LOSS AND RECAPTURE OF ORTHOGONALITY IN THE MODIFIED GRAM-SCHMIDT ALGORITHM *

Å. BJÖRCK† AND C. C. PAIGE‡

To our close friend and mentor Gene Golub, on his 60th birthday. This is but one of the many topics on which Gene has generated so much interest, and shed so much light.

Abstract. This paper arose from a fascinating observation, apparently by Charles Sheffield, and relayed to us by Gene Golub, that the QR factorization of an $m \times n$ matrix A via MGS is *numerically* equivalent to that arising from Householder transformations applied to the matrix A augmented by an n by n zero matrix. This is explained in a clear and simple way, and then combined with a well known rounding error result to show the upper triangular matrix R from MGS is about as accurate as R from other QR factorizations. The special structure of the product of the Householder transformations is derived, and then used to explain and bound the loss of orthogonality in MGS. Finally this numerical equivalence is used to show how orthogonality in MGS can be regained in general. This is illustrated by deriving a numerically stable algorithm based on MGS for a class of problems which includes solution of nonsingular linear systems, minimum 2-norm solution of underdetermined linear systems, and linear least squares problems. A brief discussion on the relative merits of such algorithms is included.

Key Words. orthogonal matrices, QR factorization, Householder transformations, least squares, minimum norm solution, numerical stability, Gram-Schmidt, augmented systems.

AMS Subject Classifications: 65F25, 65G05, 65F05, 65F20

1. Introduction. We consider a matrix $A \in \mathbf{R}^{m\times n}$ with rank $n \le m$. The Modified Gram-Schmidt algorithm (MGS) in theory produces Q_1 and R in the QR factorization

$$A = Q\begin{pmatrix} R \\ 0 \end{pmatrix} = Q_1 R, \qquad Q = (\, Q_1 \quad Q_2 \,) \tag{1.1}$$

where Q is orthogonal and R upper triangular. In practice if the condition number $\kappa = \kappa(A) \equiv \sigma_1/\sigma_n$ is large, ($\sigma_1 \ge \ldots \ge \sigma_n$ being the singular values of A), then the columns of Q_1 are not accurately orthogonal [3]. If orthogonality is crucial, then usually either rotations or Householder transformations have been used to compute the QR factorization. Here we show how MGS can be used just as stably for many problems requiring this orthogonality.

We derive some important properties of MGS in the presence of rounding errors. In particular we show that the R obtained from MGS is numerically as good as that obtained from rotations or Householder transformations. We present new insights on the loss of orthogonality in Q_1 from MGS, and show how this can effectively be regained in computations which use Q_1, without altering the MGS algorithm or reorthogonalizing the columns of Q_1. As a practical example of this, we indicate how Q_1 and R from MGS may be used to solve an important class of problems reliably, despite the loss of orthogonality in Q_1. This new approach seems applicable to most problems for which MGS is in theory relevant.

* This research was partially supported by NSERC of Canada Grant No. A9236.

† Mathematics, Linköping University, S-581 83, Linköping, Sweden, (akbjo@math.liu.se).

‡ Computer Science, McGill University, Montreal, Quebec, Canada, H3A 2A7, (chris@cs.mcgill.ca).

The class of problems we consider is that of solving the symmetric indefinite linear system involving $A \in \mathbf{R}^{m\times n}$ with rank n

$$\begin{pmatrix} I & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}. \tag{1.2}$$

In general we call (1.2) the Augmented System Formulation (ASF) of the following two problems, since it represents the conditions for their solution:

$$\min_x \|b - x\|_2, \qquad A^T x = c, \tag{1.3}$$

$$\min_y \{\|b - Ay\|_2^2 + 2c^T y\}. \tag{1.4}$$

We examine these problems more fully in [5]. The ASF can be obtained by differentiating the Lagrangian $\|b-x\|_2^2 + 2y^T(A^T x - c)$ of (1.3), and equating to zero. Here y is the vector of Lagrange multipliers. The ASF can also be obtained by differentiating (1.4) to give $A^T(b - Ay) = c$, and setting x to be the "residual" $x = b - Ay$.

The ASF covers two important special cases. Setting $b = 0$ in (1.3), and so in (1.2), gives the problem of finding the minimum 2-norm solution of a Linear Underdetermined System (LUS). Setting $c = 0$ in (1.4) gives the much used Linear Least Squares (LLS) problem. The ASF also occurs in its full form (1.2) in the iterative refinement of least squares solutions [2].

Using the QR factorization (1.1) we can transform (1.2) into

$$\begin{pmatrix} I & \begin{pmatrix} R \\ 0 \end{pmatrix} \\ \begin{pmatrix} R^T & 0 \end{pmatrix} & 0 \end{pmatrix} \begin{pmatrix} Q^T x \\ y \end{pmatrix} = \begin{pmatrix} Q^T b \\ c \end{pmatrix}.$$

This gives one method for solving (1.2):

$$z = R^{-T}c, \qquad \begin{pmatrix} d \\ f \end{pmatrix} = Q^T b, \qquad x = Q\begin{pmatrix} z \\ f \end{pmatrix}, \qquad y = R^{-1}(d - z). \tag{1.5}$$

Using $x = Q_1 z + Q_2 f = Q_1 z + Q_2 Q_2^T b = Q_1 z + (I - Q_1 Q_1^T)b$ an obvious variant is

$$z = R^{-T}c, \qquad d = Q_1^T b, \qquad x = b - Q_1(d - z), \qquad y = R^{-1}(d - z). \tag{1.6}$$

Björck [2] showed (1.5) is backward stable for (1.2) using the Householder QR factorization. Since (1.5) uses Q, (1.6) seems preferable if x is required and only Q_1 is available. However, as we shall see it cannot in general be recommended when Q_1 is obtained by MGS. We will show how to develop more reliable algorithms based on Q_1 from MGS.

In Section 2 we illustrate the important but not widely appreciated result that MGS is *numerically* equivalent to the Householder QR factorization applied to A augmented with a block of zeros. From this we show in Section 3 that the computed R from MGS is numerically as satisfactory as that obtained using Householder QR on A. The product P of the Householder transformations from the QR factorization of $\begin{pmatrix} O_n \\ A \end{pmatrix}$ is crucial for a full understanding of MGS. P has a simple and important structure, and this is derived in the theorem in Section 4. This structure shows exactly how the computed $\bar{Q}_1$ from MGS can lose orthogonality. In Section 5 this structure is used to bound the loss of orthogonality of $\bar{Q}_1$, while Section 6 shows how the lost orthogonality can be compensated for just by using $\bar{Q}_1$ differently without altering $\bar{Q}_1$ or MGS. We illustrate this by producing a new backward stable algorithm for (1.2) using the computed $\bar{Q}_1$ and $\bar{R}$ from MGS. In Section 7 we consider when we might use MGS in preference to the Householder QR factorization of A.

