Abstract

In numerical continuation methods, one often encounters linear systems of the form:

$$M = \begin{bmatrix} A & b \\ c^T & d \end{bmatrix},$$

where A may be nearly singular but the vectors b and c are such that M is nonsingular. If A is *large and sparse*, use of *iterative* methods seems attractive. In this paper, a number of algorithms based on the nonsymmetric *conjugate gradient* method are considered. Estimates of eigenvalues of M based on those of A are derived. A primary issue is the exploitation of special properties of A, e.g. *symmetry*, in these algorithms. Often, a good preconditioning for A is available, and we show various ways of exploiting it in the algorithms. Results of numerical experiments on a nonlinear elliptic problem will be presented and some general conclusions concerning the relative performance of these algorithms will be made.

Iterative Methods for Solving Bordered Systems with Applications to Continuation Methods

Tony F. Chan and Youcef Saad[1]

18 May 1982

Technical Report #235

---

[1]Computer Science Department, Yale University, Box 2158, Yale Station, New Haven, CT 06520.

The authors were supported in part by Department of Energy Contract DE-ACO2-81ER10996.

Table of Contents

1 Introduction 1
2 Estimates of Eigenvalues for Bordered Matrices 3
2.1 General Case 3
2.2 Case when b or c is an Eigenvector of A. 4
2.3 Application to Pseudo Arclength Continuation for Nonlinear Elliptic Eigenvalue Problems 5
3 Iterative Method: 5
3.1 Block-Elimination 7
3.2 Methods that work on M directly 8
3.3 Symmetric splittings. 8
3.4 Conjugate Gradient Method on the Normal Equations 10
4 Preconditionings 10
5 Numerical experiments 12
6 Conclusion 17
7 Acknowledgement. 18

1. Introduction

In this paper, we shall be concerned with solving linear systems of the form :

$$M \begin{vmatrix} x \\ y \end{vmatrix} = \begin{vmatrix} A & b \\ c^T & d \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} = \begin{vmatrix} f \\ g \end{vmatrix}, \tag{1}$$

or

$$M z = p$$

where the n by n matrix A is *bordered* by the vectors b and c to form a larger system of dimension (n+1) by (n+1).[2] In particular, we are primarily interested in the use of conjugate gradient type *iterative* methods for solving (1) which may be appropriate when the matrix A (and consequently M) is *large and sparse.* We shall present algorithms for solving systems of this form and perform numerical experiments to compare their relative efficiency.

Systems of this form arise in many applications. The matrix A often represents the *regular* part of the problem, whereas the vectors b and c represent either a *constraint* on the solution x or a *coupling* among the variables x and y. An important class of application that we are primarily interested in here is the class of *path-following continuation methods* for solving *nonlinear* problems. These include continuation procedures for solving *nonlinear eigenvalue problems* of the form $G(u,\lambda) = 0$ where u represents the usual "solution" and λ is a real parameter intrinsic to the problem [7, 14, 20]. Usually, one is interested in tracing the solution curves $[u(\lambda),\lambda]$. However, due to the fact that these solution curves may possess *multiple* solutions for a fixed value of λ and the existence of *singular points* (where the Jacobian G_u becomes singular) such as *turning points* and *bifurcation points*, it may not be best, or even possible, to parametrize the solution curves by the naturally occuring parameter λ. Often it is better to paramterize these solution curves either by another independent variable in u or by an *arclength* parameter. In these cases, the matrix A represents the Jacobian G_u, the vector b represents G_λ, and the last equation in (1) represents the "arclength constraint" on $[u,\lambda]$.

---

[2] In general, b and c could be *rectangular* matrices although in this paper we shall only be concerned with the vector case. All the algorithms presented can be easily generalized to the higher dimensional cases.

Another related application area is the class of *homotopy continuation* methods designed to improve the *global* convergence of iterative algorithms (e.g. Newton's method) for solving general nonlinear systems and fixed-point problems [11]. In these homotopy techniques, one transforms a nonlinear system $F(x) = 0$ by a homotopy into a larger system, for example, $H(x,t) = (1-t)(x-x_0) + tF(x) = 0$. Note that $H(x_0,0) = 0$ and $H(x,1) = F(x)$. Thus, one yan start from the known solution x_0 at $t = 0$ and trace the solution curve of $H(x,t)$ until the solution of $H(x,t) = 0$ at $t = 1$ is reached, which by construction is a solution of $F(x) = 0$.

The matrix M is partitioned in the way exhibited in (1) because in many applications, such as the ones mentioned above, the matrix A possesses special structures which one would like to exploit. Two structures that we would like to consider here are the *sparseness* and the *symmetry* of A, both of which become critical in the use of *iterative* methods for solving (1). With many iterative methods for solving large sparse linear systems, the symmetry of the coefficient matrix often plays a critical role in both the efficiency and the convergence of the method [12]. For example, for conjugate gradient type methods ([6], [10]), efficient methods and rather complete theories exist for symmetric problems, whereas the situation for nonsymmetric problems are not as well-understood. In many applications, although A is symmetric, the vectors b and c in (1) are unequal in general, resulting in a matrix M that is *nonsymmetric*. An obvious approach for solving (1) is to apply a *nonsymmetric* iterative method directly, without explicitly taking advantage of the symmetry of A. In this paper, we shall also consider other algorithms for solving (1) which do exploit the symmetry of A. However, all of these algorithms require solving *two symmetric* systems for each system of the form (1). One of the main issues that we would like to address in this paper is the obvious trade-offs among these approaches. Another related issue is the construction of effective *preconditioning* techniques to be used with these algorithms, assuming that a preconditioning matrix is available for the matrix A.

