Overview of Krylov subspace methods with applications to control problems

Youcef Saad

Abstract

This paper gives an overview of projection methods based on Krylov subspaces with emphasis on their application to solving matrix equations that arise in control problems. The main idea of Krylov subspace methods is to generate a basis of the Krylov subspace Span$\{v, Av, \ldots, A^{m-1}v\}$, and seek an approximate solution to the original problem from this subspace. Thus, the original matrix problem of size N is approximated by one of dimension m, typically much smaller than N. Krylov subspace methods have been very successful in solving linear systems (Conjugate Gradients, GMRES,..) and eigenvalue problems (Lanczos, Arnoldi,..) and are now just becoming popular for solving nonlinear equations. We will show how they can be used to solve partial pole placement problems, Sylvester's and Lyapunov's equations.

1 Introduction

Krylov subspace methods have become a very useful and popular tool for solving large sets of linear and nonlinear equations, and large eigenvalue problems. One of the reason for their popularity is their simplicity and their generality. These methods have been increasingly accepted as efficient and reliable alternative to the more expensive methods that are usually employed for solving dense problems. This trend is likely to accelerate as models are becoming more complex and give rise to larger and larger matrix problems.

It is interesting to observe that, surprisingly, there has been very little done in developing algorithms to solve the very large matrix problems that arise in control. Yet, there are now applications in this area which lead to systems of equations involving very large sparse matrices. This is the case for example in models that involve partial differential equations in several space dimensions, or in applications related to large space structures [5]. Another typical example is that of electrical networks [4].

In [24] we proposed a method for partial pole placement, which consists of placing a few of the poles of the matrix, namely only those that are unstable. The methods proposed are based on projecting the problem onto a small invariant subspace of A associated with the unstable eigenvalues. Datta and Saad [9] considered several ways of solving special Sylvester equations and some related problems. Recently, we have considered a few numerical methods for solving large Lyapunov equations [25].

The purpose of this paper is to describe the general concepts used in Krylov subspace methods and to give an overview of the different ways in which they are used. As will be seen the method is fairly universal in that it can be used in various forms to provide solution methods for virtually any linear problem. However, the success of the method depends critically on the nature of the matrices at hand. For example conjugate gradient type methods are very successful in solving symmetric positive definite or nonsymmetric positive real linear systems but have been rather unsuccessful with highly indefinite problems.

The next section is a brief introduction to Krylov subspaces. Section 3 discusses the application of the method to linear systems, while section 4 is on eigenvalue problems. Section 5 will be on evaluating exponentials of A times a vector and some applications to Lyapunov equations.

2 Krylov subspaces

Given a square matrix A and a nonzero vector v, the subspace defined by

$$K_m \equiv span\ \{v, Av, A^2v, ...A^{m-1}v\} \tag{1}$$

is referred to as a the m-th Krylov subspace associated with the pair (A, v) and is denoted by $K_m(A, v)$ or simply by K_m if there is no ambiguity. We start by stating a few elementary properties of Krylov subspaces. Recall that the minimal polynomial of a vector v is the nonzero monic polynomial p of lowest degree such that $p(A)v = 0$. Clearly, the Krylov subspace K_m is the subspace of all vectors in $\mathbb{C}^N$ which can be written as $x = p(A)v$, where p is a polynomial of degree not exceeding $m - 1$.

Proposition 2.1 *The Krylov subspace K_m is of dimension m if and only if the degree of the minimal polynomial of v with respect to A is not less than m.*

In practice it is uncommon that the degree of the minimal polynomial is less than N, even in exact arithmetic. If this were to happen then it is usually helpful rather than harmful because of the following proposition.

Proposition 2.2 *Let μ be the degree of the minimal polynomial of v. Then K_μ is invariant under A and $K_m = K_\mu$ for all $m \geq \mu$.*

Thus, in case μ is small we can work work in subspace of dimension μ and be able to solve the problem exactly in this small subspace.

Working directly with the basis $\{A^j v\}_{j=0,\ldots,m-1}$ is likely to lead to serious numerical difficulties. Most Krylov subspace methods utilize either orthogonal or bi-orthogonal bases of K_m. Thus, the procedure introduced by Arnoldi [1] builds an orthogonal basis of the Krylov subspace K_m by the following algorithm.