2. Modified Gram-Schmidt as a Householder Method. The MGS algorithm computes a sequence of matrices $A = A^{(1)}, A^{(2)}, \ldots, A^{(n+1)} = Q_1 \in \mathbf{R}^{m\times n}$, where $A^{(k)} = (q_1, \ldots, q_{k-1}, a_k^{(k)}, \ldots, a_n^{(k)})$. Here the first $(k-1)$ columns are final columns in Q_1, and $a_k^{(k)}, \ldots, a_n^{(k)}$ have been made orthogonal to $q_1, \ldots, q_{k-1}$. In the kth step we take

$$q'_k = a_k^{(k)}, \quad \rho_{kk} = \|q'_k\|_2, \quad q_k = q'_k/\rho_{kk}, \tag{2.1}$$

and orthogonalize $a_{k+1}^{(k)}, \ldots, a_n^{(k)}$ against q_k using the orthogonal projector $I - q_k q_k^T$,

$$\begin{aligned} a_j^{(k+1)} &= (I - q_k q_k^T) a_j^{(k)} = a_j^{(k)} - q_k \rho_{kj}, \\ \rho_{kj} &= q_k^T a_j^{(k)}, \qquad j = k+1, \ldots, n. \end{aligned} \tag{2.2}$$

We see $A^{(k)} = A^{(k+1)} R_k$ where R_k has the same kth row as upper triangular $R \equiv (\rho_{ij})$, but is the unit matrix otherwise. After n steps we have obtained the factorization

$$A = A^{(1)} = A^{(2)} R_1 = A^{(3)} R_2 R_1 = A^{(n+1)} R_n \ldots R_1 = Q_1 R, \tag{2.3}$$

where in exact arithmetic the columns of Q_1 are orthonormal by construction. Note that in the modified Gram-Schmidt algorithm, as opposed to the classical version, all the projections $q_k \rho_{kj}$ are subtracted from the $a_j^{(k)}$ sequentially as soon as q_k is computed. In practice a square root free version is often used, where one computes Q'_1, R', and $D = \mathrm{diag}(\gamma_1, \ldots, \gamma_n)$ in the scaled factorization, taking q'_k as above,

$$A = Q'_1 R', \qquad Q'_1 = (q'_1, \ldots, q'_n), \qquad \gamma_k = (q'_k)^T q'_k, \qquad k = 1, \ldots, n, \tag{2.4}$$

with $R' = (\rho'_{kj})$ *unit* upper triangular, and $\rho'_{kj} = (q'_k)^T a_j^{(k)}/\gamma_k$, $j > k$.

It was reported in [4] that the modified Gram-Schmidt algorithm for the QR factorization can be interpreted as Householder's method applied to the matrix A *augmented with a square matrix of zero elements on top.* This is not only true in theory, but in the presence of rounding errors as well. This observation is originally due to Charles Sheffield, and was communicated to the authors by Gene Golub. Because it is such an important but unexpected result, we will discuss this relationship in some detail. First we look at the theoretical result.

Let $A \in \mathbf{R}^{m\times n}$ have rank n, and let $O_n \in \mathbf{R}^{n\times n}$ be a zero matrix. Consider the two QR factorizations (here we use Q for $m \times m$ and P for $(m+n) \times (m+n)$ orthogonal matrices),

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix} = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R \\ 0 \end{pmatrix},$$

$$\tilde{A} \equiv \begin{pmatrix} O_n \\ A \end{pmatrix} = P \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix} = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}. \tag{2.5}$$

Since A has rank n, P_{11} is zero, P_{21} is an $m \times n$ matrix of orthonormal columns, and $A = Q_1 R = P_{21}\tilde{R}$. If upper triangular R and $\tilde{R}$ are both chosen to have positive diagonal elements in $A^T A = R^T R = \tilde{R}^T \tilde{R}$, then $R = \tilde{R}$ by uniqueness, so $P_{21} = Q_1$ can be found from any QR factorization of the augmented matrix. The last m columns of P are then arbitrary up to an $m \times m$ orthogonal multiplier. The important result is that the *Householder* QR factorization of the augmented matrix is *numerically* equivalent to MGS applied to A.

To see this, remember that with e_k the kth column of the unit matrix, the Householder transformation $Pa = e_1\rho$ uses $P = I - 2vv^T/\|v\|_2^2$, $v = a - e_1\rho$, $\rho = \pm\|a\|_2$. If (2.5) is obtained using Householder transformations, then

$$P^T = P_n \dots P_2 P_1, \qquad P_k = I - 2\hat{v}_k\hat{v}_k^T/\|\hat{v}_k\|_2^2, \qquad k = 1, \dots, n, \tag{2.6}$$

where the vectors $\hat{v}_k$ are described below. Now from MGS applied to $A^{(1)} = A$, $\rho_{11} = \|a_1^{(1)}\|_2$ and $a_1^{(1)} = q_1' = q_1\rho_{11}$, so for the first Householder transformation applied to the augmented matrix

$$\begin{aligned} \tilde{A}^{(1)} &\equiv \begin{pmatrix} O_n \\ A^{(1)} \end{pmatrix}, \qquad \tilde{a}_1^{(1)} = \begin{pmatrix} 0 \\ a_1^{(1)} \end{pmatrix}, \\ \hat{v}_1^{(1)} &\equiv \begin{pmatrix} -e_1\rho_{11} \\ q_1' \end{pmatrix} = \rho_{11}v_1, \qquad v_1 = \begin{pmatrix} -e_1 \\ q_1 \end{pmatrix}, \end{aligned}$$

(since there can be no cancellation we take $\rho_{kk} \geq 0$). But $\|v_1\|_2^2 = 2$, giving

$$P_1 = I - 2\hat{v}_1\hat{v}_1^T/\|\hat{v}_1\|_2^2 = I - 2v_1v_1^T/\|v_1\|_2^2 = I - v_1v_1^T,$$

and

$$P_1\tilde{a}_j^{(1)} = \tilde{a}_j^{(1)} - v_1v_1^T\tilde{a}_j^{(1)} = \begin{pmatrix} 0 \\ a_j^{(1)} \end{pmatrix} - \begin{pmatrix} -e_1 \\ q_1 \end{pmatrix} q_1^T a_j^{(1)} = \begin{pmatrix} e_1\rho_{1j} \\ a_j^{(2)} \end{pmatrix},$$

so

$$P_1\tilde{A}^{(1)} = \begin{pmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1n} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \\ 0 & a_2^{(2)} & \cdots & a_n^{(2)} \end{pmatrix},$$

where these values are clearly *numerically* the same as in the first step of MGS on A. We see the next Householder transformation produces the second row of R and $a_3^{(3)}, \dots, a_n^{(3)}$, just as in MGS. Carrying on this way we see this Householder QR is numerically equivalent to MGS applied to A, and that every P_k is effectively defined by Q_1, since

$$P_k = I - v_kv_k^T, \qquad v_k = \begin{pmatrix} -e_k \\ q_k \end{pmatrix}, \qquad k = 1, \dots, n. \tag{2.7}$$

P gives us a key to understanding the *numerical* behavior of MGS. First note that *in theory* $v_i^Tv_j = e_i^Te_j + q_i^Tq_j = 0$ if $i \neq j$, so $P_iP_j = I - v_iv_i^T - v_jv_j^T$, and $P^T = P_n \cdots P_1 = I - v_1v_1^T - v_2v_2^T - \cdots - v_nv_n^T$ is symmetric, so using Householder transformations in (2.5),

$$\begin{aligned} P_{11} &= 0, \\ P_{12}^T &= P_{21} = q_1 e_1^T + \cdots + q_n e_n^T = Q_1, \\ P_{22} &= I - q_1 q_1^T - \cdots - q_n q_n^T = I - Q_1 Q_1^T = Q_2 Q_2^T. \end{aligned}$$

