Lecture Notes in Mathematics

Edited by A. Dold and B. Eckmann

878

Numerical Solution of Nonlinear Equations

Proceedings, Bremen 1980

Edited by E. L. Allgower, K. Glashoff and H.-O. Peitgen

Springer-Verlag
Berlin Heidelberg New York 1981

Editors

Eugene L. Allgower
Mathematics Department, Colorado State University
Fort Collins, Colorado 80523, USA

Klaus Glashoff
Institut für Angewandte Mathematik
Universität Hamburg, Bundesstr. 55
2000 Hamburg 13, Federal Republic of Germany

Heinz-Otto Peitgen
Fachbereich Mathematik, FS „Dynamische Systeme"
Universität Bremen
2800 Bremen 33, Federal Republic of Germany

AMS Subject Classifications (1980): 35Q20, 35R35, 35R45, 47H10, 65-06, 65H10, 65H15, 65L10, 65L15, 65N20, 65N99, 65Q05

ISBN 3-540-10871-8 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-10871-8 Springer-Verlag New York Heidelberg Berlin

CIP-Kurztitelaufnahme der Deutschen Bibliothek
Numerical solution of nonlinear equations : proceedings, Bremen, 1980 / ed. by E. L. Allgower . . . – Berlin; Heidelberg; New York: Springer, 1981
(Lecture notes in mathematics; Vol. 878)
ISBN 3-540-10871-8 (Berlin, Heidelberg, New York)
ISBN 0-387-10871-8 (New York, Heidelberg, Berlin)
NE: Allgower, Eugene L. [Hrsg.]; GT

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr.
2141/3140-543210

dedicated to the memory of

Emanuel Sperner

A DERIVATIVE-FREE ARC CONTINUATION METHOD AND A BIFURCATION TECHNIQUE

by

Ralph Baker Kearfott

ABSTRACT

Algorithms and comparison results for a derivative-free predictor-corrector method for following arcs of $H(x,t) = \theta$, where $H : R^n \times [0,1] \rightarrow R^n$ is smooth, are given. The method uses a least-change secant update for H', adaptive controlled predictor stepsize, and Powell's indexing procedure to preserve linear independence in the updates. Considerable savings in numbers of theoretical function calls are observed over high order methods requiring explicit H'. The framework of a promising technique for handling general bifurcation problems is presented.

key words: arc continuation, quasi-Newton methods, least change secant updates, Brouwer degree, numerical computation, nonlinear algebraic systems, Powell's method.

1. Introduction

An approach to the numerical analysis of nonlinear systems in R^n is to study arcs of $H(x,t) = \theta$, where $H : R^n \times [0,1] \rightarrow R^n$ is smooth (cf. e.g., [2], pp. 39-48). This technique is used, for example, to study nonlinear eigenvalue problems and to solve algebraic systems for which Newton's method is not globally convergent. The original such methods, referred to as the "Davidenko" or "continuous Newton" methods, have been improved. Present solution techniques involve integrating the initial value problem:

$$H'(y(s))y'(s) = \theta, \quad \|y'(s)\| = 1, \quad y(0) = y^0 \tag{1}$$

where H' is the n by $n+1$ Jacobi matrix of H and $y'(s)$ is the componentwise derivative of $y = (x,t) \in R^{n+1}$ relative to arclength s.

The integration may be effected by high-order methods [16] or by "predictor-corrector" techniques (e.g., [13], [4], [5], [14]). In the latter, a direction $b^0 \in R^{n+1}$ is found to approximately satisfy $H'(y^0)b^0 = \theta$, and the

predicted value $y(\delta_0)$ is set to:

$$z^0 = y^0 + \delta_0 b^0 . \tag{2}$$

Corrections to z^0 are made by applying Newton's method to the system:

$$G(z) = \begin{pmatrix} H(z) \\ (b^0)^t (z - z^0) \end{pmatrix} = \theta . \tag{3}$$

Note that this corrects z^0 in a hyperplane perpendicular to the step b^0 (cf. [4]). The entire process is repeated to obtain sequences y^i, b^i, z^i, and δ_i. The length δ_{i+1} can be chosen according to the angle between b^{i-1} and b^i ([14]), but in any case must be such that both the algorithm functions efficiently and the corrector iterations are stable.

Disadvantages of such arc continuation methods include the necessity of computing H' several times per predictor-corrector step. Also, new derivative-free techniques to handle multiple bifurcations (i.e., to pass points y, $H(y) = \theta$ where the null space of $H'(y)$ is of dimension greater than 1) are desirable.

