AN EFFICIENT INTERVAL STEP CONTROL FOR TWO DIMENSIONAL CONTINUATION METHODS

R. B. KEARFOTT AND ZHAOYUN XING

Department of Mathematics
University of Southwestern Louisiana

ABSTRACT. We present a step control for continuation methods for curves in two dimensions that is deterministic in the sense that (i) it computationally but rigorously verifies that the corrector iteration will converge to a point on the same curve as the previous point (i.e. the predictor / corrector iteration will never jump across paths), and (ii) each predictor step is as large as possible, subject to verification that the curve is unique with the given interval extension. We use interval analysis techniques that, particular to two dimensions, result in a more efficient step control than a general technique we previously developed. We present performance data and comparisons with a non interval step control method (PITCON version 6.1). Tests using the Topologist's sine curve, which has changes in curvature of increasing magnitude, suggest that this interval step control is very efficient when rapid changes in curvature occur, and that it is much faster than the non interval step control for such functions, assuming both step controls follow the curve successfully. Additionally, comparison of plots obtained from both step controls reveals that a non interval step control will behave erratically in situations where the interval step control leads to orderly progression along the curve.

1. INTRODUCTION

We assume a knowledge of continuation methods, as can be found in [2]. We also assume an elementary knowledge of interval arithmetic, as appears in [1], [5], or [7]. The reference [7] is particularly appropriate here.

Throughout this paper, we use boldface lowercase letters (such as $\mathbf{x}$) for interval vectors, except $\mathbf{H}$ stands for the interval function extension. We use uppercase letters (such as X) for scalar vectors, and lowercase letters with subscripts (such as x_i) for scalar variables.

1991 *Mathematics Subject Classification.* 65H20.

Key words and phrases. automatic verification,interval evaluation, uniqueness, continuation methods, step control algorithms, PITCON.

In this paper, we concentrate on following problem:

> Compute a sequence of points on the solution curve
>
> $$\mathcal{Z} = \left\{ Y \in \mathbb{R}^2 | H(Y) = 0 \right\}, \text{ where } H : \mathbb{R}^2 \to \mathbb{R}^1,$$
>
> with a guarantee that all the points are on the same continuous path.

To date, most continuation methods appearing in the literature, such as [3], are based on non interval step control. Non interval step control methods are successful and fast when the curve is smooth and isolated, but problems arise when there are many paths near some points. In that case, algorithms based on non interval step controls may jump from one path to another, as the numerical results in §4 show. Also, if rapid changes in curvature occur along the path, a method based on non interval step control sometimes even erroneously reverses direction. However, appropriate use of interval analysis gives us a guarantee that the predictor algorithm will not jump from one path to another, or, indeed, jump over different legs of the same path. Also, we have observed the special two-dimensional interval step control proposed in this paper to be faster at following curves with many close branches or with rapid changes in curvature, assuming the heuristic parameters governing the non interval step control are set so it successfully tracks the curve.