This shows such special orthogonal matrices are fully defined by their $(1,2)$ blocks,

$$P = \begin{pmatrix} O_n & Q_1^T \\ Q_1 & I - Q_1 Q_1^T \end{pmatrix}. \tag{2.8}$$

3. Accuracy of R from Modified Gram-Schmidt. A rounding error analysis of MGS was given in [3]. There it was shown that the *computed* $\bar{Q}_1$ and $\bar{R}$ satisfy

$$\begin{aligned} A + \bar{E} = \bar{Q}_1 \bar{R}, \qquad \|\bar{E}\|_2 &\leq \bar{c}_1 u \|A\|_2, \\ \|I - \bar{Q}_1^T \bar{Q}_1\|_2 &\leq \bar{c}_2 \kappa u, \end{aligned} \tag{3.1}$$

where $\bar{c}_i$ are constants depending on m, n and the details of the arithmetic, and u is the unit roundoff. Hence $\bar{Q}_1 \bar{R}$ accurately represents A and the departure from orthogonality can be bounded in terms of the condition number $\kappa = \sigma_1/\sigma_n$.

From the numerical equivalence shown in the previous section it follows that the backward error analysis for the Householder QR factorization of the augmented matrix in (2.5) can also be applied to the modified Gram-Schmidt algorithm on A. Here we will do this, and in this section and Section 5 we will rederive (3.1) as well as give some new results. This is a simple and unified approach, in that the one analysis of orthogonal transformations can be used to analyse the QR factorization via both Householder transformations and MGS. It also deepens our understanding of the MGS algorithm and its possible uses.

Let $\bar{Q}_1 = (\bar{q}_1, \ldots, \bar{q}_n)$ be the matrix of vectors computed by MGS, and for $k = 1, \ldots, n$ define

$$\begin{aligned} \bar{v}_k &= \begin{pmatrix} -e_k \\ \bar{q}_k \end{pmatrix}, \qquad \bar{P}_k = I - \bar{v}_k \bar{v}_k^T, \qquad \bar{P} = \bar{P}_1 \bar{P}_2 \ldots \bar{P}_n, \\ \tilde{q}_k = \bar{q}_k / \|\bar{q}_k\|_2, \quad \tilde{v}_k &= \begin{pmatrix} -e_k \\ \tilde{q}_k \end{pmatrix}, \quad \tilde{P}_k = I - \tilde{v}_k \tilde{v}_k^T, \quad \tilde{P} = \tilde{P}_1 \tilde{P}_2 \ldots \tilde{P}_n. \end{aligned} \tag{3.2}$$

Then $\bar{P}_k$ is the computed version of the Householder matrix applied in the kth step of the Householder QR factorization of $\binom{O_n}{A}$, and $\tilde{P}_k$ is its orthonormal equivalent, so that $\tilde{P}_k^T \tilde{P}_k = I$. Wilkinson [11, pp. 153–162] has given a general error analysis of orthogonal transformations of this type. From this it follows that for $\bar{R}$ computed by MGS, the equivalent of (2.5) is

$$\begin{pmatrix} E_1 \\ A + E_2 \end{pmatrix} = \tilde{P} \begin{pmatrix} \bar{R} \\ 0 \end{pmatrix}, \quad \bar{P} = \tilde{P} + E',$$

$$\|E_i\|_2 \leq c_i u \|A\|_2, \quad i = 1, 2; \quad \|E'\|_2 \leq c_3 u, \tag{3.3}$$

where again c_i are constants depending on m, n and the details of the arithmetic.

To show this $\bar{R}$ from MGS, or the Householder QR factorization of the augmented matrix, is numerically about as good as that from the ordinary Householder QR factorization of A, we use the following general result.

LEMMA 3.1. *For any matrices satisfying*

$$\begin{pmatrix} E_1 \\ A+E_2 \end{pmatrix} = \begin{pmatrix} P_{11} \\ P_{21} \end{pmatrix} R, \qquad P_{11}^T P_{11} + P_{21}^T P_{21} = I,$$

there exist $\hat{Q}_1$ *and* E *such that*

$$A + E = \hat{Q}_1 R, \qquad \hat{Q}_1^T \hat{Q}_1 = I,$$

$$\|\hat{Q}_1 - P_{21}\|_2 \leq \|P_{11}\|_2^2, \tag{3.4}$$

$$\|(\hat{Q}_1 - P_{21})R\|_2 \leq \|P_{11}\|_2 \|E_1\|_2, \tag{3.5}$$

$$\|E\|_2 \leq \|P_{11}\|_2\|E_1\|_2 + \|E_2\|_2 \leq \|E_1\|_2 + \|E_2\|_2. \tag{3.6}$$

Proof. Consider the CS decomposition (see for example [7, p.77]) $P_{11} = U_1 C W^T$, $P_{21} = V_1 S W^T$, where $U = (U_1, U_2)$, $V = (V_1, V_2)$ are square orthonormal matrices and C and S are nonnegative diagonal matrices with $C^2 + S^2 = I$. Define $\hat{Q}_1 \equiv V_1 W^T$, the closest orthonormal matrix to P_{21} in any unitarily invariant norm, then since $(I+S)(I-S) = C^2$,

$$\begin{aligned}
\hat{Q}_1 - P_{21} &= V_1(I-S)W^T = V_1(I+S)^{-1}W^T W C U_1^T U_1 C W^T \\
&= V_1(I+S)^{-1}W^T P_{11}^T P_{11}, \\
(\hat{Q}_1 - P_{21})R &= V_1(I+S)^{-1}W^T P_{11}^T E_1,
\end{aligned}$$

from which the first two bounds follow. Next

$$E = \hat{Q}_1 R - A = (\hat{Q}_1 - P_{21})R + E_2,$$

from which the third bound follows. □

Using these results we see when $\bar{R}$ is computed using MGS, so $\bar{R}$ satisfies (3.3), there exists *orthonormal* $\hat{Q}_1$ such that, writing $c = c_1 + c_2$,

$$A + E = \hat{Q}_1 \bar{R}, \qquad \hat{Q}_1^T \hat{Q}_1 = I, \qquad \|E\|_2 \leq cu\|A\|_2. \tag{3.7}$$

This means if $\bar{\sigma}_1 \geq \ldots \geq \bar{\sigma}_n$ are the singular values of $\bar{R}$, and $\sigma_1 \geq \ldots \geq \sigma_n$ are those of A,

$$|\,\bar{\sigma}_i - \sigma_i\,| \leq cu\sigma_1, \qquad i = 1, \ldots, n. \tag{3.8}$$

Thus $\bar{R}$ from MGS is not only the same as $\bar{R}$ from the Householder QR factorization applied to A augmented by a square block of zeros, but (3.7) shows it is comparable in accuracy to the upper triangular matrix from the Householder or Givens QR factorization applied to A alone. Also (3.8) shows the singular values of $\bar{R}$ are very close to those of A, which means we could use MGS as a first step in finding the singular values of A, and justifies an algorithm by Longley in [9, Ch. 9]. Since we have not required A to be full rank as yet in this section, this fact also ensures $\bar{R}$ from MGS can be used in any computation for finding the rank of A. Here we will just use this knowledge to simplify our bounds below.

