Note on the End Game in Homotopy Zero Curve Tracking

MARIA SOSONKINA,
LAYNE T. WATSON,
Virginia Polytechnic Institute and State University
and
DAVID E. STEWART
Texas A&M University

Homotopy algorithms to solve a nonlinear system of equations $f(x) = 0$ involve tracking the zero curve of a homotopy map $\rho(a, \lambda, x)$ from $\lambda = 0$ until $\lambda = 1$. When the algorithm nears or crosses the hyperplane $\lambda = 1$, an "end game" phase is begun to compute the solution $\bar{x}$ satisfying $\rho(a, \lambda, \bar{x}) = f(\bar{x}) = 0$. This note compares several end game strategies, including the one implemented in the normal flow code FIXPNF in the homotopy software package HOMPACK.

Categories and Subject Descriptors: G.1.5 [**Numerical Analysis**]: Roots of Nonlinear Equations – *systems of equations*; G.4 [**Mathematics of Computing**]: Mathematical Software

General Terms: Algorithms

Additional Key Words and Phrases: curve tracking, fixed point, globally convergent, homotopy methods, polynomial systems, zero

1. INTRODUCTION

Homotopy algorithms are applicable to a wide variety of zero-finding, fixed point, and optimization problems [Watson 1989]. The theoretical foundation of all probability-one globally convergent homotopy methods is given by the following definition and theorem from differential geometry.

Definition. Let $U \subset \mathbf{R}^m$ and $V \subset \mathbf{R}^p$ be open sets, and let $\rho : U \times [0, 1) \times V \to \mathbf{R}^p$ be a C^2 map. ρ is said to be transversal to zero if the Jacobian matrix $D\rho$ has full rank on $\rho^{-1}(0)$.

Theorem. If $\rho(a, \lambda, x)$ is transversal to zero, then for almost all $a \in U$ the map

$$\rho_a(\lambda, x) = \rho(a, \lambda, x)$$

is also transversal to zero; i.e., with probability one the Jacobian matrix $D\rho_a(\lambda, x)$ has full rank on $\rho_a^{-1}(0)$.

This work was supported in part by Air Force Office of Scientific Research Grant F49620-92-J-0236, Department of Energy Grant DE-FG05-88ER25068/A004, and National Aeronautics and Space Administration Grant NAG-1-1562. Authors' current addresses: M. Sosonkina, L. T. Watson, Department of Computer Science, Virginia Polytechnic Institute & State University, Blacksburg, VA 24061-0106; D. E. Stewart, Mathematics Department, Texas A&M University, College Station, TX 77843.

To solve the nonlinear system of equations

$$f(x) = 0,$$

where $f : \mathbf{R}^p \to \mathbf{R}^p$ is a C^2 map, one need only to construct a C^2 homotopy map $\rho : U \times [0,1) \times \mathbf{R}^p \to \mathbf{R}^p$ such that

1) $\rho(a, \lambda, x)$ is transversal to zero,

for each fixed $a \in U$,

2) $\rho_a(0, x) = \rho(a, 0, x) = 0$ is trivial to solve and has a unique solution x_0,
3) $\rho_a(1, x) = f(x)$,
4) $\rho_a^{-1}(0)$ is bounded.

Then for almost all $a \in U$ there exists a zero curve γ of ρ_a, along which the Jacobian matrix $D\rho_a$ has rank p, emanating from $(0, x_0)$ and reaching a zero $\bar{x}$ of f at $\lambda = 1$. This zero curve γ does not intersect itself, is disjoint from any other zeros of ρ_a, and has finite arc length in every compact subset of $[0,1) \times \mathbf{R}^p$. Furthermore, if $Df(\bar{x})$ is nonsingular, then γ has finite arc length. The general idea of the homotopy algorithm is to follow the zero curve γ emanating from $(0, x_0)$ until a zero $\bar{x}$ of $f(x)$ is reached (at $\lambda = 1$). Note that λ need not increase monotonically along γ, a feature which distinguishes homotopy from classical continuation methods.

The numerical software package HOMPACK [Watson 1987] supports three general path tracking methods: ordinary differential equation-based, normal flow, and augmented Jacobian matrix, denoted DF, NF, and QF, respectively.

