RAL-TR-1999-045

A class of incomplete orthogonal factorization methods. I: methods and theories[1]

Zhong-Zhi Bai[2], Iain S. Duff[3], and Andrew J. Wathen[4]

ABSTRACT

We study the solution of large sparse nonsingular and unsymmetric systems of linear equations. We present a class of incomplete orthogonal factorization methods based on Givens rotations. These methods include: Incomplete Givens Orthogonalization (IGO-method) and Generalized Incomplete Givens Orthogonalization (GIGO-method), which drop entries from the incomplete orthogonal and upper triangular factors by position; Threshold Incomplete Givens Orthogonalization (TIGO(τ)-method), which drops entries dynamically by their magnitudes; and Generalized Threshold Incomplete Givens Orthogonalization (GTIGO(τ, p)-method), which drops entries dynamically by both their magnitudes and positions. Theoretical analyses show that these methods can produce a nonsingular sparse incomplete upper triangular factor and either a complete orthogonal factor or a sparse nonsingular incomplete orthogonal factor for a general nonsingular matrix. Therefore, these methods can potentially generate efficient preconditioners for Krylov subspace methods for solving large sparse systems of linear equations. Moreover, the upper triangular factor is an incomplete Cholesky factorization preconditioner for the normal equations matrix from least-squares problems.

Keywords: nonsingular and nonsymmetric sparse matrices, sparse least squares, modified Gram-Schmidt orthogonalization process, Givens rotations, incomplete orthogonal factorizations.

AMS(MOS) subject classifications: 65F05, 65F50, 65H10. **CR:** G1.3.

---

[1]Current reports available by anonymous ftp from ftp.numerical.rl.ac.uk in the directory pub/reports. This report is in file badwRAL99045.ps.gz. Reports are also available through Web site www.numerical.rl.ac.uk/reports/reports.html.

[2]Supported by the National Natural Science Foundation project 19601036 and EPSRC Visiting Fellowship Grant GR/L 76617. LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, P.O.Box 2719, Beijing 100080, P.R.China. bzzlsec.cc.ac.cn.

[3]I.S.Duff@rl.ac.uk, www.cse.clrc.ac.uk/Person/I.S.Duff

[4]Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD, U.K. Andy.Wathencomlab.ox.ac.uk.

Computational Science and Engineering Department
Atlas Centre
Rutherford Appleton Laboratory
Oxon OX11 0QX

June 23, 1999.

Contents

1 Introduction

We consider the solution of the sparse unsymmetric system of linear equations

$$Ax = b, \qquad A \in I\!R^{n\times n} \quad \text{nonsingular}, \quad x, b \in I\!R^n. \tag{1.1}$$

that arises in very many areas of scientific computing.

The main drawback of direct methods for solving (1.1) is that the number of nonzero entries can strongly increase during the elimination process, and this often makes these methods too expensive in computer storage and CPU time. Compared to direct methods, the main weaknesses of iterative methods are that, besides possibly having unacceptably slow convergence rates due to the poor quality of the preconditioner, they are not robust enough to solve as wide a class of systems. Recently, however, many authors have suggested combining the advantages of direct methods and iterative methods by using Krylov subspace iterations combined with incomplete factorization preconditioners. This class of methods has been surprisingly successful for many cases of general unsymmetric matrices; see for example, Saad (1996) and Wang, Gallivan and Bramley (1997). Here, the accuracy and stability of the incomplete factorizations are key factors for guaranteeing the success of the preconditioned Krylov subspace iterations.

Two well known classes of basic incomplete factorization preconditioners are incomplete triangular factorization preconditioners and incomplete orthogonal factorization preconditioners.

The former, commonly known as incomplete LU (or ILU) factorizations, compute a sparse lower triangular matrix L and a sparse upper triangular matrix U by the usual Gaussian elimination process coupled with some dropping rules so that the error matrix $E = LU - A$ satisfies certain constraints, such as having zero entries in some positions. One important special example is incomplete Cholesky factorization. This class of incomplete factorization techniques was originally developed for M-matrices by Meijerink and van der Vorst (1977) and then was extended to H-matrices and block H-matrices, for which theoretical properties such as existence, stability and accuracy can be established. For details one can refer to Axelsson (1994, 1985), Concus, Golub and Meurant (1985), Donato and Chan (1992), Elman(1986, 1989), Gustafsson (1978), Manteuffel (1980), Saad (1996), and references therein. For general unsymmetric matrices, although a number of efficient incomplete LU factorization techniques have been presented (see Axelsson (1994) and Saad (1996)), it is more difficult to give theoretical assurances about the feasibility and efficiency of these incomplete triangular factorization preconditioners. There can be breakdowns in the factorization process due to zero pivots, inaccuracies of the incomplete triangular factors due to small pivots and inefficient dropping rules, as well as instability of the triangular solves due to the poorly conditioned incomplete triangular factors.

The latter, normally known as incomplete QR factorizations, compute a sparse and generally non-orthogonal matrix Q and a sparse upper triangular matrix R by the modified Gram-Schmidt process incorporating some dropping rules. This class of incomplete factorization techniques was initially developed for general sparse matrices by Saad (1988). Thereafter, much attention was paid to practical applications rather than theoretical properties since its theoretical properties such as existence, stability and accuracy are currently too complicated for analysis. For details, one can refer to Saad (1988,1996). Recently, for a special strategy that only drops entries of the upper triangular matrix R, Wang et al. (1997) have proved the existence and stability of the associated incomplete QR factorization preconditioner. For particular sparsity patterns, this special strategy produces an R factor identical to that produced by the incomplete Cholesky method applied to the normal equations.

In addition to the drawbacks of breakdowns, inaccuracies and instabilities as in the incomplete LU factorization methods, one major problem about the above mentioned incomplete QR factorization methods is that the matrix Q is not in general orthogonal, and nothing guarantees that it is even nonsingular unless we adopt a strategy that does not drop many entries. However, this makes the resulting incomplete factors Q and R likely to be too dense to be useful in practice (see Saad, 1988).