Arnoldi's algorithm:

1. Start: Choose a vector v_1 of norm 1.

2. Iterate: for $j = 1, 2, \ldots, m$ compute,

$$h_{ij} = (Av_i, v_j) \quad i = 1, 2, \ldots, j \tag{2}$$

$$w = Av_j - \sum_{i=1}^{j} h_{ij} v_i \tag{3}$$

$$h_{j+1,j} = \|w\|_2 \tag{4}$$

$$v_{j+1} = w/h_{j+1,j} \tag{5}$$

This algorithm is mathematically equivalent to a Gram-Schmidt process applied to the power sequence $v, Av, ..., A^{m-1}v$, in that it would deliver the same sequence of v_i's *in exact arithmetic.* The algorithm will stop if the vector w computed in (4) vanishes which happens if the degree of the minimal polynomial for v is j. This is referred to a 'lucky' breakdown since as was seen above it means that the original problem (linear system, eigenvalue problem) can be solved exactly in a j-th dimensional subspace.

The following are a few simple but important properties satisfied by the algorithm.

Proposition 2.3 *The vectors $v_1, v_2, \ldots, v_m$ form an orthonormal basis of the subspace $K_m = span\{v_1, Av_1, \ldots, A^{m-1}v_1\}$.*

Proposition 2.4 *Denote by V_m the $N \times m$ matrix with column vectors $v_1, \ldots, v_m$ and by H_m the $m \times m$ Hessenberg matrix whose nonzero entries are defined by the algorithm. Then the following relations hold:*

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T \tag{6}$$

$$V_m^T A V_m = H_m \tag{7}$$

Note that when A is symmetric then (7) implies that the matrix H_m is tridiagonal symmetric and as a result Arnoldi's algorithm simplifies into an algorithm which involves only three consecutive vectors at each step. The corresponding algorithm is the well-known Lanczos algorithm.

The second of the relations in the proposition indicates that the Hessenberg matrix H_m is nothing but the matrix representation of the projection of A onto K_m, with respect to the orthogonal basis V_m. Analysis of various projection methods based on Krylov subspaces, indicate that, loosely speaking, K_m contains the most significant information of A, in that the outermost eigenvalues of A are well represented by those of its projection onto K_m, for large enough m. The main idea of Krylov subspace methods is to project the original problem into K_m. In the next sections we will see how this is done via simple Galerkin type procedures, for standard linear algebra problems.

The relation (6) has been exploited in [9] for solving special Sylvester's equations that arise in the design of reduced-dimensional state estimator. The Arnoldi and block-Arnoldi algorithms have been used in [6] to compute numerically the controllability of a linear system.

3 Krylov subspace methods for solving linear systems

Given an initial guess x_0 to the linear system

$$Ax = b, \tag{8}$$

a general *projection method* seeks an approximate solution x_m from an affine subspace $x_0 + K_m$ of dimension m by imposing the Petrov-Galerkin condition

$$b - Ax_m \perp L_m \tag{9}$$

where L_m is another subspace of dimension m. A Krylov subspace method is a method for which the subspace K_m is the Krylov subspace

$$K_m(A, r_0) = span\{r_0, Ar_0, A^2r_0, \ldots, A^{m-1}r_0\}, \tag{10}$$

in which $r_0 = b - Ax_0$. The different versions of Krylov subspace methods arise from different choices of the subspaces K_m and L_m and from the ways in which the system is preconditioned. The most common choices of K_m and L_m are the following.

1. $L_m = K_m = K_m(A, r_0)$. The conjugate gradient method is a particular instance of this method when the matrix is symmetric positive definite. Another method in this class is the Full Orthogonalization Method (FOM) [21] which is closely related to Arnoldi's method for solving eigenvalue problems [1]. Also in this class is ORTHORES [14], a method that is mathematically equivalent to FOM. Axelsson [2] also derived a similar algorithm for general nonsymmetric matrices.

As an example we outline here the FOM method for solving linear systems. Assume that we take $v_1 = r_0/\|r_0\|_2$ and run m steps of Arnoldi's method described in the previous section. Then, the approximate solution is of the form $x_0 + V_m y_m$ where y_m is some m-vector. The Galerkin condition (9) with $L_m = K_m$ gives immediately that $y_m = H_m^{-1}\|r_0\|_2 e_1$.

