NUMERICAL LINEAR ALGEBRA ASPECTS OF GLOBALLY CONVERGENT HOMOTOPY METHODS

Layne T. Watson

The University of Michigan

Technical Report 86–7

February 28, 1986

NUMERICAL LINEAR ALGEBRA ASPECTS OF GLOBALLY CONVERGENT HOMOTOPY METHODS

LAYNE T. WATSON†

Abstract. Probability one homotopy algorithms are a class of methods for solving nonlinear systems of equations that are globally convergent with probability one. These methods are theoretically powerful, and if constructed and implemented properly, are robust, numerically stable, accurate, and practical. The concomitant numerical linear algebra problems deal with rectangular matrices, and good algorithms require a delicate balance (not always achieved) of accuracy, robustness, and efficiency in both space and time. The author's experience with globally convergent homotopy algorithms is surveyed here, and some of the linear algebra difficulties for dense and sparse problems are discussed.

Key words. nonlinear equations, globally convergent, homotopy algorithm, null space, conjugate gradient, curve tracking.

1. Introduction. Homotopies are a traditional part of topology, and have found significant application in nonlinear functional analysis and differential geometry. The concepts of homotopy maps, continuation, incremental loading, and invariant imbedding are widely used and intertwined. Because these terms are widely used and sometimes used interchangeably, the differences (and there are fundamental differences) are obscured. The context here is solving nonlinear systems of algebraic (as opposed to differential) equations, recognizing that these homotopy ideas have a much wider applicability. For the sake of discussion, let *homotopy methods* be the generic term, including continuation, parameter continuation, incremental loading, displacement incrementation, invariant imbedding, and continuous Newton methods.

The *raison d'être* for homotopy methods is global convergence (or at least a greatly expanded domain of convergence), as opposed to the local convergence of most iterative methods. For low dimensional and/or well understood problems, finding a good starting point for locally convergent methods is not difficult or expensive. However, as computer technology and our ambitions for solving ever larger and harder problems have increased, finding good initial estimates has become a difficult problem by itself. When computer time was unavailable or very expensive, human analysis to linearize the problem or otherwise manually calculate a starting point for a computer implemented locally convergent iterative method made good economic sense. Also, because of the slow early computers and high cost of computer time, numerous algorithms with endless variations were invented to achieve an epsilon improvement. Now, with computer power widely available and relatively cheap, computer time is far cheaper than the human analysis time required to devise a sufficiently good starting point for a locally convergent iterative method. The argument that locally convergent methods are more efficient than homotopy methods (even if that were true by a factor of ten) is less convincing now, considering the total (human plus computer) time and cost. If blackbox software for globally convergent homotopy algorithms can be made available which relieves the user of the burden of supplying a good initial estimate of the solution, the relative inefficiency of homotopy algorithms may be irrelevant. In effect, the intent is to transfer intelligence from the user to the homotopy algorithm, which, depending on your point of view, may or may not be a worthwhile goal.

Continuation and parameter continuation are well established techniques in numerical analysis, with the basic idea being to solve a series of problems as some parameter (intrinsic to the problem

† Department of Computer Science, Virginia Polytechnic Institute & State University, Blacksburg, VA 24061. This work was supported in part by Control Data Corporation Grant 84V101 and AFOSR Grant 85-0250. Current address: Departments of Electrical Engineering and Computer Science, Industrial and Operations Engineering, and Mathematics, University of Michigan, Ann Arbor, MI 48109.

or not) is slowly varied, using a locally convergent iterative technique for each problem, and the solution to the previous problem as starting point for the current problem [4, 24, 26, 36–39]. Similar techniques in engineering are known as incremental loading and displacement incrementation [9, 22, 27, 35, 40, 77, 79, 81]. A related idea, used in the specialized context of boundary value problems, is invariant imbedding [23, 46]. For the most part, all these methods had rather restrictive hypotheses and the connection with topology had not been made. A fundamental breakthrough occurred with the truly globally convergent simplicial fixed point algorithms of Scarf [45], Kuhn, Merrill [30], Eaves [13], Saigal [41], and Todd [42]. These algorithms were grounded in topology, constructive, potentially extremely powerful, but horribly inefficient in their early forms. Simplicial algorithms are now competitive, but will not be discussed further here, since their story is told very well in [4].

Another significant advance was the differential equation formulation of continuation, proposed in various forms by Davidenko [10], Smale [48], Boggs [5], Georg [17–18], Keller [24], Klopfenstein [25], Kubicek [26], Mejia [28], Menzel and Schwetlick [29], Meyer [31], Rheinboldt [36–39], Watson [69], and many others. Although the underlying homotopy map in these differential equation forms may have been the same as in "classical" continuation, it is important to realize that the *algorithms* and *implementations* are fundamentally different. Despite considerable success on practical problems and a large amount of supporting theory, all these homotopy methods suffered from a fatal flaw. A Jacobian matrix somewhere could become singular, and the computer implementation would either experience great difficulty or the method would fail completely (the more common occurrence).

The next advance was the development in 1976 by S. N. Chow, J. Mallet-Paret, and J. A. Yorke [7] of probability one homotopy methods (discovered independently in 1978 by H. B. Keller). The thorn of singular Jacobian matrices was finally removed, since these methods were specifically constructed to not have any singular points. The phrase "probability one" refers to the supporting theory, which says that for almost all (in the technical sense of Lebesgue measure) choices of some parameter vector involved in the homotopy map, there are no singular points and the method is globally convergent. Again, while globally convergent probability one homotopy algorithms may have a superficial resemblance to earlier homotopy algorithms, it is important to note that the *philosophy* is fundamentally new. Furthermore, because of this philosophical difference, there are subtle differences in mathematical software for probability one homotopy algorithms from the other continuation methods.

Beginning with the first robust computer implementation of probability one homotopy algorithms described in [69], a remarkable array of difficult problems have yielded to these techniques. Most of these problems either had not been solved or were difficult to solve by locally convergent iterative techniques (including state-of-the-art quasi-Newton algorithms as in [11] and [32]). These applications include fluid mechanics [20, 44, 49, 50, 53, 57, 58, 60, 70, 78], solid mechanics [22, 27, 51–57, 59, 61–67, 77, 79], chemical engineering [33, 34], nonlinear programming [12, 71, 73, 76], finite difference boundary value approximations [23, 75], spline collocation approximations [80], and fixed point computation [68, 69, 72, 76]. The remaining sections describe globally convergent probability one homotopy algorithms, with emphasis on the linear algebra. Sections 2 and 3 decribe algorithms for dense and sparse Jacobian matrices, respectively. More complete details can be found in the references. Section 4 mentions some gaps in our knowledge, and future research directions.