In fact, the main motivation for developing incomplete orthogonal factorization preconditioners derives from the power of the *complete* orthogonal factorization process. Some advantages of complete QR factorizations over complete LU factorizations are:

(a) they will never break down and will always produce an orthogonal matrix Q and an upper triangular matrix R such that $A = QR$;

(b) orthogonal factorization is strongly robust and numerically stable; and

(c) the orthogonality of the matrix Q makes the solution of the original system of linear equations (1.1) easily obtainable through solving the upper triangular linear system with the triangular factor R.

Another useful feature is that the triangular factor R *is* the Cholesky factor of the normal equations matrix. Here, the orthogonality of the factor Q is the key point for ensuring the success of these orthogonal factorization methods. For a good incomplete QR factorization preconditioner, we naturally expect that it will roughly inherit the above three advantages of the complete QR factorization, or at least, it should possess some of the following properties:

(I) Q is an orthogonal matrix and R is a sparse upper triangular matrix such that the error matrix $E = QR - A$ is 'small';

(II) Q is a sparse nonsingular matrix and R is a sparse upper triangular matrix such that the error matrices $E = QR - A$ and $E_0 = Q^T Q - I$ are small, where I is the identity matrix; and

(III) if the original matrix A is nonsingular, then the matrix R must also be nonsingular.

Throughout this paper, the meaning of 'small' can be understood in the sense that either the entries dropped during the factorization process are small enough, or the error matrices satisfy certain constraints such as having zero entries in some positions. We will show, in the analyses in the next section, that incomplete QR factorization preconditioners based on the modified Gram-Schmidt orthogonalization process do not satisfy either property (I) or (II). However, we will show that incomplete QR factorization preconditioners based on Givens rotations, which we will construct and study in this paper, do satisfy property (III), and at least one of the properties (I) and (II). That such incomplete Givens strategies can always compute an orthogonal factor Q (orthogonal to the limits of finite precision arithmetic) is a particular feature: one consequence is that the R factor is always an incomplete Cholesky factor of the normal equations. For this situation, Q is not generally required and therefore need not be stored.

The rest of this paper is organized as follows: After reviewing the Incomplete Modified Gram-Schmidt methods and their properties in Section 2, we describe the Incomplete Givens Orthogonalization method (IGO-method) and the Generalized Incomplete Givens Orthogonalization method (GIGO-method) and analyse their theoretical properties in Section 3. The Threshold Incomplete Givens Orthogonalization method (TIGO(τ)-method) and the Generalized Threshold Incomplete Givens Orthogonalization method (GTIGO(τ, p)-method) are described in Section 4. Finally, some conclusions and remarks are made in Section 5.

Throughout this paper we use the term 'incomplete orthogonal factor' for the factor Q of an incomplete orthogonalization method even though such incomplete factors are not necessarily orthogonal matrices.

2 The incomplete modified Gram-Schmidt methods

The Classical Gram-Schmidt (CGS) method is one of the oldest methods for computing a QR factorization

$$A = QR$$

of a given matrix $A = (a_{ij})_{n\times n} = (a_1, a_2, \ldots, a_n)$, where $Q = (q_{ij})_{n\times n} = (q_1, q_2, \ldots, q_n)$ is an orthogonal matrix and $R = (r_{ij})_{n\times n}$ is an upper triangular matrix. This method does not break down if and only if the matrix A is of full

rank and, in this case, the QR factorization is well defined. A numerically stable alternative of the standard Gram-Schmidt process is known as the Modified Gram-Schmidt (MGS) method.

Method 2.1: MGS-method
1. Define $r_{11} := \|a_1\|_2$. If $r_{11} = 0$ Stop, else $q_1 := a_1/r_{11}$
2. For $j = 2, \ldots, n$ Do:
3. Define $\hat{q} := a_j$
4. For $i = 1, \ldots, j-1$ Do:
5. Compute $r_{ij} := (\hat{q}, q_i)$
6. Compute $\hat{q} := \hat{q} - r_{ij} q_i$
7. EndDo
8. Compute $r_{jj} := \|\hat{q}\|_2$
9. If $r_{jj} = 0$ then Stop, else $q_j := \hat{q}/r_{jj}$
10. EndDo

The MGS-method is quite efficient for solving the sparse unsymmetric system of linear equations (1.1) because:

(a) it is numerically stable and the inverse of the matrix Q is given explicitly by Q^T; and

(b) it can be simply implemented in a similar way to left-looking LU factorization where, at each step, a given column is combined with previous columns and then normalized.

To define the corresponding Incomplete Modified Gram-Schmidt (IMGS) method, dropping strategies or nonzero patterns for the incomplete factorization matrices Q and R must be defined. This can be done in a very general way as follows. Introduce two sets of integer pairs

$$P_n = \{(i,j) \mid 1 \leq i,j \leq n\} \quad \text{and} \quad P_U = \{(i,j) \mid i \leq j, \quad 1 \leq i,j \leq n\},$$

and let P_Q and P_R be the chosen nonzero patterns for the matrices Q and R, respectively. The only restriction on P_R is that $P_R \subseteq P_U$. As for P_Q, we require that $P_Q \subseteq P_n$ and for each row there must be at least one nonzero entry. The two sets P_Q and P_R can be selected in similar ways to those defined for ILU factorizations. For details one can refer to Saad (1988).

Method 2.2: IMGS-method
1. For $j = 1, \ldots, n$ Do:
2. Set $q_j := a_j$
3. For $i = 1, \ldots, j-1$ Do:
4. If $(i,j) \in P_R$, compute $r_{ij} := (q_i, q_j)$,

5. Else Set $r_{ij} := 0$
6. Compute $q_j := q_j - r_{ij} q_i$
7. EndDo
8. For $i = 1, \ldots, n$ and $(i,j) \notin P_Q$ Do:
9. Set $q_{ij} := 0$
10. EndDo
11. Compute $r_{jj} := \|q_j\|_2$
12. If $r_{jj} = 0$ then Stop, else $q_j := q_j / r_{jj}$
13. EndDo

Simple algebraic manipulation shows that this IMGS-method produces a sparse matrix Q and a sparse upper triangular matrix R satisfying $A = QR + E$, where E is the error matrix whose j-th column ϵ_j is the column of entries that were dropped from column q_j in lines 8-10. Typically, the error matrix E is small because of the strategy adopted in dropping entries. One major problem with the above decomposition is that the matrix Q is not usually orthogonal. In fact, nothing guarantees that it is even nonsingular unless P_Q is of large cardinality. For the purposes of this discussion, we will assume that line 6 of Method 2.2 is replaced by an ideal orthogonalization process, for example Daniel, Gragg, Kaufman and Stewart (1976), in which q_j is forced to be orthogonal to all $q_1, q_2, \ldots, q_{j-1}$.

