

Inexact Newton Preconditioning Techniques for Eigenvalue Problems*

Kesheng Wu[†] Yousef Saad[‡] Andreas Stathopoulos[§]

March 11, 1998

Abstract

The focus of this paper is on numerical methods for finding a few eigenvalues and eigenvectors of a large sparse matrix. New preconditioning schemes are proposed for improving the effectiveness of a few methods for computing eigenvalues and eigenvectors. The basic framework of the preconditioned eigenvalue methods we consider is that of the Arnoldi method and the related Davidson method. Within this framework, it is possible to unravel new and more effective alternatives by varying the right-hand side and the matrix of the preconditioning equation. This paper first studies the effects of selecting various such right-hand sides. These comparisons, indicate that a scheme based on the inexact-Newton method outperforms the others. We further study a number of Newton schemes for eigenvalue problems and test their potential as preconditioners. The experiments reveal that two schemes related to the Newton preconditioning can constitute good alternatives to other commonly used schemes.

1 Introduction

Many scientific and engineering applications requires the solution of large eigenvalue problems, $Ax = \lambda x$. One of the goals of this paper, is to explore ways of enhancing the performance of eigenvalue methods by improving the preconditioning schemes. The term *preconditioning* is often associated with iterative solution of a large linear system of equations, $Ax = b$ and refers to an operation which enhance the convergence rate of the linear system solvers. The preconditioned solvers compute the solution of one of the following equations instead of the original one [4, 39],

$$M^{-1}Ax = M^{-1}b, \quad AM^{-1}(Mx) = b, \quad M_L^{-1}AM_R^{-1}(M_Rx) = M_L^{-1}b.$$

With appropriate choices of preconditioners, M or $M_L M_R$, the linear system solvers converge faster on the above equations than on the original equation. Research on preconditioners for linear systems continues to be very active [1, 3, 22, 28, 37, 55]. One motivation of this paper is to take advantage of this research and apply it for solving eigenvalue problems.

*Work supported by NSF/DMR-95 25885 and by the Minnesota Supercomputer Institute.

Most of the successful iterative linear system solvers are ‘preconditioned’ projection methods, which build ‘Krylov’ bases from the iteration matrix, $M^{-1}A$, AM^{-1} , or $M_L^{-1}AM_R^{-1}$, then perform projection to generate approximate solutions. Preconditioned projection methods for eigenvalue problems have been defined similarly. One simple scheme is simply use the same bases built by the preconditioned linear system solvers in these eigenvalue methods. Because the eigenvalues and eigenvectors of the preconditioned matrices

$$M^{-1}A, \quad AM^{-1}, \quad \text{or} \quad M_L^{-1}AM_R^{-1}$$

are different from those of A in general, the bases built for solving linear systems will not be effective for eigenvalue problems. We observe that one crucial operation in these preconditioning schemes is the solution of an auxiliary linear system, $Mz = s$. The preconditioner M , or $M_L M_R$, is selected to approximate A when solving linear systems. In this case, we regard the *exact* auxiliary system as $Az = s$. The process of solving this system often (but not always) consists of replacing A by an approximation M . We refer to this auxiliary equation as the *preconditioning equation*. Research on preconditioning linear systems focuses on generating an approximate solution z of this preconditioning equation. Some of the common strategies include applying iterative methods on the preconditioning equation [2, 19, 35], constructing an approximation to A that is easy to invert [1, 3, 37, 55], and constructing an approximation to the inverse of A [9, 20, 22]. This paper will attempt to exploit results of this research and apply them to precondition eigenvalue problems. One of our first goals for this purpose is to identify an effective preconditioning equation for eigenvalue problems.

Davidson’s method is one of the best known preconditioned eigenvalue solution methods. Its original version employs the following preconditioning step [11],

$$\text{diag}(A - \lambda I)z = r,$$

where r is the residual vector of the current approximate solution (λ, x) , $r = Ax - \lambda x$. Morgan and Scott [29] were among the first researchers to generalize this to the following form,

$$(A - \lambda I)z = r. \tag{1}$$

They then adapted standard preconditioning techniques developed for linear systems for solving the above equation. The matrix in the above preconditioning equation is the same as the one used in the Rayleigh-Quotient Iterations (RQI) and the shift-and-inverted Lanczos algorithm. However, a crucial difference is that the Davidson method does not require Equation (1) to be solved exactly. In the original Davidson method, the diagonal of $(A - \lambda I)$ is taken as an approximation to the whole matrix [11]. Later, incomplete LU factorization schemes were used [12, 29]. In recent years, a number of modifications to the preconditioning Equation (1) have been developed. For example, Olsen et al. [31] proposed to modify the right-hand side of Equation (1) to make the solution z orthogonal to computed approximate eigenvectors x . Stathopoulos and colleagues have shown that using a biased shift (see Section 2) δ in the preconditioning matrix, $A - \delta I$, can significantly reduce the number of iterations [46]. More recently, the following Jacobi-Davidson preconditioning has attracted much attention [6, 15, 16, 41, 42, 43].

$$(I - xx^T)(A - \lambda I)(I - xx^T)z = r, \tag{2}$$

in which again, x is a current approximation to the eigenvector being computed. The Jacobi-Davidson preconditioning can often significantly outperform the original Davidson scheme.

Other preconditioned eigenvalue methods include the preconditioned Arnoldi method [27, 36, 51], the preconditioned subspace techniques [7, 21, 40], and the inexact rational Krylov method [24]. All these eigenvalue methods generate an basis first, then use the Rayleigh-Ritz projection on the basis to obtain the desired eigenvalue approximation. Most of them build orthonormal basis progressively. A simple way of generalizing the Davidson method to encompass all the above methods is to append the results of preconditioning to augment the current basis. In this framework, the differences among the different methods include different restarting schemes and different preconditioning schemes. The process of restarting can have a strong influence on the overall effectiveness of the eigenvalue method [23, 45, 48]. However, the focus of this paper will be on the preconditioning aspects only. Different preconditioners can be obtained by using different right-hand sides and different matrices in the preconditioning equation. We will compare the results of using a number of different preconditioning schemes. From the comparisons, we would like to identify what underlying mechanism might be crucial to their effectiveness and explore the possibility of developing new preconditioning equations.

This paper is organized as follows. Section 2 describes a number of variations which use different right-hand sides in the preconditioning equation. The comparisons of this section lead us to single out inexact Newton-type methods for further investigation. Section 3 discusses a number of Newton methods for eigenvalue problems. In Section 4, we turn the Newton methods into preconditioners for the Davidson method and study how well they work on actual problems. A brief summary is presented in Section 5.

2 Right-hand sides for the preconditioning equation

The eigenvalue methods studied here have two logically distinct parts, one to generate a basis V and the other to project the original eigenvalue problem onto the basis and generate approximate solutions [32, 53]. The Rayleigh-Ritz projection method is used to project a large eigenvalue problem onto a small basis [8, 18, 32, 36, 50]. The preconditioning equation is used to introduce new vectors to the basis. For convenience, the basis $V = [v_1, \dots, v_m]$ used in the Rayleigh-Ritz projection will be an orthonormal basis. The following algorithm shows the essence of the basis generation process. The goal of the algorithm is to generate an orthonormal basis V_m from a set of k input vectors. The basis is also required to be orthogonal to another orthonormal set X . In the algorithm, V_j represents the matrix consisting of the column vectors v_1, \dots, v_j . In addition, $x = MGSA(W, r)$ is the vector resulting from orthonormalizing the vector r against the column vectors of W which are assumed to be already orthonormal. In the algorithm, m is a user defined maximum size for the basis and β , the block size, is another parameter that can be set by the user and which can be varied dynamically.