Another property of the matrix A that plays an important role in our discussion is its *indefiniteness*. In fact, in the path following applications mentioned above, A can actually become *singular* near the singular points. However, the vectors b and c are constructed so that the matrix M remains *nonsingular*. Since the convergence of conjugate gradient type methods depends critically on the distribution of the eigenvalues of the coefficient matrix, for example, positive definiteness and the spread of the eigenvalues, it is important to understand the relationship of the

eigenvalues of M to those of A. In Section (2), we shall present some general results in this direction. In Section (3), we shall present three algorithms for solving systems of the form (1), two of which exploit the symmetry of A. Preconditioning techniques will be discussed in Section (4). Extensive numerical experiments have been carried out on applying these algorithms to a pseudo-arclength continuation procedure([7, 14]) for tracing the solution curve of a two dimensional nonlinear elliptic eigenvalue problem that possesses a turning point. The numerical results will be presented in Section (5) and we attempt to draw some general conclusions from these results in Section (6).

2. Estimates of Eigenvalues for Bordered Matrices

In order to be able to interpret the behaviours of various iterative methods, we study in this section the eigenvalues of matrices of the form :

$$M = \begin{vmatrix} A & b \\ c^T & d \end{vmatrix} \tag{2}$$

We assume that A is symmetric.

2.1. General Case

Let us start by block factoring the matrix (2) as follows

$$M = \begin{vmatrix} A & b \\ c^T & d \end{vmatrix} = \begin{vmatrix} I & 0 \\ q^T & 1 \end{vmatrix} \begin{vmatrix} A & b \\ 0 & y \end{vmatrix}$$

with

$$q = A^{-1} b \tag{3}$$

$$y = d - q^T c = d - b^T A^{-1} c \tag{4}$$

Let the eigenvalues of A be λ_i, i=1,..n, where we assume a labelling in increasing order, i.e. $\lambda_1 \leq \lambda_2 \leq .... \leq \lambda_n$. We will denote by μ_i the eigenvalues of M, and label them according to increasing real parts.

Since A is symmetric we can write $A = Q^T \Lambda Q$, where Q is orthonormal and Λ=diag($\lambda_1,\lambda_2,..\lambda_n$). We will denote by γ_i (resp. δ_i) the i-th component of the vector Q^Tb (resp. Q^Tc). From the above factorization applied to M-μI, we get for all μ not in the spectrum of A:

$$\text{Det}(M-\mu I) = \text{Det}(A-\mu I).[d - \mu - \sum_{i=1}^{n} \frac{\gamma_i \delta_i}{\lambda_i - \mu}] \tag{5}$$

From (5) we can easily show the following properties:

Proposition 1:

1) If $\gamma_i \delta_i = 0$, then λ_i is an eigenvalue of M.

2) If $\gamma_i \delta_i$ and $\gamma_{i+1} \delta_{i+1}$ have the same sign, then there is an eigenvalue of M between λ_i and λ_{i+1}. If $\gamma_n \delta_n$ (resp. $\gamma_1 \delta_1$) is positive, then M has an eigenvalue larger than λ_n (resp. less than λ_1).

3) In particular if all $\gamma_i \delta_i$'s are positive, then all the eigenvalues of M are real and interlace with those of A.

The assumption for the last property is in particular satisfied when c=b, i.e. when M is symmetric, which gives the well known Cauchy interlace theorem for symmetric bordered matrices, see e.g. [19],p.186.

2.2. Case when b or c is an Eigenvector of A.

Next we consider the particular case, when either b or c is an eigenvector of A. This situation arises around singular points in continuation methods. If c happens to be an eigenvector of A associated with λ_i, then all the δ_j's are zero, except for δ_i. Likewise if b is an eigenvector of A associated with λ_i, all of the γ_j's are zero except for γ_i. Hence in either case, the first point of Proposition 1 yields:

Proposition 2: Assume that b (resp. c) is an eigenvector of A associated with the eigenvalue λ and let $\gamma = c^Tb$. Then all eigenvalues of A, other than λ, are eigenvalues of

