Abstract: We survey numerical techniques for solving the nonlinear and linear systems arising from applying continuation methods to tracing solution manifolds of parameterized nonlinear systems of the form $G(u,\lambda) = 0$. We concentrate on large and sparse problems, e.g. discretizations of partial differential equations, for which this part of the computation dominates the overall cost. The basic issue is a tradeoff of the exploitation of the sparsity structure of the Jacobian G_u and the numerical treatment of its singularity. Among the techniques to be discussed are: Newton and quasi-Newton methods, low rank correction methods, implicit deflation techniques, Krylov subspace iterative methods and multi-grid methods.

Techniques for Large Sparse Systems Arising from Continuation Methods

Tony F. Chan[1]

Technical Report YALEU/DCS/TR-297

December, 1983

[1]Computer Science Dept., Yale University, Box 2158, Yale Station, New Haven, CT06520, USA. This paper was prepared for the Proceedings of the Conference: *Numerical Methods for Bifurcation Problems*, Dortmund, 1983, to be published in the ISNM series of Birkhauser Verlag in 1984. This work was supported in part by the Department of Energy under contract DE-AC02-81ER10996 and by the Army Research Office under grant DAAG-83-0177.

1 Introduction

In this paper, we are concerned with the numerical solution of parameterized nonlinear systems of the form

$$G(u,\lambda) = 0, \tag{1}$$

where $u \in R^n$, $\lambda \in R^m$ and $G: R^n \times R^m \rightarrow R^n$. Such systems arise in many problems in scientific computing. In the modelling of nonlinear physical phenomena, u may correspond to a field variable and λ to a set of physical parameters. Another source of such parameterized systems is the class of homotopy continuation methods [32] for improving the global convergence of locally convergent methods (e.g. Newton's method) for solving nonlinear sytems and fixed point problems. In these homotopy techniques, one transforms a nonlinear system $F(u) = 0$ by a homotopy, e.g. $G(u,\lambda) = (1-\lambda)(u-u_0) + \lambda F(u) = 0$, so that one starts from the known solution u_0 at $\lambda = 0$ and trace the solution curve of $G(u,\lambda)$ until $\lambda = 1$ to obtain the solution of $F(u) = 0$.

In general, the equation $G(u,\lambda) = 0$ defines a m-dimensional manifold in R^{n+m}. Very often, in addition to obtaining the solution u at a few selected parameter values, more physical insight can often be gained by knowing some general features of the solution manifold as a result of varying the parameters. A continuation procedure can generally be defined as a method for tracing parts of the solution manifold [2, 46, 63, 73]. The design of such a procedure would be straightforward if the solution manifold can be parameterized by the naturally occuring parameters λ. However, this cannot always be done because the solution manifold may contain singular points where the Jacobian G_u is singular and where this parameterization breaks down. Most continuation methods overcome this problem by using a different parameterization of the solution manifold implicitly defined by an additional set of equations

$$N(u,\lambda) = 0, \tag{2}$$

where $N: R^n \times R^m \rightarrow R^m$, so that a solution (u,λ) of (1) is always a *regular* solution of the coupled system (1) and (2), *even at points where G_u is singular*. Conventional methods can then be used to solve this coupled system, e.g. Newton's method, in a predictor-corrector fashion in which a nearby computed solution (u_0,λ_0) is used to generate an initial guess.

In this paper, we shall concentrate on large and sparse problems, for example, where $G(u,\lambda)$ may represent the discretization of nonlinear partial differential equations. For these problems, the solution of the coupled nonlinear system (1) and (2) constitutes the major computational cost of the continuation procedure. It is therefore important to exploit sparsity and structures in G (or G_u) so as to increase the computational efficiency. This can be achieved by any algorithm for solving (1) and (2) each step of which involves solving a subproblem involving G (or G_u) *with* λ *fixed*. However, this approach conflicts with the desire to avoid dealing with the possible singularity of G_u which was the reason for introducing the new parameterization in the first place. Therein lies the basic issue: how does one find a way to exploit the structures in G without running into numerical problems with the singularity of G_u?

This basic conflict not only occurs in the basic continuation procedure, but also in many related algorithms. For example, many types of singular points of the solution manifold, such as turning points, bifurcation points and cusp points, can be characterized as regular solutions to coupled systems of the form of (1) and (2) [1, 8, 44, 51, 53, 55, 57, 60, 62, 67, 68] and consequently computational algorithms derived from this approach must deal with the same conflict. Many techniques to be discussed here have straightforward applications to these problems as well.