Theorem 2.1 (see Saad, 1988) Assume that at every step $j \leq i-1$ of the IMGS-method, column q_j is orthogonal to $q_1, \ldots, q_{j-1}$ and let $\prod_{i-1}$ be the orthogonal projector onto the span of the columns $q_1, \ldots, q_{i-1}$. Then the matrix $Q_i = (q_1, q_2, \ldots, q_i)$ is of full rank if and only if $\| \prod_{i-1} \epsilon_i \|_2 < r_{ii}$. *A fortiori*, Q_i is of full rank if $\|\epsilon_i\|_2 < r_{ii}$.

Although this theorem provides conditions under which the constructed matrix Q_i remains of full rank at every step, it is not likely to be useful in practice because, at every step, the vector q_j must be made orthogonal to the previous vectors $q_1, \ldots, q_{j-1}$. In addition, even if we were to obtain a reasonably simple criterion for ensuring the nonsingularity of the matrix Q, this criterion might be so severe that the only way in which it could be satisfied is by making ϵ_i very small, which means allowing more fill-in in the matrices Q and R.

The case where the entries in Q are not dropped, that is the case when $P_Q = P_n$, is of particular interest. Indeed, in this situation, the error matrix $E = 0$ and we have the exact relation $A = QR$. Of course, Q is still not orthogonal and it may be dense. However, in this case, the nonsingularity of the matrices Q and R can be easily guaranteed.

Theorem 2.2 (see Wang et al., 1997) Let $A \in I\!R^{n \times n}$ be nonsingular and $P_Q = P_n$. Then the IMGS-method computes an incomplete QR factorization $A = QR$, in which Q is nonsingular and R is upper triangular with positive diagonal entries.

Wang et al. (1997) claim that an attraction of this special IMGS-method is that it can efficiently produce an incomplete Cholesky factorization preconditioner for the normal equations of linear least-squares problems.

3 Incomplete Givens orthogonalization methods

Another way to compute the QR factorization is to use Givens rotations. A Givens rotation (or plane rotation) $G(i,j,\theta) \in I\!R^{n\times n}$ is equal to the identity matrix except that

$$G([i,j],[i,j]) = \begin{pmatrix} c & s \\ -s & c \end{pmatrix},$$

where $c = \cos\theta$ and $s = \sin\theta$. The operation $y = G(i,j,\theta)x$ rotates x through θ radians clockwise in the (i,j) plane. Algebraically,

$$y_k = \begin{cases} x_k, & \text{for } k \neq i,j, \\ cx_i + sx_j, & \text{for } k = i, \\ -sx_i + cx_j, & \text{for } k = j, \end{cases} \qquad 1 \leq k \leq n,$$

and so, $y_j = 0$ if

$$s = \frac{x_j}{\sqrt{x_i^2 + x_j^2}}, \qquad c = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}.$$

Givens rotations are therefore useful for introducing zeros into a vector one at a time. Note that there is no need to compute the angle θ, since c and s in the above are all that are needed to apply the rotation.

To define an incomplete QR factorization of the matrix $A = (a_{ij})_{n\times n}$ based on Givens rotations, in addition to the nonzero patterns P_Q, P_U and P_R described in Section 2 for the incomplete factorization matrices Q and R, we need to introduce the following sets of integer pairs:

$$\begin{aligned} P_{A,L} &= \{(i,j) \mid a_{ij} \neq 0, \quad i \geq j, \quad 1 \leq i,j \leq n\}, \\ P_{A,U} &= \{(i,j) \mid a_{ij} \neq 0, \quad i \leq j, \quad 1 \leq i,j \leq n\}, \\ P_A &= \{(i,j) \mid a_{ij} \neq 0, \quad 1 \leq i,j \leq n\}, \text{ and} \\ P_L &= \{(i,j) \mid i \geq j, \quad 1 \leq i,j \leq n\}. \end{aligned} \tag{3.1}$$

That is $P_{A,L}$ and $P_{A,U}$ are the nonzero patterns of the lower and upper triangular parts of the matrix A, respectively, P_A is the nonzero pattern of the matrix A, and P_L and P_U (defined in Section 3) are the nonzero patterns of any lower and upper triangular matrices in $I\!R^{n\times n}$, respectively. Now, for given sets of integer pairs P_l and

P_u satisfying $P_{A,L} \subseteq P_l \subseteq P_L$ and $P_{A,U} \subseteq P_u \subseteq P_U$, we can define the Incomplete Givens Orthogonalization (IGO) method.

3.1 IGO-method for nonsingular matrices

The IGO-method consists of the following three elementary processes:

For each column in turn:

(a) Annihilate, using Givens rotations, the nonzero entries located in the strictly lower triangular part of the matrix $A \in I\!R^{n\times n}$ from the bottom up to the first sub-diagonal;

(b) Update the incomplete orthogonal matrix $Q \in I\!R^{n\times n}$ by postmultiplying by the transpose of the Givens rotation using some dropping rule;

(c) After steps (a) and (b) have been done for all nonzeros in the current column, form the corresponding row of the incomplete upper triangular matrix $R \in I\!R^{n\times n}$ using some dropping rule.

