12-6-1999

An Adaptation of Krylov Subspace Methods to Path Following Problems

Homer F. Walker
Worcester Polytechnic Institute, walker@wpi.edu

SIAM J. SCI. COMPUT.
Vol. 21, No. 3, pp. 1191–1198

TIMELY COMMUNICATION

AN ADAPTATION OF KRYLOV SUBSPACE METHODS TO PATH FOLLOWING PROBLEMS*

HOMER F. WALKER†

Abstract. A procedure is outlined for adapting Krylov subspace methods to solving approximately the underdetermined linear systems that arise in path following (continuation, homotopy) methods. This procedure, in addition to preserving the usual desirable features of Krylov subspace methods, has the advantages of satisfying orthogonality constraints exactly and of not introducing ill-conditioning through poor scaling.

Key words. Krylov subspace methods, path following methods, continuation methods, homotopy methods, underdetermined linear systems, Householder transformations

AMS subject classifications. 65H10, 65F10

PII. S1064827597315376

1. Introduction. Krylov subspace methods for iteratively solving linear systems have enjoyed wide success and great popularity in recent years; see [5] and [7] for surveys. These methods have been effectively adapted to solving nonlinear systems through Newton–Krylov methods (cf. [2]), in which they are used to solve approximately the linear systems that characterize steps of Newton's method. In this note, we outline a particular way of adapting Krylov subspace methods to the context of *path following problems*, including continuation and homotopy problems (see, e.g., [1] and [14]), in which the goal is to determine or follow a curve of solutions of a parameter-dependent nonlinear system over some range of parameter values. Solution methods for these problems typically give rise to underdetermined linear systems, and our specific goal is to describe a simple procedure for applying Krylov subspace methods to these underdetermined systems. To keep the presentation appropriately brief and straightforward, only essential issues will be addressed.

We consider a path following problem in the general form.

Problem PF. *Given* $F : \mathbb{R}^n \times \mathbb{R}^1 \to \mathbb{R}^n$, *solve* $F(x, \lambda) = 0$ *over a range of* (x, λ)*-values.*

For convenience, we denote $(x, \lambda) = \bar{x}$ and regard it as a vector in $\mathbb{R}^{n+1}$; we also write $F(x, \lambda) = F(\bar{x})$, $F'(x, \lambda) = F'(\bar{x})$, etc., where F' denotes the Jacobian (matrix) of F. It will usually be convenient to assume that Problem PF has an associated smooth solution curve, each point $\bar{x}$ of which is *regular* or *nonsingular* in that $F'(\bar{x}) \in \mathbb{R}^{n\times(n+1)}$ is of full rank n. Treatment of points such as bifurcation points at which F' is rank deficient is an important issue but will not be considered here.

Solution algorithms for Problem PF vary but are typically of predictor–corrector form, in which the next point on the solution curve is "predicted" by some easy but

*Received by the editors January 22, 1997; accepted for publication (in revised form) May 15, 1998; published electronically December 3, 1999.
http://www.siam.org/journals/sisc/21-3/31537.html
†Mathematical Sciences Department, Worcester Polytechnic Institute, Worcester, MA 01609-2280 (walker@wpi.edu). This work was supported in part by United States Department of Energy grant DE-FG03-94ER25221, by National Science Foundation grants DMS-9400217 and DMS-9727128, and by subcontract B341493, University of California, Lawrence Livermore National Laboratory.

relatively inaccurate means, and then a series of Newton-like corrector iterations is employed to return approximately to the curve. We shall not elaborate on the nature of predictor methods here, except to note that they typically make key use of one or more tangent vectors to the solution curve. These tangent vectors are characterized as null vectors of F' and often have to be computed with considerable accuracy in difficult applications.

