(2)

Technical Report 1399
April 1991

Parallel Block Methods for Sparse Symmetric Linear Systems of Equations

A. K. Kevorkian

Approved for public release; distribution is unlimited.

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152-5000

J. D. FONTANA, CAPT, USN
Commander

H. R. TALKINGTON, Acting
Technical Director

ADMINISTRATIVE INFORMATION

This work was performed under project RS34C77 of the Computer Technology Block Program. Block NO2D is managed by Naval Ocean Systems Center under the guidance and direction of the Office of Naval Technology. The work was funded under program element 0602234N and was performed by members of Code 7304, Naval Ocean Systems Center, San Diego, CA 92152-5000.

Under authority of
J. A. Roese, Head
Signal and Information
Processing Division

MCA

SUMMARY

In this work, we present a new graph-theoretic algorithm for the purpose of exploiting parallelism in the sparsity structure of large symmetric matrices. The key objectives of the algorithm are to identify full blocks in a symmetric matrix M that can be factored independently of each other and also to keep the number of fill elements generated in the process of factoring the blocks as small as possible. The graph-theoretic algorithm has running-time proportional to the number of vertices and edges in an undirected graph, making the algorithm very suitable for extremely large sparse symmetric problems.

Using this graph-theoretic algorithm, we develop a block method for the solution of sparse symmetric linear systems of equations. This block method takes full advantage of the parallel capabilities of high-performance computers and makes good use of the standard routines in the quality linear algebra library LAPACK to perform the numerical computations in terms of Level 2 and Level 3 Basic Linear Algebra Subprograms (BLAS) operations.

There are many defense critical applications in the Navy in the areas of signal processing, structural mechanics, and computational hydrodynamics that give rise to large sparse symmetric matrices. We strongly recommend the implementation of the presented algorithms and their application to these problems.

CONTENTS

FIGURES

INTRODUCTION

The availability and widespread use of high-performance computers and quality linear algebra programs has contributed toward gaining greater understanding of the deep interplay between data movement to and from high-speed memory and linear algebra operations. Today's high-performance computer architectures incorporate complex hierarchy of memory levels with registers at the top, followed by cache and main memory at the bottom. Data movement between these memory levels is computationally costly and should be reduced as much as possible. The key strategy that has emerged from high-speed memory traffic considerations is the restructuring of matrices into blocks and to perform block matrix operations such as matrix-matrix multiplication, solving triangular systems with multiple right-hand sides, and rank-k matrix updates. These block matrix operations are referred to as Level 3 Basic Linear Algebra Subprograms (BLAS) (Anderson et al., 1990). When all the blocks are of dimension n, block algorithms provide $O(n^3)$ floating-point operations with $O(n^2)$ data movement. In sharp contrast to this, algorithms operating only on vectors of data (Level 1 BLAS) or performing matrix-vector operations (Level 2 BLAS) are incapable of attaining such high ratios of floating-point operations to memory references. Block algorithms are also suited for high-performance computers with multiple processors since all scalar, vector, and matrix operations on individual blocks may be performed in parallel.

For dense and banded matrices, highly efficient block algorithms have been developed for the solution of systems of linear equations, finding least-squares solutions of overdetermined systems of equations and solving eigenvalue problems (Anderson et al., 1990). In the case of general sparse matrices, the subject is less understood.

Let M be any structurally symmetric sparse matrix with a nonzero main diagonal. The objective of this work is to develop a block algorithm that takes full advantage of the vector and parallel capabilities of today's high-performance computers for the solution of the system of linear equations

$$Mx = b \ .$$

There are two key goals within our objective. First, we want to exploit all parallelism in the sparsity structure of the matrix M. The main aim here is to identify full blocks in M that can be factored in parallel on the different processors in a parallel machine. Second, we wish to exploit the sparsity of the matrix M, to keep the number of fill elements generated in the process of factoring the blocks as small as possible. It is worth noting that the problem of minimizing the number of fill elements in the unsymmetric case is nondeterministic polynomial-time (NP) complete (Rose and Tarjan, 1975) and thus computationally intractable.

Algorithmic research and software development over the last two decades has brought about many methods for exploiting sparsity in symmetric and unsymmetric matrices. Among these methods, the minimum degree algorithm (George and Liu, 1981; Tinney, 1969) stands out as the most popular and widely used ordering scheme for keeping the number of fill elements in symmetric matrices small. In this work, we develop a new method for exploiting the sparsity structure of symmetric matrices. The key idea is based on the graph-theoretic concept of simplicial vertex (Dirac, 1961; Lekkerkerker and Boland, 1962). Simplicial vertices were first introduced by Dirac (1961) to show that a rigid circuit graph (a chordal graph in today's graph-theoretic terminology) has a simplicial vertex. Lekkerkerker and Boland (1962) formally defined and used simplicial vertices in their study of interval graphs. Here, we use simplicial vertices to introduce the concept of an elite clique in an undirected graph G.

An elite clique in G is a clique with the following two properties: (a) every vertex in the clique is a simplicial vertex and (b) an elite clique is not a proper subset of any other elite clique in G.

Suppose G is the undirected graph of the structurally symmetric matrix M. Let $G_1, \ldots, G_k$ be any k distinct elite cliques in G. If A_{ii} is a block in M such that G_i is its graph for $i=1,\ldots,k$, then we prove the following three properties of an elite clique. First, each of the blocks $A_{11}, \ldots, A_{kk}$ can be factored without giving rise to any fill element in A. Second, the blocks $A_{11}, \ldots, A_{kk}$ can be factored in parallel independently of each other. Third, the set of all elite cliques in G is unique. From the second property of an elite clique, it becomes immediately apparent that by exploiting sparsity through the concept of elite cliques we automatically achieve our goal for exploiting parallelism in a symmetric matrix.

While the concept of elite cliques provides a solid theoretical background for the study of sparse symmetric matrices on parallel machines, there remains the algorithmic problem of efficiently constructing the elite cliques in a general undirected graph G. We carry out this objective by introducing the concept of the core of a clique. As an immediate consequence to a result we give on core cliques, we derive an algorithm that partitions the vertex set in the graph G into two disjoint sets X and S such that every elite clique in G is a connected component of the subgraph of G induced by the vertex set X. The partitioning algorithm is linear in the number of vertices and edges in G. Using this partitioning we obtain a set of cliques in G with the property that every elite clique in G is an element of this set, and furthermore, no two cliques in this set are connected by any edge in G. We call such a set of cliques an independent cliques set. Thus, a key contribution of this work is a linear algorithm that constructs an independent cliques set in G such that every elite clique in G is an element of this set. If the graph G does not contain any elite clique, we show that the ordering scheme resulting from the construction of the independent cliques set is as competitive as the minimum degree algorithm.

Given an independent cliques set of size k, we are able to restructure a structurally symmetric matrix M into a 2 by 2 block matrix

$$PMP^T = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

where P is a permutation matrix and A is a k by k block diagonal matrix with full square diagonal blocks. Using PMT^T, we derive a block algorithm for the solution of symmetric linear systems of equations with special emphasis on the case where M is a positive definite matrix. The entire block algorithm is organized to perform three types of numerical computations. First, the k square diagonal blocks of A are factored in parallel (a Level 2 BLAS operation). Second, k full triangular systems with multiple right-hand sides are solved in parallel (a Level 3 BLAS operation). Third, the Schur complement $D-CA^{-1}B$ is computed using matrix-matrix multiplications (another Level 3 BLAS operation). If the Schur complement is dense, the algorithm proceeds by solving a dense linear system of equations in which the coefficient matrix is the Schur complement. Otherwise, the matrix M is set to $D-CA^{-1}B$ and the entire process is repeated.

In summary, the block algorithm presented in this work takes full advantage of the parallel and vector capabilities of high-performance computers. Furthermore, the algorithm makes extensive use of standard routines for dense problems to perform the numerical computations in terms of Levels 2 and 3 BLAS operations.

NOTATION AND BACKGROUND

A graph $G=(V,E)$ consists of a finite, nonempty set of vertices V and a set of edges E. If the edges are ordered pairs (u,v) of vertices, G is said to be directed. If the edges are unordered pairs of distinct vertices, also denoted by (u,v), G is said to be undirected. A bipartite graph $B=(U,V,E)$ is an undirected graph consisting of two disjoint vertex sets U and V and a set of edges E such that every edge has one end point in U and the other in V. For a set of vertices U in $G=(V,E)$, the graph $G(U)=(U, E(U))$, where $E(U)=\{(u,v) \in E \mid u,v \in U\}$, is called the subgraph of G induced by the vertex set U. An induced subgraph $G(U)$ in G is called a clique if every vertex in $G(U)$ is connected to every other vertex in $G(U)$. A clique is maximal if it is not a proper subgraph of another clique. A set of cliques $G(U_1),\ldots, G(U_r)$ in $G=(V,E)$ is called a clique partition of G if the U_i's form a partition of the vertex set V. A set of vertices S is a separator of the connected graph $G=(V,E)$ if the induced subgraph $G(V-S)$ is disconnected. A set of vertices F in $G=(V,E)$ is a feedback vertex set if $G(V-F)$ is acyclic. All graphs considered in this paper are undirected.

A vertex u is said to be adjacent to another vertex v in the graph $G=(V,E)$ if $(u,v) \in E$. The set

$$adj_G u = \{v \in V-\{u\} \mid (u,v) \in E\}$$

is the set of vertices adjacent to u.

The degree of a vertex u in G, denoted by $deg_G u$, is the number of edges incident with u in G and so $deg_G u=|adj_G u|$. A set of vertices I in G is an independent (stable) set if no two vertices in I are adjacent. An independent set is maximal if it is not a proper subset of another independent set.

For a vertex u in the graph $G=(V,E)$, the deficiency of u in G is the set of edges $def_G u$ defined by

$$def_G u = \{(v,w) \mid v,w \in adj_G u, (v,w) \notin E, v \neq w\} .$$

Clearly, every vertex u in G with $deg_G \leq 1$ has empty deficiency.

For a vertex u in $G=(V,E)$, the graph

$$G_u = (V-\{u\}, E(V-\{u\}) \cup def_G u)$$

obtained by adding $def_G u$ to $G(V-\{u\})$ is called the u-elimination graph (Rose, Tarjan, and Lueker, 1976).

The concept of u-elimination graph first introduced by Parter (1961) plays a fundamental role in the study of Gaussian elimination on symmetric matrices. To establish the key matrix-theoretic property of a u-elimination graph, assume G is the graph of an n by n structurally symmetric matrix M and that the leading block A in the 2 by 2 block matrix PMT^T is a 1 by 1 nonsingular matrix. Let u be the vertex in G representing the row of A. Assuming no cancellation of nonzero elements, the following result by Parter (1961) on elimination graphs is well-known.

Lemma 1. G_u is the graph of the Schur complement $D-CA^{-1}B$.

Since the induced subgraph $G(V-\{u\})$ is the graph of the block D, by the construction of G_u it is immediate that every element of the deficiency set $def_G u$ represents a fill element in the Schur complement $D-CA^{-1}B$. Alternately, the block D and the Schur complement $D-CA^{-1}B$ have identical zero-nonzero structure if $def_G u=\emptyset$. This observation elucidates the importance of vertices with empty deficiencies in Gaussian elimination. Accordingly, we call a vertex u with $def_G u = \emptyset$ a perfect elimination vertex. In the same spirit, we call a graph without any perfect elimination vertex an *imperfect elimination graph*.

Another important property of a u-elimination graph, which immediately follows from the definition of the deficiency set $def_G u$ is stated next.

Lemma 2. The subgraph induced by the vertex set $adj_G u$ is a clique in G_u.

This simple but very useful result highlights the important fact that cliques are inherent in elimination graphs, and thus the study of cliques in graphs may contribute to the efficient solution of systems of equations by Gaussian elimination.

ELITE CLIQUES

We begin by introducing some clique-related concepts that play key roles in subsequent developments.

For any clique $G(U)$ in the graph $G=(V,E)$, the interior of $G(U)$ is the vertex set $int(U)$ defined by

$$int(U) = \{u \in U | adj_G = U - \{u\}\} \ .$$

If $int(U) \neq \emptyset$, then the subgraph induced by the vertex set $int(U)$ is called an elite clique in G.

From the construction of the vertex set $int(U)$ it is immediate that for every vertex u in $int(U)$ the induced subgraph $G(adj_G u)$ is a clique since $G(U)$ is a clique. A vertex satisfying this property has been called a simplicial vertex (Lekkerkerker and Boland (1962)). Thus, every vertex in an elite clique is a simplicial vertex. In addition to this property of an elite clique, it is easy to see that no elite clique is a proper subgraph of any other elite clique in G.

Since the set of vertices adjacent to a simplicial vertex induces a clique, by Lemma 2 it follows that a simplicial vertex is precisely a perfect elimination vertex. Thus, simplicial vertices are ideally suited for efficient Gaussian elimination. In what follows, we show that an entire elite clique shares this important property.

