LAPACK working note 51
Qualitative Properties of the Conjugate Gradient and Lanczos Methods in a Matrix Framework*

Victor Eijkhout
University of Tennessee
Department of Computer Science
107 Ayres Hall, Knoxville, TN 37996-1301
`eijkhout@cs.utk.edu`

January 17, 1995

Abstract

This paper presents the conjugate gradient and Lanczos methods in a matrix framework, focusing mostly on orthogonality properties of the various vector sequences generated. Various aspects of the methods, such as choice of inner product, preconditioning, and relations to other iterative methods will be considered. Minimization properties of the methods and the fact that they can compute successive approximations to the solution of a linear system will be proved as corollary.

Contents

*. This work was supported by DARPA under contract number DAAL03-91-C-0047

1 Introduction

Many articles have already been written about the conjugate gradient method and the Lanczos algorithm. Some of these focus on preconditioners, some treat different variants of the basic method, and others derive error or convergence speed properties of the methods (for the history of the conjugate gradient and Lanczos methods and an extensive bibliography we refer to [9]). In this article we will give a unified presentation of the conjugate gradient method and the Lanczos algorithm, and derive properties regarding orthogonality and equivalence of various formulations. We will stress showing what the minimal assumptions are for various properties.

In general we will only be concerned with qualitative properties of the methods, leaving all quantitative results, such as convergence speed, aside. The conjugate gradient and Lanczos methods will here be derived as orthogonalization methods for Krylov sequences, and minimization properties will be derived from this orthogonality. Also, the fact that such methods can be used for the iterative solution of linear systems will be presented as a corollary rather than as a starting point of the discussion.

Additionally, the presentation here will differ from most others in the literature (with the exception of [12]) in that it is given in terms of matrices instead of vector sequences. Although an occasional part of the discussion may feel somewhat forced by this, in general the presentation is very concise.

2 Tools: sequences as matrices

Most of the analysis in this paper will talk about matrices, instead of vector sequences. The matrix corresponding to a sequence has the elements of that sequence as its columns. For instance, a matrix X will have the vectors x_1, x_2, ... as its columns. Subscripted uppercase letters will indicate initial parts of sequences: X_n is the matrix with columns $x_1 \ldots x_n$. Doubly subscripted lowercase characters denote elements of matrices: h_{ij} is the i-th element of h_j or the (i,j) element of H.

Combinations of the elements of a sequence can be expressed by upper triangular matrices: if X and Y are sequences and U is upper triangular, then $X = YU$ expresses the fact $x_i = \sum_{k=1}^{i} y_k u_{ki}$.

In addition to the identity matrix I, we will often need the unit lower subdiagonal matrix

$$J = (\delta_{i,j+1}).$$

Right multiplication by J shifts a matrix left one column, so $X = YJ$ corresponds to the relation $x_i = y_{i+1}$ between sequences. An update relation

$$x_{i+1} - x_i = y_i$$

can be expressed as $X(J - I) = Y$.

Lemma 1 *An update relation* $x_{i+1} - x_i = \sum_{k=1}^{i} y_k u_{ki}$, *or*

$$X(J - I) = YU$$

with U *upper triangular can be rewritten as* $x_{i+1} = x_1 + \sum_{k=1}^{i} y_i v_{ki}$, *or*

$$X(J - E_1) = YV$$

with V *upper triangular and* $E_1 = (\delta_{i1})$.

Proof: the sequences can be rewritten by algebraic manipulation. Here is a proof using matrix analysis. Let E_N be the square $N \times N$ matrix (δ_{iN}), then $(J - E_1)^{-1} = J^t - E_N$. Suppose $X(J-I) = YU$, then using $J - E_1 = (J-I)(I-J^t)^{-1}$ we get

$$\begin{aligned} X(J - E_1) &= YU(J - I)(J - E_1)^{-1} \\ &= YU(I - J^t)^{-1}. \end{aligned}$$

Again $V = U(I - J^t)^{-1}$ is an upper triangular matrix. •

3 Krylov sequences and Hessenberg matrices

In this section we will look at Krylov sequences, that is, vector sequences derived from successive application of a matrix product. Linear combinations of subsequences of a Krylov sequence can be associated with an upper Hessenberg matrix and with matrix polynomials in the generating matrix of the sequence. It is important to note that all of the properties derived here are independent of any concept of orthogonality.

Consider the Krylov sequence $x_{i+1} = Ax_i$. This sequence can be written in matrix form as

$$AX = XJ. \tag{1}$$

Identities such as $x_{i+2} = A^2 x_i$ follow from

$$A^k X = A^{k-1} XJ = \cdots = XJ^k.$$

Taking linear combinations of Krylov vectors gives rise to Hessenberg matrices.

Lemma 2 *If $AR = RH$ and $r_1 \parallel x_1$ (that is, $r_1 = \alpha x_1$ for some $\alpha \neq 0$), then H is an irreducible upper Hessenberg matrix iff $R = XU$ with X the Krylov sequence (1) and U a nonsingular upper triangular matrix satisfying $H = U^{-1} JU$.*

Proof: If U is a nonsingular upper triangular matrix and $R = XU$, then

$$AR = RU^{-1} JU$$

where $U^{-1}JU$ is of irreducible upper Hessenberg shape, so the r_i vectors satisfy a recurrence

$$h_{i+1,i} r_{i+1} = A r_i - \sum_{j=1}^{i} h_{ji} r_j.$$

Conversely, if H is an upper Hessenberg matrix and $AR = RH$, then an upper triangular matrix U can be determined such that $R = XU$, namely U has to satisfy

$$UH = JU$$

and this can be solved recursively. For instance, noting that the first row of JU is zero (and picking u_{11} arbitrarily):

$$\begin{aligned} u_{11}h_{11} + u_{12}h_{21} = 0 \Rightarrow u_{12} = \dots \\ u_{11}h_{12} + u_{12}h_{22} + u_{13}h_{32} = 0 \Rightarrow u_{13} = \dots \end{aligned}$$

Then for the second row

$$\begin{aligned} u_{11} = u_{22}h_{21} \Rightarrow u_{22} = \dots \\ u_{12} = u_{22}h_{22} + u_{23}h_{32} \Rightarrow u_{23} = \dots \end{aligned}$$

We see that U can be solved if all $h_{i+1,i} \neq 0$, that is, if H is irreducible. Now

$$ARU^{-1} = RU^{-1}(UHU^{-1}) = RU^{-1}J$$

so if $r_1 = \alpha x_1$, then $R = X(\alpha U)$. •

For future reference we prove a lemma on Hessenberg matrices.

Lemma 3 *Let H be a Hessenberg matrix that can be factored without pivoting, then the column sums of H are zero iff the factorization is $H = (I - J)V$ with V upper triangular.*

Proof: Let $e^t = (1, \dots)$. In general, H can be factored as $H = (I - L)V$. Now $e^t(I-L) = (1-\ell_{21}, 1-\ell_{32}, \dots)$, so if the column sums are zero, $\ell \equiv 1$. Conversely, if $H = (I - J)V$, then from $e^t(I - J) = 0^t$ the column sums of H are zero. •

A combination of Krylov vectors can be expressed as a matrix polynomial times the initial vector.

Theorem 1 *If $AX = XJ$ and $r_1 \parallel x_1$, then $R = XU$ with U upper triangular, if and only if*

$$r_{n+1} = P_n(A)x_1$$

where P_n is an n-th degree polynomial with coefficients in the $n+1$-st column of U.

Proof: Suppose $R = XU$, and consider the splitting of the upper triangular matrix U as

$$U = \sum_{k=0}^{N} (J^t)^k U^{(k)}, \qquad U^{(k)} = \mathrm{diag}(0, \ldots, 0, u_{1k}, \ldots, u_{N-k+1,N})$$

where $U^{(k)}$ has k initial zeros. For the columns of R (which are the vectors r_k) we find

$$\begin{aligned} r_{*,n+1} &= \sum_{k=0}^{n} x_{*,n+1-k} (J^t)^k U^{(k)}_{n+1,n+1} \\ &= \sum_{k=0}^{n} x_{*,n+1-k} J^{n-k} (J^t)^{n-k} (J^t)^k u_{n+1-k,n+1} \\ &= \left\{ \sum_{k=0}^{n} A^{n-k} x_{*,1} u_{n+1-k,n+1} \right\} (J^t)^n \end{aligned}$$

where we use $A^k X = X J^k$ and the fact that $J^k (J^t)^k$ is the identity matrix apart from the first k diagonal elements, which are zero. For the inverse assertion, read the above proof backwards. •

In the context of iterative methods for linear systems, where these r_i vectors will turn out to be residuals, the measure of success of the method depends on how closely the polynomials approximate the zero function. Results regarding this can be found for instance in [1], [22]; in this paper we will not pay any attention to this matter.