ALGORITHM 1 Algorithm for generating an orthonormal basis

Input: k input vectors $[w_0, \dots, w_k]$. Algorithm parameters: m and β and an existing orthonormal set X .

Output: orthonormal V_m , which is also orthogonal to X .

0. Set $v_i = w_i$, for $i = 1, \dots, k$. Orthonormalize the v_i 's and orthogonalize them against X . Let $j = k$.
1. Compute β vectors, s_1, \dots, s_β , to be processed by the preconditioner.
2. Solve the preconditioning equations: $Mv_{j+i} = s_i$ for $i = 1, \dots, \beta$.
3. Orthonormalization. For $i = 1, \dots, \beta$ do: $v_{j+i} := MGSA([X, V_{j+i-1}], v_{j+i})$
4. $j := j + \beta$. If $j + \beta < m$, go to step 1, if $j < m$, let $\beta = m - j$, go to step 1, else stop.

The matrix X can be a set of converged eigenvectors or external constraints to the eigenvalue problem. It is included to enable the explicit deflation, a process of removing the converged eigenvectors from basis V .

In practice, m is chosen so that the basis can be stored in the main memory of the computer. In most cases β can simply be equal to one. It may be set to a larger value to increase the effectiveness in case of eigenvalue clusters or to take advantage of effective block operations on certain architectures. The matrix M appearing in step 2 is commonly referred to as the preconditioner. In most preconditioned eigenvalue methods, the preconditioner is an approximation of $A - \delta I$ where the shift δ approximates the eigenvalue being computed. In the numerical examples we will present later, only symmetric matrices are used. Because Ritz values approach their limit from inside the spectrum, we can under- or over-estimate the eigenvalue depending on whether the smallest or the largest eigenvalue is wanted. When computing the smallest eigenvalues we define:

$$\delta = \lambda - \epsilon_\lambda,$$

where ϵ_λ is an estimate of the error on the Ritz value λ . For largest eigenvalues the minus sign in the above equation is replaced by a plus. Experiments show that this biased estimate of the eigenvalue is more effective as a shift for the preconditioners than the Ritz value [46, 53]. This biased estimate is used as shift in all reported test cases on preconditioning. Comparison results on nonsymmetric eigenvalue problems can be found elsewhere [25].

The above algorithm starts with k initial vectors and increment the basis size by β vectors at a time. When the basis size reaches m and the eigenvalue approximations are not satisfactory, the basis is compressed to a smaller one and the above algorithm is repeated. This process of restarting is a crucial part of an eigenvalue routine [5, 45, 47]. Since it is not the main concern of this paper, we will only use a very simple scheme. In fact for all tests, we set $k = m/2$, which is a good choice for most test problems, even though it is not optimal [47].

In step 2 of the above algorithm, the right-hand sides of the preconditioning equation are called s_i to suggest that they do not have to be residual vectors as in the Davidson method. To simplify the discussion, assume that $\beta = 1$, so there is only one right-hand side say, s . In the Arnoldi method, s is Av_j at step j . In the Davidson method, the vector s is the residual vector r .

| | |
|------------------------|-----------------------|
| Davidson | r |
| Arnoldi | Av_j |
| Cayley-Arnoldi | $(A - \lambda I)v_j$ |
| Orthogonalized Arnoldi | $(I - V_j V_j^T)Av_j$ |

Table 1: The right-hand sides of four preconditioning schemes.

| preconditioning | Arnoldi | Orthogonalized Arnoldi | Cayley- Arnoldi | Davidson |
|-----------------|---------|---------------------------|--------------------|----------|
| (NONE) | 18 | 18 | 18 | 18 |
| diagonal | 9 | 10 | 9 | 28 |
| SOR | 6 | 15 | 5 | 34 |
| ILU0 | 5 | 13 | 3 | 29 |
| ILUTP | 4 | 15 | 2 | 27 |

Table 2: Number of eigenvalue problems solved by each method.

Extensive tests have been conducted for a number of different choices for s [53]. Here, we will recall the results of four schemes: the Arnoldi method, the Davidson method, the orthogonalized Arnoldi method [51], and the Cayley-Arnoldi method [27]. The different right-hand sides used in the corresponding preconditioning equations are shown in Table 1. The rationale for the orthogonalized Arnoldi method is that the right-hand side used by the Davidson method is orthogonal to the basis. In the orthogonalized Arnoldi method, s is explicitly made orthogonal to the current basis. The cost of the orthogonalization is about half as high as that of computing the residual r . Thus, the orthogonalized Arnoldi method has the same orthogonality property as the Davidson method but it is slightly less expensive. The Cayley-Arnoldi scheme builds a basis for the Krylov subspace with the matrix $M^{-1}(A - \lambda I)$. If λ is an eigenvalue of A , then zero is an eigenvalue of $M^{-1}(A - \lambda I)$. In addition, the corresponding eigenvectors are identical [27]. This is an advantage over the Arnoldi method because usually $M^{-1}A$ and A have no eigenvector in common.

| preconditioning | Arnoldi | Orthogonalized Arnoldi | Cayley- Arnoldi | Davidson |
|-----------------|---------|---------------------------|--------------------|----------|
| (NONE) | 16 | 2 | 0 | 0 |
| diagonal | 0 | 0 | 0 | 28 |
| SOR | 0 | 0 | 0 | 34 |
| ILU0 | 1 | 1 | 0 | 27 |
| ILUTP | 0 | 2 | 0 | 26 |

Table 3: Number of cases where the method reached convergence in the least amount of time.

To compare the four preconditioning schemes, we applied them on 45 non-diagonal symmetric matrices in the Harwell-Boeing collection [14]. We used each one of them to form an standard eigenvalue problem, then compute 5 smallest eigenvalues and the corresponding eigenvectors of each problem. All methods start with the same initial guess, a vector of all ones: $[1, 1, \dots, 1]^T$. Only one initial guess is used so that comparisons can be made with ARPACK [44] to validate the implementation of the methods. A Ritz pair is considered to have converged if its residual norm is less than $10^{-12}\|A\|_F$. The timing results are obtained on a SPARC-10 workstation. The preconditioning equations are solved with selected schemes from SPARSKIT: diagonal preconditioner, SOR, ILU0 and ILUTP [34]. The diagonal preconditioner only inverts the diagonal of $(A - \lambda I)$ as in the original Davidson scheme. The SOR preconditioner solves the preconditioning equation with one iteration of Gauss-Seidel iteration which is a special case of the Successive Over-Relaxation (SOR) method. ILU0 is an incomplete factorization where the LU factors have the same nonzero pattern as the original matrix [4, 28, 39]. ILUTP modifies ILU0 in three ways, (1) elements of LU factors with small absolute values are dropped, (2) a maximum number of nonzero elements in a row of LU factors, i.e., level of fill, is controlled by the user, (3) column pivoting is performed if the diagonal element is significantly smaller than another element on the same row [38]. Generally, the preconditioners are considered more effective for linear systems toward the bottom of the Tables 2 and 3, i.e., ILUTP is better than ILU0, which is in turn better than SOR and diagonal preconditioning [52].