We note that it is possible to avoid dealing with such singularities if one stays away from singular points of the solution manifold. It then becomes possible to use G_u^{-1} explicitly in a computational algorithm and exploit the structures in G_u. Many large problems using continuation methods have been solved using this approach [19, 48, 50, 71] and we shall not elaborate on them in this paper. We feel that the regularization of the problem by introducing the N-equation presupposes the necessity of dealing with *possible* singularities of G_u and thus it is desirable to have computational procedures that automatically handle such singularities. Moreover, such a capability becomes indispensable if one is interested in computing the singular points themselves, such as locating turning points and bifurcation points and following folds in the solution manifold [64]. It is one of the themes in this paper to show that the extra cost in doing so is not too high for many numerical techniques suitable for solving large problems.

For ease of presentation, we shall restrict our discussions to the special case of $m = 1$ (i.e. one parameter). This case is also the most common because in practice one often alternately freezes all parameter values except one in tracing the solution manifold. We shall also only treat the case where the dimension of the null space of G_u is at most one. All of the techniques to be presented can be generalized to the higher dimensional cases in a straightforward manner.

2 Nonlinear Techniques

The coupled nonlinear system (1) and (2) can be considered as a single nonlinear system in the variable $z = (u,\lambda)$, namely:

$$F(z) = \begin{bmatrix} G(u,\lambda) \\ N(u,\lambda) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \tag{3}$$

Since we are seeking a regular solution of (3), a regular nonlinear iterative methods can be applied. Most of the methods to be presented here are of this nature.

An obvious approach is to use Newton's method or one of its its many variants (chord, damped, discrete, truncated ...) [59], applied directly to the coupled system (3). At each iteration, a linear system involving the Jacobian:

$$M = \begin{bmatrix} G_u & G_\lambda \\ N_u & N_\lambda \end{bmatrix}$$

must be solved. If the parameterization N is chosen appropriately, the Jacobian M is nonsingular [46] and thus Newton's method has local quadratic convergence, *even when G_u is singular*. The usual drawback of lack of global convergence for Newton's method is not severe in continuation methods because the continuation step size can be controlled to insure local convergence. Newton's method, however, does require evaluation and storage of the Jacobians of G and N. For sparse problems, sparse estimation techniques can be used [20, 22, 61].

Georg [35] and Kearfott [45] have considered dense quasi-Newton methods [24] for solving (3). No Jacobian is needed and only evaluations of G and N are required. Superlinear rate of convergence is usually achievable. For problems where the Jacobians are not available or costly to evaluate, this represents an advantage. However, since the Jacobian plays a central role in bifurcation problems, it may be needed for other purposes anyway, for example, for branch switching [46]. For large and sparse problems, sparse update of the approximate Jacobian is

needed and experience has shown that these do not perform as efficiently as Newton-like methods [36]. The successful application of these methods to large continuation methods remains to be proven.

A very interesting idea has been proposed in [37] and later used in [11]. The nonlinear system (3) is transformed into a least squares minimization problem for the functional $G^2 + N^2$, and a preconditioned nonlinear conjugate gradient method [31, 36] is used for finding the minimum. This technique is especially convenient in situations where a least squares method is already used in a finite element variational setting for solving the system $G = 0$ with fixed λ. However, since the use of least squares approach squares the condition number of G_u, it is extremely important for efficiency reasons to use a good preconditioning.

Lastly, nonlinear relaxation techniques can be attempted. For example, point nonlinear SOR methods [59] can be applied. Another possibility is to use block nonlinear SOR by relaxing u with the G equation and λ with the N equation alternatingly. By design, these methods exploit sparseness in G. However, straightforward application usually encounters convergence difficulty because the Jacobian M is often not positive definite or diagonally dominant. The point nonlinear SOR method, however, can be used as a smoother in a nonlinear multi-grid method (see Section 4).

3 Linear Techniques

Among the nonlinear methods discussed in the last section, the class of Newton-like methods is by far the most commonly used. It is the most general method and has fast local convergence. The need to evaluate Jacobians is compensated by the central role the Jacobians play in bifurcation problems. In the rest of the paper, we shall only deal with this class of methods.

In each iteration of a Newton-like method, a linear system of the form

$$M \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} A & b \\ c^T & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} , \tag{4}$$

needs to be solved, where the n by n matrix A ($\equiv G_u$) is bordered by the vectors b and c to form a larger system of dimension (n+1) by (n+1). When A is large and sparse, one would like to exploit the structures in A when solving this system. However, M does not necessarily inherit desirable structures of A, such as bandedness, symmetry, positive definiteness and separability (for fast direct solvers). It is thus natural to consider algorithms for solving (4) that do exploit these structures in A. On the other hand, dealing with A directly necessarily leads to numerical problems with its possible singularity. These two competing goals constitute the fundamental issue that must be resolved by any practical algorithm. In this section, we shall discuss how some commonly used linear algorithms for large problems can be modified to handle this problem.

3.1 The Deflated Block-Elimination Algorithm

An algorithm that fully exploits structures in A is the following block-elimination algorithm (corresponding to block Gaussian Elimination on M):

Algorithm BE [46]

(1) Solve $A\,v = b$, (5)
$A\,w = f$. (6)
(2) Compute $y = (g - c^T w) / (d - c^T v)$.
(3) Compute $x = w - y\,v$.

