Predicting Fill for Sparse Orthogonal Factorization

Thomas F. Coleman*
Anders Edenbrandt†
John R. Gilbert+

83-578
October 1983

Department of Computer Science
Cornell University
Ithaca, New York 14853

*The work of this author was supported in part by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under Contract DE-AC02-83ER10369.

†The work of this author was supported in part by an IBM fellowship.

+The work of this author was supported in part by National Science Foundation grant MCS-82-02948.

Predicting Fill
for Sparse Orthogonal Factorization

Thomas F. Coleman[*]
Anders Edenbrandt[†]
John R. Gilbert[‡]

Computer Science Department
Cornell University
Ithaca, New York

October 1983

Abstract

In solving large sparse linear least squares problems $A\mathbf{x}\simeq\mathbf{b}$, several different numeric methods involve computing the same upper triangular factor R of A. It is of interest to be able to compute the nonzero structure of R, given only the structure of A. The solution to this problem comes from the theory of matchings in bipartite graphs. The structure of A is modeled with a bipartite graph and it is shown how the rows and columns of A can be rearranged into a structure from which the structure of its upper triangular factor can be correctly computed. Also, a new method for solving sparse least squares problems, called block back-substitution, is presented. This method assures that no unnecessary space is allocated for fill, and that no space is needed for intermediate fill.

[*] The work of this author was supported in part by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under Contract DE-AC02-83ER13069.

[†] The work of this author was supported in part by an IBM fellowship.

[‡] The work of this author was supported in part by National Science Foundation grant MCS-82-02948.

1. Introduction

Consider solving a system of linear equations

$$A\mathbf{x}\simeq\mathbf{b}$$

where A is an $m\times n$, $m\geq n$, large, sparse matrix. In the following we will always assume that A has full column rank. When A is rectangular the system is overdetermined, and we seek the least squares solution—i.e., the solution that minimizes

$$||A\mathbf{x}-\mathbf{b}||_2 .$$

There are several different methods to find this solution [18]. We will take a brief look at two of them, *orthogonalization* and the *normal equations* method.

In orthogonalization we factor A into

$$A=Q\begin{pmatrix}R\\O\end{pmatrix}$$

where Q is an $m\times m$ orthogonal matrix, and R is an $n\times n$ upper triangular matrix, and then solve the upper triangular system

$$\begin{pmatrix}R\\O\end{pmatrix}\mathbf{x}=Q^T\mathbf{b} .$$

Numerically, we can compute Q and R using Householder reflections [14] or Givens rotations.

With the normal equations approach, we multiply by A^T and solve the square system

$$A^TA\mathbf{x}=A^T\mathbf{b} .$$

If A is of full rank, then A^TA is symmetric and positive definite, and we can factor A^TA into its Cholesky factors LL^T.

From the following equations, and the uniqueness of L,

$$LL^T = A^T A = (QR)^T (QR) = R^T R \ ,$$

we see that L^T is equal to R except for possible sign differences in some rows. Hence, in both of the methods described above we seek to compute the same upper triangular matrix R. Since we are dealing with large, sparse systems, it is desirable to be able to determine the nonzero structure of R, working only from the structure of A. In the following we shall describe two different methods that have been suggested, and we shall show that both can overestimate the number of nonzeros in R.

The first method we look at is based on the numeric method of factoring A by Givens rotations. We call it the Local Givens Rule.

One Givens rotation is usually thought to result in fill (i.e., the creating of new nonzeros) in the following way (see Figure 1.1):

$$\begin{array}{lccccccccccccc} i: & 0\ldots0 & * & 0 & * & 0 & 0 & * & 0 & * & 0 & 0 & 0 \\ j: & 0\ldots0 & * & 0 & 0 & * & 0 & * & 0 & 0 & 0 & * & 0 \\ & & c & & & & & & & & & & \end{array}$$

Figure 1.1.

Here row i is used as a pivot row to zero the entry in column c of row j. The Local Givens Rule for determining the fill says that entry (j,c) becomes zero, entry (i,c) remains nonzero, and in the other columns we take the union of the nonzeros for both rows. In our example, the result would be as shown in Figure 1.2.

$$\begin{array}{llccccccccccc} i: & 0 \ldots 0 & * & 0 & * & * & 0 & * & 0 & * & 0 & * & 0 \\ j: & 0 \ldots 0 & 0 & 0 & * & * & 0 & * & 0 & * & 0 & * & 0 \\ & & c & & & & & & & & & & \end{array}$$

Figure 1.2.

However, as was pointed out in [10], the Local Givens Rule as a way of computing the fill is not fully correct and might predict too much fill. We illustrate this in the example in Figure 1.3.

Figure 1.3.

In this example the triangularization is in reality completed after step 3, because both nonzeros in the last row will be zeroed simultaneously in step 3. The Local Givens Rule does not take this into account and we might predict a fill of $\Omega(n^2)$, although the actual fill is only $O(1)$, and the matrix should remain sparse.

The second method we will look at is the George-Heath algorithm [11]. They use the normal equations to predict the structure of R, and then do the numeric computation by orthogonalization.

1) Starting with the structure of A, compute the structure of A^TA.

2) From the structure of A^TA, compute the structure of L^T—using a symbolic Gaussian elimination algorithm.

George and Heath proved that the structure that results from the algorithm above will have room for all the nonzeros of R.

Lemma 1.1: [11] If we predict position $R_{ij}\neq 0$ when we do symbolic Givens factoring using the Local Givens Rule, then we also predict $R_{ij}\neq 0$ when we compute the structure of R doing symbolic Gaussian Elimination on the structure of A^TA. □

However, this method also may be too generous in allocating space for nonzeros in R, as is shown in the example in Figure 1.4.

Figure 1.4.

Here A is already in upper triangular form, so $R=A$. But, since A has one full row, A^TA is full and R will be predicted to be full.

We have thus seen that both of the previously suggested methods for computing the structure of R may predict too much fill. It should be emphasized that for both of the methods described above the occurrence of so called "bogus fill" is purely a consequence of the nonzero structure of the matrix, and does not depend on the actual numeric values of the nonzero entries.

Rather than coming up with a new rule for computing fill, we have asked the following questions: first, can we exhibit a class of matrix structures for which the George-Heath algorithm gives the correct result? Second, can we reorder the columns and rows of A so that the Local Givens Rule is guaranteed to give the correct result when applied to this rearranged matrix? We answer these questions in the affirmative.

Since the nonzero structure of a matrix is a combinatoric rather than a numeric property, it is convenient to represent this structure by a graph—in our case a bipartite graph—and to model the operations performed on the matrix with operations performed on the vertices and edges of the graph. In Section 2 of this paper we give some basic graph-theoretic definitions, and introduce a structural concept called "the strong Hall property". In Section 3 we analyze the computation of the structure of R using the George-Heath algorithm, and show that this gives the correct result if A has the strong Hall property. This fact can be exploited to provide an algorithm for computing the structure of R using the Local Givens Rule for an arbitrary matrix A, as is shown in Section 4. We also give an algorithm for solving the original system of equations without using the whole structure of R. Finally, Section 5 contains a summary of our results.

2. Definitions and notation

2.1. Basic graph-theoretic notation

An *undirected graph*, $G=(V,E)$, consists of a set V of *vertices* and a set E of *edges*. An *edge* (v,w) is an unordered pair of distinct vertices. If (v,w) is an edge the vertices v and w are *adjacent*. The edge (v,w) is *incident* on vertices v and w, which are its *endpoints*. The *degree* of a vertex, written deg(v), is the number of edges incident on v. Given a vertex v, the set $\Gamma(v)=\{x\colon (v,x)\in E\}$ is called the *adjacency set* of v. Similarly, for a set of vertices, $X\subseteq V$, we say that $\Gamma(X)=\{y\colon (x,y)\in E$ for some $x\in X\}$ is the *adjacency set* of X.