In fact $\bar{R}$ is usually even better than (3.7) suggests. We see $\bar{R}$ is nonsingular if $cu\sigma_1 < \sigma_n$, that is if $cu\kappa < 1$, so we make the following assumption and definition,

$$cu\kappa < 1, \qquad \eta \equiv (1 - cu\kappa)^{-1}, \tag{3.9}$$

where usually $\eta \sim 1$. Then

$$\|A\|_2\|\bar{R}^{-1}\|_2 = \sigma_1/\bar{\sigma}_n \le \sigma_1/(\sigma_n - cu\sigma_1) = \eta\kappa, \tag{3.10}$$

and $E_1 = \tilde{P}_{11}\bar{R}$, so

$$\|\tilde{P}_{11}\|_2 = \|E_1\bar{R}^{-1}\|_2 \le c_1 u\eta\kappa, \qquad \|\bar{P}_{11}\|_2 \le (c_1\eta\kappa + c_3)u. \tag{3.11}$$

From (3.6)

$$\|E\|_2 \le \|\tilde{P}_{11}\|_2\|E_1\|_2 + \|E_2\|_2 \le c_1^2u^2\eta\kappa\|A\|_2 + \|E_2\|_2 \tag{3.12}$$

showing that the first term on the right will be negligible if $\eta c_1 u\kappa \ll 1$, which is usually true.

We will show how all of $\tilde{P}$ and $\bar{P}$ depend crucially on $\tilde{P}_{11}$ and $\bar{P}_{11}$ respectively, so the bounds in (3.11) are important in understanding the loss of orthogonality in MGS. Since $\bar{R}$ is numerically about as good as we can hope for, it is clear that the main drawback of MGS is this lack of orthogonality in $\bar{Q}_1 = (\bar{q}_1, \ldots, \bar{q}_n)$, so we examine this in the next two sections. (As is mentioned in Section 7, another less important drawback is that the operation count is slightly higher for MGS than for the Householder QR factorization.)

4. Structure of P, $\bar{P}$ and $\tilde{P}$ from the Householder QR factorization of the augmented matrix. It is well known that the orthogonality of the ideal Q_1 is lost in MGS because of cancellation in the subtractions in (2.2), and that this can give a severely nonorthogonal computed $\bar{Q}_1$. In order to understand this loss fully and later to bound it, the following theorem provides the detailed structures of $\bar{P}$ and $\tilde{P}$ in (3.2) as functions of the computed $\bar{Q}_1$ and the normalized $\tilde{Q}_1 \equiv (\tilde{q}_1, ..., \tilde{q}_n)$ respectively. Note the theorem is for general $Q_1 = (q_1, \ldots, q_n)$, and so will apply to P, $\bar{P}$ and $\tilde{P}$. The idea is that *any* matrix $P = P_1P_2\ldots P_n$ with $P_k = I - v_kv_k^T$ and $v_k^T = (-e_k^T, q_k^T)$ has a very special structure, and the theorem reveals this. In this structure the whole matrix is seen to depend only on the leading $n \times n$ block P_{11} of P, and on Q_1. But we have bounds on *our* $\tilde{P}_{11}$ and $\bar{P}_{11}$ in (3.11), and so will be able to understand and bound the loss of orthogonality in $\tilde{Q}_1$ or $\bar{Q}_1$ from MGS. What is more all such P_{11} have special structure too, being strictly upper triangular.

THEOREM 4.1. *Let $Q_1 = (q_1, \ldots, q_n) \in \mathbf{R}^{m\times n}$, and for $k = 1, \ldots, n$, define*

$$M_k = I - q_kq_k^T, \qquad v_k = \begin{pmatrix} -e_k \\ q_k \end{pmatrix} \in \mathbf{R}^{m+n}, \qquad P_k = I - v_kv_k^T.$$

Then with the partitioning we use throughout this theorem

$$P \;\equiv\; P_1P_2\ldots P_n \equiv \begin{array}{c} \\ n \\ m \end{array}\!\!\begin{array}{c} \begin{array}{cc} n & m \end{array} \\ \left(\begin{array}{c|c} P_{11} & P_{12} \\ \hline P_{21} & P_{22} \end{array}\right) \end{array}$$

$$(4.1) = \left(\begin{array}{cccccc|c} 0 & q_1^Tq_2 & q_1^TM_2q_3 & \cdots & q_1^TM_2M_3\cdots M_{n-1}q_n & & q_1^TM_2M_3\cdots M_n \\ 0 & 0 & q_2^Tq_3 & \cdots & q_2^TM_3M_4\cdots M_{n-1}q_n & & q_2^TM_3M_4\cdots M_n \\ \vdots & \vdots & \vdots & \cdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & q_{n-1}^Tq_n & & q_{n-1}^TM_n \\ 0 & 0 & 0 & \cdots & 0 & & q_n^T \\ \hline q_1 & M_1q_2 & M_1M_2q_3 & \cdots & M_1M_2\cdots M_{n-1}q_n & & M_1M_2\cdots M_n \end{array}\right)$$

$$(4.2) = \left(\begin{array}{c|c} P_{11} & (I-P_{11})Q_1^T \\ \hline Q_1(I-P_{11}) & I - Q_1(I-P_{11})Q_1^T \end{array}\right).$$

P is orthonormal if and only if $\|q_k\|_2 = 1$ for $k = 1, \ldots, n$; and $P_{11} = 0$ if and only if $Q_1^T Q_1$ is diagonal.

There is a short proof that does not give (4.1), but since (4.1) reveals the detailed structure of P, we give a longer proof. Note if q_k has length 1 then M_k is a projector, and from (4.1) the second column of P_{21} is that part of q_2 orthogonal to q_1, the third is q_3 orthogonalized against q_2 and the result orthogonalized against q_1, and so on. However this is not the same as reorthogonalizing the q_k.