Table 2 sums up the total number of eigenvalue problems solved with each method. Without preconditioning, the four methods converged on the same problems. With preconditioning, the Davidson method converges on more problems than any other method. In fact, it is the only one that solves more problems with preconditioning than without preconditioning. Table 3 counts the number of cases where the method heading the column converged in the least amount of time. In the unpreconditioned case, the Arnoldi method uses the least amount of time to reach convergence. This is expected since the four methods are mathematically equivalent and the Arnoldi method uses the least amount of arithmetic operations per step. Without preconditioning, there are two examples where the orthogonalized Arnoldi method converges faster than the regular Arnoldi method. In this case the difference between the two methods is that the orthogonalized Arnoldi method performs one more orthogonalization than for the regular Arnoldi method. The test results indicate that this extra orthogonalization helps the Arnoldi method in some cases. However, in most cases, the extra orthogonalization only increase the overall execution time. The Lanczos method is also equivalent to the unpreconditioned Arnoldi method and it uses even fewer arithmetic operations per step. If an eigenvalue problem can be solved without preconditioning, the Lanczos method is a more efficient method than the Arnoldi method [32]. Because the Lanczos method cannot use the preconditioning scheme described here, it is not discussed further.

In the preconditioned cases, the Davidson method reaches convergence faster than most others no matter what preconditioner is used. Since the spectrum of $M^{-1}A$ can differ arbitrarily from that of A , we can accept that the preconditioned Arnoldi method is not more effective than its unpreconditioned counterpart. The right-hand sides of the preconditioning equation of both the orthogonalized Arnoldi method and the Davidson method are orthog-

onal to their bases. Table 2 shows that the orthogonalized Arnoldi converged on much less problems than the Davidson method, which indicates that this orthogonality is not the dominant factor in the overall effectiveness of the Davidson method. If both the preconditioner M and the shift λ are fixed, then the Cayley-Arnoldi method and the Davidson method produce the same Krylov subspace $K(M^{-1}(A - \lambda I))$. However, because the shift λ is in fact varying, the performance of the Cayley-Arnoldi method and the Davidson method are markedly different. Among the various schemes tested, using the residual vector as the right-hand side of the preconditioning equation is the only effective choice for the eigenvalue problems.

There are a number of implementation details that may alter the overall performance of the methods tested. However, the data shown in Tables 2 and 3 are representative of their relative strength [53].

3 Newton methods for eigenvalue problems

In Davidson's original paper, the proposed eigenvalue method was given as a combination of the Lanczos method and the Newton method for minimizing the Rayleigh quotient [11]. The first two derivatives of the Rayleigh quotient, $\lambda = x^T Ax / x^T x$, are as follows,

$$\begin{aligned} \frac{d\lambda}{dx} &= \frac{2Ax}{x^T x} - \frac{2x^T Ax x}{(x^T x)^2} \\ \frac{d^2\lambda}{dx^2} &= \frac{2A}{x^T x} - \frac{2(2Axx^T + x^T AxI + 2xx^T A)}{(x^T x)^2} + \frac{8x^T Ax xx^T}{(x^T x)^3} \end{aligned}$$

Since the second derivative of the Rayleigh quotient is complicated, Davidson proposed a simple approximation to the Newton recurrence, namely $\text{diag}(A - \lambda I)z = r$. Later on, the proper preconditioning equation was generally considered to be $(A - \lambda I)z = r$ [10, 12, 29, 30]. A number of potential problems have been observed [53]. For example, if the preconditioning equation is actually solved accurately, the solution is $z = x$, which will cause the Davidson method to stagnate. This indicates that the preconditioners for the eigenvalue problems should not solve the preconditioning equation accurately which is contrary to the requirement on preconditioners for linear systems. A simple solution to this dilemma is to find other formulations of the preconditioning equation for the eigenvalue problems where accurate solutions can lead to faster convergence. Following Davidson, we start by examining a few Newton methods for eigenvalue problems.

If the exact eigenvalue λ^* is known, finding the corresponding eigenvector can be achieved by applying the following Newton recurrence to solve equation $r = (A - \lambda^* I)x = 0$,

$$x_{i+1} = x_i - (A - \lambda^* I)^{-1} r_i, \tag{3}$$

where $r_i = (A - \lambda^* I)x_i$. The Davidson preconditioning can be regarded an approximate form of this Newton iteration. The biased estimate may also be regarded as an attempt to estimate λ^* based on the Ritz value [46]. The above equation is also known as the correction equation since it was first used to refine the eigenvectors found by other means [13].

Since λ^* is an exact eigenvalue, the Jacobian matrix $(A - \lambda^*I)$ is singular, and the above Newton recurrence is not well defined. To correct this, the following recursion can be used instead of Equation (3),

$$x_{i+1} = x_i - (A - \lambda^*I)^+ r_i, \quad (4)$$

where the superscript $+$ indicates a pseudoinverse [18]. By the definition of pseudoinverses, $AA^+A = A$. The change in x_i still satisfy the correction equation.

$$\begin{aligned} (A - \lambda^*I)(x_i - x_{i+1}) &= (A - \lambda^*I)(A - \lambda^*I)^+ r_i \\ &= (A - \lambda^*I)(A - \lambda^*I)^+(A - \lambda^*I)x_i \\ &= (A - \lambda^*I)x_i \equiv r_i. \end{aligned}$$

If the initial guess x_0 is not orthogonal to the exact eigenvector x^* , then x_1 computed by Equation (4) is a non-trivial solution of $r = 0$. In other word, x_1 is the wanted eigenvector.

The above argument indicates that solving the correction equation using pseudoinverses, see Equation (4), has a significant advantage over the alternative, see Equation (3). In practice, we don't know the exact eigenvalue, therefore $(A - \lambda I)$ is generally not singular, and thus $(A - \lambda I)^+ = (A - \lambda I)^{-1}$. We continue to face the same problems as before. Since the pseudoinverse of a matrix is different from the regular inverse only if the matrix is singular, we could force the matrix to be singular as is done in the Jacobi-Davidson method. Even though the Jacobi-Davidson method was not derived based on this idea, this constitutes a valid interpretation. In the Jacobi-Davidson method, a Krylov subspace method is used to solve Equation (2). The Krylov subspace method approximates pseudoinverse by computing a solution in the range of the iteration matrix [4]. One difference between Equation (3) and Equation (4) is that the later one generates a correction that is orthogonal to the exact eigenvector x^* , i.e., the correction is in the range of the iteration matrix $A - \lambda^*I$. Both the Olsen preconditioning scheme [31] and the Jacobi-Davidson preconditioning scheme [42] generate orthogonal corrections. They can viewed as ways of mimicking the Newton recurrence with pseudoinverse. An alternative strategy to address the same issue is to find a Newton recurrence with a nonsingular Jacobian matrix. Searching for a well behaved Newton method is the main objective in the remainder of this section.