More precisely, this method can be described as follows:

Method 3.1: IGO-method
1. Set $Q = I$
2. For $j = 1, \ldots, n-1$ Do:
3. Define $k_r(j) := \max\{i \mid i \geq j, a_{ij} \neq 0\}$
4. If $k_r(j) = j$ then cycle
5. For $i = k_r(j)$ DownTo $j+1$ and $a_{ij} \neq 0$ Do: *% All nonzero subdiagonals in column j are annihilated*
6. Compute $\varrho := \sqrt{a_{jj}^2 + a_{ij}^2}$
7. Compute $c := a_{jj}/\varrho$
8. Compute $s := a_{ij}/\varrho$
9. Set $a_{jj} := \varrho$
10. For $k = j+1, \ldots, n$ and a_{ik} and $a_{jk} \neq 0$ Do: *% Update matrix*
11. Compute $temp := -sa_{jk} + ca_{ik}$
12. Compute $a_{jk} := ca_{jk} + sa_{ik}$
13. Set $a_{ik} := temp$
14. EndDo
15. For $k = 1, \ldots, n$ Do: *% Update Q and respect sparsity pattern for Q*
16. Compute $temp := -sq_{kj} + cq_{ki}$ if $(k,i) \in P_Q$
17. Compute $q_{kj} := cq_{kj} + sq_{ki}$ if $(k,j) \in P_Q$
18. Set $q_{ki} := temp$ if $(k,i) \in P_Q$
19. EndDo

20. EndDo
21. For $k = j, \ldots, n$ and $(j,k) \in P_R$ Do: *% Respect sparsity pattern for R*
22. Set $r_{jk} := a_{jk}$
23. EndDo
24. EndDo
25. Set $r_{nn} := a_{nn}$

In the actual computation, there is no need to store the matrix $R = (r_{ij})$ separately. The matrix $A = (a_{ij})$ is updated successively and, at the end of the algorithm, its upper triangular part gives the matrix R.

To analyse the numerical properties of the IGO-method, we denote the Givens rotation defined in lines 6-8 by $G(i,j)$, and define matrices

$$\begin{cases} G_j = G(j+1,j)\cdots G(k_r(j),j) \equiv \prod_{i=j+1}^{k_r(j)} G(i,j), \\ \\ R_j = G_j R_{j-1} + E_j^{(R)}, \quad R_0 = A, \end{cases} \tag{3.2}$$

where $E_j^{(R)}$ is the error matrix determined by lines 21-23. Then, it is clear that $R = R_{n-1}$, and for $j = 1, 2, \ldots, n-1$, G_j are orthogonal matrices, $E_j^{(R)}$ are strictly upper triangular matrices with their bottom-right $(n-j)\times(n-j)$ blocks zero, and R_j are upper triangular matrices except for their bottom-right $(n-j)\times(n-j)$ blocks which are nonsingular submatrices in the case that A is nonsingular. In addition, if we denote the matrix determined by lines 15-19, by $Q(i,j)$, the corresponding error matrix by $E^{(Q)}(i,j)$, and let $Q_0 = I$, then we easily see that $Q = Q_{n-1}$ and it can be computed by the following procedure:

Procedure for Generating the Incomplete Orthogonal Matrix
1. For $j = 1, \ldots, n-1$ Do:
2. Set $Q(k_r(j)+1, j) := Q_{j-1}$
3. For $i = k_r(j) - 1$ DownTo $j+1$ Do:
4. Compute $\tilde{Q}(i,j) := Q(i+1,j)G(i,j)^T$
5. Set $Q(i,j) := \tilde{Q}(i,j) + E^{(Q)}(i,j)$
6. EndDo
7. Set $Q_j := Q(j+1,j)$
8. EndDo

We assume that the original matrix $A \in I\!R^{n\times n}$ is nonsingular and let

$$G = G_{n-1}G_{n-2}\cdots G_1 \equiv \prod_{j=n-1}^{1} G_j. \tag{3.3}$$

Then, from the construction of the IGO-method and the structure of the matrices R_j and $E_j^{(R)}$ in (3.2), we immediately know that R_j is a nonsingular matrix and its first j diagonal entries are positive. Therefore, the incomplete upper triangular matrix R must be nonsingular. Moreover, if all $E^{(Q)}(i,j) = 0$, that is, $P_Q = P_n$ or the entries in the matrix Q are not dropped, then $Q = G^T$ is an orthogonal matrix, and therefore, is nonsingular. We state this property in the following.

Theorem 3.1 Let $A \in I\!R^{n\times n}$ be nonsingular, and $Q, R \in I\!R^{n\times n}$ be the incomplete orthogonal and upper triangular matrices, respectively, produced by the IGO-method. Then

(i) R is sparse and nonsingular, and its diagonal entries are positive except except possibly for the last one;

(ii) $Q = G^T$ is orthogonal, provided $P_Q = P_n$.

This follows directly from (3.2) and (3.3).

Theorem 3.2 Let the conditions of Theorem 3.1 be satisfied. Then $Q, R \in I\!R^{n\times n}$ are sparse matrices and

(i) $A = G^T R - E^{(R)}$ and R is nonsingular, where $E^{(R)} = \sum_{j=1}^{n-1} (\prod_{i=1}^{j} G_j)^T E_j^{(R)}$;

(ii) $Q = G^T + E^{(Q)}$, where $E^{(Q)} = \sum_{j=1}^{n-1} E_j^{(Q)} (\prod_{i=j}^{n-1} G_i)^T$,

$E_j^{(Q)} = \sum_{i=j+1}^{k_r(j)} E^{(Q)}(i,j) \prod_{k=i}^{k_r(j)} G(k,j)$;

(iii) Q is nonsingular if $\|E^{(Q)}\| < 1$. Furthermore, $\sum_{j=1}^{n-1} \sum_{i=j+1}^{k_r(j)} \|E^{(Q)}(i,j)\| < 1$ implies $\|E^{(Q)}\| < 1$;

(iv) $A = QR - E$, where $E = E^{(R)} + E^{(Q)} R$.