2. $L_m = AK_m; K_m = K_m(A, r_0)$. With this choice of L_m, it can be shown, see e.g., [26] that the approximate solution x_m minimizes the residual norm $\|b - Ax\|_2$ over all candidate vectors in $x_0 + K_m$. In contrast, there is no similar optimality property known for methods of the first class when A in nonsymmetric. Because of this, many methods of this type have been derived for the nonsymmetric case [3, 14, 11, 27]. The Conjugate Residual method [7] is the analogue of conjugate gradient method that is in this class. The GMRES algorithm [27] is an extension of the Conjugate Residual method to nonsymmetric problems.

3. $L_m = K_m(A^T, r_0); K_m = K_m(A, r_0)$. Clearly, in the symmetric case this class of methods reduces to the first one. In the nonsymmetric case, the biconjugate gradient method (BCG) due to Lanczos [15] and Fletcher [12] is a good representative of this class. There are various mathematically equivalent formulations of the biconjugate gradient method [22], some of which are more numerically viable than others. An efficient variation on this method, called CGS (Conjugate gradient squared) was proposed by Sonneveld [28].

Apart from the above three basic methods there are a number of techniques for nonsymmetric problems that are mathematically equivalent to solving the normal equations $A^TAx = A^Tb$ or $AA^Ty = b$ by the conjugate gradient method. We will comment that these methods are often too quickly dismissed as inferior because of the fact that the condition number of the original problem is squared. For problems that are strongly indefinite they do represent, however, the only viable alternative, since none of the above three types of methods would work in this situation.

One of the possible applications of the methods described here is in the frequency response calculations in input-output analysis. For example, in the single input single output case, one needs to compute $c(A - j\omega I)^{-1}b$ for many values of ω. We observe that the Krylov subspaces are invariant under arbitrary shifts to the matrix A, i.e., $K_m(A, v) = K_m(A - sI, v)$ for any s. This suggests using the same Krylov subspace $K_m(A, v)$ to get approximations to all of the solution vectors $(A - j\omega I)^{-1}b$ via the formula

$$x_m(\omega) = \beta V_m(H_m - j\omega I)^{-1}e_1 \tag{11}$$

where we have set $\beta = \|b\|$ and where we assume that the Arnoldi algorithm is started with $v_1 = b/\|b\|_2$, i.e., $x_0 = 0$. This technique was suggested in [10] and can be regarded as an extension of the earlier technique proposed by Laub in [16] for dense problems. Numerical experiments are currently being performed.

An important factor in the success of conjugate gradient-like methods is the preconditioning technique. This typically consists of replacing the original linear system (8) by, for example, the equivalent system

$$M^{-1}Ax = M^{-1}b \tag{12}$$

In the classical case of the incomplete LU preconditionings, the matrix M is of the form $M = LU$ where L is a lower triangular matrix and U is an upper triangular matrix such that L and U have the same structure as the lower and upper triangular parts of A respectively. In the general sparse case, the incomplete factorization is obtained by performing the standard LU factorization of A and dropping all fill-in elements that are generated during the process. This is referred to as ILU(0), or IC(0) in the symmetric case.

4 Krylov subspace methods for eigenvalue problems

An idea that is basic to sparse eigenvalue calculations is that of projection processes [23]. Given a subspace K spanned by a system of m orthonormal vectors $V \equiv [v_1, \ldots, v_m]$ a projection process onto $K \equiv span\ \{V\}$ computes an approximate eigenpair $\tilde{\lambda} \in \mathbb{C}, \tilde{u} \in K$ that satisfy the Galerkin condition,

$$(A - \tilde{\lambda}I)\tilde{u} \perp K \tag{13}$$

The approximate eigenvalues $\tilde{\lambda}$ are the eigenvalues of the $m \times m$ matrix $C = V^TAV$. The corresponding approximate eigenvectors are the vectors $\tilde{u}_i = Vy_i$ where y_i are the eigenvectors

of C. Similarly, the approximate Schur vectors are the vector columns of VU, where $U = [u_1, u_2, \ldots, u_m]$ are the Schur vectors of C, i.e., $U^T C U$ is quasi-upper triangular. Thus, one possible method for computing eigenvalues/ eigenvectors of large sparse matrices is to use the Arnoldi process [1, 20] which is a projection process onto $K_m = span\{v_1, Av_1, \ldots, A^{m-1}v_1\}$. Once the Arnoldi vectors $v_1, ..., v_m$ have been generated we can use V_m for a projection process onto K_m. The matrix $V_m^T A V_m$ which is needed for this purpose is nothing but the upper Hessenberg matrix H_m generated by the algorithm.

