Experimental Results: An Interval Step Control for Continuation Methods

by

R. Baker Kearfott

Department of Mathematics
University of Southwestern Louisiana
U.S.L. Box 4-1010
Lafayette, LA 70504-1010 U.S.A.

Abstract

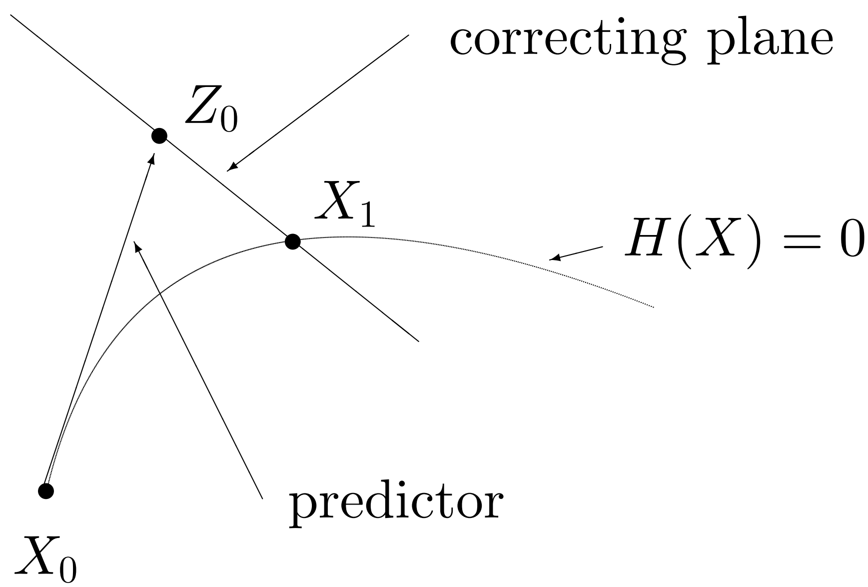
In predictor / corrector continuation methods, a step control adjusts the size of each predictor step. In traditional step controls, a heuristic is used to try to prevent these iterates from jumping to separate branches. Such heuristics are not foolproof.

In interval step controls, the step control can be made *rigorous* in the sense that it is mathematically impossible for the iterates to jump across paths or for bifurcation points to remain undetected, even in finite precision arithmetic. Recent experiments illustrate that such step controls are effective and, in some instances practical. Recent programming language and software developments make interval step control technology more widely accessible.

Outline of Talk

1. Predictor–Corrector Methods
2. Traditional Step Controls
3. General Properties of Interval Newton Methods
4. Parametrized Interval Newton Methods and Theory
5. The step control
6. Examples of Comparison
7. Speed of Interval Arithmetic
8. Availability of Interval Arithmetic Software

General Predictor–Corrector Methods



Heuristic Step Controls

The predictor step length δ is adjusted (say doubled or halved) to control

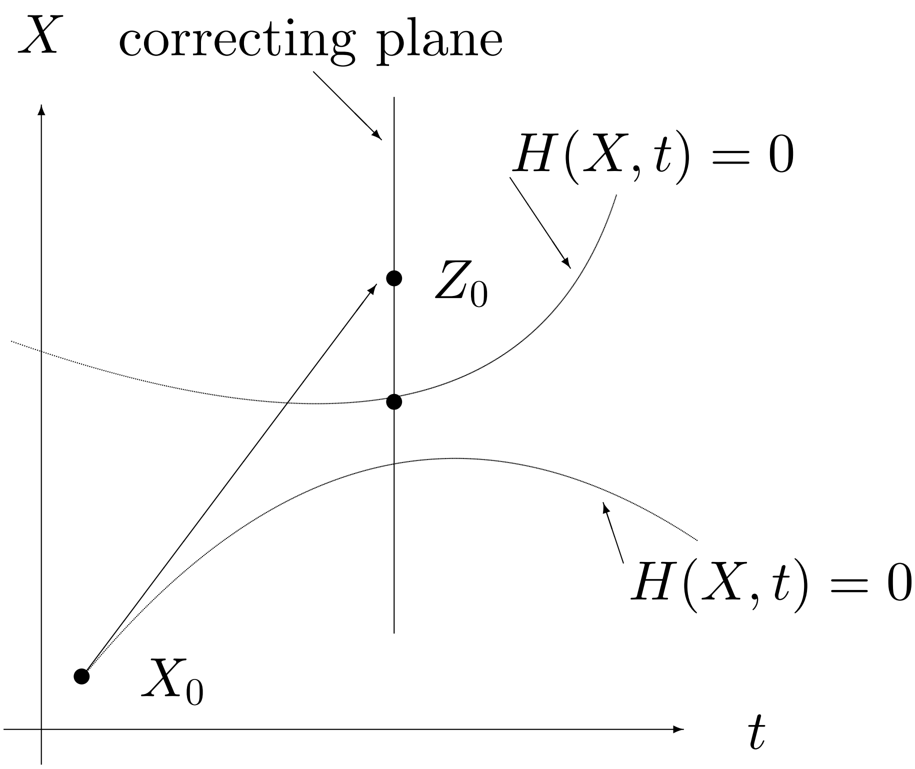
1. the residual or
2. the number of steps of corrector iteration, etc.