Let $G(U)$ be any clique in the graph $G=(V,E)$ with $|U|=m$ and let α be any ordering of the vertices in U. Suppose we wish to generate a sequence of elimination graphs using the vertices $\alpha(1)$, $\alpha(2)$, ..., $\alpha(m)$ in that order. Suppose $G_0=G$. Then, using the notion of elimination graph adopted earlier, we obtain the following sequence of graphs $G_1=(G_0)_{\alpha(1)}$, $G_2=(G_1)_{\alpha(2)}$, ..., $G_m=(G_{m-1})_{\alpha(m)}$. If we let $G_i=(V_i,E_i)$, for i=0, 1,..., m, then by the construction of an elimination graph we obtain

$$V_i = V_{i-1} - \{\alpha(i)\}$$

and

$$E_i = E_{i-1}(V_i) \cup def_{G_{i-1}}\alpha(i) \ , \qquad i = 1, \ldots, m.$$

The endmost graph $G_m = G_{|\alpha|}$ in the sequence of elimination graphs $G_1, \ldots, G_m$ is of particular interest in subsequent developments. We call $G_{|\alpha|}$ the α-elimination graph.

As each edge in the edge sets $def_{G_{i-1}}\alpha(i)$ for i=1,..., m represents a fill element, the set $F_G\alpha$ defined by

$$F_G\alpha = \bigcup_{i=1}^{m} def_{G_{i-1}}\alpha(i)$$

is called the α-fill in G. Consistent with the definition of a perfect elimination vertex in the graph G, we call a clique $G(U)$ a perfect elimination clique if

$$F_G\alpha = \emptyset$$

for any ordering α of the vertices in U. Thus, if $G(U)$ is a perfect elimination clique then

$$G_{|\alpha|} = G(V-U) \ .$$

This is an extremely desirable property in Gaussian elimination since the graph G is reduced to the induced subgraph $G(V-U)$ without creating any fill element in the matrix corresponding to G.

We are now in position to state the key sparsity-preserving property of an elite clique.

Theorem 1. Every elite clique in $G=(V,E)$ is a perfect elimination clique.

Proof

Let $G(U)$ be any elite clique in G and let α be any ordering of the vertices in U. We will prove this theorem by induction on $|U|$. Since $G(U)$ is an elite clique, every vertex in $G(U)$ is a simplicial vertex and so the first vertex $\alpha(1)$ in the ordering α is a perfect elimination vertex in G. Thus, the deficiency $def_{G_0}\alpha(1)$ of vertex $\alpha(1)$ in $G_0=G$ is the empty set. Now assume that $def_{G_0}\alpha(1) = \ldots = def_{G_{i-1}}\alpha(i)=\emptyset$ for any $i<|U|$. Then, the elimination graph G_{i-1} is a subgraph of the original graph G and so the subgraph induced by the vertex set $U'=U-\{\alpha(1), \ldots, \alpha(i)\}$ is an elite clique. This means that the vertex $\alpha(i+1)$ is a simplicial vertex in G_{i-1} and so we get $def_{G_{i-1}}\alpha(i+1)=\emptyset$. Hence, $F_G\,\alpha=\emptyset$ and the proof is complete.

$\square$

Up to now we have been exploiting sparsity from the viewpoint of minimizing fill elements. Our next task is to exploit parallelism inherent in the sparsity structure of a structurally symmetric matrix. We begin by introducing the concept of an independent cliques set.

Given any graph $G=(V,E)$, the collection of induced subgraphs

$$I_c=\{G(U)|G(U) \text{ is a clique, } G(U) \text{ is a connected component of } G(\textstyle\bigcup_{G(U)\in I_c} U)$$

is called an independent cliques set.

From the construction of the set I_c, it is immediate that for any two vertices u and v in V such that u and v are in two different cliques in I_c we have $(u,v) \notin E$. Thus, an independent cliques set in a graph G gives rise to a family of independent sets of size $|I_c|$ in G, and so the concept of independent cliques set is a natural extension of an independent set.

The next property of an elite clique shows that the exploitation of sparsity through the concept of elite cliques automatically carries out our other goal of exploiting parallelism.

Theorem 2. Let $G(U_i), \ldots, G(U_k)$ be any k distinct elite cliques in $G=(V,E)$. Then the following statements hold.

(a) For any $u_i \in U_i$, $i=1, \ldots, k$, the set $\{u_1, \ldots, u_k\}$ is an independent set of size k.

(b) $I_c=\{G(U_i)|\ i=1, \ldots, k\}$ is an independent cliques set of size k.

(c) If $k>1$, then the vertex set $V-\bigcup_{i=1}^{k} U_i$ is a separator of G.

Proof

(a) If $k=1$, there is nothing to prove. Suppose $k>1$, and let $G(U_i)$ and $G(U_j)$ be any two of the k given distinct elite cliques. Assume for contradiction that $U_i \cap U_j \neq \emptyset$. Then, for any vertex u in the set $U_i \cap U_j$ we have $deg_G u \geq |U_i|-1+|U_j-(U_i \cap U_j)|>|U_i|-1$ since $U_i \neq U_j$. This, however, is a contradiction since every vertex u in the elite clique $G(U_i)$ satisfies $deg_G u=|U_i|-1$. Thus, $U_i \cap U_j=\emptyset$, for all $i \neq j$, $i, j=1, \ldots, k$. Let u_i be any vertex in $G(U_i)$ and let u_j be any vertex in $G(U_j)$. Assume for contradiction that $(u_i, u_j) \in E$. Since $G(U_i)$ is an elite clique we have $deg_G\, u_i = deg_{G(U_i)} u_i$ and so every edge incident with u_i must be in $G(U_i)$. This means that the vertex u_j is in the clique $G(U_i)$, and so we have a contradiction since $U_i \cap U_j=\emptyset$. Hence $(u_i, u_j) \notin E$, and the proof of statement (a) is complete.

(b) By statement (a) no vertex in any elite clique $G(U_i)$ is adjacent to any vertex in another elite clique $G(U_j)$, and so we have statement (b).

(c) If $k>1$, then by statement (b) the induced subgraph $G(\bigcup_{i=1}^{k} U_i)$ is a disconnected graph, and so the vertex set $V-\bigcup_{i=1}^{k} U_i$ is a separator of G. This completes the proof.

$\square$

By statement (b) in Theorem 2, every set of elite cliques in a graph is an independent cliques set, and in view of this property of elite cliques we introduce the following definition.

For any set of elite cliques I_c in G, we call I_c an elite cliques set. Also, an elite cliques set of maximum size is called a maximum elite cliques set.

An interesting property of a maximum elite cliques set is presented in the next result.

Lemma 3. A maximum elite cliques set in $G=(V,E)$ is unique.

Proof

Let I_c be any maximum elite cliques set in G. If I_c is the empty set, we have nothing to prove. Suppose $I_c \neq \oslash$, and assume for contradiction that there exists in G another maximum independent elite cliques set I_c' such that $I_c' \neq I_c$. By statement (b) in Theorem 2, however, no two elite cliques in G have a common vertex. Thus, I_c must equal I_c'. This establishes a contradiction and completes the proof.

$\square$

Other interesting properties of elite cliques are summarized in the following series of results.

Lemma 4. For every elite clique $G(U)$ in $G=(V,E)$, there exists exactly one maximal clique $G(U')$ in G such that $U \subseteq U'$.

Proof

Assume for contradiction that there is in G a second clique $G(U'')$ with $U'' \neq U'$ such that $U \subseteq U''$. Then, for any vertex u in the set U we have $deg_G u \geq |U'|-1+|U''-(U' \cap U'')|>|U'|-1$ since $U' \neq U''$. This, however, is a contradiction since every vertex u in the elite clique $G(U)$ satisfies $deg_G u=|U'|-1$, and so the proof is complete.

$\square$

Lemma 5. Let $G(U)$ be any maximal clique in $G=(V,E)$ such that

$$int(U) = \oslash .$$

Then, no subgraph of $G(U)$ forms part of any elite clique in G.

Proof

Since the clique $G(U)$ is maximal, $G(U)$ can never be a proper subgraph of any clique in G. Thus, $G(U)$ can never be an elite clique in G since $deg_G u>|U|-1$, for all $u \in U$. Assume for contradiction that U contains a proper subset U' such that $G(U')$ is an elite clique in G. Thus, there exists in G a clique $G(W)$ such that $U'=int(W)$. Also, since $deg_G u>|U|-1$, for all $u \in U$, and $deg_G u=|W|-1$, for all $u \in U'$, we get $|W| > |U|$ and so $G(U')$ is a proper subgraph of $G(W)$. Now one of the following two cases must hold.

Case 1. $U-U'$ is a subset of W. Then $G(U)$ is a proper subgraph of $G(W)$ since $U' \subset W$ and $|W|>|U|$. This, however, is a contradiction since $G(U)$ is a maximal clique in G.

Case 2. $U-U'$ is not a subset of W. Let u be any vertex in U', and let v be any vertex in $U-U'-W$. Clearly, the edge (u,v) is in G but not in $G(W)$ since both u and v are in the clique $G(U)$

and $v \notin W$. However since $deg_G u=|V|-1$, every edge incident with u in G must be in $G(W)$ and a contradiction. This complete the proof.

$\square$

Lemma 6. Let $G(U)$ be any clique in G. Then for any vertex u in $int(U)$, the following statements are true.

(a) There exists in G a maximum independent set S with $u \in S$.

(b) There exists in G a minimum feedback vertex set F with $U-\{u\} \subseteq F$.

Proof

If $int(U)=\emptyset$, there is nothing to prove. Suppose $int(U) \neq \emptyset$, and let u be any vertex in $int(U)$. Then, by the property of $int(U)$ we have $adj_G u=U-\{u\}$ and so we get

$$\begin{aligned} deg_G u &= |adj_G u| \ , \\ &= |U| - 1 \ , \\ &= deg_{G(U)} u \ , \end{aligned}$$

But Kevorkian (1980) has already shown (Lemma 1 and Theorem 1) that both statements (a) and (b) hold if the above relation is satisfied. With this, the proof of Lemma 6 is complete.

$\square$

BORDERED BLOCK DIAGONAL MATRICES

Theorem 2 provides a worthwhile matrix theoretic interpretation. Let $G=(V,E)$ be the graph of an n by n structurally symmetric matrix M. Let $I_c=\{G(U_i)|\ i=1, \ldots, k\}$ be any independent cliques set in G such that $\bigcup_{i=1}^{k} U_i$ is a proper subset of V, and let A_{ii} be a submatrix of M such that the clique $G(U_i)$ is the graph of A_{ii} for $i=1, \ldots, k$. Then, by statement (b) in Theorem 2, there exists a permutation matrix P such that PMP^T is a $(k+1)$ by $(k+1)$ block matrix of the following form

$$PMP^T = \begin{bmatrix} A_{11} & & & & B_1 \\ & A_{22} & & & B_2 \\ & & \ddots & & \vdots \\ & & & A_{kk} & B_k \\ C_1 & C_2 & \cdots & C_k & D \end{bmatrix} .$$

Matrices having the block form of PMP^T are called bordered block diagonal matrices. In the case that I_c is an independent elite cliques set, we will refer to PMP^T as a bordered block diagonal matrix with respect to an elite cliques set. If each of the k cliques in any independent cliques set I_c consists of a single vertex, then each of the k diagonal blocks A_{ii} in PMP^T is a 1 by 1 nonzero matrix. This special case of PMP^T is called a bordered diagonal matrix. Bordered block diagonal matrices have been used extensively by researchers Branin, 1975; Chua and Chen, 1976; and Erisman, 1973, in the analysis of electrical circuits and in the solution of the equations deriving from these circuits. The origins of bordered block diagonal matrices, however, can be traced to the early works of Gabriel Kron (1958).

Let A be the k by k block diagonal matrix

$$A = \begin{bmatrix} A_{11} & & & & \\ & A_{22} & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & A_{kk} \end{bmatrix} ,$$

and let B and C be the 1 by k and k by 1 block matrices

$$B = [B_1 \ B_2 \ \ldots \ B_k]^T$$

and

$$C = [C_1 \ C_2 \ \ldots \ C_k] \ .$$

Also, let n_i denote the order of the the ith diagonal block in A, and let s denote the order of the diagonal block D. Since permutation matrices are orthogonal, $P^T = P^{-1}$, the system of equations $Mx=b$ is equivalent to the following system

$$(PMP^T) \ (Px) \ = \ (Pb) \ . \tag{1}$$

Now, let the vectors Px and Pb be partitioned as follows:

$$Px = \begin{bmatrix} y \\ z \end{bmatrix} \qquad \text{and} \qquad Pb = \begin{bmatrix} u \\ v \end{bmatrix} ,$$

where y and u are $n-s$ vectors and z and v are s vectors. Also, let the vectors y and u be further partitioned as follows:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_k \end{bmatrix} \qquad \text{and} \qquad u = \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_k \end{bmatrix} ,$$

where y_i and u_i are n_i vectors, for $i=1, \ldots, k$. Assuming now that A is nonsingular, the application of generalized Gaussian elimination to (1) results in the following special block upper triangular system

$$\begin{bmatrix} A_{11} & & & & B_1 \\ & A_{22} & & & B_2 \\ & & \cdot & & \cdot \\ & & & \cdot & \cdot \\ & & & \cdot & \cdot \\ & & & A_{kk} & B_k \\ & & & & D - CA^{-1}B \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_k \\ z \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_k \\ v - CA^{-1}u \end{bmatrix} .$$

Thus, using the bordered block diagonal matrix PMP^T, the system of equations $Mx=b$ is reduced to the following equivalent system of equations

$$\begin{aligned} (D - CA^{-1}B)z \ &= \ v - CA^{-1}u \ , \\ A_{ii} \, y_i \ &= \ u_i - B_i z \ , \end{aligned} \qquad i = 1, \ldots, k \ . \tag{2}$$

If the Schur complement $D-CA^{-1}B$ is sparse, the entire process can be repeated by setting the matrix M to $D-CA^{-1}B$ and the vectors x and b to z and $v-CA^{-1}u$ respectively. This process is repeated until the Schur complement has no worthwhile sparsity to exploit.

