# MULTIFRONTAL COMPUTATION WITH THE ORTHOGONAL FACTORS OF SPARSE MATRICES *

SZU-MIN LU† AND JESSE L. BARLOW‡

**Abstract.** This paper studies the solution of the linear least squares problem for a large and sparse $m$ by $n$ matrix $A$ with $m \geq n$ by $QR$ factorization of $A$ and transformation of the right-hand side vector $b$ to $Q^T b$. A multifrontal-based method for computing $Q^T b$ using Householder factorization is presented.

A theoretical operation count for the $K$ by $K$ unbordered grid model problem and problems defined on graphs with $\sqrt{n}$-separators shows that the proposed method requires $O(N_R)$ storage and multiplications to compute $Q^T b$, where $N_R = O(n \log n)$ is the number of nonzeros of the upper triangular factor $R$ of $A$.

In order to introduce BLAS-2 operations, Schreiber and Van Loan's Storage-Efficient-WY Representation [SIAM J. Sci. Stat. Computing, 10(1989),pp. 55-57] is applied for the orthogonal factor $Q_i$ of each frontal matrix $F_i$. If this technique is used, the bound on storage increases to $O(n(\log n)^2)$.

Some numerical results for the grid model problems as well as Harwell-Boeing problems are provided.

**Key words.** Multifrontal $QR$ factorization, $\sqrt{n}$-separable graphs, Householder matrices

**AMS subject classifications.** 15A23, 65F50

## 1. Introduction.

We study the linear least squares problem:

$$\min_x ||Ax - b||_2, \tag{1}$$

where $x$ is a real $n$-vector and $b$ is a real $m$-vector. Orthogonal factorization is often used in methods for solving the linear least squares and eigenvalue problems. Let $A$ be an $m$ by $n$ large sparse matrix of full column rank with $m \geq n$. The $QR$ factorization of $A$ is $A = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$, where $R$ is an $n$ by $n$ upper triangular matrix, $Q$ is an $m$ by $m$ orthogonal matrix.

We apply $QR$ factorization of $A$ and transform the right-hand side vector $b$ to $Q^T b$, as follows.

$$\begin{aligned}
\min_x ||Ax - b||_2^2 &= || \begin{pmatrix} R \\ 0 \end{pmatrix} x - Q^T b||_2^2 \\
&= || \begin{pmatrix} R \\ 0 \end{pmatrix} x - \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} ||_2^2 \\
&= ||Rx - c_1||_2^2 + ||c_2||_2^2.
\end{aligned}$$

* The authors' research was supported by the National Science Foundation under grant no. CCR-9201612.

† Computer Science and Engineering Department, The Pennsylvania State University, University Park, PA 16802 (lu@cse.psu.edu).

‡ Computer Science and Engineering Department, The Pennsylvania State University, University Park, PA 16802 (barlow@cse.psu.edu).

The least squares solution is given by solving

$$Rx = c_1.$$

The problem is that if $b$ is not known in advance or if we have more than one $b$, we need to save the orthogonal matrix $Q$. Unfortunately, $Q$ is often larger and much denser than the factor $R$.

Instead of storing the first $n$ columns of $Q$, one often stores the orthogonal factor $Q$ implicitly, as follows. Let $A = H_1 H_2 \cdots H_n R$. The orthogonal matrix $Q$ is then expressed as

$$Q = H_1 H_2 \cdots H_n \ ,$$

where $H_i = I - h_i h_i^T$ is a Householder reflection that zeros out column $i$ of $A$ below the main diagonal. The vector $h_i$ is often referred to as a Householder vector and is zero in positions 1 through $i-1$. The orthogonal factor $Q$ can therefore be represented implicitly by the $m$ by $n$ lower trapezoidal matrix $H$:

$$H = (h_1 \ h_2 \ \cdots \ h_n),$$

which is referred to as the Householder matrix. A matrix-vector product $Q^T b$ can be computed efficiently from $H$ and $b$. The LINPACK routines SQRDC and SQRSL employ $H$ [7].

In [17], Gilbert, Ng, and Peyton analyzed the nonzero counts of the factors $Q$, $R$, and $H$ in terms of the sizes of separators in the column intersection graph $G_\cap(A)$ of $A$, where $G_\cap(A)$ is an undirected graph in which an edge joins two vertices whose columns share a nonzero row in $A$. This graph corresponds to the matrix of the normal equations $A^T A$. If $A$ is such that $G_\cap(A)$ has $\sqrt{n}$-separators for all its subgraphs, and if $m - n$ is of the same order as $n$, then $H$ is smaller than $Q$ only by a constant factor [17]. That is both $|Q|$, the number of nonzeros in the first $n$ columns of $Q$, and $|H|$, the number of nonzeros in $H$, are of $O(n\sqrt{n})$. Moreover, the difference between $|Q|$ and $|H|$ is likely to be relatively small if $m$ is much larger than $n$. Other results on the nonzero structures of the Householder matrix $H$ and the orthogonal factor $Q$ for a sparse matrix using $G_\cap(A)$ are, for example, given in [14, 26].

In this paper, we study the computation of orthogonal factor using the multifrontal $QR$ factorization [20, 24]. Associated with each row of the upper triangular factor $R$ , is a frontal matrix $F_i$. Likewise for each $F_i$, there is a frontal Householder matrix $Y_i$. Note that $Y_i$ is the $H$ matrix for $F_i$. Figure 1 is a small sample matrix $A$ and its column intersection graph. Figure 2 is the Householder matrix $H$ of $A$ and the elimination tree of $A^T A$. The frontal Householder matrices $Y_i$'s are given in Figure 3. The size of $H$ is $|H| = 106$, where the sum of sizes of $Y_i$'s is $\sum_{i=1}^{n} |Y_i| = 65$. The result of this paper provide an explanation for this dramatic difference.

We are going to present an efficient method for computing $Q^T b$ by using the frontal Householder matrices $Y_i$'s. In addition, this method is suitable for parallel computation because of the special structure of multifrontal matrices [25].

For the theoretical part, we study the $K$ by $K$ grid model problem and problems which are defined on $\sqrt{n}$-separable graphs under one assumption for the initial step.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| 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 | | | | |

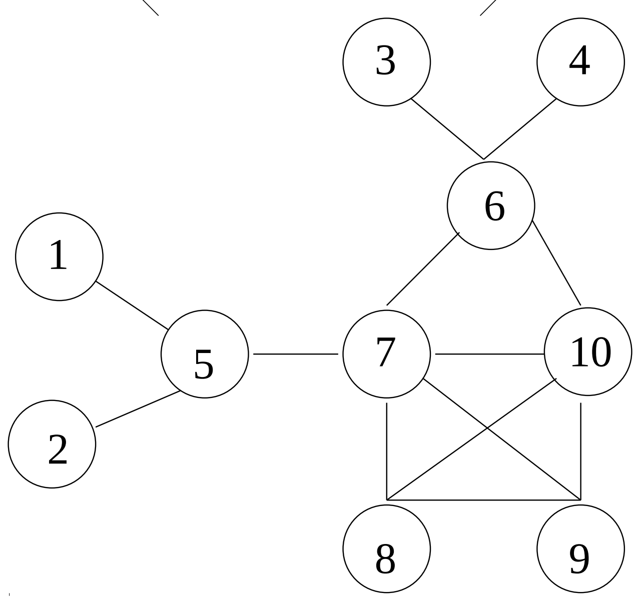


FIG. 1. *A sample matrix A and its column intersection graph*

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| x | | | | | | | | | |
| | x | | | | | | | | |
| | | x | | | | | | | |
| | | | x | | | | | | |
| | | | | x | | | | | |
| | | | | | x | | | | |
| | | | | | x | x | | | |
| | | | | | | | x | | |
| | | | | | | | | x | |
| | | | | | | x | x | x | x |
| | x | | | f | | f | f | f | f |
| | x | | | f | | f | f | f | f |
| | x | | | f | | f | f | f | f |
| | x | | | f | | f | f | f | f |
| | | | x | | f | f | f | f | f |
| | | | x | | f | f | f | f | f |
| | | | x | | f | f | f | f | f |
| | | | x | | f | f | f | f | f |
| | | x | | | f | f | f | f | f |
| | | x | | | f | f | f | f | f |
| | | x | | | f | f | f | f | f |
| | | x | | | f | f | f | f | f |
| | | | | | x | f | f | f | f |
| | | | | | x | f | f | f | f |
| | | | | | x | f | f | f | f |
| | | | | | x | f | f | f | f |