Simplicial methods (cf. [2]) are derivative-free and have been applied to nonlinear bifurcation (see [8] and references therein). However, difficulties remain in the interplay between the triangulation, mesh, tracing of bifurcation branches, and the proximity of approximate solution arcs to true ones.

The purpose of this paper is to present a derivative-free arc continuation algorithm modelled on the predictor-corrector approach. In Section 2 the arc continuation algorithm is presented and explained. In Section 3 we give some numerical comparisons. In Section 4 a derivative-free method for bifurcation problems is presented.

It should be mentioned that Kurt Georg has independently developed similar derivative-free path-following algorithms, to appear in the SIAM Journal on Scientific and Statistical Computing [7] and in these proceedings [6]. Georg also gives a method of handling odd-order bifurcation points. Specific techniques from those methods and the method given below may be merged to effect improvements.

2. The Algorithm

The algorithm follows the general pattern outlined above and in [14].

The techniques herein may be applied to a variety of settings. For explanatory purposes, however, we assume H is of the form:

$$H(z) = H(x,t) = t\,f(x) + (1-t)\,g(x) \tag{4}$$

where $z = (x,t) \in R^n \times R$, $f : R^n \to R^n$, and $g : R^n \to R^n$. We also assume that a root x^0; $g(x^0) = \theta$ is known; the object is to find roots of F by following (possibly bifurcating) arcs of $H(z) = \theta$ from $(x^0, 0)$ to the $t = 1$ hyperplane.

The main modifications of the general scheme ([14] and above) are: (1) use of a least-change secant update to H' [3] instead of complete evaluation of H'; (2) use of Powell's indexing ([15], pp. 133-138) to assure accuracy in H'; and (3) special choice and adjustment of the stepsize to assure stability and accurate H'. The least-change update is given in Algorithm 2.1, Step 8, and is documented in [3], while the Powell indexing procedure is given in Algorithm 2.2 (infra) and is documented in [15].

Several parameters in Algorithm 2.1 are chosen to control the inner iteration and stepsize. These include the initial predictor stepsize δ_0, the maximum allowable predictor step δ_{max}, the criterion c_i for doubling the stepsize, and the criterion c_d for halving the stepsize. As in [4], $0 < c_d < c_i < 1$, where $\delta_{i+1} \leftarrow 2\delta_i$ if $b^i \circ b^{i-1} > c_i$, but $\delta_{i+1} \leftarrow \delta_i/2$ if $b^i \circ b^{i-1} < c_d$. (In all tests, the algorithm functioned well with $c_d = .95$ and $c_i = .99$.)

Additional parameters include the predictor function magnitude tolerance ε_δ, the maximum number of inner iterations N_i, the singular matrix indicator ε_{mat}, the inner iteration convergence criterion ε_y, the $t = 1$ convergence criterion ε_t, the relative stepsize for finite differences Δ, and the eigenvalue criterion ε_e. Upon taking a predictor step: $z \leftarrow y + \delta b$ (Steps 5-6 of Algorithm 2.1), δ is halved and z is revised if:

$$\|H(z)\|(n+1)^{1/2} / \|H'\|_F > \varepsilon_\delta, \tag{5}$$

where $\|\cdot\|_F$ is the Frobenius norm. If the number of inner iterations between successive predictor steps exceeds $N_i + n + 1$, H' is reinitialized using finite differences, δ is halved, and the initial predictor step is revised. In solving (3) during the inner iteration, it is necessary to solve $G'X = -G$ repeatedly; if, after normalizing G', a maximum Gaussian elimination pivot element of magnitude less than ε_{mat} is found in partial pivoting, G' is considered singular. In that case, the technique in Section 4 can be used to continue. The inner iteration is terminated when $\|H\|(n+1)^{1/2} / \|H'\|_F < \varepsilon_y$, whereas

outer iteration is terminated when $\|y_{n+1} - 1\| = |t - 1| < \varepsilon_t$. The additional parameter ε_e is used to determine when eigenvalues of H'^tH' are approximately equal to 0; this will be explained in Section 4. M_z is an estimate of the maximum magnitude of any point on the arc.