Proof: We first verify (i). According to Theorem 3.1, $R \in I\!R^{n\times n}$ is nonsingular. By using (3.2) recursively, we have

$$\begin{aligned} R_j &= G_j R_{j-1} + E_j^{(R)} \\ &= G_j (G_{j-1} R_{j-2} + E_{j-1}^{(R)}) + E_j^{(R)} \\ &= G_j G_{j-1} R_{j-2} + G_j E_{j-1}^{(R)} + E_j^{(R)} \\ &= \ldots \\ &= G_j G_{j-1} \cdots G_1 R_0 + G_j G_{j-1} \cdots G_2 E_1^{(R)} + \ldots + G_j E_{j-1}^{(R)} + E_j^{(R)}. \end{aligned}$$

Letting $j = n-1$ we get

$$R = GA + G_{n-1}G_{n-2}\cdots G_2 E_1^{(R)} + \ldots + G_{n-1}E_{n-2}^{(R)} + E_{n-1}^{(R)}. \tag{3.4}$$

Because of the orthogonality of the matrices G_j, $j = 1, 2, \ldots, n-1$, (3.3) implies that

$$\begin{aligned} A &= G^T R - G_1^T E_1^{(R)} - \ldots - (G_{n-2}G_{n-3}\cdots G_1)^T E_{n-2}^{(R)} - (G_{n-1}G_{n-2}\cdots G_1)^T E_{n-1}^{(R)} \\ &= G^T R - E^{(R)}. \end{aligned}$$

To verify (ii), from the *Procedure for Generating the Incomplete Orthogonal Matrix* we have

$$\begin{aligned} Q(i,j) =& Q(i+1,j)G(i,j)^T + E^{(Q)}(i,j) \\ =& [Q(i+2,j)G(i+1,j)^T + E^{(Q)}(i+1,j)]G(i,j)^T + E^{(Q)}(i,j) \\ =& Q(i+2,j)[G(i,j)G(i+1,j)]^T + E^{(Q)}(i+1,j)G(i,j)^T + E^{(Q)}(i,j) \\ =& \ldots \\ =& Q(k_r(j)+1,j)[G(i,j)G(i+1,j)\cdots G(k_r(j),j)]^T \\ & + E^{(Q)}(k_r(j),j)[G(i,j)G(i+1,j)\cdots G(k_r(j)-1,j)]^T \\ & + \ldots \\ & + E^{(Q)}(i+1,j)G(i,j)^T + E^{(Q)}(i,j). \end{aligned}$$

Letting $i = j+1$ we get

$$\begin{aligned} Q_j &= Q_{j-1}G_j^T + \sum_{i=j+1}^{k_r(j)} E^{(Q)}(i,j)\Big(\prod_{k=j+1}^{i-1} G(k,j)\Big)^T \\ &= Q_{j-1}G_j^T + \sum_{i=j+1}^{k_r(j)} E^{(Q)}(i,j)\prod_{k=i}^{k_r(j)} G(k,j)G_j^T \\ &= (Q_{j-1} + E_j^{(Q)})G_j^T, \end{aligned}$$

where we have stipulated that $\prod_{k=k_1}^{k_2} G(k,j) = I$ if $k_1 > k_2$. That is to say,

$$Q_{j-1} = Q_j G_j - E_j^{(Q)},$$

and therefore

$$
\begin{aligned}
Q_{j-1} =& Q_j G_j - E_j^{(Q)} \\
=& (Q_{j+1} G_{j+1} - E_{j+1}^{(Q)}) G_j - E_j^{(Q)} \\
=& Q_{j+1} G_{j+1} G_j - E_{j+1}^{(Q)} G_j - E_j^{(Q)} \\
=& \ldots \\
=& Q_{n-1} G_{n-1} G_{n-2} \cdots G_j \\
& - E_{n-1}^{(Q)} G_{n-2} G_{n-3} \cdots G_j \\
& - \ldots \\
& - E_{j+1}^{(Q)} G_j - E_j^{(Q)}.
\end{aligned}
$$

Taking $j = 1$ we have

$$
\begin{aligned}
I &= QG - \sum_{j=1}^{n-1} E_j^{(Q)} G_{j-1} G_{j-2} \cdots G_1 \\
&= QG - \sum_{j=1}^{n-1} E_j^{(Q)} (G_{n-1} G_{n-2} \cdots G_j)^T G \\
&= (Q - E^{(Q)}) G.
\end{aligned}
$$

This equality is equivalent to

$$
Q = G^T + E^{(Q)}.
$$

Finally, (iii) follows directly from (ii) since G is orthogonal; and (iv) follows directly from (i) and (ii).

From Theorems 3.1 and 3.2 we see that the matrices Q and R satisfy the property (III), and at least one of the properties (I) and (II) stated in Section 1. In actual applications, the matrix $B = QR$, given by the IGO-method, is used as a preconditioner for Krylov subspace iterations. Moreover, it follows from Theorem 3.2(i) that if $P_Q = P_n$ the IGO-method produces an incomplete Cholesky factorization preconditioner for the normal equations of the linear least-squares problem.

3.2 Generalized IGO-method

The generalized IGO-method, designated as the GIGO-method, consists of the following four elementary processes:

(a) Choose sparse nonzero patterns P_Q and P_R of the incomplete orthogonal and upper triangular matrices Q and R, respectively, and determine sparse nonzero patterns P_l and P_u for which the entries of the matrix $A \in \mathbb{R}^{n \times n}$ need to

be annihilated and updated, respectively, during the incomplete orthogonal factorization process;

For each column in turn:

(b) Annihilate all the nonzero entries of the matrix $A \in \mathbb{R}^{n\times n}$ in P_l from the bottom up to the first sub-diagonal by Givens rotations, and update the entries of the matrix $A \in \mathbb{R}^{n\times n}$ in P_l or P_u, correspondingly;

(c) Update the incomplete orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ by postmultiplying by the transpose of the Givens rotation, with the entries not in P_Q being dropped;

(d) After steps (b) and (c) have been done for all nonzeros in the current column, form the corresponding row of the incomplete upper triangular matrix $R \in \mathbb{R}^{n\times n}$ using some dropping rule.