The polynomials derived above satisfy a simply recurrence.

Lemma 4 *Denoting the polynomials from the previous lemma ϕ_n and defining $H = U^{-1} J U$, the polynomials satisfy*

$$h_{n+1n} \phi_{n+1}(t) = (t - h_{nn}) \phi_n(t) - \sum_{k=1}^{n-1} h_{kn} \phi_k(t)$$

Proof: From the relation $UH = JU$ relating the upper Hessenberg matrix and the upper triangular matrix corresponding to certain spans of Krylov spaces we see that U can be solved one column at a time. Suppose the n-th column u_{*n} has been solved, then

$$u_{in+1}h_{n+1n} = u_{i-1n} - \sum_{k=1}^{n} u_{ik}h_{kn} \qquad \text{for } i = 1, \ldots, n+1. \tag{2}$$

This relation implies the recurrence for the polynomials ϕ_n. •

4 Orthogonalization

The conjugate gradient algorithm and the Lanczos method can be derived by imposing various orthogonality conditions on the sequence R derived in the previous section. Equivalently, they correspond to a certain construction for the Hessenberg matrix H.

4.1 Simple orthogonalization: the Galerkin condition

A sequence R defined by $AR = RH$ can be chosen semi-orthogonal to another sequence S by suitable construction of the Hessenberg matrix H.

Algorithm 1 *Let r_1 and any sequence S be given, and suppose inductively that $S_n^t R_n$ is nonsingular. Solve the $n \times 1$ vector h_n from*

$$S_n^t A r_n = S_n^t R_n h_n. \tag{3}$$

Then pick a value for $h_{n+1,n}$, for instance 1 or $1/\|Ar_n - R_n h_n\|$, and define r_{n+1} from

$$r_{n+1}h_{n+1,n} = Ar_n - R_n h_n. \tag{4}$$

(In section 6.1 we shall see factors that dictate certain choices of $h_{i+i,i}$.)

Lemma 5 *Algorithm 1 constructs a sequence R that is semi-orthogonal to S in the sense that $s_i^t r_j = 0$ for $j > i$; the algorithm is well-defined if R_n is not an invariant subspace of A, and if the breakdown condition $s_{n+1}^t r_{n+1} = 0$ does not occur.*

Proof: Suppose inductively that $S_n^t R_n$ is lower triangular and nonsingular. Unless R_n is an invariant subspace of A, the vector r_{n+1} will be nonnull. Furthermore, $S_n^t r_{n+1} = 0$ so, $S_{n+1}^t R_{n+1}$ is again a lower triangular matrix. It is nonsingular depending on whether $s_{n+1}^t r_{n+1}$ is zero. •

The condition that r_i is orthogonal to the $i-1$-dimensional space spanned by S_{i-1} is often called a 'Galerkin condition', or perhaps more correctly, a 'Petrov-Galerkin condition' [19].

The classical conjugate gradient method stems from the choice $S = R$.

Lemma 6 *The conjugate gradient method corresponds to a transformation of A to upper Hessenberg form. If A is symmetric this is a tridiagonalization; if A is positive definite the breakdown condition will not occur.*

Proof: If $S = R$, then the lower triangular matrix $S^t R$ is diagonal, and $S^t AR = R^t AR = R^t RH$ is a transformation of A to Hessenberg form. If A is symmetric, then $R^t RH = H^t R^t R$, so H is of both upper and lower Hessenberg form, hence tridiagonal. •

Lemma 7 *Normalizing the r_i vectors in the conjugate gradient method for a symmetric matrix A leads to a symmetric Hessenberg matrix H.*

Proof: This follows immediately from $R^t AR = R^t RH = H$. •

The question whether for unsymmetric matrices A the upper Hessenberg matrix can take on a banded form with a small bandwidth is of practical importance, since such a form limits the length of the recurrence for the r_i vectors in (4). This question was answered largely negatively by [25] and more generally by [6]: the conjugate gradient algorithm gives a tridiagonal matrix if the spectrum of A lies on a line in the complex plane; for other matrices the bandwidth is a large fraction of the matrix size.

4.2 Bi-orthogonalization

If the sequence $\{s_n\}$ is based on a Krylov recurrence with A^t, that is,

$$A^tS = SK \qquad \text{or} \qquad K^tS^t = S^tA$$

with K of upper Hessenberg shape, the columns of K can be solved in a similar manner to those of H. Assume $S_n^tR_n$ is diagonal and non-singular. Solve k_n from

$$s_n^tAR_n = k_n^tS_n^tR_n$$

so that s_{n+1} defined from

$$k_{n+1,n}s_{n+1} = A^ts_n - S_nk_n$$

satisfies $s_{n+1}^tR_n = 0$, so that $S_{n+1}^tR_{n+1}$ is again diagonal, but maybe singular. Obviously, if $r_1 = s_1$ and A is symmetric, then $S_n = R_n$ for all n, and $S_{n+1}^tR_{n+1}$ is singular iff $r_{n+1} = 0$. Choosing r_1 and s_1 independently corresponds to the Lanczos method.

Theorem 2 *The Lanczos method generates Hessenberg matrices H and K that are of tridiagonal form; under some normalization of the R and S sequences they are equal. The matrix S^tAR is tridiagonal and it is symmetric (independent of the symmetry of A) if $H = K$.*

Proof: We have that S^tR is a diagonal matrix, so for S^tAR we find

$$S^tAR = S^tRH = K^tS^tR$$

that is, it is both equal to an upper and to a lower Hessenberg matrix. Therefore it is tridiagonal. Note that, unlike above, this conclusion does not depend on symmetry, or equality of the sequences R and S.

Inspecting the elements of S^tAR, we find from the (n,n) position that

$$s_n^tAr_n = s_n^tr_nh_{nn} = k_{nn}s_n^tr_n \Rightarrow h_{nn} = k_{nn} = \frac{s_n^tAr_n}{s_n^tr_n}$$

and similarly

$$(n+1,n): \quad k_{nn+1}s_n^tr_n = s_{n+1}^tr_{n+1}h_{n+1n}$$

and

$$(n,n+1): \quad k_{n+1n}s_{n+1}^tr_{n+1} = s_n^tr_nh_{nn+1}.$$

The matrices H and K are related by

$$H = \Omega^{-1}K^t\Omega$$

where Ω is the diagonal matrix S^tR. We see again that $h_{nn} = k_{nn}$ for any scaling of R and S; if we choose $h_{n+1n} = k_{n+1n}$, then

$$h_{nn+1} = k_{nn+1} = \frac{s_{n+1}^t r_{n+1}}{s_n^t r_n} h_{n+1n}$$

so $H = K$. In general, any scaling of the R or S sequence that makes $\Omega = I$ causes H and K to be equal.

If H and K are equal, we can combine the equalities $S^t AR = K^t S^t R = H^t S^t R$ and $(S^t AR)^t = (S^t RH)^t = H^t R^t S = H^t S^t R$ to find that $S^t AR$ is symmetric. •

4.3 Different choice of inner product and the symmetrizable case

If we don't orthogonalize R and S directly, but make $S^t MR$ diagonal for some matrix M (if this matrix is symmetric positive definite it induces an inner product, however this fact will not be used in the following derivation), the above formulas alter slightly. First of all, if we still generate R from $AR = RH$, then h_n is to be solved from

$$S_n^t MAr_n = S_n^t MR_n h_n$$

so that $S_n^t Mr_{n+1} = 0$ if we define

$$r_{n+1} h_{n+1n} = Ar_n - R_n h_n.$$

For the conjugate gradient method, the upper Hessenberg matrix is usually only tridiagonal if A is symmetric (see the remark in section 4.1). However, if MA is symmetric, then it is clear from

$$R^t MAR = R^t MRH \tag{5}$$

that under M-orthogonalization the upper Hessenberg matrix H is again tridiagonal.

4.4 Different choice of inner product in the Lanczos method

Consider a generalization of the Lanczos method with a matrix M inducing the inner product with respect to which the sequences S and R are orthogonalized. We want to be able to write $S^t MRH = K^t S^t MR$. Since $S^t MAR = S^t MRH$, we have to solve k_n from

$$k_n^t S_n^t MR_n = s_n^t MAR_n$$

and define

$$s_{n+1}^t k_{n+1n} = s_n^t MAM^{-1} - k_n^t S_n^t$$

so that $s_{n+1}^t M R_n = 0$. Apparently, S is no longer generated from a Krylov sequence of A^t, but from one of $(MAM^{-1})^t$.