The corrector iterations typically begin with $\bar{x}_0$ determined by the predictor and produce iterates $\bar{x}_{k+1} = \bar{x}_k + \bar{s}_k$ through steps $\bar{s}_k$ characterized by analogues of the linear systems that determine steps of Newton's method, viz.,

$$F'(\bar{x}_k)\,\bar{s}_k = -F(\bar{x}_k). \tag{1.1}$$

The system (1.1) is underdetermined, and an additional condition must be imposed to specify $\bar{s}_k$ uniquely. This can be done in various ways; here, we take the additional condition to be of the form

$$\bar{t}_k^T \bar{s}_k = 0, \qquad \bar{t}_k \in \mathbb{R}^{n+1}. \tag{1.2}$$

In (1.2), $\bar{t}_k$ may be an approximate unit null vector of $F'(\bar{x}_k)$, as in the normal flow algorithm (cf. [15] and [16]), or it may be an approximate unit tangent to the solution curve at the previous point that does not change with k, as in the augmented Jacobian algorithm of [15] and [16] or in pseudo-arclength continuation as described, e.g., in [3], [6], and [10]. In either case, computing one or more approximate null vectors of F' may be an important ancillary task.

In section 2 below, we first describe our procedure for adapting Krylov subspace methods to computing a corrector step, i.e., to computing an approximate solution of (1.1) subject to the constraint (1.2). We then outline a variant for finding an approximate unit null vector of $F'(\bar{x})$ for a given $\bar{x}$; as indicated above, this may be required by either the predictor method or the corrector iterations or both. In section 3 we demonstrate the effectiveness of these procedures through numerical experiments on representative model problems. A summary is offered in section 4.

2. The adaptation.

2.1. Computing a corrector step. An obvious way to use a Krylov subspace method to compute a corrector step is to apply it to the (square) linear system obtained by augmenting (1.1) with (1.2). However, previous experience in [8] and [4] suggests that Krylov subspace methods may not perform well on these augmented systems. Possible difficulties evident in this approach include the following:

- The augmented system may be ill-conditioned if the scale of (1.1) differs greatly from that of (1.2). Although it may be possible to rescale (1.2) to obtain a well-conditioned system, it may not be clear how best to do this.
- Iterates produced by a Krylov subspace method applied to the augmented system only approximately satisfy (1.2). This difficulty may be aggravated by poor scaling as just discussed. It may be a serious concern when only a fairly inaccurate approximate solution of the augmented system is desired.

The approach proposed here circumvents these potential difficulties by imposing (1.2) directly as a constraint on the iterates. The abstract framework is as follows:

1. Find $Q \in \mathbb{R}^{(n+1)\times n}$ such that $\text{Range}\,(Q) = \{\bar{t}_k\}^{\perp}$ and $\|Qy\|_2 = \|y\|_2$ for all $y \in \mathbb{R}^n$. Then $F'(\bar{x}_k)Q \in \mathbb{R}^{n\times n}$.
2. Apply the Krylov subspace method to solve approximately $F'(\bar{x}_k)Qy_k = -F(\bar{x}_k)$ for $y_k \in \mathbb{R}^n$. Then set $\bar{s}_k = Qy_k$.

Under this approach, $\bar{s}_k = Qy_k$ satisfies (1.2) regardless of how well it approximately satisfies (1.1). Scaling associated with (1.2) is not an issue. Moreover, since $\|Qy\|_2 = \|y\|_2$ for $y \in \mathbb{R}^n$, conditioning problems are not worsened as long as $\bar{t}_k$ is an accurate unit null vector. Indeed, if $F'(\bar{x}_k)$ is of full rank and $\bar{t}_k$ is exact, i.e., $F'(\bar{x}_k)\bar{t}_k = 0$, then the singular values of $F'(\bar{x}_k)Q$ are precisely the positive singular values of $F'(\bar{x}_k)$.

The particular way of specifying Q that we propose here is based on Householder transformations, as follows:

1. Choose i, $1 \le i \le n+1$, and let $\bar{e}_i$ be the ith column of $I \in \mathbb{R}^{(n+1)\times(n+1)}$. Determine a Householder transformation P such that $P\bar{t}_k = \pm\bar{e}_i$.
2. Set $Q = P\hat{I}_i$, where $\hat{I}_i$ is obtained by deleting the ith column of I.