More precisely, this method can be described as follows:

Method 3.2: GIGO-method
1. Set $Q = I$
2. For $j = 1, \ldots, n-1$ Do:
3. Define $k_r(j) := \max\{i \mid i \geq j, (i,j) \in P_l\}$
4. If $k_r(j) = j$ then cycle
5. For $i = k_r(j)$ DownTo $j+1$, $a_{ij} \neq 0$ and $(i,j) \in P_l$ Do:
6. Compute $\varrho := \sqrt{a_{jj}^2 + a_{ij}^2}$
7. Compute $c := a_{jj}/\varrho$
8. Compute $s := a_{ij}/\varrho$
9. Set $a_{jj} := \varrho$
10. Define $k_c(i) := \max\{k \mid k \geq j+1, (i,k) \in P_l \cup P_u\}$
11. For $k = j, \ldots, k_c(i)$ and $(i,k) \in P_l \cup P_u$ Do:
12. Compute $temp_k := -sa_{jk} + ca_{ik}$
13. EndDo
14. Define $k_c'(j) := \max\{k \mid k \geq j, (j,k) \in P_u\}$
15. For $k = j+1, \ldots, k_c'(j)$ and $(j,k) \in P_u$ Do:
16. Compute $a_{jk} := ca_{jk} + sa_{ik}$
17. EndDo
18. For $k = j, \ldots, k_c(i)$ and $(i,k) \in P_l \cup P_u$ Do:
19. Set $a_{ik} := temp_k$
20. EndDo
21. For $k = 1, \ldots, n$ Do:
22. Compute $temp_k := -sq_{kj} + cq_{ki}$ if $(k,i) \in P_Q$
23. Compute $q_{kj} := cq_{kj} + sq_{ki}$ if $(k,j) \in P_Q$
24. Set $q_{ki} := temp_k$ if $(k,i) \in P_Q$

25. EndDo
26. EndDo
27. For $k = j, \ldots, n$ and $(j,k) \in P_R$ Do:
28. Set $r_{jk} := a_{jk}$
29. EndDo
30. EndDo
31. Set $r_{nn} := a_{nn}$

Similarly to the IGO-method, in the actual computation in the GIGO-method there is no need to store the matrix $R = (r_{ij})$ separately. The matrix $A = (a_{ij})$ is updated successively, and finally, its upper triangular part gives the matrix R, which may be used as an incomplete Cholesky factorization preconditioner for the normal equations of the linear least-squares problem. More generally, the matrix $B = QR$ given by the GIGO-method can be used as a preconditioner for Krylov subspace iterations. Evidently, when $P_l = P_L$ and $P_u = P_U$, the GIGO-method naturally reduces to the IGO-method.

Denote the Givens rotation determined by lines 6-8 by $G(i,j)$; the incomplete orthogonal matrix determined by lines 21-25 by $Q(i,j)$ with the corresponding error matrix $E^{(Q)}(i,j)$; and the error matrix determined by lines 10-20 for the i-loop, together with lines 27-29, by $E_j^{(R)}$. If we further define

$$G_j = \prod_{\substack{i = j+1 \\ (i,j) \in P_l}}^{k_r(j)} G(i,j) \quad \text{and} \quad G = \prod_{j=1}^{n-1} G_j,$$

then, following a similar analysis as for the IGO-method, we find that the GIGO-method has the following properties.

Theorem 3.3 Let $A \in I\!R^{n \times n}$, and $Q, R \in I\!R^{n \times n}$ be the sparse incomplete orthogonal, upper triangular matrices, respectively, produced by the GIGO-method. Then

(i) $A = G^T R - E^{(R)}$, where $E^{(R)} = \sum_{j=1}^{n-1} (\prod_{i=1}^{j} G_j)^T E_j^{(R)}$;

(ii) $Q = G^T + E^{(Q)}$, where

$$E^{(Q)} = \sum_{j=1}^{n-1} E_j^{(Q)} (\prod_{i=j}^{n-1} G_i)^T, \; E_j^{(Q)} = \sum_{\substack{i = j+1 \\ (i,j) \in P_l}}^{k_r(j)} E^{(Q)}(i,j) \prod_{\substack{k = i \\ (k,j) \in P_l}}^{k_r(j)} G(k,j);$$

(iii) $A = QR - E$, where $E = E^{(R)} + E^{(Q)} R$.

Theorem 3.3 shows that even if the matrix $A \in I\!R^{n \times n}$ is nonsingular, nothing guarantees that the incomplete orthogonal matrix Q generated by the GIGO-method is nonsingular unless we make the dropping strategy drop only a few

entries. However, from the construction of the GIGO-method, we easily see that the nonsingularity of the incomplete upper triangular matrix R is guaranteed if, for all j, $j = 1, 2, \ldots, n-1$, the integer sets $\{(i,j) \in P_l \mid j \leq i \leq k_r(j), a_{ij} \neq 0\}$ determined during the annihilating process are nonempty.

In the actual computation, the nonzero patterns P_l and P_u in the GIGO-method can be determined according to the nonzero structure of the matrix $A \in I\!R^{n\times n}$. One practical way is to simply take $P_l = P_{A,L}$ and $P_u = P_{A,U}$, another is suggested by the following procedure. Note that this procedure also determines the quantities $k_r(j)$, $k_c(j)$ and $k'_c(j)$ in the GIGO-method.

Procedure for Generating the Nonzero Patterns P_l and P_u

1. Compute $k_r(1) := \max\{i \mid (i,1) \in P_{A,L}\}$
2. Set $k_r := k_r(1)$
3. Compute $k_c(1) := \max\{j \mid (1,j) \in P_{A,U}\}$
4. Set $k_c := k_c(1)$
5. Set $j := 1$
6. Do while $j < n$
7. Set $P_l := \{(i,j) \mid j \leq i \leq k_r\}$
8. Set $P_u := \{(j,i) \mid j \leq i \leq k_c\}$
9. Compute $k_r(j+1) := \max\{k \mid (k, j+1) \in P_{A,L}\}$
10. If $k_r(j+1) < k_r(j)$ then Set $k_r := k_r(j)$, Else
11. Compute $k_c(j+1) := \max\{k \mid (j+1, k) \in P_{A,U}\}$
12. If $k_c(j+1) < k_c(j)$ then Set $k_c := k_c(j)$, Else
13. Set $j := j+1$
14. Endwhile
15. For $j = 1, \ldots, n$ Do
16. Set $k'_c(j) := k_c(j)$
17. EndDo

Note that, unlike the IMGS-method, the computation of the incomplete orthogonal matrix Q and the incomplete upper triangular matrix R of the IGO-method and the GIGO-method are independent. Therefore, if we only need an incomplete Cholesky factorization preconditioner $R \in I\!R^{n\times n}$ for the normal equations of the linear least-squares problem, there is no need to compute and store the incomplete orthogonal matrix $Q \in I\!R^{n\times n}$. This not only significantly reduces the operation and storage requirements, but also greatly simplifies the programming of both the IGO-method and the GIGO-method.