Note that the Arnoldi algorithm utilizes the matrix A only to compute successive matrix by vector products $w = Av$, so sparsity can be exploited. As m increases, the eigenvalues of H_m that are located in the outermost part of the spectrum start converging towards corresponding eigenvalues of A. However, the difficulty with the above algorithm is that as m increases cost and storage increase rapidly. One solution is to use the method iteratively: m is fixed and the initial vector v_1 is taken at each new iteration as a linear combination of some of the approximate eigenvectors. Moreover, there are several ways of accelerating convergence by preprocessing v_1 by a Chebyshev iteration before restarting, i.e., by taking $v_1 = t_k(A)z$ where z is again a linear combination of eigenvectors.

A technique related to Arnoldi's method is the nonsymmetric Lanczos algorithm [19, 8] which produces a nonsymmetric tridiagonal matrix instead of a Hessenberg matrix. Unlike Arnoldi's process, this method requires multiplications by both A and A^T at every step. On the other hand it has the big advantage of requiring little storage (5 vectors). Although no comparisons of the performances of the Lanczos and the Arnoldi type algorithms have been made, the Lanczos methods are usually recommended whenever the number of eigenvalues to be computed is large.

Finally, if the matrix is banded an efficient solution is the shift and invert strategy which consists of using one of the above iterative methods (subspace iteration, Arnoldi, or Lanczos) for the matrix $(A-\sigma I)^{-1}$, where σ is some shift chosen say at the center of some small region of the complex plane where eigenvalues are sought. The matrix $(A-\sigma I)^{-1}$, need not be explicitly computed: all we need is to factor $(A - \sigma I)$ into LU and subsequently at each step of the iterative method solve two triangular systems one with L and the other with U. Thus band structure can be fully exploited. In [18] several implementations of the shift and invert strategy are considered and the problem of avoiding complex arithmetic when A is real is addressed.

An important application of the eigenvalue algorithms is to provide a small invariant subspace that will represent the critical modes of the system. For example unstable modes, assuming that there are just a few of them, can then be displaced by a technique described in [24]. The main idea in [24] is now summarized.

Let A be an N x N real nonsymmetric matrix whose eigenvalues are

$$\lambda_1, \lambda_2, \ldots, \lambda_k, \lambda_{k+1}, \ldots, \lambda_N$$

and b be a given real vector. The problem considered is to find a vector f so that the modified matrix

$$M = A - bf^T$$

has the given eigenvalues

$$\mu_1, \mu_2, \ldots, \mu_k, \lambda_{k+1}, \ldots, \lambda_N.$$

In other words we would like to assign the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$ of A into $\mu_1, \mu_2, \ldots, \mu_k$, while leaving the rest of the spectrum of A unchanged, with the rank one perturbation $-bf^T$. This is referred to as the *partial pole assignment problem.*

To solve this problem we assume that we have computed the partial Schur factorization for A^T:

$$A^T Q = QR,$$

where Q is an N x k matrix whose columns form an orthonormal basis of the left invariant subspace associated with $\lambda_i, i = 1, \ldots k$ and R is a k x k upper quasi-triangular matrix. We then seek a solution f in the form $f = Qg$. Denoting by s the vector $s = Q^T b$, the matrix $M^T Q$ is such that,

$$M^T Q = \left[A^T - fb^T\right] Q = QR - Qgb^T Q = Q\left[R - gs^T\right] = Q\left[R^T - sg^T\right]^T$$

The above equation means that the choice $f = Qg$ makes the subspace spanned by Q also invariant under M^T. Moreover, the eigenvalues of the matrix M associated with this invariant subspace are the eigenvalues of the k x k matrix $C_k = R^T - sg^T$. Thus it suffices to assign the eigenvalues of this small matrix to be $\mu_i, i = 1, \ldots k$, by an appropriate choice of the vector g. It was shown in [24] that this solves the problem and that the solution is feasible under some mild conditions on b and the μ_i's. Nichols [17] proposed improvements on this scheme to provide robust partial pole placement techniques. In [9] alternative techniques for partial pole placement were derived from a special technique to solve Sylvester's equation.