Proof. To determine the first n columns of $P = P_1 P_2 \cdots P_n$ note

$$P_k = I - v_k v_k^T = I - \left(\frac{-e_k}{q_k}\right)\left(-e_k^T \mid q_k^T \right) = \left(\begin{array}{c|c} I - e_k e_k^T & e_k q_k^T \\ \hline q_k e_k^T & M_k \end{array}\right)$$

and let $1 \le j \le n$. If $j \ne k$ then $P_k e_j = e_j$, while $P_j e_j = \begin{pmatrix} 0 \\ I_m \end{pmatrix} q_j$, so

$$\begin{aligned} Pe_j = P_1 P_2 \cdots P_n e_j &= P_1 P_2 \cdots P_j e_j = P_1 P_2 \cdots P_{j-2} \left(\frac{e_{j-1} q_{j-1}^T}{M_{j-1}}\right) q_j \\ &= P_1 P_2 \cdots P_{j-3} \left(\frac{e_{j-1} q_{j-1}^T + e_{j-2} q_{j-2}^T M_{j-1}}{M_{j-2} M_{j-1}}\right) q_j \end{aligned}$$

$$= \left(\begin{array}{c} q_1^T M_2 \cdots M_{j-1} q_j \\ q_2^T M_3 \cdots M_{j-1} q_j \\ \vdots \\ q_{j-2}^T M_{j-1} q_j \\ q_{j-1}^T q_j \\ 0 \\ \vdots \\ 0 \\ \hline M_1 M_2 \cdots M_{j-1} q_j \end{array}\right) = \left(\frac{P_{11}}{P_{21}}\right) e_j = \left(\frac{p_{1j}}{p_{2j}}\right) \equiv \left(\begin{array}{c} \pi_{1j} \\ \pi_{2j} \\ \vdots \\ \pi_{j-2,j} \\ \pi_{j-1,j} \\ \pi_{jj} \\ \vdots \\ \pi_{nj} \\ \hline p_{2j} \end{array}\right) \tag{4.3}$$

say, which gives the $(1,1)$ and $(2,1)$ blocks of (4.1). For the last m columns we have

$$\begin{aligned} \left(\frac{P_{12}}{P_{22}}\right) &= P \left(\frac{0}{I_m}\right) = P_1 P_2 \cdots P_{n-1} \left(\frac{e_n q_n^T}{M_n}\right) \\ &= P_1 P_2 \cdots P_{n-2} \left(\frac{e_n q_n^T + e_{n-1} q_{n-1}^T M_n}{M_{n-1} M_n}\right) = \left(\begin{array}{c} q_1^T M_2 \cdots M_n \\ q_2^T M_3 \cdots M_n \\ \vdots \\ q_{n-1}^T M_n \\ q_n^T \\ \hline M_1 \cdots M_n \end{array}\right), \end{aligned} \tag{4.4}$$

which completes the proof of (4.1). Next from (4.3)

$$\begin{aligned} P_{21} e_j &= (I - q_1 q_1^T) M_2 M_3 \cdots M_{j-1} q_j \\ &= M_2 M_3 \cdots M_{j-1} q_j - q_1 \pi_{1j} \\ &= M_3 M_4 \cdots M_{j-1} q_j - q_1 \pi_{1j} - q_2 \pi_{2j} \end{aligned}$$

$$
\begin{aligned}
&= M_{j-1}q_j - q_1\pi_{1j} - q_2\pi_{2j} - \cdots - q_{j-2}\pi_{j-2,j} \\
&= q_j - q_1\pi_{1j} - q_2\pi_{2j} - \cdots - q_{j-1}\pi_{j-1,j} \\
&= Q_1 e_j - Q_1 P_{11} e_j,
\end{aligned}
$$

so $P_{21} = Q_1(I - P_{11})$, giving the $(2,1)$ block of (4.2). Next from (4.4)

$$
\begin{aligned}
e_i^T P_{12} &= q_i^T M_{i+1}\cdots M_{n-1}M_n \\
&= q_i^T M_{i+1}\cdots M_{n-1} - q_i^T M_{i+1}\cdots M_{n-1}q_n q_n^T \\
&= q_i^T M_{i+1}\cdots M_{n-1} - \pi_{in}q_n^T \\
&= q_i^T M_{i+1}\cdots M_{n-2} - \pi_{i,n-1}q_{n-1}^T - \pi_{in}q_n^T \\
&= q_i^T M_{i+1} - \pi_{i,i+2}q_{i+2}^T - \cdots - \pi_{in}q_n^T \\
&= q_i^T - \pi_{i,i+1}q_{i+1}^T - \cdots - \pi_{in}q_n^T \\
&= e_i^T Q_i^T - e_i^T P_{11} Q_1^T,
\end{aligned}
$$

so $P_{12} = (I - P_{11})Q_1^T$, giving the $(1,2)$ block of (4.2). We can now use the structure of P_{21} in (4.1) to give

$$
\begin{aligned}
P_{22} &= M_1 M_2 \cdots M_n \\
&= M_1 M_2 \cdots M_{n-1} - M_1 M_2 \cdots M_{n-1} q_n q_n^T \\
&= M_1 M_2 \cdots M_{n-1} - P_{21} e_n q_n^T \\
&= M_1 M_2 \cdots M_{n-2} - P_{21} e_{n-1} q_{n-1}^T - P_{21} e_n q_n^T \\
&= I - q_1 q_1^T - P_{21} e_2 q_2^T - P_{21} e_3 q_3^T - \cdots - P_{21} e_n q_n^T \\
&= I - P_{21}(e_1 q_1^T + e_2 q_2^T + \cdots + e_n q_n^T) \\
&= I - P_{21} Q_1^T = I - Q_1(I - P_{11})Q_1^T,
\end{aligned}
$$

completing the proof of (4.2).

Clearly P_k is orthonormal if $q_k^T q_k = 1$, so if $\|q_k\|_2 = 1$ for $k = 1, \ldots, n$, then P is orthonormal. Now suppose P is orthonormal, then $Pe_1 = P_1 e_1 = (0, q_1^T)^T$ must have length 1, so $\|q_1\|_2 = 1$ and P_1 and so $P_2 P_3 \cdots P_n$ is orthonormal. But then $P_2 P_3 \cdots P_n e_2 = P_2 e_2 = (0, q_2^T)^T$ must have length 1, and so on. Finally we see from (4.1) that the ith row of P_{11} is zero if and only if $q_i^T q_j = 0$ for $j = i+1, \ldots, n$, proving $P_{11} = 0$ if and only if $Q_1^T Q_1$ is diagonal. □

Since each of P (see (2.6) and (2.7)), $\bar{P}$ and $\tilde{P}$ (see (3.2)) has the structure of P in the theorem, P has the form (2.8), and

$$
\tilde{P} = \begin{pmatrix} \tilde{P}_{11} & (I - \tilde{P}_{11})\tilde{Q}_1^T \\ \tilde{Q}_1(I - \tilde{P}_{11}) & I - \tilde{Q}_1(I - \tilde{P}_{11})\tilde{Q}_1^T \end{pmatrix}, \tag{4.5}
$$

for some strictly upper triangular $\tilde{P}_{11}$, with $\bar{P}$ having a similar form. This shows how $\tilde{Q}_1$ loses orthogonality when $\tilde{P}_{11}$ is nonzero. Clearly P and $\tilde{P}$ are orthogonal matrices, so their first n columns form orthonormal sets. Since P_{11} is zero, Q_1 is clearly an $m \times n$ matrix of orthonormal columns, but all we can say about the size of $\tilde{P}_{11}$ is $\|\tilde{P}_{11}\|_2 \leq c_1 u\eta\kappa$, from (3.11). If κ is not very much greater than 1, then $\tilde{P}_{11}$ is small, and from (4.5) $\tilde{Q}_1$ has nearly orthonormal columns. For larger κ (4.5) shows how the columns of $\tilde{Q}_1$ can become less and less orthogonal, losing all likelihood of orthogonality when $c_1 u\eta\kappa \simeq 1$. Clearly column pivoting would be useful in maintaining orthogonality as long as possible, and in revealing the rank of rank deficient A. Since $\tilde{Q}_1$ is just $\bar{Q}_1$

with normalized columns, the same comments on orthogonality apply to $\bar{Q}_1$. We will bound these losses of orthogonality in the next section, and show how to avoid them after that.