2. Algorithms for dense Jacobian matrices. Let E^p denote p-dimensional real Euclidean space. The following four lemmas from [74], [74], [73], [75] respectively, which follow from the results of [7], will be useful.

Lemma 1. Let $g : E^p \to E^p$ be a C^2 map, $a \in E^p$, and define $\rho_a : [0,1) \times E^p \to E^p$ by

$$\rho_a(\lambda, y) = \lambda g(y) + (1-\lambda)(y-a).$$

Then for almost all $a \in E^p$ there is a zero curve γ of ρ_a emanating from $(0,a)$ along which the Jacobian matrix $D\rho_a(\lambda, y)$ has full rank.

Lemma 2. If the zero curve γ in Lemma 1 is bounded, it has an accumulation point $(1,\bar{y})$, where $g(\bar{y}) = 0$. Furthermore, if $Dg(\bar{y})$ is nonsingular, then γ has finite arc length.

Lemma 3. Let $F : E^p \to E^p$ be a C^2 map such that for some $r > 0$, $xF(x) \geq 0$ whenever $\|x\| = r$. Then F has a zero in $\{x \in E^p \mid \|x\| \leq r\}$, and for almost all $a \in E^p$, $\|a\| < r$, there is a zero curve γ of

$$\rho_a(\lambda, x) = \lambda F(x) + (1-\lambda)(x-a),$$

along which the Jacobian matrix $D\rho_a(\lambda, x)$ has full rank, emanating from $(0,a)$ and reaching a zero $\bar{x}$ of F at $\lambda = 1$. Furthermore, γ has finite arc length if $DF(\bar{x})$ is nonsingular.

Lemma 3 is a special case of the following more general lemma.

Lemma 4. Let $F : E^p \to E^p$ be a C^2 map such that for some $r > 0$ and $\tilde{r} > 0$, $F(x)$ and $x - a$ do not point in opposite directions for $\|x\| = r$, $\|a\| < \tilde{r}$. Then F has a zero in $\{x \in E^p \mid \|x\| \leq r\}$, and for almost all $a \in E^p$, $\|a\| < \tilde{r}$, there is a zero curve γ of

$$\rho_a(\lambda, x) = \lambda F(x) + (1-\lambda)(x-a),$$

along which the Jacobian matrix $D\rho_a(\lambda, x)$ has full rank, emanating from $(0,a)$ and reaching a zero $\bar{x}$ of F at $\lambda = 1$. Furthermore, γ has finite arc length if $DF(\bar{x})$ is nonsingular.

The general idea of the algorithm is apparent from the lemmas: just follow the zero curve γ emanating from $(0, W)$ until a zero $\tilde{Y}$ of $F(Y)$ is reached (at $\lambda = 1$). Of course it is nontrivial to develop a viable numerical algorithm based on that idea, but at least conceptually, the algorithm for solving the nonlinear system of equations $F(Y) = 0$ is clear and simple. The homotopy map is

$$\rho_W(\lambda, Y) = \lambda F(Y) + (1-\lambda)(Y - W), \tag{1}$$

which has the same form as a standard continuation or embedding mapping. However, there are two crucial differences. In standard continuation, the embedding parameter λ increases monotonically from 0 to 1 as the trivial problem $Y - W = 0$ is continuously deformed to the problem $F(Y) = 0$. The present homotopy method permits λ to both increase and decrease along γ with no adverse effect; that is, turning points present no special difficulty. The second important difference is that there are never any "singular points" which afflict standard continuation methods. The way in which the zero curve γ of ρ_W is followed and the full rank of $D\rho_W$ along γ guarantee this. Observe that Lemma 1 guarantees that γ cannot just "stop" at an interior point of $[0,1) \times E^p$.

It should be mentioned that the form of the homotopy map $\rho_W(\lambda, Y)$ in (1) is just a special case used here for clarity of exposition. The more general theory can be found in [69, 73–75, 81], and practical engineering problems requiring a ρ_W nonlinear in λ are in [59] and [66].

The homotopy map ρ_a is a map from a $(p+1)$-dimensional space into a p-dimensional space. This mismatch of dimensions is a double-edged sword: The extra dimension permits "lifting" the original problem to obtain a problem with no singularities and a full rank Jacobian matrix, and which can be solved from an arbitrary starting point. However, the Jacobian matrix

$$D\rho_a(\lambda, x) = \Big[\overbrace{D_\lambda \rho_a(\lambda, x)}^{1} \quad \overbrace{D_x \rho_a(\lambda, x)}^{p}\Big] \Big\} p$$

of ρ_a is rectangular $(p \times (p+1))$, and this is an essential aspect of the lifted problem. By contrast, the Hessian matrices in nonlinear optimization and the Jacobian matrices in locally convergent algorithms for nonlinear systems of equations are always square. The $p \times (p+1)$ shape adds a combinatorial aspect to the numerical linear algebra, and this subtle difference is important in computer implementations. Even a symmetric $D_x\rho_a$ (as Hessian matrices are) does not help much (unless $D_x\rho_a$ is well-conditioned along the entire zero curve γ), because maintaining symmetry is at the expense of numerical stability. One proposal is to convert the $p \times (p+1)$ shape into a $(p+1) \times (p+1)$ shape maintaining symmetry and full rank; while mathematically feasible, this is fraught with practical difficulties, and is at best unnatural.

The Jacobian matrices associated with nonlinear least squares problems are also rectangular, but in the other direction $(m \times n,\ m > n)$. Redundant parameter and large residual nonlinear least squares problems have some of the same combinatorial flavor as the homotopy problem, but those problems have not been completely resolved yet either. Furthermore, if some part of the numerical linear algebra fails during a nonlinear least squares optimization, any descent direction will suffice. There is no similar fall back position for a homotopy algorithm.

The zero curve γ of the homotopy map $\rho_W(\lambda, Y)$ in (1) can be tracked by many different techniques; refer to the excellent survey [4] and recent work by Rheinboldt and Burkhardt[39] and Mejia[28]. The numerical results in [49–80] were obtained with preliminary versions of HOMPACK [81], a software package currently under development at Sandia National Laboratories, General Motors Research Laboratories, Virginia Polytechnic Institute and State University, and the University of Michigan. There are three primary algorithmic approaches to tracking γ : 1) an ODE-based algorithm based on that in [68], with several refinements; 2) a predictor-corrector algorithm whose corrector follows the flow normal to the Davidenko flow [16–18] (a "normal flow" algorithm); 3) a version of Rheinboldt's linear predictor, quasi-Newton corrector algorithm [39] (an "augmented Jacobian matrix" method).

