Numerical Aspects of Gram-Schmidt Orthogonalization of Vectors

Axel Ruhe
Informationsbehandling
Universitetet
90187 Umeå, Sweden

Dedicated to Professor A. M. Ostrowski on his ninetieth birthday.

Submitted by W. Gautschi

ABSTRACT

Several variants of Gram-Schmidt orthogonalization are reviewed from a numerical point of view. It is shown that the classical and modified variants correspond to the Gauss-Jacobi and Gauss-Seidel iterations for linear systems. Further it is shown that orthogonalization with respect to elliptic norms and biorthogonalization can be formulated as orthogonalization by oblique projections.

1. INTRODUCTION

In diverse areas of applied mathematics use is made of orthogonal sets of vectors, or one needs to make a vector orthogonal to the linear span of a given set of vectors. The Gram-Schmidt algorithm is instrumental for these purposes.

The numerical behavior of this algorithm has been clarified by Björck [1], who showed that the modified variant, working rowwise on the vectors, was to be preferred to the classical variant. To be certain of orthogonality, reorthogonalization is sometimes necessary; Gragg and coworkers [3] have studied the classical variant and given a rigorous method to ascertain convergence.

The most common case when orthogonalization is needed is in the least-squares solution of overdetermined linear systems. There the Gram-Schmidt algorithm has met strong competition from methods based on Householder transformations, as introduced by Golub [7] and implemented by LINPACK [9].

52 Vanderbilt Ave., New York, NY 10017 0024-3795/83/$3.00

The Lanczos algorithm for eigenvalues and its descendants (the Arnoldi, nonsymmetric Lanczos, etc.) are all built up around the orthogonalization of a set of basis vectors. There, a Gram-Schmidt formulation is more natural, since the vectors are taken one at a time. Note however that even here Householder transformations can be used; see Golub, Underwood, and Wilkinson [8].

In the present contribution we will concentrate on one substep of the Gram-Schmidt algorithm, that is, the orthogonalization of one vector towards the linear span of a given set of vectors. Usually that set has been assumed to be orthonormalized, but we will study what happens when that is not fully the case, as e.g. when rounding errors are present, or when an incomplete algorithm has been used to compute the set. See e.g. the IOM method of Saad [12]. In Section 2, we show that there the classical and modified variants of the Gram-Schmidt algorithm correspond to the Gauss-Jacobi and Gauss-Seidel iterations for solving the system of normal equations. We compare this with the findings by Björck, and indicate how more powerful iterations can be used.

We conclude Section 3 by studying more general notions of orthogonality. We formulate orthogonalization by oblique projections, and show that both orthogonalization with respect to elliptic norms and biorthogonalization can be formulated this way. These variants are especially interesting in conjunction with nonsymmetric or generalized eigenproblems.

A word about notation. Capital letters are used for matrices, and lowercase for vectors. A_k is a matrix with k columns. a^k denotes the kth vector in a sequence of iterates, while a_i^k is the ith element of that vector.

2. ORTHOGONALIZATION FORMULATED AS ITERATION

Let us now formulate, in an algorithmic notation, what is done in the classical and modified Gram-Schmidt algorithms, and then show how this fits into the formulations of the Gauss Jacobi and Gauss Seidel iterative methods for linear systems. We concentrate our attention on one step, when we already have p allegedly orthonormal vectors forming an $n \times p$ matrix Q, and a vector a, which is to be orthogonalized eventually to form the $p+1$st column of Q. Note that when formulating the modified algorithm, we put off all operations on the vector a till this step, as e.g. Gander [6] or Schwarz and Rutishauser [15] have suggested, instead of the rowwise ordering used e.g. by Björck [1]. There is no numerical difference between these two variants of the modified algorithm, since it is the same operations performed in a different sequence.

ALGORITHM CGSA (Classical Gram-Schmidt algorithm).