3.1 Constrained Newton Recurrence

One way of stating the eigenvalue problem is to write it as follows:

$$Ax - xx^T Ax = 0, \quad \|x\| = 1. \quad (5)$$

With the eigenvalue problem stated in this form, Tapia's algorithm for constrained optimization can be directly applied after we evaluate the Jacobian matrix of this Newton recurrence [49]. Letting $\lambda \equiv x^T Ax$, the Jacobian for this constrained Newton recurrence is

$$J_C \equiv A - \lambda I - xx^T(A + A^T). \quad (6)$$

Given an initial guess x_0 , the constrained Newton recurrence to find a solution for Equation (5) is as follows,

$$J_C \delta_x = r_i \quad (7)$$

| i | λ_i | $\ r_i\ $ | κ_i |
|-----|--------------|----------------|------------|
| 1 | -90685.22644 | 911704.3057 | 762 |
| 2 | 2500.224244 | 34828.83139 | 2.567e+04 |
| 3 | 0.211851563 | 672.661174 | 2.807e+05 |
| 4 | 0.2109448726 | 0.01285108761 | 4.407e+11 |
| 5 | 0.2106508436 | 0.001874826371 | 4.598e+11 |
| 6 | 0.2097380738 | 0.001081499539 | 5.698e+11 |
| 7 | 0.2094248973 | 0.000100055371 | 2.218e+12 |
| 8 | 0.2094224436 | 6.79586972e-08 | 2.853e+14 |
| 9 | 0.2094224436 | 8.26389478e-09 | 2.134e+18 |

Table 4: The results of applying Rayleigh quotient iteration on EX2.

$$x_{i+1} = \frac{x_i - \delta_x}{\|x_i - \delta_x\|}, \quad (8)$$

where $r_i = Ax_i - \lambda_i x_i$ and $\lambda_i = x_i^T Ax_i$. It is easy to show that the above constrained Newton recurrence is well defined near a nonzero simple eigenvalue [53, Lemma 3.5]. Based on a result of Tapia [49, Theorem 3.3], the above Newton recurrence should converge quadratically near a nonzero simple eigenvalue. In addition, it is also very easy to show that this Newton recurrence is mathematically equivalent to the Rayleigh quotient iteration [53, Lemma 3.6].

To test this constrained Newton recurrence, we have chosen a small finite element matrix named EX2 as a test problem. This test matrix is generated from solving for a fully coupled Navier-Stokes equation using the FIDAP package¹ and is available from MatrixMarket². It is a symmetric matrix with only simple eigenvalues ranging from -7×10^8 to 3×10^6 . The matrix size is 441×441 . The largest negative eigenvalue is $-48,501$, the smallest positive eigenvalue is 0.068 . There are 28 well separated negative eigenvalues and 160 eigenvalues between zero and one. The condition number of the matrix is 10^{10} . The initial guess used in all tests is $[1, 1, \dots, 1]^T$. The experiments are carried on using matlab³ which is internally using 64-bit IEEE floating-point arithmetic.

Table 5 shows the Ritz values and their corresponding residual norms computed during iterations of the constrained Newton recurrence. We iterated until the residual norm does not decrease. For comparison, we also show the intermediate results from the Rayleigh quotient iteration applied on the same problem with the same initial guess. The Ritz values and the residual norms in the two tables substantially matches each other. This confirms that the constrained Newton recurrence is equivalent to the Rayleigh quotient iteration.

Our goal is to use the linear systems of the constrained Newton recurrence as the preconditioning equations for eigenvalue methods. Thus, we are interested in finding how difficult it is to solve the linear systems, by considering their condition numbers. In Tables 4 and 5, the last columns headed by κ_i show the condition numbers of the Jacobian matrices at each step i . The results from the two tables confirm that the condition number of $A - \lambda_i I$ grows

¹FIDAP is a trademark of Fluid Dynamics International.

²MatrixMarket URL: <http://math.nist.gov/MatrixMarket/>.

³Matlab is a trade mark of MathWorks.

| i | λ_i | $\ r_i\ $ | κ_i |
|-----|--------------|-----------------|------------|
| 1 | -90685.22644 | 911704.3057 | 759.3 |
| 2 | 2500.224244 | 34828.83139 | 2.550e+04 |
| 3 | 0.211851563 | 672.661174 | 3.237e+05 |
| 4 | 0.2109448741 | 0.0128510811 | 1.549e+12 |
| 5 | 0.210650847 | 0.001874827738 | 2.271e+12 |
| 6 | 0.2097380775 | 0.001081505247 | 5.322e+11 |
| 7 | 0.2094248973 | 0.000100056626 | 1.739e+12 |
| 8 | 0.2094224436 | 6.730182732e-08 | 7.831e+12 |
| 9 | 0.2094224436 | 2.973326427e-09 | 8.052e+12 |

Table 5: The results of applying the constrained Newton recurrence on EX2.

rapidly as λ_i approaches the exact eigenvalue and the Jacobian matrix of the constrained Newton method approaches constant near convergence. The condition number of J_C in this case is about 100 times larger than the condition number of EX2.

3.2 Augmented Newton Recurrence

Another way of formulating the eigenvalue problem is to treat it as an $(n + 1)$ -dimensional optimization problem which can again be solved using a Newton method [33]. Earlier researchers have formulated the eigenvalue problem using different normalization schemes. Here we choose to normalize the eigenvectors using the 2-norm. The eigenvalue problem can be restated as follows,

$$\begin{cases} (A - \lambda I)x = 0, \\ -\frac{1}{2}x^T x + \frac{1}{2} = 0. \end{cases} \quad (9)$$

This is an unconstrained quadratic problem. Given an initial guess (λ_0, x_0) , the Newton recurrence can be described as follows,

$$\begin{pmatrix} x_{i+1} \\ \lambda_{i+1} \end{pmatrix} = \begin{pmatrix} x_i \\ \lambda_i \end{pmatrix} - J_A^{-1} \begin{pmatrix} (A - \lambda_i I)x_i \\ -\frac{1}{2}x_i^T x_i + \frac{1}{2} \end{pmatrix}, \quad (10)$$

$$J_A = \begin{pmatrix} A - \lambda_i I & -x_i \\ -x_i^T & 0 \end{pmatrix}. \quad (11)$$

It is easy to see that the matrix J_A is non-singular when λ_i is near a simple eigenvalue [53, Lemma 3.1].

Table 6 shows the Ritz values and associated residual norms produced by the augmented Newton recurrence on the test problem. One observation we make here is that the eigenvalue converges to a different number compared with Tables 4 and 5. At the last few steps of the iterations, the residual norm seems to be decreasing quadratically as in Tables 4 and 5. However, the augmented Newton method takes 50% more steps than the Rayleigh quotient iteration to converge to a solution. Near convergence, the condition number of the Jacobian matrix is about the same as the condition number obtained for EX2. However, the condition numbers of the Jacobian matrices at the start of the iterations are considerably larger.

| i | λ_i | $\ r_i\ $ | κ_i |
|-----|---------------|-----------------|------------|
| 1 | 305675.0787 | 1084020.582 | 1.740e+15 |
| 2 | 18112.55342 | 165792.3299 | 1.840e+14 |
| 3 | -10808.31223 | 30650.49007 | 1.736e+13 |
| 4 | -2704.146524 | 3329.024029 | 3.633e+12 |
| 5 | -194.3086005 | 199.2559124 | 1.620e+12 |
| 6 | -0.2278873939 | 1.108173596 | 1.354e+11 |
| 7 | 0.134302848 | 0.1432648279 | 2.197e+09 |
| 8 | 0.07799833383 | 0.2413701758 | 1.112e+12 |
| 9 | 0.04437308049 | 3.650974905 | 9.400e+10 |
| 10 | 0.05875740161 | 0.8677481763 | 1.508e+10 |
| 11 | 0.06520476412 | 0.1461848604 | 2.186e+10 |
| 12 | 0.06772453288 | 0.0086436044 | 2.737e+10 |
| 13 | 0.0680408643 | 3.913476664e-05 | 3.036e+10 |
| 14 | 0.06804357153 | 4.576369965e-09 | 3.078e+10 |

Table 6: Results produced by the augmented Newton recurrence on EX2.