2.1 Algorithm

0. Input the dimension n, the function $H : R^{n+1} \to R^n$, δ_0, δ_{max}, c_i, c_d, ε_δ, N_i, ε_{mat}, ε_y, ε_t, Δ, ε_e, and M_z.
1. Set $y \leftarrow y^0$; compute $H \leftarrow H(y)$ and compute $H' \leftarrow H'(y)$ using finite differences with stepsize $\|y\|\Delta$; initialize the Powell vectors: $d^i \leftarrow e^i$, $i = 1, \ldots, n+1$, where e^i is the i-th coordinate vector in R^{n+1}, and $\omega_i \leftarrow 1$, $i = 1, \ldots, n+1$; set flag $f_1 \leftarrow 1$. ($f_1 = 1$ indicates H' has just been initialized.)
2. Initialize b so that $H'b = \theta$. (cf. [4].)
3. (Initialization of the stepsize and counter for outer iterations) $\delta \leftarrow \delta_0$, nit $\leftarrow 1$.
4. (Initialization of the counter for inner iterations) ninit $\leftarrow 1$.
5. (Take predictor step) $s \leftarrow \delta b$; $z^0 \leftarrow y + s$; $z \leftarrow z^0$; $H_{old} \leftarrow H$.
6. $H \leftarrow H(z)$.
7. (Halve predictor step if new H value is too large in magnitude) If $\|H\|(n+1)^{1/2} / \|H'\|_F > \varepsilon_\delta$, do the following:
   (a) $H \leftarrow H_{old}$.
   (b) $\delta \leftarrow \delta/2$.
   (c) Return to Step 4.
   Otherwise, continue.
8. Make a "least change" (Broyden) update to H': $H' \leftarrow H' + (H - H_{old} - H's)s^t / \|s\|^2$.
9. (Powell's checking procedure applied to the predictor step)
   (a) Compute the inner products: $a_i \leftarrow (s/\|s\|) \circ d^i$, $i = 1, \ldots, n+1$.
   (b) If $\omega_1 \geq 2(n+1)$ and $2|a_1| < 1$ do (i)-(vi); otherwise, continue to Step 9(c). (Here, a special correction update is made to H' if necessary.)
      (i) $s \leftarrow \Delta d^1$; $z \leftarrow z + s$.
      (ii) $H_{old} \leftarrow H$; $H \leftarrow H(z)$.
      (iii) $H' \leftarrow H' + (H - H_{old} - H's)s^t / \|s\|^2$.
      (iv) $a_i \leftarrow (s/\|s\|) \circ d^i$, $i = 1, \ldots, n+1$.

(v) $H \leftarrow H_{old}$; $z \leftarrow z - s$.

(vi) Proceed to Step 21.

Steps 10-23 involve refinement of the predictor step (i.e., inner iteration).

(c) Update the Powell indices ω_i and d^i, $i = 1, \ldots, n+1$.

10. (Reinitialization if maximum number of inner iterations has been exceeded) If $ninit > N_i + n + 1$, do Step 11; otherwise, continue to Step 12.

11. (Reinitialization)

(a) $\delta \leftarrow \delta/2$.

(b) $H \leftarrow H(y)$.

(c) Initialize the Powell indices and Powell vectors: $\omega_i \leftarrow 1$, $d^i \leftarrow e^i$, $i = 1, \ldots, n+1$.

(d) Set $H' \leftarrow H'(y)$, where $H'(y)$ is computed using finite differences.

(e) Set the initialization indicator flag : $f_1 \leftarrow 1$.

(f) Return to Step 4.

12. (Storage of convergence criteria) $\tau_1 \leftarrow |s_{n+1}|$; $\tau_2 \leftarrow \|H\|(n+1)^{1/2}/\|H'\|_F$.

13. Compute a new corrector step: $s \leftarrow -(G')^{-1}G$, where

$$G = \begin{pmatrix} H \\ b^t(z - z^0) \end{pmatrix} \quad \text{and} \quad G' = \begin{pmatrix} H' \\ b^t \end{pmatrix}.$$

14. If a singular G' is detected in Step 13 (cf. the explanation of ε_{mat} above) then do the following:

(a) (Reinitialize if the singularity is possibly due to the update process) If $f_1 = 0$, go to Step 11; otherwise, continue to (b).

(b) Compute and store direction vectors b, y, δ, and nit via Algorithm 4.1.

(c) Retrieve a direction vector b, y, δ and nit via Algorithm 4.2.

(d) nit $\leftarrow$ nit + 1.

(e) Go to Step 4.

Otherwise, continue.

15. (Execute Powell's special step if independence is not maintained) If $\omega_1 \geq 2(n+1)$ and $2|a_1| < 1$, go to step 9(b); otherwise, continue to Step 16.

16. (Termination of inner iteration if convergence has been achieved) If $\tau_2 < \varepsilon_y$ and $\tau_1 < |z_{n+1}|/10$ or $ninit = 1$, go to Step 24;

otherwise, continue. (Convergence in the t variable is tested separately in case of poor scaling.)