1. Start $a^0 = a$, $r^0 = 0$
2. For $k = 1,2,\ldots$ do
   1. $s^{k-1} = Q^T a^{k-1}$
   2. $r^k = r^{k-1} + s^{k-1}$
   3. $a^k = a^{k-1} - Qs^{k-1}$
3. $q_{p+1} = a^k/(r_{p+1,p+1} := \|a^k\|)$

Note here that each iteration of step 2 is a reorthogonalization. CGSA as it is usually formulated corresponds to the first step ($k = 1$) of this process. The slightly awkward notation in step 3 means that the final vector a^k is normalized to form the next column q_{p+1} of Q, and the normalization coefficient enters the R matrix below the vector r:

$$R_{p+1} = \begin{bmatrix} R_p & r \\ 0 & r_{p+1,p+1} \end{bmatrix}, \qquad Q_{p+1} = \begin{bmatrix} Q & q_{p+1} \end{bmatrix}.$$

In Algorithm MGSA, each iteration in step 2 is divided into p minor steps, each computing one element of the increment vector s:

Algorithm MGSA (Modified Gram-Schmidt algorithm). Replace steps 2.1–3 by

2.1. For $i = 1,\ldots,p$ do
   1. $s_i^{k-1} = q_i^T a^{k-1,i-1}$
   2. $a^{k-1,i} = a^{k-1,i-1} - q_i s_i^{k-1}$
   3. $r_i^k = r_i^{k-1} + s_i^{k-1}$

2.2. $a^k = a^{k-1,p}$

We first note that for each iteration in step 2 of CGSA one has

$$a^k = a - Qr^k;$$

it is the loop invariant, to use terminology from computer science, irrespective of the orthogonality of Q. Likewise

$$a^{k-1,i} = a - Qr^{k-1,i}$$

for each iteration of step 2.1 of MGSA, where $r^{k-1,i} = (r_1^k,\ldots,r_{i-1}^k, r_i^{k-1},\ldots, r_p^{k-1})^T$ is an obvious fashion.

We can use these loop invariants to see what happens during the iteration. In fact we solve an overdetermined linear system (linear least squares problem), with Q as its matrix and a its right hand side for r, seeking the point where a^k, the residual, is orthogonal to the span of Q.

For CGSA we see that

$$\begin{aligned} r^k &= r^{k-1} + Q^T a^{k-1} \\ &= r^{k-1} + Q^T(a - Qr^{k-1}) \\ &= r^{k-1} + Q^T a - Q^T Q r^{k-1}, \end{aligned}$$

which is the Gauss-Jacobi iteration applied to the normal equations

$$Q^T Q r = Q^T a. \tag{2.1}$$

For MGSA

$$\begin{aligned} r_i^k &= r_i^{k-1} + q_i^T a^{k-1,i-1} \\ &= r_i^{k-1} + q_i^T(a - Qr^{k-1,i-1}) \\ &= r_i^{k-1} + q_i^T a - \sum_{j=1}^{i-1} q_i^T q_j r_j^k - \sum_{j=i}^{p} q_i^T q_j r_j^{k-1}. \end{aligned}$$

Setting

$$Q^T Q = I + L + L^T,$$

with a strictly lower triangular L, we get

$$\begin{aligned} r^k &= r^{k-1} + Q^T a - Lr^k - (I + L^T) r^{k-1}, \\ r^k &= (I + L)^{-1} Q^T a - (I + L)^{-1} L^T r^{k-1}, \end{aligned}$$

which we recognize as the Gauss-Seidel iteration for solving (2.1).

If the matrix $Q^T Q$ has property A (see [16]), i.e. if the elements of L can be confined to a nondiagonal block, the spectral radius of the Gauss-Seidel

iteration matrix is the square root of that of the Gauss-Jacobi. A similar relation holds when L has small elements, since

$$(I+L)^{-1}L^T = (I - L + L^2 - L^3 + \cdots)L^T$$
$$= L^T - LL^T + O(\|L\|^3)$$

and L^T is nilpotent.

We are not going into a detailed analysis of rounding errors in this contribution, but it might anyhow be interesting to see how our finding relates to the analysis by Björck [1], where it is shown that MGSA can be performed without reorthogonalization, provided that we start with linearly independent vectors. Assuming that rounding errors are the sole cause of the lack of orthogonality of Q, we now ask whether Algorithm CGSA or MGSA hits convergence already at the first iteration $k = 1$.