Note that only a solver for A is needed. This solver could use any method that is appropriate for the particular problem, for example, sparse Gaussian Elimination, a fast direct solver, an iterative method or a multi-grid method. The last two cases will be discussed in more detail in later sections. In this section, we shall only consider methods based on Gaussian Elimination.

In general, 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 (e.g. in chord-Newton methods), the factorization needs to be computed only once. However, since we use A^{-1} explicitly, we can expect problems when A is nearly singular [12]. Consider the following simple example (n = 2):

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & \epsilon & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1+\epsilon \\ 1 \end{bmatrix},$$

where $|\epsilon|$ is smaller than the machine epsilon of the computer, i.e. $1 + \epsilon = 1$ in floating point arithmetic. In exact arithmetic, $v = (-1/\epsilon,\ 1/\epsilon)^T$ and $w = (1 - 1/\epsilon ,\ 1 + 1/\epsilon)^T$. In floating point arithmetic, $w = (-1/\epsilon,\ 1/\epsilon)^T$. Assuming that this is the only round-off error committed, Algorithm BE would give $x = (0 ,\ 0)^T$ which obviously has a large relative error.

In [12], deflation techniques are proposed for stabilizing Algorithm BE. Instead of computing v and w directly from (5) and (6), numerically stable representations for them are computed by working in subspaces orthogonal to approximate null vectors ϕ and ψ of A. Based on these deflated decompositions of v and w, stable variants of Algorithm BE are derived, one version of which is:

Algorithm DBE: Deflated Block-Elimination Algorithm [12].

1. Compute an approximate normalized left singular vector ψ of A.
2. Compute $\phi = \delta\, A^{-1}\psi$, where $\delta = 1 / ||A^{-1}\psi||$.
3. Compute $c_b = (\psi^T b)$ and $c_f = (\psi^T f)$.
4. Solve $Av_d = b - c_b\,\psi$ for v_d. (v is represented as: $v = v_d + (c_b/\delta)\phi$)
5. Solve $Aw_d = f - c_f\,\psi$ for w_d. (w is represented as: $w = w_d + (c_f/\delta)\phi$)
6. Compute $h_1 = g - c^T w_d$, $h_2 = d - c^T v_d$, $h_3 = h_1 c_b - h_2 c_f$, $h_4 = (c^T\phi)\,c_f - \delta h_1$, $D = (c^T\phi)\,c_b - \delta h_2$.
7. Compute $y = h_4 / D$ and $x = w_d + (h_3\phi - h_4 v_d) / D$.

Note that only two solves with A is needed, exactly the same as in Algorithm BE. The major

overhead for performing the deflation is the computation of ψ and one backsolve for ϕ. The vector ψ can be computed with only one or two backsolves. One possibility is to use a few steps of an inverse iteration [12, 13, 69]. Another algorithm that has been used in the literature is based on computing a LU-factorization of A with a small n-th pivot [16, 47, 49]. Although the usual pivoting strategies (partial and complete pivoting) [25] will exhibit such a LU-factorization for most nearly singular matrices, it is well known that there are counter-examples to this commonly assumed fallacy [16, 38, 74]. An algorithm that is guaranteed to produce such a factorization is given in [16]. This extra work in computing ϕ and ψ is compensated by the fact that they can be reused for several continuation steps and are also useful for switching branch at bifurcation points [46].

In [47, 49], a similar algorithm is independently proposed, but one that works only for the case where A is *exactly* singular. Errors occur if A is nearly but not exactly singular. On the other hand, Algorithm DBE can be proven to be numerically stable [12] *independent of the singularity of A.* Because of its robustness and low overhead, Algorithm DBE can be used at all continuation steps without necessitating a check on the singularity of A.

3.2 Sparse Matrix Methods

If A is sparse, then so is M. Therefore a sparse matrix solver [26, 28, 29, 34] can be used directly on M. However, even if A has a sparse LU factorization, M does not necessarily have a factorization that is just as sparse. This is because if A is nearly singular then some pivoting with the last row or column of M is needed for numerical stability when factoring M which may adversely affect the fill-ins. This is especially severe if a pivot with the last column or row occurs early in the elimination process, as the following simple example shows:

$$\text{(a)} \begin{bmatrix} x & & & x \\ & x & 0 & x \\ & & . & . \\ & 0 & & . \\ x & x & \cdots & x \end{bmatrix} \qquad \text{(b)} \begin{bmatrix} x & x & \cdots & x \\ x & x & & \\ . & & . & 0 \\ . & 0 & . & \\ x & & & x \end{bmatrix} .$$

Without pivoting, matrix (a) produces no fill-in, whereas pivoting with the (n+1,n+1)th element gives matrix (b) which produces a complete fill-in.