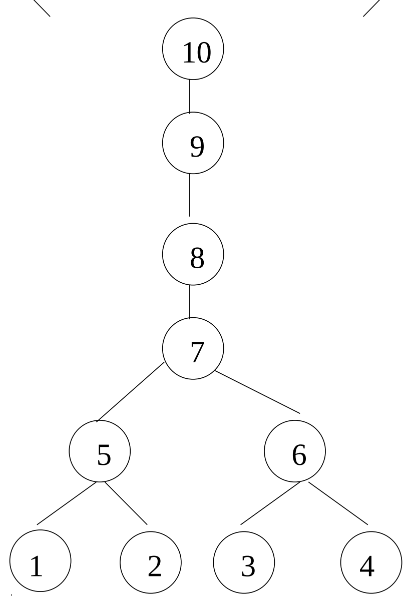


FIG. 2. *The Householder matrix H and the elimination tree for the matrix of Figure 1.*

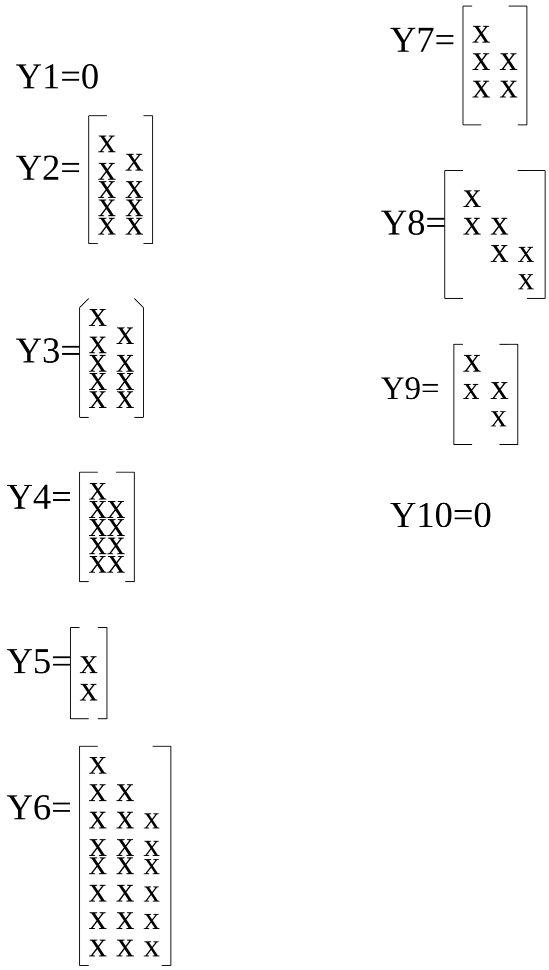


FIG. 3. *The frontal Householder matrices of A*

We are going to describe these problems in section 2. An $O(n \log n)$ bound is proven on the number of nonzeros of all frontal Householder matrices $Y_i$'s. We also count the number of nonzeros used in the WY representations of Bischof and Van Loan [5] and Schreiber and Van Loan [29]. We prove the bounds for the $K$ by $K$ grid model problem and problems which are defined on $\sqrt{n}$-separable graphs are $O(n \log n)$ and $O(n(\log n)^2)$, respectively. Note that these bounds are valid even if $m - n$ is of the same order of $n$.

The rest of the paper is organized as follows. Section 2 briefly reviews the model problem and the $\sqrt{n}$-separator problems, the multifrontal Householder $QR$ factorization, and the application of supernodes. Section 3 proposes a multifrontal-based method for computing $Q^T b$. Section 4 proves an upper bound on the nonzero counts of all $Y_i$'s. This section builds on the work of Lewis et.al. [20] for $K$ by $K$ grid problem. We extend their result to the $\sqrt{n}$-separator problem. Section 5 introduces BLAS-2 operations in computing $Q^T b$ by using the YTY representation of Schreiber and Van Loan [29] for the orthogonal factor $Q_i$ of each frontal matrix $F_i$. The upper bound on operation counts of that representation is also included. In section 6, we provide some numerical test results.

## 2. Background.

**2.1. The Model Problem.** Since the sparsity patterns of general sparse matrices are difficult to predict, our theoretical operation counts for sparse matrices are based on the model problem which is described in this section.

The model problem is motivated by the finite element method. Consider a $K$ by $K$ regular grid with $(K-1)^2$ small squares. A variable is assigned to each grid point. Associated with each square is a set of $s$ equations involving the four variables at the corners of the square. The assembly of these equations results in a large overdetermined system of equations:

$$Ax = b, \tag{2}$$

where $A$ is $m$ by $n$ with $m = s(K-1)^2$ and $n = K^2$ .

In our examples, we let $s = 4$ as in [20].

Figure 4 is an example for a 3 by 3 grid ordered by nested dissection ordering, the corresponding matrix $A$ and the upper triangular matrix $R$ are given in Figure 5.

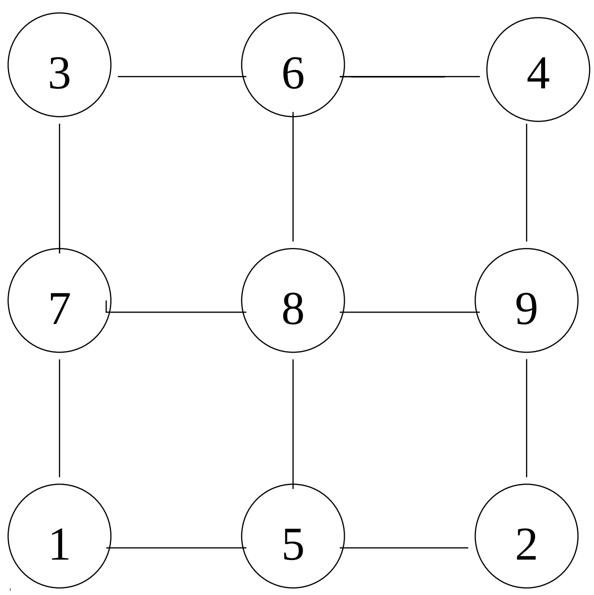


FIG. 4. *A 3 by 3 nested dissection ordered grid*

**2.2. The Extended Problem.** In addition to the model problem, we would like to study more general problems. Consider problems which are defined by graphs. Let $S$ be a class of graphs closed under the subgraph relation. That is if $G_1 \in S$ and $G_2$ is a subgraph of $G_1$, then $G_2 \in S$.

DEFINITION 2.1 ($\sqrt{n}$-SEPARABLE GRAPH). *A $\sqrt{n}$-separable graph is an $n$-vertex connected graph $G \in S$ with the following properties:*

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| 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 |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| 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 |

FIG. 5. *A sample matrix $A$ and its upper triangular factor $R$*

*There exist constants $\alpha < 1, \beta > 0$ such that $G$ can be partitioned into three sets $A, B, C$ such that no edge joins a vertex in $A$ with a vertex in $B$, neither $A$ nor $B$ contains more than $\alpha n$ vertices, and $C$ contains no more than $\beta\sqrt{n}$ vertices.*

Throughout the paper, we refer to the $\sqrt{n}$-separator matrices as the set of matrices whose column intersection graphs are members of the set of $\sqrt{n}$-separable graphs with the constants $\alpha$ and $\beta$ defined above.

**2.3. Multifrontal Householder $QR$ factorization Method.** We describe a multifrontal-based method by [20] for computing the upper triangular factor $R$ of $A$. The factorization uses the frontal structure inherent in multifrontal Cholesky algorithms and Householder transformations. A theoretical operation count for the model problem is given by Lewis et.al. [20] which indicates the factorization algorithm requires half the multiplications as Liu's algorithm [8, 22].

We begin this subsection with the definition of the elimination tree.

DEFINITION 2.2 (ELIMINATION TREE). *Given an $m$ by $n$ matrix $A$, such that $A^TA$ is irreducible, the elimination tree of $A^TA$ is a tree consisting of $n$ vertices each uniquely labeled by an integer from 1,2,3,...,n. Let $R$ denote the upper triangular factor of the $QR$ factorization of $A$. Then $j$ is the parent of vertex $i$ in the elimination tree if $j$ is the leading off-diagonal nonzero in the ith row of $R$.*

Consider the matrix $A$ and the factor $R$ of $A$ given in Figure 5. The corresponding elimination tree is given in Figure 6.

The elimination tree is a tool for ordering and organizing the computation in the multifrontal method. In order to compute the $i^{th}$ row of $R$, all the rows corresponding to node $i$'s descendants in the elimination tree must be computed. That is, row $i$ cannot be computed until its children's rows are computed.