Hence, from the viewpoint of solving a large sparse system of linear equations, the efficient computation of the Schur complement $D-CA^{-1}B$ and preserving the sparsity of the Schur complement are two central issues in the construction of a bordered block diagonal matrix.

A BLOCK ALGORITHM FOR SYMMETRIC LINEAR SYSTEMS OF EQUATIONS

We first consider the most general case among symmetric matrices and that is when M is structurally symmetric. Subsequently, we deal with the more special case where M is a symmetric matrix. Our main assumption here is that the k by k block diagonal matrix A in PMP^T is nonsingular. When this assumption is satisfied, we can factor A into the product form

$$A = LU$$

in which L is a unit triangular matrix and U is an upper triangular matrix with nonzero diagonal entries.

Let X and Y be any two matrices with the same dimensions as the blocks B and C in PMP^T respectively. If X and Y satisfy the following triangular systems with multiple right-hand sides

$$LX = B$$

and

$$YU = C \ ,$$

then it is easy to show that PMP^T can be factored into the following block product form

$$PMP^T = \begin{bmatrix} L & O \\ Y & I \end{bmatrix} \begin{bmatrix} U & X \\ O & D-YX \end{bmatrix} \tag{3}$$

in which I is an s by s identity matrix and the O's are zero matrices. The matrix $D-YX$ is the familiar Schur complement $D-CA^{-1}B$.

With this block formulation, the computation of the Schur complement $D-YX$ takes the following algorithmic form.

procedure SCHUR:

begin

1. Factorize the diagonal block A_{ii} of A into the product form

$$A_{ii} = L_{ii}U_{ii} \ , \qquad i = 1, \ldots, k \ ,$$

where L_{ii} is a unit lower triangular matrix and U_{ii} is an upper triangular matrix.

2. Solve the triangular systems with multiple right-hand sides

$$L_{ii}X_i = B_i \quad \text{and} \quad Y_iU_{ii} = C_i \ , \qquad i = 1, \ldots, k \ ,$$

where the matrices X_i and Y_i have dimensions of B_i and C_i respectively;

3. Compute the Schur complement

$$D - CA^{-1}B = D - \sum_{i=1}^{k} Y_iX_i$$

end

The computation presented in procedure Schur can be tailored to take full advantage of both the vector and parallel capabilities of high-performance computers. To begin with, the k square symmetric matrices $A_{11}, \ldots, A_{kk}$ are factored (a Level 2 BLAS operation) in parallel using any parallel machine with multiple instruction stream—multiple data stream (MIMD) taxonomy. Next a total of $2k$ triangular systems with multiple right-hand sides $L_{11}X_1 = B_1, \ldots, L_{kk}X_k = B_k$ and $Y_1U_{11} = C_1, \ldots, Y_kU_{kk} = C_k$ are solved (a Level 3 BLAS operation) in parallel. Since the A_{ii} 's are full matrices, factors $L_{11}, \ldots, L_{kk}$ and $U_{11}, \ldots, U_{kk}$ are full triangular matrices and so the routines in the portable numerical linear algebra library LAPACK Anderson, et al. (1990) can be nicely used to do both steps 1 and 2 of the procedure Schur. At this point, the matrices $X_1, \ldots, X_k$ and $Y_1, \ldots, Y_k$ are available, and so the Schur complement $D-CA^{-1}B=D-YX$ is computed using matrix-matrix multiplications (yet another Level 3 BLAS operation).

The decomposition in (2) not only requires the computation of the Schur complement $D-CA^{-1}B$ but also the term $v-CA^{-1}u$. However, it we let w_i be a vector for the same size as u_i and replace the triangular system $L_{ii}X_i = B_i$ in step 2 of the procedure Schur by

$$L_{ii}[X_i \; w_i] \;=\; [B_i \; u_i] \;,$$

the the term $v-CA^{-1}u$ on the right-hand side of the first equation in (2) can be written as

$$v - CA^{-1}B = D - \sum_{i=1}^{k} Y_i w_i$$

Thus, if we augment the blocks D and B with the vectors v and u respectively,

$$\mathbf{D} = [\mathbf{D}v] \qquad \text{and} \qquad B = [\mathbf{B}\; u]$$

and let

$$\mathbf{X}_i \;=\; [X_i \; w_i] \;, \qquad\qquad i = 1, \ldots, k \;,$$

then the computation of $D-CA^{-1}B$ and $v-CA^{-1}u$ can be combined computing the following augmented Schur complement

$$\mathbf{D} - CA^{-1}\mathbf{B} \;=\; \mathbf{D} - \sum_{i=1}^{k} Y_i \, \mathbf{X}_i$$

We now turn our attention to the special case where M is a symmetric matrix. Then we have

$$U = L^T$$

which means the blocks Y and X are related through the following relation

$$Y = X^T \;.$$

Thus, in the case the matrix M is symmetric we obtain the following simplified factored form for PMP^T.

$$PMP^T \;= \begin{bmatrix} L & O \\ X^T & I \end{bmatrix} \begin{bmatrix} U & X \\ O & D - X^T X \end{bmatrix}.$$

This factorization reduces the amount of computations performed in the procedure Schur since in this particular case, step 2 of Schur involves the solution of k triangular systems of equations with multiple right-hand sides compared to the $2k$ required in the previous more general case.

The factorization in step 1 of the procedure Schur in both cases considered above assumes that each of the diagonal blocks of A is well-conditioned and so pivoting is unnecessary. However, if any diagonal block A_{11} in A is ill-conditioned, then pivoting becomes essential, and in such case, the factorization in step 1 of the procedure Schur takes the following product form

$$P_i \, A_{ii} \;=\; L_i \, U_{ii} \;, \qquad\qquad i \;=\; 1, \ldots, k \;,$$

where L_{ii} is a unit lower triangular matrix, U_{ii} is an upper triangular matrix with nonzero diagonal elements, and P_i is a permutation matrix (allows the necessary row interchanges needed for pivoting).

SPARSITY CONSIDERATIONS IN PMP^T

With the derivation of the block factorization (3), we next focus our attention on sparsity issues relating to the construction of a bordered block diagonal matrix. But first, we need some notation that models the locations of the nonzero elements of a matrix.

For any square or nonsquare matrix $X=[x_{ij}]$, the sparsity set of X is defined by

$$S(X) = \{(i,j) | x_{ij} \neq 0\} .$$

Thus, two matrices X and Y with identical dimensions have the same sparsity set, that is, $S(X)=S(Y)$ if and only if X and Y have identical zero-nonzero structure. When X is square and symmetric and the condition $i \neq j$ is imposed in the definition of $S(X)$, the sparsity set $S(X)$ and the nonzero structure set Nonz(X) introduced by George and Liu (1986) are identical. Worth noting is that if X is a p by q matrix and the vertices of the two vertex sets in the bipartite graph of X are labeled 1, 2,..., p and 1, 2,..., q, then the sparsity set of X is precisely the edge set in the bipartite graph of X.

By the block factorization given in (3), it is clear that the ideal bordered block diagonal matrix from the viewpoint of preserving sparsity in Gaussian elimination is the one that addresses the sparsity of the blocks L, U, X, Y, and the Schur complement $D-YX$. As we shall show in the following result, a bordered block diagonal matrix with respect to any elite cliques set addresses all these sparsity issues to the fullest.

Theorem 3. Let M be any structurally symmetric matrix and let G be the graph of M. Let P be any permutation matrix such that PMP^T is a bordered block diagonal matrix with respect to an elite cliques set I_c in G. Then the following statements are true.

(a) $S(L) \cup S(U) \subseteq S(A)$

(b) $S(X) \subseteq S(B)$

(c) $S(Y) \subseteq S(C)$

(d) $S(D-YX) \subseteq S(D)$.

Proof

(a) Let $G=(V,E)$ be the graph of PMP^T. Let $G(U_i),..., G(U_k)$ be the elite cliques in I_c and such that $G(U_i)$ is the graph of the block A_{ii} in A, for i=1,..., k. Let $G(U_i)$ be any element of I_c. Since $G(U_i)$ is a clique, the block A_{ii} is a matrix with all nonzero entries. Thus, $S(L_{ii}) \cup S(U_{ii}) \subseteq S(A_{ii})$, for i=1,..., k, and so the proof of (a) is complete. The remaining statements (b) and (d) follow immediately from Theorem 1.

$\square$

If exact cancellation does not occur, equality holds in each of the three statements in Theorem 3.

As Ortega and Voight (1985) elucidate in their elegant monograph, there are two well-known methods in computational science that lead to structured matrices that have the bordered block diagonal form. These are the substructuring techniques (Noor, Kamel, & Fulton, 1978) popularized by structural engineers, and the incomplete nested dissection (George, Pool, & Voight, 1978). The results presented in this section and especially Theorem 3, complement the findings in (George, Pool, & Voight, 1978) and (Noor, Kamel, & Fulton, 1978), and hopefully shed some newer insights into these methods.

AN ILLUSTRATIVE EXAMPLE

Consider the 10-vertex graph $G=(V,E)$ shown in figure 1 and suppose we wish to use the minimum degree algorithm for finding an ordering α of the vertices in the set V. Since $deg_G v_6=2$ and $deg_G v>2$ for any $v \in V-\{v_6\}$ we get $\alpha(1)= v_6$ and so the $\alpha(1)$-elimination graph takes the form $G_{\alpha(1)}=(V-\alpha\{(1)\}, E(V-\{\alpha(1)\}) \cup def_G\alpha(1))$ where $def_G\alpha(1)=\{(v_5,v_7)\}$. Hence, any application of the minimum degree algorithm to the graph G will definitely create a fill element in the Cholesky factorization of the matrix corresponding to G since $def_G\alpha(1) \neq \emptyset$.

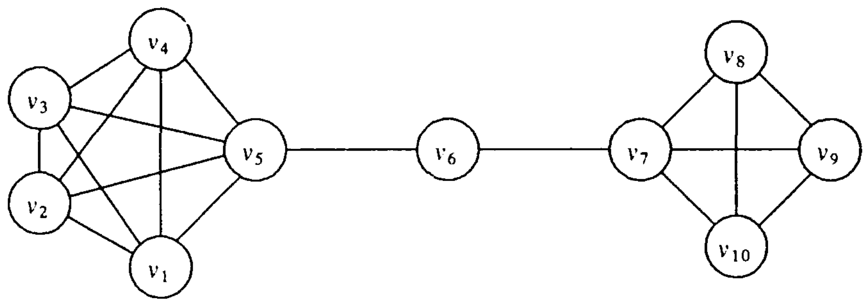


Figure 1. Ten-vertex graph $G=(V,E)$.

Next, we use the concept of elite cliques for finding an ordering of the vertices in the set V. Let $V_1=\{v_1,v_2,v_3,v_4,v_5\}$ and $V_2=\{v_7,v_8,v_9,v_{10}\}$. Then, it is easy to see from figure 1 that the induced subgraphs $G(V_1)$ and $G(V_2)$ are cliques in G and that

$$int(V_1)=V_1-\{v_5\} \quad \text{and} \quad int(V_2)=V_2-\{v_7\}.$$

Thus, the subgraphs induced by the vertex sets

$$U_1=int(V_1) \quad \text{and} \quad U_2=int(V_2)$$

are elite cliques in G and so by Theorem 1 both $G(U_1)$ and $G(U_2)$ are perfect elimination cliques. Therefore, if we let α be any ordering of the vertices in the set $U=U_1 \cup U_2=\{v_1,v_2,v_3,v_4,v_8,v_9,v_{10}\}$, then the α-elimination graph $G_{|\alpha|}=G_7$ consists of the induced subgraph $G(V-U)$, that is $G_7=G(V-U)$. Since the induced subgraph $G(V-U)$ consists of the path (v_5,v_6), (v_ϵ,v_7), it is now easy to verify that the use of the concept of elite cliques leads to a Cholesky factorization that does not suffer any fill.

THE CORE OF A CLIQUE AND MAIN RESULTS

We begin by introducing the concept of the core of a clique. This concept will subsequently be used to derive key results on the construction of elite cliques.

For a clique $G(U)$ in the graph $G=(V,E)$, the core of $G(U)$ is the vertex set $cor(U)$ defined by

$$cor(U) = \{u \in U \mid deg_G u = \min_{v \in U} deg_G v\} \ .$$

The concept of the core of a clique is a natural extension of the concept of the interior of a clique. To see this, we first express the vertex set $int(U)$ in the following form

$$int(U) \ = \ \{u \in U \mid deg_G u \ = \ |U| - 1\} \ .$$

Comparison of these two expressions reveals that

$$int(U) \ = \ cor(U)$$

if and only if

$$\min_{u \in U} deg_G u = |U| - 1 \ .$$

The fundamental property of the core of a clique is highlighted in the next result.