Fortunately, there are situations where it can be shown that the last row or column of M does not have to be pivoted until towards the end of the elimination process. This is true, for example, *if* a pivoting strategy can be found to produce a *sparse* LU factorization of A with a small n-th pivot. Using the same pivoting sequence for factoring M, possibly with the last row and column of M appropriately scaled, we obtain at the last stage of the elimination process, a coefficient matrix of the form:

$$\begin{bmatrix} U & u & t \\ 0 & \epsilon & p \\ 0 & q & s \end{bmatrix} ,$$

where U is sparse and ϵ is the small pivot. Using complete or partial pivoting for the lower right hand 2 by 2 submatrix now will handle the singularity.

3.3 Banded Matrices

Discretizations of differential equations often give rise to banded rather than generally sparse matrices. If a banded LU factorization with a small n-th pivot can be found, then the method outlined in Section 3.2 can be used. This is possible for some two point boundary value problems where the parameter λ occurs in the boundary conditions [49].

For a general banded matrix, Rheinboldt [65] proposed the following method. The matrix M is splitted according to $M = S + R$, where S has the same form as M except the vectors b and c are both replaced by the k-th unit vector and R is a rank 2 matrix. The index k is chosen so that S is as well-conditioned as possible. Using the Sherman-Morrison-Woodbury formula [42], for every system in M, one can equivalently solve three systems in S. By taking advantage of the special form of S, it can be shown that a system in S can be reduced to one for A *with its k-th row and column deleted,* which preserves bandedness. Since it requires working with the (possibly complicated) storage structures of A, this algorithm is not as modular as Algorithm DBE. Moreover, generally one more backsolve is required.

3.4 Iterative methods

For many large and sparse problems, e.g. multi-dimensional PDEs, iterative methods may become competitive with direct methods, both in terms of storage and computational time. One of the most successful iterative methods is the class of Krylov subspace iterative methods [3, 18, 21, 27, 30, 40, 41, 43, 52, 66, 72, 75]. In addition to sparseness, the symmetry of the coefficient matrix often plays a critical role in both the efficiency and the convergence of these iterative methods. In general, efficient methods and rather complete theories exist for symmetric and positive definite problems, whereas the situation for indefinite and nonsymmetric problems are not as well-understood. We shall assume in this section only that A is symmetric.

Although M inherits the sparseness properties of A, M may be nonsymmetric while A is symmetric. Therefore the obvious approach of applying a nonsymmetric iterative method directly to (4) may fail to exploit the symmetry of A. In [17], some alternative algorithms are proposed. One approach that does exploit the symmetry in A is to use Algorithm BE. However, *two* linear systems of dimension n have to be solved for each system involving M. Moreover, deflation techniques may have to be used to handle the singularity of A. In principle, deflation techniques for conjugate gradient type methods can be obtained by applying the techniques developed in [13] to the tridiagonal factor produced by the underlying Lanczos process. This is currently under development. Another method that exploits symmetry of A is a low rank correction method. For example, if we split M as $M = S + uv^T$, where S has the same form as M except that the vector b is set to be equal to c, then the solution of (4) can easily be obtained via the Sherman-Morrison formula [42] by solving two systems with the symmetric and nonsingular matrix S. Mittelman [54] has even considered choosing the parametrization N in the continuation method so that $N_u = G_\lambda$ to produce a symmetric M. Finally, one can apply a symmetric positive definite method to the normal equations derived from the M-system. However, it is well-known that the convergence rate will suffer. In short, the alternatives are solving one nonsymmetric system or two symmetric systems or one symmetric positive definite ill-conditioned system.

Another issue is the choice of a good preconditioning, which is often essential for the successful application of Krylov subspace based iterative methods. Assume that a good preconditioning is available for the matrix A in the form of a symmetric matrix B such that $B^{-1} \approx A^{-1}$ and such that the matrix-vector product $B^{-1}x$ is easy to compute. The use of

preconditioning in Algorithm BE is straightforward, because the preconditioning B^{-1} can be applied directly to the systems with A as coefficient matrix. Next, consider the matrices M and S. One way to obtain a preconditioning is to first express the exact inverse in terms of A^{-1} and then replace A^{-1} by B^{-1}. Thus, for example, we have

$$M^{-1} = \begin{bmatrix} A^{-1}(I-bu^T) & v \\ u^T & -y^{-1} \end{bmatrix} \tag{7}$$

where

$$y = c^T A^{-1} b - d,\ u = A^{-1}c/y\ ,\ v = A^{-1}b/y\ . \tag{8}$$

Replacing A^{-1} by B^{-1} in (7) and (8), one obtains the following preconditioner for M:

$$P_1 = \begin{bmatrix} B^{-1}(I-b\tilde{u}^T) & \tilde{v} \\ \tilde{u}^T & -\tilde{y}^{-1} \end{bmatrix}\ , \tag{9}$$

