Stepsize control for adaptive multiprecision path tracking

Daniel J. Bates* Jonathan D. Hauenstein †
Andrew J. Sommese ‡ Charles W. Wampler II §

Abstract

Numerical difficulties encountered when following paths using methods such as homotopy continuation may be overcome by combining adaptive stepsize and adaptive multiprecision. In the paper Adaptive multiprecision path tracking [1], precision and stepsize are adapted separately. This can lead to suboptimal performance and even failure in certain circumstances. This paper presents a strategy for adjusting precision and stepsize together to eliminate path failures while minimizing the computational effort expended per unit advance along the path.

This paper concerns path tracking algorithms for tracing out a one dimensional path defined implicitly by n equations in $n+1$ unknowns. In particular, we consider such algorithms when multiprecision calculations are available, that is, when the precision of the computations can be changed during the computation. We treat a common type of path tracker that uses an Euler predictor to step ahead along the tangent to the path and a Newton corrector to bring the predicted point closer to the path. Our objective is to adjust precision and stepsize together to minimize the computational cost of tracking the path while maintaining high reliability.

*Department of Mathematics, Colorado State University, 101 Weber Building, Fort Collins, CO 80528 (bates@math.colostate.edu, http://www.math.colostate.edu/∼bates). This author was supported by Colorado State University and the Institute for Mathematics and Its Applications (IMA).

†Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556 (jhauenst@nd.edu, http://www.nd.edu/∼jhauenst). This author was supported by the Duncan Chair of the University of Notre Dame; the University of Notre Dame Center for Applied Mathematics; and NSF grants DMS-0410047 and NSF DMS-0712910.

‡Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556 (sommese@nd.edu, http://www.nd.edu/∼sommese). This author was supported by the Duncan Chair of the University of Notre Dame; and NSF grants DMS-0410047 and NSF DMS-0712910.

§General Motors Research and Development, Mail Code 480-106-359, 30500 Mound Road, Warren, MI 48090 (Charles.W.Wampler@gm.com, www.nd.edu/∼cwample1) This author was supported by NSF grant DMS-0410047 and NSF DMS-0712910.

In fixed precision tracking, a trial-and-error approach to setting the stepsize is effective: shorten the step upon failure, and lengthen it upon repeated successes. If the level of precision is not adequate, the step may fail no matter how small the step is made, so the trial-and-error approach repeatedly shortens the stepsize until failure is declared due to lack of progress.

In the multiprecision setting, one has the flexibility of either changing the stepsize or changing the precision. In [1], the precision is set first, according to rules designed so that corrector steps computed by Newton's method have enough digits of accuracy to ensure convergence, assuming the initial guess is within the convergence zone. If the initial guess is not good enough, the corrector fails, and the algorithm responds by shortening the stepsize to try again. For a small enough step and a high enough precision, the prediction/correction cycle must succeed and the tracker advances along the path. One would hope that the only mode of failure is when the combination of high precision and small steps is so severe that one gives up due to the excessive burden on computational resources. However, in testing that algorithm, another mode of failure was discovered: for too large a stepsize, the predicted point can be far enough from the path that the rules set the precision too high that the algorithm fails before a decrease in stepsize is considered. In particular, this was observed in tracking paths defined by polynomials of high degree and occurred on the first step when the initial stepsize given by the user was too large.

One might fix this problem by trapping the precision overflow condition and responding with a decrease in the stepsize. While such an approach may work, we present a more effective alternative here.

Success of a step depends on having sufficient precision and a small enough stepsize, but increasing precision and decreasing the stepsize both inflate the computational cost. With exact arithmetic, the stepsize is limited by the requirement for the predicted point to stay within the convergence zone of the corrector. For each stepsize below this limit, there is some minimum precision necessary to converge within the allowed number of correction steps. This necessary level of precision monotonically decreases with stepsize, approaching in the limit the precision that just barely ensures that the final error given by Newton's method equals the desired accuracy. Somewhere between these two limits, there must be an optimal combination of stepsize and precision that minimizes the computational effort per unit advance along the path. However, spending too much computation to find this optimum is itself counterproductive. Accordingly, in this paper, we develop some approximate rules for finding a near optimum. At the same time, we eliminate the mode of failure previously mentioned. These new rules have been implemented in our software package, Bertini [2].