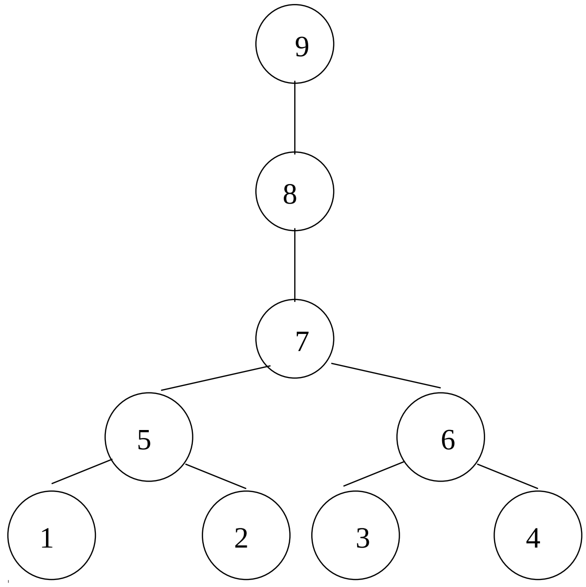


FIG. 6. *The elimination tree of $R$ in Figure 5*

The multifrontal $QR$ factorization method uses the elimination tree to determine the required information for forming each frontal matrix. We explain this in detail below.

We begin by defining the following notation:

1. Let $i$ denote node $i$ in the elimination tree as well as the $i^{th}$ column of $A$.
2. Let $A[i]$ be the matrix whose rows are those rows of $A$ which have their leading nonzeros in column $i$.

Let $j$ be a leaf in the elimination tree. During each frontal stage, only $A[j]$ contributes to building the frontal matrix $F_j$. That is the nonzero structure of the $j^{th}$ row of the upper triangular factor $R$ is completely dependent on $A[j]$. One then computes the $QR$ factorization of $F_j$ resulting in $Q_j^T F_j = R_j$, where $R_j$ is an upper triangular or usually trapezoidal factor. The first row of $R_j$ corresponds to the $j^{th}$ row of the factor $R$; the remaining part of $R_j$ is saved as update matrix $U_j$ which is used by $j$'s parent.

Now consider an internal node $i$. We assemble the frontal matrix $F_i$ by collecting all rows of $A[i]$ and all the update matrices from the children of $i$. We then compute the $QR$ factorization of $F_i$, use the first row of the upper triangular factor $R_i$ to fill the $i^{th}$ row of $R$ and save the update matrix $U_i$ for $i$'s parent.

The update matrices can be stored and retrieved in a last-in/first-out (i.e. stack) manner if the nodes of the elimination tree are ordered by a postordering. The use of a stack for update matrices is due to Duff and Reid [9].

Below we outline the multifrontal $QR$ factorization in an algorithm.

---

ALGORITHM 2.3. **(Multifrontal QR Factorization):**
**For** $j = 1$ **To** *number of tree nodes* **Do**

1. *Assemble the frontal matrix for vertex $j$, consisting of all rows of $A$ with first*

*nonzero in column $j$ and the update matrices from the children of vertex $j$.*

$$\begin{pmatrix} \leftarrow A[j] \rightarrow \\ \leftarrow U_{c_1} \rightarrow \\ \vdots \\ \leftarrow U_{c_s} \rightarrow \end{pmatrix}$$

*where the children of vertex $j$ are vertices $c_1, \ldots, c_s$.*

2. *Compute the $QR$ factorization of the frontal matrix such that*

$$Q_j^T F_j = R_j = \begin{pmatrix} r_{jj} & r_{jj_1} & \ldots & r_{jj_t} \\ 0 & U_j & & \\ \vdots & & & \\ 0 & & & \end{pmatrix}.$$

3. *Save the first row of $R_j$ , $(r_{jj}, r_{jj_1}, \ldots, r_{jj_t})$ for the jth row of $R$, save the remaining part as update matrix $U_j$ for $j$'s parent.*

**End For**

---

The data flow of multifrontal $QR$ factorization is given in Figure 7.

Matstoms [24] implemented the multifrontal method and solved (1) by the corrected semi-normal equation(CSNE).

**2.4. Supernodes.** In order to use dense operations and reduce data movement, we can apply the supernode concept to the frontal method in section 2.3.

We begin by defining the fundamental supernodes as follows.

DEFINITION 2.4 (FUNDAMENTAL SUPERNODE). *A fundamental supernode, with respect to a postordering elimination tree, is a set of maximal number of contiguous vertices, $S_j = \{i_{j_1}, i_{j_2}, \ldots, i_{j_{|S_j|}}\}$, such that $i_{j_k}$ is the only son of $i_{j_{k+1}}$ and the structure of row $i_{j_{k+1}}$ in the factor $R$ is identical to the structure of the off-diagonal part of row $i_{j_k}$, $k = 1, 2, \ldots, |S_j| - 1$. Furthermore, $|S_j|$ is called the size of the supernode.*

Duff and Reid [9] explored the use of supernodes in the multifrontal method. They amalgamate vertices if one of the following conditions is satisfied.

1. if they form a *fundamental supernode.*
2. if the number of fully summed vertices in the parent and the child is less than a user defined parameter NEMIN.

Details of the implementation and complete study on the efficiency of the value of NEMIN on the performance of multifrontal $QR$ factorization are given by Matstoms [24] and Puglisi [27].

We build a supernodal elimination tree by substituting a single node for all the nodes belonging to the same supernode in the original elimination tree. We formally define this elimination tree below.

DEFINITION 2.5 (SUPERNODAL ELIMINATION TREE). *Let $A$ be as in Definition 2.2. Let the set $\{1, 2, \ldots, n\}$ be partitioned into $\{1, 2, \ldots, n\} = S_1 \cup \ldots \cup S_{ns}$ where*

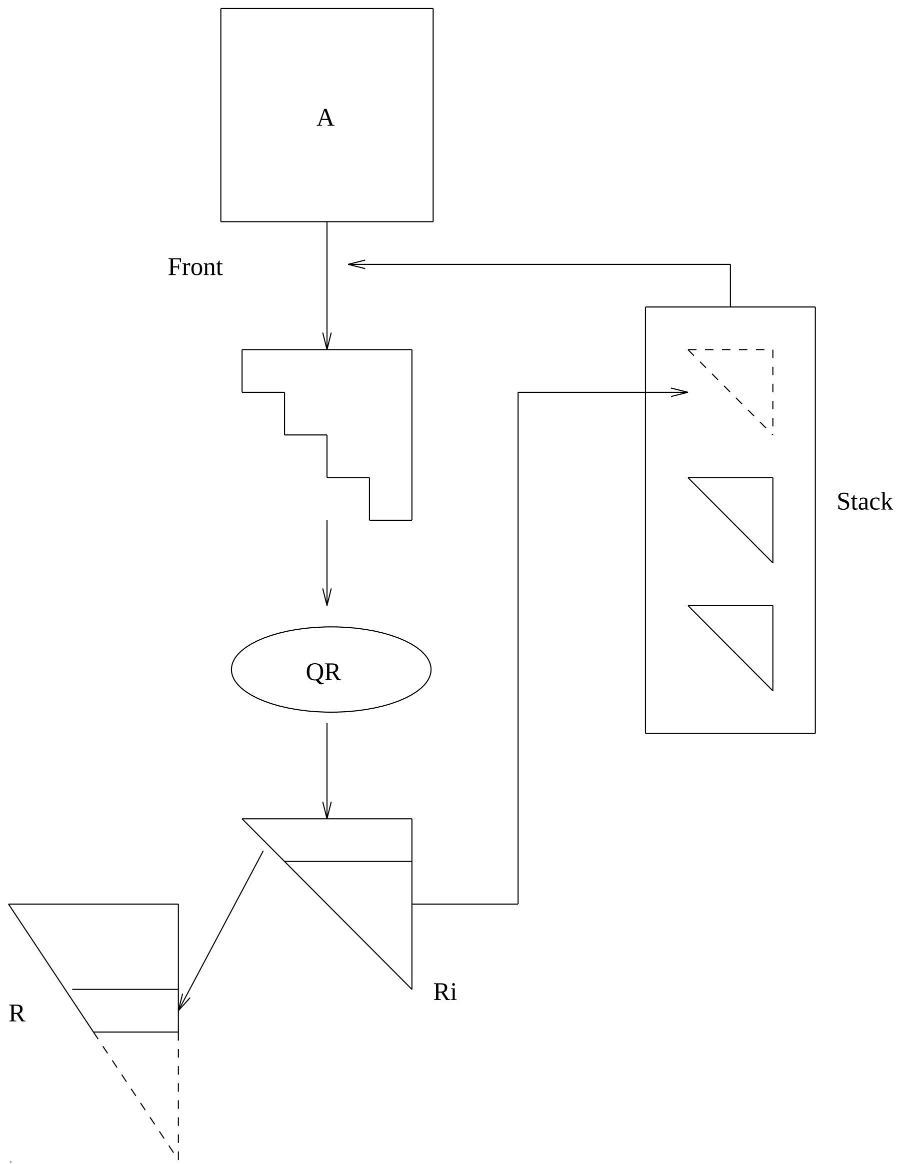


FIG. 7. *Data flow of multifrontal QR factorization*

*$S_1, \ldots, S_{ns}$ are the supernodes of $A$ by Definition 2.4. Then $S_j$ is the parent of $S_i$ in the supernodal elimination tree if for some vertex $v \in S_j$ and $w \in S_i$, $v$ is the parent of $w$ in the elimination tree of $A$.*

Associated with each supernode is an $m_j \times n_j$ frontal matrix $F_j$ where $n_j$ is number of nonzero elements in the rows of $\{S_j\} \cup Tree(S_j)$ and $Tree(S_j)$ is the tree rooted at $S_j$.