First the ODE-based algorithm will be discussed. Assuming that $F(Y)$ is C^2 and W is such that Lemma 1 holds, the zero curve γ is C^1 and can be parametrized by arc length s. Thus $\lambda = \lambda(s)$, $Y = Y(s)$ along γ, and

$$\rho_W(\lambda(s), Y(s)) = 0 \tag{2}$$

identically in s. Therefore

$$\frac{d}{ds}\rho_W(\lambda(s), Y(s)) = D\rho_W(\lambda(s), Y(s)) \begin{pmatrix} \dfrac{d\lambda}{ds} \\ \dfrac{dY}{ds} \end{pmatrix} = 0, \tag{3}$$

$$\left\| \left(\frac{d\lambda}{ds}, \frac{dY}{ds} \right) \right\|_2 = 1. \tag{4}$$

If we take

$$\lambda(0) = 0, \quad Y(0) = W, \tag{5}$$

the zero curve γ is the trajectory of the initial value problem (3–5). When $\lambda(\bar{s}) = 1$, the corresponding $Y(\bar{s})$ is a zero of $F(Y)$. Thus all the sophisticated ODE techniques currently available can be brought to bear on the problem of tracking γ [47], [69].

Typical ODE software requires $(d\lambda/ds, dY/ds)$ explicitly, and (3), (4) only implicitly define the derivative $(d\lambda/ds, dY/ds)$. (It might be possible to use an implicit ODE technique for (3–4), but that seems less efficient than the method proposed here.) The derivative $(d\lambda/ds, dY/ds)$, which is a unit tangent vector to the zero curve γ, can be calculated by finding the one-dimensional kernel of the $p \times (p+1)$ Jacobian matrix

$$D\rho_W(\lambda(s), Y(s)),$$

which has full rank by Lemma 1. It is here that a substantial amount of computation is incurred, and it is imperative that the number of derivative evaluations be kept small. Once the kernel has been calculated, the derivative $(d\lambda/ds, dY/ds)$ is uniquely determined by (4) and continuity. Complete details for solving the initial value problem (3–5) and obtaining $Y(\bar{s})$ are in [68] and [69]. A discussion of the kernel computation follows.

The Jacobian matrix $D\rho_W$ is $p\times(p+1)$ with (theoretical) rank p. The crucial observation is that the last p columns of $D\rho_W$, corresponding to $D_Y\rho_W$, may not have rank p, and even if they do, some other p columns may be better conditioned. The objective is to avoid choosing p "distinguished" columns, rather to treat all columns the same (not possible for sparse matrices). Kubicek [26] and Mejia [28] have kernel finding algorithms based on Gaussian elimination and p distinguished columns. Choosing and switching these p columns is tricky, and based on *ad hoc* parameters. Also, computational experience has shown that accurate tangent vectors $(d\lambda/ds, dY/ds)$ are essential, and the accuracy of Gaussian elimination may not be good enough. A conceptually elegant, as well as accurate, algorithm is to compute the QR factorization with column interchanges [6] of $D\rho_W$,

$$Q\, D\rho_W\, P^t P z = \begin{pmatrix} * & \cdots & * & * \\ & \ddots & \vdots & \vdots \\ 0 & & * & * \end{pmatrix} P z = 0,$$

where Q is a product of Householder reflections and P is a permutation matrix, and then obtain a vector $z \in \ker D\rho_W$ by back substitution. Setting $(Pz)_{p+1} = 1$ is a convenient choice. This scheme provides high accuracy, numerical stability, and a uniform treatment of all $p+1$ columns. Finally,

$$\left(\frac{d\lambda}{ds}, \frac{dY}{ds} \right) = \pm \frac{z}{\|z\|_2},$$

where the sign is chosen to maintain an acute angle with the previous tangent vector on γ . There is a rigorous mathematical criterion, based on a $(p+1) \times (p+1)$ determinant, for choosing the sign, but there is no reason to believe that would be more robust than the angle criterion.

Remember that tracking γ was merely a means to an end, namely a zero $\tilde{Y}$ of $F(Y)$. Since γ itself is of no interest (usually), one should not waste computational effort following it too closely.

However, since γ is the only sure way to $\tilde{Y}$, losing γ can be disastrous. The tradeoff between computational efficiency and reliability is very delicate, and a fool-proof strategy appears difficult to achieve. None of the three primary algorithms alone is superior overall, and each of the three beats the other two (sometimes by an order of magnitude) on particular problems. Since the algorithms' philosophies are significantly different, a hybrid will be hard to develop.

The following version of the normal flow algorithm is derived from [16], and differs somewhat from that of [17] and [18]. The normal flow algorithm has three phases: prediction, correction, and step size estimation. (1) and (2) are the relevant equations here. For the prediction phase, assume that several points $P^{(1)} = (\lambda(s_1), Y(s_1))$, $P^{(2)} = (\lambda(s_2), Y(s_2))$ on γ with corresponding tangent vectors $(d\lambda/ds(s_1),\ dY/ds(s_1))$, $(d\lambda/ds(s_2),\ dY/ds(s_2))$ have been found, and h is an estimate of the optimal step (in arc length) to take along γ . The prediction of the next point on γ is

$$Z^{(0)} = p(s_2 + h), \tag{6}$$

where $p(s)$ is the Hermite cubic interpolating $(\lambda(s), Y(s))$ at s_1 and s_2. Precisely,

$$p(s_1) = (\lambda(s_1), Y(s_1)), \quad p'(s_1) = (d\lambda/ds(s_1), dY/ds(s_1)),$$
$$p(s_2) = (\lambda(s_2), Y(s_2)), \quad p'(s_2) = (d\lambda/ds(s_2), dY/ds(s_2)),$$

and each component of $p(s)$ is a polynomial in s of degree less than or equal to 3.

Starting at the predicted point $Z^{(0)}$, the corrector iteration is

$$Z^{(n+1)} = Z^{(n)} - \left[D\rho_W(Z^{(n)})\right]^{\dagger} \rho_W(Z^{(n)}), \qquad n = 0, 1, \ldots \tag{7}$$

where $\left[D\rho_W(Z^{(n)})\right]^{\dagger}$ is the Moore-Penrose pseudoinverse of the $p \times (p+1)$ Jacobian matrix $D\rho_W$. Small perturbations of W produce small changes in the trajectory γ , and the family of trajectories γ for varying W is known as the "Davidenko flow". Geometrically, the iterates given by (7) return to the zero curve along the flow normal to the Davidenko flow, hence the name "normal flow algorithm".