5 Approximations to $e^A v$ and applications

Computing approximations to the exponential of a matrix is usually not too hard a problem for small dense matrices. For large matrices, this can be rather difficult to do because of the fact that e^A will in general be a dense matrix even when A is very sparse. However, it is often not the exponential of the matrix that is sought but its product with some vector v. The question of approximating $e^A v$ for any given vector v was considered in [13] where polynomial and rational approximations to the exponential were used. Here we summarize only the method proposed in [13] that is based on polynomial approximation to $e^A v$:

$$e^A v \approx p_{m-1}(A)v \tag{14}$$

where p_{m-1} is a polynomial of degree $m-1$. Thus, the above approximation is an element of the Krylov subspace (1) and it is convenient to express it in the orthonormal basis $V_m = [v_1, v_2, v_3, \ldots, v_m]$ generated by Arnoldi's algorithm seen earlier. We can write the desired approximation to $x = e^A v$ as $x_m = p_m(A)v$ or equivalently $x_m = V_m y$ where y is an m-vector. In [13], the choice $y = \beta e^{H_m} e_1$ with $\beta = \|v\|_2$ was suggested, leading to the following formula for arbitrary t,

$$e^A v \approx \beta V_m e^{H_m} e_1 \tag{15}$$

The quality of this approximation was also analyzed in [13] and the following result was shown.

Theorem 5.1 *Let A be any square matrix and let $\rho = \|A\|_2$. Then the error of the approximation (15) is such that*

$$\|e^A v - \beta V_m e^{H_m} e_1\|_2 \leq 2\beta \frac{\rho^m e^\rho}{m!}. \tag{16}$$

Experiments reported in [13], reveal that this approximation can be very accurate even for moderate values of the degree m. The theorem shows convergence of the approximation (15) for fixed t, as m increases to ∞. However, note that the above approximation is exact when $m = N$, see [13].

One application of the above formulas is that one can approximate $e^{tA}v$ for all t as

$$e^{tA}v \approx \beta V_m e^{tH_m} e_1. \tag{17}$$

This provides a way of solving ordinary differential equations of the type $\dot{x} = Ax + b$ which was the original problem considered in [13]. Moderate degrees can provide reasonably high accuracy in the solutions. Another direct application is described in [25], where the controllability Grammian,

$$X = \int_0^\infty e^{\tau A} b b^T e^{\tau A^T} d\tau. \tag{18}$$

was approximated by replacing the function $e^{\tau A}b$ by its approximation (17). Note that X is known to be the solution of the Lyapunov equation

$$AX + XA^T + bb^T = 0. \tag{19}$$

A rather unexpected result shown in [25] is that the approximation provided by the above integration process is nothing but a Galerkin method applied to (19) over the subspace of matrices of the form $V_m G V_m^T$, where V_m is fixed and G runs over the set of $m \times m$ matrices.

6 Conclusion

The techniques described in the previous section, and other recent developments elsewhere, suggest that in many of the problems in control, one can work in a Krylov subspace of reasonably small dimension. For example, although the theory is not established for this case, the above approach for Lyapunov equations can be extended to Riccati's equation using a Galerkin point of view. Many of the optimization techniques in control can be carried out by replacing the full n-dimensional variable $x(t)$ by an approximation derived from replacing x by its m dimensional approximation (17). This may open up interesting new approaches and theoretical questions as to the accuracy of the corresponding approximations.

References

[1] W. E. Arnoldi. The principle of minimized iteration in the solution of the matrix eigenvalue problem. *Quart. Appl. Math.*, 9:17–29, 1951.

[2] O. Axelsson. Conjugate gradient type-methods for unsymmetric and inconsistent systems of linear equations. *Linear Algebra Appl.*, 29:1–16, 1980.

[3] O. Axelsson. A generalized conjugate gradient, least squares method. *Num. Math.*, 51:209–227, 1987.

[4] P. Kokotovic B. H. Krogh. Feedback control of overloaded networks. *IEEE Trans. Aut. Cont.*, AC-29:704–711, 1984.