We can use the supernodal elimination tree as the representation of the order of the multifrontal factorization process as follows. The merge operation corresponds to computing the $QR$ factorization of a frontal matrix composed of all the update matrices, $U_{j_1}, U_{j_2}, \ldots, U_{j_t}$, and rows of $A$ with leading nonzeros from the set $\{j_1, j_2, \ldots, j_t\}$, the indices of the supernode. With the $QR$ factorization of a frontal matrix, we compute multiple rows of the factor $R$ (i.e., those rows belonging to the set of indices of the supernode). Moreover, we spend less time manipulating the update matrices by reducing the data movement. The use of supernodes avoids the redundancy of separate merges by increasing the size of the frontal matrix and combining these merges into the application of one block Householder transformation. The amount of fill in the factor $R$ remains unchanged.

**3. The Proposed Method.** In this section, we present an efficient method for storing frontal Householder vectors and, thus, computing $Q^T b$ by applying the multifrontal Householder $QR$ factorization method.

Instead of storing the Householder matrix $H$ itself, we store the frontal Householder matrix $Y_i$ of each multifrontal matrix $F_i$ of $A$. Recall that $Y_i$ is a lower trapezoidal matrix in which each column $k$ is a Householder vector $w_k^{(i)}$ of $F_i$ such that :

$$Q_i^T = (I - w_{n_i}^{(i)} w_{n_i}^{(i)T}) \cdots (I - w_1^{(i)} w_1^{(i)T}) \tag{3}$$

and

$$Q_i^T F_i = R_i.$$

Here, $Q_i^T$ is the orthogonal factor which is used to factor the $m_i \times n_i$ frontal matrix $F_i$. The first $|S_i|$ rows of $R_i$ are used to fill the corresponding rows of $R$, where $|S_i|$ is the size of the supernode $S_i$.

From the multifrontal process, we have

$$Q^T A \equiv Q_{ns}^T \circledast Q_{ns-1}^T \circledast \cdots \circledast Q_1^T \circledast A, \tag{4}$$

where $ns$ is the number of supernodes in the supernodal elimination tree, $Q_i^T$ is the orthogonal factor which is used to factor the $m_i$ by $n_i$ frontal matrix $F_i$. $\circledast$ is called "Extended Multiplication". The extended product in $Q_i^T \circledast A$ factors the part of $A$ which contributes to form frontal matrix $F_i$. It follows that :

$$Q^T b \equiv Q_{ns}^T \circledast Q_{ns-1}^T \circledast \cdots \circledast Q_1^T \circledast b. \tag{5}$$

From (3) we have :

$$\begin{aligned} Q^T b \equiv ((I - w_{n_{ns}}^{(ns)} w_{n_{ns}}^{(ns)T}) \cdots (I - w_1^{(ns)} w_1^{(ns)T})) \circledast \cdots \\ \circledast ((I - w_{n_1}^{(1)} w_{n_1}^{(1)T}) \cdots (I - w_1^{(1)} w_1^{(1)T})) \circledast b. \end{aligned} \tag{6}$$

Since $R_i$ in the $QR$ factorization of a frontal matrix $F_i$,

$$Q_i^T F_i = R_i,$$

is invariant under row orderings of $F_i$, we can therefore sort the rows of the frontal matrices by the column indices of their leading entries. Therefore, we achieve a block triangular structure for each frontal matrix. Block triangular matrices are efficiently factored by respecting the block structure. It follows that the corresponding frontal Householder matrix $Y_i$ is also a block matrix. Figure 8 is a small example of a block triangular frontal matrix $F_i$ and the corresponding frontal Householder matrix $Y_i$.

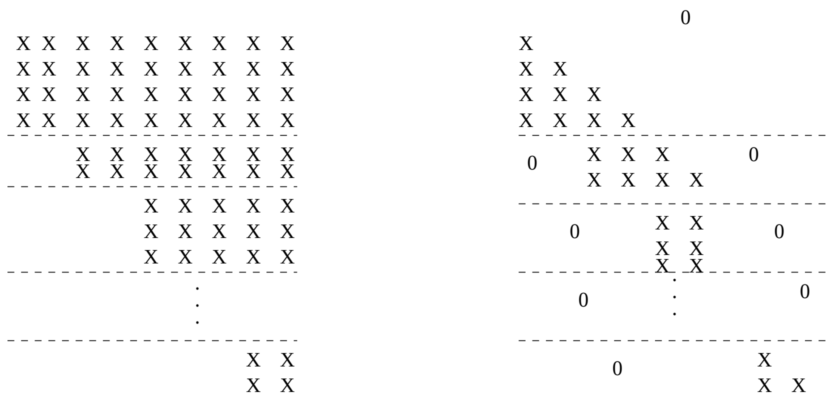

FIG. 8. *A frontal matrix of a supernode and the corresponding Y matrix*

The data structure for storing all $Y_i$'s is not complicated; we need only a real data buffer to store the nonzeros of $Y_i$'s and an integer data buffer to record the row indices of each frontal matrix.

The required storage for computing $Q^T b$ using (6) is the same as the number of nonzeros of all the $Y_i$'s. Since each Householder vector, $w_i$, in (6) is applied twice, the required number of multiplications is twice the number of nonzeros of all the $Y_i$'s.

**4. Theoretical Results.** In this section, we develop an upper bound on the number of nonzeros in all $Y_i$'s for the model problem and extend the result to the $\sqrt{n}$ -separator problem. For notational convenience, we denote $|Y_i|$ as the number of nonzeros in $Y_i$ and $|Y| = \sum_{i=1}^{ns} |Y_i|$ where $ns$ is the number of supernodes. Thus $|Y|$ is the quantity we wish to bound.

Lipton, Rose, and Tarjan's "generalized nested dissection" ordering [21], which include the separators in the recursive call, guarantees bounds of $O(n \log n)$ on fill-in and generates balanced elimination trees for the $\sqrt{n}$-separator problem, we assume all the matrices are ordered by that column ordering in our analysis for the $\sqrt{n}$-separator problem in section 4.2.

On the other hand, George's original, simpler form of nested dissection [10], which does not include the separators in the recursive call, is actually sufficient for some special classes of the $\sqrt{n}$-separable graphs: planar graphs, graphs of bounded genus or bounded excluded minor, and two-dimensional finite element meshes of bounded aspect ratio [16]. As a result, our analysis in section 4.1 for the model problem uses George's nested dissection ordering.

**4.1. The Model Problem.** If George's nested dissection ordering [10] is applied to the model problem, then each internal node has at most two children. To count the number of nonzeros in $Y_i$'s, we begin with the initial frontal matrices. Let $F_j$ be the frontal matrix associated with a leaf node of the elimination tree. Since $F_j$ has at most $4s$ rows and nine columns, the number of nonzeros in $Y_j$ is actually a constant. As a result, the sum of the number of nonzeros in all these $Y_j$'s is $O(n)$ in total.

Now, we consider the internal nodes. The special structure of the model problem implies that the merge process for forming a frontal matrix only involves two trapezoidal matrices.

To simplify the proof of the theorems and obtain an upper bound of the nonzero count, we use the two assumptions as in [20] :

1. the two update trapezoidal matrices are full triangular matrices.
2. the two update matrices are $u$ by $u$, and $v$ by $v$ respectively. Also they have t columns in common.

Let $C(u, v, t)$ denote the total number of nonzeros in all Householder vectors $w_k^{(i)}$'s such that $Q_i^T = (I - w_i^{(n_i)} w_i^{(n_i)T}) \cdots (I - w_i^{(1)} w_i^{(1)T})$ and $Q_i^T F_i = R_i$, where the first row of $R_i$ is used to fill the $i$th row of the upper triangular factor $R$ of $A$. The unreduced frontal matrix has the form given in Figure 9.