Since λ_i, x_i is a Ritz pair, and x_i is normalized in our eigenvalue routines, the preconditioning equation derived from this augmented Newton recurrence is as follows,

$$\begin{pmatrix} A - \lambda_i I & -x_i \\ -x_i^T & 0 \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_\lambda \end{pmatrix} = \begin{pmatrix} Ax_i - \lambda_i x_i \\ 0 \end{pmatrix}. \quad (12)$$

The vector δ_x replaces z in Equation (1), which is used to expand the basis in algorithm 1.

Based on this observation, we modify slightly the augmented Newton recurrence in the spirit of the constrained Newton recurrence. Each time Equation (10) is solved, we scale the resulting x_{i+1} to have norm one and replace λ_{i+1} with the Rayleigh quotient, i.e.,

$$x_{i+1} = \frac{x_i - \delta_x}{\|x_i - \delta_x\|}, \quad \lambda_{i+1} = x_{i+1}^T A x_{i+1}.$$

| i | λ_i | $\ r_i\ $ | κ_i |
|-----|--------------|-----------------|------------|
| 1 | -90685.22644 | 911704.3057 | 1.740e+15 |
| 2 | 2500.224244 | 34828.83139 | 7.433e+13 |
| 3 | 0.2118515631 | 672.661174 | 1.877e+12 |
| 4 | 0.2109448739 | 0.01285108732 | 1.011e+12 |
| 5 | 0.2106508463 | 0.00187482772 | 2.201e+12 |
| 6 | 0.2097380766 | 0.001081503671 | 5.322e+11 |
| 7 | 0.2094248973 | 0.000100056939 | 1.739e+12 |
| 8 | 0.2094224437 | 6.722151431e-08 | 7.831e+12 |
| 9 | 0.2094224435 | 3.157933201e-09 | 8.052e+12 |

Table 7: The results produced by the normalized augmented Newton recurrence on EX2.

| i | λ_i | $\ r_i\ $ | κ_i |
|-----|--------------|-----------------|------------|
| 1 | -90685.22644 | 911704.3057 | 762 |
| 2 | 2500.224244 | 34828.83139 | 2.567e+04 |
| 3 | 0.2118515633 | 672.661174 | 2.807e+05 |
| 4 | 0.2109448763 | 0.01285108116 | 1.165e+12 |
| 5 | 0.2106508532 | 0.001874830432 | 2.23e+12 |
| 6 | 0.2097380827 | 0.001081513364 | 5.322e+11 |
| 7 | 0.2094248975 | 0.0001000607567 | 1.739e+12 |
| 8 | 0.2094224435 | 7.269909007e-08 | 7.831e+12 |
| 9 | 0.2094224435 | 2.276624783e-08 | 8.052e+12 |

Table 8: The results produced by the inflated Newton recurrence on EX2.

This modified scheme is called the normalized augmented Newton recurrence. The Ritz values and associated residual norms obtained by this new recurrence on the test problem are listed in Table 7. Observe that the Ritz values and associated residual norms in this table are very close to those in Tables 4 and 5. This observation can be explained rigorously.

If we factor Equation (12) symbolically, δ_x can be expressed as follows,

$$\delta_x = (A - \lambda_i I)^{-1} \left(I - \frac{x_i x_i^T (A - \lambda_i I)^{-1}}{x_i^T (A - \lambda_i I)^{-1} x_i} \right) r_i. \quad (13)$$

This expression can be regarded as the exact form of the Olsen modification to the Davidson preconditioning scheme [31]. The goal of the Olsen modification is to make the vector resulting from preconditioning orthogonal to the current Ritz vector. It is easy to verify that δ_x from Equation (12) is orthogonal to x_i . Using Equation (13), it can be shown that the normalized augmented Newton recurrence is equivalent to the Rayleigh quotient iteration [53, Lemma 3.2]. Using the iteration matrix of the Jacobi-Davidson preconditioning scheme, we can also define δ_x as follows

$$\delta_x = \left((I - x_i x_i^T) (A - \lambda_i I) (I - x_i x_i^T) \right)^+ r_i. \quad (14)$$

It is easy to see that Equation (13) and Equation (14) produce the same results. Since Equation (2) has a singular iteration matrix, Equation (14) is a natural alternative. The Jacobi-Davidson preconditioning scheme is known to be a form of inexact Newton method. This is another interpretation in the framework of the constrained Newton recurrence. The recurrence formed from Equation (14) and (8) is equivalent to the Rayleigh quotient iteration [53].

The matrix in front of r_i in Equation (13) closely resembles the Sherman-Woodbury formula for the inverse of $A - \lambda_i I + \alpha x_i x_i^T$ where α is an arbitrary constant [18, Equation (2.1.4)],

$$(A - \lambda_i I + \alpha x_i x_i^T)^{-1} = (A - \lambda_i I)^{-1} \left(I - \frac{\alpha x_i x_i^T (A - \lambda_i I)^{-1}}{1 + \alpha x_i^T (A - \lambda_i I)^{-1} x_i} \right).$$

we can define yet another recurrence which computes δ_x as follows,

$$\delta_x = J_I^{-1} r_i, \quad J_I = A - \lambda_i I + \alpha x_i x_i^T. \quad (15)$$

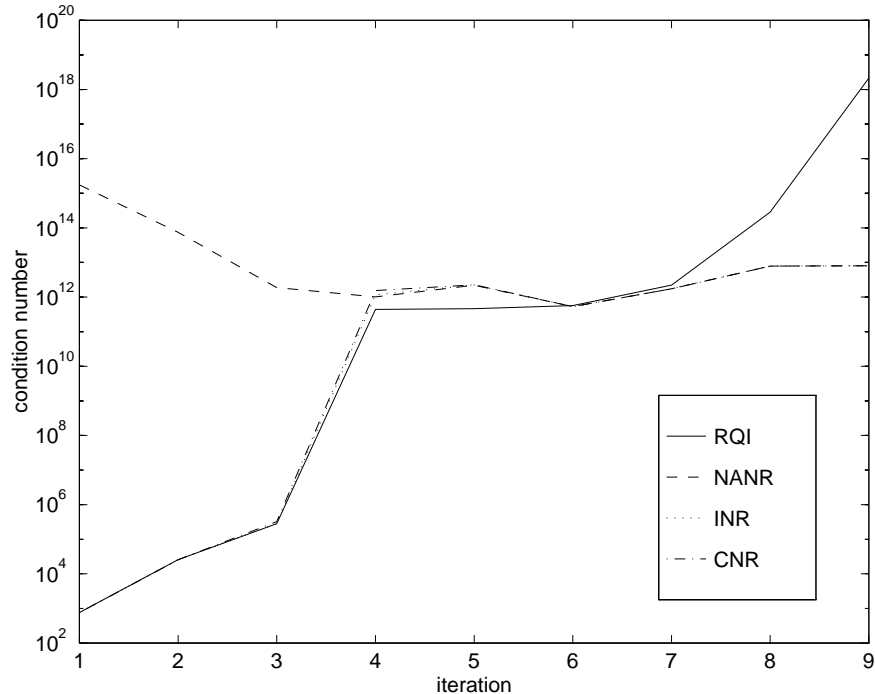


Figure 1: Condition numbers of the Jacobian matrices used by different Newton recurrences.