1 The Main Idea

As in [1], the Euler predictor and the Newton corrector can be summed up in a single relation, obtained by retaining only the linear terms in a Taylor series expansion about (z_i, t_i):

$$H_z(z_i, t_i)\Delta z = -\left(H(z_i, t_i) + H_t(z_i, t_i)\Delta t\right), \tag{1}$$

$$(z_{i+1}, t_{i+1}) = (z_i, t_i) + (\Delta z, \Delta t). \tag{2}$$

where $H_z = \frac{\partial H}{\partial z}$ and $H_t = \frac{\partial H}{\partial t}$. This formula gives the Euler prediction along the tangent when (z_i, t_i) is on the path, i.e., when $H(z_i, t_i) = 0$. After predicting with a given Δt, we may set $\Delta t = 0$, upon which Eq. 1 becomes Newton's method.

We may view this another way. Let the stepsize be s, let (z_0, t_0) be the current point approximately on the path, and let $T = t_0 + s$ be the target for t for the next point on the path. Then the desired next point is the solution of the augmented system

$$f(z,t) = \begin{pmatrix} H(z,t) \\ t - T \end{pmatrix} = 0. \tag{3}$$

Applying Newton's method to $f(z,t)$, we produce a new guess (z_{i+1}, t_{i+1}) by solving

$$\begin{bmatrix} H_z(z_i, t_i) & H_t(z_i, t_i) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta t \end{bmatrix} = -\begin{bmatrix} H(z_i, t_i) \\ t - T \end{bmatrix}, \tag{4}$$

$$(z_{i+1}, t_{i+1}) = (z_i, t_i) + (\Delta z, \Delta t). \tag{5}$$

It is easy to confirm that the sequence of iterates produced in this way are exactly the same as before: the first iterate is just the Euler prediction and subsequent ones are Newton corrections at $t = T$.

This observation is useful, because the analysis of Newton's method in [1] now applies to the prediction step whereas it applies only to the corrector steps in that paper. The only difference is that the Jacobian matrix, J, appearing in those derivations is now the augmented Jacobian appearing on the left side of Eq. 4.

2 New Rules for Stepsize

The method of [1] consists of several rules, each of which indicate when precision should be changed. Recall that rule B of [1] for setting the precision P is

$$P > \sigma_1 + \log_{10}\left[\|J^{-1}\|\left((2+\mathcal{E})\|J\| + \mathcal{E}\Phi\right) + 1\right] + (\tau + \log_{10}\|d\|)/(N - i). \tag{\textbf{B}}$$

For brevity, let $D = \log_{10}\left[\|J^{-1}\|(2+\mathcal{E})(\|J\| + \mathcal{E}\Phi) + 1\right]$, where J is the Jacobian matrix, $\mathcal{E}$ is the error in adding the differential d to the current approximation of the solution, and Φu is a bound on the error when evaluating J. Then we may write

$$P > \sigma_1 + D + (\tau + \log_{10} \|d\|)/(N - i). \tag{7}$$

Roughly speaking, D is the number of digits lost to numerical error in computing corrections. The first term, σ_1 is the number of safety digits, and $(\tau + \log_{10} \|d\|)/(N - i)$ is the number of additional digits we need to correct in our approximate solution so as to meet the final tolerance $10^{-\tau}$ within $(N - i)$ remaining iterations. In [1], this rule only applies to the corrector steps, not the predictor step.

We may apply this rule to the augmented system, where the Euler prediction step is now just the initial Newton iteration. We allow the number of iterations, N, to be one greater than before, since it now includes the prediction step. However, we have a new parameter that we can vary besides the precision P, namely, the stepsize s. In Eq. 7, $\|d\| = \|(\Delta z \ \Delta t)\|$, which on the first iteration is directly proportional to s, say $\|d\| = a\,|s|$. Accordingly, we may rewrite the equation for the first step, $i = 0$, as

$$P - \log_{10} |s|/N > \sigma_1 + D + (\tau + \log_{10} a)/N, \tag{8}$$

or, letting $|s| = 10^{-\xi}$, as

$$P + \xi/N > \sigma_1 + D + (\tau + \log_{10} a)/N. \tag{9}$$

Thus, there are two ways to satisfy this inequality: raise precision P or decrease the stepsize by raising ξ.