M. Moreover M has two extra eigenvalues which are roots of the following equation in μ:

$$\mu^2 - (\lambda + d)\,\mu + d\,\lambda - \gamma = 0. \tag{6}$$

2.3. Application to Pseudo Arclength Continuation for Nonlinear Elliptic Eigenvalue Problems

The above corollary determines completely the spectrum of M in the situation when either b or c is an eigenvector of A. In the context of pseudo arc-length continuation methods, the matrix A becomes singular near the turning point and the vector c approaches an eigenvector of A associated with the eigenvalue zero [7, 14]. Moreover, the scalar d also goes to zero. Hence the result of Proposition 2 applies. The spectrum of M consists of all the nonzero eigenvalues of A, plus the two eigenvalues that are solutions of (6). Of particular interest in the use of iterative methods is the size of these eigenvalues as compared to the nonzero eigenvalues of A, as the rate of convergence of most iterative methods depends on the spread of the eigenvalues. For two dimensional elliptic problems with a mesh spacing h, the last row of M is usually scaled in such a way that $\gamma = O(h^{-2})$, $\lambda_2 = O(1)$, $\lambda_n = O(h^{-2})$. Moreover, except for a small region next to a singular point, $d = O(h^{-2})$. Thus, we have $d^2 >> \gamma$ and $d^2 >> \lambda_1$. Based on these estimates and equation (6), we can deduce the behaviour of the two new eigenvalues as λ_1 goes to zero, as illustrated in Figure (2-1). There are two things worth noting in Figure 2-1. First, the extra eigenvalues are always in the range $[O(1), O(h^{-2})]$. Thus they do not in general increase the spread of the spectrum of A. Second, the minimum absolute value taken on by these two eigenvalues occur when λ_1 is near, but not at, zero. Although this minimum value is O(1) with this particular scaling, its actual value can be quite smaller than λ_2 or its value at the singular point, resulting in slower convergence for the iterative method. We note that the actual values of the extra eigenvalues depend on the particular scaling used for the last row of M, but the minimum value is independent of the row scaling used. For some iterative methods, it pays to scale the last row by a sign change so that the extra eigenvalues are real.

3. Iterative Method:

We will be interested in several ways of applying Krylov subspace methods for solving system (1).

A classical approach widely used in conjunction with direct methods is to work with A. Such a

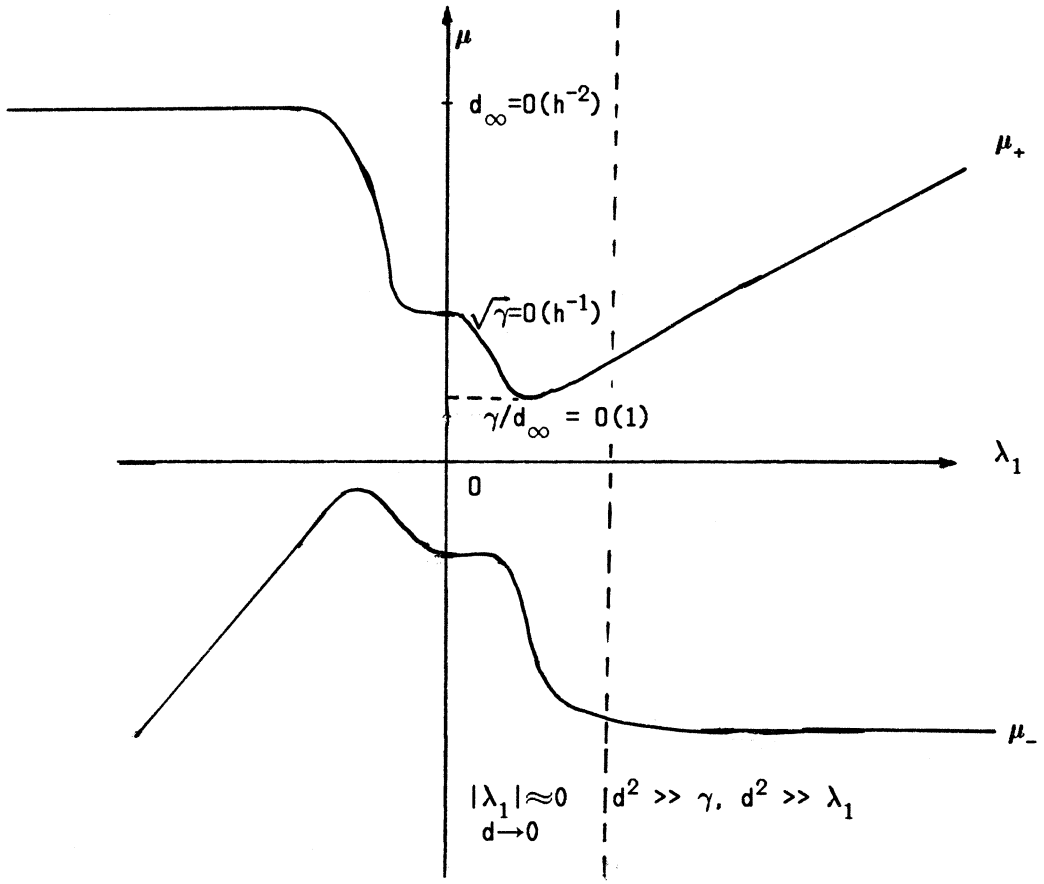


Figure 2-1: Behaviour of the Two Extra Eigenvalues as $\lambda_1 \to 0$

process, sometimes called block elimination, requires the solution of two systems with A and is described in Section 3.1.