In [den Heijer and Reinboldt, *SIAM J. Numer. Anal.* **18**, 5 (1981), pp. 925–947], δ is adjusted according to an estimate for the radius of convergence of corrector iteration.

Den Heijer and Rheinboldt observe that it is impossible to have an infallible step control that is based on information only at a finite number of points. However, *interval step controls* implicitly use global information.

Illustration of Failure

This actually happens frequently!



Interval Newton Methods

Computational Fixed Point Theorems

Interval Newton methods are based on *Brouwer's Fixed Point Theorem* or a variant of it, *Miranda's Theorem*. They are operators from interval vectors (boxes) $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ into themselves:

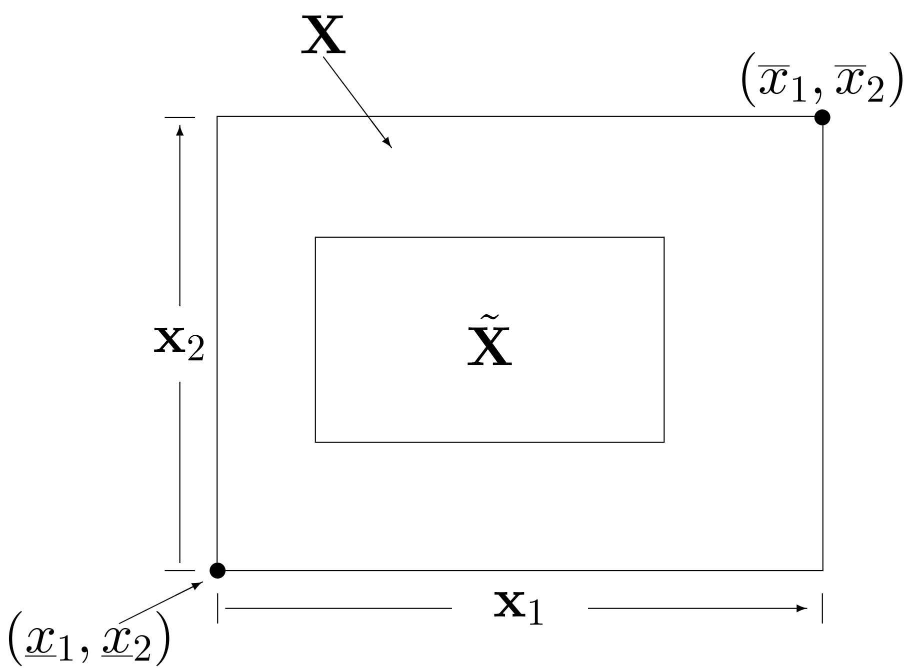
$$\tilde{\mathbf{X}} = \mathbf{N}(F; \mathbf{X}, \check{X})$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$, $\check{X} \in \mathbb{R}^n$ is a base point, and $\mathbf{N}(F; \mathbf{X}, \check{X})$ is computed via interval evaluations of the Jacobi matrix and interval numerical linear algebra. (Details can be found in various places.)

If $\mathbf{N}(F; \mathbf{X}, \check{X}) \subset \mathbf{X}$ then there exists a solution of $F(X) = 0$ within $\mathbf{X}$. For certain specific types of operators $\mathbf{N}(F; \mathbf{X}, \check{X})$, it can also be concluded that this solution is unique. (For example, see my upcoming book, or other references.)

Interval Newton Methods

Illustration



In this case, an interval Newton method proves existence.

Parametrized Interval Newton Methods

In a parametrized interval Newton method, an interval Newton method is applied to $H : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ by singling out a coordinate t. Then, with appropriate interval extensions to the Jacobi matrix,

$$\mathbf{N}(H(\cdot,\mathbf{t});\mathbf{X},\check{X}) \subset H(\mathbf{X},\mathbf{t})$$

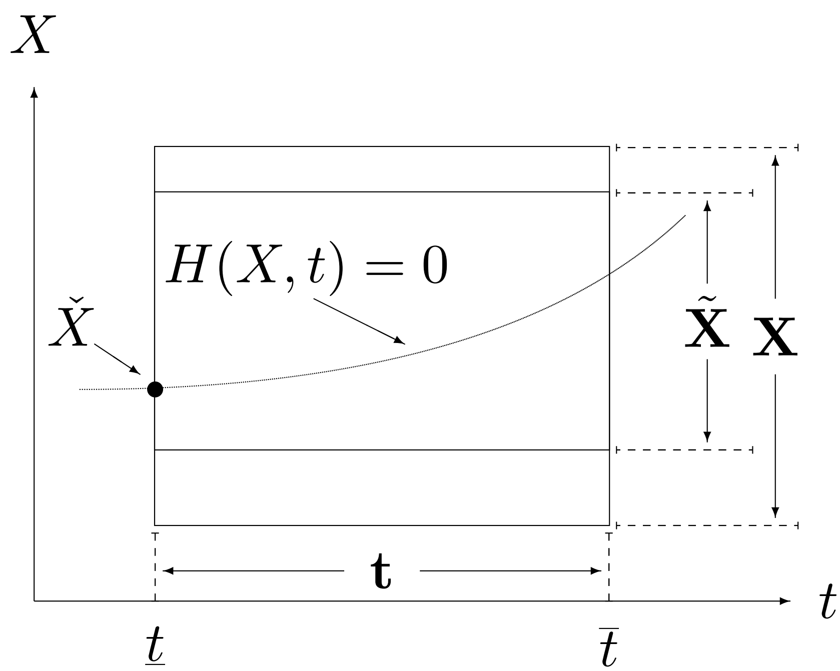
implies that, for every $t \in \mathbf{t}$, there is a unique solution of $H(X,t) = 0$.