5. Loss of Orthogonality in $\bar{Q}_1$ and $\tilde{Q}_1$ from MGS. Each column of $\tilde{Q}_1$ is just the correctly normalized column of the computed $\bar{Q}_1$ from MGS, whose columns already have norm almost one, so what we prove for $\tilde{Q}_1$ effectively holds for $\bar{Q}_1$. We saw from Theorem 4.1 that the first n columns $\tilde{P}^{(n)}$ of $\tilde{P}$ are orthonormal and

$$\tilde{P}^{(n)} = \begin{pmatrix} \tilde{P}_{11} \\ \tilde{Q}_1 - \tilde{Q}_1\tilde{P}_{11} \end{pmatrix} = \begin{pmatrix} \tilde{P}_{11} \\ \tilde{P}_{21} \end{pmatrix} I, \qquad \tilde{P}^{(n)T}\tilde{P}^{(n)} = I,$$

so an easy result is obtained by applying Lemma 3.1 with $R = I$, $A = \tilde{Q}_1$, $E_1 = \tilde{P}_{11}$ and $E_2 = -\tilde{Q}_1\tilde{P}_{11}$, showing there exist $\hat{Q}_1$ and E such that $\tilde{Q}_1 + E = \hat{Q}_1$ with $\hat{Q}_1^T\hat{Q}_1 = I$ and

$$\|E\|_2 = \|\tilde{Q}_1 - \hat{Q}_1\|_2 \leq (\|\tilde{P}_{11}\|_2 + \|\tilde{Q}_1\|_2)\|\tilde{P}_{11}\|_2.$$

But then $\|\tilde{Q}_1\|_2 \leq 1 + \|E\|_2$, giving

$$\|E\|_2 \leq \|\tilde{P}_{11}\|_2(1 + \|\tilde{P}_{11}\|_2)/(1 - \|\tilde{P}_{11}\|_2),$$

and a bound on the distance of $\tilde{Q}_1$ from an orthogonal matrix when $c_1 u\eta\kappa < 1$,

$$\|\tilde{Q}_1 - \hat{Q}_1\|_2 \leq c_1 u\eta\kappa \frac{1 + c_1 u\eta\kappa}{1 - c_1 u\eta\kappa}, \tag{5.1}$$

which for $c_1 u\eta\kappa \ll 1$ is effectively $c_1 u\eta\kappa$.

In order to bound the departure of $\tilde{Q}_1^T\tilde{Q}_1$ from the unit matrix we could use (5.1) directly, but a more revealing result follows by noting in (3.3) $E_1 = \tilde{P}_{11}\bar{R}$ is strictly upper triangular, since $\tilde{P}_{11}$ is so from Theorem 4.1. Thus

$$\tilde{P}^{(n)}\bar{R} = \begin{pmatrix} \tilde{P}_{11} \\ \tilde{Q}_1(I - \tilde{P}_{11}) \end{pmatrix} \bar{R} = \begin{pmatrix} E_1 \\ \tilde{Q}_1(\bar{R} - E_1) \end{pmatrix},$$

so that

$$\begin{aligned} (\bar{R} - E_1)^T\tilde{Q}_1^T\tilde{Q}_1(\bar{R} - E_1) &= \bar{R}^T\bar{R} - E_1^T E_1 \\ &= (\bar{R} - E_1)^T(\bar{R} - E_1) + (\bar{R} - E_1)^T E_1 + E_1^T(\bar{R} - E_1). \end{aligned}$$

Since $\bar{R}$ is nonsingular upper triangular, and E_1 is strictly upper triangular, $\bar{R} - E_1$ is nonsingular upper triangular, and

$$\tilde{Q}_1^T\tilde{Q}_1 = I + E_1(\bar{R} - E_1)^{-1} + (\bar{R} - E_1)^{-T}E_1^T, \tag{5.2}$$

with $E_1(\bar{R} - E_1)^{-1}$ the strictly upper triangular part of $\tilde{Q}_1^T\tilde{Q}_1$. This gives a clear picture of exactly how the loss of orthogonality depends on the computed $\bar{R}$. Thus from (3.3) and (3.8)–(3.10) if $(c + c_1)u\kappa < 1$ we obtain the bound

$$\|I - \tilde{Q}_1^T\tilde{Q}_1\|_2 \leq \frac{2c_1 u\kappa}{1 - (c + c_1)u\kappa}, \tag{5.3}$$

and a loss of orthogonality of this magnitude can often be observed in practice.

The bound (5.3) is of similar form to the bound (3.1) given in [3], but here we also derived the relation of $\tilde{Q}_1$ to the orthonormal matrix $\tilde{P}$, and described the relation between the loss of orthogonality in $\tilde{Q}_1$ and the deviation of $\tilde{P}$ from the ideal form of P. We also note here that if the first k columns of A in (3.3) have a small κ, then the first k columns of $\tilde{P}_{11}$ will be small, and the first k columns of $\tilde{Q}_1$ will be nearly orthonormal.

Our main purpose is not to show how $\tilde{Q}_1$ or $\bar{Q}_1$ may be improved. Instead the key point of this work is that although the computed $\bar{P}$ is very close to the exactly orthogonal $\tilde{P}$ in (3.3), the columns of $\bar{Q}_1$ need not be particularly orthonormal. Our thesis here is that as a result of this, it is usually inadvisable to use $\bar{Q}_1$ as our set of orthonormal vectors, but we *can* use $\bar{P}$ (as the theoretical product of the computed $\bar{P}_k = I - \bar{v}_k \bar{v}_k^T$, which is extremely close to $\tilde{P}$), to make use of the desired orthogonality, since we have all the necessary information in $\bar{Q}_1$, that is $\bar{v}_k^T = (-e_k^T, \bar{q}_k^T)$. Thus we can solve problems as accurately using MGS as using Householder or Givens QR factorizations if, instead of using the computed $\bar{Q}_1$ directly, we formulate the problems in terms of (2.5), see (3.3), and use the $\bar{q}_k$ to define $\bar{P}$. Ofcourse in most cases no block of $\bar{P}$ need actually be formed. We illustrate an important use of this idea in the next section, and discuss the efficiency of such an approach in Section 7.

6. Backward Stable Solution of the ASF using MGS. Björck [2] showed (1.5) is backward stable for the ASF (1.2) using the Householder QR factorization, but the same is not true when we use (1.6) with $\bar{R}$ and $\bar{Q}_1$ computed by MGS, see [5]. Here we use our new knowledge of MGS to produce a backward stable algorithm for the ASF based on $\bar{Q}_1$ and $\bar{R}$ from MGS. This new approach can be used to design good algorithms using MGS in general.