As the homotopy parameter vector a varies, the corresponding homotopy zero curve γ also varies. This family of zero curves is known as the Davidenko flow. The NF algorithm is so called because the iterates converge to the zero curve γ along the normal to the Davidenko flow. The NF algorithm has four phases: prediction, correction, step size estimation, and computation of the solution at $\lambda = 1$. The fourth phase of the NF algorithm is the "end game"; this phase is crucial for the convergence of the algorithm because the predictor-corrector algorithm alone does not guarantee convergence to a zero of $\rho_a(\lambda, x)$ at which $\lambda = 1$.

The design of the end game has two principal objectives: accuracy and reliability. Speed is of less importance, since the bulk of the computation will almost always be in the curve following part of the algorithm. Since the path following need only be done to an accuracy necessary to track the curve, it is imperative that the end game accurately locate the solution. Of particular concern for robustness of the algorithm is the treatment of singular Jacobian matrices at the solution point.

This note studies four different strategies for the end game. Three new methods are introduced and compared against the one embedded in the NF algorithm. A different scheme for locating the target point at $\lambda = 1$ is proposed which robustly handles the case of a singular Jacobian matrix at the solution. The test suite includes a number of industrial problems formulated as zero-finding problems that have proven very difficult for homotopy codes [Morgan 1989]. The numerical results indicate that end games based on the different scheme outperform the one in the original NF algorithm.

2. END GAME STRATEGIES

In the normal flow algorithm, when γ is tracked past $\lambda = 1$, the algorithm enters an "end game" phase to compute the solution $(1, \bar{x})$ in the hyperplane $\lambda = 1$. The end game of the algorithm starts from two points $P^{(1)}$ and $P^{(2)}$ such that $P_1^{(1)} < 1$ and $P_1^{(2)} \geq 1$. In pseudocode, the end game is:

1. $y := P^{(2)}$; $yold := P^{(1)}$; yp, $ypold :=$ tangents at $P^{(2)}$ and $P^{(1)}$ respectively; $ansae$, $ansre :=$ absolute, relative error tolerances for the answer.
2. Do 3.– 8. some fixed number of times.
   3. Find $\bar{s}$ such that $p_1(\bar{s}) = 1$, where $p(s)$ is the Hermite cubic interpolating polynomial generated using $yold$, y, $ypold$, and yp.
   4. Do two iterations of

$$Z^{(k+1)} := Z^{(k)} - \left[D\rho_a(Z^{(k)})\right]^\dagger \rho_a(Z^{(k)}), \quad k = 0, 1,$$

      staring with $Z^{(0)} = p(\bar{s})$, ending with $Z^{(2)}$. $[D\rho_a]^\dagger$ is the Moore-Penrose pseudoinverse of $D\rho_a$.
   5. If

$$\left|Z_1^{(2)} - 1\right| + \left\|\Delta Z^{(1)}\right\| \leq ansae + ansre \left\|Z^{(1)}\right\|,$$

      then return (solution has been found).
   6. If $Z_1^{(2)} \geq 1$, then
      7. $y := Z^{(2)}$; $yp :=$ tangent at $Z^{(2)}$;

      else
      8. $yold := Z^{(2)}$; $ypold :=$ tangent at $Z^{(2)}$.
9. Return with an error flag.

The augmented Jacobian matrix algorithm, another curve tracking method from HOMPACK, approaches the end game differently. The normal flow algorithm can be modified to incorporate that approach as follows.

The end game begins when the first point $P^{(2)}$ is generated such that $P_1^{(2)} \geq 1$. Thus initially, the solution at $\lambda = 1$ lies somewhere between the previous point $P^{(1)}$ and $P^{(2)}$, i.e., $P^{(1)}$ and $P^{(2)}$ bracket the solution. When the end game is first entered, $P^{(1)}$ and $P^{(2)}$ bracket $\lambda = 1$, and the first predicted point $Z^{(0)}$ is computed as in the original normal flow algorithm: using the Hermite cubic interpolation and root finding. In each subsequent step, the prediction $Z^{(k-2)}$ is generated by the secant method, namely

$$Z^{(k-2)} = P^{(k)} + \left(P^{(k-1)} - P^{(k)}\right) \frac{\left(1 - P_1^{(k)}\right)}{\left(P_1^{(k-1)} - P_1^{(k)}\right)}.$$

This equation results in a disastrous prediction when $\left|P_1^{(k-1)} - P_1^{(k)}\right| \ll \left|1 - P_1^{(k)}\right|$, and can also simply generate divergent approximations. To stabilize the iteration, the last point $P^{(opp)}$ on the opposite side of the hyperplane $\lambda = 1$ from $P^{(k)}$ must always be saved to bracket the solution.