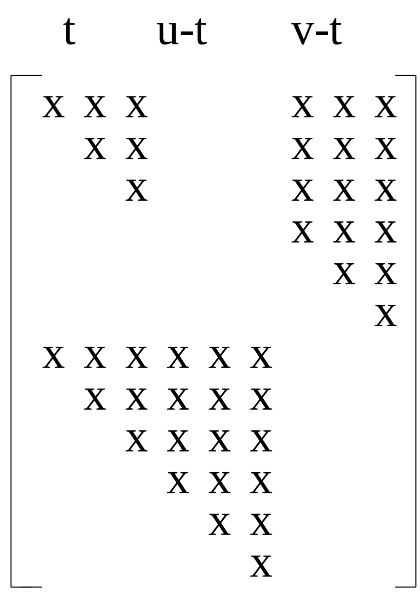


FIG. 9. *Unreduced frontal matrix*

We then have :

$$\begin{aligned} C(u, v, t) &= \sum_{i=1}^{t} (i+1) + \sum_{i=t+1}^{u+v-t} (t+1) \\ &= \frac{t(t+3)}{2} + (t+1)(u+v-2t). \end{aligned}$$

We can use the concept of "bordered $K$ by $K$ grids" [12] to perform the merge operation. Let $\Theta(k, i)$ be the number of nonzeros in the matrix $Q$, which is used to factor a $K$ by $K$ grid which is bordered on $i$ sides. According to [20, 12], the following recurrence relations are valid.

$$\begin{aligned} \Theta(K, 4) &= 4\Theta(\frac{K}{2}, 4) + 2C(2K, 2K, \frac{K}{2}) + C(3K, 3K, K) \\ \Theta(K, 3) &= 2\Theta(\frac{K}{2}, 3) + 2\Theta(\frac{K}{2}, 4) + 2C(2K, \frac{3K}{2}, \frac{K}{2}) + C(\frac{5K}{2}, \frac{5K}{2}, K) \\ \Theta(K, 2) &= \Theta(\frac{K}{2}, 2) + 2\Theta(\frac{K}{2}, 3) + \Theta(\frac{K}{2}, 4) + C(\frac{3K}{2}, K, \frac{K}{2}) + C(2K, \frac{3K}{2}, \frac{K}{2}) + C(\frac{5K}{2}, \frac{3K}{2}, K) \\ \Theta(K, 0) &= 4\Theta(\frac{K}{2}, 2) + 2C(K, K, \frac{K}{2}) + C(K, K, K) \end{aligned}$$

Because we are interested in the $K$ by $K$ unbordered grid, $\Theta(K,0)$ is desired.

Using the approaches in [12] and our definition of $C(u,v,t)$, we have:

$$|Y| = \Theta(K,0) = \frac{31}{4}K^2 log_2\, K + \frac{29}{6}K^2 + 32K log_2\, K - 35K \tag{7}$$

According to [12], the number of nonzeros of the $R$ factor of a $K$ by $K$ unbordered grid matrix is $N_R = \frac{31}{4}K^2\log_2\, K - \frac{73}{3}K^2 + O(K\log_2\, K)$. It follows that $|Y| \approx N_R$ as $n \to \infty$.

The following theorem summarizes the result in this section.

THEOREM 4.1. *For the model problem, the required storage and number of multiplications to compute $Q^T b$ as in (6) is $O(N_R)$ where $N_R$ is the number of nonzeros of the upper triangular matrix $R$.*

We note that since $|Y| \approx N_R + 30K^2$, for most practical values of $n$, $|Y| \approx cN_R$, for a constant $c > 1$.

**4.2. The Extended Model Problem.** We prove the bound of $O(n\log n)$ on $|Y|$ for $\sqrt{n}$-separator matrices. Let $A$ be a $\sqrt{n}$-separator matrix whose columns are ordered by "generalized nested dissection" ordering [21]. If $A$ has no more than $n_0 = (\beta/(1-\alpha))^2$ columns, this recursive numbering algorithm numbers the unnumbered columns arbitrarily.

In order to limit the initial $|Y_i|$, we assume that each $s_i$ ,the number of rows of $A[i]$, is much smaller than $n$ and could be treated as a constant. That is there exists a constant $s$ such that $s_i \leq s$ for all $i$. Note that this is a reasonable assumption as long as $m = O(n)$ because of the fact that $\sum_{i=1}^{n} s_i = m$ and we are studying the matrices after column ordering which generally permutes relatively fuller columns toward the end of the matrices to reduce fill-in.

LEMMA 4.2. *Let $J$ denote the set of leaf nodes of the elimination tree of $A$. Then $\sum_{j\in J} |Y_j| = O(n\log n)$.*

*Proof.* Let $j$ be a leaf node and $F_j$ is the corresponding frontal matrix with $m_j$ rows and $n_j$ columns.

From the definition of frontal matrices, $F_j$ is identical to $A[j]$ thus $m_j = s_j$. Since $|Y_j| \leq m_j \times n_j = s_j \times n_j$ we have

$$\begin{aligned}\sum_{j\in J} |Y_j| &< \sum_{j\in J} s_j \times n_j \\ &\leq s \times \sum_{j\in J} n_j \\ &< s \times N_R.\end{aligned}$$

According to [21], $N_R$ is $O(n\log n)$. We then have $\sum_{j\in J} |Y_j| = O(n\log n)$.

□

We now consider the internal nodes using the supernodal elimination tree. Since we apply the generalized nested dissection ordering and the special property of the $\sqrt{n}$-separable graphs, an internal supernode $S_j$ of the supernodal elimination tree is actually a collection of the tree nodes corresponding to those vertices of $C$ which are not previously numbered, where $C$ is the separator of the subgraph corresponding to

$S_j$ and the subtree rooted at $S_j$.

From the process of multifrontal $QR$ factorization, we have that $n_j$ is the number of nonzeros in the $i_1^{th}$ row of the upper triangular factor $R$, where $i_1$ is the vertex in $S_j$ with lowest number. That is $n_j$ is the number of fill-in edges whose lower numbered vertex is $i_1$. Suppose the recursive numbering algorithm is applied to an $n$-vertex graph $G$ with $\ell$ vertices previously numbered. If $G$ has $n$ vertices, then by the definition of separator $|S_j| \leq \beta\sqrt{n}$, thus we have

$$\begin{aligned} n_j &= |S_j| + \ell \\ &\leq \beta\sqrt{n} + \ell. \end{aligned}$$

LEMMA 4.3. *Let $I$ denote the set of internal supernodes of the supernodal elimination tree. Then $\sum_{j \in I} |Y_j| = O(n \log n)$.*

*Proof.* The proof is similar to the one for the fill-in bound of Lipton, Rose, and Tarjan's work in [21].

The construction of $F_j$ involves only those $A[i]'s$, where $i$ is a vertex in $S_j$, and the two upper triangular or trapezoidal update matrices from the two children of $S_j$ in the supernodal elimination tree. Let the two update matrices be $u$ by $u$ and $v$ by $v$ respectively. Then $u + v \leq \ell + 2\beta\sqrt{n}$. Note that the two update matrices have $\beta\sqrt{n}$ columns in common. That is the size of the separator. In order to get the maximum of $|Y_j|$, we assume the first $\beta\sqrt{n}$ columns are those common columns. A frontal matrix $F_j$ with row reordering according to their leading nonzeros has the form given in Figure 10.

We then have:

$$\begin{aligned} |Y_j| &\leq \sum_{i=1}^{\beta\sqrt{n}} (i + 1 + s\beta\sqrt{n}) + \sum_{i=\beta\sqrt{n}+1}^{u+v-\beta\sqrt{n}} ((s+1)\beta\sqrt{n} + 1) \\ &+ \frac{1}{2}((s+1)\beta\sqrt{n} + 1)^2 \\ &\leq ((s + \frac{1}{2})\beta\sqrt{n} + \frac{3}{2})\beta\sqrt{n} + (s+2)\beta\ell\sqrt{n} \\ &+ \frac{1}{2}((s+1)\beta\sqrt{n} + 1)^2 \\ &= \frac{1}{2}(s^2 + 4s + 2)\beta^2 n + (s+2)\beta\ell\sqrt{n} + (s + \frac{5}{2})\beta\sqrt{n} + \frac{1}{2} \\ &\leq c_1 n + c_2 \ell\sqrt{n} + c_3\sqrt{n}, \end{aligned}$$

where $c_1 = \frac{1}{2}(s^2 + 4s + 2)\beta^2$, $c_2 = (s+2)\beta$, and $c_3 = (s+3)\beta$.