To see more clearly what happens, let us look at the example of Läuchli discussed by Björck [1]:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix},$$

where ϵ is such that $fl(1+\epsilon^2) = 1$. Taking $p = 2$, we have

$$Q = \begin{bmatrix} 1 & 0 \\ \epsilon & -1/\sqrt{2} \\ 0 & 1/\sqrt{2} \\ 0 & 0 \end{bmatrix}, \qquad a = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \epsilon \end{bmatrix}.$$

One step of CGSA yields

$$s^0 = Q^T a = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad r^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad a^1 = \begin{bmatrix} 0 \\ -\epsilon \\ 0 \\ \epsilon \end{bmatrix},$$

$$\text{with} \quad \frac{Q^T a^1}{\|a^1\|} = \begin{bmatrix} \epsilon/\sqrt{2} \\ \frac{1}{2} \end{bmatrix},$$

which shows that we have not yet achieved the goal, but

$$s^1 = \begin{bmatrix} -\epsilon^2 \\ \epsilon/\sqrt{2} \end{bmatrix}, \qquad r^1 = \begin{bmatrix} 1 \\ \epsilon/\sqrt{2} \end{bmatrix},$$

$$a^2 = \begin{bmatrix} \epsilon^2 \\ -\epsilon/2 \\ -\epsilon/2 \\ \epsilon \end{bmatrix},$$

$$\frac{Q^T a^2}{\|a^2\|} = \begin{bmatrix} \epsilon/\sqrt{3} \\ 0 \end{bmatrix},$$

which is as good as we can ask for. Following MGSA, we get

$$s_1 = q_1^T a,$$

$$a^{0,1} = \begin{bmatrix} 0 \\ -\epsilon \\ 0 \\ \epsilon \end{bmatrix}$$

$$s_2 = q_2^T a^{0,1} = \epsilon/\sqrt{2},$$

$$a^{0,2} = \begin{bmatrix} 0 \\ -\epsilon/2 \\ -\epsilon/2 \\ \epsilon \end{bmatrix},$$

hitting the result already after the first iteration.

In this case, the iteration matrices of the Gauss-Jacobi and Gauss-Seidel algorithms are

$$\begin{bmatrix} 0 & \epsilon/\sqrt{2} \\ \epsilon/\sqrt{2} & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & \epsilon/\sqrt{2} \\ 0 & -\epsilon^2/2 \end{bmatrix},$$

respectively, and we see that is is the near-nilpotency of the Gauss-Seidel matrix together with the fact that the right hand side,

$$Q^T a = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

is graded with all the weight in the beginning that explains the success of MGSA. We also see that the order of the minor steps in MGSA is important; reversing it would yield the same iteration matrix, but a starting vector that is not graded in the appropriate way. We then need to perform a reorthogonalization.

When the vectors of Q are less than perfectly orthogonalized, we get slower and slower convergence of the reorthogonalization process, but as long as the columns of Q remain linearly independent, we will get convergence eventually, since the Gauss-Jacobi and Gauss-Seidel algorithms converge for all positive definite matrices. The convergence criterion used to stop the iteration is then critical. Gragg et al. [3] have from other assumptions arrived at the criterion

$$\|a^k\| > \eta \|a^{k-1}\| \tag{2.2}$$

with η a moderately sized number, e.g. $1/\sqrt{2}$ (they use CGSA). In our case we ask two questions: first, is (2.2) an appropriate criterion to guarantee orthogonality, and second, can we promise that (2.2) will ever be satisfied, if we start out with a nonorthogonal Q?

After an intricate navigation through the labyrinth of proofs in [3], we find that when Q is far from orthogonal, it is necessary to choose η close to one in order to get an affirmative answer to the first question. Moreover, to be applicable, (2.2) supposes certain further, so-called τ, conditions on the orthogonality, which rule out cases far from orthogonality.