In the case when $\left\|Z^{(k-2)} - P^{(k)}\right\| > \left\|P^{(k)} - P^{(opp)}\right\|$, the chord method

$$Z^{(k-2)} = P^{(k)} + \left(P^{(opp)} - P^{(k)}\right) \frac{\left(1 - P_1^{(k)}\right)}{\left(P_1^{(opp)} - P_1^{(k)}\right)}$$

is substituted for the secant method. Since the chord method has a slower rate of convergence, its application is limited to this special case.

The correction is implemented as a single Newton step

$$P^{(k+1)} = Z^{(k-2)} + \Delta Z^{(k-2)},$$

where $\Delta Z^{(k-2)}$ is the minimum norm solution to

$$\left[D\rho_a(Z^{(k-2)})\right] \Delta Z^{(k-2)} = -\rho_a(Z^{(k-2)}).$$

$P^{(opp)}$ is updated appropriately. This two step process (linear interpolation followed by a Newton correction) is repeated until the solution at $\lambda = 1$ is found, or a limit on the number of iterations is exceeded.

Another implementation of the end game adds one Newton iteration to the correction stage of the algorithm described above. The fourth version considered computes the first predicted point $Z^{(0)}$ as in the others. However, for the rest of the points, a projected Newton iteration is performed (pure Newton step followed by a projection along the tangent direction onto the hyperplane $\lambda = 1$): with

$$W = Z^{(k)} - \left[D\rho_a(Z^{(k)})\right]^{\dagger} \rho_a(Z^{(k)}),$$

and T the unit tangent vector satisfying

$$\left[D\rho_a(Z^{(k)})\right] T = 0, \quad \|T\|_2 = 1,$$

the new point $Z^{(k+1)}$ is given by

$$Z^{(k+1)} = W + \frac{1 - W_1}{T_1} T.$$

Note that if the solution $\bar{Z} = (1, \bar{x})$ occurs at a turning point of γ, so that $D_x\rho_a(\bar{Z})$ is singular and $T_1 = d\lambda/ds = 0$ at $\bar{Z}$, this division by T_1 is ill advised.

Several other projection alternatives were considered: an orthogonal projection of W into the hyperplane $\lambda = 1$, and a scaled projection where W is divided by its first component to get the new point $Z^{(k+1)}$ in the hyperplane $\lambda = 1$. Comparing the performance of the end game for any of the three projections to the other end game versions considered, no significant difference between the alternative projections is observed. Thus, the orthogonal and scaled projections are excluded from further discussion.

3. NUMERICAL RESULTS

For convenient reference, the following short names are assigned to the versions of the end game being tested:

E — original end game;
EL1N — end game modified with a secant predictor and one Newton step in the correction stage;
EL2N — end game modified with a secant predictor and two Newton steps in the correction stage;
EPN — end game modified to use projected Newton iterations.

The experiments involved 14 polynomial systems [Morgan 1989] modeling real world problems. To reach their solutions, the normal flow algorithm was required to track $204,622$ paths (many of which are extremely difficult, involving highly rank deficient Jacobian matrices at the solution). Along with successful finishes, failures were encountered due to detected (exact) singularity of the Jacobian matrix, or due to the nonconvergence of the end game. Only the latter aspect was monitored in the experiments. Precisely, all the paths where the iteration failed to converge for at least one of the tested end games, but not for all of them, were extracted for analysis. There were only 59 such paths. The total number of Jacobian matrix evaluations an algorithm required while tracking a curve is the basis for performance analysis. A two-stage ranking scheme is used to present the data.

Preprocessing. The ranking involves only the 59 mentioned cases. Thus, for each of them, there is at least one end game that failed to converge. To account for the failure, the algorithm is penalized by weighting the total number of Jacobian matrix evaluations by a positive number greater than one (say ten). On the other hand, for a successfully finished end game along a given path, the total number of Jacobian matrix evaluations for the corresponding algorithm is weighted by one (no penalty assigned).

Rank assignment. Given a path, the numbers obtained from the preprocessing stage are sorted in ascending order, and each algorithm is assigned a rank equal to the place its weighted number holds in the sequence; smaller rank indicates better performance. The weight assignment defined in the preprocessing stage guarantees that all four algorithms cannot be ranked equally for a given path.