A corrector step ΔZ is the unique minimum norm solution of the equation

$$\left[D\rho_W\right]\Delta Z = -\rho_W. \tag{8}$$

Fortunately ΔZ can be calculated at the same time as the kernel of $\left[D\rho_W\right]$, and with just a little more work. Normally for dense problems the kernel of $\left[D\rho_W\right]$ is found by computing a QR factorization of $\left[D\rho_W\right]$, and then using back substitution. By applying this QR factorization to $-\rho_W$ and using back substitution again, a *particular* solution v to (8) can be found. Let $u \neq 0$ be any vector in the kernel of $\left[D\rho_W\right]$. Then the minimum norm solution of (8) is

$$\Delta Z = v - \frac{v^t u}{u^t u} u. \tag{9}$$

Since the kernel of $\left[D\rho_W\right]$ is needed anyway for the tangent vectors, solving (8) only requires another $O(p^2)$ operations beyond those for the kernel. The number of iterations required for convergence of (7) should be kept small (say < 4) since QR factorizations of $\left[D\rho_W\right]$ are expensive.

The alternative of using $\left[D\rho_W\left(Z^{(0)}\right)\right]$ for several iterations, which results in linear convergence, is rarely cost effective.

When the iteration (7) converges, the final iterate $Z^{(n+1)}$ is accepted as the next point on γ, and the tangent vector to the integral curve through $Z^{(n)}$ is used for the tangent–this saves a Jacobian matrix evaluation and factorization at $Z^{(n+1)}$. The step size estimation briefly described next attempts to balance progress along γ with the effort expended on the iteration (7).

Define a contraction factor

$$L = \frac{\left\|Z^{(2)} - Z^{(1)}\right\|}{\left\|Z^{(1)} - Z^{(0)}\right\|}, \tag{10}$$

a residual factor

$$R = \frac{\left\|\rho_W\left(Z^{(1)}\right)\right\|}{\left\|\rho_W\left(Z^{(0)}\right)\right\|}, \tag{11}$$

a distance factor $(Z^* = \lim_{n\to\infty} Z^{(n)})$

$$D = \frac{\left\|Z^{(1)} - Z^*\right\|}{\left\|Z^{(0)} - Z^*\right\|}, \tag{12}$$

and ideal values $\bar{L}$, $\bar{R}$, $\bar{D}$ for these three. The goal is to compute the "optimal" step size $\bar{h}$ for the next step based on the current step size h, the actual factors L, R, D, and the ideal factors $\bar{L}$, $\bar{R}$, $\bar{D}$. $\bar{h}$ must be reasonable, chosen in a way that prevents chattering, and consistent with h and the number of iterations in (7) required for convergence. The complete step size estimation scheme is sophisticated; details are in [80] and [81].

Rheinboldt's augmented Jacobian matrix algorithm together with step size strategies has been described very well elsewhere [39], and will not be repeated here.

3. Algorithms for sparse Jacobian matrices.

Large nonlinear systems of equations with sparse symmetric Jacobian matrices occur in many engineering disciplines, and each class of problems has special characteristics. Nonlinear structural mechanics problems [22, 27, 35, 40, 77, 79, 82] will be considered here, because they are representative of many problems outside structural mechanics, and yet have enough special features to admit efficient solution. The notation and terminology of engineering mechanics (which differs from the mathematicians' notation of the previous section) will be used in this section. The homotopy equation $\rho_W = 0$ becomes the equilibrium equations written in the form

$$F(x, \lambda) = 0 \tag{13}$$

where x, F are n-vectors and λ is a scalar (note the order of the arguments). x is the *displacement vector* and λ is the *load parameter*. The analog of the vector W in (1) is called the *load distribution* vector and is considered fixed. Assuming there are no bifurcation points, the zero set of $F(x, \lambda)$ is a smooth curve γ (the *equilibrium curve*) which does not intersect itself, and along which $DF(x, \lambda) = [D_xF(x, \lambda), D_\lambda F(x, \lambda)]$ has rank n. At a *limit point* (a turning point in mathematical jargon) $D_xF(x, \lambda)$ is singular, but the entire Jacobian matrix $DF(x, \lambda)$ still has rank n. It is this fact which is exploited by homotopy methods. The matrix $D_xF(x,\lambda)$ is symmetric (because it is the Hessian matrix of some energy potential function) and sparse with a "skyline" structure, such as

$$\begin{pmatrix}
\bullet_{1} & \bullet_{3} & & & & & & & & \\
\bullet & \bullet_{2} & \bullet_{5} & & \bullet_{11} & \bullet_{16} & & & & \\
 & \bullet & \bullet_{4} & \bullet_{7} & \bullet_{10} & \bullet_{15} & & & & \bullet_{31} \\
 & & \bullet & \bullet_{6} & \bullet_{9} & \bullet_{14} & & & & \bullet_{30} \\
 & \bullet & \bullet & \bullet & \bullet_{8} & \bullet_{13} & & & & \bullet_{29} \\
 & \bullet & \bullet & \bullet & \bullet & \bullet_{12} & \bullet_{18} & & & \bullet_{28} \\
 & & & & & \bullet & \bullet_{17} & \bullet_{20} & \bullet_{23} & \bullet_{27} \\
 & & & & & & \bullet & \bullet_{19} & \bullet_{22} & \bullet_{26} \\
 & & & & & & \bullet & \bullet & \bullet_{21} & \bullet_{25} \\
 & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet_{24}
\end{pmatrix}$$

Typically such matrices are stored in packed skyline format, in which the upper triangle is stored in a one-dimensional array indexed as shown above. The auxiliary array (1, 2, 4, 6, 8, 12, 17, 19, 21, 24, 32) of diagonal indices is also required. By convention the auxiliary integer array has length $n+1$ with the $(n+1)$st element containing the length of the packed real array plus one.

The differences mentioned in Section 2 between the structure of homotopy Jacobian matrices, Hessian matrices, nonlinear system Jacobian matrices, and nonlinear least squares Jacobian matrices are even more pronounced for large sparse problems. For realistic structural mechanics problems, with tens of thousands of variables, maintaining the symmetry and sparsity of $D_xF(x,\lambda)$ is absolutely essential. For postbuckling analysis, $D_xF(x,\lambda)$ will be singular and indefinite, so the $(n+1)$st column $D_\lambda F(x,\lambda)$ must be used in an essential way. This section illustrates again how subtle yet important facets of computational linear algebra are fundamental to the resolution of nonlinear problems.