Another (more symmetric) way of looking at this is to diagonalize $S^t MR$, and let R and S be generated from

$$AR = RH, \qquad B^t S = SK.$$

We conclude that

$$S^t MAR = S^t MRH, \qquad S^t BMR = K^t S^t MR,$$

so in order to conclude the tridiagonality characteristic of the Lanczos recurrence, we need that $MA = BM$, that is,

$$B = MAM^{-1}.$$

In practice, this limits M to powers of A or powers of A^{-1}.

5 Minimization

The choice of orthogonal sequences for R and S in the Lanczos method (or R orthogonal to itself for the conjugate gradient method) leads to some minimization properties. Additionally, minimization properties can be imposed by taking vector sequences that consist of linear combinations of the R sequence.

5.1 Lanczos' Minimized iterations

Theorem 3 *Choosing the sequences R and S to be orthogonal minimizes the inner products $s_i^t r_i$ (modulo some normalization of the sequences).*

Proof: Let X, Y be the Krylov sequences following from $AX = XJ$ and $A^t Y = YJ$, and assume that $r_1 \parallel x_1$, $s_1 \parallel y_1$, and $AR = RH$, $A^t S = SH$ for some upper Hessenberg matrix H, such that $R^t S$ is diagonal with positive diagonal elements.

Let further $\tilde{r}_1 = r_1$, $\tilde{s}_1 = s_1$ and $A\tilde{R} = \tilde{R}\tilde{H}$, $A^t \tilde{S} = \tilde{S}\tilde{H}$ for some upper Hessenberg matrix $\tilde{H}$. From lemma 2 we find that there are upper triangular matrices U, $\tilde{U}$ such that $R = XU$, $S = YU$, $\tilde{R} = X\tilde{U}$, $\tilde{S} = Y\tilde{U}$. Making the substitutions $H \leftarrow HU^{-1}$, $\tilde{H} \leftarrow \tilde{H}\tilde{U}^{-1}$, we find $AX = RH = \tilde{R}\tilde{H}$, $A^t Y = SH = \tilde{S}\tilde{H}$. Assuming that the sequences have been normalized such that H, $\tilde{H}$ can be written as $H = J + V$, $\tilde{H} = J + \tilde{V}$ for some upper triangular matrices V, $\tilde{V}$, we find

$$\tilde{R}J = RJ + X\hat{V}, \qquad \tilde{S}J = SJ + Y\hat{V},$$

where $\hat{V}$ is the upper triangular matrix $\tilde{U}\tilde{V} - UV$.

We now get

$$J^t\tilde{S}^t\tilde{R}J = J^tS^tRJ + J^tS^tX\hat{V} + \hat{V}^tY^tRJ + \hat{V}^tY^tX\hat{V}$$

in which the second and third term are strictly upper and lower triangular respectively. (Here we use for instance that $S^tX = S^tRU^{-1}$ is upper triangular; furthermore, $Y^tX = U^{-t}S^tRU^{-1}$.) Therefore, $\tilde{s}_n^t\tilde{r}_n \geq s_n^tr_n$, with equality only in the case that $\hat{V} = 0$, that is, if $\tilde{R} = R$, $\tilde{S} = S$. •

For the symmetric case of $A = A^t$, $S = R$, this says that the orthogonalizing algorithm minimizes the length of the r_n vectors in each iteration. For the general Lanczos method it gives the minimization of the inner product $s_i^tr_i$, but this implies no minimization for either the r_i or the s_i vectors. This minimization property led Lanczos [13, 14] to name this method 'minimized iterations'. Another name is the 'biconjugate gradient method' [7].

5.2 Symmetric CG: Minimization in the A^{-1} norm

In addition to theorem 3 we can prove a further minimization property for the conjugate gradient method for spd systems.

Theorem 4 *Orthogonalizing the residuals r_i for the conjugate gradient method applied to symmetric positive definite systems A minimizes their lengths in the A^{-1}-norm (modulo some normalization condition).*

Proof: Assume that $AR = RH$, $A\tilde{R} = \tilde{R}\tilde{H}$ and $r_1 \parallel \tilde{r}_1 \parallel x_1$, then there are upper triangular matrices U, $\tilde{U}$ such that $R = XU$, $\tilde{R} = X\tilde{U}$. Suppose the sequences have been normalized such that H, $\tilde{H}$ admits a factorization $H = (I-J)V$, $\tilde{H} = (I-J)\tilde{V}$ with V, $\tilde{V}$ upper triangular (in section 6.1 we shall see that this can be achieved by scaling the R sequence, and this is in fact the conventional conjugate gradient method); we find that $R(J-I) = -ARV^{-1}$, $\tilde{R}(J-I) = -A\tilde{R}\tilde{V}^{-1}$, and using $(J-I)(I-J^t)^{-1} = J - E_1$ with the matrix E_1 from section 2

$$R(J-E_1) = AXW, \qquad \tilde{R}(J-E_1) = AX\tilde{W}$$

where $W = -UV^{-1}(I-J^t)^{-1}$, $\tilde{W} = -\tilde{U}\tilde{V}^{-1}(I-J^t)^{-1}$ are upper triangular matrices.

Taking the difference we find $\tilde{R}J = RJ + AX\hat{W}$ with $\hat{W} = \tilde{W} - W$, so (using the symmetry of A)

$$\begin{aligned} J^t\tilde{R}^tA^{-1}\tilde{R}J &= J^tR^tA^{-1}RJ + J^tR^tA^{-1}AX\hat{W} + \hat{W}^tX^tA^tA^{-1}RJ \\ &\quad + \hat{W}^tX^tA^tA^{-1}AX\hat{W} \\ &= J^tR^tA^{-1}RJ + J^tR^tX\hat{W} + \hat{W}^tX^tRJ + \hat{W}^tX^tAX\hat{W}, \end{aligned}$$

in which the second and third term are strictly upper and lower triangular because of the upper triangularity of $R^tX = R^tRU^{-1}$. Thus $\tilde{r}_i^tA^{-1}\tilde{r}_i \geq r_i^tA^{-1}r_i$ with equality only if $\hat{W} = 0$, that is if $\tilde{R} = R$. •

It is easy to see that orthogonalizing the R sequence under the A^k inner product will similarly lead to minimization in the A^{k-1} norm. For the case of $k = 1$ this even holds for the nonsymmetric case, as will be shown in the next section.

5.3 Minimization in nonsymmetric conjugate gradient methods

Neither of the two preceding sections stated a minimization property for nonsymmetric conjugate gradient methods, that is, for the choice $S = R$ with A not a symmetric matrix. We can derive a minimization property for this case if R is orthogonalized under the A-inner product.

Theorem 5 *The sequence $R = XU$ (where X is the Krylov sequence satisfying $AX = XJ$) is A-orthogonal if and only if the norms of the r_n vectors are minimized (see also [19, theorem 1]).*

Proof: As in the proof of theorem 4 we derive that any choice $\tilde{R}$ in the same Krylov space as R is related to R by $\tilde{R}J = RJ + AXW$, so we find that

$$J^t\tilde{R}^tRJ = J^tR^tRJ + J^tR^tAXW + W^tX^tA^tRJ + W^tX^tA^tAXW.$$

From diagonality of R^tAR we find that $R^tAX = R^tARU^{-1}$ is upper triangular, so the second and third term in the above right hand side are strictly upper and lower triangular respectively. Therefore $\tilde{r}_n^t\tilde{r}_n \geq r_n^tr_n$ with equality only if A^tA is semi-definite or if $W = 0$.

Conversely, for any nonsingular upper triangular matrix W the function $f(\alpha) = \|RJ + \alpha ARW\|$ is minimized for $\alpha = 0$. Differentiating and substituting $\alpha = 0$ gives $(R^tARW)_{nn} = 0$, that is, R is A-orthogonal. •

5.4 Minimal residual methods: GMRES and QMR

Two methods have been proposed that generate a vector sequence satisfying a minimization property by first generating the sequence R in the usual way, and subsequently taking combination of this sequence. This procedure can be applied both to the Lanczos method and the conjugate gradient method.

Let then R satisfy $AR = RH$, and construct a sequence G by taking combinations, specifically, $G = RV_1$ with V_1 upper triangular with column sums equal to 1. Then $V_1(J - E_1)$ is a Hessenberg matrix with zero column sums, so by lemma 3 it can be written as $V_1(J - E_1) = (I - J)V_2$. Therefore, for some upper triangular V_3 and V_4:

$$G(J - E_1) = RV_2(J - E_1) = R(I - J)V_2 = RHV_3 = -ARV_4.$$

Thus we arrive at the following formulation for these combining methods:

$$G(J - E_1) = -ARV, \qquad g_1 = r_1$$

where V is an upper triangular matrix.