An interesting alternative to this approach is to work directly with M, which unlike A, is not singular near the singular point. Here only one system with M, has to be solved but the problem becomes unsymmetric. This will be described in Section 3.2.

Somewhere in between these two approaches lies a whole class of methods in which one attempts to split M into the sum of a symmetric matrix and a low rank matrix. These will be descibed in Section 3.3.

3.1. Block-Elimination

One algorithm for solving linear systems of the form (1) that has been proven to be very useful with *direct* methods is the following ***block-elimination*** algorithm :

Algorithm BE: [7, 14]

(1) Solve $A\,v = b$, (7)
$A\,w = f$. (8)
(2) Compute $y = (g - c^T w) / (d - c^T v)$. (9)
(3) Compute $x = w - y\,v$. (10)

With direct methods, the work consists mainly of one factorization of A and two backsolves with the LU factors of A. Moreover, for problems with many right hand sides, the factorization need only be computed once. Since the factorization usually costs much more than a backsolve, the cost of Algorithm BE is approximately the same as that of factoring the matrix M directly, at least in the general dense case. With iterative methods, however, the situation is rather different because *two* linear systems of dimension n have to be solved for each system of the form (1). On the other hand, Algorithm BE does exploit fully any special properties that may be possessed by A, for example, symmetry or positive definiteness, which are important for the convergence of the iterative methods but are usually not inherited by the matrix M.

In situations where A is nearly singular, the iterative methods may encounter some convergence difficulties in solving (7), (8). In the path following applications mentioned in Section 1, this usually does not present great difficulties for the purpose of tracing the solution curves, ***unless*** one is actually interested in computing the singular points themselves accurately [1, 8, 17, 16]. For direct methods, Algorithm BE can be modified through ***deflation techniques*** [5, 23] to deal with the near singularity of A, while retaining most of its desirable properties with minimal overhead [4]. We are currently investigating similar deflation techniques to be used with iterative methods.

3.2. Methods that work on M directly

Consider the linear system

$$M z = p \tag{11}$$

where M is unsymmetric. If z_0 is an initial approximation of z, and r_0 the corresponding residual vector r_0=b-Az_0, then one defines the j-th Krylov subspace K_j as the linear span of the finite sequence r_0,Mr_0, ...$M^{j-1}r_0$. The Krylov subspace methods consist in finding an approximate solution to (11) belonging to the affine subspace $z_0 + K_j$, such that the residual vector r_j of z_j satisfies certain Galerkin conditions. Among such methods let us mention the Orthomin(k) methods studied by Vinsome [24], and by Eisenstat, Elman and Schlutz [9], the method of Axelsson [3], the ORTHODIR and ORTHORES methods due to Jea and Young [13] and the Incomplete Orthogonalization Method (IOM) described by Saad in [22]. Most of the above methods require that the symmetric part of A be positive definite, in order that they do not break down.

In this paper we will use the IOM method a full description of which may be found in [22]. The main properties characterizing the method are the following:

$$z_j \in z_0 + \{r_0, Mr_0, \ldots, M^j r_0\}, \tag{12}$$

$$(r_i, r_j) = 0, \quad j\text{-}k \leq i < j, \tag{13}$$

where z_i and r_i denote the iterate and the corresponding residual at the i-th iteration and k is an integer parameter larger than 1. In other words the residual vector r_j is orthogonal to the previous k residuals. Note that k vectors from the previous iterations have to be stored.

In the Krylov subspace methods, the only operations performed with M are operations of the form y = Mz, i.e. matrix-vector multiplications. Such operations are easy to perform for bordered matrices and cost at most 2n more multiplications than the corresponding operation with A. This feature makes it possible to take full advantage of sparsity.

3.3. Symmetric splittings

Consider the matrix M in (1) If A is symmetric, then clearly M is a small rank perturbation of a symmetric matrix, and one would like to take advantage of this fact. Let us split M as follows:

$$M = \begin{vmatrix} A & c \\ c^T & d \end{vmatrix}_{n+1} + \begin{vmatrix} b-c \\ 0 \end{vmatrix} e_{n+1}^T \tag{14}$$

We will denote by S_1 the first matrix on the right hand side of (14). Then writing the second matrix of (14) as $u\ v^T$, the solution of (1) can easily be obtained via the Sherman and Morrison formula:

$$(S_1 + u\ v^T)^{-1}\ p = S_1^{-1}\ p + \sigma\ S_1^{-1}\ u \tag{15}$$

with

$$\sigma = \frac{v^T S_1^{-1} p}{1 + v^T S_1^{-1} u}.$$

To apply the above formula, one needs to solve two systems with S_1, namely $S_1^{-1}p$ and $S_1^{-1}u$. Note however that unlike in the methods where M is used directly (Section 3.2), the systems are symmetric.