Let the smooth equilibrium curve γ be parametrized by arc length s, so $x = x(s)$, $\lambda = \lambda(s)$ on γ and

$$F(x(s),\lambda(s)) = 0 \tag{14}$$

identically as a function of s. Then observe that $(x(s),\lambda(s))$ is the trajectory of the initial value problem

$$\frac{d}{ds}F(x(s),\lambda(s)) = \left[D_xF(x(s),\lambda(s)), D_\lambda F(x(s),\lambda(s))\right]\begin{pmatrix}\dfrac{dx}{ds}\\[2mm] \dfrac{d\lambda}{ds}\end{pmatrix} = 0, \tag{15a}$$

$$\left\|\left(\frac{dx}{ds},\frac{d\lambda}{ds}\right)\right\|_2 = 1, \tag{15b}$$

$$x(0) = x_0, \quad \lambda(0) = \lambda_0, \tag{15c}$$

where x_0, λ_0 is some initial point on γ . Since the Jacobian matrix has rank n along γ , the derivative $(dx/ds, d\lambda/ds)$ is uniquely determined by (15a, b) and continuity, and the initial value problem (15a, b, c) can be solved for $x(s)$, $\lambda(s)$. As before, the equilibrium curve γ can be tracked by an ODE-based, normal flow, or augmented Jacobian matrix algorithm. Note that in this context the interest is usually in γ itself, and not just a single point $x(\bar{s})$ corresponding to a specified value of $\lambda(\bar{s})$, although this latter situation does occur. If computing γ itself is the goal, then a Newton-iteration-based algorithm is more appropriate than an ODE-based algorithm, because the latter will drift away from γ (although computational experience on some very difficult problems suggests that this drift is negligible [77]).

Regardless of whether γ itself or $x(\bar{s})$ corresponding to $\lambda(\bar{s}) = 1$ is required, the problem is still to solve the initial value problem (15), which requires calculating the implicitly defined derivative (tangent vector) $(dx/ds, d\lambda/ds)$. The difficulty now is that the first n columns of the Jacobian matrix $DF(x, \lambda)$ are definitely special, and any attempt to treat all $n+1$ columns uniformly would be disastrous from the point of view of storage allocation. Any algorithm requires computing the kernel of the $n \times (n+1)$ matrix $DF(x, \lambda)$, which has rank n. This can be elegantly and efficiently done for small dense matrices, but the large sparse Jacobian matrix of structural mechanics presents special difficulties. Numerous engineering approaches [9, 22, 27, 35, 40, 77, 82] exist, but they all either fail completely or destroy the matrix structure if they hit exactly at a limit point on γ. The novel approach taken here is to solve $DF\, y = 0$ using a preconditioned conjugate gradient algorithm that takes full advantage of sparsity and symmetry, and is mathematically correct and numerically stable even exactly at limit points. This conjugate gradient algorithm will now be described.

Let $(\bar{x}, \bar{\lambda})$ be a point on the equilibrium curve γ , and $\bar{y}$ the unit tangent vector to γ at $(\bar{x}, \bar{\lambda})$ in the direction of increasing arc length s. Let $|\bar{y}_k| = \max_i |\bar{y}_i|$. Then the matrix

$$A = \begin{bmatrix} DF(x, \lambda) \\ e_k^t \end{bmatrix} \tag{16}$$

where e_k is a vector with 1 in the kth component and zeros elsewhere, is invertible at $(\bar{x}, \bar{\lambda})$ and in a neighborhood of $(\bar{x}, \bar{\lambda})$ by continuity. Thus the kernel of DF can be found by solving the linear system of equations

$$Ay = \bar{y}_k e_{n+1} = b. \tag{17}$$

Given any nonsymmetric, nonsingular matrix A, the system of linear equations $Ay = b$ can be solved by considering the linear system

$$AA^t z = b.$$

Since the coefficient matrix for this system is both symmetric and positive definite, the system can be solved by a conjugate gradient algorithm. Once a solution vector z is obtained, the vector y from the original system can be computed as $y = A^t z$. An implementation of the conjugate gradient algorithm in which y is computed directly, without reference to z, any approximations of z, or AA^t, was originally proposed by Hestenes [21], and is commonly known as Craig's method [15]. Each iterate y^i minimizes the Euclidean error norm $\|y - y^i\|$ over the translated Krylov space

$$y^0 + span\{r^0, AA^t r^0, (AA^t)^2 r^0, \ldots, (AA^t)^{i-1} r^0\},$$

where $r^0 = b - Ay^0$. Below $\langle u, v \rangle$ denotes the inner product $u^t v$.

Craig's Method:

```
    Choose $y^0$;
    Compute $r^0 = b - Ay^0$;
    Compute $p^0 = A^t r^0$;
For $i = 0$ step 1 until convergence do
    BEGIN
    $\alpha_i = \langle r^i, r^i \rangle / \langle p^i, p^i \rangle$
    $y^{i+1} = y^i + \alpha_i p^i$
    $r^{i+1} = r^i - \alpha_i A p^i$
    $\beta_i = \langle r^{i+1}, r^{i+1} \rangle / \langle r^i, r^i \rangle$
    $p^{i+1} = A^t r^{i+1} + \beta_i p^i$
    END
```

Let Q be any nonsingular matrix. The solution to the system $Ay = b$ can be calculated by solving the system

$$By = (Q^{-1}A)y = Q^{-1}b = g. \tag{18}$$

The use of such a matrix is known as preconditioning. Since the goal of using preconditioning is to decrease the computational effort needed to solve the original system, Q should be some approximation to A. Then $Q^{-1}A$ would be close to the identity matrix, and the iterative method described above would converge more rapidly when applied to (18) than when applied to (17). In the following algorithm B and g are never explicitly formed. The algorithm given above can be obtained by substituting the identity matrix for Q.

Craig's method using a preconditioner:

```
    Choose $y^0$, $Q$;
    Compute $r^0 = b - Ay^0$;
    Compute $\tilde{r}^0 = Q^{-1} r^0$;
    Compute $p^0 = A^t Q^{-t} \tilde{r}^0$;
For $i = 0$ step 1 until convergence do
    BEGIN
    $\alpha_i = \langle \tilde{r}^i, \tilde{r}^i \rangle / \langle p^i, p^i \rangle$
    $y^{i+1} = y^i + \alpha_i p^i$
    $\tilde{r}^{i+1} = \tilde{r}^i - \alpha_i Q^{-1} A p^i$
    $\beta_i = \langle \tilde{r}^{i+1}, \tilde{r}^{i+1} \rangle / \langle \tilde{r}^i, \tilde{r}^i \rangle$
    $p^{i+1} = A^t Q^{-t} \tilde{r}^{i+1} + \beta_i p^i$
```

```
END
```

For this algorithm, a minimum of $5(n+1)$ storage locations is required (in addition to that for A). The vectors y, $\tilde{r}$, and p all require their own locations; $Q^{-t}\tilde{r}$ can share with Ap; $Q^{-1}Ap$ can share with $A^tQ^{-t}\tilde{r}$. The computational cost per iteration of this algorithm is:

1) two preconditioning solves $(Q^{-1}v$ and $Q^{-t}v)$;

2) two matrix-vector products $(Av$ and $A^tv)$;

3) $5(n+1)$ multiplications (the inner products $\langle p,p\rangle$ and $\langle\tilde{r},\tilde{r}\rangle$, αp, βp, and $\alpha Q^{-1}Ap$).