Lemma 8 *The elements of the sequence G are linear combinations of the same Krylov sequence that underlies R.*

Proof: Let $R = XU$ where is U is upper triangular and X is a Krylov sequence (see lemma 2), then $G(J - E_1) = -ARV = -AXUV = -XJUV$. Since $GE_1 = RE_1 = XE_1$, we find

$$GJ = X(E_1 - JUV) \Rightarrow GJJ^t = X(E_1 - JUV)J^t.$$

In this equation, JJ^t is the identity matrix except for a zero $(1,1)$ element, and $(E_1 - JUV)J^t$ is an upper triangular matrix with a zero $(1,1)$ element. Thus we can augment the first column of this matrix equation to find $G = XW$ with $W = E_1 - JUVJ^t$, which is a nonsingular upper triangular matrix. •

From the choice $g_1 = r_1$ we find that $GJ = R(E_1 - HV)$, and the columns v_n of V can be derived by solving the least squares problem

$$\min \|H_n v_n - e_1^{(n+1)}\|_2.$$

If R is constructed by the conjugate gradient method this method is called GMRES (Generalized Minimal Residual [20]), and since R is orthogonal this minimizes the g_n vectors (which will turn out to be residuals) in the 2-norm. For a matrix R constructed by the Lanczos process this method is called QMR (Quasi Minimal Residual [8]; the authors propose basing QMR on a block-Lanczos process called 'look-ahead Lanczos', but that is not relevant for the current discussion), and it does not directly minimize the residual, only the coefficient vector.

Lemma 9 *The GMRES method is equivalent to the conjugate gradient method under the A-inner product.*

Proof: This follows immediately from theorem 5 and lemma 8, using the fact that the method minimizes the norms of the residual vectors g_n. •

The least squares solution to $HV = E_1$ can for instance be found by making a QR decomposition $H = QU$ with Q orthonormal and U upper triangular. For the QMR method this will result in U being upper tridiagonal. We find that $V = U^{-1}\tilde{Q}^t$, where $\tilde{Q}$ is formed from Q as

$$\tilde{Q}^t = \begin{pmatrix} q_{11} & q_{11} & \cdots & q_{11} \\ & q_{12} & \cdots & q_{12} \\ & \emptyset & \ddots & \vdots \\ & & & q_{1n} \end{pmatrix}.$$

The residuals are then computed as

$$G(J - E_1) = -AP\tilde{Q}^t \qquad \text{where } P = RU^{-1}.$$

For an updating formula for G we find the even simpler formula

$$G(J - I) = -AP\, diag(q_{1i}).$$

The elements of P can be easily updated from the r_n vectors, in the case of QMR with a three-term recurrence. In GMRES updating p_n requires all previous such vectors to be stored. Hence, people have considered truncated or restarted versions of the method.

6 The iterative formulation of the conjugate gradient and Lanczos methods

The matrix equation $AR = RH$ implies in cases where H is tridiagonal a three-term recurrence

$$r_{n+1}h_{n+1n} + r_n h_{nn} + r_{n-1}h_{n-1n} = Ar_n.$$

Traditionally this was the way Lanczos [13] derived his bi-orthogonalization method. The conjugate gradient method, however, was presented by its discoverers Hestenes and Stiefel [11] as two coupled two-term recurrences. Such a formulation was later given by Fletcher [7] for the Lanczos method, who called it the 'biconjugate gradient method'.

In this section we will show how the two formulations are equivalent, and how the view as coupled two-term recurrences arises from factoring the Hessenberg matrix.

6.1 Search directions and the solution of linear systems

The generating recurrence $AR = RH$ can be split into two coupled recurrences. Factor H as $H = (I - L)D^{-1}(I - U)$ where L is a strictly lower diagonal matrix, and U is strictly upper triangular (single upper diagonal for symmetric conjugate gradients or the Lanczos method). Introducing $P = R(I - U)^{-1}$ we get the coupled recurrences

$$APD = R(I - L); \qquad R = P(I - U).$$

The elements of the sequence P are usually called 'search directions'.

If vectors r_n are replaced by scalar multiples $c_n r_n$ of themselves, this corresponds to a transformed recurrence for the sequence RC (where $C = \mathrm{diag}(c_i)$)

$$ARC = (RC)(C^{-1}HC)$$

of the original form. In particular, for $H = (I - L)D^{-1}(I - U)$ we can define a scalar recurrence

$$c_1 = 1; \qquad c_{i+1} = L_{i+1i}c_i$$

so that

$$C^{-1}(I - L)C = I - J.$$

As a result,

$$C^{-1}HC = (I - J)C^{-1}D^{-1}C(I - C^{-1}UC) = (I - J)D^{-1}(I - C^{-1}UC)$$

is a factorization of the Hessenberg matrix of the transformed recurrence for RC. Note that the matrix D is invariant under scalings of the R sequence.

Table 1 gives orthogonality properties for the gradients and search directions thus derived.

For the transformed recurrence, now simply denoted R, we thus get coupled formulas as above:

$$APD = R(I - J), \qquad R = P(I - U). \tag{6}$$

Equation /	Shape	Relation	Valid for
R^tR	diagonal	$r_i^t r_j = 0$	$i \neq j$
$R^tAR = R^tRH$			
	upper Hessenberg	$r_i^t A r_j = 0$	$i > j+1$
	symm: tridiagonal	$r_i^t A r_j = 0$	$\|i-j\| > 1$
$P^tAP = (I-U)^{-t}R^tR(I-J)D^{-1}$			
	lower triangular	$p_i^t A p_j = 0$	$j > i$
	symm: diagonal	$p_i^t A p_j = 0$	$i \neq j$
$P^tR = (I-U)^{-t}R^tR$		$p_i^t r_i = r_i^t r_i$	
	lower triangular	$p_i^t r_j = 0$	$j > i$
$R^tAP = R^tR(I-J)D^{-1}$			
	lower bidiagonal	$r_i^t A p_j = 0$	$j > i$, $i > j+1$
$R^tAP = (I-U)^t P^tAP$		$r_i^t A p_i = p_i^t A p_i$	

Table 1: Orthogonality properties of the conjugate gradient algorithm.

In this we recognize the classical formulation of the conjugate gradient method: if x_n are iterates[1], and p_n is a search direction, then scalars d_{nn} are chosen such that

$$x_{n+1} = x_n - d_{nn} p_n, \tag{7}$$

with for the gradients $r_n = Ax_n - f$:

$$r_{n+1} = r_n - d_{nn} A p_n,$$

which is the first half of the coupled recurrence. In order to ensure consistency, we do have to choose explicitly $r_1 = Ax_1 - f$. Since the residuals r_i are orthogonal, for some value n we will have $r_n = 0$. For that n, x_n is the solution of the linear system $Ax = f$. Note that this is the theoretical exact convergence of the conjugate gradient method, discovered by [11]; in practice the method usually converges to a given tolerance in far fewer iterations. This 'speed of convergence' depends mostly on the eigenvalues of the matrix; see for instance [1] or [22].

6.2 The scalars in the CG algorithm

If the Hessenberg matrix is factored $H = (I-J)D^{-1}(I-U)$ with U strictly upper triangular, then for the conjugate gradient method there are only two sets of scalars that we need to know: the quantities d_{ii} and $u_{i-k,i}$ ($k > 0$; for symmetric problems only $k = 1$ is needed).

1. In section 3 we used x_n to denote elements of a Krylov sequence; from now on x_n will denote solely the n-th iterate.

Algorithm 2 *(Conjugate Gradient Method) In order to solve $Ax = f$, choose x_1 arbitrarily. Choose $p_1 = r_1 = Ax_1 - f$. Then perform the following steps for $i = 1, \ldots$:*

$$\text{compute } d_{ii} \tag{8}$$

Update the iterate

$$x_{i+1} = x_i - p_i d_{ii} \tag{9}$$

and the residual

$$r_{i+1} = r_i - Ap_i d_{ii} \tag{10}$$

$$\text{compute } u_{ki+1} \text{ for } k = 1..i \tag{11}$$

Update the search direction

$$p_{i+1} = r_{i+1} + \sum_{k=1}^{i} p_k u_{ki+1} \tag{12}$$

If Ar_i is computed, Ap_i can be derived recursively from

$$Ap_{i+1} = Ar_{i+1} + \sum_{k=1}^{i} Ap_k u_{ki+1} \tag{13}$$

where we note that $Ap_1 = Ar_1$.