It is easy to verify that Q specified in this way has the desired properties $\text{Range}\,(Q) = \{\bar{t}_k\}^{\perp}$ and $\|Qy\|_2 = \|y\|_2$ for all $y \in \mathbb{R}^n$. The cost of applying such a Q is just that of applying P, viz., one dot product and one "saxpy." The Q of this form that has been most effective in our testing, and which was used in the experiments in section 3, is $Q = P\hat{I}_{n+1}$, i.e., $Qv = P\begin{pmatrix} v \\ 0 \end{pmatrix}$ for $v \in \mathbb{R}^n$. This was very successful in the experiments in section 3 and in other limited testing, while several other choices of the form $Q = P\hat{I}_i$ were not very successful. The success of this choice is perhaps not surprising, given the special nature of the $(n+1)$st coordinate in path following problems.

This procedure preserves an important property of Krylov subspace methods: Applying a method requires only products of $F'(\bar{x}_k)$ and perhaps also $F'(\bar{x}_k)^T$ with vectors. If these products are computed analytically, then sparsity of $F'(\bar{x}_k)$ can be fully exploited. If analytic computation is infeasible, then products involving $F'(\bar{x}_k)$, if not $F'(\bar{x}_k)^T$, can be approximated in a well-known way with finite differences of F-values.

Preconditioning is essential for success on most large scale problems and is a particularly important issue here. In difficult applications, there is often a major investment in constructing a preconditioning operator $M : \mathbb{R}^n \to \mathbb{R}^n$ that is effective in the "fixed-parameter" case, i.e., for solving $F(x, \lambda) = 0$ for a fixed value of λ. It would be highly desirable if such a preconditioner could be used for path following as well. In our approach, left preconditioning with such an M can be accomplished in the usual way, i.e., by replacing (1.1) with $M^{-1}F'(\bar{x}_k)\,\bar{s}_k = -M^{-1}F(\bar{x}_k)$. Right preconditioning with M can be done in two ways. First, one can approximately solve $F'(\bar{x}_k)QM^{-1}z_k = -F(\bar{x}_k)$ for $z_k \in \mathbb{R}^n$ and then set $\bar{s}_k = QM^{-1}z_k$, corresponding to the usual implementation of right preconditioning. Second, writing

$$\bar{M} = \begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix},$$

one can approximately solve $F'(\bar{x}_k)\bar{M}^{-1}Qz_k = -F(\bar{x}_k)$ for $z_k \in \mathbb{R}^n$ and set $\bar{s}_k = \bar{M}^{-1}Qz_k$. These alternatives were equally effective in limited testing; since the first is somewhat simpler and more natural, it was used in the experiments in section 3.

2.2. Computing a unit null vector of F'. In our approach to computing an approximate unit null vector $\bar{t}$ of $F'(\bar{x})$ for given $\bar{x}$, we assume that an initial approximate null vector $\bar{t}_0$ is given. In practice, this $\bar{t}_0$ can be postulated at the first continuation step or obtained from previous continuation steps or corrector iterations once the path following algorithm is under way. The proposed procedure is as follows:

1. Solve $F'(\bar{x})\,\bar{s} = -F'(\bar{x})\,\bar{t}_0$ subject to $\bar{t}_0^T\bar{s} = 0$.
2. Set $\bar{t} = (\bar{t}_0 + \bar{s})/\|\bar{t}_0 + \bar{s}\|_2$.

Step 1 of this procedure is precisely the same mathematically as solving (1.1) subject to (1.2), and so for step 1 we propose using the procedure outlined in section 2.1 for computing a corrector step, with the appropriate straightforward changes. As in section 2.1, conditioning problems should not be worsened if $\bar{t}_0$ is reasonably accurate. Even if serious ill-conditioning occurs because of inaccuracy of $\bar{t}_0$, the resulting $\bar{t}$ should be a better approximate unit null vector than $\bar{t}_0$, and the procedure can be repeated.