The coefficient matrix A in the linear system of equations (17), whose solution y yields the kernel of $DF(\bar{x},\bar{\lambda})$, has a very special structure which can be exploited if (17) is attacked indirectly as follows. Note that the leading $n\times n$ submatrix of A is D_xF, which is symmetric and sparse, but possibly indefinite. Write

$$A = M + L \tag{19}$$

where

$$M = \begin{bmatrix} D_xF(\bar{x},\bar{\lambda}) & c \\ c^t & d \end{bmatrix},$$

$$L = ue^t_{n+1}, \quad u = \begin{pmatrix} D_\lambda F(\bar{x},\bar{\lambda}) - c \\ 0 \end{pmatrix}.$$

The choice of e^t_k as the last row of A to make A invertible is somewhat arbitrary, and in fact any vector (c^t, d) outside a set of measure zero (a hyperplane) could have been chosen. Thus for almost all vectors c the first n columns of M are independent, and similarly almost all $(n+1)$-vectors are independent of the first n columns of M. Therefore for almost all vectors (c^t, d) both A and M are invertible. Assume that (c^t, d) is so chosen.

Using the Sherman-Morrison formula (L is rank one), the solution y to the original system $Ay = b$ can be obtained from

$$y = \left[I - \frac{M^{-1}ue^t_{n+1}}{(M^{-1}u)^te_{n+1} + 1}\right] M^{-1}b. \tag{20}$$

which requires the solution of two linear systems with the sparse, symmetric, invertible matrix M. It is the systems $Mz = u$ and $Mz = b$ to which Craig's preconditioned conjugate gradient algorithm is actually applied.

The only remaining detail is the choice of the preconditioning matrix Q. Q is taken as the modified Cholesky decomposition of M, as described by Gill and Murray [19]. If M is positive definite and well conditioned, $Q = M$. Otherwise, Q is a well conditioned positive definite approximation to M. The Gill-Murray factorization algorithm can exploit the symmetry and sparse skyline structure of M, and this entire scheme, Equations (17–20), is built around using the symmetry and sparse skyline structure of the Jacobian matrix D_xF.

The splitting in (19) is a key idea, since M is nicer to deal with than A both theoretically and algorithmically. In typical applications M is a low rank modification of a symmetric positive definite matrix, so M can be preconditioned very well making Craig's method and (20) efficient.

The sparse skyline structure of M also facilitates storage management. Computational results verify that the approach of Equations (17–20) is superior to a preconditioned conjugate gradient algorithm applied directly to A, which has been done in the engineering literature.

4. Future prospects. One noteworthy difference between general curve tracking algorithms and homotopy curve tracking algorithms is the objective: the curve itself for general curve tracking and the solution at the end of the curve for homotopy algorithms. This means that the curve itself is not important, and sophisticated homotopy algorithms (as in HOMPACK, e.g.) actually *change* the curve that is being tracked as they proceed. This strategy has a rigorous theoretical justification, since changing the curve amounts to changing the parameter vector in the homotopy map, and for almost all parameter vectors the zero curve of the homotopy map is guaranteed to reach a solution. Furthermore, homotopy algorithms are inherently stable because all the zero curves are confluent at the solution–the ideal situation where "all roads lead to Rome". However, following the zero curve γ too loosely can be disastrous [69], so there is a delicate balance between efficiency and robustness. This balance needs further study, perhaps leading to fundamentally different algorithms.

Each of the three primary algorithms described here has several parameters that can be "fine tuned" for a given problem. Normally these parameters don't matter, with a poor choice producing inefficiency but not failure. Work needs to be done on the significance and sensitivity of these parameters, and on how they can be adapted as the algorithm proceeds.

Currently, for both dense and sparse problems, the $n \times (n+1)$ Jacobian matrix is evaluated, factored, and a kernel calculated at every step. There are many possibilities for savings here, and these need to be systematically explored and compared. The evidence to date is that the current scheme is the best, but no conclusive results exist.

The most challenging numerical linear algebra problem is how to handle large sparse Jacobian matrices elegantly. The (inelegant) scheme proposed here works reasonably well for structural mechanics problems, but sparse homotopy algorithms are just in their infancy. Much more research needs to be done on the iterative scheme used to calculate the kernels, the matrix splitting (Rheinboldt [38] uses a different splitting from that proposed here), and the preconditioning matrix. While the skyline sparsity structure (and nearly positive definite symmetric Jacobian matrices $D_xF(x,\lambda)$) considered here covers a wide class of problems, it by no means covers all common sparsity patterns or arbitrary sparsity patterns. A leading candidate for solution next is the nonsymmetric "bordered" pattern (common in chemical engineering):

$$\begin{bmatrix} \mathcal{A}_1 & 0 & \cdots & 0 & \mathcal{C}_1 & * \\ 0 & \mathcal{A}_2 & \cdots & 0 & \mathcal{C}_2 & * \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \mathcal{A}_{k-1} & \mathcal{C}_{k-1} & * \\ \mathcal{B}_1 & \mathcal{B}_2 & \cdots & \mathcal{B}_{k-1} & \mathcal{A}_k & * \end{bmatrix}$$

5. Acknowledgement. The author is indebted to the referees for several very good suggestions, and to all the colleagues and students who over the years have contributed to this work.

6. References.

[1] **J. C. Alexander**, *The topological theory of an imbedding method*, in Continuation Methods (H. G. Wacker, Ed.), Academic Press, New York, 1978, pp. 37–68.

[2] **J. C. Alexander and J. A. Yorke**, *The homotopy continuation method: numerically implementable topological procedures*, Trans. Amer. Math. Soc., 242 (1978), pp. 271–284.

[3] **J. C. Alexander, T.-Y. Li, and J. A. Yorke**, *Piecewise smooth homotopies*, in Homotopy Methods and Global Convergence, B. C. Eaves, F. J. Gould, H.-O. Peitgen, and M. J. Todd, eds., Plenum, New York, 1983, pp. 1–14.

[4] **E. Allgower and K. Georg**, *Simplicial and continuation methods for approximating fixed points*, SIAM Rev., 22 (1980), pp. 28–85.

[5] **P. Boggs**, *The solution of nonlinear systems of equations by A-stable integration techniques*, SIAM J. Numer. Anal., 8 (1971), pp. 767–785.