The same iteration matrix J_I has been used before in a so-called Inflated Inverse Iterations [17]. Thus we refer to the recurrence formed from Equations (15) and (8) as the Inflated Newton Recurrence. This recurrence is well defined since it is always possible to choose an α to make J_I non-singular. It is also easy to verify that the eigenpairs are the stationary points of the recurrence. However, because we directly generated the iteration matrix from Equation (13), we do not know whether J_I is a Jacobian matrix and we cannot analyze the convergence rate of the recurrence. Since the recurrence is a modified form of the normalized augmented Newton iteration, theoretical results to analyze its convergence might be derived, though this is not attempted in this paper. To see how the recurrence performs in practice, we set α to 1 and apply it on the EX2 test problem, see Table 8. Notice that the Ritz values and the residual norms generated by this recurrence are effectively the same as the constrained Newton recurrence and the Rayleigh quotient iteration, see Tables 4 and 5. This justifies the name inflated Newton recurrence. The 9th iteration of this recurrence does not achieve the same residual norm as the Rayleigh quotient iteration. This is due to the floating-point round-off error. The unit round-off error of 64-bit IEEE arithmetic is 10^{-16} and the norm of EX2 is 7×10^8 , the round-off error of evaluating $Ax - \lambda x$ can be as large as 7×10^{-8} . The difference between $\|r_9\|$ in Table 8 and in Table 4 can be regarded as zero because it is smaller than the round-off error.

We show in Figure 1 the condition numbers of the preconditioning systems obtained for the four recurrences, namely, the Rayleigh quotient iteration, the normalized augmented Newton recurrence, the inflated Newton recurrence, and the constrained Newton recurrence.

The condition numbers become large in two cases: near convergence for the Rayleigh quotient iteration and at the start of the normalized augmented Newton recurrence. In all other cases, the condition number of the iteration matrices are less than 10^{12} which is 100 times of the condition number of the matrix EX2.

4 Newton based preconditioners

This section contains a number of small examples to show how the different preconditioning schemes work in practice. Due to the large number of possible implementations, it is not possible to cover every case. We have selected to implement the following schemes.

- **Incomplete factorizations of J_A .** We wanted to test the potential benefit of using Equation (12) as the preconditioning equation. This preconditioning scheme will be called augmented Newton preconditioning. The preconditioning equation will be solved with two incomplete LU factorizations: ILU0, ILUTP [34]. ILUTP uses a level of fill that is equal to half of the average number of nonzero elements per row. In other word, the ILUTP factorization stores slightly more nonzero elements than ILU0. The drop tolerance is 3×10^{-5} and the pivot threshold is 0.1, see documentation of SPARSKIT for definition of these parameters [34].
- **Olsen preconditioners.** The Olsen preconditioning scheme is implemented on top of the regular Davidson preconditioning schemes. We replace $(A - \delta I)$ in Equation (13) by the following three schemes: the diagonal preconditioner, ILU0, and ILUTP.
- **Iterative solvers.** We apply Conjugate Gradient (CG) method to solve Equations (1), (2), (7), and (15). This is how the Jacobi-Davidson method is usually implemented. It is similar to the inner-outer iteration schemes for linear systems [2, 19, 35, 54]. For simplicity, a maximum of 100 steps are allowed for each call to the CG routine. The CG routine is stopped when the residual norm has decreased by a factor of 10^{-4} .

Our first set of tests is to apply Davidson method with above preconditioning schemes on the EX0 test problem. The basis size for the Davidson method is 20. The maximum number of matrix-vector multiplications allowed is 5,000. The test is performed on a SPARC 10 running at 40MHz. We will also show results from another test problem PLAT362. The matrix PLAT362 is in the Harwell/Boeing collection [14]. It is positive definite, but the smallest eigenvalues are 12 order of magnitude smaller than the largest ones and the smallest ones are very close to each other. As before we apply Davidson method with basis size 20 to computed the smallest five eigenvalues and the corresponding eigenvectors. Since we are unable to compute the smallest eigenvalue of PLAT362 in most test cases, Tables 9 – 11 only show the results of EX2.

Table 9 shows the results of using the Davidson method with some of the commonly used preconditioners on EX2. The preconditioners used here are also used in section 2. It is easy to find the smallest eigenvalues of EX2 because they are well separated. Matrix EX2 has more than 60 nonzero elements per row which makes the ILU preconditioners significantly more expensive than the diagonal preconditioner. Even though the Davidson method using the two

| | MATVEC | time(sec) |
|----------|--------|-----------|
| (NONE) | 35 | 0.6 |
| diagonal | 88 | 1.7 |
| ILU0 | 63 | 15.4 |
| ILUTP | 41 | 10.6 |

Table 9: Matrix-vector multiplications and time used by the Davidson method with commonly used preconditioners on EX2.

| | MATVEC | time(sec) |
|-------|--------|-----------|
| ILU0 | 525 | 19.8 |
| ILUTP | >5000 | - |

Table 10: Matrix-vector multiplications and time used by the Davidson method with augmented Newton preconditioners on EX2.

ILU preconditioners take significantly less iterations than using the diagonal preconditioner, they require more time. Since the problem is easy, the unpreconditioned Davidson method uses the least number of matrix-vector multiplication and is also the fastest in terms of CPU time.

Table 10 shows the results of using ILU factorizations on J_A as preconditioners. Neither of the two ILU preconditioners is successful in speeding up the Davidson method. This is due to the fact that J_A is indefinite, and the ILU preconditioners are ineffective in such cases. Another reason that these preconditioning schemes are not effective is that the condition number of J_A is large when the Ritz pair is far from convergence, see Figure 1 and Table 7.

Table 11 shows the results of using the Olsen preconditioning schemes with the diagonal preconditioner, ILU0 and ILUTP. Compared to Table 9, the Olsen scheme improves the effectiveness of the three incomplete factorizations. The improvement with the ILU0 preconditioner is particularly pronounced [31].

Table 12 shows the results of using CG as preconditioner. The results of both EX2 and PLAT362 are shown. The table shows the number of matrix-vector multiplications (MATVEC) used by the Davidson method, the total number of matrix-vector multiplications, and the time. The four schemes show different relative strengths in the two test cases. Using CG on the Davidson preconditioning scheme, see Equation 1, is more effective on EX2

| | MATVEC | time(sec) |
|----------|--------|-----------|
| diagonal | 85 | 1.7 |
| ILU0 | 40 | 10.2 |
| ILUTP | 35 | 10.0 |

Table 11: Matrix-vector multiplications and time used by the Davidson method with the Olsen preconditioning scheme on EX2.

| EX2 | MATVEC (Davidson) | MATVEC (total) | time (sec) |
|--------------------------------------|----------------------|-------------------|---------------|
| $A - \delta I$ | 155 | 2223 | 12.1 |
| $A - \delta I + xx^T$ | 178 | 2742 | 15.2 |
| $A - \delta I - 2x(Ax)^T$ | 162 | 3076 | 16.5 |
| $(I - xx^T)(A - \delta I)(I - xx^T)$ | 485 | 83976 | 510.5 |
| PLAT362 | MATVEC (Davidson) | MATVEC (total) | time (sec) |
| $A - \delta I$ | 146 | 26982 | 6.7 |
| $A - \delta I + xx^T$ | 267 | 49381 | 12.9 |
| $A - \delta I - 2x(Ax)^T$ | 170 | 31600 | 8.2 |
| $(I - xx^T)(A - \delta I)(I - xx^T)$ | 131 | 24400 | 6.7 |

Table 12: Matrix-vector multiplications and time used by the Davidson method using CG as preconditioner on EX2 and PLAT362.

| | converged | fastest |
|--------------------------------------|-----------|---------|
| $A - \delta I$ | 23 | 9 |
| $A - \delta I + xx^T$ | 23 | 6 |
| $A - \delta I - 2x(Ax)^T$ | 22 | 3 |
| $(I - xx^T)(A - \delta I)(I - xx^T)$ | 21 | 6 |

Table 13: Applying the Davidson method with CG preconditioner on Harwell-Boeing test problems.

while using CG on the Jacobi-Davidson scheme, see Equation 2, is slightly more effective on PLAT362. It is remarkable that these four preconditioning schemes are now capable of finding the smallest eigenvalue of the matrix PLAT362, a sign of strong robustness.