Theorem 4. Let $G(U_1)$, ..., $G(U_r)$ be any clique partition of the graph $G=(V,E)$. Then, for any elite clique $G(U)$ in G, the following statements are true.

(a) $U \subseteq \bigcup_{i=1}^{r} cor(U_i)$

(b) If $U \cap cor(U_i) \neq \emptyset$, for any $i = 1,\ldots, r$ then $cor(U_i) \subseteq U$.

Proof

(a) Assume for contradiction that statement (a) does not hold. Then, we have $U \cap (U_i - cor(U_i)) \neq \emptyset$ for some $i=1$, ..., r since the U_i's form a partition of the vertex set V in G. Since $G(U)$ is an elite clique, there exists in G a clique $G(C)$ such that $U=int(C)$ and so for any vertex u of $U \cap U_i$, we have $adj_G u = C - \{u\}$. But we also have $U_i - \{u\} \subseteq adj_G u$ since $u \in U_i$ and $G(U_i)$ is a clique, and so we obtain $U_i \subseteq C$. Now let v be any vertex of $cor(U_i)$. Then, by the property of $cor(U_i)$ we have $deg_G v < deg_G w$ for any $w \in U_i - cor(U_i)$. Thus, for any vertex u of $U \cap (U_i - cor(U_i))$ we get $deg_G v < deg_G u$. But we know that $deg_G u \leq deg_G w$ for all $w \in C$ since $u \in U$ and $G(U)$ is an elite clique. Consequently, we obtain $deg_G u \leq deg_G v$ since $v \in C$ and a contradiction. This completes the proof of statement (a).

(b) Let u be any vertex of $U \cap cor(U_i)$ and let v be any vertex of $cor(U_i) - U$. Then, we have $deg_G u = deg_G v$ since both u and v are vertices of $cor(U_i)$. Also, since $U_i \subseteq C$ we have $cor(U_i) \subseteq C$ and so both u and v are vertices of C. As a result, we obtain $deg_G u < deg_G v$ since $u \in U$ and $v \notin U$. This establishes a contradiction and completes the proof.

$\square$

Theorem 4 provides a number of worthwhile results and algorithms.

Corollary 4.1. The following two statements are equivalent.

(a) I_c is an elite cliques set in G.

(b) I_c is an elite cliques set in $G(\bigcup_{i=1}^{r} cor(U_i))$.

Proof

This is an immediate consequence of statement (a) in Theorem 4.

$\square$

By statement (a) in Theorem 4 or Corollary 4.1, the vertex set V in a graph G is partitioned into two disjoint sets

$$V' = \bigcup_{i=1}^{r} cor(U_i) ,$$

and

$$V'' = V - \bigcup_{i=1}^{r} cor(U_i) ,$$

such that every elite clique in G is a subgraph of $G(V')$. This partitioning of G into two subgraphs $G(V')$ and $G(V'')$ has a worthwhile matrix theoretic interpretation. In particular, if the matrix A corresponding to the graph G has decoupled blocks that can be factored in parallel and without giving rise to any fill element, then by Corollary 4.1 the vertices representing the rows of these decoupled blocks are fully contained in the induced subgraph $G(V')$.

As an illustration, consider the 11-vertex graph $G=(V,E)$ shown in figure 2. Looking at figure 2, we can easily see that the subgraphs induced by the vertex sets

$$\begin{aligned} U_1 &= \{v_1, v_2, v_3, v_4\} \\ U_2 &= \{v_5, v_6, v_7\} \\ U_3 &= \{v_8, v_9\} \\ U_4 &= \{v_{10}, v_{11}\} \end{aligned}$$

form a clique partition of the graph G.

Also it can be readily verified that

$$\begin{aligned} deg_G v_2 &< deg_G v_i , & i = 1, 3, 4 , \\ deg_G v_5 &= deg_G v_7 < deg_G v_6 , \\ deg_G v_8 &= deg_G v_9 , \\ deg_G v_{10} &= deg_G v_{11} , \end{aligned}$$

and so by the construction of the core of a clique in G we obtain

$$\begin{aligned} cor(U_1) &= \{v_2\} , \\ cor(U_2) &= \{v_5, v_7\} , \\ cor(U_3) &= \{v_8, v_9\} , \\ cor(U_4) &= \{v_{10}, v_{11}\} , \end{aligned}$$

Since $deg_G v_2 = |U_1|-1$, we have $cor(U_1)=int(U_1)$ which means $G(cor(U_1))$ is an elite clique in G. Furthermore, if we let $U' = \{v_4, v_5, v_{10}, v_{11}\}$, then it is easy to verify that $G(U)$ is a clique in G with $int(U)=\{v_{10}, v_{11}\}$. This means that the induced subgraph $G(cor(U_4))$ is also an elite clique in G.

That $G(cor(U_1))$ and $G(cor(U_4))$ are the only two elite cliques in G can be verified by applying Lemma 4. Let $U' = \{v_1, v_2, v_3, v_6\}$, $U'' = \{v_4, v_5, v_6, v_7\}$ and $U''' = \{v_6, v_8, v_9\}$. Then, it is easy to verify that each of the induced subgraphs $G(U')$, ..., $G(U''')$ is a maximal clique in G, and that

$$int(U') = int(U'') = int(U''') = \emptyset.$$

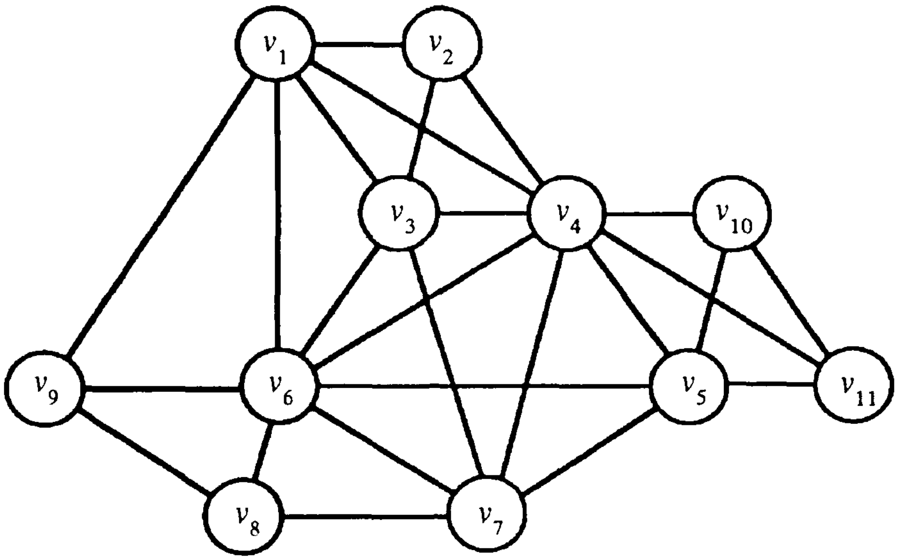


Figure 2. Eleven-vertex graph $G=(V,E)$.

So, by the application of Lemma 5 it follows that no vertex in the set

$$U' \cup U'' \cup U''' = \{v_1, v_3, v_4, v_5, v_6, v_7, v_8, v_9\} = V - cor(U_1) - cor(U_4)$$

is contained in an elite clique in G. This completes the illustration of statements (a) and (b) in Theorem 4 as well as Corollary 4.1.

The next corollary to Theorem 4 has an efficient algorithmic interpretation.

Corollary 4.2. Let (v,w) be any edge in the graph $G=(V,E)$ such that

$$deg_G v < deg_G w.$$

Then, the vertex w can never be in any elite clique in G.

Proof

Let $U_1=\{v,w\}$. Then, $G(U_1)$ is clique in G since $(v,w) \in E$, and so for any clique partition $G(U_2),\ldots, G(U_r)$ of the induced subgraph $G(V-U_1)$ the cliques $G(U_1),\ldots, G(U_r)$ form a clique partition of G. Now by the hypothesis of the corollary we have $deg_G v < deg_G w$, which means that $w \notin cor(U_1)$ and so by statement (a) of Theorem 4 the vertex w can never be in an elite clique in G. This completes the proof.

$\square$

Corollary 4.2 suggests the following way of visiting the vertices of a graph $G=(V,E)$. We select and "visit" a starting vertex v. Then, for each unexplored edge (v,w) incident with v we apply Corollary 4.2. If $deg_G v = deg_G w$, we pick the next unexplored edge incident with v since Corollary 4.2 does not apply. If $deg_G v \neq deg_G w$, then either $deg_G v < deg_G w$ or $deg_G w < deg_G v$. If the former holds, we mark w as a vertex that can never be in an elite clique, and if the latter holds, we mark v. Once all unexplored edges incident with v have been considered, we pick the next unvisited vertex in G and repeat the steps we took at the starting vertex until all vertices in G have been visited. We will call this method of visiting the vertices of an undirected graph an elite search since such a search of an undirected graph $G=(V,E)$ partitions the vertex set V into two disjoint sets R and S so that all elite cliques in G are subgraphs of the induced subgraph $G(R)$.

A procedure for elite search in a pseudo-Algol language adopted by Aho, Hopcroft and Ullman (1976) is presented next. The input for the procedure is an undirected graph $G=(V,E)$ represented by adjacency lists $L[v]$ for all $v \in V$. We assume that each vertex v in V is marked "new" and $F[v]=0$. Also we assume that the vertex sets R and S are initially set to the empty set.

```
procedure SEARCH:
begin
    while there exists a vertex v in V marked "new" do
        begin
            mark v "old";
            for each vertex w on L[v] do
                if w is marked "new" then
                    if DEG[v] ≠ DEG[w] then
                        if DEG[v]<DEG[w] then F[w] ← 1
                        else F[v] ← 1
        end;
    for all v in V do
        if F[v]=0 then
            begin
                mark v "new";
                add v to R
            end
        else add v to S
end
```

An elite search of the graph $G=(V,E)$ partitions the vertices in V into two sets R and S such that

$$S=\{w \mid \text{there exists in } G \text{ an edge } (v,w) \text{ such that } deg_G v < deg_G w\}.$$

By Corollary 4.2, this means that every elite clique in G is a subgraph of $G(R)$.

Initially, all vertices in G are marked "new." We associate with each vertex v in G a nonnegative integer $DEG[v]$ representing the degree of v in G and a Boolean integer $F[v]$. Initially, all these Boolean integers are set to zero. At the completion of the **while** loop in *SEARCH*, a Boolean integer $F[v]$ takes the value of 1 if and only if for some unexplored edge (v,w) in E the vertices v and w satisfy the inequality $DEG[w]<DEG[v]$. By Corollary 4,2, this means that each vertex v in G that can never be in an elite clique in G will have $F[v]=1$.

Suppose we apply the procedure *SEARCH* to the graph G in figure 2. Initially, all vertices in G are marked "new." Suppose the vertices $v_1, v_2, \ldots, v_{11}$ are visited in that order. We select v_1 and mark it "old." The unexplored edges incident with v_1 are (v_1, v_2), (v_1, v_3), (v_1, v_4), (v_1, v_6) and (v_1, v_9). Now since

$$\deg_G v_2 < \deg_G v_1 ,$$
$$\deg_G v_1 < \deg_G v_4 ,$$
$$\deg_G v_1 < \deg_G v_6 ,$$

the procedure *SEARCH* sets $F[v_1]=F[v_4]=F[v_6]=1$. Next, vertex v_2 is visited and marked "old." The edges incident with v_2 are (v_2, v_1), (v_2, v_3), and (v_2, v_4). The edge (v_2, v_1) has already been explored and will not be considered further since the vertex v_1 is marked "old." The remaining two edges are presently unexplored and so the procedure *SEARCH* sets $F[v_3]=1$ since $deg_G v_2 < deg_G v_3$. Next, v_3 is visited and marked "old." The unexplored edges incident with v_3 are (v_3, v_4), (v_3, v_6) and (v_3, v_7). However, since $deg_G v_3 = deg_G v_7$ and $F[v_4]=F[v_6]=1$, vertex v_4 is visited next and marked "old." The unexplored edges incident with v_4 are (v_4, v_5), (v_4, v_6), (v_4, v_7), (v_4, v_{10}), and (v_4, v_{11}). Again, the Boolean numbers associated with the vertices v_5, v_6, v_7, v_{10}, and v_{11} remain unchanged with this visit of the vertex v_4 since the degree at this vertex is greater than the degree of any adjacent vertex. It can readily be verified that the visits to the remaining vertices $v_5, v_6, \ldots, v_{11}$ brings in the following changes $F[v_5]=F[v_7]=1$ in that order. Thus, at the completion of the **while** loop we have

$$F[v_2] = F[v_8] = F[v_9] = F[v_{10}] = F[v_{11}] = 0$$

and

$$F[v_1] = F[v_3] = F[v_4] = F[v_5] = F[v_6] = F[v_7] = 1$$

which means that

$$R = \{v_2, v_8, v_9, v_{10}, v_{11}\}$$

and

$$S = \{v_1, v_3, v_4, v_5, v_6, v_7\} .$$

It is interesting to note that the induced subgraphs $G(\{v_2\})$, $G(\{v_8, v_9\})$, and $G(\{v_{10}, v_{11}\})$ constitute the connected components of $G(R)$, and furthermore, these connected components are induced by the vertex sets $cor(U_1)$, $cor(U_3)$, and $cor(U_4)$. As we shall show later on, this is not a coincidence but a property of the procedure *SEARCH*.