Suppose that $C(P)$ is the cost of computing N iterations in precision P. Then the cost per unit advance along the path is $C(P)/|s|$. We wish to minimize $C(P)/|s|$ subject to Eq. 8. Since Bertini uses MPFR for multiprecision arithmetic, precision is available at discrete packets of 32 bits. Thus it is straightforward to step using the current level of P, evaluate $C(P)$, set $|s|$ to just satisfy Eq. 8, and thereby identify the minimum of $C(P)/|s|$.

We must remember that this rule assumes that we are within the convergence zone of Newton's method. If not, additional precision will not be effective and we must cut the stepsize. Therefore, we must retain the previous algorithmic step of cutting the stepsize when convergence is not obtained within N iterations.

Remark 1: An alternative is to compute another bound that guarantees that we stay within the convergence zone. From the analysis in [1], we have the new distance from a solution $\bar{\Delta}$ is bounded in terms of the previous distance Δ as

$$\begin{aligned}\bar{\Delta} \leq & K\|J^{-1}\|(1+u)^2(\alpha+\beta)\Delta^2 + \\ & \big(K\|J^{-1}\|[(2+\mathcal{E}+u)\,(u\|J\|+\phi)]+u\big)\,\Delta + K\|J^{-1}\|(1+u)\psi + u\|v_*\|.\end{aligned} \tag{10}$$

Here α and β are constants bounding the error in the linear approximations to the function whose zero we are seeking. They can be approximated by finding the second derivatives of the function. All the other quantities are at least approximately known as functions of the precision $u = 10^{-P}$, so that we may rewrite (10) as

$$\bar{\Delta} \leq e(u)\Delta^2 + f(u)\Delta + g(u), \tag{11}$$

with functions $e(u), f(u), g(u)$ known. We want $\bar{\Delta} < \Delta$ for convergence. As above, the first step has magnitude $\Delta = a|s|$. Moreover, for small u, $e(u)$ is approximately constant, say $e(u) \approx \hat{e}$ and $f(u)$ and $g(u)$ are proportional to u, say $f(u) \approx \hat{f}u, g(u) \approx \hat{g}u$. Consequently, if we have

$$\hat{e}\Delta^2 + \hat{f}u\Delta + \hat{g}u < a, \tag{12}$$

then convergence is assured. For small enough u and Δ, this inequality can always be satisfied. If we add this constraint to the minimization of $C(P)/|s|$, we have a minimization problem whose solution guarantees progress at approximately minimum cost.

The constants α and β may be hard to evaluate for a multivariate system. However, when the original problem to be solved involves just one variable, $z \in \mathbb{C}$, these could be computed. However, since the idea seems unattractive for multivariate systems, our main interest, we have not implemented it in Bertini.

Remark 2: The analysis given in Section 2 can be extended to all Runge-Kutta methods using a block diagonal matrix in which each diagonal block corresponds to a function evaluation for the Runge-Kutta method. Similar to the case of Euler prediction as treated above, when Newton's method is applied to the extended system, the first iteration produces the Runge-Kutta prediction step and subsequent Newton iterations on the extended system give the same sequence of corrections as Newton's method applied to the original system. In this way, step length optimization can be extended to Runge-Kutta prediction followed by Newton correction. Other higher-order predictors could be accommodated in a similar manner.

3 Implementation Details and Computational Experiments

Adaptive multiprecision tracking with stepsize control is implemented in the software package Bertini [2]. All the examples discussed here were run on an Opteron 250 processor running 64-bit Linux.

3.1 Implementation Details

In the examples below, the number of safety digits was set to 1. To avoid constantly changing precision and stepsize upon success, we required 5 successful steps before attempting to increase the stepsize and 10 successful steps before attempting to decrease the precision. Failures always cause the stepsize and precision to be changed according to the rules provided.

minimum fixed precision		method of [1]	new method
96 bits	185.94	39.84	34.65

Table 1: Comparison for average time of 10 runs of the IPP system, in seconds.

To obtain a minimization of cost per unit advance, $C(P)/|s|$, an approximation of $C(P)$ was found using an average cost of computation in MPFR with different precisions compared with IEEE double precision. At various precisions, we computed the time of common operations used in homotopy continuation, e.g. straight-line program evaluation, matrix multiplication and linear solving. Based on this data, the cost function that we used in the following examples, with P in bits of precision, was

$$C(P) = \begin{cases} 1, & \text{if } P \text{ corresponds to double precision;} \\ 10.35 + 0.04P, & \text{otherwise.} \end{cases}$$

As new versions of MPFR are released, this cost function will be recomputed.