Assume the subgraph corresponding to $Tree(S_j) \cup \{S_j\}$ has $n$ vertices , of which $\ell$ are previously numbered. Let $f(\ell, n)$ be the maximum of $\sum_{i \in SN_j} |Y_i|$, where $SN_j$ is a set whose members are the supernodes in $Tree(S_j) \cup \{S_j\}$. Then

$$f(\ell, n) \leq |Y_j| + max\{f(\ell_1, k_1) + f(\ell_2, k_2)\} \tag{8}$$

$$\leq c_1 n + c_2 \ell\sqrt{n} + c_3\sqrt{n} + max\{f(\ell_1, k_1) + f(\ell_2, k_2)\}, \tag{9}$$

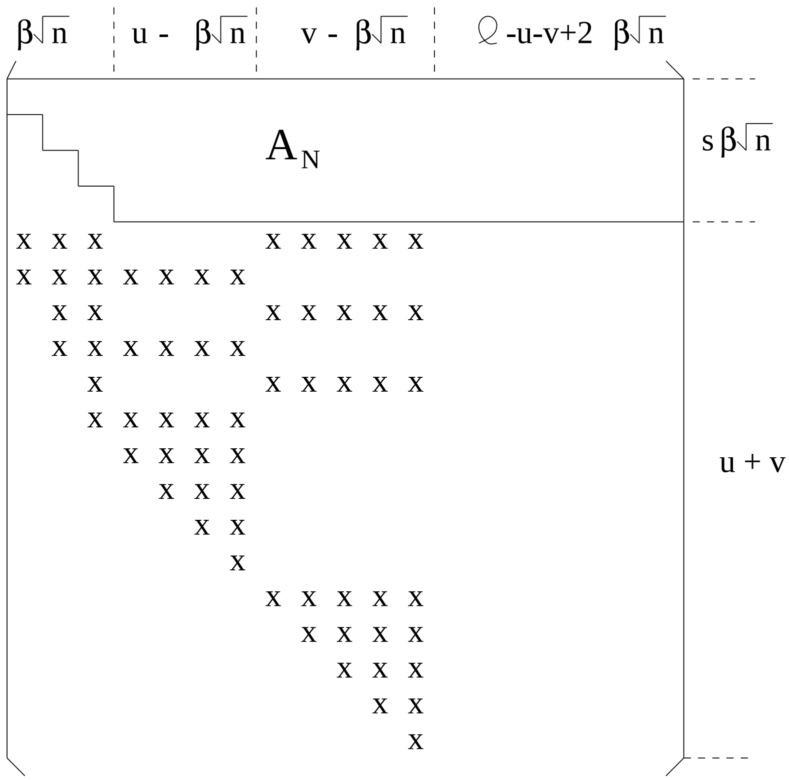


FIG. 10. *A sample frontal matrix with row-reordering*

where the maximum is taken over values satisfying

$$\ell_1 + \ell_2 \le \ell + 2\beta\sqrt{n}, \tag{10}$$

$$n \le k_1 + k_2 \le n + \beta\sqrt{n}, \text{ and} \tag{11}$$

$$(1-\alpha)n \le k_i \le \alpha n + \beta\sqrt{n} \text{ for } i = 1, 2. \tag{12}$$

An analysis similar to [21] (pp. 349-350, Theorem 2) shows that

$$f(\ell, n) \le c_4(n + \ell)\log n + c_5 \ell\sqrt{n},$$

where $c_4$ and $c_5$ are some suitably large constants.

Since $\sum_{S_j \in I} |Y_j| = f(0, n)$, the desired bound of $O(n \log n)$ on $\sum_{S_j \in I} |Y_j|$ follows. □

From Lemma 4.2 and 4.3 we have $|Y| = O(n \log n)$.

The following theorem summarizes the results of this section.

THEOREM 4.4. *Let $A$ be an $m$ by $n$ matrix which is defined on a generalized nested dissection ordered $\sqrt{n}$-separable graph. If the number of rows of each $A[i]$ is bounded by a constant , then the proposed method for computing $Q^T b$ requires $O(n \log n)$ storage and multiplications.*

From Theorem 4.4, the proposed method for computing $Q^T b$ is more efficient than using the Householder matrix $H$ or the orthogonal factor $Q$ itself when $m - n$ is of the order of $n$.

**5. Introducing BLAS-2 Operations.** In order to introduce BLAS-2 operations in (6), we use the YTY Representation [29], also called the "storage-efficient WY-representation. Here each $Q_i^T$ can be written as:

$$Q_i^T = I - Y_i T_i Y_i^T, \tag{13}$$

where $Y_i$ is the frontal Householder matrix of $F_i$ as defined before and $T_i$ is an $n_i$ by $n_i$ lower triangular matrix which is computed by the following algorithm:

$$T_i^{(1)} = 1$$

$$T_i^{(k)} = \begin{pmatrix} T_i^{(k-1)} & 0 \\ Z_i^{(k)} & 1 \end{pmatrix}, \; k = 2, \ldots, n_i$$

where $Z_i^{(k)} = -w_i^{(k)T} Y_i^{(k-1)} T_i^{(k-1)}$. and $Y_i = Y_i^{(n_i)}$, $T_i = T_i^{(n_i)}$.

From (6) and (13) we have :

$$(14) Q^T b = (I - Y_{ns} T_{ns} Y_{ns}^T) \circledast (I - Y_{ns-1} T_{ns-1} Y_{ns-1}^T) \circledast \cdots \circledast (I - Y_1 T_1 Y_1^T) \circledast b.$$

The structure in (14) is suitable for parallel computing; each $Y_i$ and $Y_j$ are independent blocks if node $i$ and node $j$ are not ancestor and descendant in the elimination tree. That is also true for the $T$ matrices. As a result, the matrix-vector computations $(I - Y_i T_i Y_i^T) \circledast b$ and $(I - Y_j T_j Y_j^T) \circledast b$ can be done simultaneously and each matrix-vector computation can be performed in parallel. It follows that the computing time of (14) is based on the height of the supernodal elimination tree and the communication time among the processors.

Note that the required number of multiplications to compute $Q^T b$ by (14) is $2|Y| + |T|$.

LEMMA 5.1. *The required storage and number of multiplications to compute $Q^T b$ for the $K$ by $K$ grid model problem using BLAS-2 operations is $O(N_R)$.*

*Proof. By the same argument as in section 4.1 and redefine $C(u, v, t) = \frac{1}{2}(u + v - t)^2$, we can prove that $|T| = \frac{99}{4} K^2 log_2 K + O(K^2) \approx 3N_R$. It follows that the storage requirement is $4N_R$ and the required number of multiplications is $5N_R$.*

□

THEOREM 5.2. *Let $A$ be an $m$ by $n$ matrix which is defined on a generalized nested dissection ordered $\sqrt{n}$-separable graph. If the number of rows of each $A[i]$ is a constant , then the required storage and number of multiplications for computing $Q^T b$ using BLAS-2 operations is $O(n(\log n)^2) = O(N_R \log n)$.*

*Proof.* Use the same argument as in section 4.2 and redefine $f(\ell, n)$ as the maximum of $\sum_{i \in SN_j} |T_j|$, where

$$|T_j| = \frac{1}{2}(n_j)^2$$
$$= \frac{1}{2}(\beta\sqrt{n} + \ell)^2.$$

Here again $\ell$ is number of vertices that have already been labeled. Similar to (8)-(9) we define

$$f(\ell, n) \leq \frac{1}{2}n^2 \leq \frac{1}{2}(n_0)n, \;\; n \leq n_0$$

$$f(\ell, n) \leq |T_j| + max\{f(\ell_1, k_1) + f(\ell_2, k_2)\} \tag{15}$$

$$\leq \frac{1}{2}\beta^2 n + \beta\ell\sqrt{n} + \ell^2 + max\{f(\ell_1, k_1) + f(\ell_2, k_2)\}, \;\; n > n_0 \tag{16}$$

where the maximum is again taken over the set (10)-(12). Here $n_0 < n$ which is independent of $n$. We claim that the solution for all $n \geq 1$,

$$f(\ell, n) \leq c_4 n(\log_2 n)^2 + c_5\ell^2 \log_2 n + c_6\ell\sqrt{n}\log n,$$

where $c_4$, $c_5$ and $c_6$ are constants. The value $f(0, n) = |T| = \sum_{i\in NS} |T_i|$. This claim can be shown by induction on $n$ and using the approach given by Lipton, Rose, and Tarjan in [21] (pp. 349-350, Theorem 2). The proof is as follows.

Let $n$ be large and suppose the claim is true for values smaller than $n$. Then the recurrences (15) and (16) give us

$$\begin{aligned} f(\ell, n) \leq{} & c_4 n(\log_2 n)^2 + c_5\ell^2 \log_2 n \\ & + (2c_4 \log_2(1-\epsilon) + 4c_5\beta^2 + 2c_6\beta\sqrt{\alpha})n \log_2 n + (4c_5\beta + \sqrt{\alpha}c_6)\ell\sqrt{n}\log n \\ & + (c_5 \log_2(1-\epsilon) + \frac{1}{2})\ell^2 \\ & + \{c_4(\log_2(1-\epsilon))^2 + (4c_5\beta^2 + 2c_6\beta)\log_2(1-\epsilon) + \frac{1}{2}\beta^2\}n + h(n), \end{aligned}$$

where $h(n)$ is of order $O(\sqrt{n}(\log_2 n)^2)$ and $\epsilon = \alpha + \beta/\sqrt{1+n_0}$. Clearly, $n_0$ must be large enough that $\epsilon < 1$. Suppose we choose $c_5$ such that $c_5 \log_2(1-\epsilon) + \frac{1}{2} \leq 0$, choose $c_6$ large enough such that $4c_5\beta + \sqrt{\alpha}c_6 \leq c_6$, choose $c_4$ large enough such that $\frac{3}{2}c_4 \log_2(1-\epsilon) + 4c_5\beta^2 + 2c_6\beta\sqrt{\alpha} + \frac{1}{2}\beta^2 = 0$. Then $f(\ell, n) \leq c_4 n(\log_2 n)^2 + c_5\ell^2 \log_2 n + c_6\ell\sqrt{n}\log n + (-\frac{1}{2}c_4)(\log_2(1-\epsilon))^2 n + h(n)$. Since $n$ is large, we have $f(\ell, n) \leq c_4 n(\log_2 n)^2 + c_5\ell^2 \log_2 n + c_6\ell\sqrt{n}\log n$ as desired.