The next theorem and its corollaries highlight the main properties of elite search.

Theorem 5. Every elite clique in the graph $G=(V,E)$ is a connected component of the induced subgraph $G(R)$.

Proof

Let $G(U)$ be any elite clique in G. Let u be any vertex in U. If $deg_G u=0$, then there is no edge incident with u in G. Thus, $F[u] = 0$ at the completion of the **while** loop in *SEARCH*, and so u will be in R at the completion of *SEARCH*. Suppose $deg_G u>0$, and let (u,v) be any edge incident with u in G. By the construction of an elite clique, there exists in G a maximal clique $G(U')$ such that $U=int(U')$. Then either the vertex v is in the set U or $U'-U$ since by Lemma 4 every elite clique is contained in exactly one maximal clique in G. If v is in U, then by the construction of an elite clique we have $deg_G u=deg_G v$ which means that $F[v]=0$ at the completion of the **while** loop in *SEARCH* and so every vertex in U will be R at the completion of *SEARCH*. Suppose v is in $U'-U$. Then we have $U'-\{v\} \subset adj_G v$ since v is a vertex in the clique $G(U')$ and $v \notin U$. This means that $deg_G v>deg_G u$ since $adj_G u=U'-\{u\}$. Thus, we get $F[v]=1$ at the completion of the **while** loop in *SEARCH* since (u,v) is an edge in G. Hence, $(U'-U) \cap R=\emptyset$ at the completion of *SEARCH*, and so $G(U)$ must be a connected component of the induced subgraph $G(R)$. This complete the proof.

□

Corollary 5.1. If the graph $G=(V,E)$ contains more than one elite clique, then the vertex set

$$S=V-R$$

is a separator of G.

Proof

Suppose G contains more than one elite clique. Then, by Theorem 5 these elite cliques are contained in a subgraph of G induced by the vertex set $R=V-S$, and so by statement (b) of Theorem 2 the induced subgraph $G(V-S)$ is a disconnected graph. This completes the proof.

□

Corollary 5.2. Let $G=(V,E)$ be any connected graph. If every edge (u,v) in G satisfies

$$deg_G u = deg_G v,$$

then one of the following two conditions must hold:

(a) G is an elite clique.
(b) G is an imperfect elimination graph.

Proof

Suppose every edge (u,v) satisfies the hypothesis. Then, we have $R=V$ and so $G(R)$ consists of exactly one connected component since G is a connected graph. Consequently, by Theorem 5, either G is an elite clique or not. If it is an elite clique, then we have statement (a). Suppose G is not an elite clique. Then, by Theorem 5 no part of G is an elite clique, and so by Theorem 1, no vertex in G is a perfect elimination vertex. This means that G is an imperfect elimination graph [statement (b)] and as a result the proof is complete.

□

The next lemma establishes a connection between the minimum degree algorithm and elite search.

Lemma 7. Every vertex u in V with

$$deg_G u = \min_{v \in V} deg_G v$$

is contained in the set R.

□

Proof

Let u be any vertex with minimum degree in G. Then there exists no edge (u,v) in G such that $deg_G v < deg_G u$. This means that u can never be in the set S at the completion of SEARCH. So the vertex u must be in the set R since R and S form a partition of the vertex set V. This completes the proof.

□

If every connected component of the induced subgraph $G(R)$ is a clique, then the connected components of $G(R)$ form an independent cliques set I_c in G. By Theorem 5, this particular independent cliques set I_c has the property that every elite clique in G must be in I_c. In practice, however, one or more connected components of $G(R)$ may not be cliques. Thus, to construct an independent cliques set of $G(R)$, we first require a procedure to construct the connected components of $G(R)$ and test each connected component for the clique property. If a connected component $G(U)$ of $G(R)$ is a clique, there is nothing further to do. Otherwise, we require a procedure to construct an independent cliques set in $G(U)$. To do this, we need a procedure to construct a clique in $G(U)$. Let $CLIQUES(U)$ be such a procedure and let $N_G U$ be the set of vertices defined by

$$N_G U = \Big(\bigcup_{u \in U} adj_G u \Big) - U \ .$$

Let $G(U')$ be any clique constructed by the procedure $CLIQUES(U)$. Then by the construction of the vertex set $N_G U$ it is easy to see that for any clique $G(U'')$ in $G(U-U'-N_G U')$ the cliques $G(U')$ and $G(U'')$ form an independent cliques set in $G(U)$. Thus, a way of constructing an independent cliques set of $G(U)$ is to recursively construct a clique $G(U')$; set U to $U-U'-N_G U'$ and repeat these two steps until U is the empty set.

The procedure *SEARCH* combined with a procedure *COMPONENT* for constructing connected components of a graph as well as the procedure *CLIQUES* gives rise to a procedure *ICS* (independent cliques set) for finding an independent cliques set in an undirected graph $G=(V,E)$. We assume that the input graph G is not a clique.

```
procedure ICS(V):
begin
     R ← ∅;
     S ← ∅;
     for all v in V do
          begin
               mark v "new" ;
               F[v] ← 0
          end;
     SEARCH;
     NUMEDGES ← 0;
     U ← ∅;
     while there exists a vertex v in R marked "new" do
          begin
               COMPONENT(v);
               if NUMEDGES≠|U| × (|U|-1) then
                  CLIQUES(U)
          end
end

procedure COMPONENT(v):
begin
     mark v "old";
     add v to U;
     for each vertex w on L[v] do
          if F[w]=0 then
               begin
                  NUMEDGES ← NUMEDGES + 1;
                  if w is marked "new" then
                     COMPONENT(w)
               end
end
```

```
procedure CLIQUES(U):
begin
    LIST ← ∅;
    for all u in U do mark u "new";
    while there exists a vertex u in U marked "new" do
          begin
              mark u "old";
              for each v on L[u] do
                 if v is marked "new" then
                    begin
                        add v to LIST;
                        F[v] ← 1
                    end;
              while LIST≠∅ do
                 begin
                    select any vertex v in LIST and mark it "old";
                    delete v from LIST;
                    for each vertex w on L[v] do
                        if w is marked "new" then
                            if F[w]=1 then F[w] ← -1;
                            else
                                begin
                                      mark w "old";
                                      add w to S;
                                      delete w from R
                                end;
                    for each vertex w on LIST do
                        if F[w]=-1 then F[w] ← 1
                        else
                            begin
                                delete w from LIST;
                                mark w "old";
                                add w to S;
                                delete w from R
                            end
                 end
          end
end
```

At the completion of *SEARCH* in the procedure *ICS*, the vertex set R has the property that every elite clique in G is a connected component of $G(R)$. The procedure *COMPONENT* constructs the connected components of $G(R)$ using depth-first search (Aho, Hopcroft, and Ullman, 1976). The integer *NUMEDGES* defines the number of edges visited by *COMPONENT* in the process of constructing $G(U)$. Let $G(U)$ be any connected component constructed by the procedure *COMPONENT*. If $G(U)$ is a clique, then we have $|E(U)|=|U|\times(|U|-1)/2$ and so $G(U)$ is a clique if $NUMEDGES=|U|(|U|-1)$ since depth-first search visits every edge of an undirected graph exactly twice. Thus, if $NUMEDGES=|U|\times(|U|-1)$, the procedure *ICS* proceeds with the construction of the next connected component of $G(R)$. Otherwise, the procedure *CLIQUES* is called.

The main purpose of the procedure *CLIQUES*(U) is to construct an independent cliques set in $G(U)$. *CLIQUES*(U) is designed so that each pass of the outer **while** loop generates one element of the independent cliques set. The approach adopted for constructing a maximal clique in $G(U)$ is as follows. We select any vertex u in U marked "new" and use the **for** loop in the outer **while** to construct the vertex set $LIST$ defined by

$$LIST=\{v\ L[u] \mid v \text{ is marked "new" and } F[v]=1\}.$$

Let $W=\{u\}$ and suppose we insert the statement $W \leftarrow W \cup \{v\}$ immediately following the selection of the vertex v in the inner **while** loop in $CLIQUES(U)$. We will now use induction to prove that at the completion of each pass of the inner **while** loop the set W induces a clique in $G(U)$. Since $W=\{u\}$ initially and u is adjacent to each vertex in $LIST$, it follows that the vertex set $W=W \cup \{v\}=\{u,v\}$ induces a clique in $G(U)$ at the completion of the first pass of the loop. Now assume that $G(W)$ is a clique at the completion of the ith pass of the loop. Since $G(W)$ is a clique and the vertex v in W was arbitrarily chosen at the ith pass of the loop, any vertex u in the set $W-\{v\}$ is adjacent to every vertex in the set $LIST$. Our next objective is to see which vertices on $L[v]$ marked "new" are in $LIST$. Since U is a subset of the vertex set R, by the construction of R in the procedure SEARCH any vertex v marked "new" in U and has $F[v]=1$ must be in $LIST$. Thus any vertex w on $L[v]$ marked "new" and with $F[w]$ =1 must be in $LIST$. The first **for** loop in the inner **while** loop sets each vertex w marked "new" on $L[v]$ and with $F[w]=1$ to have $F[w]=-1$. Thus, any vertex w in U that has $F[w]=-1$ is adjacent to v as well as to every vertex in the set $W-\{v\}$. This means that the set $W=W \cup \{w\}$ induces a clique in $G(U)$. The second **for** loop in the inner **while** loop modifies $LIST$ so that each vertex in $LIST$ is adjacent to every vertex in W. This completes the proof of the induction. This way, the inner **while** loop generates a clique that is maximal in the subgraph of $G(U)$ consisting of the vertices marked "new."

To complete the correctness of the procedure $CLIQUES(U)$, we need to show that the cliques constructed by $CLIQUES(U)$ form an independent cliques set in $G(U)$. Let $G(W)$ be any clique constructed by the procedure $CLIQUES(U)$. Then, it is easy to verify that every vertex x in $U-W$ that is adjacent to a vertex v in W is marked "old" during the construction of the clique $G(W)$ and hence never considered for the construction of another clique in $CLIQUES(U)$. Thus, for any two vertices x and y in two different cliques constructed by $CLIQUES(U)$, we have $(x,y) \notin E(U)$ and so $CLIQUES(U)$ correctly constructs an independent cliques set in $G(U)$.

If X is the vertex set produced by applying the procedure ICS to the graph $G=(V,E)$ and I_c is the set consisting of the connected components of the induced subgraph $G(R)$, then we have the following key properties of ICS.

Theorem 6. The set I_c satisfies the following two properties:

(a) I_c is an independent cliques set in G
(b) Every elite clique in G is an element of I_c.

Proof

(a) This follows immediately from the proof we gave for the correctness of procedure $CLIQUES$.

(b) Let $G(U)$ be any elite clique in the graph G. Then, by Theorem 5, $G(U)$ is a clique in the original induced subgraph $G(R)$. Thus, the procedure $CLIQUES$ is never applied to $G(U)$ and so $G(U)$ remains a connected component of $G(R)$ at the completion of ICS. With this the proof of statement (b) is complete.

□

Let T be the vertex set defined by

$$T = S - \cup_{G(I \quad c} N_G U ,$$

where $S=V-R$. The next result ignores some sparsity issues to increase the size of the independent cliques set constructed in procedure *ICS*.

Corollary 6.1. For any independent cliques set I in the induced subgraph $G(T)$, the set

$$I_c' = I_c \cup I$$

is an independent cliques set in G.

Proof

This is a direct consequence of Theorem 6 and the definition of the vertex set T.

$\square$

CONTRACTIONS WITH RESPECT TO AN INDEPENDENT CLIQUES SET

Let I_c be any independent cliques set in the graph $G=(V,E)$ and let α be any ordering of the vertices in the set $\bigcup_{G(U) \in I_c} U$. Our main objective is to derive explicit and implicit forms of the α-elimination graph $G_{|\alpha|}$. We begin by considering a single clique in G.

Theorem 7. Let $G(U)$ be any clique in the graph $G=(V,E)$ and let α be any ordering of the vertices in U. If $G_0=G$ and $G_i=(G_{i-1})_{\alpha(i)}$ for $i=1,\ldots,|\alpha|-1$, then

$$adj_{G_i}\alpha(k) - U = (adj_G\alpha(k) - U) \bigcup \left(\bigcup_{j=1}^{i} (adj_G\alpha(j) - U) \right), k = i+1, \ldots, |\alpha| .$$

Proof

We will prove this by induction on $|\alpha|$. Since $G(U)$ is a clique, we have $(\alpha(1), \alpha(k)) \in E$ for $k=2, \ldots, |\alpha|$. Thus, for any vertex u in $adj_G\alpha(1)-U$ we have $adj_G\alpha(1)-U \subseteq adj_{G_1}\alpha(k)-U$ for $k=2, \ldots, |\alpha|$ and so we obtain $adj_{G_1}\alpha(k)-U=(adj_G\alpha(k)-U) \cup (adj_G\alpha(1))-U)$ for $j=k, \ldots, |\alpha|$. Now assume that the elimination graph G_{i-1} satisfies the above equality, that is

$$adj_{G_{i-1}}\alpha(k) - U = (adj_G\alpha(k) - U) \bigcup \left(\bigcup_{j=1}^{i-1} (adj_G\alpha(j) - U) \right), \quad k = i, \ldots, |\alpha| .$$

Now since $G(U)$ is a clique, we have $\big(\alpha(i), \alpha(k)\big) \in$ E for $k = i+1, \ldots, |\alpha|$ and so for any vertex u in $adj_{G_{i-1}}\alpha(i)-U$ we get $adj_{G_{i-1}}\alpha(i)-U \subseteq adj_{G_i}\alpha(k)-U$ for $k = i+1, \ldots, |\alpha|$. This means that

$$adj_{G_i}\alpha(k) - U = (adj_G\alpha(k) - U) \bigcup (adj_{G_{i-1}}\alpha(i) - U) , \qquad k = i+1, \ldots, |\alpha| .$$

Combining the above two equalities completes the proof of the theorem.