A *matching* in a graph G is a set of edges, $F\subseteq E$, such that no two edges in F have an endpoint in common. A vertex that is the endpoint of some edge in F is *covered*; otherwise it is *exposed*. If every vertex is covered by F then F is a *perfect matching*.

A *path* between v and w in G is a sequence of distinct edges $(v_0,v_1),(v_1,v_2),\ldots,(v_{k-1},v_k)$, where $v=v_0$ and $w=v_k$ and all the vertices are distinct except possibly $v_0=v_k$. If F is a matching in G, an *alternating path* is a path in which every other edge is in F.

A *subgraph*, $G'=(V',E')$, of $G=(V,E)$ is a graph where $V'\subseteq V$ and $E'\subseteq E$. If $V'=V$ then G' is a *spanning subgraph*. The set of edges with both endpoints in V' is written as $E(V')$. If $E'=E(V')$ then G' is the subgraph of G *induced* by the vertex set V'.

For an $n \times n$ symmetric matrix A that has only nonzeros on its diagonal, we can use an undirected graph, $G(A)$, to represent the zero/nonzero structure of A. The graph $G(A)$ has n vertices and (v_i,v_j) is an edge if and only if $A_{ij} \neq 0$. The changes in the structure of A during Gaussian elimination can be modeled as a process working on the graph: when column i is zeroed out in the matrix, we mark v_i "old" and add enough new edges to turn all of v_i's unmarked neighbors into a complete graph. The final result of this process is called the ***filled graph***, G^*, and we refer to this process as ***symbolic Gaussian elimination.*** The following characterization of fill in terms of an undirected graph is very useful.

Lemma 2.1: [23] The edge (v,w) is an edge in the filled graph G^* if and only if there is a path in G from v to w going only through vertices marked earlier than both v and w. □

2.2. Bipartite graphs and the Hall property

An undirected graph whose vertex set can be divided into two disjoint sets V_1 and V_2, such that every edge in E has one endpoint in V_1 and the other in V_2, is called a ***bipartite graph.*** This graph is sometimes written $G=(V_1,V_2,E)$ to stress the partition of V.

Let $G=(V,W,E)$ be a bipartite graph with $|V| \geq |W|$. If, for every set of vertices, $X \subseteq W$, X is adjacent to at least $|X|$ vertices in V, then we say that the bipartite graph G has the ***Hall property.*** If every X is adjacent to more than $|X|$ vertices, we say that G has the ***strong Hall property;*** or, for more precise definitions:

Definition 2.1: A bipartite graph $G=(V,W,E)$ with $|V|\geq|W|$ has the *Hall property* if

$$\forall X\subseteq W,\ X\neq\emptyset:\ |\Gamma(X)|\geq|X|.$$

Definition 2.2: A bipartite graph $G=(V,W,E)$ with $|V|\geq|W|$ has the *strong Hall property* if

$$\forall X\subset W,\ X\neq\emptyset:\ |\Gamma(X)|>|X|.$$

We list some basic and useful facts about matchings in bipartite graphs.

Lemma 2.2: [16] Let $G=(V,W,E)$ be a bipartite graph. There is a matching in G of cardinality $|W|$ if and only if G has the Hall property. □

Corollary 2.3: Let $G=(V,W,E)$ be a bipartite graph with $|V|=|W|$. Then G has a perfect matching if and only if G has the Hall property. □

A *directed graph* $D=(V,A)$ is a graph whose edges are ordered pairs of distinct vertices, $<v_i,v_j>$, called *arcs*. A *directed path* from v to w in D is a sequence of distinct arcs $<v_0,v_1>,<v_1,v_2>,\ldots,<v_{k-1},v_k>$, where $v=v_0$ and $w=v_k$ and all the vertices are distinct except possibly $v_0=v_k$. If for every two vertices v and w in D there is a directed path from v to w, then D is *strongly connected*.

With any directed graph D of n vertices we can associate a bipartite graph, $G(D)$, as follows: $G(D)=(X,Y,E)$, with $|X|=|Y|=n$, and (x_i,y_j) is an edge in E if $<v_i, v_j>$ is an arc in A. In addition, all the edges (x_i, y_i) for $1 \leq i \leq n$ are in E.

The following observation, due to Johnson, Dulmage, and Mendelsohn [21], is also proved in [20].

Lemma 2.4: [21] D is strongly connected if and only if $G(D)$ has the strong Hall property. □

Note that this correspondence between the strong Hall property and strong connectedness is well defined only when the bipartite graph is "square", that is, has the same number of vertices in each part. Indeed, it is possible for a rectangular bipartite graph that has the strong Hall property not to be connected, as in Figure 2.1. However, for every bipartite graph there exists a canonical decomposition of G into subgraphs, each of which has the strong Hall property. This topic will be pursued further in Section 4.

Figure 2.1.

2.3. Representing the structure of a matrix with a bipartite graph

Let A be an $m \times n$ real matrix, with $m \geq n$. We represent the zero/nonzero structure of A with a graph, $G(A)$. $G=(V,W,E)$ is a bipartite graph with $V=\{v_1,\ldots,v_m\}$, $W=\{w_1,\ldots,w_n\}$, and $(v_i,w_j) \in E$ if and only if $A_{ij} \neq 0$. We refer to the vertices of W as ***column-vertices***, and to the vertices of V as ***row-vertices***.

Given any two sets $W_1, W_2 \subseteq W$ of column-vertices, we can define a new bipartite graph $H=(X,Y,S)$ where $|X|=|W_1|$, $|Y|=|W_2|$, and $(x_i,y_j) \in S$ if and only if $w_i \in W_1$ and $w_j \in W_2$ and $w_j \in \Gamma(\Gamma(w_i))$. We write this as $H=\Phi(G,W_1,W_2)$. See Figure 2.2 for an example.

If U_1 is the matrix containing only the columns of A represented by W_1, and U_2 is the matrix containing the columns of W_2, then $H=G((U_1)^T U_2)$, provided that no two columns that have some nonzeros in the same rows are orthogonal. This is true, for example, if all the entries are nonnegative. In particular, under this assumption $\Phi(G(A),W,W)=G(A^T A)$.

Figure 2.2.

By $N(G)$ we denote the set of $m \times n$ real matrices A such that $A_{ij}=0$ if (v_i, w_j) is not an edge in G.

For a matrix A, we define the ***structural rank*** of A to be the size of a maximum matching in $G(A)$. The ***numeric rank*** of A is the number of linearly independent columns of A. The structural rank of A is at least as large as its numeric rank.

To make it easier to remember all this notation we summarize it in a table below.

Symbol(s)	Denotes
$G=(V,E)$	Undirected graph
$G=(V,W,E)$	Bipartite graph
$D=(V,A)$	Directed graph
$\Gamma(X)$	Adjacency set of X
$\Phi(G,W_1,W_2)$	Product graph
$<G,r,p,q>$	The special product graph $\Phi(G,W_1,W_2)$ where $W_1=\{w_1,...,w_r\}\cup\{w_p\}$ and $W_2=\{w_1,...,w_r\}\cup\{w_q\}$.
A or M	Matrix
$M=<A,r,p,q>$	Special submatrix of A^TA
$N(G)$	Matrices whose structure fits into G
$G(A)$	Bipartite graph that represents the structure of A

3. Matrices for which the George-Heath algorithm works

A matrix A has an orthogonal factorization $A=QR$, and we are interested in determining the structure of the upper triangular factor R. This R is numerically the same as the triangular factor in the Cholesky factorization $LL^T=A^TA$. When A has full column rank, A^TA is positive definite, and the Cholesky factorization can be computed using Gaussian elimination with pivots from the main diagonal. There are well-known and efficient algorithms for determining the structure of the matrix that results from Gaussian elimination [12]. However, any such algorithm produces the correct structure only under the assumption that no cancellation occurs during the Gaussian elimination.