4 Threshold incomplete Givens orthogonalization methods

The incomplete Givens orthogonalization methods discussed in the previous sections are blind to numerical values because entries are dropped only using structural considerations. An alternative is to drop entries in the Givens orthogonalization process according to their magnitudes rather than their positions, as in the threshold ILU methods (Saad 1994, Saad 1996). In this approach, the nonzero patterns, for example, P_l, P_u, P_Q and P_R, are determined dynamically. This results in a Threshold Incomplete Givens Orthogonalization method (TIGO(τ)-method), based upon the GIGO-method in Section 3.2, which we show below.

Method 5.1: TIGO(τ)-method
1. Input the dropping tolerance $\tau^{(Q)}$ for the factor Q
2. Input the dropping tolerance $\tau^{(R)}$ for the factor R
3. Set $Q = I$
4. For $j = 1, \ldots, n-1$ Do:
5. Set $P_j := \{i \mid (i,j) \notin P_{A,L}, |a_{ij}| > \tau^{(R)}, j+1 \leq i \leq n\}$
6. Define $k_r(j) := \max\{i \mid (i,j) \in P_{A,L} \cup P_j\}$
7. If $k_r(j) = j$ then cycle
8. For $i = k_r(j)$ DownTo $j+1$ and $(i,j) \in P_{A,L} \cup P_j$ Do:
9. Compute $\varrho := \sqrt{a_{jj}^2 + a_{ij}^2}$
10. Compute $c := a_{jj}/\varrho$
11. Compute $s := a_{ij}/\varrho$
12. Set $a_{jj} := \varrho$
13. Set $\tilde{P}_i := \{k \mid (i,k) \notin P_A, |a_{ik}| > \tau^{(R)}, j+1 \leq k \leq n\}$
14. Define $k_c(i) := \max\{k \mid (i,k) \in P_A \cup \tilde{P}_i\}$
15. For $k = j+1, \ldots, k_c(i)$ and $(i,k) \in P_A \cup \tilde{P}_i$ Do:
16. Compute $temp_k := -sa_{jk} + ca_{ik}$
17. EndDo
18. Set $\tilde{P}_j := \{k \mid (j,k) \notin P_{A,U}, |a_{ik}| > \tau^{(R)}, j \leq k \leq n\}$
19. Define $k_c(j) := \max\{k \mid (j,k) \in P_{A,U} \cup \tilde{P}_j\}$
20. For $k = j, \ldots, k_c(j)$ and $(j,k) \in P_{A,U} \cup \tilde{P}_j$ Do:
21. Compute $a_{jk} := ca_{jk} + sa_{ik}$
22. EndDo
23. For $k = j+1, \ldots, k_c(i)$ and $(i,k) \in P_A \cup \tilde{P}_i$ Do:
24. Set $a_{ik} := temp_k$
25. EndDo
26. For $k = 1, \ldots, n$ Do:
27. Compute $temp := -sq_{kj} + cq_{ki}$
28. Compute $q_{kj} := cq_{kj} + sq_{ki}$

29. If $|temp| \leq \tau^{(Q)}$ then Set $q_{ki} := 0$, Else
30. Set $q_{ki} := temp$
31. If $|q_{kj}| \leq \tau^{(Q)}$ then Set $q_{kj} := 0$
32. EndDo
33. EndDo
34. For $k = j, \ldots, n$ and $|a_{jk}| > \tau^{(R)}$ Do
35. Set $r_{jk} := a_{jk}$
36. EndDo
37. EndDo
38. Set $r_{nn} := a_{nn}$

We remark that, in the above method, the drop tolerances $\tau^{(Q)}$ and $\tau^{(R)}$ may vary according to the row or the column. For example, relative drop tolerances $\tau^{(R)}$ may be obtained by multiplying the initial drop tolerance $\tau^{(R)}$ by the current values of $|a_{jj}|$ in line 5, line 18, and lines 34-36, respectively, and by the value of $|a_{ii}|$ in line 13. The relative drop tolerances $\tau^{(Q)}$ may be obtained by multiplying the initial drop tolerance $\tau^{(Q)}$ by $|q_{ii}|$ in line 29 and by $|q_{jj}|$ in line 31.

To further control the memory use, we need to limit the number of nonzero entries in each row (or column) of the incomplete orthogonal and upper triangular matrices Q and R in the TIGO(τ)-method. This can be achieved by introducing another parameter p, the largest number of nonzero entries permitted in each row or column of the matrices Q and R. The resulting method, called the Generalized Threshold Incomplete Givens Orthogonalization method or the GTIGO(τ, p)-method, is described in the following.

Method 5.2: GTIGO(τ, p)-method
1. Input the dropping tolerance $\tau^{(Q)}$ for the factor Q
2. Input the dropping tolerance $\tau^{(R)}$ for the factor R
3. Input the memory-control tolerance $p^{(Q)}$ for the factor Q
4. Input the memory-control tolerance $p^{(R)}$ for the factor R
5. Set $Q = I$
6. For $j = 1, \ldots, n-1$ Do:
7. Set $P_j := \{i \mid (i,j) \notin P_{A,L}, |a_{ij}| > \tau^{(R)}, j+1 \leq i \leq n\}$
8. Define $k_r(j) := \max\{i \mid (i,j) \in P_{A,L} \cup P_j\}$
9. If $k_r(j) = j$ then cycle
10. For $i = k_r(j)$ DownTo $j+1$ and $(i,j) \in P_{A,L} \cup P_j$ Do:
11. Compute $\varrho := \sqrt{a_{jj}^2 + a_{ij}^2}$
12. Compute $c := a_{jj}/\varrho$
13. Compute $s := a_{ij}/\varrho$
14. Set $a_{jj} := \varrho$
15. Set $\tilde{P}_i := \{k \mid (i,k) \notin P_A, |a_{ik}| > \tau^{(R)}, j+1 \leq k \leq n\}$
16. Define $k_c(i) := \max\{k \mid (i,k) \in P_A \cup \tilde{P}_i\}$