[5] M. J. Balas. Trends in large space structure control theory: fondest dreams, wildest hopes. *IEEE Trans. Aut. Contr.*, AC-2:522–535, 1982.

[6] D. Boley and G. H. Golub. The Lanczos-Arnoldi algorithm and controllability. *Systems and Control Letters*, 4:317–324, 1987.

[7] R. Chandra. *Conjugate Gradient Methods for Partial Differential Equations.* PhD thesis, Yale University, Computer Science Dept., New Haven, CT. 06520, 1978.

[8] J. Cullum and R. Willoughby. A Lanczos procedure for the modal analysis of very large nonsymmetric matrices. In *Proceedings of the 23rd Conference on Decision and Control, Las Vegas*, 1984.

[9] B. N. Datta and Y. Saad. Arnoldi methods for large Sylvester-like observer matrix equations. in preparation.

[10] B. N. Datta and Y. Saad. Solution of large linear algebra problems in control. Presentation at the 1986 SIAM meeting on Linear Algebra and its Applications, Boston Mass. , Aug 1986.

[11] H. C. Elman. *Iterative Methods for Large Sparse Nonsymmetric Systems of Linear Equations.* PhD thesis, Yale University, Computer Science Dept., New Haven, CT., 1982.

[12] R. Fletcher. Conjugate gradient methods for indefinite systems. In G. A. Watson, editor, *Proceedings of the Dundee Biennal Conference on Numerical Analysis 1974*, pages 73–89, New York, 1975. University of Dundee,Scotland, Springer Verlag.

[13] E. Gallopoulos and Y. Saad. On the parallel solution of parabolic equations. In R. De Groot, editor, *Proceedings of the International Conference on Supercomputing 1989, Heraklion, Crete, June 5-9, 1989.* ACM press, 1989.

[14] K. C. Jea and D. M. Young. Generalized conjugate gradient acceleration of nonsymmetrizable iterative methods. *Linear Algebra Appl.*, 34:159–194, 1980.

[15] C. Lanczos. Solution of systems of linear equations by minimized iterations. *J. Res. Nat. Bur. Standards*, 49:33–53, 1952.

[16] A. J. Laub. Robust stability of linear systems- some computational considerations. In C. Wilde et al. , editor, *Information linkages between applied mathematics and industry.* Academic Press, 1980.

[17] N. K. Nichols. Robustness in partial pole placement. *IEEE Trans. Aut. Cont.*, 32:728–732, 1987.

[18] B. N. Parlett and Y. Saad. Complex shift and invert strategies for real matrices. *Linear Algebra and its Applications*, 88/89:575–595, 1987.

[19] B. N. Parlett, D. R. Taylor, and Z. S. Liu. A look-ahead Lanczos algorithm for nonsymmetric matrices. *Mathematics of Computation*, 44:105–124, 1985.

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

[21] Y. Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of Computation*, 37:105–126, 1981.

[22] Y. Saad. The Lanczos biorthogonalization algorithm and other oblique projection methods for solving large unsymmetric systems. *SIAM J. Numer. Anal.*, 19:470–484, 1982.

[23] Y. Saad. Projection methods for solving large sparse eigenvalue problems. In B. Kagstrom and A. Ruhe, editors, *Matrix Pencils, proceedings, Pitea Havsbad*, pages 121–144, Berlin, 1982. University of Umea, Sweden, Springer Verlag. Lecture notes in Math. Series, Number 973.

[24] Y. Saad. Projection and deflation methods for partial pole assignment in linear state feedback. *IEEE Trans. Aut. Cont.*, 33:290–297, 1988.

[25] Y. Saad. Numerical solution of large Lyapunov equations. In M. A. Kaashoek, J. H. van Schuppen, and A. C. Ran, editors, *Signal Processing, Scattering, Operator Theory, and Numerical Methods. Proceedings of the international symposium MTNS-89, vol III*, pages 503–511, Boston, 1990. Birkhauser.

[26] Y. Saad and M. H. Schultz. Conjugate gradient-like algorithms for solving nonsymmetric linear systems. *Mathematics of Computation*, 44(170):417–424, 1985.

[27] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7:856–869, 1986.

[28] P. Sonneveld. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Scient. Statist. Comput.*, 10(1):36–52, 1989.