The scalars d_{ii}, $u_{i-k,i}$ follow in various ways from the orthogonality conditions. From the relation $APD = R(I-J)$ we get, using the fact that $r_i^t p_j = 0$ for $i > j$:

$$p_i^t Ap_i d_{ii} = p_i^t r_i - p_i^t r_{i+1} \Rightarrow d_{ii} = \frac{p_i^t r_i}{p_i^t Ap_i} = \frac{r_i^t r_i}{p_i^t Ap_i} \tag{14}$$

and also

$$r_i^t Ap_i d_{ii} = r_i^t r_i - r_i^t r_{i+1} \Rightarrow d_{ii} = \frac{r_i^t r_i}{r_i^t Ap_i}$$

Comparing elements in the left and right hand side of $R^t AR = R^t RH$, we find from the lower diagonal

$$r_{i+1}^t Ar_i = r_{i+1}^t r_{i+1} h_{i+1,i} = -r_{i+1}^t r_{i+1} d_{ii}^{-1} \Rightarrow d_{ii} = -\frac{r_{i+1}^t r_{i+1}}{r_{i+1}^t Ar_i}$$

(which is not a useful formula because d_{ii} is needed to compute r_{i+1}) and from the upper triangle we find for $k > 0$

$$r_{i-k}^t Ar_i = -r_{i-k}^t r_{i-k} d_{i-k,i-k}^{-1} u_{i-ki} \Rightarrow u_{i-ki} = -\frac{r_{i-k}^t Ar_i}{r_{i-k}^t r_{i-k}} d_{i-k,i-k}.$$

If A is symmetric we find from $r_{i-1}^t Ar_i = r_i^t Ar_{i-1}$ that

$$u_{i,i+1} = \frac{r_{i+1}^t r_{i+1}}{r_i^t r_i}. \tag{15}$$

For nonsymmetric problems the elements of U can be computed by recursion in each column. Observing that P^tAP is lower triangular we find from P^tAP $(I-U) = P^tAR$ that for $i < j$

$$-\sum_{k=1}^{i} p_i^t A p_k u_{kj} = p_i^t A r_j. \tag{16}$$

Another way of computing U follows from the equation $D^{-1}(I-J)^t R^t R = P^t A^t R = P^t A^t P(I-U)$. We find for the columns of U

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{r_{n+1}^t r_{n+1}}{r_n^t r_n} \end{pmatrix} = \frac{1}{p_n^t A p_n} \begin{pmatrix} p_1^t A^t p_1 & \dots & p_1^t A^t p_n \\ & \ddots & \vdots \\ & & p_n^t A^t p_n \end{pmatrix} \begin{pmatrix} u_{1n+1} \\ \vdots \\ u_{nn+1} \end{pmatrix} \tag{17}$$

which reduces immediately to (15) for symmetric systems.

6.3 Iterative formulation of the Lanczos method: Bi-CG

Most authors present the Lanczos method as a three-term recurrence (see also section 9.2), and reserve the name 'biconjugate gradients' for the mathematically equivalent form involving two coupled two-term recurrences.

In order to let the presentation be as far as possible analogous to that of the conjugate gradient method, we will take the latter approach. That is, we arrive at an iterative formulation for the Lanczos method by introducing search directions corresponding to the left Krylov vectors. To this purpose we split the relation $A^tS = SH$ into

$$A^tQD = S(I-J), \qquad Q = S(I-U).$$

Algorithm 3 *(Lanczos Method) In order to solve $Ax = f$, choose x_1 arbitrarily. Choose $p_1 = r_1 = Ax_1 - f$ and $q_1 = s_1$ arbitrarily. Then perform the following steps for $i = 1, \ldots$:*

$$\textit{compute } d_{ii} \tag{18}$$

Update the iterate

$$x_{i+1} = x_i - p_i d_{ii} \tag{19}$$

and the left and right residuals

$$r_{i+1} = r_i - Ap_i d_{ii}, \qquad s_{i+1} = s_i - A^t q_i d_{ii}; \tag{20}$$

$$\textit{compute } u_{ii+1} \tag{21}$$

Update left and right search directions

$$p_{i+1} = r_{i+1} + p_i u_{ii+1}, \qquad q_{i+1} = s_{i+1} + q_i u_{ii+1} \tag{22}$$

The resulting orthogonality relations for the two sequences of iterates and search directions are in table 2.

Equation /	Shape	Relation	Valid for
$S^t R$	diagonal	$s_i^t r_j = 0$	$i \neq j$
$S^t AR = S^t RH = H^t S^t R$			
	tridiagonal	$s_i^t Ar_j = 0$	$\|i-j\| > 1$
$Q^t AP = (I-U)^{-t} S^t R(I-J)D^{-1}$			
$\quad = D^{-1}(I-J)^t S^t R(I-U)^{-1}$			
	diagonal	$q_i^t Ap_j = 0$	$i \neq j$
$Q^t R = (I-U)^{-t} S^t R$		$q_i^t r_i = s_i^t r_i$	
	lower triangular	$q_i^t r_j = 0$	$j > i$
$S^t P = S^t R(I-U)^{-1}$		$s_i^t p_i = s_i^t r_i$	
	upper triangular	$s_i^t p_j = 0$	$i > j$
$S^t AP = S^t R(I-J)D^{-1}$			
	lower bidiagonal	$s_i^t Ap_j = 0$	$j > i,\ i > j+1$
$Q^t AR = D^{-1}(I-J)^t S^t R$			
	upper bidiagonal	$q_i^t Ar_j = 0$	$i > j,\ j > i+1$
$S^t AP = (I-U)^t Q^t AP$		$s_i^t Ap_i = q_i^t Ap_i$	
$Q^t AR = Q^t AP(I-U)$		$q_i^t Ar_i = q_i^t Ap_i$	

Table 2: Orthogonality properties of the Lanczos method.

The scalar sequences d_{ii} and u_{ii+1} now can be derived as follows. Taking the inner product of Q with $APD = R(I-J)$ we find that

$$d_{ii} = \frac{s_i^t r_i}{q_i^t Ap_i}; \tag{23}$$

from $S^t AR = S^t R(I-J)D^{-1}(I-U)$ we find that

$$s_{i+1}^t Ar_i = -s_{i+1}^t r_{i+1} d_{ii}^{-1}, \qquad s_i^t Ar_{i+1} = -s_i^t r_i d_{ii}^{-1} u_{ii+1}$$

which, with the symmetry of $S^t AR$ (see theorem 2) gives

$$u_{ii+1} = \frac{s_{i+1}^t r_{i+1}}{s_i^t r_i}. \tag{24}$$

7 Preconditioning

The idea of preconditioning is usually simply presented as replacing an iteration on the system $Au = f$ by one on $MAu = Mf$. However, for conjugate gradient methods it is relevant whether the iteration matrix is symmetric or not. Hence we do not want to iterate directly with MA, which usually is an unsymmetric system, even if A and M are symmetric.

In this section we will derive how the formulas for the conjugate gradient and Lanczos methods have to be altered to incorporate a preconditioner. The actual construction of the preconditioner will be left undiscussed, this being a whole field of science in itself.

7.1 Preconditioned conjugate gradients

If M is symmetric and positive definite, we can split it (theoretically, not computationally) as $M = EE^t$, and we formulate the conjugate gradients method as iterating on the system

$$E^tAE(E^{-1}x) = E^tf,$$

that is, r_n vectors are generated from the equation

$$E^tAER = RH$$

where H is again factored as $H = (I - J)D^{-1}(I - U)$.

The first of the coupled recurrences in section 6.1 is now replaced by $E^tAEPD = R(I - J)$, and we get for the full method

$$A(EP)D = (E^{-t}R)(I - J), \qquad E^{-t}R = E^{-t}E^{-1}(EP)(I - U).$$

Introducing transformed sequences $\tilde{R} = E^{-t}R$ and $\tilde{P} = EP$, we get the coupled recurrences

$$A\tilde{P}D = \tilde{R}(I - J), \qquad M\tilde{R} = \tilde{P}(I - U). \tag{25}$$

Computationally, the algorithm is extended by a single step of forming the product MR, and the splitting of M is never explicitly required.

In order to compute the scalar quantities of the algorithm we note that $R^tR = \tilde{R}^tM\tilde{R}$. Where the conjugate gradient algorithm uses $p_i^tAp_i$, we now get, because of the transformed system, $p_i^tE^tAEp_i$, which is equal to $\tilde{p}_i^tA\tilde{p}_i$. For the methods in section 9.3 we further note that R^tAR becomes in the transformed system $R^tE^tAER = \tilde{R}^tM^tAM\tilde{R}$.

If we let $\tilde{x}$ be the solution of $E^tAE\tilde{x} = E^tf$, then in the end we are interested in obtaining $x = E\tilde{x}$. Noting that the updating formula $\tilde{X}(J - I) = PD$ follows from equation (25), we find that

$$X(J-I) = E\tilde{X}(J-I) = \tilde{P}D, \tag{26}$$

that is, using the transformed search directions we can actually update approximations to the solution of the original system.

Also,

$$\tilde{R} = E^{-t}R = E^{-t}(E^tAE\tilde{x} - E^tf) = Ax - f,$$

that is, the transformed residuals are the residuals of the original system.