17. (Reinitialize if the ratio of magnitudes of the corrector step to the predictor step is too large) If $\|s\|/\delta > 2(1 - c_d^2)/c_d$, go to Step 11; otherwise, continue.

18. $a_i \leftarrow (s/\|s\|) \circ d^i$, $i+1, \ldots, n+1$.

19. (Take the corrector step) $z \leftarrow z + s$; $H_{old} \leftarrow H$; $H \leftarrow H(z)$.

20. (Make a Broyden update as in Step 8 and reset initialization indicator flag)
    (a) $H' \leftarrow H' + (H - H_{old} - H's)s^t/\|s\|^2$.
    (b) $f_1 \leftarrow 0$.

21. Update the Powell indices ω_i and the Powell vectors d^i, $i = 1, \ldots, n+1$ via Algorithm 2.2.

22. (Advance counter for inner iterations) ninit $\leftarrow$ ninit + 1.

23. (Do another inner iteration) Return to Step 10.

24. (Reset initialization flag in case no inner iterations were necessary) $f_1 \leftarrow 0$.

The remaining steps consider possible reasons for ending the outer iterations.

25. (Divergence or a return to the $t = 0$ hyperplane) If $z_{n+1} \leq 0$ or if $z \geq M_z$, then do the following:
    (a) Print an appropriate message.
    (b) If there are no more bifurcation branches to be considered, then stop; otherwise, continue to step (c).
    (c) Retrieve a vector b, y, δ, and nit via Algorithm 4.2.
    (d) nit $\leftarrow$ nit + 1; go to Step 4.

26. (Termination if the $t = 1$ hyperplane has been successfully reached) If $|z_{n+1} - 1| < \varepsilon_t$ do the following:
    (a) Store z.
    (b) If there are no more bifurcation branches to be considered, then stop; otherwise, continue to Step (c).
    (c) Retrieve a vector b, y, δ, and nit via Algorithm 4.2.
    (d) nit $\leftarrow$ nit + 1; go to Step 4.

27. (Interpolation if the $t = 1$ hyperplane has been passed) If $z_{n+1} > 1$, do the following:
    (a) Find the point $q = (q_1, q_2, \ldots, q_{n+1})$ on the line connecting y and

z such that $q_{n+1} = 1$.

(b) $y \leftarrow q$; $s \leftarrow y - z$; $z^0 \leftarrow y$; $z \leftarrow z^0$; $b \leftarrow e^{n+1}$; $H \leftarrow H(y)$.

(c) (Refinement of the interpolation at the $t = 1$ hyperplane by inner iteration) Go to Step 8.

In the remaining steps, δ is adjusted and a new b is computed for further outer iteration.

28. $b_{old} \leftarrow b$; $b \leftarrow (z - y)/\|z - y\|$; $y \leftarrow z$.

29. (Adjusting the stepsize according to the angle between the previous and present direction)

(a) If $b \circ b_{old} > c_i$, set $\delta \leftarrow 2\delta$.

(b) If $\delta > \delta_{max}$, set $\delta \leftarrow \delta_{max}$.

(c) If $b \circ b_{old} < c_d$, set $\delta \leftarrow \delta/2$.

30. (Increment counter for number of outer iterations) nit $\leftarrow$ nit + 1.

31. (Do another outer iteration) Go to Step 4.

The following algorithm, given and explained in [15], pp. 133-138, is for keeping track of the directions in which generalized secant updates are made. It is repeated here for convenience.

Algorithm 2.2 (Powell's indexing)

0. Input the vectors s and d^i, $i = 1, \ldots, n+1$ and the scalars ω_i and a_i, $i = 1, \ldots, n+1$ from Algorithm 2.1.
1. Set m equal to the smallest k: $1 \le k \le n + 1$ such that $\sum_{i=1}^{k} a_i^2 \ge 1/4$.
2. $\omega_j \leftarrow \omega_j + 1$ for $j = 1$ to $m - 1$.
3. $\omega_j \leftarrow \omega_{j+1} + 1$ for $j = m$ to n.
4. $\omega_{n+1} \leftarrow 1$.
5. $d^1_{new} \leftarrow d^m$; $d^i_{new} \leftarrow d^{i-1}$ for $i = 2$ to m; $d^i \leftarrow d^i_{new}$, for $i = 1$ to m.
6. $a_{1,new} \leftarrow a_m$; $a_{i,new} \leftarrow a_{i-1}$, $i = 2$ to m; $a_i \leftarrow a_{i,new}$, $i = 1$ to m.
7. Set $r \leftarrow a_1^2$ and $\sigma_i \leftarrow 0$, for $i = 1$ to $n + 1$.
8. Repeat the following in sequence for $i = 2$ to $n + 1$:

(a) denom $\leftarrow (r(r + a_i^2))^{1/2}$.