An analogue of the alternative approach noted at the outset in section 2.1 is available here, viz., solving the augmented system

$$\begin{pmatrix} F'(\bar{x}) \\ \bar{t}_0^T \end{pmatrix} \bar{s} = \begin{pmatrix} -F'(\bar{x})\,\bar{t}_0 \\ 0 \end{pmatrix}.$$

However, this would be subject to the potential difficulties noted in section 2.1. Another possibility is to solve the $n \times n$ system $D_x F(\bar{x})s = -F'(\bar{x})\bar{t}_0$ for $s \in \mathbb{R}^{n\times n}$ and then to take $\bar{t} = [\bar{t}_0 + (s,0)]/\|\bar{t}_0 + (s,0)\|_2$. However, this system is singular at fold points (also known as turning points or limit points) and ill-conditioned near them.

3. Numerical experiments. We implemented and tested the procedures outlined in section 2 in a rudimentary predictor–corrector algorithm using a simple *Euler predictor* (cf. [1]), in which prediction is along an approximate tangent to the solution curve. The stepsize h along the approximate tangent was adaptively adjusted to give satisfactory corrector convergence in a modest number of corrector iterations. We note in each case below how the approximate tangent was constructed and how the vectors $\bar{t}_k$ in (1.2) were selected. The Krylov subspace methods used were GMRES(m) [11] and BiCGSTAB [13]. Only left preconditioning was used; particular preconditioners in each case are described below. All computing was done in MATLAB on a Sun Microsystems SPARCstation 20.

The test problems all involved elliptic PDEs in $\mathcal{D} \equiv [0,1] \times [0,1]$, as described further below. Discretization was with the usual centered differences on a regularly spaced rectangular grid of interior points in each case.

3.1. Two nonlinear eigenvalue problems. These two problems have been suggested as simplified models of combustion phenomena. The first is the Bratu (or Gelfand) problem, which is a classical test problem for continuation methods (see, e.g., [6]), given by

$$\Delta u + \lambda e^u = 0 \text{ in } \mathcal{D}, \qquad u = 0 \text{ on } \partial\mathcal{D}. \tag{3.1}$$

The second is a problem considered by Chan [3] and given by

$$\Delta u + \lambda \left(1 + \frac{u + u^2/2}{1 + u^2/100}\right) = 0 \text{ in } \mathcal{D}, \qquad u = 0 \text{ on } \partial\mathcal{D}. \tag{3.2}$$

The Bratu problem has one fold point at $\lambda \approx 6.81$, and the Chan problem has two at $\lambda \approx 7.98$ and $\lambda \approx 6.41$.

Determining curves of solutions of these problems as λ varies is rather easy for path following methods, and the algorithm was kept appropriately simple. In the predictor–corrector method, each approximate unit tangent used in the Euler predictor was just the normalized difference of the current and previous points on the solution curve. In the corrector iterations, we took $\bar{t}_k$ to be this approximate unit tangent for each k. In particular, no null vectors of F' were computed.

Our main interest is in assessing the performance of the Krylov solvers on the corrector iterations with a fast Poisson solver preconditioner.[1] This preconditioner is very effective on (3.1) and (3.2) with fixed small λ and gives *mesh independent* Krylov solver convergence, i.e., convergence that does not slow as the grid is refined.

To investigate the effectiveness of this preconditioner and its mesh independence in the present context, we ran the algorithm on the two problems with four different grid sizes, using GMRES(40) and BiCGSTAB as the Krylov solvers. (In fact, convergence was always so fast that there were no GMRES restarts.) In each case, we calculated the geometric mean of successive linear residual norm ratios $\|r_{k+1}\|_2/\|r_k\|_2$ over all Krylov iterations at all corrector steps along the computed solution curve over a specified arclength interval containing the fold point(s).[2] The results are shown in Tables 3.1 and 3.2. In these tables, the convergence of both Krylov solvers is seen to be very fast overall. The convergence of GMRES is quite mesh independent; that of BiCGSTAB slows slightly but not greatly as the mesh is refined. These results suggest that convergence is at most mildly dependent on the grid size.