Since the matrix S_1 is symmetric but not positive definite in general, we must resort to some generalization of the conjugate gradient similar to the SYMMLQ algorithm for solving the systems involving S_1 [18]. We have used a method based on the IOM algorithm which is equivalent to the SYMMLQ algorithm, but slightly less expensive [22]

Note that there are other ways of splitting M. Here are two other possibilities:

$M = S_2 - v\ u^T$, with u and v defined earlier and

$$S_2 = \begin{vmatrix} A & b \\ b^T & d \end{vmatrix} \tag{16}$$

and $M = S_3 + (M - S_3)$ where S_3 is

$$S_3 = \begin{vmatrix} A & e \\ e^T & d \end{vmatrix} \tag{17}$$

with $e = \frac{1}{2}(b+c)$.

Only the splitting S_1 defined by (14) will be considered in this paper.

Although symmetry is an important factor in iterative methods, it is not clear a priori whether solving two symmetric linear systems instead of one unsymmetric linear system of the same dimension will be more costly. The main objective of the numerical experiments to be described in Section(5) is to provide some empirical clarifications in this direction.

An interesting observation concerning the computational work of block elimininations and symmetric splitting is that in both cases we have to solve two linear systems with matrices of dimensions differing by one. Note also that S_1 is a low rank perturbation of A and therefore we may expect the methods to converge in approximately the same number of steps. There is however a big difference in the context of pseudo arc-length continuation methods, which is that A becomes nearly singular near the singular point while S_1 is nonsingular by construction. Thus the symmetric splitting algorithm seems to be more robust.

3.4. Conjugate Gradient Method on the Normal Equations

Another classical way of preserving symmetry, is via the normal equations

$$M^T M z = M^T p \quad . \tag{18}$$

The regular conjugate gradient algorithm can then be applied to (18) which is positive definite. However, not only is the amount of work per step doubled but, as is well known, the condition number of M^TM is the square of that of M, thus rendering the method slowly convergent.

4. Preconditionings

The use of a good preconditioning is often essential for the successful application of Krylov subspace based iterative methods. In this section, we shall discuss the use of preconditioning techniques in the algorithms presented in Section 3. For this purpose, we shall assume that a good preconditioning is available for the matrix A in the form of a matrix B such that $B^{-1} \approx A^{-1}$ and such that the matrix-vector product $B^{-1}x$ is easy to compute. Since we wish to exploit the symmetry of A, we shall also assume that B is symmetric, so that a symmetric method can be used with the preconditioned systems in some of the methods.

The use of preconditioning in the block-elimination algorithm is straightforward, because the

preconditioning B^{-1} can be applied directly to the systems with A as coefficient matrix. A possible difficulty with this approach occurs when A is *indefinite*. The reason is that all of the *symmetric* preconditioned conjugate gradient methods require a preconditioner that is *symmetric and positive definite* [6], and therefore when A is indefinite, it may not be easy to find a positive definite preconditioner B^{-1} that is "close" to A^{-1} in some sense. If A is not too indefinite, however, the situation may not be too serious because one can use a *shifted* incomplete factorization of A to obtain a reasonably good preconditioner [15].

For the other two algorithms presented in Section 3, the construction of a preconditioner is not as straightforward. We shall only consider the more general unsymmetric case here (Section 3.2), as the same techniques can be applied to the symmetric splittings as well. One way to obtain a preconditioning for M based on one for A is to first express the *exact inverse* of M in terms of A^{-1} and then replacing A^{-1} by B^{-1}. Thus, we have:

$$M^{-1} = \begin{vmatrix} A^{-1}(I-bu^T) & v \\ u^T & -y^{-1} \end{vmatrix} \tag{19}$$

where

$$\begin{aligned} y &= c^T A^{-1} b - d\,, \\ u &= A^{-1} c / y\,, \\ v &= A^{-1} b / y\,. \end{aligned} \tag{20}$$

Replacing A^{-1} by B^{-1} in (19), one obtains the following preconditioner for M:

$$P_1 = \begin{vmatrix} B^{-1}(I-b\tilde{u}^T) & \tilde{v} \\ \tilde{u}^T & -\tilde{y}^{-1} \end{vmatrix}\,, \tag{21}$$

where the "~" quantities are defined analogous to (20), but with A^{-1} replaced by B^{-1}. We assume here that $\tilde{y}$ is nonzero.

The above preconditioning requires some preprocessing to compute the quantities y, u and v and the matrix-vector product $P_1 z$ requires a few more inner-products to compute than $B^{-1}z$. For this reason, we shall also consider the following simpler preconditioning:

$$P_2 = \begin{vmatrix} B^{-1} & 0 \\ & \\ 0 & 1 \end{vmatrix} \tag{22}$$

It is of interest to compare the two preconditionings P_1 and P_2, assuming that we have the same preconditioning B for A. Let $E=I-AB^{-1}$ and consider first the "error" $I-MP_2$, in the preconditioning P_2. We clearly have:

$$I - M P_2 = \begin{vmatrix} E & -b \\ & \\ -c^T B^{-1} & 1-d \end{vmatrix} \tag{23}$$

For P_1 a similar, but somewhat more complicated computation, leads to the equality:

$$I - M P_1 = \begin{vmatrix} E(I-b\tilde{u}^T) & \tilde{y}^{-1}Eb \\ \hline 0 & o \end{vmatrix} \tag{24}$$

A comparision of (23) and (24) indicates that if E is small then the preconditioning P_1 will be more accurate than P_2 in general. This is not surprising because of the way this preconditioning is constructed. In general however the norm of E will not be small. The effect of a preconditioning P of M is not to provide a small error E in the inverse but rather to transform the eigenvalues of $P^{-1}M$ in such a way that most of them will be close to one. With this point of view the two preconditionings P_1 and P_2 should not behave too differently as they only differ by a low rank perturbation. This fact is confirmed by the numerical experiments.