(b) For $j = 1$ to $n + 1$:

(i) $\sigma_j \leftarrow \sigma_j + a_{i-1} d_j^{i-1}$

(ii) $d_j^{i-1} \leftarrow (r d_j^i - a_i \sigma_j)/\text{denom}$.

(c) $r \leftarrow r + a_i^2$.

9. $d^{n+1} \leftarrow s/\|s\|$.

10. Return to Algorithm 2.1.

3. Numerical Results

Introduction of the Broyden update with special Powell steps, the choice of predictor direction, and special control of predictor and corrector step lengths have these ends: (1) provision of an arc-continuation algorithm applicable where derivatives are difficult to obtain; (2) provision of a more efficient algorithm; and (3) provision of a more reliable algorithm. To test the achievement of these ends, we have made comparisons on the following four problems used by Watson in [12]:

(1) $f_k(x) = x_k - (1/2n)(\sum_{i=1}^{n} x^3 + k)$, $k = 1$ to n;

(2) $f_k(x) = .01(\sum_{i=k-1}^{k+1} x_i + 1)^3$, $k = 1$ to n;

(3) $f_k(x) = \exp(\cos(k \sum_{k=1}^{n} x_i))$, $k = 1$ to n;

(4) $f_1(x) = x_1 - (\prod_{i=1}^{n} x_i - 1)$, $f_j(x) = x_j - (\sum_{i=1}^{n} x_i + x_j - (n+1))$, $j = 2$ to n.

In all cases, the homotopy used was $H(x,t) = (1-t)x + t(x - F(x))$, where $F(x) = (f_1(x), f_2(x), \ldots, f_n(x))$ and the object was to find fixed points of F.

The results for these functions are presented in Tables 1 to 4, respectively. In each case, the dimension, the number of function evaluations, the number of inner iterations, and the number of function evaluations per inner iteration are given in columns 1 through 4. "Equivalent" function evaluations, given in the fifth column, are computed for Watson's test runs by multiplying the number of Jacobi matrix evaluations Watson's algorithm required by the dimension. The ratio of equivalent function evaluations to function evaluations is given in the last column.

Double precision was used in the Fortran program on a Honeywell 68/80 (36 bit word length). In all cases, $\delta_0 = .1$, $c_i = .99$, and $c_d = .95$. The maximum predictor steplength δ_{max} was not limited except in the fourth problem, for $n = 10$, $n = 20$, $n = 25$, and $n = 45$. This was necessary since H' is ill-conditioned in the fourth problem near $t = 1$.

In all results listed, the fixed point of F was found to at least 12 significant digits.

Except for the extremely nonlinear function (Table 3) and except for certain runs with the ill-conditioned function (Table 4), the derivative-free method showed a definite advantage. In many cases, the total number of function evaluations for an outer iteration was less than that required for a single Jacobi matrix evaluation. Furthermore, roundoff and truncation possibly accumulate in Watson's direct high-order scheme, but such errors are corrected in general predictor-corrector methods.

It should be pointed out that it is often possible to compute an n by $n + 1$ Jacobi matrix with less than the equivalent of n evaluations of H. For this reason, our method of comparison would be most valid for complicated functions and functions which are difficult to encode.

In this preliminary version, G' was not updated directly; hence, $(n + 1)^3$ were required per inner iteration to solve the algebraic system given in (3).

n	eval.	nit	eval./nit	eq. eval.	eq. eval./eval.
10	32	4	8.	500	15.6
20	46	5	9.2	800	17.4
30	57	5	11.4	1020	17.9
40	67	5	13.4	1840	27.5
50	76	5	15.2	1800	23.7
60	86	5	17.2	2280	26.5
70	95	5	19.	3500	36.8
80	105	5	21.	4320	41.1
90	114	5	22.8	5040	44.2
100	132	6	21.	3400	25.8

Table 1. $f_k(x) = x_k - (1/2n)\left(\sum_{i=1}^{n} x_i^3 + k\right)$, $k = 1$ to n.

n	eval.	nit	eval./nit	eq. eval.	eq. eval./eval.
10	52	4	13.	360	6.9
20	57	4	14.3	720	12.6
30	72	4	18.	1080	15.
40	82	4	20.5	1440	17.6
50	89	4	22.3	1800	20.2
60	99	4	24.8	2160	21.8
70	109	4	27.3	2520	23.1
80	119	4	29.8	2880	24.2
90	129	4	32.3	3240	25.1
100	139	4	34.8	3600	25.9