where the "hatted" quantities are defined by analogy to (8), but with A^{-1} replaced by B^{-1}. In addition to P_1, we can use the following simpler preconditioning:

$$P_2 = \begin{bmatrix} B^{-1} & 0 \\ 0 & 1 \end{bmatrix}\ . \tag{10}$$

In [17], numerical experiments were carried out to compare some of the above techniques. We applied them to the model nonlinear elliptic problem $G(u,\lambda) = \Delta u + \lambda e^u = 0$ with zero Dirichlet boundary condition on a unit square. This problem has a simple turning point. For the preconditioning, we use $B = \Delta$. We briefly summarize the results here. It was found that the use of a good preconditioner is extremely important. In particular the methods that do not use preconditioning are slow and sensitive to nonsymmetry near the turning point whereas symmetry is not as important for preconditioned systems. If a good preconditioning is available, it seems best to work directly with the nonsymmetric M than with the symmetric systems. In fact, the method P_2M gives the best results in execution time. As expected, the normal equations approach is not competitive.

Lastly, we point out that a Newton-Krylov subspace method can be implemented by directional differencing techniques without computing or storing the Jacobian matrix [14, 33, 58] and can be used in conjunction with inexact Newton algorithms [23].

4 Multi-Grid Methods

If G is a discretization of a differential or integral operator, then one may consider using a multi-grid (MG) method [10] for solving (3). For this, we need a hierarchy of nested grids on which the discretizations of the operators G, N and their Jacobians are defined. In addition, we need a smoother on each grid, for example, a point relaxation method or a conjugate gradient method. For the MG method to work, the operators and smoothers on the grids must be appropriately chosen to work together in a concerted manner. Although this is rather standard

procedure for a large class of differential systems, very little of the theory and literature on MG is on solving coupled systems. It must be noted that the operator N may have very different smoothing and approximation properties on the hierarchy of grids than G and thus it is not obvious that MG can be made to work as efficiently on the coupled system (linear or nonlinear) directly as on G itself.

There are at least two ways in which multi-grid methods can be applied: (1) solve the linear systems that arise in Newton's method, or (2) solve the coupled nonlinear system directly. Consider case (1) first. The most straightforward approach is to use MG as the black box solver for A in Algorithm BE. However, the singularity of A again causes problems. It was first reported in [15] that MG diverges when A is nearly singular. This divergence is not caused by round-off errors but by the corrections from a coarse grid on which A is nearly singular. As a result of this singularity, the magnitude of the component of the correction in the null vector ϕ direction could be completely wrong. This could happen even if A on the finest grid is reasonably nonsingular. Fortunately, deflation techniques together with Algorithm DBE can be used to overcome this problem [6, 7]. The basic idea is to compute the deflated decompositions of the vectors v and w in Algorithm DBE by MG methods. Similar in spirit to algorithms proposed in [15], approximate null vectors ψ and ϕ are computed on all grids and the iterates on all grids except the finest are purged of these components after smoothing but before a transfer to another grid. The ϕ component on the finest grid is accumulated by a projection process. Again the overhead is low and the algorithm can be used without a check on the singularity of A. Another natural idea for handling the singularity of A is to add a small diagonal shift to M to make A nonsingular [5]. However, besides losing quadratic convergence in the Newton process, the shift has to be chosen carefully and thus is not as easy to implement robustly as the deflation techniques.

In addition to being used on A in conjunction with Algorithm BE, with modifications standard MG techniques can be applied to M directly. Such an idea was proposed by Mittelmann and Weber [56]. On all the grids except the coarsest, smoothing is done only to x using the Ax + yb = f equation with a fixed value for y. On the coarsest grid, a direct solver is used to solve the M-system with pivoting and y is updated there.

Now consider case (2). It is known that a version of MG, called the Full Approximation Scheme [10] (FAS), can be applied directly to a nonlinear system without first applying a linearization. Hackbusch [39] proposed a technique similar to Mittelmann and Weber's [56] except that a FAS is used on G(u,λ) to smooth u on all the grids except the coarsest and λ is updated only on the coarsest grid. Stuben and Trottenberg [70], based on an idea of Brandt, proposed a slightly different algorithm in which both u and λ are updated on all grids. After a FAS smoothing step on a particular grid, u is scaled such that the constraint equation N is satisfied with the current λ, after which λ is updated by "averaging" the G equations on that grid. Recently, Bolstad and Keller [9] have combined the FAS with the τ-extrapolation technique [10] for solving similar problems.