It is worthwhile at this point to say what we mean by cancellation. Gaussian elimination (for our purposes) is an algorithm that zeros the subdiagonal elements of one column at a time, meanwhile replacing some of the elements in other columns to its right. Cancellation occurs when a nonzero is replaced by a zero outside the column being eliminated.

There are two kinds of cancellation. ***Lucky cancellation*** is cancellation that occurs only for certain values of the nonzeros in the given matrix. We shall disregard lucky cancellation because it cannot be predicted from the structure alone, and because lucky zeros are not likely to be computed as exact floating-point zeros. ***Essential cancellation*** occurs no matter what the nonzero values are.

It can be shown [8] that if we are given the structure of a positive definite matrix M, then there is no essential cancellation when computing its Cholesky factorization $R^TR=M$. Here, however, we begin not with the structure of A^TA, the matrix whose Cholesky factor we want, but with the structure of A. The entries of A^TA are not independent: the structure of A may be such that cancellation will occur when we do Gaussian elimination on A^TA, regardless of the values of the nonzeros in A. This essential cancellation is the reason why the George-Heath algorithm may predict too much fill. In the following we will show that, provided A has the strong Hall property, no essential cancellation will occur when we perform Gaussian elimination on A^TA. From this we can conclude that the George-Heath algorithm gives the correct result for matrices that have the strong Hall property.

Our way to the main theorem of this section is via three important lemmas. The first of these lemmas says that we need only consider certain special submatrices of A^TA. The second lemma talks about bipartite graphs, and shows how we can use the strong Hall property to make sure that each special submatrix has the property we desire. The third lemma tells us that it is enough to look at just one special submatrix at a time, rather than all of them simultaneously.

3.1. A special sort of submatrix

The following lemma shows that if cancellation occurs at position (p,q) in A^TA during Gaussian elimination, then this position is in the lower right-hand corner of a singular submatrix of A^TA of a special form described below.

Figure 3.1.

Consider the square matrix M consisting of the principal $r \times r$ submatrix of $A^T A$, plus the first r entries of row p and column q, plus entry (p,q), where $1 \leq r < p \leq q \leq n$. See Figure 3.1. As a shorthand notation we will write $M = <A,r,p,q>$.

Lemma 3.1: If cancellation occurs in $A^T A$ at entry (p,q) during the r-th stage of the Gaussian elimination, when row r is used to zero out entry (p,r), then $M = <A,r,p,q>$ has full structural rank and is singular.

Proof: To show that M has full structural rank, let $C = G(M)$ be the bipartite graph that represents the structure of M, and let $H = (X,Y,S) = G(A^T A)$ be the bipartite graph that represents the structure of $A^T A$. Clearly, C is a subgraph of H.

Since $A^T A$ is a symmetric matrix with a nonzero diagonal, $F = \{(x_1,y_1),...,(x_n,y_n)\}$ is a perfect matching in H. By Lemma 2.1 it then follows that position (p,q) in $A^T A$ fills in if and only if there is an alternating path in H from vertex x_p to vertex y_q going only through lower-numbered vertices. Hence, if we have cancellation at position (p,q), such a path exists, and by interchanging the matching and non-matching edges of this path we can construct a perfect matching in C.

To see that M is singular, note that performing Gaussian elimination on a square matrix does not change the value of its determinant; and since the resulting matrix is upper triangular and has one diagonal entry equal to zero, it must be singular. □

3.2. A structural lemma

In this section we show that if A's bipartite graph has the strong Hall property, then we can make any one of the special submatrices of $A^T A$ nonsingular by choosing appropriate values for the nonzeros of A.

As before, $G=(V,W,E)$ is a bipartite graph with the strong Hall property. Let $W_1=\{w_1,...,w_r\}\cup\{w_p\}$ and $W_2=\{w_1,...,w_r\}\cup\{w_q\}$. Let $C=\Phi(G,W_1,W_2)$. We also use the shorthand notation $C=<G,r,p,q>$ for this graph.

If $G=G(A)$, and $M=<A,r,p,q>$, then $M\in N(C)$; and if, e.g., all entries in A are nonnegative, then $C=G(M)$.

Now, we are given the structure of a matrix, i.e., a bipartite graph G. We want to show that provided that G has the strong Hall property, we can find a matrix A that fits into the given structure, and is such that for each subgraph $C=<G,r,p,q>$ that has a perfect matching, the corresponding submatrix $M=<A,r,p,q>$ is nonsingular. If we can show this, then by Lemma 3.1 we can say that there will be no essential cancellation during Gaussian elimination in $A^T A$.

We will take on a somewhat easier task, though, namely showing that for any single subgraph $C=<G,r,p,q>$ that has a perfect matching, we can find a matrix that fits into G and has $M=<A,r,p,q>$ nonsingular.

Each nonzero term in the determinant of a square matrix corresponds to a different perfect matching in its bipartite graph. Hence, if C has exactly one perfect matching, then the determinant must be nonzero—i.e., M is nonsingular.

On the other hand, if C has no perfect matching at all, this means that all terms in the determinant of M are zero, and the matrix is singular.

Lemma 3.2: Let $G=(V,W,E)$ be a bipartite graph, and let W_1 and W_2 be subsets of W such that $|W_1|=|W_2|=k$, and $|W_1 \cap W_2| \geq k-1$. If G has the strong Hall property, and if $C=\Phi(G,W_1,W_2)$ has a perfect matching, then there is a spanning subgraph $\hat{G}$ of G such that $\hat{C}=\Phi(\hat{G},W_1,W_2)$ has exactly one perfect matching.

Proof: Let $C=(S,T,U)$, with $S=\{s_1,...,s_k\}$ and $T=\{t_1,...,t_k\}$. It is convenient to distinguish between two cases:

Case 1, $|W_1 \cap W_2|=k$, i.e., $W_1=W_2$.

Let $G_1=(V,W_1,E)$. Since G_1 has the Hall property, we know there is a matching F_1 in G_1 that covers W_1. Let $\hat{G}=(V,W,F_1)$. It is easy to see that any vertex s_i in $\hat{C}$ is adjacent only to t_i. Therefore, $\hat{C}$ has precisely one perfect matching, namely $\{(s_1,t_1),...,(s_k,t_k)\}$.

Case 2, $|W_1 \cap W_2|=k-1$, i.e. there is a vertex $w_p \in W_1$ such that $w_p \notin W_2$, and a vertex $w_q \in W_2$ such that $w_q \notin W_1$. Without loss of generality, we may assume that $W_1=\{w_1,...,w_{k-1}\} \cup \{w_p\}$, $W_2=\{w_1,...,w_{k-1}\} \cup \{w_q\}$. For the proof of this second case, we need two more lemmas.

The following lemma is due to Ford and Fulkerson.

Lemma 3.3: [9, Corollary 10.9] Let $G_1=(V,W_1,E_1)$ and $G_2=(V,W_2,E_2)$ be two bipartite graphs with one vertex set in common, and $|W_1|=|W_2|=k$.

There is a set $P \subseteq V$ of k vertices, with a matching F_1 in G_1 covering P, and a matching F_2 in G_2 also covering P, if and only if:

$$\forall X \subseteq W_1 \, \forall Y \subseteq W_2\colon \ |X| + |Y| \leq k + |\Gamma(X) \cap \Gamma(Y)| \ . \qquad (1)$$

□

Lemma 3.4: Let $G_1=(V,W_1,E(V \cup W_1))$ and $G_2=(V,W_2,E(V \cup W_2))$. If $C=\Phi(G,W_1,W_2)$ has a perfect matching, then there is a matching F_1 in G_1 and a matching F_2 in G_2, both of cardinality k and both covering the same set $V' \subseteq V$, where $|V'|=k$.

Proof: By Lemma 3.3, it is enough to show that condition (1) holds for G_1 and G_2.