We have also applied the Davidson method with these four preconditioning schemes on the 45 Harwell-Boeing test matrices mentioned in Section 2. The results of this test are summarized in Table 13. The second column shows the number of test problems converged and the third column shows the number of cases where the particular preconditioning scheme is the most effective among the four, see also Tables 2 and 3. Overall, the four preconditioners are almost equally effective. The eigenvalue method with the Davidson preconditioning scheme ($\text{CG}(A - \delta I)$) and the inflated Newton preconditioning scheme ($\text{CG}(A - \delta I + xx^T)$) converge on the same problems. Because the Davidson preconditioner is less expensive to apply, it uses slightly less time than the others; thus, it uses the least amount of time to converge in more cases than the others. The Davidson method with the constrained Newton preconditioner ($\text{CG}(A - \delta I - 2x(Ax)^T)$) is the fastest in only three cases. In all these three cases, the Davidson method with the Jacobi-Davidson preconditioning fails to converge. This seems to suggest a certain complementarity between the constrained Newton preconditioner and the Jacobi-Davidson preconditioner.

There are a total of 24 eigenvalue problems converged using these four preconditioning schemes. Out of the 24 cases, there are 14 cases where the Davidson method with these four preconditioning scheme uses less time than with other preconditioning schemes, see Tables 2 and 3. Among the 14 cases, six are with the Davidson preconditioning scheme ($\text{CG}(A - \delta I)$), three cases each with the inflated Newton preconditioning ($\text{CG}(A - \delta I + xx^T)$) and the constrained Newton preconditioning ($\text{CG}(A - \delta I - 2x(Ax)^T)$), two with the Jacobi-Davidson preconditioning ($\text{CG}((I - xx^T)(A - \delta I)(I - xx^T))$). In short, Krylov methods can be effective preconditioners for eigenvalue problems.

5 Summary

In this paper, we examined various preconditioning schemes with an emphasis on finding appropriate preconditioning equations for the eigenvalue problems. We studied the effectiveness of various right-hand sides and matrices for the preconditioning equation, a linear system of equations. A comparison of various choices of right-hand sides indicates that the residual vector of the current approximate eigenpair is the most effective choice.

In order to vary the matrix of the preconditioning equation, it was useful to consider Newton methods for solving eigenvalue problems. Newton methods constitute a natural framework because the linear systems that arise at each Newton steps have residual vectors as their right-hand sides. We found a number of Newton schemes that always generate well-conditioned Jacobian matrices, such as J_I and J_C . The most common preconditioners for linear systems are the incomplete LU-type factorizations. The augmented Newton preconditioning leads to an indefinite iteration matrix for which it is relatively hard to obtain a good incomplete LU factorization. The matrices J_I , J_C , and J_J are low rank modifications of the matrix $A - \delta I$, the matrix of the Davidson preconditioning scheme, and it is not easy to generate incomplete factorization for them either. In the experiments reported here,

the related preconditioning equations were solved with the CG algorithm. The four related schemes are found to be very robust and have similar efficiency.

The Davidson preconditioning scheme is slightly less expensive to apply, and it is therefore faster in some cases. The iteration matrix of the Davidson preconditioning scheme tends to be ill-conditioned near convergence. As is illustrated in Figure 1, the condition number of $A - \delta I$ only becomes high if the approximate solutions are very close to the exact solutions. In most applications, the eigenvalues do not have to be very accurate. Therefore, it is unlikely that the Davidson preconditioning equation will become too ill-conditioned for practical use. This explains why the use of CG to solve the Davidson preconditioning equation is effective. On the other hand, the advantages of using the other preconditioning schemes are significant in some cases. For example, there are a number of reports on the successes of the Jacobi-Davidson method [6, 15, 16, 41, 42, 43].

We recommend two new preconditioning schemes to the users of the Davidson method. These are the inflated Newton preconditioning and the constrained Newton preconditioning. The inflated Newton scheme is only slightly more complex than the Davidson scheme. Far away from convergence, its iteration matrix J_I has almost the same condition number as $A - \delta I$. Near convergence, its condition number is guaranteed to be bounded. It was found to be very effective in our experiments, as Table 13 shows for example. Our tests also show that the constrained Newton preconditioning is effective in the test problems where the Jacobi-Davidson scheme is ineffective. This suggests that if the Jacobi-Davidson preconditioning is not able to converge on a particular problem, it might be worthwhile to apply the constrained Newton scheme as an alternative.

References

- [1] O. Axelsson and P. S. Vassilevski. A survey of multilevel preconditioned iterative methods. *BIT*, 29(4), 1989.
- [2] O. Axelsson and P. S. Vassilevski. A block generalized Conjugate Gradient solver with inner iterations and variable step preconditioning. *SIAM J. Matrix Anal. Appl.*, 12, 1991.
- [3] Owe Axelsson. A survey of preconditioned iterative methods for linear systems of equations. *BIT*, 25:166–187, 1985.
- [4] Owe Axelsson. *Iterative Solution Methods*. Press Syndicate of the University of Cambridge, Cambridge, UK, 1994.
- [5] J. Baglama, D. Calvetti, and L. Reichel. Iterative methods for the computation of a few eigenvalues of a large symmetric matrix. *BIT*, 36:400–421, 1996.
- [6] J. G. L. Booten, H. A. van der Vorst, P. M. Meijer, and H. J. J. te Riele. A preconditioned Jacobi-Davidson method for solving large generalized eigenvalue problems. Technical Report NM-R9414, Mathematisch Centrum, Centrum voor Wiskunde en Informatica, Amsterdam, 1994.

- [7] J. H. Bramble, A. V. Knyazev, and J. E. Pasciak. A subspace preconditioning algorithm for eigenvector/eigenvalue computation. Technical Report 66, Center for Computational Mathematics, University of Colorado at Denver, 1995.
- [8] F. Chatelin. *Eigenvalues of Matrices*. Wiley, 1993.
- [9] E. Chow and Y. Saad. Approximate inverse preconditioners via sparse-sparse iterations. *SIAM J. Sci. Comput.*, 18(6):1657–1675, 1997.
- [10] M. Crouzeix, B. Philippe, and M. Sadkane. The Davidson method. *SIAM J. Sci. Comput.*, 15:62–76, 1994.
- [11] Ernest R. Davidson. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *J. Comput. Phys.*, 17:87–94, 1975.
- [12] Ernest R. Davidson. Super-matrix methods. *Computer Physics Communications*, 53:49–60, 1989.
- [13] J. J. Dongarra, C. B. Moler, and J. H. Wilkinson. Improving the accuracy of computed eigenvalues and eigenvectors. *SIAM J. Numer. Anal.*, 20:23–45, 1983.
- [14] I. S. Duff, R. G. Grimes, and J. G. Lewis. Sparse matrix test problems. *ACM Trans. Math. Soft.*, pages 1–14, 1989.
- [15] A. Edelman, T. Arias, and S. T. Smith. Conjugate Gradient and Newton’s method on the Grassmann and Stiefel manifolds, 1996. Submitted to SIAM J. Matrix Anal. Appl.
- [16] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst. Jacobi-Davidson style QR and QZ algorithms for the partial reduction of matrix pencils. Technical Report nr. 941, Department of mathematics, Universiteit Utrecht, 1996.
- [17] Albert T. Galick. *Efficient solution of large sparse eigenvalue problems in microelectronic simulation*. PhD thesis, University of Illinois at Urbana-Champaign, 1993.
- [18] G. H. Golub and C. F. van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, MD 21211, third edition, 1996.
- [19] A. Greenbraum and L. N. Trefethen. GMRES/CR and Arnoldi/Lanczos matrix approximation problems. *SIAM J. Sci. Comput.*, 15(2):359–368, March 1994.
- [20] M. J. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.*, 18(3):838–853, 1997.
- [21] A. V. Knyazev and A. L. Skorokhodov. Preconditioned gradient-type iterative methods in a subspace for partial generalized symmetric eigenvalue problems. *SIAM J. Numer. Anal.*, 31:1226–1239, 1994.
- [22] L. Yu. Kolotilina and A. Yu. Yeremin. Factorized sparse approximate inverse preconditionings I. theory. *SIAM J. Matrix Anal. Appl.*, 14(1):45–58, 1993.