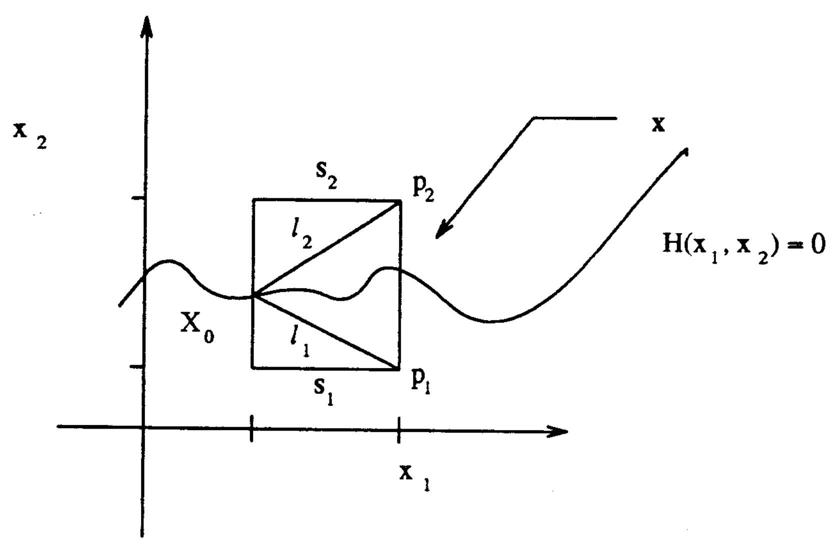

Figure 1

We assume that our algorithms start with a point $X_0 = (x_{01}, x_{02}) \in \mathcal{Z} \subset \mathbb{R}^2$ (i.e. $H(X_0) = 0$). If the gradient vector $H'(X)$ is of full rank at X_0, then the tangent vector $B(X_0)$ can be computed to within length and direction. This information can then be used to choose an appropriate parameter coordinate. Following [4], we choose the parameter coordinate changeably, so that curve following is as fast as possible and is successful. That is, we choose i_0 such that

$$|B_{i_0}(X)| = \max\{|B_1(X)|, |B_2(X)|\}.$$

We call x_{i_0} the parameter coordinate. If $\text{sgn}(B_{i'_0}) = -\text{sgn}(B_{1,i'_0})$, where i'_0 and B_1 are the previous parameter number and tangent vector, respectively, we change the direction of the current tangent vector B. This action rigorously maintains the orientation, as is proven in Theorem 3.4 below.

We say the continuation method is successful if the correct orientation is maintained and if successively computed points lie on the same path. We prove that the continuation method presented in this paper must be successful. In particular, suppose without loss of generality that the parameter coordinate corresponding to X_0 is x_1, and suppose that $B_1 > 0$. Construct a two dimensional box $\mathbf{x}$ containing X_0 on one of its faces and a portion of the curve in the positive direction (defined by B) from X_0. (See fig. 1 and Algorithm 2.2.) We define the shape of the box with a width parameter η and the predictor step length δ. Theorems 3.2 and 3.3 then guarantee that, if $\eta_{\min}$ large enough and some other assumptions hold, then the curve passing through X_0 passes out the face opposite X_0, and otherwise does not pass through the boundaries of $\mathbf{x}$. Define the faces

$$\mathbf{s}_1 = \{X = (x_1, x_2) \mid \quad X \in \mathbf{x} \quad \text{and} \quad x_2 = \inf(\mathbf{x}_2)\} \tag{1}$$

$$\mathbf{s}_2 = \{X = (x_1, x_2) \mid \quad X \in \mathbf{x} \quad \text{and} \quad x_2 = \sup(\mathbf{x}_2)\} \tag{2}$$

of $\mathbf{x}$. Now consider an interval extension $\mathbf{H}$ for the range of H. By Theorem 3.1, if the interval extensions $\mathbf{H}(\mathbf{s}_1)$, $\mathbf{H}(\mathbf{s}_2)$ and $\mathbf{H}_{x_2}(\mathbf{x})$ do not contain 0, where $\mathbf{H}_{x_j}$ represents an interval extension of the partial derivative of h with respect to x_j, then the curve does not intersect $\mathbf{s}_1$ and $\mathbf{s}_2$, there is at most one curve in the box $\mathbf{x}$, and the curve cannot have any turning points with respect to x_2 in $\mathbf{x}$.

2. ALGORITHMS

Algorithm 2.1. *(Overall continuation method)*

1. *Input the known point X_0 on the curve, the function H (and its partial derivatives), the stepsize δ and parameters $\delta_{\max}$, η, $\eta_{\min}$, $\eta_{\max}$, η_1, i'_0, and initial orientation B. Also set*
   $\eta \leftarrow \eta_{\min}$ and
   $B_1 \leftarrow B$.
2. *Compute the new tangent vector B and orient it so that the parameter coordinate of B_1 has the same sign as the parameter coordinate of B.*
3. *Construct the box $\mathbf{x}$ according to Algorithm 2.2.*
4. **If** $0 \in \mathbf{H}(\mathbf{s}_1)$,$0 \in \mathbf{H}(\mathbf{s}_2)$, *or* $0 \in \mathbf{H}_{x_j}(\mathbf{x})$, *where $j \neq i_0$ and i_0 is the parameter index,* **then**
   $\eta \leftarrow 2\eta$.
   If $\eta > \eta_{\max}$ **then**