[6] **P. Businger and G. G. Golub**, *Linear least squares solutions by Householder transformations*, Numer. Math., 7 (1965), pp. 269–276.

[7] **S. N. Chow, J. Mallet-Paret, and J. A. Yorke**, *Finding zeros of maps: homotopy methods that are constructive with probability one*, Math. Comp., 32 (1978), pp. 887–899.

[8] **S. N. Chow, J. Mallet-Paret, and J. A. Yorke**, *A homotopy method for locating all zeros of a system of polynomials*, in Functional Differential Equations and Approximation of Fixed Points, H.-O. Peitgen and H. O. Walther, eds., Springer-Verlag Lecture Notes in Math. 730, New York, 1979, pp. 228–237.

[9] **M. A. Crisfield**, *Incremental/iterative solution procedures for nonlinear structural analysis*, Internat. Conf. on Numer. Methods for Nonlinear Problems, Swansea, UK, 1980, pp. 261–290.

[10] **D. F. Davidenko**, *On the approximate solution of systems of nonlinear equations*, Ukrain. Mat. Ž., 5 (1953), pp. 196–206.

[11] **J. E. Dennis and R. Schnabel**, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[12] **J. P. Dunyak, J. L. Junkins, and L. T. Watson**, *Robust nonlinear least squares estimation using the Chow-Yorke homotopy method*, J. Guidance, Control, Dynamics, 7 (1984), pp. 752–755.

[13] **B. C. Eaves**, *Homotopies for the computation of fixed points*, Math. Programming, 3 (1972), pp. 1–22.

[14] **B. C. Eaves and R. Saigal**, *Homotopies for computation of fixed points on unbounded regions*, Math. Programming, 3 (1972), pp. 225–237.

[15] **D. K. Fadeev and V. N. Fadeeva**, *Computational Methods of Linear Algebra*, Freeman, London, 1963.

[16] **K. Georg**, private communication, 1981.

[17] **K. Georg**, *A note on stepsize control for numerical curve following*, in Homotopy Methods and Global Convergence, B. C. Eaves, F. J. Gould, H.-O. Peitgen, and M. J. Todd, eds., Plenum, New York, 1983, pp. 145–154.

[18] **K. Georg**, *Numerical integration of the Davidenko equation*, in Numerical Solution of Nonlinear Equations, E. Allgower, K. Glashoff, H.-O. Peitgen, eds., Springer-Verlag Lecture Notes in Math. 878, New York, 1981.

[19] **P. E. Gill and W. Murray**, *Newton type methods for unconstrained and linearly constrained optimization*, Math. Programming, 7 (1974), pp. 311–350.

[20] **M. W. Heruska, L. T. Watson, and K. K. Sankara**, *Micropolar flow past a porous stretching sheet*, Comput. & Fluids, to appear.

[21] **M. R. Hestenes**, *The conjugate gradient method for solving linear systems*, in Proc. Symp. Appl. Math., 6, Numer. Anal., AMS, New York, 1956, pp. 83–102.

[22] **M. P. Kamat, L. T. Watson, and V. B. Venkayya**, *A quasi-Newton versus a homotopy method for nonlinear structural analysis*, Comput. & Structures, 17 (1983), pp. 579–585.

[23] **H. B. Keller**, *Numerical Solution of Two-point Boundary Value Problems*, Society for Industrial and Applied Mathematics, Philadelphia, 1976.

[24] ——, *Numerical solution of bifurcation and nonlinear eigenvalue problems*, in Applications of Bifurcation Theory, Academic Press, New York, 1977.

[25] **R. W. Klopfenstein**, *Zeros of nonlinear functions*, J. Assoc. Comput. Mech., 8 (1961), pp. 336–373.

[26] **M. Kubicek**, *Dependence of solutions of nonlinear systems on a parameter*, ACM-TOMS, 2 (1976), pp. 98–107.

[27] **H. H. Kwok, M. P. Kamat, and L. T. Watson**, *Location of stable and unstable equilibrium configurations using a model trust region quasi-Newton method and tunnelling*, Comput. & Structures, 21 (1985), pp. 909–916.

[28] **R. Mejia**, *CONKUB: A conversational path-follower for systems of nonlinear equations*, J. Comput. Phys., to appear.

[29] **R. Menzel and H. Schwetlick**, *Zur Lösung parameterabhängiger nichtlinearer Gleichungen mit singulären Jacobi-Matrizen*, Numer. Math., 30 (1978), pp. 65–79.

[30] **O. Merrill**, *Applications and extensions of an algorithm to compute fixed points of upper semi-continuous mappings*, Doctoral thesis, I. O. E. Dept., Univ. of Michigan, Ann Arbor, 1972.

[31] **G. Meyer**, *On solving nonlinear equations with a one-parameter operator embedding*, SIAM J. Numer. Anal., 5 (1968), pp. 739–752.

[32] **J. J. Moré, B. S. Garbow, and K. E. Hillstrom**, *User Guide for MINPACK-1*, ANL-80-74, Argonne National Laboratory, Argonne, IL (1980).

[33] **A. P. Morgan**, *A transformation to avoid solutions at infinity for polynomial systems*, Appl. Math. Comput., 18 (1986), pp. 77–86.

[34] ——, *A homotopy for solving polynomial systems*, Appl. Math. Comput., 18 (1986), pp. 87–92.

[35] **J. Padovan**, *Self-adaptive predictor-corrector algorithm for static nonlinear structural analysis*, NASA CR–165410, 1981.

[36] **W. C. Rheinboldt and C. Den Héyér**, *On steplength algorithms for a class of continuation methods*, SIAM J. Numer. Anal., 18 (1981), pp. 925–947.

[37] **W. C. Rheinboldt**, *On the computation of critical boundaries on equilibrium surfaces*, SIAM J. Numer. Anal., 19 (1982), pp. 653–669.

[38] **W. C. Rheinboldt and R. Melhem**, *A comparison of methods for determining turning points of nonlinear equations*, Computing, 29 (1982), pp. 103–114.

[39] **W. C. Rheinboldt and J. V. Burkardt**, *Algorithm 596: A program for a locally parameterized continuation process*, ACM Trans. Math. Software, 9 (1983), pp. 236-241.

[40] **E. Riks**, *An incremental approach to the solution of snapping and buckling problems*, Internat. J. Solids Structures, 15 (1979), pp. 529–551.

[41] **R. Saigal**, *On the convergence rate of algorithms for solving equations that are based on methods of complementary pivoting*, Math. Operations Res., 2 (1977), pp. 108–124.