Our original ASF (1.2) is equivalent to the augmented system

$$\begin{pmatrix} I & 0 & 0 \\ 0 & I & A \\ 0 & A^T & 0 \end{pmatrix} \begin{pmatrix} w \\ x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ b \\ c \end{pmatrix}, \tag{6.1}$$

so applying Householder transformations as in (2.5) gives the augmented version of the method (1.5) is

$$z = R^{-T}c, \quad \begin{pmatrix} d \\ h \end{pmatrix} = P^T \begin{pmatrix} 0 \\ b \end{pmatrix}, \quad \begin{pmatrix} w \\ x \end{pmatrix} = P \begin{pmatrix} z \\ h \end{pmatrix}, \quad y = R^{-1}(d - z). \tag{6.2}$$

But as we saw in Section 2 we can use the q_k from MGS to produce $P_k = I - v_k v_k^T$, $v_k^T = (-e_k^T, q_k^T)$, and use $P^T = P_n \dots P_2 P_1$ in (6.2). We show in [5] that this algorithm is *strongly* stable (see [6]) for (6.1), and also strongly stable for (1.2).

We now show how to take advantage of the structure of the P_k, then we will summarize this numerically stable use of MGS for the ASF. To compute d and h in (6.2) note that $P^T = P_n \dots P_1$, and define

$$\begin{pmatrix} d^{(1)} \\ h^{(1)} \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix}, \qquad \begin{pmatrix} d^{(k+1)} \\ h^{(k+1)} \end{pmatrix} = P_k \dots P_1 \begin{pmatrix} 0 \\ b \end{pmatrix} = P_k \begin{pmatrix} d^{(k)} \\ h^{(k)} \end{pmatrix}.$$

Now using induction we see $d^{(k)}$ has all but its first $k-1$ elements zero, and

$$\begin{pmatrix} d^{(k+1)} \\ h^{k+1)} \end{pmatrix} = \begin{pmatrix} d^{(k)} \\ h^{(k)} \end{pmatrix} - \begin{pmatrix} -e_k \\ q_k \end{pmatrix} \begin{pmatrix} -e_k^T & q_k^T \end{pmatrix} \begin{pmatrix} d^{(k)} \\ h^{(k)} \end{pmatrix} = \begin{pmatrix} d^{(k)} + e_k(q_k^T h^{(k)}) \\ h^{(k)} - q_k(q_k^T h^{(k)}) \end{pmatrix},$$

giving the computation starting with $h^{(1)} := b$,

$$\textbf{for } k = 1, \ldots, n \ \ \textbf{do}\{\delta_k := q_k^T h^{(k)}; \quad h^{(k+1)} := h^{(k)} - q_k \delta_k\},$$

so $h = h^{(n+1)}, d = d^{(n+1)} = (\delta_1, \ldots, \delta_n)^T$. This costs $2mn$ flops (1 flop = 1 multiplication and 1 addition in floating point arithmetic), compared with the mn flops required to form $d = Q_1^T b$ in (1.6). The computation for d and h is exactly the same as the one that would arise if the n MGS steps in (2.1)–(2.3) had been applied to (A, b) instead of just A, so that h is theoretically the component of b orthogonal to the columns of A. Note that now d has elements $q_k^T h^{(k)}$ instead of $q_k^T b$ as would be the case in (1.6).

To compute x in (6.2), define

$$\begin{pmatrix} w^{(n)} \\ x^{(n)} \end{pmatrix} = \begin{pmatrix} z \\ h \end{pmatrix}, \qquad \begin{pmatrix} w^{(k-1)} \\ x^{(k-1)} \end{pmatrix} = P_k \ldots P_n \begin{pmatrix} z \\ h \end{pmatrix} = P_k \begin{pmatrix} w^{(k)} \\ x^{(k)} \end{pmatrix},$$

so that

$$\begin{pmatrix} w^{(k-1)} \\ x^{(k-1)} \end{pmatrix} = \begin{pmatrix} w^{(k)} \\ x^{(k)} \end{pmatrix} - \begin{pmatrix} -e_k \\ q_k \end{pmatrix} (\, -e_k^T w^{(k)} + q_k^T x^{(k)} \,),$$

which shows in this step only the kth element of $w^{(k)}$ is changed from $\zeta_k = e_k^T z$ to $\omega_k = q_k^T x^{(k)}$. This gives the computation starting with $x^{(n)} := h = h^{(n+1)}$,

$$\textbf{for } k = n, \ldots, 1 \ \ \textbf{do}\{\omega_k := q_k^T x^{(k)}; \quad x^{(k-1)} := x^{(k)} - q_k(\omega_k - \zeta_k)\},$$

so $x = x^{(0)}$, $w = (\omega_1, \ldots, \omega_n)^T$. This costs $2mn$ flops compared with mn flops for $x = b - Q_1(d - z)$ in (1.6). From (2.8) we see in theory (6.2) gives $x = Q_1 z + Q_2 Q_2^T h$ where $h = Q_2 Q_2^T b$, so $x = h + Q_1 z$. Note that $w = (\omega_1, \ldots, \omega_n)^T$ is ideally zero, see (6.1), but can be significant when $\kappa(A)$ is large. The computation of x here can be seen to reorthogonalize each $x^{(k)}$ against the corresponding q_k before adding on $q_k \zeta_k$ to give $x^{(k-1)}$. The complete algorithm is then:

Algorithm 6.1. **Backward Stable Algorithm for the ASF based on MGS**

1. *Carry out MGS on A to give $Q_1 = (q_1, \ldots, q_n)$ and R,*
2. *Solve $R^T z = c$ for $z = (\zeta_1, \ldots, \zeta_n)^T$,*
3. **for** $k = 1, \ldots, n$ **do**$\{\delta_k := q_k^T b; \quad b := b - q_k \delta_k\}$;
4. **for** $k = n, \ldots, 1$ **do**$\{\omega_k := q_k^T b; \quad b := b - q_k(\omega_k - \zeta_k)\}$; $\quad x := b$;
5. *Solve $Ry = d - z$ for y, where $d = (\delta_1, \ldots, \delta_n)^T$.*

A weakness in some other MGS-based algorithms is that the reorthogonalization in step 4 is not done. This is the case for the two algorithms denoted (3.4) and (3.6) in [1]. The first is equivalent to (1.6) and the second is the Huang algorithm [8] which, instead of steps 3 and 4, does (using our notation)

$$\textbf{for } k = 1, \ldots, n \ \ \textbf{do}\{\delta_k := q_k^T b; \quad b := b - q_k(\delta_k - \zeta_k)\}; \quad x := b;$$

The following implementation issues and specializations of the algorithm are fairly obvious. Steps 1, 2 and 3 can be combined, and there is a lot of parallelism inherent in these. When these are complete, steps 4 and 5 can be carried out independently.

For (1.3), step 5 can be omitted if the vector of Lagrange multipliers y is not needed, while for (1.4), step 4 can be omitted if the residual x is not needed.

If $b = 0$, corresponding to LUS, then $d = 0$ and step 3 will be omitted, and step 5 too if the Lagrange multipliers are not needed. If $c = 0$, corresponding to LLS, then $z = 0$ and step 2 will be omitted, and step 4 too if the LLS residual x is not needed. Then the algorithm is equivalent to the following variant of MGS:

$$(A, b) = (Q_1, h) \begin{pmatrix} R & d \\ 0 & 1 \end{pmatrix}, \qquad y = R^{-1}d, \tag{6.3}$$

where d is computed as part of MGS. This is the approach recommended for LLS in [3]. The work here is another way of proving the backward stability of this approach, and adds insight into why it works. For LUS however, the numerically stable algorithm made of steps 1,2 and 4 constitutes a new algorithm which is superior to the usual approach that omits the ω_k in step 4.