Summarizing, we get the following algorithm (where we now simply write r_i for the $\tilde{r}_i$ of the above theory).

Algorithm 4 *(Preconditioned Conjugate Gradient method) Choose x_1 arbitrarily, and let $r_1 = Ax_1 - f$ and $p_1 = \hat{r}_1 = Mr_1$. Then perform the following steps for $i = 1, \ldots$:*

$$\text{compute } d_{ii} = \hat{r}_i^t r_i / p_i^t A p_i \tag{27}$$

Update the iterate

$$x_{i+1} = x_i - p_i d_{ii} \tag{28}$$

and the residual

$$r_{i+1} = r_i - Ap_i d_{ii}. \tag{29}$$

Apply the preconditioner

$$\hat{r}_{i+1} = Mr_{i+1}, \tag{30}$$

$$\text{compute } u_{ii+1} = \hat{r}_{i+1}^t r_{i+1} / \hat{r}_i^t r_i. \tag{31}$$

Update the search direction

$$p_{i+1} = \hat{r}_{i+1} + p_i u_{ii+1}. \tag{32}$$

7.2 Preconditioning as a change of matrix and inner product

We started formulating the preconditioned iteration as arising from a Krylov sequence with the matrix E^tAE. However, the resulting sequence R satisfying $E^tAER = RH$ is never formed. Instead we compute a sequence $\tilde{R} = E^{-t}R$. This sequence can be interpreted as arise from a Krylov sequence of another system, orthogonalized under another inner product.

Lemma 10 *The sequence $\tilde{R}$ consists of linear combinations of a Krylov sequence of the matrix AM, and it is orthogonal under the M-inner product.*

Proof: The sequence $\tilde{R}$ satisfies $AM\tilde{R} = \tilde{R}H$, that is, it arises from a Krylov sequence of the matrix AM (see lemma 2). For a general symmetric choice for M this matrix will be nonsymmetric, but noting that the scalars in the iterative method are determined from the equation

$$\tilde{R}^t MAM\tilde{R} = \tilde{R}^t M\tilde{R}H$$

we find by comparing to equation (5) that we have orthogonalized $\tilde{R}$ under the M inner product, which symmetrizes the matrix AM again. •

Although the discussion so far used a symmetric preconditioner $M = EE^t$, it is easy to see that taking $M = FE^t$ and iterating on a system with coefficient matrix E^tAF will also lead to a formula $M\tilde{R} = \tilde{P}(I - U)$, that is, involving only the unfactored preconditioner.

7.3 Preconditioned Lanczos

Preconditioning the Lanczos iteration is largely analogous to preconditioning the conjugate gradient method as described above. (The only exposition of a preconditioned Lanczos method in the literature that we are aware of is in [4]; apart from an implementational detail it is identical to the method derived here.)

Since the Lanczos method can handle nonsymmetric matrices, assume that the preconditioner is nonsymmetric too, and that we iterate on a system with matrix E^tAF. From

$$E^tAFPD = R(I - J), \qquad F^tA^tEQD = S(I - J)$$

we find

$$A\tilde{P}D = \tilde{R}(I - J), \qquad A^t\tilde{Q}D = \tilde{S}(I - J)$$

where

$$\tilde{R} = E^{-t}R, \qquad \tilde{S} = F^{-t}S, \qquad \tilde{P} = FP, \qquad \tilde{Q} = EQ.$$

For the scalar quantities we find first of all that

$$S^tR = \tilde{S}M\tilde{R}$$

where $M = FE^t$ is the preconditioner. Furthermore, the quantity $q_i^tAp_i$ in the conjugate gradient method now becomes $q_i^tE^tAFp_i$, which is equal to $\tilde{q}_i^tA\tilde{p}_i$.

The update relations $R = P(I - U)$ and $S = Q(I - U)$ for the search directions become

$$M\tilde{R} = \tilde{P}(I - U), \qquad M^t\tilde{S} = \tilde{Q}(I - U)$$

for the transformed sequences. We see that applications of both the preconditioner and its transpose are necessary.

8 Polynomial squaring methods: CGS

It has been observed [21] that the Lanczos method computes both $r_n = \phi_n(A)r_1$ and $s_n = \phi_n(A^t)s_1$, where ϕ_n is an $n-1$-st degree polynomial, and r_n, s_n are the right and left residuals. Since presumably both residuals tend to zero, and

$$s_n^t r_n = (\phi_n(A^t)s_1)^t(\phi_n(A)r_1) = s_1^t(\phi_n^2(A)r_1)$$

it may seem like a good idea to compute to compute the sequence $\phi_n^2(A)r_1$ since, ideally, this sequence would have double the convergence speed of the original sequence. Here is why computing this sequence is possible.

Let X be a Krylov sequence of A, that is, $AX = XJ$, and let R consist of linear combination of the X sequence: $R = XU$. From the foregoing discussion we know that we have $AR = RH$ with $H = U^{-1}JU$. Suppose that H is tridiagonal, for instance because R is generated with the Lanczos iteration. For the n-th residual we have $r_n = \phi_n(A)x_1$ where the polynomials ϕ_n are related by

$$\phi_{n+1}(t)h_{n+1n} + \phi_n(t)h_{nn} + \phi_{n-1}(t)h_{n-1n} = t\phi_n(t); \tag{33}$$

see lemma 4.

We now define in each n-th step a new Krylov vector sequence $Y^{(n)}$ by

$$Y^{(n)} = \phi_n(A)X \Rightarrow AY^{(n)} = Y^{(n)}J.$$

Using again the upper triangular matrix U to take linear combinations, this time from $Y^{(n)}$, we define $S^{(n)}$ by

$$S^{(n)} = Y^{(n)}U \Rightarrow AS^{(n)} = S^{(n)}H. \tag{34}$$

Note that the matrix H is the same as above, and in particular independent of n.

The justification for these new sequences is that

$$y_1^{(n)} = r_n, \qquad s_n^{(n)} = \phi_n(A)y_1^{(n)} = \phi_n^2(A)r_1$$

Naturally, we do not wish to build the whole sequence $S^{(n)}$ in the n-th step. It turns out that of each $S^{(n)}$ we need only four elements. From (33) we find that

$$S^{(n+1)}h_{n+1n} + S^{(n)}h_{nn} + S^{(n-1)}h_{n-1n} = AS^{(n)}$$

and combining this with (34) we find the following computation:

$$
\begin{aligned}
s_{n-1}^{(n+1)}h_{n+1n} &= As_{n-1}^{(n)} - (s_{n-1}^{(n)}h_{nn} + s_{n-1}^{(n-1)}h_{n-1n}) \\
s_{n}^{(n+1)}h_{n+1n} &= As_{n}^{(n)} - (s_{n}^{(n)}h_{nn} + s_{n}^{(n-1)}h_{n-1n}) \\
s_{n+1}^{(n+1)}h_{n+1n} &= As_{n}^{(n+1)} - (s_{n}^{(n+1)}h_{nn} + s_{n-1}^{(n+1)}h_{n-1n}) \\
s_{n+2}^{(n+1)}h_{n+2n+1} &= As_{n+1}^{(n+1)} - (s_{n+1}^{(n+1)}h_{n+1n+1} + s_{n}^{(n+1)}h_{nn+1})
\end{aligned}
$$

Of each equation, the left hand side is the computed result. Additionally, in the last two steps we perform a matrix-vector product; the results of these can be reused in the first two steps of the next iteration. We see that this method requires two matrix-vector products, but neither is with the transpose of the matrix.

For the computation of the coefficients of H we have to go back to the Lanczos method. In section 6.3 we saw that we need the expressions (now denoting the left sequence of residuals by $\tilde{r}_n$)

$$\tilde{r}_n^t A r_n \quad (23), \qquad \tilde{r}_n^t r_n \quad (23) \text{ and } (24).$$

From the above it is easy to see that

$$\tilde{r}_n^t r_n = \tilde{r}_1^t \phi_n^2(A) r_1 = \tilde{r}_1^t s_n^{(n)}, \qquad \tilde{r}_n^t A r_n = \tilde{r}_1^t A \phi_n^2(A) r_1 = \tilde{r}_1^t A s_n^{(n)}.$$

Thus the left sequence can be eliminated completely, and only its first element is ever needed. Using the above two expression, the elements of H can be computed, for instance as in equation (35).

There is in fact an unused degree of freedom in the above method: we can choose a different upper triangular matrix V to form the linear combinations $S^{(n)}$, so that

$$S^{(n)} = Y^{(n)}V \Rightarrow AS^{(n)} = S^{(n)}K$$

where $K = V^{-1}JV$. We will not further pursue this approach here.