Sidestepping the issue, we note that the deviation from orthogonality of a^k is just the residual of (2.1):

$$Q^T a^k = s^k,$$

and we stop whenever $\|s^k\|$ is small enough compared to $\|a^k\|$. Now, however, we have no guarantee that the second question gets a positive answer; the convergence guarantee in [3] is built on the supposition that rounding errors eventually "take over" if a is nearly in the span of Q.

In some cases, we need to orthogonalize against vectors that we know are not orthogonal. Consider e.g. the incomplete orthogonalization method of Saad [12]. Then more than a few iterations will be needed for convergence of the reorthogonalization, and it is advisable to use a more powerful algorithm. The preferred choice, from the author's point of view, is the LSQR algorithm by Paige and Saunders [10], applied to the system

$$\min_r \|Qr - a\|_2.$$

It will considerably decrease the number of reorthogonalizations necessary.

3. OBLIQUE PROJECTIONS

Let us now turn to the oblique case, when we make a orthogonal to a subspace spanned by W_k by subtracting a vector in another subspace, spanned by V_j; neither the W_k nor the V_j are assumed to be orthonormal for the moment. Such oblique projections are discussed by Saad [14] and Ruhe [11].

We see that now

$$a^* = a - V_j r,$$

$$W_k^T a^* = W_k^T a - W_k^T V_j r$$

showing that for $W_k^T a^* = 0$ we now get the system

$$W_k^T V_j r = W_k^T a \tag{3.1}$$

corresponding to (2.1), and that

$$a^* = \left[I - V_j \left(W_k^T V_j \right)^- W_k^T \right] a, \tag{3.2}$$

an oblique projection ($^-$ denotes left inverse). It is evident that the solvability of the system (3.1) is a condition for an orthogonalization to be possible. Typically, assume that $k \leqslant j$, and W_k are linearly independent columns. Then the QR factorization

$$V_j^T W_k = QR$$

exists, and one gets

$$a^* = a - V_j c,$$

$$c = QR^{-T} W_k^T a.$$

The classical Gram-Schmidt algorithm now computes a^* this way, by first forming $W_k^T a$, then making a forward substitution to get $R^{-T} W_k^T a$, then multiplying by Q, and last forming a linear combination of V_j. We also note that Q and R can be formed one column at a time for increasing k.

The notable fact now is that the modified Gram-Schmidt algorithm does not need a forward substitution and makes no explicit use of the nondiagonal elements of R. Then a is successively orthogonalized to $w_1, w_2, \ldots$ until w_k. If $w_1^T a = \cdots = w_{i-1}^T a = 0$, then $W_i^T a = (0, \ldots, w_i^T a)^T$ and $R^{-T} W_i^T a = (0, \ldots, r_{ii}^{-1} w_i^T a)^T$, and finally

$$V_j Q R^{-T} W_i^T a = V_j q_i \cdot r_{ii}^{-1} w_i^T a,$$

only using the ith column of Q and W_k. This is a considerable simplification, compared to the straightforward CGSA approach, and is also likely to be more stable numerically.

In algorithmic notation we get:

ALGORITHM OMGS (Oblique modified Gram-Schmidt algorithm).

1. Start $a^{(0)} = a$
2. for $i := 1$ to k do
   1. $q_i := V_j^T w_i$
   2. for $l := 1$ to $i - 1$ do
      1. $q_i := q_i - q_l \times (r_{li} := q_l^T q_i)$
   3. $q_i := q_i / (r_{ii} := \|q_i\|)$
   4. $a^{(i)} := a^{(i-1)} - V_j q_i r_{ii}^{-1} w_i^T a^{(i-1)}$
3. Now $a^{(k)}$ satisfies $W_k^T a^{(k)} = 0$

Breakdown may occur in step 2.3 if $W_i^T V_j$ does not have linearly independent rows.