5. Numerical experiments

When the preconditioning techniques presented in Section 4 are combined with the basic algorithms of Section 3, a large number of methods result. In order to focus our discussions on a few representative methods and to facilitate the presentation of numerical results, we shall limit our attention to the combinations in Table 5-1.

The algorithms in Table 5-1 have been implemented in a path-following continuation program package written by the authors for tracing solution curves of nonlinear eigenvalue problems. All our numerical experiments have been performed in the context of applying this program package to solve

Table 5-1: List of Algorithms

BE : Block-Elimination, symmetric Conjugate Gradient for A

M : Nonsymmetric Conjugate Gradient for M

SS : Symmetric splitting, symmetric Conjugate Gradient for A

NE : Normal equation on M, Symmetric Positive Definite Conjugate Gradient

PBE : Preconditioned BE, symmetric Conjugate Gradient for $B^{-1}A$

P1M : Nonsymmetric Conjugate Gradient for P_1M

P2M : Nonsymmetric Conjugate Gradient for P_2M

P2SS : Preconditioned Symmetric Splitting, symmetric Conjugate Gradient for P_2S_1

the following model nonlinear elliptic eigenvalue problem :

$$G(u,\lambda) \equiv \Delta u + \lambda e^u = 0 \,, \tag{25}$$

on [0,1] x [0,1] with zero Dirichlet boundary conditions. This problem is discretized by a standard five-point finite difference formula on an uniform m by m grid, resulting in a system of nonlinear equations of size $n = (m-1)^2$. The solution curve for this problem has one simple turning point at [$\lambda \approx 6.808$, $u(0.5,0.5) \approx 1.3$] (for $m = 20$), where the Jacobian G_u is singular [7]. The matrix $A \equiv G_u$ is symmetric, and sparse (banded and no more than 5 nonzeros per row). On the lower branch of the solution curve A is *negative definite*, whereas on the upper branch, it is *indefinite*, with one eigenvalue being *positive*. For the preconditioner, we choose $B \equiv \Delta$, which is symmetric positive definite. For the basic continuation procedure, we use the *pseudo arclength* parametrization of Keller [7, 14], corresponding to using the unit tangent vector $[\dot{u},\dot{\lambda}]$ for the vector [c, d]. At each step of the continuation procedure, a Newton iteration is used to bring a predicted solution to converge to the solution curve. At each step of the Newton iteration, a linear system of the form (1) has to be solved.

The iterative methods that we have used is the usual conjugate gradient method for symmetric positive definite matrices, a method similar to SYMMLQ [22, 18] for symmetric indefinite problems,

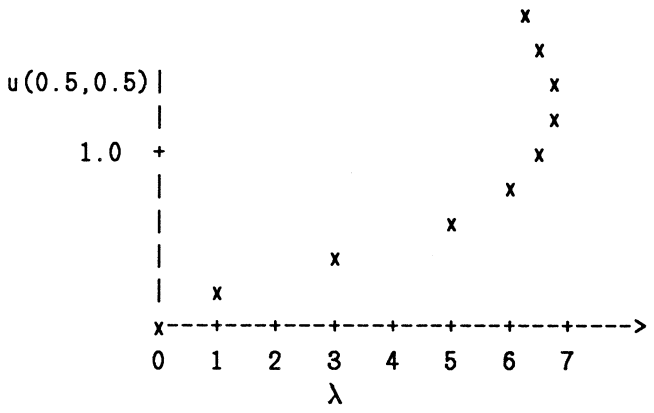
and the Incomplete Orthogonalization Method (IOM) for general nonsymmetric systems [22].

A form of *truncated Newton method* is used for the corrector: the conjugate gradient like inner iteration is stopped when the norm of the residual has been reduced by a factor less than 10^{-3}, and the Newton iteration is stopped when the norm of the residual for the nonlinear equations is less than 10^{-4}. All computations were performed on a VAX-780 with mantissa of 24 bits, corresponding to a relative machine precision of about 10^{-7}.

The experiments were carried out by starting at the trivial solution [0, 0] and tracing the solution curve slightly past the turning point. For each of the methods listed in Table 5-1, we record the total number of inner iterations used by the iterative method. We note that exactly the same continuation steps are taken in the outer iteration independent of which iterative methods is used in the inner Newton iteration. The continuation steps are illustrated in Fig. 5-1. They are automatically chosen by an adaptive strategy analogous to those used in ODE solvers. The results are tabulated in Table 5-2 for m = 10 and 20. Since the matrix M is qualitatively different around the turning point (due to the fact that A is nearly singular), the number of iterations taken near the turning point (denoted by (6,7) in the table) is presented separately from the part away from it (denoted by (0,6) in the table).