9 Other aspects of the iterative solution process

9.1 Iterative solution as solving reduced systems

In equation (7) we saw that the orthogonalization process of a Krylov sequence can be used to compute iteratively approximations of the solution of a linear system. Writing this vector recurrence again in block form $X(I-J) = PD$, we find that the iterates can be computed iteratively or from an initial approximation as

$$X(I-J) = R(I-U)^{-1}D \qquad \text{or} \qquad X(E_1 - J) = RV$$

where $V = (I-U)^{-1}D(I-J^t)^{-1}$ is an upper triangular matrix. Thus, every iterate differs from the previous and from the initial one by a span of residuals.

For the columns v_n of V we find after some elementary throwing around of formulas that

$$R^t ARV = R^t R(E_1 - J)$$

so

$$R_n^t A R_n v_n = \|r_1\|^2 e_1^{(n)}$$

where $e_1^{(n)}$ is the first length n unit vector. We see that each of the columns of V follows from the solution of a reduced system that is derived from the previous by the addition of a row and column.

9.2 Three-term recurrence for iterates

Some authors [5, 10], have considered a three-term recurrence for the iterands in the conjugate gradient algorithm, corresponding to the three-term recurrence for the residuals. Such a three-term recurrence is the usual mode of presentation for the Lanczos method and the Chebyshev semi-iterative method [10].

From the splitting $H = (I-J)D^{-1}(I-U)$ of the Hessenberg matrix in $AR = RH$ we get

$$-r_{n+1}d_{nn}^{-1} + r_n(d_{nn}^{-1} + d_{n-1n-1}^{-1}u_{n-1n}) - r_{n-1}d_{n-1n-1}^{-1}u_{n-1n} = Ar_n. \quad (35)$$

Since the sum of the weights of the residuals in equation (35) is zero, we can extract the matrix A from this equation and derive for the iterates

$$x_{n+1} = (1 + u_{n-1n}d_{nn}/d_{n-1n-1})x_n - r_n - u_{n-1n}(d_{nn}/d_{n-1n-1})x_{n-1}.$$

Conversely, a different choice of H, or equivalently a different scaling of the r_n sequence, will not have an analogous relation for updating iterates. Thus, lemma 3 is seen to be a consistency condition for solving linear systems with H. We will return to this matter in section 9.5.

A three-term recurrence can also directly be derived from the elements of the Hessenberg matrix:

$$r_{i+1}h_{i+1i} + r_i h_{ii} + r_{i-1}h_{i-1i} = Ar_i \quad (36)$$

where from orthogonalization properties we get the expressions

$$h_{ii} = \frac{r_i^t A r_i}{r_i^t r_i}, \qquad h_{ii+1} = \frac{r_i^t A r_{i+1}}{r_i^t r_i}, \qquad h_{i+1i} = \frac{r_{i+1}^t A r_i}{r_{i+1}^t r_{i+1}}. \quad (37)$$

Note that the h_{i+1i} are arbitrary normalization factors.

If A is symmetric, the scalar h_{ii+1} can be derived as

$$h_{ii+1} = \frac{r_{i+1}^t r_{i+1}}{r_i^t r_i} h_{i+1i}, \tag{38}$$

which can be shown [16, 17] to be a more stable variant of the method.

The three-term form (36) is often written as two coupled two-term recurrences

$$r_{i+1}h_{i+1i} + r_i h_{ii} = t_i \qquad \text{where} \qquad t_i = Ar_i - r_{i-1}h_{i-1i}.$$

Although this splitting is of less mathematical significance than the introduction of search directions in (6), it does allow for a slight variant of the computation by noting that $r_i^t A r_i = r_i^t t_i$, which is used in the computation of h_{ii}. This variant is as stable as the original method [16].

9.3 Equivalent formulas for the scalar quantities

In a parallel implementation, the conjugate gradient algorithm as formulated above has two synchronization points per iteration, located at the inner product calculations. These synchronization points cannot be coalesced immediately: the scalar d_{ii} needed to compute r_{i+1} needs an inner product with p_i, and the scalar u_{ii+1} necessary to compute p_{i+1} requires an inner product with r_{i+1}. Thus the two inner products are interdependent. However, for the conjugate gradient method for spd systems the scalars can be computed in ways that eliminate one synchronization point, perhaps at the expense of some extra computation.

At the moment, two complementary approaches are known. One can either eliminate the inner product $r_i^t r_i$ and express it in terms of inner products with the search directions, or one can try to eliminate the inner product $p_i^t A p_i$ and express it in terms of inner products involving gradients.

The former approach was discovered by Saad [18]. From the orthogonality of R, the equation $APD = R(I - J)$ leads to

$$r_{i+1}^t r_{i+1} + r_i^t r_i = d_{ii}^2 (Ap_i)^t (Ap_i), \tag{39}$$

so the norms of r_i can be computed recursively without the need for an inner product. However, a simple analysis shows that this method is unstable, so extra measures are necessary. Hence, Meurant [15] proposed computing $r_i^t r_i$ explicitly, together with $(Ap_i)^t (Ap_i)$. Thus, the $r_{i+1}^t r_{i+1}$ value serves only as a predictor; it is later computed exactly. The resulting method takes three inner products per iteration, and is as stable as the classical formulation of the conjugate gradient method.

Recently, several methods based on elimination of computing $p_i^t A p_i$ (which is needed for $d_{ii} = r_i^t r_i / p_i^t A p_i$) have been discovered by the present author and [3]. First of all, combine $R^t AP = (I-U)^t P^t AP$ and $R^t AR = R^t AP(I-U)$ to

$$(I-U)^t P^t AP = R^t AR + R^t APU$$

and consider the diagonal of the left and right hand side. From

$$p_i^t A p_i = r_i^t A r_i + r_i^t A p_{i-1} u_{i-1i} \tag{40}$$

we see that $p_i^t A p_i$ can be computed from $r_i^t A r_i$ and $r_i^t A p_{i-1} = p_{i-1}^t (A r_i)$. Hence, one extra inner product is needed, and Ar_i is computed instead of Ap_i. This latter vector can be computed recursively as in equation (13).

Expanding $R^t AR$ one step further into $(I-U)^t P^t AP(I-U)$ gives

$$P^t AP = R^t AR + P^t APU + U^t P^t AP - U^t P^t APU.$$

Using the A-orthogonality of P and the fact that the second and third term in the rhs are strictly upper and lower triangular respectively, we find that

$$p_i^t A p_i = r_i^t A r_i - u_{i-1i}^2 p_{i-1}^t A p_{i-1}. \tag{41}$$

For the resulting variant of the conjugate gradient method ([3]) Ap_i is again computed recursively from Ar_i, the inner products $r_i^t r_i$ and $r_i^t A r_i$ are computed simultaneously, and the scalar $p_i^t A p_i$ needed for $d_{ii} = r_i^t r_i / p_i^t A p_i$ is computed from the above recurrence.

There is a fairly strong argument for the stability of this last rearrangement: using equations (37), (38) and (14) the recurrence for $p_n^t A p_n$ can be derived from the recurrence

$$d_{nn}^{-1} = h_{n+1n+1} - h_{n+1n} d_{nn} h_{nn+1}$$

for pivot generation in the factorization of the Hessenberg matrix. For symmetric positive definite systems this is a stable recurrence.

Another variant of this scheme results from considering the diagonal of

$$(I-U)^t PAP = R^t AP = R^t AR(I-U)^{-1}.$$

Since $R^t AR$ is tridiagonal the infinite expansion of $(I-U)^{-1}$ terminates quickly, and we find that

$$p_i^t A p_i = r_i^t A r_i + r_i^t A r_{i-1} u_{i-1i}.$$

However, in the presence of rounding errors this method, being based on an infinite number of orthogonalities, becomes unstable in contrast to the two previous methods which only use a single orthogonality relation.

Methods for removing a synchronization point generalize to the Lanczos method. We find that

$$q_{i+1}^t A p_{i+1} = s_{i+1}^t A r_{i+1} - s_i^t A p_i u_{ii+1}$$

from $(I-U)^t Q^t AP = S^t AR + S^t APU$;

$$q_{i+1}^t A p_{i+1} = s_{i+1}^t A r_{i+1} - u_{ii+1}^2 q_i^t A p_i$$

using A-orthogonality of Q and P in

$$Q^t AP = S^t AR + Q^t APU + U^t Q^t AP - U^t Q^t APU,$$

and we get

$$s_{i+1}^t r_{i+1} - s_i^t r_i = d_{ii}^2 (A^t q_i)^t (A p_i)$$

from the equations $APD = R(I-J)$ and $A^t QD = S(I-J)$ using orthogonality of S and R.

Another approach for a reduced synchronization overhead was proposed by [23] and [2], based on computing a number of Krylov vectors and orthogonalizing these as a block. However, this approach seems inadvisable from a point of stability.