□

As a direct consequence of Theorem 7 we have the following result.

Corollary 7.1. The α-elimination graph $G_{|\alpha|}=(V_{|\alpha|},E_{|\alpha|})$ satisfies

$$V_{|\alpha|}=(V-U)$$

and

$$E_{|\alpha|} = E(V_{|\alpha|}) \bigcup \Big(\bigcup_{v,w \in N_G U} \{(v,w)\} \Big) , \qquad v \neq w .$$

Proof

Let $m=|U|$. Then $|\alpha|=m$, and so by Theorem 7 we obtain

$$\begin{aligned} adj_{G_{m-1}}\alpha(m) &= \bigcup_{j=1}^{m} (adj_G\alpha(j) - U) , \\ &= N_G U . \end{aligned}$$

Hence, by Lemma 2 the set of vertices $N_G U$ form a clique in the elimination graph G. This completes the proof since $G_m = G_{|\alpha|}$.

Applying Corollary 7.1 to every element of an independent cliques set leads to the next result.

Corollary 7.2. Let I_c be any independent cliques set in the graph $G=(V,E)$ and let a be any ordering of the vertices in the set $\bigcup_{G(U) \in I_c} U$. Then, the α-elimination graph $G_{|\alpha|}=(V_{|\alpha|}, E_{|\alpha|})$ satisfies

$$V_{|\alpha|} = V - \bigcup_{G(U) \in I_c} U \ .$$

$$E_{|\alpha|} = E(V_{|\alpha|}) \ \bigcup_{G(U) \in I_c} \Big(\bigcup_{v, w \in N_G U} \{(v, w)\} \Big) \ , \qquad v \neq w \ .$$

Proof

Since the elements of the independent cliques set I_c are the connected components of the subgraph induced by the vertex set $\bigcup_{G(U) \in I_c} U$, the proof follows immediately from Corollary 7.1

□

If α' and α'' are any two distinct orderings of the same set of vertices in a graph $G=(V,E)$, then the α'-elimination graph $G_{|\alpha''|}$ may not be the same as the α''-elimination graph $G_{|\alpha''|}$. In sharp contrast to this, the α-elimination graph $G_{|\alpha|}$ for any ordering α of the vertices in the set $\bigcup_{G(U) \in I_c} U$ is unique. Moreover, Corollary 7.2 suggests that the α-elimination graph $G_{|\alpha|}$ can be obtained in parallel.

To illustrate Theorem 7 and its corollaries, we consider the 15-vertex graph $G=(V,E)$ shown in figure 3. This graph is slightly more challenging than the one we considered earlier in figure 2.

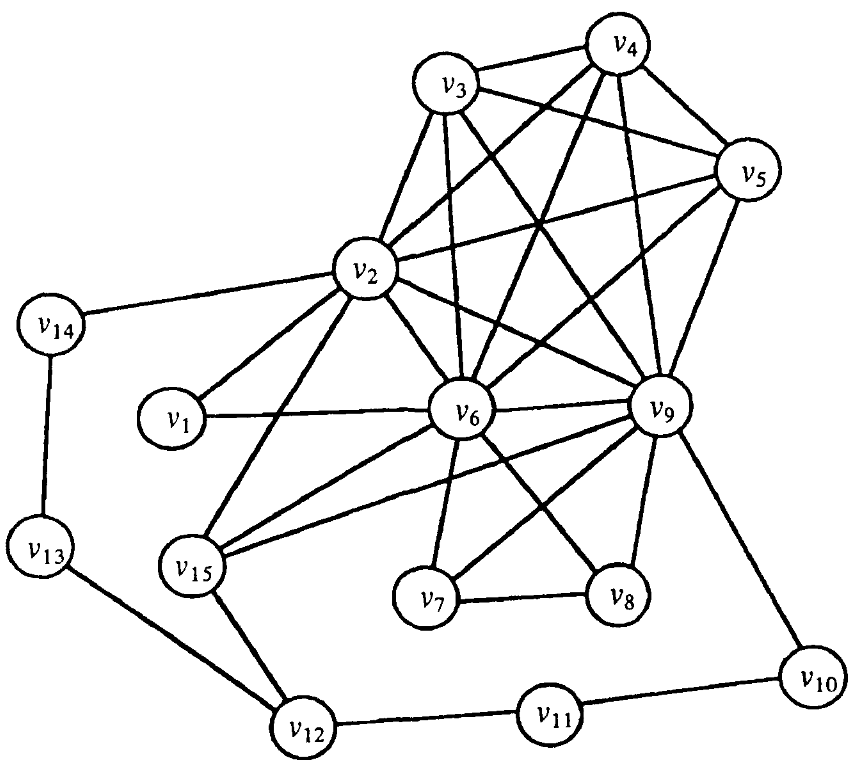


Figure 3. Fifteen-vertex graph $G=(V,E)$.

It can be easily verified that the application of the procedure *ICS* to the graph G produces the sets

$$X = \{v_1, v_3, v_4, v_5, v_7, v_8, v_{10}, v_{11}, v_{13}, v_{14}\}$$

and

$$S = V - X = \{v_2, v_6, v_9, v_{12}, v_{15}\} .$$

There are five connected components in the induced subgraph $G(X)$. The vertex sets inducing these five connected components are

$$\begin{aligned} U_1 &= \{v_1\} , \\ U_2 &= \{v_3, v_4, v_5\} , \\ U_3 &= \{v_7, v_8\} , \\ U_4 &= \{v_{10}, v_{11}\} , \\ U_5 &= \{v_{13}, v_{14}\} . \end{aligned}$$

From figure 3 it is easy to see that each of the five connected components of the induced subgraph $G(X)$ is a clique. Furthermore, we can show that each of the three connected components, $G(U_1)$, $G(U_2)$ and $G(U_3)$ is an elite clique in G. There are no other elite cliques in G. Thus, the procedure *ICS* generates the independent cliques set

$$I_c = \{G(U_i) \mid i=1, \ldots, 5\}$$

that has the maximum elite cliques set $\{G(U_1), G(U_2), G(U_3)\}$ in G as its subset.

Since $G(U_1)$, $G(U_2)$, and $G(U_3)$ are elite cliques, by Corollary 1.1 the elimination of the vertices in these three elite cliques in any order will not create any fill edge. So we will focus our attention on the remaining cliques $G(U_4)$ and $G(U_5)$ in G.

First, we apply Theorem 7 to the cliques $G(U_4)$ and $G(U_5)$. Let $\alpha'=(v_{10}, v_{11})$ and $\alpha''=(v_{13}, v_{14})$ be orderings of the vertices in the sets U_4 and U_5 in G respectively. Since $adj_G\alpha'(1)-U_4=\{v_9\}$, by Theorem 7 the $\alpha'(1)$-elimination graph G_1 satisfies

$$adj_{G_i}\alpha'(2) - adj_G\alpha'(2) - U_4 = adj_G\alpha'(1) - U_4 = \{v_9\} .$$

Similarly, the $\alpha''(1)$-elimination graph G_1 satisfies

$$adj_{G_1}\alpha''(2) - adj_G\alpha''(2) - U_5 = adj_G\alpha''(1) - U_5 = \{v_{12}\} .$$

Thus, (v_9, v_{11}) and (v_{12}, v_{14}) are the two fill edges that result from the elimination of the vertices v_{10} and v_{13} respectively.

Next, we apply Corollary 7.1 to the cliques $G(U_4)$ and $G(U_5)$. Since the neighborhoods of the vertex sets U_4 and U_5 are

$$\begin{aligned} N_G U_4 &= \{v_9, v_{12}\} , \\ N_G U_5 &= \{v_2, v_{12}\} , \end{aligned}$$

by Corollary 7.1 it follows that the elimination graphs $G_{|\alpha'|}$ and $G_{|\alpha''|}$ satisfy

$$E_{|\alpha'|} = E(V - U_4) \bigcup \{(v_9, v_{12})\} .$$

and

$$E_{|\alpha''|} = E(V - U_5) \bigcup \{(v_2, v_{12})\} .$$

Thus, if we let α denote any ordering of the vertices in the set $\bigcup_{i=1}^{5} U_i = R$, then by Corollary 7.2 it follows that the α-elimination graph $G_{|\alpha|}$ satisfies

$$V_{|\alpha|} = S$$

and

$$E_{|\alpha''|} = E(S) \bigcup \{(v_2, v_{12}), (v_9, v_{12})\} .$$

Hence, (v_2, v_{12}), (v_9, v_{11}) and (v_9, v_{12}), and (v_{12}, v_{14}) are the four fill edges that result from the elimination of the 10 vertices in the vertex R in any arbitrary order. Figure 4 shows the α-elimination graph $G_{|\alpha|}=(V_{|\alpha|}, E_{|\alpha|})$. The fill edges (v_2, v_{12}) and (v_9, v_{12}) are shown in bold lines.

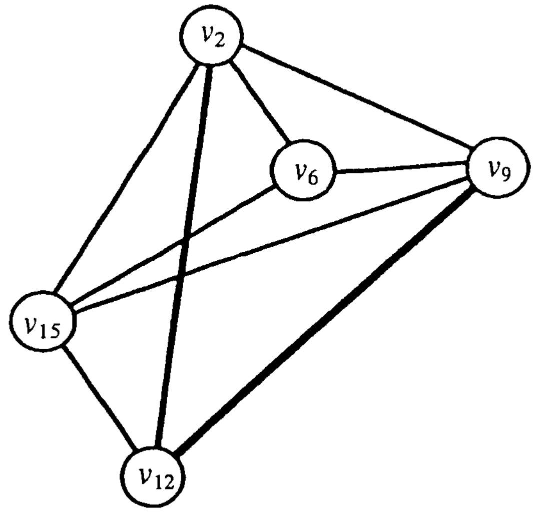


Figure 4. The α-elimination graph $G_{|\alpha|} = (V_{|\alpha|}, E_{|\alpha|})$.

Once the α-elimination graph $G_{|\alpha|}$ is computed, the graph G may be replaced by $G_{|\alpha|}$ and the entire process repeated until $G_{|\alpha|}$ is a clique. Using $ICS(V)$ as the basic procedure, this recursive version of constructing independent cliques sets and elimination graphs takes the form given below. Again, we assume that the input graph $G=(V,E)$ to the procedure *NESTEDICS* is not a clique. Also, we assume that the adjacency lists $L[v]$ and the integers $DEG[v]$ are known for all v in V at each pass of the **repeat** loop in *NESTEDICS*.

```
procedure NESTEDICS:
repeat
    begin
        ICS(V);
        for each connected component G(U) of G(R) do
            begin
                let α be any ordering of the vertices in U;
                let G|α|=(V|α|),E|α| be the α-elimination graph;
                V ← V|α|;
                E ← E|α|
            end
    end
until G(V) is a clique
```

The correctness of the procedure *NESTEDICS* follows from the correctness of the procedure *ICS* and Corollaries 7.1 and 7.2.

The first pass of the **repeat** loop in the procedure *NESTEDICS* has already been illustrated. The output at the completion of this first pass consists of the five-vertex α-elimination graph $G_{|\alpha|}$ shown in figure 4. Thus, at the start of the second pass of the **repeat** loop, we have $G=G_{|\alpha|}$.

The application of the procedure *SEARCH* to the graph in figure 4 yields $F[v_6]=F[v_{12}]=0$ and $F[v_2]=F[v_9]=F[v_{15}]=1$. Thus, at the completion of the procedure *ICS* at the second pass of the **repeat** loop in *NESTEDICS* produces the vertex set

$$R=\{v_6, v_{12}\} .$$

The connected components of the induced subgraph $G(R)$ are $G(U_1')$ and $G(U_2')$, where $U_1'=\{v_6\}$ and $U_2'=\{v_{12}\}$. If we let $U'=\{v_2, v_6, v_9, v_{15}\}$ and $U''=\{v_2, v_9, v_{12}, v_{15}\}$, we get $int(U')=U_1'$ and $int(U'')=U_2'$, which means that both of the induced subgraphs $G(U_1')$ and $G(U_2')$ are elite cliques in G.

At the completion of the second pass of the **repeat** loop, we have $V=\{v_2, v_9, v_{15}\}$ and as a result, the procedure *NESTEDICS* halts since $G(V)$ is a clique. Note that the second pass of the **repeat** loop in NESTEDICS did not give rise to any fill edge since both of the connected components $G(U_1')$ and $G(U_2')$ of the induced subgraph $G(R)$ were elite cliques.