- [23] R. B. Lehoucq. Restart an Arnoldi reduction. Technical Report MCS-P591-0496, Argonne National Laboratory, Argonne, IL, 1996. Submitted to SIAM Journal on Scientific Computing.
- [24] R. B. Lehoucq and K. Meerbergen. The inexact rational Krylov sequence method. Technical Report MCS-P612-1096, Argonne National Laboratory, Argonne, IL, 1997.
- [25] R. B. Lehoucq and J. A. Scott. An evaluation of software for computing eigenvalues of sparse nonsymmetric matrices. Technical Report MCS-P547-1195, Argonne National Laboratory, Argonne, IL, 1996. Submitted to ACM TOMS.
- [26] T. Manteuffel, S. McCormick, L. Adams, S. Ashby, H. Elman, R. Freund, A. Greenbaum, S. Parter, P. Saylor, N. Trefethen, H. van der Vorst, H. Walker, and O. Wildlund, editors. *Proceedings of Copper Mountain Conference on Iterative Methods*, Copper Mountain, Colorado, 1996.
- [27] Karl Meerbergen. *Robust methods for the calculation of rightmost eigenvalues of non-symmetric eigenvalue problems*. PhD thesis, K. U. Leuven, Heverlee, Belgium, 1996.
- [28] J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.*, 31:148–162, 1977.
- [29] R. B. Morgan and D. S. Scott. Generalizations of Davidson’s method for computing eigenvalues of sparse symmetric matrices. *SIAM J. Sci. Statist. Comput.*, 7:817–825, 1986.
- [30] R. B. Morgan and D. S. Scott. Preconditioning the Lanczos algorithm for sparse symmetric eigenvalue problems. *SIAM J. Sci. Comput.*, 14:585–593, 1993.
- [31] J. Olsen, P. Jorgensen, and J. Simons. Passing the one-billion limit in full configuration-interaction (FCI) calculations. *Chemical Physics Letters*, 169:463–472, 1990.
- [32] Beresford N. Parlett. *The symmetric eigenvalue problem*. Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [33] G. Peters and J. H. Wilkinson. Inverse iteration, ill-conditioned equations and Newton’s method. *SIAM Review*, 21(3):339–360, July 1979.
- [34] Yousef Saad. SPARSKIT: A basic toolkit for sparse matrix computations. Technical Report 90-20, Research Institute for Advanced Computer Science, NASA Ames Research Center, Moffet Field, CA, 1990. Software currently available at <ftp://ftp.cs.umn.edu/dept/sparse/>.
- [35] Yousef Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, 14(2):461–469, March 1993.
- [36] Yousef Saad. *Numerical Methods for Large Eigenvalue Problems*. Manchester University Press, 1993.

- [37] Yousef Saad. Highly parallel preconditioners for general sparse matrices. In G. Golub, A. Greenbaum, and M. Luskin, editors, *Recent advances in Iterative Methods*, pages 165–199. Springer-Verlag, 1994.
- [38] Yousef Saad. ILUT: a dual threshold incomplete ILU factorization. *Numerical Linear Algebra with Applications*, 1:387–402, 1994. Technical Report 92-38, Minnesota Supercomputer Institute, University of Minnesota, 1992.
- [39] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. PWS publishing, Boston, MA, 1996.
- [40] A. H. Sameh and J. A. Wisniewski. A trace minimization algorithm for the generalized eigenvalues problem. *SIAM J. Numer. Anal.*, 19(6):1243–1259, 1982.
- [41] G. L. G. Sleijpen, A. G. L. Booten, D. R. Fokkema, and H. A. van der Vorst. Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems: Part I. Technical Report nr. 923, Department of mathematics, Universiteit Utrecht, 1995.
- [42] G. L. G. Sleijpen and H. A. van der Vorst. A Jacobi-Davidson iteration method for linear eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 17(2), 1996.
- [43] G. L. G. Sleijpen and H. A. van der Vorst. The Jacobi-Davidson method for eigenvalue problems and its relation with accelerated inexact newton schemes, 1996.
- [44] D. Sorensen, R. Lehoucq, P. Vu, and C. Yang. ARPACK: an implementation of the Implicitly Restarted Arnoldi iteration that computes some of the eigenvalues and eigenvectors of a large sparse matrix. Available from [ftp.caam.rice.edu](ftp://caam.rice.edu/pub/people/sorensen/ARPACK), directory `pub/people/sorensen/ARPACK`, 1995.
- [45] D. S. Sorensen. Implicit application of polynomial filters in a K-step Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 13(1):357–385, 1992.
- [46] A. Stathopoulos, Y. Saad, and C. F. Fischer. Robust preconditioning of large, sparse, symmetric eigenvalue problems. *Journal of Computational and Applied Mathematics*, 64:197–215, 1995.
- [47] A. Stathopoulos, Y. Saad, and K. Wu. Dynamic thick restarting of the davidson and the implicitly restarted arnoldi methods. Technical Report UMSI 96/123, University of Minnesota Supercomputer Institute, 1996.
- [48] A. Stathopoulos, Y. Saad, and K. Wu. Thick restarting of the Davidson method: an extension to implicit restarting. In Manteuffel et al. [26].
- [49] R. A. Tapia. Newton’s method for optimization problems with equality constraints. *SIAM J. Numer. Anal.*, 11:874–886, 1974.
- [50] J. H. Wilkinson. *Algebraic Eigenvalue Problem*. Monographs on Numerical Analysis. Oxford Science Publications, New York, 1965.

- [51] K. Wu, Y. Saad, and A. Stathopoulos. Preconditioned Krylov subspace methods for eigenvalue problems. In Manteuffel et al. [26].
- [52] Kesheng Wu. An experimental study of Krylov subspace accelerators. Technical Report UMSI 96/20, Minnesota Supercomputing Institute, University of Minnesota, 1996.
- [53] Kesheng Wu. *Preconditioned Techniques for Large Eigenvalue Problems*. PhD thesis, University of Minnesota, 1997. An updated version also appears as Technical Report TR97-038 at the Computer Science Department.
- [54] Ulrike Meier Yang. Preconditioned Conjugate Gradient-like methods for nonsymmetric linear system. Technical Report 1210, CSRD, University of Illinois at Urbana-Champaign, 1994.
- [55] Harry Yserentant. Old and new convergence proofs for multigrid methods. *Acta Numerica*, pages 285–326, 1993.