All the methods encountered some convergence difficulties near the turning point. As anticipated by the results in Section 2, this is directly due to the near-singularity of A. We have tabulated in Table 5-3 some of the eigenvalues of the matrix M as the turning point is approached, together with the estimated values for the two extra eigenvalues $\bar{\mu}_1$ and $\bar{\mu}_2$ as given by the solution of (6). The eigenvalues of M are computed by a version of Arnoldi's method [2, 21]. We note that the estimates are rather accurate, especially around the singular point. For this particular problem, the minimum absolute value for the extra eigenvalues is an order of magnitude smaller than λ_2 and their values at the singular point. We can also observe a correlation between the larger number of iterations in Table 5-2 and the small value of the extra eigenvalue in Table 5-3. For direct methods this is not as important since this effect manifests itself through the loss of a few digits, which usually is not so drastic as to make the Newton process diverge.

Although we have presented the total number of inner iterations for each method in Table 5-2, the work per iteration is different for each method. In Table 5-4, we tabulated the work per step



Value of λ at each continuation step on the solution curve:

(0,6) : (0.0, 1.0, 3.0, 5.0, 6.0, 6.47)

(6,7) : (6.59, 6.80, 6.70, 6.47)

Figure 5-1: Continuation Steps

Table 5-2: Total number of Iterations

Method	m = 10		m = 20	
	(0,6)	(6,7)	(0,6)	(6,7)
BE	305	465	633	1554
M (k=9)	206	328	623	2217
SS	350	533	737	1193
NE	433	1426	large	large
PBE	72	134	72	143
P1M (k=9)	48	88	48	134
P1M (k=4)	49	87	49	150
P2M (k=9)	48	87	48	90
P2SS	100	186	106	201

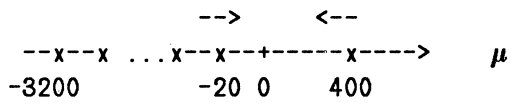
and the storage requirement of each of the competing methods.

Table 5-3: Eigenvalues of M

m = 20

$$\mu_n \leq \cdots\cdots \leq \mu_2 \leq \mu_1$$

λ	μ_n	μ_2	μ_1	$\bar{\mu}_2$	$\bar{\mu}_1$	
3.0	-3179.0	-16.0	399.0	-45.5	400.0	
6.0	-3169.0	-9.2	398.0	-8.8	398.0	
6.47	-3166.0	-6.5	395.0	-6.1	395.0	
6.73	-3163.0	-3.6	364.0	-3.7	364.0	
6.8	-3162.0	-2.2	311.0	-2.2	311.0	
6.77	-3162.0	-31.4	11.8	-90.2	11.8	*



* $\bar{\mu}_2$ is not the second largest eigenvalue of M.
μ_2 computed corresponds to λ_2.

Table 5-4: Work and Storage Per Iteration

Method	Work				*Storage
	Ax,Mx	Mult.	B^{-1}	Vector	
BE	1	7n	0	6n	
M	1	(3k+2)n	0	(2k+2)n	
SS	1	7n	0	6n	
NE	2	5n	0	4n	
PBE	1	7n	1	6n	
P1M	1	(3k+2)n	1	(2k+2)n	
P2M	1	(3k+2)n	1	(2k+2)n	
P2SS	1	7n	1	6n	

* Storage does not include A or B.

6. Conclusion

In this section, we wish to to draw some conclusions based on the numerical results presented in Section 5.

By comparing the performance of methods (BE) or (SS) versus method (M) tabulated in Table 5-2, we can observe the importance of ***symmetry*** in case no preconditioning is used. By comparing the corresponding preconditioned methods (PBE) and (P1M, k=4), and their corresponding operation counts in Table 5-4, we find that, when a preconditioning is available , it seems best to work directly with an iterative nethod on the unsymmetric matrix M. Note that from Table 5-4, each step of method (P1M) with k=4 is less expensive than with k=9, and is not much more expensive than each step of method (PBE).

Also of importance is the observation that despite the fact that the matrices do not have positive definite symmetric parts, a simple preconditioning based on the direct solver associated with a shift of the matrix can be quite effective. Surprisingly the preconditioning continues to work around the singular point for which the matrix is badly conditioned. The two preconditionings P_1 and P_2 yield almost identical results, and this is not unexpected from the discussion of Section 4. This may change if a better approximation of the inverse of A is available.

Finally we can assess the various methods tested as follows:

- Symmetry is not as important for well conditioned problems as for ill-conditioned ones. In particular as shown in Table (5-2), the iterative methods that do not use preconditioning are slow and sensitive to symmetry, especially near the singular point.
- Normal equations approach is to be avoided if unpreconditioned.
- Method (M) takes slightly fewer iterations than (BE) and (SS) except near singular points.
- Method (SS) is slighter superior to method (BE).
- Preconditioned Block Elimination (PBE) is slightly superior to Preconditioned Symmetric Splitting (PSS).
- If a good preconditioning is available then the preconditioned unsymmetric conjugate gradient method (P1M) gives the best results both in number of iterations and in execution time.