We also note that the convergence of GMRES and BiCGSTAB with this preconditioner was not adversely affected near fold points in these experiments. To illustrate this, for a 64×64 grid, we show in Figure 3.1 geometric means of the ratios $\|r_{k+1}\|_2/\|r_k\|_2$ at each continuation step along the curves, plotted over the $(\lambda, \|u\|_\infty)$-plane.

3.2. The driven cavity problem. This problem is a standard test problem for Navier–Stokes solvers and for continuation methods (see [6] and [12]), considered here in stream function form

$$\begin{aligned} &(1/Re)\Delta^2\psi + \frac{\partial\psi}{\partial x_1}\frac{\partial}{\partial x_2}\Delta\psi - \frac{\partial\psi}{\partial x_2}\frac{\partial}{\partial x_1}\Delta\psi = 0 \text{ in } \mathcal{D}, \\ &\psi = 0 \text{ and } \frac{\partial\psi}{\partial n} = g \text{ on } \partial\mathcal{D}, \end{aligned} \tag{3.3}$$

where $g = 0$ on the bottom and sides of the square and $g = 1$ on the top.

Our goal here is to demonstrate the ability of the methods in section 2 to determine curves of solutions of (3.3) as the Reynolds number Re varies when the "standard" centered-difference discretization of (3.3) is applied on a regular coarse grid. This can be a difficult problem, as explained further below. The discretization involves unknowns ψ_{ij} at interior gridpoints (ih, jh), $1 \le i, j \le m$, where $h = 1/(m+1)$. To treat the boundary conditions, gridpoints are added on and are exterior to the boundary by allowing $i, j = -1, 0, m+1, m+2$. The boundary conditions are imposed on the top boundary by requiring

$$\psi_{i,m+1} = 0, \quad \frac{\psi_{i,m+2} - \psi_{im}}{2h} = 1, \quad 1 \le i \le m.$$

Imposition of the boundary conditions on other boundary segments is similar.

It has been shown in [12] that this "standard" discretization gives rise to spurious nonphysical solutions on coarse grids. In particular, it is demonstrated in [12] that, on several coarse grids, the branch of physically significant solutions has two fold points, with loss of physical significance beyond the first. Following the solution curve through these fold points is difficult: care must be taken to remain near the curve in

[1] The MATLAB code used for this was the code accompanying [9].

[2] Arclength was normalized so that the curves traversed were similar for all discretizations.

TABLE 3.1
Bratu problem, geometric means of $\|r_{k+1}\|_2/\|r_k\|_2$.

Grid size	16×16	32×32	64×64	128×128
GMRES(40)	.0291	.0294	.0282	.0285
BiCGSTAB	.0681	.0961	.1091	.1278

TABLE 3.2
Chan problem, geometric means of $\|r_{k+1}\|_2/\|r_k\|_2$.

Grid size	16×16	32×32	64×64	128×128
GMRES(40)	.0207	.0197	.0196	.0205
BiCGSTAB	.0575	.0655	.0789	.0935