As a result, the bound on $|T|$ is $O(n(\log n)^2)$. It follows that the required storage and number of multiplications is $O(n(\log n)^2)$. □

Note that refering to Lemma 5.1, there is an extra $\log n$ term in the result of Theorem 5.2. This is because that in the grid model problem case, the $\ell$ term in the boundary of $|T|$ is replaced by a lower order term based on the information from the separators.

REMARK 5.3. *The above complexity results apply to Bischof and Van Loan [5] WY-representation. This would generate*

$$Q_i^T = I - W_i Y_i^T$$

TABLE 1
*Numerical operation count for the model problem.*

| K | $\|T\|/N_R$ | $\|Y\|/N_R$ | $(2\|Y\| + \|T\|)/N_R$ |
|---|---|---|---|
| 20 | 3.21 | 3.12 | 9.45 |
| 40 | 3.38 | 2.55 | 8.48 |
| 60 | 3.42 | 2.35 | 8.12 |
| 80 | 3.45 | 2.25 | 7.95 |
| 100 | 3.45 | 2.17 | 7.79 |

TABLE 2
*Numerical operation count for general problem.*

| Prob. | m | n | NZ | $\|T\|/N_R$ | $\|Y\|/N_R$ | $\frac{m}{n}$ |
|---|---|---|---|---|---|---|
| ILLC1033 | 1033 | 320 | 4732 | 2.12 | 3.40 | 3.23 |
| WELL1033 | 1033 | 320 | 4732 | 2.14 | 3.43 | 3.23 |
| ILLC1850 | 1850 | 712 | 8758 | 2.73 | 3.27 | 2.60 |
| WELL1850 | 1850 | 712 | 8758 | 2.73 | 3.27 | 2.60 |
| CONVEC8 | 3362 | 484 | 13997 | 3.30 | 7.24 | 6.95 |
| DUNES8 | 5514 | 771 | 24796 | 3.49 | 6.72 | 7.15 |
| MIMBUS | 23871 | 1325 | 181972 | 4.93 | 15.78 | 18.02 |
| STRAT8 | 16640 | 2205 | 66192 | 3.43 | 6.32 | 7.55 |

*where $Y_i$ is the same as above, but $W_i$ is computed according to*

$$W_i^{(1)} = (w_i^{(1)})$$

$$W_i^{(k)} = \left( \begin{array}{cc} W_i^{(k-1)} & z^{(k)} \end{array} \right); \; z^{(k)} = (I - W_i^{(k-1)} Y_i^{(k-1)T}) w_i^{(k)}$$

*thus $W_i = W_i^{(n_i)}$. To our knowledge, this was never stated formally , but from an easy induction argument it is evident that $W_i = Y_i T_i$. Since $T_i$ is full, it is reasonable to assume that $W_i$ will be as well. Moreover, our bounds will apply to the Bischof-Van Loan representation, but with slightly different constants. The $YTY$ representation always requires less storage. In our experiments, the $WY$ representation tended to compute $Q^T b$ somewhat faster, but that result varies among architectures [29].*

**6. Numerical Results.** In this section, we examine the performance of our method for computing $Q^T b$ and solve the linear least squares problem given in (1) by $QR$ method using our method to compute $Q^T b$.

We first check the ratio of the required number of multiplications for computing $Q^T b$ versus $N_R$ for model problem in Table 1. In Table 2 we do the same test for the general problems from Harwell-Boeing test collection and Bramley's test matrices. The results show our method also performs well for those problems. ( i.e. the required number of multiplications is less than $\frac{3m}{n} N_R = O(N_R)$, where $m$ and $n$ are the number of rows and columns of the problem respectively.) Here, we use $\frac{m}{n}$ as an approximation of the average value of $s$ in section 4.2, since $\sum_i^n s_i = m$, and $s = \max_i \; s_i$ and is presumed to be constant.

We also solve the linear least squares problems given in (1) by $QR$ method by [18] using our method for computing $Q^T b$. We compare the $QR$ method with the method of correct semi-normal equations method (CSNE) by [6]. The CSNE method

TABLE 3
*Residuals for the model problem.*

| K | QR | CSNE |
|---|---|---|
| 10 | 4.08 | 4.08 |
| 20 | 9.21 | 9.21 |
| 30 | 13.99 | 13.99 |
| 40 | 19.54 | 19.19 |
| 50 | 24.27 | 24.27 |
| 60 | 24.98 | 28.98 |
| 70 | 31.32 | 33.94 |

TABLE 4
*Timing for the model problem ( in seconds).*

| K | QR(YTY BLAS-2) | QR(WY BLAS-2) | QR(BLAS-1) | CSNE |
|---|---|---|---|---|
| 10 | 0.04 | 0.04 | 0.03 | 0.05 |
| 20 | 0.45 | 0.44 | 0.37 | 0.32 |
| 30 | 1.36 | 1.35 | 1.34 | 0.87 |
| 40 | 2.99 | 2.98 | 2.81 | 1.65 |
| 50 | 5.07 | 4.72 | 4.80 | 2.34 |
| 60 | 10.00 | 7.31 | 7.93 | 4.58 |
| 70 | 13.39 | 9.03 | 10.13 | 4.89 |
| 80 | 18.87 | 15.02 | 16.95 | 7.81 |

is in fact the method of semi-normal equations with one or more correction steps,

$$
\begin{aligned}
r &= b - Ax \\
R^T R \delta x &= A^T r \\
x &\leftarrow x + \delta x.
\end{aligned}
$$

The tests of CSNE were on QR27 routines by [24]. Two correction steps are used as suggested in QR27.

In Table 3 and 4, we list the residuals, $||r||_2 = ||b - Ax||_2$, and timing for solving (1) by both CSNE and $QR$ method on the model problem. Here, we assume the upper triangular factor $R$ has been pre-computed, the timing results do not include the factorization time. All the tests in this part were run on a Sun 4. We see from Table 3 that the residuals from both methods are about the same. However, the $QR$ method is more time- consuming than the CSNE method. This should be expected because $QR$ method requires more operations and, therefore, is slower. Also, we compare the performance of the BLAS-2 method (14) and the BLAS-1 method (6). Both WY and storage-efficient-WY (YTY) representations are applied in the BLAS-2 method. The results are listed in Table 4. The WY BLAS-2 method is faster than BLAS-1 method for larger problems. The BLAS-1 method seems faster than the YTY BLAS-2 method, but the gap between them narrows with larger problems. This suggests that the BLAS-1 method requires fewer floating point operations, but BLAS-2 method takes more advantage of features of the architecture.

In [6], Björck stated that the CSNE method does not obtain good accuracy on "stiff " problems:

$$\begin{pmatrix} \omega A_1 \\ A_2 \end{pmatrix} x = \begin{pmatrix} \omega b_1 \\ b_2 \end{pmatrix},$$

where the rows are of widely differing norms.

It has been shown that the $QR$ method performs well for "stiff" problems [1]. There is some comment on this problem in [2, 3]. In order to confirm that the $QR$ method using the proposed method maintains this property, we apply the QR and CSNE methods on a sample "stiff" problem given in Figure 11. We take the exact solution $x$ as:

$$x = (10.0, 1.0, 10^{-1}, 10^{-2}, 10^{-3})^T$$

and set the right-hand side vector $b$ to be $Ax$. We define error by:

$$error = ||\tilde{x} - x||_2.$$

$$\begin{bmatrix} \omega & \omega & \omega & \omega & \\ \omega & 2\omega & 4\omega & 8\omega & \\ 1. & 3. & 9. & 27. & \\ & 4. & & 64. & 256. \\ & 5. & & 125. & 625. \\ & 6. & & 216. & 1296. \\ & 7. & & 343. & 2401. \\ 1. & 8. & & & 4096. \\ 1. & 9. & & & 6561. \end{bmatrix}$$

FIG. 11. *A sample "stiff" problem*

The results for the $QR$ and CSNE methods using single precision with $\omega = 10^4, 10^5, 10^6$, and $10^7$ are given in Table 5.