The above techniques of applying MG methods to continuation algorithms compute the solution (u,λ) on the finest grid as a solution of the coupled nonlinear system (3). This presupposes that a fine grid solution is needed on all points on a solution branch. Very often it suffices to compute the *qualitative* behaviour of the solution manifold, perhaps on a coarser grid, and a fine grid solution at a few selected points. If the main features of the solution manifold can be captured by a coarse grid, then a direct method based on Gaussian Elimination can be used on it without incurring a large computational effort. At a point where high accuracy is desired, a MG algorithm can be used to refine the solution. This idea is implemented in the MG-

Continuation program PLTMGC [6]. This package can handle a general class of self-adjoint mildly nonlinear elliptic problems with a parameter dependence on a general two dimensional domain, and can compute target values in λ and $||u||$ with an adaptive stepping algorithm, detect and locate simple turning points and bifurcation points and switch branch at simple bifurcation points. It is based on an earlier package PLTMG [4] and uses Rayleigh-Ritz Galerkin techniques on piecewise linear triangular elements with adaptive mesh refinements. For refining the coarse grid solution using MG, MG deflation techniques [7] and Algorithm DBE are applied to ensure numerical stability.

References

[1] J.P. Abbott. An Efficient Algorithm for the Determination of Certain Bifurcation Points. *Journal of Computational and Applied Mathematics* 4 :19 - 27, 1978.

[2] E. Allgower and K. Georg. Simplicial and Continuation Methods for Approximating Fixed Points and Solutions to Systems of Equations. *SIAM Review* 22(1):28 - 85, 1980.

[3] O. Axelsson. Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations. *Lin. Alg. Appl.* 29:1-16, 1980.

[4] R.E. Bank. *PLTMG User's Guide, June, 1981 version.* Technical Report, Dept. of Mathematics, University of Calif. at San Diego, 1982.

[5] Randolph E. Bank. Analysis of a Multilevel Inverse Iteration Procedure for Eigenvalue Problems. *SIAM J. Numer. Anal.* 19:886-898, 1982.

[6] R.E. Bank and T.F. Chan. *PLTMGC: A Multigrid-Continuation Program for Parameterized Nonlinear Elliptic Systems.* Technical Report 261, Dept. of Computer Science, Yale Univeristy, 1983.

[7] R E. Bank and T. F. Chan. *Multi-Grid Deflation.* 1983. In preparation.

[8] W. Beyn. Defining Equations for Singular Solutions and Numerical Applications. In T. Kupper, H. Mittelmann and H. Weber, Editors, *Numerical Methods for Bifurcation Problems*, Birkhauser Verlag, Basel, 1984 .

[9] J. Bolstad and H.B. Keller. *A Multi-Grid Continuation Method for Elliptic Problems with Turning Points.* 1983. Paper presented at the Siam Fall Meeting, Norfolk, Virginia.

[10] A. Brandt. Multi-level Adaptive Solution to Boundary Value Problems. *Math. Comp.* 31:333-390, 1977.

[11] M.O. Bristeau, R. Glowinski, J. Perriaux, G. Poirier. Non unique solutions of the transonic equation by arc length continuation techniques and finite element least squares methods. In *Proceedings of 5^{th} international conference on finite elements and flow problems, Austin, Texas,* , Jan. 23-26, 1984 .

[12] T.F. Chan. Deflation Techniques and Block-Elimination Algorithms for Solving Bordered Singular Systems. *Siam J. Sci. Stat. Comp.* 5 (1) March 1984.

[13] T.F. Chan. *Deflated Decomposition of Solutions of Nearly Singular Systems.* Technical Report 225, Computer Science Department, Yale Univ., 1982. To appear in Siam J. Numer. Anal., 1984.

[14] T.F. Chan and Ken Jackson. *Nonlinearly Preconditioned Krylov Subspace Methods for Discrete Newton Algorithms.* Technical Report 259, Dept. of Computer Science, Yale Univ., 1983. To appear in Siam J. Sci. Stat. Comp., 1984.

[15] T.F. Chan and H.B. Keller. Arclength Continuation and Multi-Grid Techniques for Nonlinear Eigenvalue Problems. *SIAM J. Sci. Stat. Comp.* 3(2):173-194, June 1982.

[16] T.F. Chan. On the Existence and Computation of LU-factorizations with Small Pivots. *Math. Comp.* 42 (166) April 1984.

[17] T.F. Chan and Y. Saad. *Iterative Methods for Solving Bordered Systems with Applications to Continuation Methods.* Technical Report 235, Computer Science Dept., Yale University, 1982. To appear in Siam J. Sci. Stat. Comp., 1984.

[18] R. Chandra. *Conjugate gradient methods for partial differential equations.* Ph.D. Thesis, Dept. of Computer Science, Yale Univ., 1978.

[19] B. Chen and P. Saffman. Numerical Evidence for the Existence of New Types of Gravity Waves of Permanent Form on Deep Water. *Studies in Applied Math.* 62:1-21, 1980.