Let $X \subseteq W_1$ and $Y \subseteq W_2$. If $|X|+|Y| \leq k$ then the inequality in (1) is immediate. Assume that $|X|+|Y|=k+\delta$, where $\delta \geq 1$. We must prove that $|\Gamma(X) \cap \Gamma(Y)| \geq \delta$. Let $Z=X \cap Y$. Clearly, $\Gamma(Z) \subseteq \Gamma(X) \cap \Gamma(Y)$.

Since X and Y are chosen from sets of size k that have k-1 elements in common, we can conclude that $|Z| \geq \delta$-1. We consider separately the cases $Z=\emptyset$ and $Z \neq \emptyset$.

If $Z=\emptyset$ then $\delta=1$. By assumption C has a perfect matching, which by Corollary 2.3 implies that $|\Gamma_C(X)|=|\Gamma(\Gamma(X)) \cap W_2| \geq |X|$. Since $\Gamma_C(X) \subseteq W_2$ and

$$|\Gamma_C(X)|+|Y| \geq |X|+|Y|=k+1 \ ,$$

$\Gamma_C(X)$ and Y must have at least one vertex in common. That is, there must be at least one vertex $w \in Y$ such that $w \in \Gamma_C(X)$, and thus

$$|\Gamma(X)\cap\Gamma(Y)|\geq 1=\delta\,.$$

Next, assume that $Z\neq\emptyset$. By the strong Hall property $|\Gamma(Z)|>|Z|\geq\delta-1$. Then

$$|\Gamma(X)\cap\Gamma(Y)|\geq|\Gamma(Z)|\geq\delta\,.$$

We have thus shown that condition (1) is satisfied, and Lemma 3.4 now follows from Lemma 3.3. □

We continue case 2 of the proof of Lemma 3.2.

Among the matchings F_1 in G_1 and F_2 in G_2 that satisfy the conclusion of Lemma 3.4, choose that pair F_1, F_2 that maximizes $|F_1\cap F_2|$. Let $\hat{G}=(V,W,F_1\cup F_2)$ be the graph with the vertices of the original graph G and the edges of the two matchings F_1 and F_2. We will show that $\hat{C}=\Phi(\hat{G},W_1,W_2)$ has exactly one perfect matching. In fact, we will see that $\hat{C}$ consists of a path of odd length plus some isolated edges.

First note that every vertex in $\hat{G}$ has degree at most 2, and that $\deg(w_p)=\deg(w_q)=1$, since w_p and w_q are only covered by one matching each.

Let s_i be any vertex in $\hat{C}$. If $\deg(w_i)=1$ and $i\neq p$ and $i\neq q$, then w_i was matched to the same vertex v_j in both F_1 and F_2. This implies that $\deg(v_j)=1$ in $\hat{G}$ and that (s_i,t_i) is the only edge incident on s_i and also the only edge incident on t_i, in $\hat{C}$. In other words, s_i can be matched with exactly one other vertex, t_i.

If $\deg(w_i)=1$ and $i=p$, so that s_i corresponds to w_p, then some edge $e_1=(w_p,v_\alpha)$ is in F_1. Some other edge $e_2=(v_\alpha,w_\beta)$ must be in F_2. Thus, s_i in $\hat{C}$

is adjacent to exactly one vertex, namely t_β. It follows that (s_i, t_β) must be a matching edge in $\hat{C}$.

If we delete e_1 and e_2 from $\hat{G}$, then we can repeat the argument above, beginning at w_β, to find another edge that must be in the matching on $\hat{C}$. This procedure will continue until deg(t_β)=1. Since the only other vertex of degree 1 that we have not dealt with is w_q, this will happen precisely when we reach w_q.

Finally, if deg(w_i)=2, and if w_i is not on the path between w_p and w_q, then w_i must lie on a cycle in $\hat{G}$. The cycle edges alternate between F_1 and F_2. We can increase $|F_1 \cap F_2|$ by putting every other cycle edge into both F_1 and F_2. This contradicts the assumption that the matchings were chosen to maximize $|F_1 \cap F_2|$.

This completes the proof that there is only one perfect matching in $\hat{C}$. □

3.3. An exercise in field theory

Lemma 3.2 in the previous section implies that if the graph $<G,r,p,q>$ has a perfect matching, then the special submatrix $<A,r,p,q>$ of A^TA is made nonsingular by some assignment of values to A—that is, by some $A \in N(G)$. In this section we show that there is a single assignment of values to A that makes all these special submatrices nonsingular at once. For more details on notation and background in field theory, the reader is referred to [3].

Lemma 3.5: Let A be a matrix with t nonzeros, where in place of each nonzero we have put a variable, $x_1,x_2,...,x_t$. Let $M_1,M_2,...,M_k$ be submatrices of A^TA. Suppose that for each i with $1 \leq i \leq k$ there is an assignment of ~~values~~ $a_1^{(i)},a_2^{(i)},...,a_t^{(i)}$ to the nonzero entries of A such that M_i is nonsingular. Then there is a single assignment $a_1,...,a_t$ such that every M_i, $1 \leq i \leq k$, is nonsingular.

Proof: The proof is from the following lemmas.

Lemma 3.6: There is a sequence $a_1,...,a_t$ of real numbers such that if the fields $F_0,F_1,...,F_{t-1}$ are defined by

$$\begin{array}{ll} F_0 = Q & \text{(i.e., the rationals)} \\ F_{k+1} = F_k(a_{k+1}) & \text{(i.e., } F_{k+1} \text{ is the smallest field} \\ & \text{ containing } F_k \text{ and } a_{k+1}) \end{array}$$

then a_{k+1} is not a zero of any nonzero polynomial in $F_k[z]$ (i.e. of any polynomial with coefficients from F_k).

Proof: Every element of F_{k+1} is equal to $p(a_{k+1})/q(a_{k+1})$ for some polynomials p and q in $F_k[z]$. By induction on k, then, each F_k is countable. Thus $F_k[z]$ has only countably many polynomials, each with only finitely many zeros. Thus there are only countably many zeros of polynomials over F_k; let a_{k+1} be one of the uncountably many other real numbers. □

Lemma 3.7: Let $a_1,...,a_t$ be the sequence from Lemma 3.6. Then $\mathbf{a}=(a_1,...,a_t)$ is not a zero of any nonzero polynomial in t variables over Q, the rationals.

Proof: By induction on t. Suppose $\mathbf{a}$ is a zero of $p(z_1,...,z_t)= p(\mathbf{z}) \in Q[z_1,...,z_t]$. Then a_t is a zero of the one-variable polynomial $q(z)=p(a_1,...,a_{t-1},z) \in F_{t-1}[z]$. By Lemma 3.6, q is the zero polynomial, so all the coefficients of q are zero. Each coefficient of q is a polynomial in $t-1$ variables over Q, evaluated at $(a_1,...,a_{t-1})$. Thus by the inductive hypothesis each of these is the zero polynomial. Thus p is the zero polynomial. □

Proof of Lemma 3.5: Let $a_1,...,a_t$ be as in Lemma 3.6. For each i, $\det(M_i)$ is a polynomial with integer coefficients in the variables $x_1,...,x_t$, so $\det(M_i)=p_i(x_1,...,x_t)$. Since $p_i(\mathbf{a}^{(i)})\neq 0$, p is not the zero polynomial. Thus $p_i(\mathbf{a})\neq 0$, so M_i is nonsingular for $\mathbf{x}=\mathbf{a}$. □

3.4. Main theorem

With all the necessary lemmas in hand, we are ready for the main theorem of this section. Our theorem below is stated twice: once informally, and once in more formal terms. The formal version deserves a few words of explanation.

When we say that a computed structure H_R is "correct", we do not necessarily mean that the factor R of the matrix we started with fits exactly into H_R. Recall that we are computing H_R not from A, but from the structure of A. What we require of a "correct" structure is rather

a) that it is sufficiently large; i.e., for every matrix A of the given structure, its factor R fits into H_R

b) that it is not too large; i.e., there is some matrix A of the given structure whose factor R needs all of H_R.