Table 2. $f_k(x) = .01\left(\sum_{i=k-1}^{k+1} x_i + 1\right)^3$, k = 1 to n.

n	eval.	nit	eval./nit	eq. eval.	eq. eval./eval.
2	71	7	10.	88	1.2
3	536	32	16.8	663	1.2
4	840	56	15.	892	1.1
5	1485	93	16.	2565	1.7
6	1787	117	15.3	7272	4.1
7	3039	183	16.6	13860	4.6
8	4849	294	16.5	23792	4.9
9	5950	371	16.	33210	5.6
10	8078	476	17.	46440	5.7

Table 3. $f_k(x) = \exp\left(\cos\left(k \sum_{i=1}^{n} x_i\right)\right)$, k = 1 to n.

n	eval.	nit	eval./nit	eq. eval.	eq. eval./eval.
5	115	6	19.2	260	2.3
10*	451	38	11.8	740	1.6
15	424	14	30.3	1455	3.4
20*	1320	62	19.4	1460	1.1
25*	1371	67	20.5	2025	1.5
30	1000	17	58.8	3240	3.2
35	898	17	52.8	4235	4.7
40	851	16	53.2	4840	5.7
45*	3419	370	9.2	5535	1.6
50	897	17	52.8	6450	7.2

Table 4. $f_1(x) = x_1 - (\prod_{i=1}^{n} x_i - 1)$,

$f_j(x) = x_j - (\sum_{i=1}^{n} x_i + x_j - (n+1))$, $j = 2$ to n.

Special parameters were used for the starred dimensions (see text).

4. Bifurcation

Corrector iteration in Algorithm 2.1 fails when H' becomes ill-conditioned or singular, i.e., when the null space of H' effectively has dimension greater than 1. Indeed, at such (and only such) points y^*, $H(y^*) = \theta$, the manifold structure of $H^{-1}(\theta)$ may break down, and two or more arcs of $H^{-1}(\theta)$ may intersect.

H. B. Keller [13] has proposed several procedures for following all such arcs emanating from y^*. However, these involve evaluation of second partial derivatives of H or other drawbacks. Here, we outline a different general technique which will always work in theory and which can be expected to function reasonably well when the effective dimension of the null space of H' is 5 or less.

Suppose that the solution arcs $\{y(s)\} \subset R^{n+1}$ of $H(y) = \theta$ are smooth except at bifurcation points y^*. It can then be shown that all such arcs intersecting at y^* must be tangent to the tangent space Π to y^* generated by the null space of $H'(y^*)$. If $\{z^1, z^2, \ldots, z^k\}$ is a basis for the null space of $H'(y^*)$ and $\delta > 0$, consider $\Pi_\delta = \{y^* + v \mid v = \sum_{i=1}^{k} a_i z^i, \ -\delta \le a_i \le \delta\}$. For δ

sufficiently small, the local minimum points $\{m^1, m^2, \ldots, m^q\}$ of $\|H\| \mid \partial\Pi_\delta$ correspond to points on solution curves bifurcating from y^*. Thus, predictor directions for such curves at y^* are given by:

$$b^i = (m^i - y^*)/\|m^i - y^*\|, \quad i = 1 \text{ to } q. \tag{6}$$

When $k \leq 5$, finding the m^i involves searching $2k$ cubes of dimension 4 or less. That can possibly be done by using numerical computation of the topological degree and a generalized method of bisection to find roots of grad $(\|H\|)$ within the low dimensional parameter space [9], [10], [11]. This approach will find all such m^i since the Kronecker index of any continuous grad $(\phi) \neq 0$ at minima of ϕ, where its sign depends upon the dimension k of the space. Furthermore, grad $(\|H\|)$ need not be approximated with high accuracy for proper functioning of the bisection algorithm.

Algorithm 4.1 uses the above approach. It consists of two phases: (1) computation of an orthonormal basis for the null space of $H'(y^*)$; and (2) computation of the direction vectors b^i in (5) and their storage, along with other information necessary to continue Algorithm 2.1 from y^*.

The first phase may be executed by computing the eigenvalues and eigenvectors of the symmetric matrix $H'^t H'$. First, the eigenvectors $\{u^i\}_{i=1}^{n+1}$ and corresponding eigenvalues $\{\lambda_i\}_{i=1}^{n+1}$ are computed; the basis $\{v^i\}_{i=1}^{k}$ for the tangent hyperplane is then obtained by selecting those u^i for which $|\lambda_i| < \varepsilon_e$.