[20] T.F. Coleman and J.J. More. Estimation of Sparse Jacobian Matrices and Graph Coloring Problems. *Siam J. Numer. Anal.* 20:187-209, 1983.

[21] P. Concus, G.H. Golub and Dianne P. O'leary. A Generalized Conjugate Gradient Method for the Numerical Solution of Elliptic Partial Differential Equations . In J.R. Bunch and D.J. Rose, Editor, *Proceedings of the Symposium on Sparse Matrix Computations*, Academic Press, New York, 1975, pp. 309-332.

[22] A.R. Curtis, M.J.D. Powell and J.K. Reid. On the Estimation of Sparse Jacobian Matrices. *J. Inst. Maths. Applics.* 13:117-119, 1974.

[23] R.S. Dembo, S. Eisenstat and T. Steihaug. Inexact Newton Methods. *SIAM J. Numer. Anal.* 18(2):400-408, 1982.

[24] J.E. Dennis and J.J. More. Quasi-Newton Methods, Motivation and Theory. *SIAM Review* 19:46 - 89, 1977.

[25] J.J. Dongarra, J.R. Bunch, C.B. Moler and G.W. Stewart. *LINPACK User's Guide.* SIAM, Philadelphia, 1979.

[26] I. Duff. A Survey of Sparse Matrix Research. *Proc. IEEE* 65:500-535, 1977.

[27] Stanley C. Eisenstat, Howard C. Elman and Martin H. Schultz. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM Journal on Numerical Analysis* 20:345-357, 1983.

[28] S.C. Eisenstat, M.C. Gursky, M.H. Schultz and A.H. Sherman. *Yale sparse matrix package I: The symmetric codes.* Technical Report 112, Dept. of Computer Science, Yale Univ., 1977.

[29] S.C. Eisenstat, M.C. Gursky, M.H. Schultz and A.H. Sherman. *Yale sparse matrix package II: The nonsymmetric codes.* Technical Report 114, Dept. of Computer Science, Yale Univ., 1977.

[30] Howard C. Elman. *Iterative Methods for Large, Sparse, Nonsymmetric Systems of Linear Equations.* Ph.D. Thesis, Yale University, 1982. Techreport # 229.

[31] R. Fletcher and C.M. Reeves. Function Minimization by Conjugate Gradients. *Comput. J.* 7:149-154, 1964.

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

[33] W. C. Gear and Y. Saad. Iterative Solution of Linear Equations in ODE Codes. *Siam J. Sci. Stat. Comp.* 4(4) December 1983.

[34] A. George and J.W. Liu. *Computer Solution of Large Sparse Positive Definite Systems.* Prentice Hall, New Jersey, USA, 1981.

[35] K. Georg. On Tracing an Implicitly Defined Curve by Quasi-Newton Steps and Calculating Bifurcation by Local Perturbations. *Siam J. Sci. Stat. Comp.* 2(1) March 1981.

[36] P.E. Gill, W. Murray and M. Wright. *Practical Optimization.* Academic Press, New York, 1981.

[37] R. Glowinski, H.B. Keller and L. Reinhart. *Continuation-Conjugate Gradient Methods for the Least Squares Solution of Nonlinear Boundary VAlue Problems.* Technical Report, Institut National de Recherche en Informatique et en Automatique, 1982.

[38] G.H. Golub, G.W. Stewart and V. Klema. *Rank Degeneracy and Least Squares Problems.* Technical Report STAN-CS-76-559, Computer Science Dept., Stanford University, 1976.

[39] W. Hackbusch. Multi-Grid Solution of Continuation Problems. In R. Ansorge, T. Meis and W. Torning, Editors, *Iterative Solution of Nonlinear Systems*, Springer-Verlag, Berlin, 1982 .

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

[41] M.R. Hestenes and E. Stiefel. Methods of Conjugate Gradient for solving Linear Systems. *Journal of Research of the National Bureau of Standards* 49:409-436, 1952.

[42] A.S. Householder. *Theory of Matrices in Numerical Analysis.* Blaisdell Pub. Co., Johnson, Colo., 1964.

[43] Kang Chang Jea. *Generalized Conjugate Gradient Acceleration of Iterative Methods.* Ph.D. Thesis, University of Texas at Austin, 1982.

[44] A. Jepson and A. Spence. Singular Points and Their Computations. In T. Kupper, H. Mittelmann and H. Weber, Editors, *Numerical Methods for Bifurcation Problems*, Birkhauser Verlag, Basel, 1984 .

[45] R.B. Kearfott. A Derivative-Free Arc Continuation Method and a Bifurcation Technique. In E.L. Allgower, K. Glashoff and H.-O. Peitgen, Editors, *Numerical Solution of Nonlinear Equations*, Springer Verlag, New York, 1981 .

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

[47] H.B. Keller. *Practical Procedures in Path Following Near Limit Points.* 1982. In *Computing Methods in Applied Sciences and Engineering,* eds. Glowinski and Lions, North-Holland Pubs. Inc.