Theorem 3.8: Let A be an $m \times n$ matrix, $m \geq n$, with the strong Hall property, and an orthogonal factorization $A = QR$. The following algorithm will correctly compute the nonzero structure of the upper triangular factor R, disregarding lucky cancellation.

1) Determine the structure of $A^T A$.

2) Apply symbolic Gaussian elimination to the structure of $A^T A$, yielding the structure of R.

We can state the theorem more precisely as follows: Let $G=(V, W, E)$ be a bipartite graph with the strong Hall property, with $|V| = m$ and $|W| = n$. Let $H = \Phi(G, W, W)$ be as defined above. Use symbolic Gaussian elimination on H, to form H_R. The theorem now states that

1) If $A \in N(G)$ has an orthogonal factorization $A = QR$, then $G(R)$ is a subgraph of H_R.

2) There exists a matrix $A \in N(G)$, with an orthogonal factorization $A = QR$, such that $G(R)$ is equal to H_R.

Proof: The first part of the theorem is just Lemma 1.1, and was proved by George and Heath [11].

For the second part of the theorem, recall that the George-Heath algorithm gives an incorrect result when the symbolic Gaussian elimination does not produce the same structure as that from the actual Gaussian elimination. Furthermore, this happens when cancellation occurs during the actual Gaussian elimination. Hence, we want to show that there is a matrix $A \in N(G)$ in which no cancellation occurs during Gaussian elimination in $A^T A$. By Lemma 3.1, it is enough if we can show that there is a matrix $A \in N(G)$ such that for each subgraph $C = <G,r,p,q>$ that has a perfect matching, $M = <A,r,p,q>$ is nonsingular.

Let $C = <G,r,p,q>$, $1 \leq r < p \leq q \leq n$, be a subgraph of H that has a perfect matching. By Lemma 3.2, for every such C there is a subgraph G' of G, such that $C' = <G',r,p,q>$ has precisely one perfect matching.

Form A' by setting $A'_{ij} = 1$ if (w_i, w_j) is an edge in G', and setting $A'_{ij} = 0$ otherwise. Clearly, $A' \in N(G)$, and $M' = <A',r,p,q>$ is nonsingular—since $C = G(M')$, and hence its determinant has exactly one nonzero term.

Thus, for each subgraph $C = <G,r,p,q>$ with a perfect matching, there is a matrix $A' \in N(G)$ such that $M' = <A',r,p,q>$ is nonsingular. Finally, by Lemma 3.5, we conclude that there is one matrix, A, such that for all subgraphs $C = <G,r,p,q>$ with a perfect matching, $M = <A,r,p,q>$ is nonsingular. For this matrix A, no essential cancellation will occur during Gaussian elimination in $A^T A$, and the theorem follows. □

4. Finding the structure of R

Section 3 showed us that, provided A has the strong Hall property, we can determine the structure of R correctly with the George-Heath algorithm. But what if we want to use the Local Givens Rule? Or, most important, what can we do when A does not have the strong Hall property? We answer these questions below. First, we show that the Local Givens Rule also works correctly on matrices with the strong Hall property.

Theorem 4.1: Let A be an $m \times n$ matrix, $m \geq n$, with the strong Hall property. If we compute the nonzero structure of R by doing symbolic Givens rotations using the Local Givens Rule, then the result is correct.

Proof: Let H_R be the structure that results from using the George-Heath algorithm on the structure of A. As was shown in Section 3, H_R is the correct structure. Lemma 1.1 says that the Local Givens Rule gives at most the fill present in H_R. The Local Givens Rule never gives too little fill, so it must give all the fill in H_R. □

In the general case—when the structure of A does not necessarily have the strong Hall property—we can permute A into parts, each of which has the desired property. This canonical reordering scheme was first studied by Dulmage and Mendelsohn [5], [6], [7], [21].

The reordering we want is what is known as "block upper triangular form" [15]. It is a permutation of the columns and rows of A that leaves square submatrices on the diagonal, and a rectangular submatrix in the lower right-hand corner. See Figure 4.1. The area below and to the left of these submatrices has only zeros in it, whereas the area above and to the right may have nonzeros. All the submatrices on the main diagonal have the strong Hall property. The diagonal elements are all nonzero.

As we pointed out in Section 2, the strong Hall property is closely related to the strong connectedness of a directed graph. In particular, if A is a square matrix we can do our reordering by finding the strongly connected components in the directed graph that corresponds to A [15], [24]:

1) Find a perfect matching in A and, with this matching fixed, construct the directed graph $D(A)$.

2) Find the strongly connected components of D.

Regardless of which perfect matching we choose in Step 1, we will always get the same number of components in Step 2.

Figure 4.1.

If A is not square, we can proceed as follows. First find a matching in A that covers all the columns. Then the last, rectangular, component of A consists of every column and row that can be reached from an uncovered row via an alternating path. The remaining rows and columns are a square submatrix that can be reordered into strongly connected components as above.

The time to find a maximum matching is $O(t\sqrt{m+n})$ [19], [22]. The rectangular component can be found by depth-first search in $O(t)$ time. The strong components can also be found in $O(t)$ time [24], [1]. (In fact, the same depth-first search can be used to find the rectangular component and the strong components.) Thus, the worst-case time for the reordering is bounded by $O(t\sqrt{m+n})$. Duff's experiments with a heuristic maximum-matching algorithm [4] indicate that in practice a maximum matching can often be found in $O(t)$ time; in such cases, the entire reordering takes $O(t)$ time.

Theorem 4.2: Let A be an $m\times n$ rectangular matrix with a factorization $A=QR$. If we use the algorithm above to permute A into A', and use the Local Givens Rule on A' to find the structure of R, then the result is correct.

Proof: Let M be any of the square submatrices on the diagonal. If there are none, A itself has the strong Hall property, and the theorem follows from Theorem 4.1.

Let $\mathbf{c}_i,\ldots,\mathbf{c}_j$ be the columns that are in M, and let $\mathbf{c}_k$ be any other column in A' to the right of M. If $\mathbf{c}_k$ has only zeros in rows $i,\ldots,j$, then clearly the factoring of M cannot affect the structure of $\mathbf{c}_k$ in these rows.

If $\mathbf{c}_k$ has at least one nonzero in rows $i,\ldots,j$, consider the rectangular submatrix, U, made up from columns $\mathbf{c}_i,\ldots,\mathbf{c}_{j-1},\mathbf{c}_k$. Since column $\mathbf{c}_k$ has a nonzero in row k, U has the strong Hall property. Hence, by Theorem 4.1, the Local Givens Rule works on this matrix, and it follows that the Local Givens Rule works correctly on the whole of A. □

Although the result in the theorem above is interesting in its own right, it turns out that in practice it is probably better to solve the system by "block back-substitution"—and never compute the structure of R. This is done as follows. (Refer to Figure 4.2 for notation.)

Figure 4.2.

Algorithm 4.1:

```
1) Reorder $A$ as shown in Figure 4.2, with square diagonal blocks
   $M_1, M_2, \ldots, M_k$ and a rectangular diagonal block $M_{k+1}$.

2) Solve the rectangular least-squares system
       $M_{k+1} \mathbf{x}_{k+1} \simeq \mathbf{b}_{k+1}$.

3) For $i=k, k-1, \ldots, 1$ do:
       Update the vector $\mathbf{b}$, by setting
       $(\mathbf{b}_1, \ldots, \mathbf{b}_i) \leftarrow (\mathbf{b}_1, \ldots, \mathbf{b}_i) - U_{i+1} \mathbf{x}_{i+1}$.
       Solve the square system $M_i \mathbf{x}_i = \mathbf{b}_i$.
```

Theorem 4.3: Algorithm 4.1 correctly computes the solution to the least-squares system $A\mathbf{x} \simeq \mathbf{b}$.