7. Acknowledgement

The authors would like to thank Prof. Stanley Eisenstat for his many helpful suggestions throughout this project.

References

[1]

J.P. Abbott, *An Efficient Algorithm for the Determination of Certain Bifurcation Points,* Journal of Computational and Applied Mathematics, 4 (1978), pp. 19 - 27.

[2]

W.E. Arnoldi, *The principle of minimized iteration in the solution of the matrix eigenvalue problem,* Quart. Appl. Math., 9 (1951), pp. 17-29.

[3]

O. Axelsson, *Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations,* Lin. Alg. and its Appl., 29 (1980), pp. 1-16.

[4]

T.F. Chan, *Deflation Techniques and Block-Elimination Algorithms for Solving Bordered Singular Systems,* Tech. Rep. 226, Yale Computer Science Department, New Haven, CT06520, 1982.

[5]

T.F. Chan, *Deflated Decomposition of Solutions of Nearly Singular Systems,* Tech. Rep. 225, Yale Computer Science Department, New Haven, CT06520, 1982.

[6]

R. Chandra, *Conjugate gradient methods for partial differential equations,* Ph.D. Thesis, Dept. of Computer Science, Yale U., New Haven, CT, 1978.

[7]

T.F. Chan and H.B. Keller, Arclength Continuation and Multi-Grid Techniques for Nonlinear Eigenvalue Problems, to appear in SIAM J. Sci. Stat. Comp., June, 1982.

[8]

T.F. Chan, *Newton-Like Pseudo-Arclength Methods for computing Simple Turning Points,* Tech. Rep. 233, Yale Computer Science Department, New Haven, CT06520, 1982.

[9]

S.C. Eisenstat, H.C. Elman and M.H. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations,* Tech. Rep. 209, Yale University, New Haven, Connecticut, 1980.

[10]

H.C. Elman, *Iterative methods for Large Sparse Nonsymmetric systems of Linear Equations,* Tech. Rep. 229, Yale University, New Haven, 1982. Phd-Thesis.

[11]

C.B. Garcia and W.I. Zangwill, *Pathways to Solutions, Fixed Points and Equilibria,* Prentice-Hall, Englewood Cliffs, N.J., 1981.

[12]

A.L. Hageman and D.M. Young, *Applied Iterative Methods,* Academic Press, New York, 1981.

[13]

K.C. Jea and D.M. Young, *Generalized conjugate gradient acceleration of nonsymmetrizable iterative methods,* Lin. Alg. and its Appl., 34 (1980), pp. 159-194.

[14]

H.B.Keller, *Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems,* Applications of Bifurcation Theory, P. Rabinowitz, ed., Academic Press, New York, 1977, pp. 359-384.

[15]

T.A. Manteuffel, *An incomplete Factorization technique for positive definite linear systems,* Math. Comp., 34 (1980), pp. 473-497.

[16]

R.G. Melhem and W.C. Rheinboldt, *A Comparison of Methods for Determining Turning Points of Nonlinear Equations,* Tech. Rep. ICMA-82-32, Institute for Computational Mathematics and Applications, Department of Mathematics and Statistics, University of Pittsburg, Pittsburg, 1982.

[17]

H.D. Mittelmann and H. Weber, *Numerical Methods for Bifurcation Problems - A Survey and Classification,* Bifurcation Problems and their Numerical Solution, Workshop on Bifurcation Problems and their Numerical Solution, January 15-17, Dortmund, 1980, pp. 1-45.

[18]

C.C. Paige and M.A. Saunders, *Solution of sparse indefinite systems of linear equations ,* SIAM j. on Numer. Anal., 12 (1975), pp. 617-624.

[19]

B.N. Parlett, *The Symmetric Eigenvalue Problem,* Prentice Hall, Englewood Cliffs, 1980.

[20]

W.C. Rheinboldt, *Solution Fields of Nonlinear Equations and Continuation Methods,* SIAM J. Numer. Anal., 17 (1980), pp. 221-237.

[21]

Y. Saad, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices,* Linear Algebra and its Applications, 34 (1980), pp. 269-295.

[22]

Y. Saad, *Practical use of some Krylov subspace methods for solving indefinite and unsymmetric linear systems,* Tech. Rep. 214, Yale University, New Haven, Connecticut, 1982.

[23]

G.W. Stewart, *On the Implicit Deflation of Nearly Singular Systems of Linear Equations,* SIAM J. Sci. Stat. Comp., 2 (1981), pp. 136-140.

[24]

P.K.W. Vinsome, *ORTHOMIN, an iterative method for solving sparse sets of simultaneous linear equations,* Proceedings of the Fourth Symposium on Resevoir Simulation, Society of Petroleum Engineers of AIME, , 1976, pp. 149-159.