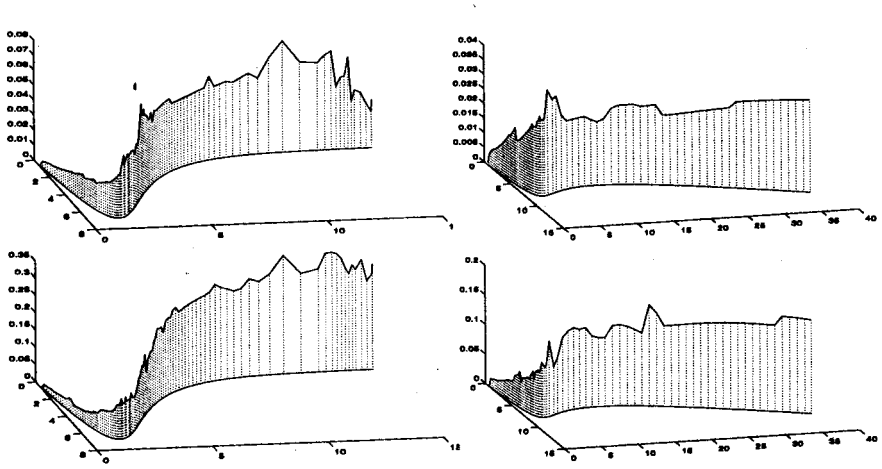

FIG. 3.1. *Bratu problem (left) and Chan problem (right), geometric means of* $\|r_{k+1}\|_2/\|r_k\|_2$ *along the solution curves for GMRES (top) and BiCGSTAB (bottom),* 64×64 *grid.*

the predictor steps and to return very close to it in the corrector steps. Accordingly, for these experiments, we modified the simple predictor–corrector algorithm outlined in section 3.1 to use the procedure in section 2.2 both to compute each unit tangent used by the Euler predictor and to compute each $\bar{t}_k$ in (1.2) to be a unit null vector of $F'(\bar{x}_k)$ as in the normal flow algorithm. The preconditioner was a biharmonic solver implemented by initially performing a Cholesky factorization of the discrete biharmonic operator and then using the factors for each preconditioner solve. This preconditioner is very effective for small values of Re but loses its effectiveness as Re grows. For the values of Re of ultimate interest here, BiCGSTAB was not very effective in our experiments. In the experiments noted below, GMRES(m) was used with a large restart value of $m = 200$ in order to obtain convergence in a tolerable number of iterations.

With very tight tolerances for computing approximate tangents and corrector steps and for terminating the corrector iterations, the algorithm succeeded in following the solution curves through the fold points on a number of coarse grids. The curves for four coarse grids are shown in Figure 3.2. Aside from slow convergence

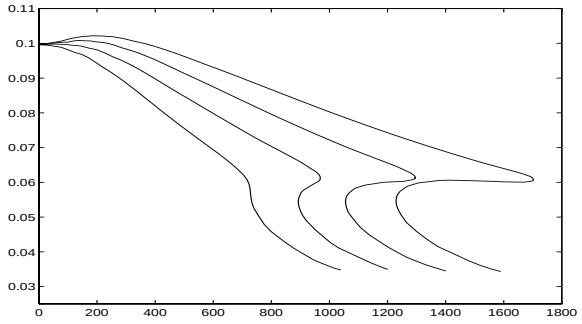

FIG. 3.2. *Continuation curves for the driven cavity problem, Re versus* $\|\psi\|_\infty$*, "standard" discretization,* $m \times m$ *grids with (left to right)* $m = 24, 28, 32, 36$.

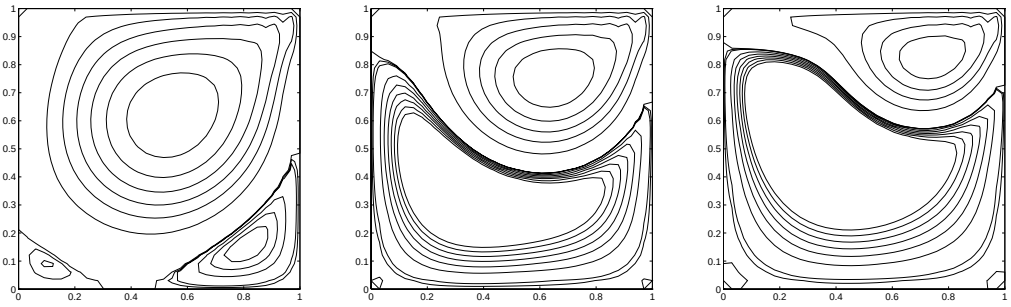

FIG. 3.3. *Three solutions of the driven cavity problem at* $Re = 1100$*, "standard" discretization,* 32×32 *grid. The left, middle, and right solutions are on the upper, middle, and lower branches of the curve, respectively.*

associated with large Reynolds numbers, no undue difficulties were observed with GMRES convergence near the fold points, either in computing corrector steps or in computing approximate null vectors of F'; such difficulties as occurred were associated with problem nonlinearity near the fold points. For an additional perspective on the nature of the spurious solutions, three solutions for $Re = 1100$ on a 32×32 grid are shown in Figure 3.3. The most physically significant solution, which suffers some degradation due to the coarse grid, lies on the upper branch of the curve and is shown on the left in Figure 3.3.