$\delta \leftarrow \delta/2$
endif
Goto step 3.
endif

5. *Using the traditional Newton's method, compute an approximation to the point on the curve on the face* $\mathbf{s}_i$ *of* $\mathbf{x}$ *opposite the face containing* X_0.
6. **If** *some stopping criterion is met,* **then**
   Stop
   else
   $\delta \leftarrow 10\delta$,
   If $\delta > \delta_{\mathbf{max}}$, **then**
   $\delta \leftarrow \delta_{\mathbf{max}}$.
   endif
   $\eta_{\mathbf{max}} \leftarrow 10|B_i|$ *and*
   $\eta_{\mathbf{min}} \leftarrow 0.1|B_i|$, *where* i *is not the parameter coordinate index.*
   $\eta \leftarrow \eta_{\mathbf{min}}$.
   Goto step 2.
   endif

Algorithm 2.2. (Use the known point X_0 and tangent $B(X_0)$ to construct the box shape.)

1. *Input* δ, η, X_0 *and* $B(X_0)$.
2. **If** $i = i_0$, **then**

$$\mathbf{x}_i = \begin{cases} (x_{0i}, x_{0i} + \delta) & \text{if } B_{i_0} \geq 0 \text{ and} \\ (x_{0i} - \delta, x_{0i}) & \text{if } B_{i_0} < 0. \end{cases}$$

else

$$p_1 = x_{0i} + B_i\delta - \eta\delta$$
$$p_2 = x_{0i} + B_i\delta + \eta\delta$$

and

$$\mathbf{x}_i = \begin{cases} (2x_{0i} - p_2, p_2) & \text{if } x_{0i} < p_1, \\ (p_1, 2x_{0i} - p_1) & \text{if } x_{0i} > p_2, \text{ and} \\ (p_1, p_2) & \text{otherwise.} \end{cases}$$

end if

3. Mathematical Theorems

Assumption 1. Assume $H : \mathbb{R}^2 \to \mathbb{R}^1$ has continuous derivatives. Assume x_{i_0} is the parameter coordinate chosen in Algorithm 2.2. For notational simplicity but without loss of generality, throughout this section assume that the parameter coordinate $i_0 = 1$. Also assume that $\dfrac{\partial H}{\partial X_2} = H_{x_2} \neq 0$ in the box $\mathbf{x}$.

Theorem 3.1. *Suppose Assumption 1 is true. Also assume that* $\mathbf{H}(\mathbf{s}_1)$, $\mathbf{H}(\mathbf{s}_2)$ *and* $\mathbf{H}_{x_2}(\mathbf{x})$ *do not contain* 0. *Then there exists a unique smooth curve contained in the box* $\mathbf{x}$.

Proof. Since $\dfrac{\partial H}{\partial X_2} \neq 0$ in the box $\mathbf{x}$, the implicit function theorem implies that, for any $x_1 \in \mathbf{x}_1$, there is a point $(x_1, x_2(x_1))$ on the curve, and this set of points constitutes a continuous curve. Since the range of H over $\mathbf{s}_i$ is contained in the interval extension, i.e.

$$\{H(X) \mid \quad X \in \mathbf{s}_i\} \subset \mathbf{H}(\mathbf{s}_i),$$

the assumption implies that 0 is not contained in the range of H over $\mathbf{s}_i$, i.e. the curve will not intersect $\mathbf{s}_i$.

If there are two different curves in $\mathbf{x}$, then must exist an x_1 and x_2^1 and x_2^2 in $\mathbf{x}_2$ such that

$$H(x_1, x_2^1) = H(x_1, x_2^2).$$

By the mean value theorem, there must exist a $\xi \in \mathbf{x}_2$ such that

$$H_{x_2}(x_1, \xi) = 0,$$

where $\mathbf{x}_2$ is the first coordinate of the interval vector $\mathbf{x}$. This implies that the range of H_{x_2} over $\mathbf{x}$ contains 0, and thus $0 \in \mathbf{H}_{x_2}(\mathbf{x})$, contradicting the assumption. □

Theorem 3.2. *Let* $\mathbf{x}$ *be the box described in Algorithm 2.2. If Assumption 1 is true and if*

$$\eta_{\min} - 1 > M = \max \left| \frac{\partial H}{\partial X_1} \right| \Big/ \left| \frac{\partial H}{\partial X_2} \right|,$$

for all $X \in \mathbf{x}$, *then the curve starting at the point* X_0 *on the face* (1) *of the box* $\mathbf{x}$ *will pass out of the box* $\mathbf{x}$ *through the face* (2).