Three interrelated parameters from Algorithm 2.1 govern the selection of $\{v^i\}_{i=1}^{k}$: δ_{max}, ε_{mat}, and ε_e. If all bifurcation points are to be found, ε_{mat} must be sufficiently large, since the iterates y^i cannot be expected to be closer than $\delta_{max}/2$ to bifurcation points; likewise, ε_e must be sufficiently large to detect the approximate null space. On the other hand, if ε_{mat} and ε_e are too large, false directions may be given or Algorithm 4.1 will be invoked repeatedly in the vicinity of a bifurcation point, reducing efficiency or causing redundant tracing of a single arc. In such cases, all three of δ_{max}, ε_{mat}, and ε_e may need to be reduced.

The second phase can be executed by computation of the topological degree and a generalized method of bisection. (In the case $k = 1$, H' is not really singular; $b \leftarrow \pm v^1$ without further computation.) A good bisection method is documented in [10]; details will not be given here. Search of the $(k-1)$-dimensional parallelopiped:

(7) $$\Pi_{\delta,i} = \{y^* \pm \delta v^i + \sum_{\substack{j=1 \\ j\neq i}}^{k} a_j v^j, \quad -\delta \leq a_j \leq \delta\}$$

is begun by computing the Brouwer degree of grad ($\|H\|$) at θ relative to a canonically shaped simplex S containing $\Pi_{\delta,i}$; if this degree is 0, no relative minima exist on $\Pi_{\delta,i}$. Otherwise, "Whitney" bisection (ibid.) is applied to S, each iteration of which yields simplexes with the same shape as S but with diameters equal to precisely half that of S. A sufficient number of such iterations is done to reduce the diameter, nominally, to $.1\delta$, while all simplexes having non-zero Brouwer degree are considered. The m^i are set equal to the resulting barycenters.

Computing the Brouwer degree requires either choice of a heuristic parameter or coding of a bound on the modulus of continuity of the components of grad ($\|H\|$) whenever $k > 2$.

When an m^i and b^i are computed, y^*, b^i, nit, and δ' are stored in a stack. Here, nit, y, and δ are from Algorithm 2.1, while $\delta' = 20\delta$. We need δ' so large that the next iteration of Algorithm 2.1 will not encounter an effectively singular H', but small enough so that the arc can be followed. (The factor of 20 was found to be satisfactory in test cases.)

Algorithm 4.1

0. Input n, H', y, nit, δ, b, and ε_e.
1. Compute the $n+1$ by $n+1$ matrix H'^tH'.
2. Compute the eigenvalues and orthonormal set of eigenvectors of H'^tH'.
3. Store those eigenvectors $\{v^i\}_{i=1}^k$ whose corresponding eigenvalues λ_i satisfy $|\lambda_i| < \varepsilon_e$.
4. If $k = 1$, do the following:
   (a) $\delta \leftarrow 20\delta$.
   (b) If $v^1 \cdot b > -.7$, $b \leftarrow v^1$; otherwise, $b \leftarrow -v^1$.
   (We make this check to avoid going in the direction opposite to that Algorithm 2.1 traversed to get to y^*.)
   (c) Store δ, nit, y, and b in stack $\mathcal{S}$.
   (d) Return to Algorithm 2.1.
5. If $k > 1$, do Steps 6-9 for $i = 1$ to k and sgn = -1 and sgn = 1:
6. $p \leftarrow \text{sgn} \cdot v^i$.

7. Set $w^j \leftarrow v^j$, $j = 1$ to $i - 1$ and set $w^{j-1} \leftarrow v^j$, $j = i + 1$ to k.

8. Using $X = (x_1, x_2, \ldots, x_{k-1})$, find roots of $\phi(X) = \text{grad}\,(\|H(z_X)\|)$, where $z_X = y + \delta p + \sum_{j=1}^{k-1} x_j w^j$.

9. Suppose $(x_1^1, x_2^1, \ldots, x_{k-1}^1), \ldots, (x_1^q, x_2^q, \ldots, x_{k-1}^q)$ are the roots found in Step 8. Define:
$b^\ell = (\delta p + \sum_{j=1}^{k-1} x_j^\ell w^j) / \|\delta p + \sum_{j=1}^{k-1} x_j^\ell w^j\|$, $\ell = 1$ to q. If $b \circ b > -.7$, store 20δ, nit, y, and b^l in stack $\mathcal{S}$, for $l = 1$ to q.