[48] H.B. Keller and R. Meyer-Spache. Numerical Study of Taylor-Vortex Flows Between Rotating Cylinders. *J. Comp. Phys.* 35(1):100-109, 1980.

[49] H.B. Keller. The Bordering Algorithm and Path Following Near Singular Points of Higher Nullity. *SIAM J. Sci. and Stat. Comp.* 4(4) 1983.

[50] H.B. Keller and R. Schreiber. Accurate Solutions for the Driven Cavity. *J. Comp. Phys.* 49(2):310-333, 1983.

[51] M. Kubicek ad M. Holodniok. Numerical Determination of Bifurcation Points in Steady State and Periodic Solutions - Numerical Algorithms and Examples. In T. Kupper, H. Mittelmann and H. Weber, Editors, *Numerical Methods for Bifurcation Problems*, Birkhauser Verlag, Basel, 1984 .

[52] Thomas A. Manteuffel. The Tchebychev Iteration for Nonsymmetric Linear Systems. *Numer. Math.* 28:307-327, 1977.

[53] R.G. Melhem and W.C. Rheinboldt. A Comparison of Methods for Determining Turning Points of Nonlinear Equations. *Computing* 29:201-226, 1982.

[54] H.D. Mittelmann. An Efficient Algorithm for Bifurcation Problems of Variational Inequalities. *Math. Comp.* 41:473-485, 1983.

[55] H.D. Mittelmann and H. Weber, Editors. *Bifurcation Problems and their Numerical Solution.* Birkhauser, Basel, 1980.

[56] H.D. Mittelmann and H. Weber. *Multi-Grid Solution of Bifurcation Problems.* Technical Report 65, Abteiling Mathematik, Universitat Dortmund, 1983. To appear in Siam J. Sci. Stat. Comp.

[57] G. Moore and A. Spence. The Calculation of Turning Points of Nonlinear Equations. *SIAM J. Numer. Anal.* 17:567-576, 1980.

[58] Dianne P. O'Leary. A Discrete Newton Algorithm for Minimizing a Function of Many Variables. *Math. Progr.* 23:20 - 33, 1982.

[59] J.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.* Academic Press, New York, 1970.

[60] G. Pönisch and H. Schwetlick. Computing Turning Points of Curves Implicitly Defined by by Nonlinear Equations Depending on a Parameter. *Computing* 26:107-121, 1981.

[61] M.J.D. Powell and Ph. L. Toint. On the Estimation of Sparse Hessian Matrices. *SIAM J. Numer. Anal.* 16:1060-1073, 1979.

[62] G.W. Reddien. Computation of Turning and Bifurcation Points for Two Point Boundary Value Problems. In T. Kupper, H. Mittelmann and H. Weber, Editors, *Numerical Methods for Bifurcation Problems,* Birkhauser Verlag, Basel, 1984 .

[63] W.C. Rheinboldt and J.V.Burkardt. A Locally Parameterized Continuation Process. *ACM Trans. Math. Soft.* 9(2):215-235, June 1983.

[64] W.C. Rheinboldt. Computation of Critical Boundaries on Equilibrium Manifolds. *Siam J. Numer. Anal.* 19:653-669, 1982.

[65] W.C. Rheinboldt. Numerical Analysis of Continuation Methods for Nonlinear Structural Problems. *Computers and Structures* 13:103-113, 1981.

[66] Y. Saad. Krylov Subspace Methods for Solving Large Unsymmetric Linear Systems. *Math. Comp.* 37:105-126, 1981.

[67] H. Schwetlick. Algorithms for Finite Dimensional Turning Point Problems from Viewpoint to Relationships with Constrained Optimization Methods. In T. Kupper, H. Mittelmann and H. Weber, Editors, *Numerical Methods for Bifurcation Problems,* Birkhauser Verlag, Basel, 1984 .

[68] R. Seydel. Numerical Computation of Branch Points in Nonlinear Equations. *Numer. Math.* 33:339-352, 1979.

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

[70] K. Stuben and U. Trottenberg. Multi-Grid Methods: Fundamental Algorithms, Model Problem Analysis and Applications. In W. Hackbusch adn U. Trottenberg, Editors, *Multigrid Methods,* Springer Verlag, Berlin, 1982 .

[71] K.K. Tung, T.F. Chan and T. Kubota. Large Amplitude Internal Waves of Permanent Form. *Studies in Applied Math.* 66:1-44, February 1982.

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

[73] H. Wacker, Editors. *Continuation Methods.* Academic Press, New York, 1978.

[74] J.H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford University Press, London, 1965.

[75] David M. Young and Kang C. Jea. Generalized conjugate gradient acceleration of nonsymmetrizable iterative methods. *Linear Algebra and Its Applications* 34:159-194, 1980.