In [11] the OMGS algorithm is used in conjunction with the Arnoldi method for computing an orthogonal basis $V_j = (v_1, v_2, \ldots, v_j)$ of the right Krylov subspace of a nonsymmetric matrix,

$$K_R^j(B, v_1) = \{v_1, Bv_1, \ldots, B^{j-1} v_1\},$$

with a residual, $v_{j+1} = v_{j+1} - V_j r$, which is orthogonal to the left Krylov subspace,

$$K_L^k(B, w_1^H) = \{w_1^H, w_1^H B, \ldots, w_1^H B^{k-1}\}.$$

Another class of problems which can be formulated as orthogonalization by oblique projections is orthogonalization in elliptic norms. Assume that we want to make

$$V_i^T M a = 0,$$

by subtracting a vector from V_j. Then

$$a^* = \left[I - V_j \left(V_j^T M V_j \right)^{-1} V_j^T M \right] a,$$

which is the same as (3.2) if we choose $W_k = MV_j$. Note that now the matrix of (3.1) is symmetric.

If V_j has M orthonormal columns, then a simple modified Gram-Schmidt orthogonalization is possible. Such a procedure is used, e.g., in [5] for reorthogonalization of basis vectors in a Lanczos program, and an analysis of rounding errors which takes the effect of the condition of M into account is forthcoming [4].

The nonsymmetric Lanczos algorithm computes two bases Q_j and P_j^H of the right and left Krylov subspaces. They are kept biorthogonal,

$$P_j^H Q_j = I_j,$$

and a reorthogonalization is done by (3.2), now also with a nearly diagonal but nonsymmetric system (3.1). See the analysis by Saad [13].

Some of these results dawned upon the author during enlightening discussions with W. B. Gragg. The work was written up while the author enjoyed the kind hospitality and stimulating environment provided by Gene Golub and his group at Stanford. Supported in part by U.S. Department of Energy contract DE-AT-03-76ER71030 and a travel grant by Stiftelson J. C. Kempes Minne at Umeå University.

REFERENCES

1 Å. Björck, Solving linear least squares problems by Gram-Schmidt orthogonalization, *BIT* 7:1–21 (1967).
2 Å. Björck, Iterative refinement of linear least squares solutions, part 1, *BIT* 7:257–278 (1967); part 2, 8:8–30 (1968).
3 J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization, *Math. Comp.* 30:772–795 (1976).
4 T. Ericsson, An analysis of orthogonalization in elliptic norms, to appear.
5 T. Ericsson, and A. Ruhe, The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems, *Math. Comp.* 35:1251–1268 (1980).
6 W. Gander, Algorithms for the QR decomposition, Res. Rep. 80-02, Sem. Angew. Math., ETH, Zurich, 1980.

7 G. H. Golub, Numerical methods for solving linear least squares problems, *Numer. Math.* 7:206–216 (1965).

8 G. H. Golub, R. Underwood, and J. H. Wilkinson, The Lanczos algorithm for the symmetric $Ax = \lambda Bx$ problem, Tech. Rep. STAN-CS-72-270, Computer Science Dept., Stanford Univ., 1972.

9 J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart, LINPACK *Users Guide*, SIAM, Philadelphia, 1979.

10 C. C. Paige, and M. Saunders, LSQR, an algorithm for sparse linear equations and sparse least-squares, *ACM Trans. Math. Software* 8:45–71 (1982); Alg 583 LSQR, *ibid.* 8:195–201 (1982).

11 A. Ruhe, The two-sided Arnoldi algorithm for nonsymmetric eigenvalue problems, in *Proceedings of the Conference on Matrix Pencils*, Piteå, 1982, Springer LNM, to appear.

12 Y. Saad, Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices, *Linear Algebra Appl.* 34:269–295 (1980).

13 Y. Saad, The Lanczos biorthogonalization algorithm and other oblique projection methods for solving large unsymmetric systems, *SIAM J. Numer. Anal.* 19:485–506 (1982).

14 Y. Saad, Projection methods for solving large sparse eigenvalue problems, in *Proceedings of the Conference on Matrix Pencils*, Piteå, 1982, Springer LNM, to appear.

15 H. R. Schwarz, H. Rutishauser, and E. Stiefel, *Matrizen-Numerik*, Teubner, 1968.

16 R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1962.

Received W. Cautschi