If A is square and nonsingular, (1.3) becomes the solution of $A^T x = c$, and x is independent of b, so if y is not wanted then b can be taken as zero in the algorithm, and steps 3 and 5 dropped. Similarly if A is square and nonsingular and $c = 0$ then (1.4) becomes $Ay = b$, and steps 2 and 4 can be dropped. This gives *two different* backward stable algorithms for solving nonsingular systems using MGS. Note that the first algorithm applies MGS to the *rows* of the matrix (here A^T), and is numerically invariant under row scalings. The second algorithm applies MGS to the *columns* of A, and is invariant under column scalings. Hence the first algorithm is to be preferred if the matrix is badly row scaled, the second if A is badly column scaled.

A square root free version of Algorithm 6.1 is obtained if we instead use the factorization (2.4) $A = Q_1' R'$, where R' is *unit* upper triangular:

ALGORITHM 6.2.

1. *Carry out MGS on A to give $Q_1' = (q_1', \ldots, q_n')$, R', and $D = diag(\gamma_1, \ldots, \gamma_n)$, where $\gamma_i = \|q_i'\|_2^2$.*
2. *Solve $(R')^T D z' = c$ for $z' = (\zeta_1', \ldots, \zeta_n')^T$.*
3. **for** $k = 1, \ldots, n$ **do**$\{\delta_k' := (q_k')^T b/\gamma_k; \quad b := b - q_k' \delta_k'\}$;
4. **for** $k = n, \ldots, 1$ **do**$\{\omega_k' := (q_k')^T b/\gamma_k; \quad b := b - q_k'(\omega_k' - \zeta_k')\}; \quad x := b;$
5. *Solve $R'y = d' - z'$ for y, where $d' = (\delta_1', \ldots, \delta_n')^T$.*

This section has not only shown how MGS can be used in a numerically stable way to solve the very useful linear system (1.2), along with its many specializations, but it has hopefully shown how MGS can be used more effectively in general.

7. Comparison of MGS and Householder Factorizations. There are four main approaches we need to compare:

1. MGS on A producing computed $\bar{R}$ and $\bar{Q}_1$, and using these.
2. MGS on A producing computed $\bar{R}$ and $\bar{Q}_1$, and using $\bar{R}$ and $\bar{P}_1, \ldots, \bar{P}_n$.
3. Householder transformations on $\begin{pmatrix} O_n \\ A \end{pmatrix}$ producing $\bar{R}$ and $\bar{P}_1, \ldots, \bar{P}_n$ and using these.
4. Householder transformations on A producing $\hat{R}$ and $\hat{P}_1, \ldots, \hat{P}_n$ say, and using these.

We call these *approaches* rather than algorithms, since each includes a reduction algorithm, plus a choice of tools to use in problems that use the reduction. We only consider the case of a single processor computer, and a dense matrix A.

Approaches 2 and 3 are numerically equivalent, but it is clearly more efficient for computer storage to use approach 2 via (2.1) and (2.2) than to use 3, even though we may *think* in terms of 3 to design algorithms which use the $\bar{P}_1, \ldots, \bar{P}_n$, these of course being "stored" as $\bar{q}_1, \ldots, \bar{q}_n$. Thus we would use the new approach 2 rather than 3 computationally, while being aware of both their properties theoretically.

The most usual case is where we wish to use the orthogonality computationally, but cannot rely on $\kappa(A)$ being small. Then the choice is between 2 and 4. For the initial QR factorization MGS requires mn^2 flops compared to $mn^2 - n^3/3$ for Householder. MGS also needs $n(n-1)/2$ more storage locations. Hence approach 4 has an advantage with respect to both storage and operation count for the initial factorization, although this is small when $m \gg n$.

If accurately orthogonal Q or Q_1 in (1.1) is required as an entity in itself, then since such orthogonal matrices are not immediately produced by 2 when $\kappa(A)$ is large, the obvious choice is 4, where Q (or Q_1) is available as the product (or part of it) of the $\hat{P}_k$. To produce Q_1 doubles the cost using 4. To produce an accurately orthogonal Q_1 with MGS in general, we apparently need to reorthogonalize. This also approximately doubles the factorization cost, and again the operation count is higher than for Householder.

For both approaches 2 and 4 we have shown backward stability in the usual normwise sense. In agreement with this both these approaches tend to give similar accuracy, although experience shows that MGS has a small edge here, in particular if the square root free version is used.

If the matrix A is not well row scaled then row interchanges may be needed in 4 to give accurate solutions for problem LLS, see [10]. In this context it is interesting to note that MGS is *numerically invariant* under row permutations of A as long as inner products are unaltered by the order of accumulation of terms. That is, if $\bar{Q}_1$ and $\bar{R}$ are the computed factors for A then $\Pi\bar{Q}_1$ and $\bar{R}$ are the computed factors of ΠA. This shows that 2 is more stable than 4 without row interchanges. However, if row interchanges are included in 4, this approach is more accurate for problems where the row norms of A vary widely. In approach 2 a second order error term $\sim O((wu)^2)$ appears, where w is the maximum ratio of row norms. This error term can be eliminated by reorthogonalization, which however increases the cost of MGS.

We finally mention that sometimes $\bar{R}$ is used alone to solve our problems, and then approaches 1 and 2 are identical. We will discuss this case in [5].

REFERENCES

[1] M. Arioli and A. Laratta, *Error analysis of algorithms for computing the projection of a point onto a linear manifold*, Linear Algebra Appl., 82 (1986), pp. 1–26.

[2] Å. Björck, *Iterative refinement of linear least squares solutions I*, BIT, 7 (1967), pp. 257–278.

[3] ———, *Solving linear least squares problems by Gram-Schmidt orthogonalization*, BIT, 7 (1967), pp. 1–21.

[4] ———, *Methods for sparse least squares problems*, in Sparse Matrix Computations, J. Bunch and D. J. Rose, eds., Academic Press, New York, 1976, pp. 177–199.

[5] Å. Björck and C. Paige, *Solution of augmented linear systems using orthogonal factorizations*, BIT, 34 (1994), pp. 1–24.

[6] J. R. Bunch, *The weak and strong stability of algorithms in numerical linear algebra*, Linear Algebra Appl., 88/89 (1987), pp. 49–66.

[7] G. H. GOLUB AND C. F. V. LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Maryland, 2nd ed., 1989.
[8] H. Y. HUANG, *A direct method for the general solution of a system of linear equations*, J. Optim. Theory Appl., 16 (1975), pp. 429–445.
[9] J. W. LONGLEY, *Least Squares Computations Using Orthogonalization Methods*, Marcel Dekker, Inc., New York, 1984.
[10] M. J. D. POWELL AND J. K. REID, *On applying Householder's method to linear least squares problems*, in Proceedings IFIP Congress, 1968, pp. 122–126.
[11] J. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.