Proof. Without loss of the generality, assume $B_1 > 0$ so that

$$x_1 = \inf(\mathbf{x}_1) = x_{01},$$

where $\mathbf{x}_1$ is the first coordinate of the interval vector $\mathbf{x}$. Since $\dfrac{\partial H}{\partial X_2}$ is non zero at X_0, the implicit function theorem implies that there is a $\delta_{\mathbf{x}_1}$ such that there is a unique continuously differentiable path

(3) $$\{(x_1, x_2(x_1))^T \mid x_1 \in \mathbf{x}_1^*, \quad H(x_1, x_2(x_1)) = 0\}$$

in the interval $\mathbf{x}_1^* = [x_{01}, x_{01} + \delta_{\mathbf{x}_1}]$. By the chain rule,

$$\frac{dx_2}{dx_1} = -\frac{\partial H}{\partial X_1} \Big/ \frac{\partial H}{\partial X_2},$$

and by Assumption 1, H has continuous derivatives and $\dfrac{\partial H}{\partial X_2}$ is non zero in $\mathbf{x}$. Therefore, there must exist m_2 and m such that

$$\left| \frac{\partial H}{\partial X_1} \right| \leq m_2,$$

and

$$\left|\frac{\partial H}{\partial X_2}\right| \geq m > 0.$$

Thus,

$$\left|\frac{dx_2}{dx_1}\right| \leq \frac{m_2}{m} = M$$

on the curve (3). By the mean value theorem,

$$x_2(x_1) = x_2(x_{01}) + \frac{dx_2}{dx_1}(\xi)(x_1 - x_{01}), \qquad \text{for some } \xi \in \mathbf{x}_1^*.$$

Thus,

$$x_{02} - M(x_1 - x_{01}) < x_2(x_1),$$

and

$$x_{02} + M(x_1 - x_{01}) > x_2(x_1),$$

Therefore, $x_2(x_1)$ is contained within the two straight lines

$$x_2 = x_{01} + (B_2 - \eta)(x_1 - x_{10})$$

and

$$x_2 = x_{01} + (B_2 + \eta)(x_1 - x_{10}),$$

illustrated as the oblique lines l_1 and l_2 in fig. 1. Hence, by the construction in Algorithm 2.2, $x_2(x_1) \in \mathbf{x}_2$. Thus, the locally unique curve is contained in the box $\mathbf{x}$. □

Theorem 3.3. *Make Assumption 1, so that there is a curve passing through X_0. Further assume that H has continuous derivatives up to order 2, so that the derivatives $\frac{d^2x_2}{dx_1^2}$ are continuous*[1]. *Assume that*

$$\eta_{\min} > \frac{1}{2}\max\left|\frac{d^2x_2}{dx_1^2}\right|\delta$$

for all $X \in \mathbf{x}$. Then the curve passes through the faces (1) and (2) of the box $\mathbf{x}$ described in Algorithm 2.2.

Proof. By Taylor's theorem,

$$x_2(x_1) = x_2(x_{01}) + \frac{dx_2}{dx_1}(x_{01})(x_1 - x_{01}) + \frac{1}{2}\frac{d^2x_2}{dx_1^2}(\xi)(x_1 - x_{01})^2 \tag{4}$$

for some $\xi(x_1)$, for every $x_1 \in \mathbf{x}_1^* = [x_{01}, x_{01} + \delta]$. Comparing (4) to the expressions for p_1 and p_2 in Algorithm 2.2, and observing that the assumption implies

$$\eta > \frac{1}{2}\max\left|\frac{d^2x_2}{dx_1^2}\right|(x_1 x_{01})$$

for all $x_1 \in \mathbf{x}_1^*$, we obtain

$$p_1 < x_2(x_1) < p_2.$$

Thus, by the construction of $\mathbf{x}_2$, $x_2(x_1) \subset \mathbf{x}_2$. □

---

[1]Actually, in this situation the chain rule implies that $\frac{d^2x_2}{dx_1^2}$ can be expressed in terms of H and its derivatives.