[42] **R. Saigal and M. J. Todd**, *Efficient acceleration techniques for fixed point algorithms*, SIAM J. Numer. Anal., 15 (1978), pp. 997–1007.

[43] **R. Saigal**, *An efficient procedure for traversing large pieces in fixed point algorithms*, in Homotopy Methods and Global Convergence, B. C. Eaves, F. J. Gould, H.-O. Peitgen, and M. J. Todd, eds., Plenum, New York, 1983, pp. 239–248.

[44] **K. K. Sankara and L. T. Watson**, *Micropolar flow past a stretching sheet*, Z. Angew. Math. Phys., to appear.

[45] **H. Scarf**, *The approximation of fixed points of a continuous mapping*, SIAM J. Appl. Math., 15 (1967), pp. 1328–1343.

[46] **M. R. Scott**, *Invariant Imbedding and its Applications to Ordinary Differential Equations*, Addison-Wesley, Reading, MA, 1973.

[47] **L. F. Shampine and M. K. Gordon**, *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*, W. H. Freeman, San Francisco, 1975.

[48] **S. Smale**, *Convergent process of price adjustment and global Newton methods*, J. Math. Econom., 3 (1976), pp. 107–120.

[49] **C. Y. Wang and L. T. Watson**, *Squeezing of a viscous fluid between elliptic plates*, Appl. Sci. Res., 35 (1979), pp. 195–207.

[50] ——, *Viscous flow between rotating discs with injection on the porous disc*, Z. Angew. Math. Phys., 30 (1979), pp. 773–787.

[51] ——, *On the large deformations of C-shaped springs*, Internat. J. Mech. Sci., 22 (1980), pp. 395–400.

[52] ——, *Theory of the constant force spring*, J. Appl. Mech., 47 (1980), pp. 956–958.

[53] ——, *The fluid-filled cylindrical membrane container*, J. Engrg. Math., 15 (1981), pp. 81–88.

[54] ——, *Equilibrium of heavy elastic cylindrical shells*, J. Appl. Mech., 48 (1981), pp. 582–586.

[55] ——, *The elastic catenary*, Internat. J. Mech. Sci., 24 (1982), pp. 349–357.

[56] ——, *Free rotation of a circular ring about a diameter*, Z. Angew. Math. Phys., 34 (1983), pp. 13–24.

[57] **C. Y. Wang, L. T. Watson, and M. P. Kamat**, *Buckling, postbuckling and the flow through a tethered elastic cylinder under external pressure*, J. Appl. Mech., 50 (1983), pp. 13–18.

[58] **L. T. Watson, T. Y. Li and C. Y. Wang**, *Fluid dynamics of the ellilptic porous slider*, J. Appl. Mech., 45 (1978), pp. 435–436.

[59] **L. T. Watson and C. Y. Wang**, *A homotopy method applied to elastica problems*, Internat. J. Solids Structures, 17 (1981), pp. 29–37.

[60] ——, *Deceleration of a rotating disc in a viscous fluid*, Phys. Fluids, 22 (1979), pp. 2267–2269.

[61] ——, *The circular leaf spring*, Acta Mech., 40 (1981), pp. 25–32.

[62] ——, *Hanging an elastic ring*, Internat. J. Mech. Sci., 23 (1981), pp. 161–168.

[63] ——, *Overhang of a heavy elastic sheet*, Z. Angew. Math. Phys., 33 (1982), pp. 17–23.

[64] ——, *Periodically supported heavy elastic sheet*, J. Engrg. Mech., 109 (1983), pp. 811–820.

[65] ——, *The equilibrium states of a heavy rotating column*, Internat. J. Solids Structures, 19 (1983), pp. 653–658.

[66] **L. T. Watson and W. H. Yang**, *Optimal design by a homotopy method*, Applicable Anal., 10 (1980), pp. 275–284.

[67] ——, *Methods for optimal engineering design problems based on globally convergent methods*, Comput. & Structures, 13 (1981), pp. 115–119.

[68] **L. T. Watson and D. Fenner**, *Chow-Yorke algorithm for fixed points of zeros of C^2 maps*, ACM TOMS, 6 (1980), pp. 252–260.

[69] **L. T. Watson**, *A globally convergent algorithm for computing fixed points of C^2 maps*, Appl. Math. Comput., 5 (1979), pp. 297–311.

[70] ——, *Numerical study of porous channel flow in a rotating system by a homotopy method*, J. Comput. Appl. Math., 7 (1981), pp. 21–26.

[71] ——, *Computational experience with the Chow-Yorke algorithm*, Math. Programming, 19 (1980), pp. 92–101.

[72] ——, *Fixed points of C^2 maps*, J. Comput. Appl. Math., 5 (1979), pp. 131–140.

[73] ——, *Solving the nonlinear complementarity problem by a homotopy method*, SIAM J. Control Optim., 17 (1979), pp. 36–46.

[74] ——, *An algorithm that is globally convergent with probability one for a class of nonlinear two-point boundary value problems*, SIAM J. Numer. Anal., 16 (1979), pp. 394–401.

[75] ——, *Solving finite difference approximations to nonlinear two-point boundary value problems by a homotopy method*, SIAM J. Sci. Stat. Comput., 1 (1980), pp. 467–480.

[76] ——, *Engineering applications of the Chow-Yorke algorithm*, Appl. Math. Comput., 9 (1981), pp. 111–133.

[77] **L. T. Watson, S. M. Holzer, and M. C. Hansen**, *Tracking nonlinear equilibrium paths by a homotopy method*, Nonlinear Anal., 7 (1983), pp. 1271–1282.

[78] **L. T. Watson, K. K. Sankara, and L. C. Mounfield**, *Deceleration of a porous rotating disk in a viscous fluid*, Internat. J. Engrg. Sci., 23 (1985), pp. 131–137.

[79] **L. T. Watson, M. P. Kamat, and M. H. Reaser**, *A robust hybrid algorithm for computing multiple equilibrium solutions*, Engrg. Comput., 2 (1985), pp. 30–34.

[80] **L. T. Watson and M. R. Scott**, *Solving spline collocation approximations to nonlinear two-point boundary value problems by a homotopy method*, Tech. Rep. CS84015-R, Dept. of Computer Sci., VPI&SU, Blacksburg, VA, 1984.

[81] **L. T. Watson, S. C. Billups, and A. P. Morgan**, *HOMPACK: A suite of codes for globally convergent homotopy algorithms*, Tech. Rep. 85–34, Dept. of Industrial and Operations Eng., Univ. of Michigan, Ann Arbor, MI, 1985.

[82] **G. A. Wempner**, *Discrete approximations related to nonlinear theories of solids*, Internat. J. Solids Structures, 7 (1971), pp. 1581–1599.