The $QR$ method performs consistently well and gives indeed better accuracy for increasing $\omega$. These results are consistent with the work by Björck and Matstoms [6, 24].

**7. Conclusion.** In this paper, we have provided a multifrontal-based method for storing $Q$ and, thus, computing $Q^T b$ by using the frontal Householder matrices for an $m$ by $n$ large and sparse matrix with $m \geq n$. We have shown that the use of multifrontal paradigm requires $O(N_R)$ storage and multiplications for the $K$ by $K$ grid

TABLE 5
*Errors for a sample "stiff" problem.*

| $\omega$ | No. of refinements | Errors $QR$ | Errors CSNE |
|---|---|---|---|
| $10^4$ | 0 | 9.83E-5 | 1.78E+1 |
| | 1 | | 1.11E-5 |
| | 2 | | 1.07E-6 |
| $10^5$ | 0 | 3.32E-4 | 1.56E+3 |
| | 1 | | 8.78E-3 |
| | 2 | | 2.32E-2 |
| | 3 | | 9.74E-5 |
| $10^6$ | 0 | 1.84E-4 | 1.81E+5 |
| | 1 | | 1.64E+3 |
| | 2 | | 1.06E+2 |
| | 3 | | 2.45E+0 |
| | 4 | | 9.62E-2 |
| | 5 | | 9.68E-7 |
| $10^7$ | 0 | 3.45E-4 | 5.60E+7 |
| | 1 | | 1.02E+7 |
| | 2 | | 3.38E+7 |
| | 3 | | 2.84E+7 |
| | 4 | | 2.33E+7 |
| | 5 | | 1.96E+7 |

model problem and problems defined on the $\sqrt{n}$-separable graphs, where $N_R = n \log n$ represents the number of nonzeros in the upper triangular factor $R$. This method is more efficient than using the Householder matrix $H$ or $Q$ directly if the matrix is $m$ by $n$ with $m - n = O(n)$ and defined on a $\sqrt{n}$-separable graph. In that case both $H$ and $Q$ have $O(n\sqrt{n})$ nonzeros. Thus one can solve sparse linear least squares problems by the orthogonal method using the proposed method for computing $Q^T b$ efficiently.

In order to introduce BLAS-2 operations, we also use the " Storage- Efficient-WY Representation" for the orthogonal factor of each frontal matrix. This representation brings the bound on storage and operation counts up to $O(n(\log n)^2)$ for matrices defined on $\sqrt{n}$-separable graphs. This is still more efficient than using $Q$ itself or the " Storage- Efficient-WY Representation" of $Q$ directly if the matrix is $m$ by $n$ with $m - n = O(n)$ and defined on a $\sqrt{n}$-separable graph; under which $Q$ and its " Storage- Efficient-WY Representation" have $O(n\sqrt{n})$ and $O(n^2)$ nonzeros respectively.

The proposed method has possibilities for parallel computing as seen on the iPSC/2 [23]. In a future report, we will test different representations of the orthogonal factors on $\sqrt{n}$-separator matrices and various practical problems such as the geodesy problems [19] and the equilibrium systems problem [30] on advanced architectures which support BLAS-2 and BLAS-3 operations.

**Acknowledgements.** The authors would like to thank Åke Björck and Pontus Matstoms for making the QR27 software available to us and to thank John Gilbert and Esmond Ng for introducing us to this problem. The second author thanks Don Beaver for some helpful discussion.

REFERENCES

[1] J. L. BARLOW AND S. L. HANDY, *The direct solution of weighted and equality constrained least squares problems*, SIAM J. Sci. Computing 9(1988), pp. 704–716.

[2] J. L. BARLOW, *Error analysis and implementation aspects of deferred correction for equality constrained least squares problems*, SIAM J. Numer. Anal., 25(1988), pp. 1340–1358.

[3] J. L. BARLOW AND U. B. VEMULAPATI, *A note on deferred correction for equality constrained least squares problems*, SIAM J. Numer. Anal., 29(1992), pp. 249-256.

[4] P. BERMAN AND G. SCHNITGER, *On the performance of the minimum degree ordering for Gaussian elimination*, SIAM J. Matr. Anal., 11(1990), pp. 83–88.

[5] C.H. BISCHOF AND C.F. VAN LOAN, *The WY representation for products of Householder matrices*, SIAM J. Sci. Stat. Comput., 8(1987), pp.s2-s13.

[6] Å. BJÖRCK, *Stability analysis of the method of seminormal equations for linear least squares problems*, Linear Algebra Appl., 88/89(1987), pp. 31–48.

[7] J. J. DONGARRA, J. R. BUNCH, C. B. MOLER, AND G. STEWART, *LINPACK users' guide*, SIAM Press, Philadulphia.(1979).

[8] E. CHU, *Orthogonal decomposition of dense and sparse matrices on multiprocessors*, Technical Report CS-88-08, University of Waterloo, Waterloo, Ontario, March, 1988.

[9] I. S. DUFF AND J. K. REID, *The multifrontal solution of indefinite sparse symmetric linear systems*, ACM Trans. Math. Software, 9(1983), pp. 302–325.

[10] A. GEORGE, *Nested dissection of a regular finite element mesh*, SIAM J. Numerical Analysis, 10(1973), pp. 345–363.

[11] A. GEORGE AND M. T. HEATH, *Solution of sparse linear least squares problems using Givens rotations*, Linear Algebra and its Applications, 34(1980), pp. 69–83.

[12] A. GEORGE AND J. W. LIU, *Computer solution of large sparse positive definite systems*, Prentice-Hall Inc,Englewood Cliffs, NJ, (1981).

[13] ———, *Householder reflections versus Givens rotations in sparse orthogonal decomposition*, Linear Algebra and its Appl., 88/89 (1987), pp. 223–238.

[14] A. GEORGE, J. W. LIU, AND E. G. NG, *A data structure for sparse QR and LU factorizations*, SIAM J. Sci. Computing, 9(1988), pp. 100–121.

[15] A. GEORGE, J. W. LIU, *Communication results for parallel sparse Cholesky factorization on a hypercube* , Parallel Comput., 10(1989), pp. 287–298.

[16] J. R. GILBERT AND R. E. TARJAN, *The analysis of a nested dissection algorithm*, Numerische Mathematik. 50(1987), pp. 377–404.

[17] J. R. GILBERT, E. G. NG, B. W. PEYTON, *Separators and structure prediction in sparse orthogonal factorization*,(1993).

[18] G. H. GOLUB, *Numerical methods for solving linear least squares problems*, Numer.Math., 7(1965), pp. 206–216.

[19] G. H. GOLUB, P. MANNEBACK, PH. L. TOINT, *A comparison between direct and iterative methods for certain large scale geodetic least squares problems*, SIAM J. Sci.Stat.Comput. 7(1986), pp. 799–816.

[20] J. G. LEWIS, D. J. PIERCE, AND D. K. WAH, *Multifrontal Householder QR factorization*, Technical Report ECA-TR-127, Boeing Computer Services, Seattle WA, Nov. (1989).

[21] R. J. LIPTON, D. J. ROSE, AND R. E. TARJAN, *Generalized nested dissection*, SIAM J. on numerical analysis, 16(1979), pp. 346–358.

[22] J. W. LIU, *On general row merging schemes for sparse givens transformations*, SIAM J. Sci. Stat. Comput., 7(1986), pp. 1190–1211.

[23] S. M. LU AND J. L. BARLOW, *Parallel computation of orthogonal factors of sparse matrices*, Proc. Sixth SIAM Conference on Parallel processing for Scientific Computing, Norfolk, VA, (1993), pp. 486–490.

[24] P. MATSTOMS, *The multifrontal solution of sparse linear least squares problems*, Thesis No. 293, LIU-TEK-LIC-1991:33, Department of Mathematics, Linköping University (1991).

[25] P. MATSTOMS, *Parallel sparse QR factorization on Shared Memory Architectures*, LiTH-MAT-R-1993-18, Department of Mathematics, Linköping University (1993).

[26] E. G NG AND B. W. PEYTON, *A tight and explicit representation of Q in sparse QR factorization*, Technical Report ORNL/TM-12059, Oak Ridge National Laboratory, 1992.

[27] C. PUGLISI, *QR factorization of large sparse overdetermined and square matrices using the multifrontal method in a multiprocessor environment*, Ph.D. thesis, De L'institut National Polytechnique de Toulouse (1993).

[28] D. J. ROSE, R. E. TARJAN, AND G. S. LUEKER, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. on computing, 5(1976), pp. 266–283.

[29] R. SCHREIBER AND C. VAN LOAN, *A storage-efficient WY representation for products of Householder transformations*, SIAM J. Sci.Stat.Comput. 10 (1989), pp. 53–57.

[30] S. A. VAVASIS, *Stable numerical algorithms for equilibrium systems*, SIAM J. Matr. Anal. Appl., to appear 1993.