Theorem 3.4. *Suppose i_0', $B_{i_0'}$, and $B1_{i_0'}$ are as in Algorithm 2.1. Also assume that the computed tangent vectors B and B_1 are contained in the set consisting of normalizations of vectors from $(\mathbf{H}_{x_2}(\mathbf{x}_0), -\mathbf{H}_{x_1}(\mathbf{x}_0))$. Then, after execution of Step 3 of Algorithm 2.1, B represents the same direction along the curve as in the previous step.*

Proof. In the previous box, there is a unique curve that can be parametrized in terms of the previous parameter $x_{i_0'}$. However, if the i_0'-th component $B_{i_0'}$ changes sign within that box, then there must be a turning point with respect to i_0', which contradicts the fact that there is unique curve with respect to i_0' in the previous box. Therefore, to maintain orientation, it is necessary and sufficient that $\text{sgn}(B_{i_0'}) = \text{sgn}(B1_{i_0'})$, provided B and B_1 are both in the null space of H' at the previous and present points on the curve, respectively.

We have verified that $0 \notin \mathbf{H}_{x_{i_1}}(\mathbf{x})$ in each constructed box $\mathbf{x}$, where x_{i_1} is not the parameter coordinate x_{i_0}. However, $\text{sgn}(B_{i_0})$ and $\text{sgn}(B_{i_0'})$ either equal corresponding values of $\text{sgn}(H_{x_{i_1}})$ or $\text{sgn}(-H_{x_{i_1}})$, depending on whether $i_1 = 1$ or $i_1 = 2$. Thus, by the assumption on the computed values for $B_{i_0'}$ and $B1_{i_0'}$, $\text{sgn}(B_{i_0'}) = \text{sgn}(B1_{i_0'})$. □

4. Numerical Experiments

A *rigorous* step control method, interval step control guarantees that the curve followed is unique, i.e. that it is not possible to jump from one path to another. Thus, the interval step control follows the curve properly, regardless of how the initial, minimum, and maximum stepsizes are chosen. In contrast, success of non interval step controls depends on how we choose these parameters, and we may need to be overly conservative. The following numerical results illustrate this.

We used FORTRAN-SC ([3]) on an IBM 3090 for the interval step control experiments. We used the software package PITCON 6.1 on the IBM 3090 for the non interval step control. We chose PITCON as a readily available high quality package with a representative non interval step control, although other packages may have served equally well[2].

In PITCON, we set the absolute error tolerance to 10^{-14} and relative error tolerance to 10^{-11}. We did not request the algorithm to locate limit points. We supplied a subroutine with an analytic representation of the Jacobian matrix, rather than using finite differences. Furthermore, we allowed the program to choose the parameter coordinate. Finally, we configured PITCON to reevaluate the Jacobian matrix after each step of the corrector iteration[3].

Behavior on the topologist's sine curve not near $x = 0$. The topologist's sine curve is defined as

$$\{(x,t) = (x, \sin(1/x)) \mid x \neq 0\}.$$

[2]See [8] and [9] for an explanation of the original version of PITCON.

[3]That is, we configured PITCON to use the classical Newton's method as its corrector iteration.

Table 4.1 shows the CPU time for interval step control is comparable with the CPU time needed by PITCON, when smooth parts of the curve are followed and when the parameters in PITCON are set to ensure that it succeeds. The table was obtained by running Algorithm 2.1 and PITCON on the topologist's sine curve from various starting points, proceeding forward in x to the target value $x = 10.0$. The CPU time and average step size are given. For both algorithms, the maximum allowable step size was set to 0.1. Here, making the starting x smaller than 5.0×10^{-2} caused PITCON to lose its orientation.