Proof: Refer to Figure 4.3 for notation. To find the least squares solution we want to minimize $||\mathbf{r}||_2$, where $\mathbf{r}=A\mathbf{x}-\mathbf{b}$. In Algorithm 4.1 we split $\mathbf{r}$ into two parts, $\mathbf{r}=(\mathbf{r}_1,0)+(0,\mathbf{r}_2)$. Here, $\mathbf{r}_1=A_1\mathbf{x}_1+U_2\mathbf{x}_2-\mathbf{b}_1$, and $\mathbf{r}_2=A_2\mathbf{x}_2-\mathbf{b}_2$. Since $(\mathbf{r}_1,0)$ and $(0,\mathbf{r}_2)$ never have nonzeros in the same coordinates, $||\mathbf{r}||_2^2=||\mathbf{r}_1||_2^2+||\mathbf{r}_2||_2^2$.

Figure 4.3.

Algorithm 4.1 finds a solution that minimizes $||\mathbf{r}_2||_2$ and makes $||\mathbf{r}_1||_2$ equal to zero. It follows that this solution must also minimize $||\mathbf{r}||_2$, and hence that Algorithm 4.1 computes the correct solution to the least squares problem. □

This way of solving the system has several advantages. First, in step (2), we know that M_{k+1} has the strong Hall property, so by Theorem 4.1 we will not use any unnecessary space—whether we use the George-Heath algorithm or the Local Givens Rule. In case we use the George-Heath algorithm, note that the strong Hall property is preserved under column and row permutations. This means we can use a heuristic, like minimum degree, on the structure of A^TA to find a column ordering that gives small fill, and still be guaranteed that we are not allocating any extra space for R.

Second, all the systems we solve in step (3) are square, which means that we can use any standard algorithm to solve these, and to find orderings that give little fill. Of course, one way to solve the square system $M_i\mathbf{x}_i=\mathbf{b}_i$ is by a QR factorization of M_i [17]. If we do this, Theorem 3.8 says that the George-Heath algorithm correctly predicts the structure of M_i's triangular factor.

Third, by updating the vector $\mathbf{b}$ every time we have determined new values for some variables in $\mathbf{x}$, we avoid having to store any fill at all in U, the part of A above the diagonal blocks.

5. Summary

We described the nonzero structure of matrices and singled out a class of structures, namely those with the strong Hall property. For matrices with this property we showed that the George-Heath algorithm correctly predicts the structure of the upper triangular factor R. We also showed that this condition is sufficient to say that the Local Givens Rule will correctly predict the structure of R. However, in neither case is it a necessary condition: there are matrices that do not have the strong Hall property, and in which no bogus fill occurs.

We gave a way of reordering a matrix of arbitrary structure into blocks for which the George-Heath algorithm correctly predicts fill. We outlined an algorithm that uses this reordering to solve a least squares problem block by block, storing no unnecessary fill within each block and storing no fill at all between blocks. The reordering also gave a way to permute columns so that the Local Givens Rule would correctly predict the structure of the triangular factor of the whole matrix.

Our analysis suggests that the fill from a Givens rotation cannot be characterized by a local rule—i.e., a rule based on the structure of only a fixed number of rows—as opposed to Gaussian elimination. We have found it necessary to use both a local rule—the Local Givens Rule—and a concept that summarizes the structure of the whole matrix—the strong Hall property. On the other hand, the fill in Gaussian elimination can be given a precise *static* characterization: we can predict which positions will be filled in terms of paths in the initial structure, as in Lemma 2.1. This characterization has been a valuable theoretic tool. For this reason it might be interesting to find a similar static characterization for Givens rotations. George and Ng explore this theme in [13].

6. References

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, Mass., 1974.

2. Bunch, J.R., Rose, D.J., (Eds.). *Sparse Matrix Computations.* Academic Press, New York, 1976.

3. Dean, R.A. *Elements of Abstract Algebra.* John Wiley & Sons, New York, 1966.

4. Duff, I.S. On algorithms for obtaining a maximum transversal. *ACM Trans. Math. Software 7,* (Sept. 1981), 315-330.

5. Dulmage, A.L., Mendelsohn, N.S. Coverings of bipartite graphs. *Can. J. Math. 10,* (1958), 517-534.

6. Dulmage, A.L., Mendelsohn, N.S. A structure theory of bipartite graphs of finite exterior dimension. *Trans. Roy. Soc. Canada Sect III 53,* (1959), 1-13.

7. Dulmage, A.L., Mendelsohn, N.S. Two algorithms for bipartite graphs. *J. Soc. Indust. Appl. Math. 11,* (1963), 183-194.

8. Edenbrandt, A., Gilbert, J.R. Predicting fill in Gaussian elimination. Tech. Report (in preparation), Dept. of Comp. Sci., Cornell University, Ithaca, N.Y., (1983).

9. Ford, L.R., Fulkerson, D.R. *Flows in Networks.* Princeton University Press, Princeton, N.J., 1962.

10. Gentleman, W.M. Row elimination for solving sparse linear systems and least squares problems. In *Proceedings Dundee Conf. on Numerical Analysis*, Lecture Notes in Math. 506, Watson, G.A. (Ed.), 1975.

11. George, A., Heath, M.T. Solution of sparse linear least squares problems using Givens rotations. *Lin. Alg. and Appl. 34*, (1980), 69-83.

12. George, A., Liu, J.W. *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall, Englewood Cliffs, N.J., 1981.

13. George, A., Ng, E. On row and column orderings for sparse least squares problems. Res. Rep. CS-81-09, Dept. of Comp. Sci., Univ. of Waterloo, Waterloo, Ontario, Canada, 1981.

14. Golub, G.H., Businger, P.A. Linear least squares solutions by Householder transformations. *Numer. Math. 7*, H.B. Series Linear Algebra, (1965), 269-276.

15. Gustavson, F. Finding the block lower triangular form of a sparse matrix. In [2], 275-289.

16. Hall, P. On representations of subsets. *J. London Math. Soc. 10*, (1935), 26-30.

17. Heath, M.T. Some extensions of an algorithm for sparse linear least squares problems. *SIAM J. Sci. Stat. Comput. 3*, (1982), 223-237.

18. Heath, M.T. Numerical methods for large sparse linear least squares problems. Rep. no. ORNL/CSD-114, Oak Ridge National Laboratory, (1983).

19. Hopcroft, J.E., Karp, R. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. of Comp. 2*, (1973), 225-231.

20. Howell, T.D. Partitioning using PAQ. In [2], 23-38.

21. Johnson, D.M., Dulmage, A.L., Mendelsohn, N.S. Connectivity and reducibility of graphs. *Canad. J. Math. 14*, (1962), 529-539.

22. Papadimitriou, C.H., Steiglitz, K. *Combinatorial Optimization*, Prentice Hall, Englewood Cliffs, N.J., 1982.

23. Rose, D.J., Tarjan, R.E., Lueker, G.S. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comp. 5*, (1976), 266-283.