The reason for comparing failed curve tracking attempts, rather than successful tracks, is that when all four methods succeed there is little difference between them. Furthermore, unless a successful method is egregiously worse than another method, there is little justification for abandoning the successful method. The comparison of the four end game versions is made by summing the ranks over all 59 paths (see Table 1).

4. DISCUSSION AND CONCLUSIONS

Table 1 demonstrates rather different performance of all four end games. The reason for this is the predicted point computation since the Newton iteration is common to the correction stage of all the end games. EL1N and EL2N noticeably outperform E and EPN. The computational advantage of EL1N and EL2N over E is attributable to the secant method instead of the chord method (in E), which has linear convergence. In EL1N and EL2N, the secant method is used for every predicted point computation except when (chord) bracketing is necessary for stability. EPN has better performance than E, because the projected Newton iterates converge quadratically, while the chord method in E is converging strictly linearly when one of the points bracketing the solution at $\lambda = 1$ remains fixed over several iterations. A closer look at EL1N and EL2N reveals that one Newton step in the correction stage of the end game is better than two (or more) Newton steps. The second Newton iteration in EL2N is apparently wasted, generating a point closer to the zero curve γ but *not* closer to the solution $(1, \bar{x})$ in the hyperplane $\lambda = 1$. In conclusion, the experiments show that the end game with a secant predictor and one Newton step in the correction stage is a clear winner among the end games tested. Thus, EL1N should be used as the final stage of the normal flow algorithm FIXPNF in HOMPACK.

REFERENCES

MORGAN, A.P., AND WATSON, L.T. 1989. A globally convergent parallel algorithm for zeros of polynomial systems. *Nonlinear Anal. 13*, 1339–1350.

WATSON, L.T. 1989. Globally convergent homotopy methods: a tutorial. *Appl. Math. Comput. 31BK*, 369–396.

WATSON, L.T., BILLUPS, S.C., AND MORGAN, A.P. 1987. HOMPACK: A suite of codes for globally convergent homotopy algorithms. *ACM Trans. Math. Softw. 13*, 3 (Sept.), 281–310.

Table 1. Rank assignment for the end games tested on the distinguished 59 paths.

Problem	E	EL1N	EL2N	EPN	Problem	E	EL1N	EL2N	EPN
pb0102	2	2	4	1	pb0103	3	4	2	1
pb0103	2	4	3	1	pb0103	2	4	3	1
pb0601	4	1	2	3	pb0601	4	1	2	3
pb0601	4	1	2	3	pb0601	4	1	2	3
pb0601	3	1	2	4	pb0601	4	1	2	3
pb0601	4	1	2	3	pb0601	4	1	2	3
pb0601	1	2	3	4	pb0601	4	1	2	3
pb0601	4	1	1	3	pb0601	4	1	2	3
pb0601	4	1	2	3	pb0601	4	1	2	3
pb0601	4	1	1	3	pb0601	4	1	2	3
pb0601	4	3	1	2	pb0601	4	1	2	3
pb0601	3	1	2	4	pb0602	4	1	1	3
pb0602	4	1	1	3	pb0602	4	1	2	3
pb0602	4	2	3	1	pb0602	4	1	1	3
pb0602	2	3	1	4	pb0602	4	3	2	1
pb0602	4	1	1	3	pb0603	2	1	3	4
pb0603	4	1	2	2	pb0603	1	2	3	4
pb1704	4	1	3	1	pb5001	3	1	1	4
pb5001	3	1	4	2	pb5001	4	1	2	3
pb5001	3	2	4	1	pb5001	4	1	2	3
pb5001	4	1	2	3	pb5001	4	1	1	3
pb5001	4	3	1	2	pb5001	4	1	2	3
pb5001	4	1	2	3	pb5001	1	3	4	2
pb5001	3	1	4	2	pb5001	4	1	2	3
pb5001	4	1	2	3	pb5001	4	1	2	3
pb5001	4	1	2	3	pb5001	3	1	2	4
pb5001	3	1	2	4	pb5001	4	3	1	2
pb5001	3	2	4	1	pb5001	1	3	4	2
pb5001	4	1	2	3	pb5001	4	1	2	3
pb5001	4	1	2	3	Sum:	204	88	127	160