MATRIX INTERPRETATION OF THE PROCEDURE NESTEDICS

Let M be any structurally symmetric matrix with a nonzero main diagonal and let $G=(V,E)$ be the graph of M. Suppose *NESTEDICS* is applied to the graph $G=(V,E)$ and let $I_c^{(i)}$ denote the independent cliques set consisting of the connected components of the induced subgraph $G(R)$ at the ith pass of the **repeat** loop in *NESTEDICS*. Also assume that *NESTEDICS* terminates at the completion of the $(r-1)$th pass of the **repeat** loop. Then, in matrix terms, the procedure *NESTEDICS* restructures M so that for some permutation matrix P, $PMP^T=[M_{ij}]$ is an r by r block matrix satisfying the following two properties:

(a) M_{ii} is an $|I_c^{(i)}|$ by $|I_c^{(i)}|$ block diagonal matrix with full diagonal blocks, $i=1, \ldots, r-1$.

(b) M_{rr} is a full matrix.

By property (a) the restructured matrix PMP^T is a bordered block diagonal matrix with respect to the independent cliques set $I_c^{(i)}$. If we delete the first k block rows and columns of PMP^T, then it is easy to verify that the remaining $(r-k)$ by $(r-k)$ block matrix is a bordered block diagonal matrix with respect to the independent cliques set $I_c^{(k+1)}$ for $k=1, \ldots, r-2$. Thus, in matrix terms the procedure *NESTEDICS* restructures the matrix M so that PMP^T is a nested bordered block diagonal matrix.

To illustrate this, consider the 15 by 15 matrix M shown in figure 5. It is easy to verify that the graph $G=(V,E)$ in figure 3 is the graph of the matrix M in figure 5.

$$
\left[\begin{array}{ccccccccccccccc}
x & x & & & & x & & & & & & & & & \\
x & x & x & x & x & x & & & x & & & & & x & x \\
 & x & x & x & x & x & & & x & & & & & & \\
 & x & x & x & x & x & & & x & & & & & & \\
 & x & x & x & x & x & & & x & & & & & & \\
x & x & x & x & x & x & x & x & x & & & & & & x \\
 & & & & & x & x & x & x & & & & & & \\
 & & & & & x & x & x & x & & & & & & \\
 & x & x & x & x & x & x & x & x & x & & & & & x \\
 & & & & & & & & x & x & x & & & & \\
 & & & & & & & & & x & x & x & & & \\
 & & & & & & & & & & x & x & x & & x \\
 & & & & & & & & & & & x & x & x & \\
 & x & & & & & & & & & & & x & x & \\
 & x & & & & x & & & x & & & x & & & x
\end{array}\right]
$$

Figure 5. The 15 by 15 matrix M.

Since $G(U_1)$, ..., $G(U_5)$ are the connected components of $G(R)$ at the first pass of the **repeat** loop in *NESTEDICS* and $G(U_1')$ and $G(U_2')$ are the connected components of $G(R)$ at the second pass, we have $|I_c^{(1)}|=5$ and $I_c^{(2)}|=2$. Also, we have $r=3$ since at the completion of the second pass of the **repeat** loop the subgraph induced by the vertex set $V=\{v_2, v_9, v_{15}\}$ is a clique. Thus, there exists a permutation matrix P such that $PMP^T=[M_{ij}]$ is a 3 by 3 block matrix while M_{11}, M_{22} and M_{33} are 5 by 5, 2 by 2, and 1 by 1 block diagonal matrices respectively with full diagonal blocks. The block matrix PMP^T is shown in figure 6.

$$
\begin{array}{r}
1 \\ 3 \\ 4 \\ 5 \\ 7 \\ 8 \\ 10 \\ 11 \\ 13 \\ 14 \\ 6 \\ 12 \\ 2 \\ 9 \\ 15
\end{array}
\left[\begin{array}{cccccccccc|cc|ccc}
x & & & & & & & & & & x & & x & & \\
 & x & x & x & & & & & & & x & & x & x & \\
 & x & x & x & & & & & & & x & & x & x & \\
 & x & x & x & & & & & & & x & & x & x & \\
 & & & & x & x & & & & & x & & & x & \\
 & & & & x & x & & & & & x & & & x & \\
 & & & & & & x & x & & & & & & x & \\
 & & & & & & x & x & & & & x & & & \\
 & & & & & & & & x & x & & x & & & \\
 & & & & & & & & x & x & & & x & & \\
\hline
x & x & x & x & x & x & & & & & x & & x & x & x \\
 & & & & & & & x & x & & & x & & & x \\
\hline
x & x & x & x & & & & & & x & x & & x & x & x \\
 & x & x & x & x & x & x & & & & x & & x & x & x \\
 & & & & & & & & & & x & x & x & x & x
\end{array}\right]
$$

Figure 6. Restructured PMP^T.

If we factor the structurally symmetrix matrix PMP^T so that

$$PMP^T = LU,$$

then the L and U factors take the forms in figure 7.

The eight dark circles in figure 7 correspond to the four fill edges (v_{12}, v_{14}), (v_2, v_{12}), (v_9, v_{11}), and (v_9, v_{12}). Ignoring these eight fill elements, the L and U factors have precisely the forms of the lower and upper triangular part of the restructured matrix PMP^T.

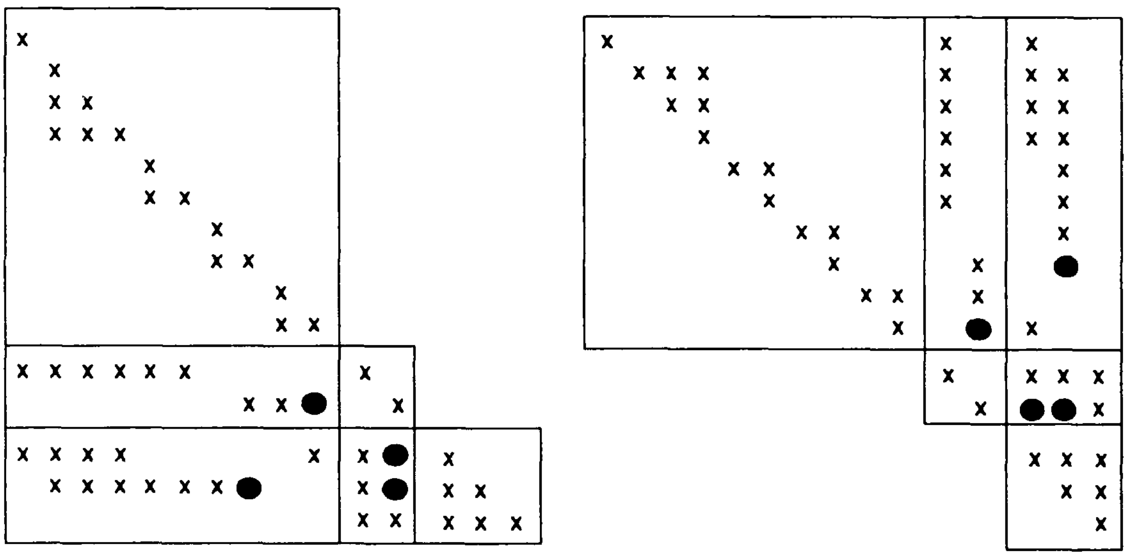


Figure 7. The L and U factors of PMP^T.

IMPLICIT FORM OF AN ELIMINATION GRAPH

Computer implementation of elimination graphs having the explicit form given in Corollaries 7.1 and 7.2 can be very costly. George and Liu (1981) cover this subject in great depth and detail. Explicit elimination graphs suffer two disadvantages. First, they require sophisticated data structures to accommodate possible changes in the structure of a graph. Second, additional space is required to store the fill elements that can be large. Another difficulty is the unpredictability of the amount of storage needed at various stages of the elimination.

George and Liu (1981) advocate the use of implicit elimination graphs to eliminate these disadvantages. Implicit elimination graphs have small and predictable storage needs and use the original adjacency list representation of the graph throughout the elimination process. The key idea in the implicit model is based on the concept of reachability.

For any set of vertices U in the graph $G=(V,E)$ and for any vertex s in the set $V-U$, the reachability of s in G with respect to the set U is the vertex set $Reach_G(s,U)$ defined by

$$Reach_G(s,U) = \{v \in V-U-\{s\} \mid \text{there exists a path from } v \text{ to } s \text{ in } G(U \cup \{s,v\})\}.$$

In other words, all information about the structure of an elimination graph can be extracted from the original graph through the concept of reachability.

While the concept of reachability eliminates the major drawbacks of an explicit representation of an elimination graph, the amount of work required to traverse paths in the original graph to generate a reachability set can be costly in the case when the vertex set U is large. To circumvent this problem, George and Liu (1981) advocate the use of quotient graphs. In subsequent developments, we borrow from this idea to facilitate the computation of implicit elimination graphs.

Let I_c be any independent cliques set in the graph $G=(V,E)$. Let $G(U)$ be any element of I_c and let $S=V-U$. Since $G(U)$ is a clique and thus a connected graph, it is easy to verify that the reachability of any vertex $s \in N_G U$ with respect to the vertex set U satisfies the following relation.

$$Reach_G(s,U)=adj_{G(V-U)}s \cup (N_G U-\{s\}).$$

If U is a large set, the construction of $Reach_G(s,U)$ may require the traversal of long paths in G. As we show next, a minor modification of the graph G solves this problem.

Suppose we introduce a single vertex c, an edge set E_c defined by

$$E_c=\{(s,c)\,|\,s \in N_G U\}\ ,$$

and the graph

$$G' = (S \cup \{c\},\ E(S) \cup E_c).$$

Then, by the construction of the edge set E_c it is immediate that the reachability of the vertex s in the graph G' is given by

$$Reach_{G'}(s,\{c\})=adj_{G(V-U)}s \cup (N_G U-\{s\}),$$

and so we get

$$Reach_{G'}(s,\{c\})=Reach_G(s,U), \tag{4}$$

for all $s \in N_G U$. Thus, the reachability of a vertex s in the set $N_G U$ remains unchanged if the original graph G is replaced by a combination of the induced subgraph $G(S)$, the vertex c and the edge set E_c.

The graph G' is extremely attractive from computer implementation standpoint. Every path traversed in G' to construct $Reach_{G'}(s,U)$ is either of length one or two whereas the paths traversed in G to construct $Reach_G(s,U)$ can be arbitrarily long depending on the size of U. Thus, replacing the original graph G by G' speeds up the computation of the reachability set.

Since the elements of an independent cliques set are disconnected cliques in the induced subgraph $G(X)$, the transformation of G to G' can be naturally extended to account for all the elements of an independent cliques set. This is our next objective.

Suppose $|I_c|=k$. Let $G(U_1)$, ..., $G(U_k)$ be the k elements of I_c and let

$$S=V-(U_1 \cup \dots \cup U_k).$$

Following the previous approach, suppose we introduce a vertex set C consisting of k vertices:

$$C=\{c_1,\ \dots,\ c_k\},$$

an edge set E_c with k disjoint parts:

$$E_c=E_{c_1} \cup \dots \cup E_{c_k}\ ,$$

where

$$E_{c_i}=\{(s,c_i)\mid s\in N_G U_i\}, \qquad i=1,\ \dots,\ k,$$

and the graph

$$G'=\left(V(S)\cup C,\ E(S)\cup E_c\right).$$

Then, by the property that the elements of the independent cliques set are disconnected in the induced subgraph $G(V-S)$ it follows

$$Reach_{G'}(s,C)=Reach_G(s,(U_1\cup\dots\cup U_k)), \tag{5}$$

for all $s\in S$. The similarities between relations (4) and (5) are obvious.

Note that the graph G' contains two distinct sets of vertices S and C. Each vertex in the set S is a vertex in the original graph G, whereas each vertex in C represents an element of the independent cliques set I_c. Since each element of I_c is a clique, we call a vertex in C a clique vertex. Similarly, we call a vertex in the set S a separator vertex since all the elements of I_c are disconnected graphs in $G(V-S)$. Also note that no edge in the graph G' connects two clique vertices, and in view of this property and the fact that G' consists of two distinct types of vertex sets, we call G' the semibipartite form *of* G with respect to the independent cliques set I_c. George and Liu (1981) called a minor variant of G' a quotient graph while a clique vertex is referred to as a supernode.