4. Summary. We have proposed a simple procedure for adapting Krylov subspace methods to the two major linear algebra tasks arising in path following (continuation, homotopy) methods, viz., solving the underdetermined linear systems that arise in Newton-like corrector iterations and computing approximate null vectors of the Jacobian (matrix). With this procedure, orthogonality constraints on approximate solutions are satisfied exactly, and problem conditioning is not worsened through poor scaling. Moreover, applying a Krylov subspace method using this procedure requires only products of the Jacobian (and perhaps its transpose) with vectors, as is usual with Krylov subspace methods. Preconditioners developed for the "fixed-parameter" case can be applied in a straightforward way. The effectiveness of the procedure has been demonstrated on several representative 2D elliptic PDE model problems.

In the interest of brevity, a number of important issues have been left open for future consideration. These include, most notably, the usefulness of the proposed pro-

cedure in treating bifurcation, the possibility of exploiting special problem structure, such as symmetry of D_xF, and the effectiveness of the proposed procedure with a broader range of test problems, Krylov subspace methods, and preconditioners.

REFERENCES

[1] E. Allgower and K. Georg, *Continuation and path following*, in Acta Numerica 1993, 2 (1993), pp. 1–64.

[2] P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.

[3] T. F. Chan, *Newton-like pseudo-arclength methods for computing simple turning points*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 135–148.

[4] C. Desa, K. M. Irani, C. J. Ribbens, L. T. Watson, and H. F. Walker, *Preconditioned iterative methods for homotopy curve tracking*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 30–46.

[5] R. W. Freund, G. H. Golub, and N. M. Nachtigal, *Iterative solution of linear systems*, in Acta Numerica 1992, 1 (1992), pp. 57–100.

[6] R. Glowinski, H. B. Keller, and L. Reinhart, *Continuation-conjugate gradient methods for the least-squares solution of nonlinear boundary value problems*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 793–832.

[7] M. H. Gutknecht, *Lanczos-type solvers for nonsymmetric linear systems of equations*, in Acta Numerica 1997, 6 (1997), pp. 271–398.

[8] K. M. Irani, M. P. Kamat, C. J. Ribbens, H. F. Walker, and L. T. Watson, *Experiments with conjugate gradient algorithms for homotopy curve tracking*, SIAM J. Optim., 1 (1991), pp. 222–251.

[9] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.

[10] B. N. Lundberg and A. B. Poore, *Variable order Adams–Bashforth predictors with an error-stepsize control for continuation methods*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 695–723.

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

[12] R. Schreiber and H. B. Keller, *Spurious solutions in driven cavity calculations*, J. Comput. Phys., 49 (1983), pp. 165–172.

[13] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.

[14] L. T. Watson, *Globally convergent homotopy algorithms for nonlinear systems of equations*, Nonlinear Dynam., 1 (1990), pp. 143–191.

[15] L. T. Watson, S. C. Billups, and A. P. Morgan, *Algorithm* 652: *HOMPACK: A suite of codes for globally convergent homotopy algorithms*, ACM Trans. Math. Software, 6 (1980), pp. 252–260.

[16] L. T. Watson, M. Sosonkina, R. C. Melville, A. P. Morgan, and H. F. Walker, *HOMPACK*90: *A suite of FORTRAN*90 *codes for globally convergent homotopy algorithms*, ACM Trans. Math. Software, 23 (1997), 514–549.