10. Return to Algorithm 2.1.

The stack $\mathcal{S}$ is expanded whenever bifurcation points are encountered in Algorithm 2.1. Algorithm 2.1 follows an arc until it is determined to diverge, terminate at $t = 1$, or terminate at $t = 0$. Upon such termination, the algorithm continues with values of δ, nit, y, and b retrieved from $\mathcal{S}$ via the following:

Algorithm 4.2 (Retrieval)

0. Let $\mathcal{S}$ be the stack of δ, nit, y, and b.
1. If $\mathcal{S}$ is empty, stop.
2. Retrieve δ, nit, y, and b from $\mathcal{S}$.
3. $z \leftarrow y + b$.
4. Initialize the Powell indices: $\omega_i \leftarrow 1$, $d^i \leftarrow e^i$, $i = 1$ to n.
5. $H' \leftarrow H'(z)$, where H' is computed using finite differences.
6. $H \leftarrow H(y)$.
7. Return to Algorithm 2.1 .

Actual detection of the bifurcation points may be effected by monitoring any of several determinants. For example, the determinants of $G(z)$ (formula 3) could be averaged over sequences of $2n + 3$ iterations to obtain approximate values. Alternatively, the reciprocals of the condition numbers of H' (i.e., the ratios of the smallest to largest singular values of H') could be computed directly. The condition number of H' may even be approximated in $o(n^2)$ operations [1].

Detailed algorithms for approximating and refining even-order bifurcation points, in addition to specifics on finding the $m^{(i)}$ in the general tangent hyperplane setting, appear in [12]. There results of several numerical experiments involving bifurcation are given. These results look promising with regard to the handling of arbitrary types of bifurcation points, even in cases where the

arcs intersect tangentially; it is feasible to apply the techniques to various classes of problems. However, the algorithms in [12] are subject to improvement; also, additional tests on realistic problems and comparisons with other methods are needed to determine the practical value of each of the individual techniques.

References

1. A.K. Cline, C.B. Moler, G.W. Stewart, and J.H. Wilkinson, "An estimate for the condition number of a matrix," SIAM J. Numer. Anal. 16, #2, 1979.
2. E. Allgower and K. Georg, "Simplicial and continuation methods for approximating fixed points and solutions to systems of equations," SIAM Review 22, no. 1, January, 1980.
3. J.E. Dennis, Jr. and R.B. Schnabel, "Least change secant updates for quasi-Newton methods," SIAM Review 21 no. 4, 1979 (pp. 443-459).
4. C.B. Garcia and T.Y. Li, "On a path-following method for systems of equations," MRC Technical Summary Report #1983, Mathematics Research Center, University of Wisconsin - Madison, July, 1979.
5. C.B. Garcia and W.I. Zangwill, "Finding all solutions to polynomial systems and other systems of equations," Math. Prog. 16, 1979 (pp. 159-176).
6. Kurt Georg, "Numerical integration of the Davidenko equation," these proceedings.
7. Kurt Georg, "On tracing an implicitly defined curve by quasi-Newton steps and calculating bifurcation by local perturbations," to appear in the SIAM Journal on Scientific and Statistical Computing.
8. Harmut Jürgens, Heinz-Otto Peitgen, and Dietmar Saupe, "Topological perturbations in the numerical nonlinear eigenvalue and bifurcation problems," Proc. Conf. Analysis and Computation of Fixed Points, Academic Press, New York-London, 1980, 139-181.
9. R.B. Kearfott, "An efficient degree-computation method for a generalized method of bisection," Numer. Math. 32, 1979 (pp. 109-127).
10. R.B. Kearfott, "An improved program for generalized bisection," to appear.
11. R.B. Kearfott, "A summary of recent experiments to compute the topological degree," Applied Nonlinear Analysis, ed. V. Lakshmikantham, A.P., 1979.
12. R.B. Kearfott, "Some general derivative-free bifurcation techniques," submitted to the SIAM Journal on Scientific and Statistical Computing.
13. H.B. Keller, "Numerical solution of bifurcation and nonlinear eigenvalue problems." Applications of Bifurcation Theory, ed. P.H. Rabinowitz, Academic Press, New York, 1979, (pp. 359-384).
14. T.Y. Li and J.A. Yorke, "A simple reliable numerical algorithm for following homotopy paths," Technical Summary Report #1984, Mathematics Research Center, University of Wisconsin - Madison, Wisconsin 53706.

15. M. J. D. Powell, "A Fortran subroutine for solving systems of nonlinear algebraic equations," Numerical Methods for Nonlinear Algebraic Equations, ed. P. Rabinowitz, Gordon and Breach, 1970.

16. L. T. Watson, "A globally convergent algorithm for computing fixed points of C^2 maps," Appl. Math. Comput., to appear.