24. Tarjan, R.E. Depth-first search and linear graph algorithms. *SIAM J. Comp. 1*, (1972), 146-160.

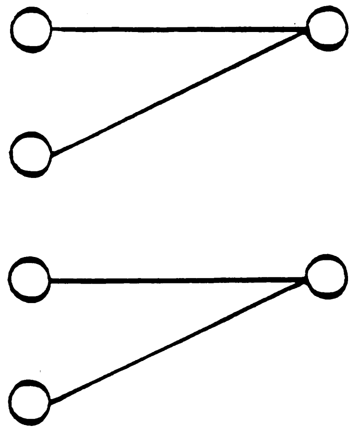
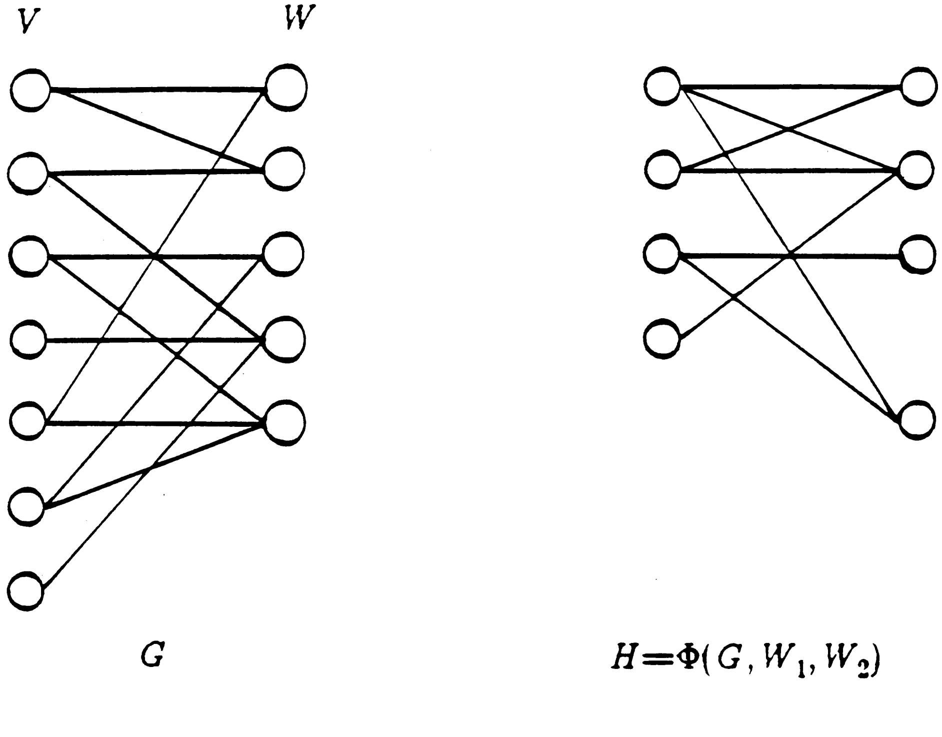


Figure 2.1



A product-bipartite graph where $W_1=\{w_1, w_2, w_3, w_4\}$ and $W_2=\{w_1, w_2, w_3, w_5\}$.

Figure 2.2

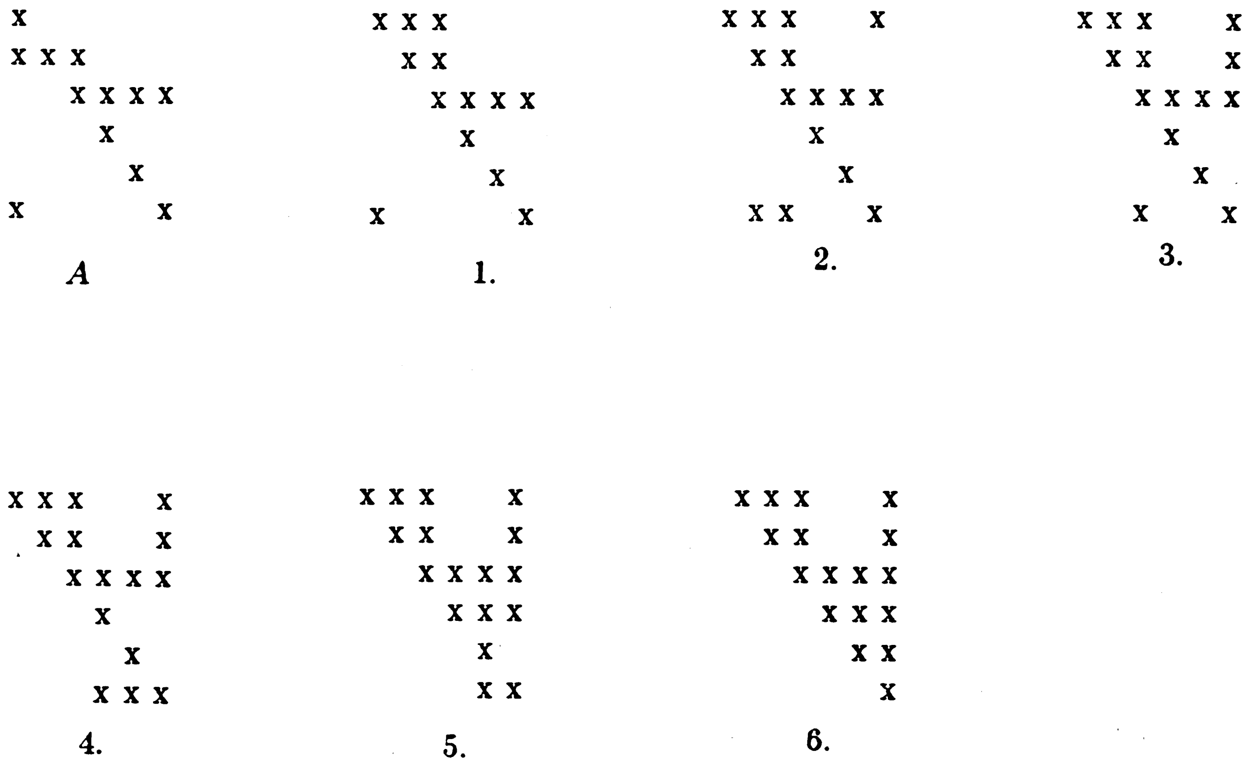


Figure 1.3

$$\begin{array}{ccc}
\begin{matrix} x & x & x & x & x & x \\ & x & & & & \\ & & x & & & \\ & & & x & & \\ & & & & x & \\ & & & & & x \end{matrix}
&
\begin{matrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \end{matrix}
&
\begin{matrix} x & x & x & x & x & x \\ & x & x & x & x & x \\ & & x & x & x & x \\ & & & x & x & x \\ & & & & x & x \\ & & & & & x \end{matrix}
\\
A & A^T A & R
\end{array}$$