9.4 Stationary iteration and the steepest descent method

The matrix framework that was used throughout this paper can also be applied to traditional stationary iterative methods and the steepest descent method (see for instance [24] and [10]).

Stationary iterative methods in their most basic form update an iterate as

$$x_{n+1} = x_n - M r_n \tag{42}$$

where $r_n = Ax_n - f$ and M approximates A^{-1}. The simplest method, Richardson iteration, results from choosing $M = D$ a diagonal matrix with $d_{ii} \equiv d$ for some scalar d. The updating formula can be written in matrix form as

$$X(I-J) = RD$$

from which we find $R(I-J) = ARD$, and thus

$$AR = R(I-J)D^{-1}. \tag{43}$$

Comparing this to the conjugate gradient method, we see that (43) can be derived from (6) by taking $P = R$ (or equivalently $U = 0$) and letting D be constant.

We see that this method generates a Hessenberg matrix $H = (I-J)D^{-1}$ of lower bidiagonal form. From lemma 2 and theorem 1 we find that $r_{n+1} = P_n(A) r_1$ where P_n is an n-th degree polynomial. (This fact could of course also have been derived in a more classical way.)

A more general iterative method results from choosing an approximation M to A^{-1}, and iterating

$$x_{n+1} = x_n - d_{nn} M r_n \tag{44}$$

where $d_{nn} \equiv d$ gives a general stationary iterative method, and the steepest descent method results from choosing d_{nn} such that it minimizes $\|r_{n+1}\|$. Writing this update equation as $X(I - J) = MRD$, we find that the residuals are generated by

$$AMR = RH \qquad \text{where } H = (I - J)D^{-1}.$$

We note the close resemblance between this formula and the generating formula for the preconditioned conjugate gradient method in lemma 10.

In the case where $d_{nn} \equiv d$ we can hide both D and M in a transformed system, turning equation 42 into

$$x_{n+1} = x_n - r_n. \tag{45}$$

Lemma 11 *The residuals corresponding to the iterative method (45) are Krylov vectors of the matrix $I - A$; the iterative method converges if $\rho(I - A) < 1$.*

Proof: The residuals satisfy $R(I - J) = AR$, which can be rewritten as $(I - A)R = RJ$, showing that they are a Krylov sequence of the matrix $I - A$. Since $\|r_{n+1}\| \leq \rho(I - A)\|r_n\|$ the iterative method will converge if the spectral radius is less than 1. •

9.5 Polynomial acceleration

Methods such as conjugate gradients are sometimes considered to be an acceleration of a basic iterative method. Suppose iterates x_n have been generated by (45), then the idea behind acceleration is to take combinations

$$\tilde{x}_n = \sum_{i=1}^{n} x_i u_{in} \tag{46}$$

with the consistency condition

$$\sum_{i=1}^{n} u_{in} = 1. \tag{47}$$

Varga [24] calls (46) the semi-iterative method corresponding to the basic iterative method. The residuals $\tilde{r}_n = A\tilde{x}_n - f$ are seen to satisfy $\tilde{R} = RU$ (where U is an upper triangular matrix containing the u_{in} coefficients) because of the above consistency condition.

Writing the basic iterative method as $AR = R(I-J)$ we find for the accelerated residuals $A\tilde{R} = \tilde{R}H$ where H is the upper Hessenberg matrix $I - U^{-1}JU$.

Lemma 12 *The consistency condition (47) implies the consistency condition of lemma 3 for $H = I - U^{-1}JU$.*

Proof: We can express equation (47) as $e^tU = e^t$. Since this implies $e^tU^{-1} = e^t$, and since $e^tJ = e^t$, we find that $e^tH = 0$. •

These acceleration methods are also called 'polynomial acceleration' methods because of the following fact.

Lemma 13 *The accelerated residuals are related to the original residuals as*

$$\tilde{r}_n = P_n(I-A)r_1$$

where P_n is an $n-1$-st degree polynomial with its coefficients in the n-th column of U, where U contains the u_{in} coefficients of equation (46).

Proof: Lemma 11 established that the residuals of the basic method are Krylov vectors in a sequence with matrix $I-A$. It then follows from theorem 1 that the combinations $\tilde{R}$ are obtained by multiplying by a matrix polynomial as stated above. •

10 Conclusion

We have presented the conjugate gradient-like methods in a matrix framework. A clear separation between the Hessenberg matrix associated with Krylov sequences, orthogonalization under different inner products, preconditioning, and minimization properties was made. Facts normally taken for granted, such as the three-term form of recurrences, or the fact that these methods can be used for iterative solution of linear systems, were derived, not taken as a premise. Additionally, we have shown how cg methods can be considered as polynomial accelerations of basic stationary iterative method.

Using the matrix framework for talking about vector sequences, we have given short derivations of the conjugate gradients method, both for symmetric and unsymmetric systems, the Lanczos algorithm and the equivalent biconjugate gradient method, the least squares methods GMRES and QMR and the conjugate gradients squared method.

References

[1] Owe Axelsson and Gunhild Lindskog. On the rate of convergence of the preconditioned conjugate gradient method. *Numer. Math.*, 48:499–523, 1986.

[2] A. Chronopoulos and C.W. Gear. s-step iterative methods for symmetric linear systems. *Journal of Computational and Applied Mathematics*, 25:153–168, 1989.

[3] E.F. D'Azevedo and C.H. Romine. Reducing communcation costs in the conjugate gradient algorithm on distributed memory multiprocessors. Technical Report ORNL/TM-12192, Oak Ridge National Lab, 1992.

[4] Jack J. Dongarra, Iain S. Duff, Danny C. Sorensen, and Henk A. van der Vorst. *Solving Linear Systems on Vector and Shared Memory Computers*. SIAM, Philadelphia, 1991.

[5] M. Engeli, M. Ginsburg, H. Rutishauser, and E. Stiefel. Refined iterative methods for the computation of the solution and the eigenvalues of self-adjoint boundary value problems. Technical Report 8, Mitt. Inst. Angew. Math. ETH Zürich, 1959.

[6] V. Faber and T. Manteuffel. Necessary and sufficient conditions for the existence of a conjugate gradient method. *SIAM J. Numer. Anal.*, 21:352–362, 1984.

[7] R. Fletcher. Conjugate gradient methods for indefinite systems. In G.A. Watson, editor, *Numerical Analysis Dundee 1975*, pages 73–89, New York, 1976. Springer Verlag.

[8] Roland W. Freund and Noël M. Nachtigal. QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Numer. Math.*, 60:315–339, 1991.

[9] Gene H. Golub and Dianne P. O'Leary. Some history of the conjugate gradient and Lanczos methods. *SIAM Review*, 31:50–102, 1989.

[10] Louis A. Hageman and David M. Young. *Applied Iterative Methods*. Academic Press, New York, 1981.

[11] M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Nat. Bur. Stand. J. Res.*, 49:409–436, 1952.

[12] Alston S. Householder. *The theory of matrices in numerical analysis*. Blaisdell Publishing Company, New York, 1964. republished by Dover Publications, New York, 1975.

[13] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research, Nat. Bu. Stand.*, 45:255–282, 1950.

[14] C. Lanczos. Solution of systems of linear equations by minimized iterations. *Journal of Research, Nat. Bu. Stand.*, 49:33–53, 1952.

[15] Gerard Meurant. Multitasking the conjugate gradient method on the CRAY X-MP/48. *Parallel Computing*, 5:267–280, 1987.

[16] C.C. Paige. Computational variants of the Lanczos method for the eigenproblem. *J. Inst. Maths. Applics.*, 10:373–381, 1972.

[17] C.C. Paige. Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. *J. Inst. Maths. Applics.*, 18:341–349, 1976.

[18] Yousef Saad. Practical use of some krylov subspace methods for solving indefinite and nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 5:203–228, 1984.

[19] Yousef Saad and Martin H. Schultz. Conjugate gradient-like algorithms for solving nonsymmetric linear systems. *Math. Comp.*, 44:417–424, 1985.

[20] Yousef Saad and Martin H. Schultz. GMRes: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[21] Peter Sonneveld. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 10:36–52, 1989.

[22] A. van der Sluis and H.A. van der Vorst. The rate of convergence of conjugate gradients. *Numer. Math.*, 48:543–560, 1986.

[23] J. Van Rosendale. Minimizing inner product data dependencies in conjugate gradient iteration. Technical Report 172178, ICASE, NASA Langley Research Center, Hampton, Virginia, 1983.

[24] Richard S. Varga. *Matrix Iterative Analysis*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1962.

[25] V.V. Voevodin. The problem of non-self-adjoint generalization of the conjugate gradient method is closed. *USSR Comput. Maths. Math. Phys.*, 23:143–144, 1983.