17. For $k = j+1, \ldots, k_c(i)$ and $(i,k) \in P_A \cup \tilde{P}_i$ Do:
18. Compute $temp_k := -sa_{jk} + ca_{ik}$
19. EndDo
20. Set $\tilde{P}_j := \{k \mid (j,k) \notin P_{A,U}, |a_{ik}| > \tau^{(R)}, j \le k \le n\}$
21. Define $k_c(j) := \max\{k \mid (j,k) \in P_{A,U} \cup \tilde{P}_j\}$
22. For $k = j, \ldots, k_c(j)$ and $(j,k) \in P_{A,U} \cup \tilde{P}_j$ Do:
23. Compute $a_{jk} := ca_{jk} + sa_{ik}$
24. EndDo
25. Keep only the $p^{(R)}$ largest entries in the j-th row a_{j*}
26. For $k = j+1, \ldots, k_c(i)$ and $(i,k) \in P_A \cup \tilde{P}_i$ Do:
27. Set $a_{ik} := temp_k$
28. EndDo
29. Keep only the $2p^{(R)}$ largest entries in the i-th row a_{i*}
30. For $k = 1, \ldots, n$ Do:
31. Compute $temp := -sq_{kj} + cq_{ki}$
32. Compute $q_{kj} := cq_{kj} + sq_{ki}$
33. If $|temp| \le \tau^{(Q)}$ then Set $q_{ki} := 0$, Else
34. Set $q_{ki} := temp$
35. If $|q_{kj}| \le \tau^{(Q)}$ then Set $q_{kj} := 0$
36. EndDo
37. Keep only the $p^{(Q)}$ largest entries in the i-th column q_{*i}
38. Keep only the $p^{(Q)}$ largest entries in the j-th column q_{*j}
39. EndDo
40. For $k = j, \ldots, n$ and $|a_{jk}| > \tau^{(R)}$ Do
41. Set $r_{jk} := a_{jk}$
42. EndDo
43. Keep only the $p^{(R)}$ largest entries in the j-th row r_{j*}
44. EndDo
45. Set $r_{nn} := a_{nn}$

The drop tolerances $\tau^{(Q)}$ and $\tau^{(R)}$ in the GTIGO(τ, p)-method can be determined according to each row or column of the incomplete orthogonal and upper triangular matrices Q and R, respectively, in a similar way to the TIGO(τ)-method. Moreover, it is possible to dynamically adjust the parameters $p^{(Q)}$ and $p^{(R)}$ during the incomplete Givens orthogonalization process.

Following exactly the analysis in Section 3, analogous properties to the GIGO-method can be established for both the TIGO(τ)-method and the GTIGO(τ, p)-method.

5 Conclusions

We have presented an extensive sequence of incomplete orthogonal factorization methods for general nonsingular and unsymmetric matrices. These factorizations are obtained from the Givens orthogonalization process by dropping fill-in according to either position or magnitude and by limiting the number of entries according to available storage. Theoretical analyses show that these new methods can successfully produce a nonsingular sparse incomplete upper triangular factor and yield either a complete orthogonal factor or at least a sparse nonsingular incomplete orthogonal factor for any nonsingular and unsymmetric matrix. Therefore, they have the potential to provide robust, accurate and efficient preconditioners for Krylov subspace iterations for solving large sparse systems of linear equations, possibly in combination with a reordering technique and/or with some pivoting strategy. The theory presented here would equally apply with the use of such techniques. Numerical results for a variety of practical problems will be reported in a separate paper, so that the numerical behaviour and range of application of these new incomplete orthogonal factorization preconditioners can be further examined.

Acknowledgement

The authors are very much indebted to Professor G.H. Golub for his constant concern and encouragement, and Professor L.N. Trefethen for his useful discussions.

References

Axelsson, O. (1994), *Iterative Solution Methods*, Cambridge University Press, New York.

Axelsson, O. and Barker, V. (1985), *Finite Element Solutions of Boundary Value Problems. Theory and Computation*, Academic Press, New York.

Concus, P., Golub, G. and Meurant, G. (1985), 'Block preconditioning for the conjugate gradient method', *SIAM J. Scientific and Statistical Computing* **6**, 220–252.

Daniel, J., Gragg, W., Kaufman, L. and Stewart, G. W. (1976), 'Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization', *Mathematics of Computation* **30**, 772–95.

Donato, J. and Chan, T. (1992), 'Fourier analysis of incomplete factorization preconditioners for three dimensional anisotropic problems', *SIAM J. Scientific and Statistical Computing* **13**, 319–338.

Elman, H. (1986), 'A stability analysis of incomplete LU factorizations', *Mathematics of Computation* **47**, 191–217.

Elman, H. (1989), 'Relaxed and stabilized incomplete factorizations for non-self-adjoint linear systems', *BIT* **29**, 890–915.

Gustafsson, I. (1978), 'A class of first order factorizations', *BIT* **18**, 142–156.

Manteuffel, T. (1980), 'An incomplete factorization technique for positive definite linear systems', *Mathematics of Computation* **34**, 473–497.

Meijerink, J. and van der Vorst, H. (1977), 'An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix', *Mathematics of Computation* **31**, 148–162.

Saad, Y. (1988), 'Preconditioning techniques for nonsymmetric and indefinite linear systems', *J. Comput. Appl. Math.* **24**, 89–105.

Saad, Y. (1994), 'ILUT: A dual threshold incomplete ILU factorization', *Numerical Linear Algebra with Applications* **1**, 387–402.

Saad, Y. (1996), *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston.

Wang, X., Gallivan, K. and Bramley, R. (1997), 'CIMGS: A incomplete orthogonalization preconditioner', *SIAM J. Scientific Computing* **18**, 516–536.