Figure 8(a) shows the semibipartite form of graph G in figure 3 with respect to the independent cliques set $I_c^{(1)}$ while figure 8(b) shows the semibipartite form of the elimination graph $G_{|\alpha|}$ in figure 4 with respect to the independent cliques set $I_c^{(2)}$. The bold lines in figures 8(a) and 8(b) denote the edges in the edge set E_c. From figures 3, 4 and 8(a) it is easy to verify the following:

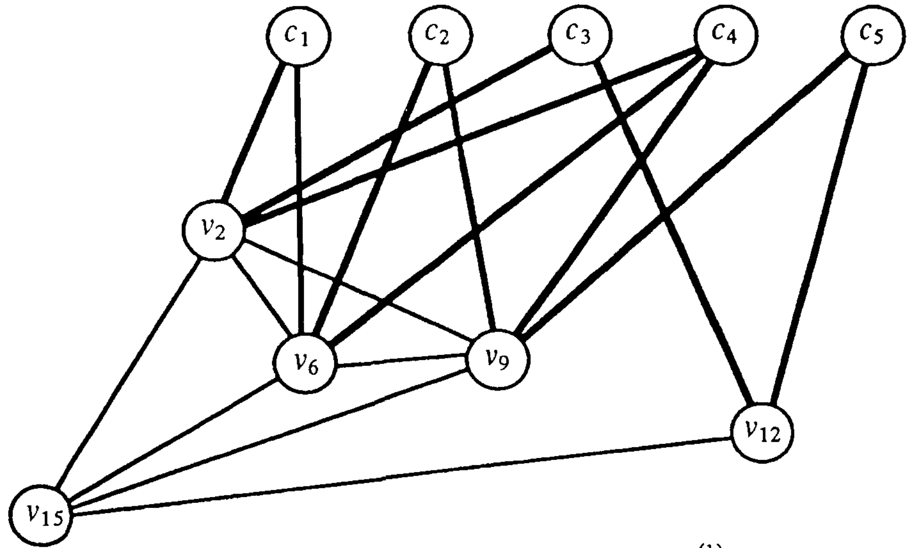
$$\begin{aligned}
Reach_G(v_2,R) &= Reach_{G'}(v_2,C) = adj_{G_{|\alpha|}}v_2 = \{v_6,v_9,v_{12},v_{15}\}\ ,\\
Reach_G(v_6,R) &= Reach_{G'}(v_2,C) = adj_{G_{|\alpha|}}v_6 = \{v_2,v_9,v_{15}\}\ ,\\
Reach_G(v_9,R) &= Reach_{G'}(v_9,C) = adj_{G_{|\alpha|}}v_9 = \{v_2,v_6,v_{12},v_{15}\}\ ,\\
Reach_G(v_{12},R) &= Reach_{G'}(v_{12},C) = adj_{G_{|\alpha|}}v_{12} = \{v_2,v_9,v_{15}\}\ ,\\
Reach_G(v_{15},R) &= Reach_{G'}(v_{15},C) = adj_{G_{|\alpha|}}v_{15} = \{v_2,v_6,v_9,v_{12}\}\ .
\end{aligned}$$

Note that the reachability sets given above are computed by traversing paths of length one or two in the semibipartite graph G'.

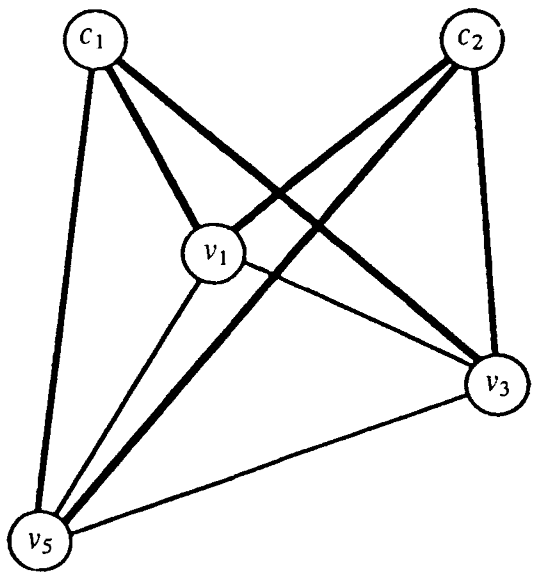
The graph G' in figure 8(a) nicely serves for demonstrating Corollary 6.1. Since the vertex v_{15} is not adjacent to any clique vertex in G', it follows that v_{15} is an element of the set $S-(N_G U_1\cup\dots\cup N_G U_5)$. Thus, if we let $U_6=\{v_{15}\}$, then by Corollary 6.1 it follows that the set of cliques $I_c\cup\{G(U_6)\}$ is an independent cliques set in G.

Let I_c be the independent cliques set consisting of the connected components of the subgraph induced by the vertex set R generated by the procedure *ICS*. Also, let $G'=(V(S)\cup C, E(S)\cup E_c)$ be the semibipartite form of G with respect to the independent cliques set. Then, the implicit version of the procedure *NESTEDICS* takes the following form.

```
procedure NESTEDICS_I:
repeat
    begin
        ICS(V);
        for each vertex s in S do
            DEG[s] ← |Reach_G'(s,C)|;
        V ← S
    end
until G(V) is a clique
```



(a) Semibipartite form of G with respect to $I_c^{(1)}$.



(b) Semibipartite form of $G_{|\alpha|}$ with respect to $I_c^{(2)}$.

Figure 8. Semibipartite forms of graph G and elimination graph $G_{|\alpha|}$.

CONCLUSION

Let A be any symmetric matrix and let G be the undirected graph of A. In this work, we develop a new separator theory for partitioning the vertex set in G into two disjoint sets such that if the matrix A contains decoupled blocks that can be factored in parallel and without giving rise to any fill element in A, then the vertices representing the rows of these decoupled blocks are contained in only one of the two disjoint vertex sets. If there are no such blocks in A, then we derive an ordering scheme that compromises between sparsity and parallel computation issues. Our partitioning algorithm has running-time proportional to the number of vertices and edges in G. Using this partitioning, we derive a block algorithm for the solution of sparse symmetric linear systems of equations. The algorithm takes full advantage of the parallel and vector capabilities of high-performance computers, and makes extensive use of standard routines for dense problems to perform the numerical computations. Finally, we establish connections between our method and some well-known block techniques such as bordered block diagonal decomposition, substructuring, and incomplete nested dissection.

REFERENCES

Aho, A. V., J. E. Hopcroft, and J. D. Ullman. 1976. *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA.

Anderson E., Z. Bai, C. Bishop, J. Demmel, J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, A. McKinney, and D. Sorenson. 1990. Lapack Working Note 20. LAPACK: A Portable Linear Algebra Library for High-Performance Computers, University of Tennessee, CS-90-105.

Branin, F. H. 1975. "A Sparse Matrix Modification of Kron's Method of Piecewise Analysis," *Proceedings of the 1975 IEEE Int. Symp. Circuits and Syst.*, pp. 383–386.

Chua, L. O. and L. K. Chen. 1976. "Diakoptic and Generalized Hybrid Analysis," *IEEE Trans. Circuits and Syst.*, CAS-23, pp. 694–705.

Dirac, G. A. 1961. "On Rigid Circuit Graphs," *Abhandlungen Aus Dem* Mathematischen Seminar der Universitat Hamburg, 25, pp. 70–76.

Erisman, A. M. 1973. "Decomposition Methods Using Sparse Matrix Techniques with Application to Certain Electrical Network Problems," in *Decomposition of Large-Scale Problems*, North-Holland Publishing Company, Amsterdam, pp. 69–80.

George, A. and J. W-H Liu. 1986. *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ.

George, A., W. Poole and R. Voight. 1978. "A Variant of Nested Dissection for Solving *n* by *n* Grid Problems," *SIAM J. Numer. Anal.*, 15, pp. 662-673.

Kevorkian, A. K. 1980. "General Topological Results on the Construction of a Minimum Essential Set of a Directed Graph," *IEEE Trans. Circuits and Syst.*, CAS-27, pp. 293–304.

Kron, G. 1958. *Diakoptics*, MacDonald, London.

Lekkerkerker C. G., and J. CH. Boland. 1962. "Representation of a Finite Graph by a Set of Intervals on the Real Line," *Polska Akademia Nauk Fundamenta Mathematicae*, LI, pp. 45–64.

Noor, A., H. Kamel and R. Fulton. 1978. "Substructuring Techniques—Status and Projections," *Computers and Structures*, 8, pp. 621-632.

Ortega, J. M. and R. G. Voight. 1985. "Solution of Partial Differential Equations on Vector and Parallel Computers," *SIAM*, Philadelphia.

Parter, S. V. 1961. "The Use of Linear Graphs in Gaussian Elimination," *SIAM* Rev. 3, pp. 119–130.

Rose, D. J. and R. E. Tarjan. 1975. "Algorithmic Aspects of Vertex Elimination," *Proc. 7th Annual Symposium on the Theory of Computing*, pp. 245–254.

Rose, D. J., R. E. Tarjan, and G. S. Lueker. 1976. "Algorithmic Aspects of Vertex Elimination on Graphs," *SIAM J. Comput.*, 5, pp. 266–283.

Tinney, W. F. and J. W. Walker. 1967. "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorizations," *IEEE Proc.*, 55, pp. 1801–1809.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1991	Final: October 1990—December 1990

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
PARALLEL BLOCK METHODS FOR SPARSE SYMMETRIC LINEAR SYSTEMS OF EQUATIONS	PE: 0602234N PROJ: RS34C77 SUBPROJ: 40-ECB2-01 ACC: DN300 086
6. AUTHOR(S) A. K. Kevorkian	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Naval Ocean Systems Center San Diego, CA 92152-5000	NOSC TR 1399

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
Naval Ocean Systems Center San Diego, CA 92152-5000	

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited.	

13. ABSTRACT (*Maximum 200 words*)

Let A be any symmetric matrix and let G be the undirected graph of A. In this work, we develop a new separator theory for partitioning the vertex set in G into two disjoint sets such that if the matrix A contains decoupled blocks that can be factored in parallel and without giving rise to any fill element in A, then the vertices representing the rows of these decoupled blocks are contained in only one of the two disjoint vertex sets. If there are no such blocks in A, then we derive an ordering scheme that compromises between sparsity and parallel computation issues. Our partitioning algorithm has running-time proportional to the number of vertices and edges in G. Using this partitioning, we derive a block algorithm for the solution of sparse symmetric linear systems of equations. The algorithm takes full advantage of the parallel and vector capabilities of high-performance computers, and makes extensive use of standard routines for dense problems to perform the numerical computations. Finally, we establish connections between our method and some well-known block techniques such as bordered block diagonal decomposition, substructuring, and incomplete nested dissection.

14. SUBJECT TERMS		15. NUMBER OF PAGES
block algorithms cliques elimination graphs	numerical linear algebra parallel computation	43
		16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAME AS REPORT

21a. NAME OF RESPONSIBLE INDIVIDUAL	21b. TELEPHONE *(include Area Code)*	21c. OFFICE SYMBOL
A. K. Kevorkian	(619) 553-2058	Code 7304

INITIAL DISTRIBUTION

Code 0012	Patent Counsel	(1)
Code 0142	K. Campbell	(1)
Code 0144	R. November	(1)
Code 411	R. A. Wasilausky	(1)
Code 423	R. Freund	(1)
Code 454	J. Dickinson	(1)
Code 541	L. Wolfgang	(1)
Code 541	B. Williams	(1)
Code 541	F. Ryan	(1)
Code 542	J. Ferguson	(1)
Code 553	B. Offord	(1)
Code 56	H. Rast	(1)
Code 605	H. Whitehouse	(1)
Code 634	L. Parnell	(1)
Code 634	T. Mautner	(1)
Code 66	P. Reeves	(1)
Code 662	D. Hoffman	(1)
Code 663	G. Meckstroth	(1)
Code 701	H. Schenck	(1)
Code 702	B. Hearn	(1)
Code 712	G. Benthien	(1)
Code 73	J. Roese	(1)
Code 7304	G. Mohnkern	(1)
Code 7304	A. Kevorkian	(15)
Code 733	D. Barbour	(1)
Code 7601	K. Bromley	(1)
Code 7601	J. Zeidler	(1)
Code 761	G. Byram	(1)
Code 761	S. I. Chou	(1)
Code 761	J. Symanski	(1)
Code 761	J. Loughlin	(1)
Code 805	J. Rockway	(1)
Code 825	J. Logan	(1)
Code 844	L. Russell	(1)
Code 9101	J. Baird	(1)
Code 913	T. Wernet	(1)
Code 931	P. Jung	(1)
Code 952B	J. Puleo	(1)
Code 961	Archive/Stock	(6)
Code 964B	Library	(3)

Defense Technical Information Center
Alexandria, VA 22304-6145 (4)

NOSC Liaison Office
Washingotn, DC 20363-5100

Defense Advanced Research Projects
Agency
Arlington, VA 22209

Center for Naval Analyses
Alexandria, VA 22302-0268

Navy Acquisition, Research & Development
Information Center (NARDIC)
Alexandria, VA 22333

NASA, Langley Research Center
Hampton, VA 23665

INITIAL DISTRIBUTION (Cont'd)

NASA, Ames Research Center
Moffett Field, CA 94035

Oak Ridge National Laboratory
Oak Ridge, TN 37831-8083

Lawrence Livermore National Laboratory
Livermore, CA 94550

Argonne National Laboratory
Argonne, IL 60439-4843

Duke University
Durham, NC 27706

Rice University
Houston, TX 77251 (2)

University of California, Los Angeles
Los Angeles, CA 90024

University of Illinois
Urbana, IL 61801

Wake Forest University
Winston-Salem, NC 27109

Boeing Computer Services
Seattle, WA 98124-0346

Scientific Computing Associates, Inc.
New Haven, CT 06510

Xerox Palo Alto Research Center
Palo Alto, CA 94304

Sandia National Laboratories
Albuquerque, NM 87185

Army Research Office
Research Triangle Park, NC 27709

IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598

Cornell University
Ithaca, NY 14853 (2)

Pennsylvania State University
University Park, PA 16801

Stanford University
Stanford, CA 94305

University of California, San Diego
La Jolla, CA 82093

University of Wisconsin
Madison, WI 53706

Yale University
New Haven, CT 06529

San Diego Supercomputer Center
San Diego, CA 92138

The Mathworks
Menlo Park, CA 94025