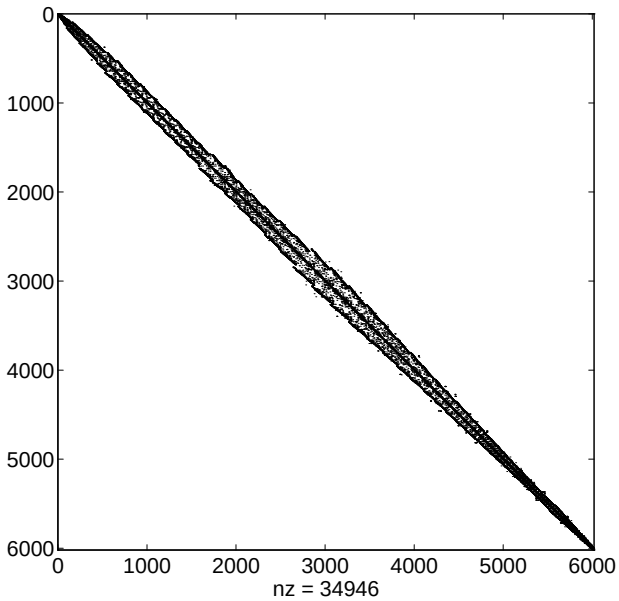

FIG. 4.2. *Structure of the Gibbs-Poole-Stockmeyer (GPS) reordering of BARTH4.*

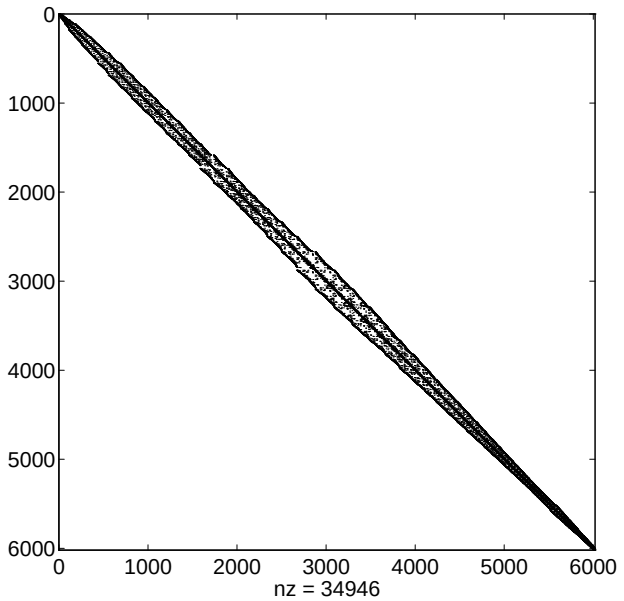

FIG. 4.3. *Structure of the Gibbs-King (GK) reordering of BARTH4.*

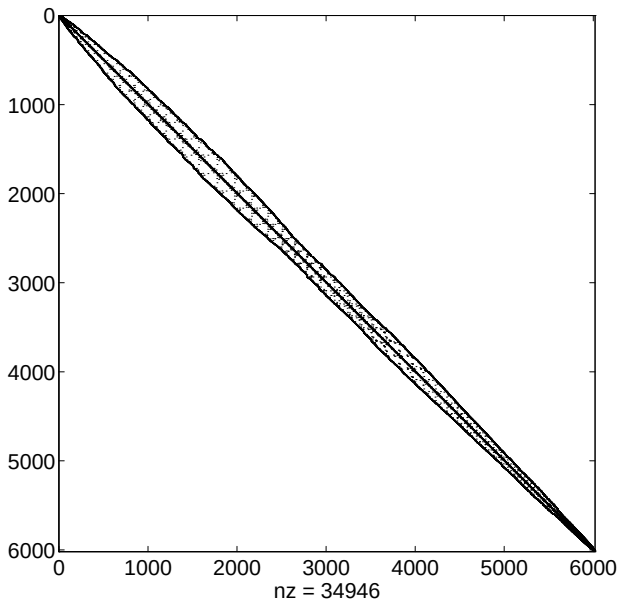

FIG. 4.4. *Structure of the Reverse Cuthill-McKee (RCM) reordering of BARTH4.*

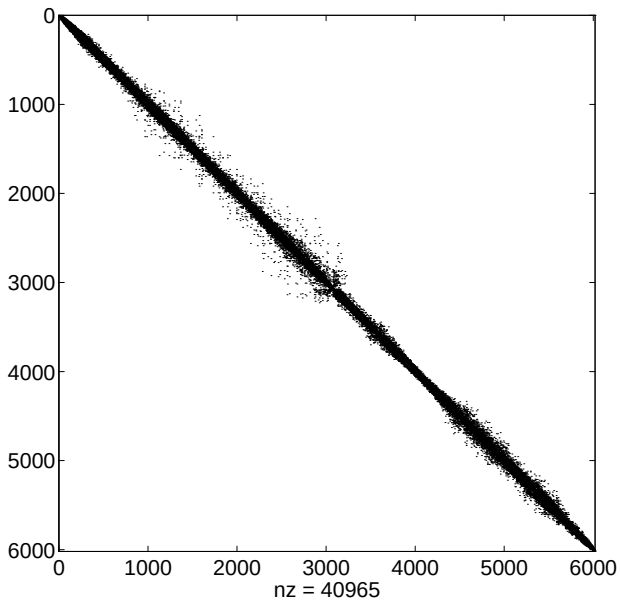

FIG. 4.5. *Structure of the Spectral reordering of BARTH4.*