Figure 1.4

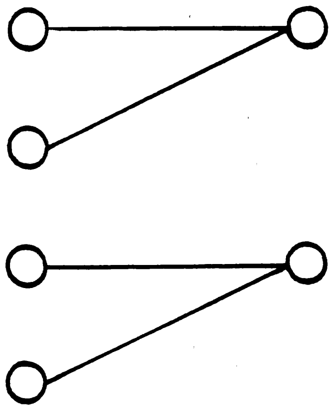
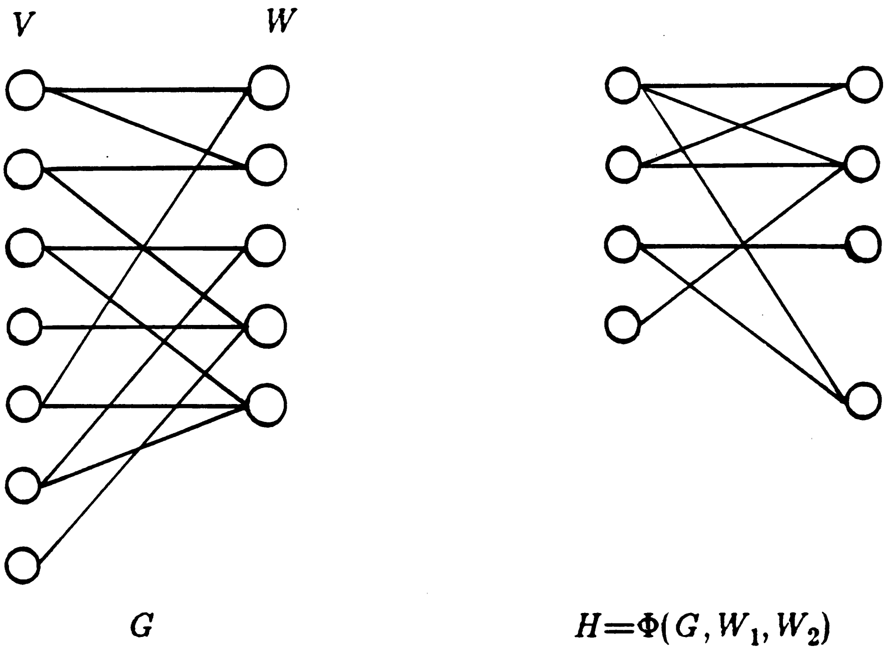
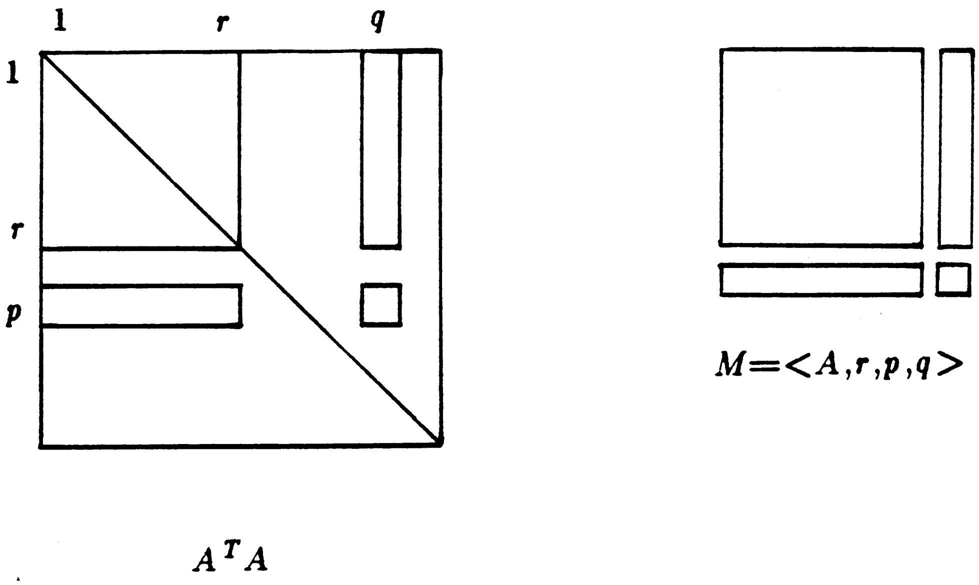


Figure 2.1



A product-bipartite graph where $W_1=\{w_1,w_2,w_3,w_4\}$ and $W_2=\{w_1,w_2,w_3,w_5\}$.

Figure 2.2

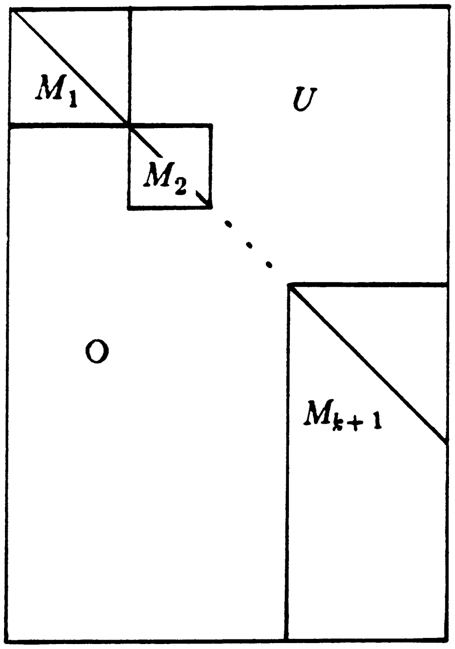


A^TA

$M = < A, r, p, q >$

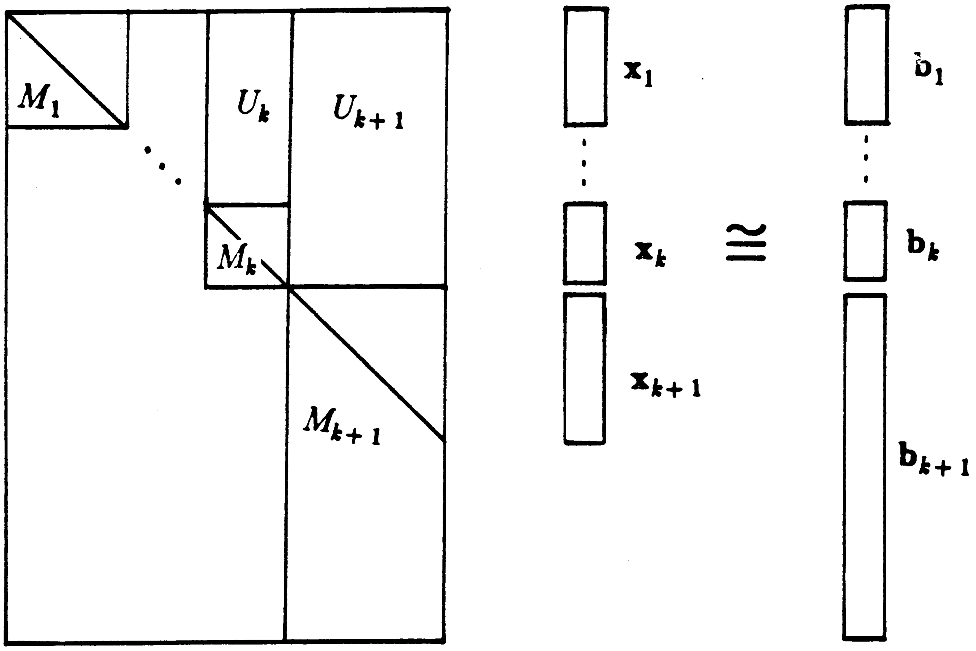
A special submatrix.

Figure 3.1



The Dulmage-Mendelsohn reordering of a matrix.

Figure 4.1



Block back-substitution.

Figure 4.2

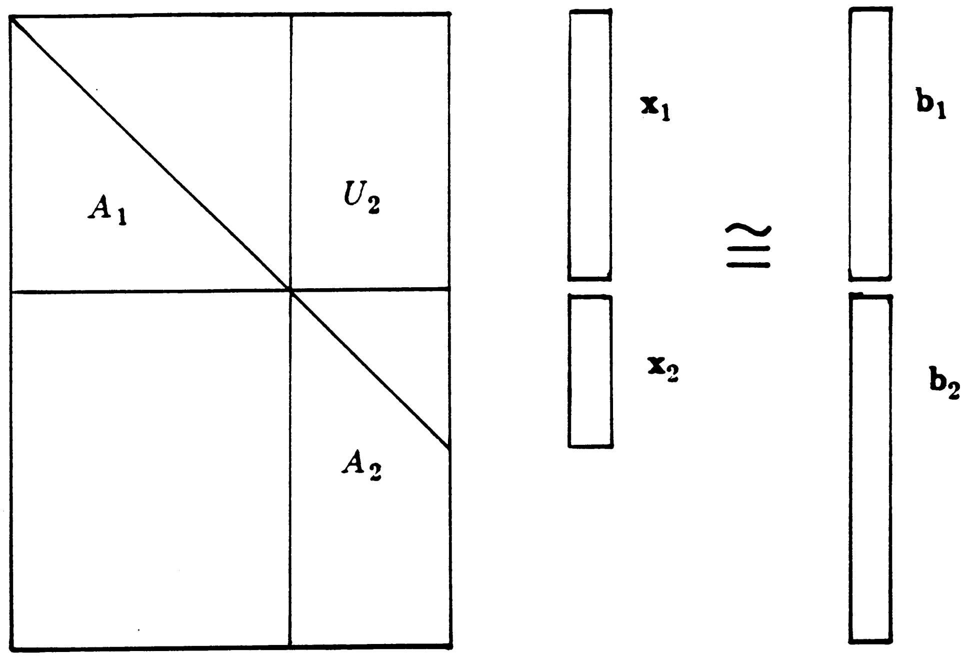


Figure 4.3