3.2 Comparing the methods

In Section 5.5 of [1], a polynomial system arising from the inverse kinematics problem for a general six-revolute serial-link robot [3] is considered. Utilizing the power series endgame with the same settings as in [1], Table 1 indicates the average time required to solve that system with fixed precision, the method of [1], and the new method of this paper.

The proactive method described in this paper causes paths to be tracked using double precision longer than the method of [1]. The key difference is that this new method allows the stepsize to be decreased rather than automatically relying on the power (and cost) of higher precision when numerical difficulties are encountered. This resulted in a 13% improvement in speed for this example since double precision computation is so much less costly than multiple precision computation.

3.3 Near singular conditions

For the homotopies utilized, with probability one, the paths do not pass directly through a singularity on $(0, 1]$. Even though the Jacobian is still nonsingular, higher precision may be needed to reveal this. It is not known, *a priori*, how many paths travel near a singularity for a given homotopy.

To demonstrate that near singularity conditions do exist, consider the formulation of the nine-point path synthesis problem for four-bar linkages in [4]. Utilizing a 2-homogeneous structure and the two-fold symmetry, the homotopy consists of $143,360$ paths of which 4326 lead to nondegenerate solutions. The precision points were selected at random and the homotopy was created using random numbers. During the tracking, 1184 of the total $143,360$ paths (0.83%)

needed to use precision higher than double to track past a near singularity before returning back to double precision. Moreover, 680 paths (0.47% of the total) needed to use at least 96 bits of precision to track past a near singularity before returning to double precision.

Figure 1 is a graph of the log of the condition number, precision and stepsize in relation to tracking parameter t for a typical path having a near singularity and requiring the use of at least 96 bits of precision before returning to double precision.

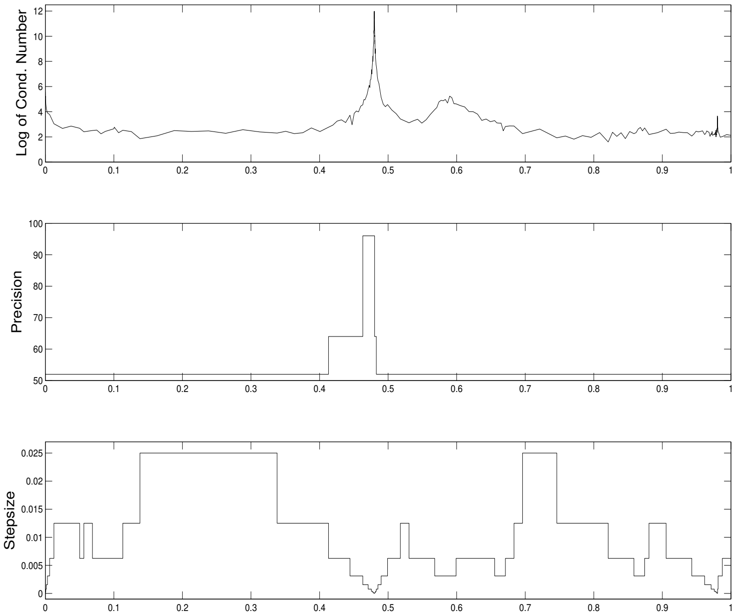

Figure 1: Graph of the log of the condition number, precision and stepsize against the tracking parameter t

References

[1] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, Adpative multiprecision path tracking, SIAM Journal on Numerical Analysis, 46(2): 722-746, 2008.

[2] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler, Bertini: Software for Numerical Algebraic Geometry. Available at www.nd.edu/∼sommese/bertini.

[3] A.P. Morgan and A.J. Sommese. Computing all solutions to polynomial systems using homotopy continuation. *Appl. Math. Comput.*, 24(2):115–138, 1987. Errata: *Appl. Math. Comput.*, 51:209, 1992.

[4] C.W. Wampler, A. Morgan, and A.J. Sommese, Complete solution of the nine-point path synthesis problem for four-bar linkages, *ASME Journal of Mechanical Design* 114(1):153–159, 1992.