Furthermore, we have proven that there is a unique path passing through the faces $(\mathbf{X},\underline{t})$ and $(\mathbf{X},\overline{t})$ of the box $(\mathbf{X},\mathbf{t}) \subset \mathbb{R}^{n+1}$ [Theorem 3.1, SIAM J. Numer. Anal. **31**, 3 (1994), pp. 892–914].

Because of this uniqueness, the iterates *cannot* jump across branches or bifurcation points with an interval step control.

Parametrized Interval Newton Methods

Illustration



In this situation, the computations *prove* that there is a unique path in the box, passing through the faces $t = \underline{t}$ and $t = \overline{t}$.

Interval Step Controls

Interval step controls involve adjusting the size of $\mathbf{X}$ and choosing the index and width of the parameter coordinate $\mathbf{t}$ to ensure

$$\mathbf{N}(H(\cdot, \mathbf{t}); \mathbf{X}, \check{X}) \subset \mathbf{X}$$

A specific step control is given by Zhaoyun Xing and me in [Theorem 3.1, SIAM J. Numer. Anal. **31**, 3 (1994), pp. 892–914]. We compare its performance to PITCON on some problems with rapidly changing curvature.

Layne Watson's Exponential Cosine $- n = 5$

With different heuristic parameter choices in PITCON, behavior was erratic and unpredictable. In the interval step control, heuristic parameters affected only efficiency, not the curve obtained.

Topologist's Sine Curve

$(x, \sin(1/x))$

Dimension-Dependence

A Discretized Eigenvalue Problem

The initial-boundary problem:

$$\begin{cases} y'' + \lambda e^y = 0 \\ y(0) = 0, \quad y'(1) = 0 \end{cases}$$

is discretized with central differences into $H(X) = 0$, where

$$H_i(X) = \begin{cases} x_1, & i = 1, \\ x_{i-1} - 2x_i + x_{i+1} + x_N e^{x_i} * d, & i = 2, \ldots N-2, \\ x_{N-1} - x_{N-2}, & i = N-1, \end{cases}$$

where $d = 1/(N-2)^2$ and $x_N = \lambda$. To include portions with changing curvature, the curve is followed from $\lambda = 0$ and $x_i = 0$, $i = 1, ..., N-1$, past a turning point of x_N with respect to x_{N-1}. A plot of the curve is in our paper.

Discretized Eigenvalue Problem

Interval Step Control

These results were with ACRITH-XSC on an IBM 3090.

N	steps	ave. δ	CPU(s)	CPU r.	N^3 r.
10	252	0.019910	27.35		
20	315	0.019921	198.00	7.24	8
30	371	0.019910	732.19	3.70	3.37
40	420	0.019950	1968.17	2.69	2.37
50	464	0.019954	4425.24	2.25	1.95
60	504	0.019957	8510.62	1.92	1.73

Average δ and number of steps were very similar with PITCON. Other comparisons to PITCON, including performance near bifurcation points, also appear in the paper.

Approximate Bifurcation Point

A Parametrized Family of Hyperbolas

$$H(x,t) = x^2 - (t-0.5)^2 - p^2$$

This picture was generated with $p = 10^{-15}$. PITCON jumped across to the other branch. The step sizes varied from appropriately large to appropriately small; a table is in the paper.

On the General Speed of Interval Arithmetic

Task	`ACRITH`	`INTARITH`
$\sum_{i=1}^{10^6}$ `1.0D0`	39.7	19.7
$\prod_{i=1}^{10^6}$ `1.0D0`	34.6	24.1
Compute sin(1) 10^6 times	17.7	104.9
Compute 1^2 10^6 times		9.6

Ratios of of interval to floating point CPU times for ACRITH-XSC on an IBM 3090 and for `INTLIB_ARITHMETIC` on a Sun Sparc 20 model 51

Interval Computations Language Support

- Software developed under Prof. Dr. Kulisch's direction (the "SC" languages and the "XSC" languages) will be described in the afternoon sessions next week.
- I have portable software.
  - INTLIB — A TOMS Algorithm, in standard `FORTRAN-77`.
  - Fortran 90 modules — an interval data type, automatic differentiation, support for interval Newton methods.
  - Reprints are available.
  - Browse the URL:

`ftp://interval.usl.edu/pub/interval_math/www/kearfott.html`