Controls	Start pt.	Aver. δ	CPU
Non interval	1.0×10^{00}	0.10×10^{00}	0.10
Interval	1.0×10^{00}	0.97×10^{-1}	0.09
Non interval	1.0×10^{-1}	0.10×10^{0}	0.18
Interval	1.0×10^{-1}	0.69×10^{-1}	0.61
Non interval	5.0×10^{-2}	0.99×10^{-1}	0.26
Interval	5.0×10^{-2}	0.43×10^{-1}	1.14

Table 4.1. Results for the topologist's sine curve not near $x = 0$.

Behavior on the topologist's sine curve near $x = 0$. The curve has a severe oscillation near $x = 0$, causing non interval curve following algorithms to "skip" or lose orientation. We obtained the graphs in Figure 4.2 by running Algorithm 2.1 and PITCON with starting point $(0.019, \sin(1/0.019))$. In PITCON, we used initial stepsize 0.1, minimum stepsize 0.05 and maximum stepsize 0.5. The interval algorithm with the same initial stepsize and maximum stepsize as we used in PITCON proceeds successfully past the point where PITCON 6.1 started to "skip." This is because the interval step control guarantees that there is only one point on the curve in the slice of the box perpendicular to the parameter coordinate. Therefore, when the algorithm approaches a point where the curvature is larger, it decreases the stepsize as much as necessary, then increases the stepsize after passing this point. But non interval step controls do not always correctly anticipate such points.

The heuristics in PITCON dictate that the stepsize be increased whenever the corrector iteration converged on the previous step. Since portions of the topologist's sine curve are nearly linear, this strategy results in a stepsize near the maximum. The algorithm thus cannot detect the numerous turning points and rapid changes in curvature, and it loses its orientation. If reliable curve following is necessary in this context, then a correspondingly smaller maximum step size is required as t becomes smaller. Unfortunately, how small the maximum step size should be is hard to predict. But this is not a concern in the interval step control. The interval step control algorithm always follows the unique curve, and never loses the orientation. It increases the step size in regions of small curvature, and decreases it otherwise. This makes the interval step control more efficient than the non interval step control for curve with similar properties. Table 4.1 contains CPU times as a function of t, assuming both step controls are successful.

Table 4.2 presents comparisons of Algorithm 2.1 and PITCON for the topologist's sine curve near $t = 0$. We started the algorithms at 1.0×10^{-2}, and chose the direction of decreasing x as the orientation. The CPU time for PITCON is the CPU time up to the point where PITCON loses its orientation, and the CPU time for the interval step control is the CPU time over the same interval. The column labeled "stop pt." gives the point where PITCON loses its orientation, for the corresponding maximum step size.

Controls	Max δ	Stop pt.	CPU
Non interval	1.0×10^{-3}	6.56×10^{-3}	36.1
Interval	1.0×10^{-1}	6.56×10^{-3}	8.8
Non interval	1.0×10^{-4}	4.99×10^{-2}	199
Interval	1.0×10^{-1}	6.56×10^{-3}	1.14

Table 4.2. Results for the topologist's sine curve near $x_1 = 0$.

It is evident from Table 4.2 that running PITCON with maximum stepsize 1.0×10^{-5} until it reached the point where it loses its orientation would be very expensive. However, when we tried Algorithm 2.1 on IBM 3090, it was able to reach the point where $x = 1.41 \times 10^{-4}$. To analyze this case, consider the positive integer n such that

$$\frac{1}{(n+1)\pi} \leq x \leq \frac{1}{n\pi}.$$

This illustrates that two consecutive zeros become very close. The distance can be estimated as

$$\begin{aligned} d_n &= \left| \frac{1}{n\pi} - \frac{1}{(n+1)\pi} \right| \\ &\leq \frac{1}{(n+1)^2\pi} \\ &\leq \pi x^2 \\ &\leq 7.0 \times 10^{-8}. \end{aligned}$$

Also, the distance between two legs of the curve near the maximum and minimum points of this function becomes even smaller. Using Algorithm 2.1, we found the smallest stepsize taken near the maximum and minimum point to be roughly 10^{-16}, near $x = 1.41 \times 10^{-4}$. The floating point accuracy prevented the algorithm from proceeding past this point.

Behavior on a parametrized family of hyperbolas. Define a parametrized family of functions $f(x,t) : \mathbb{R}^2 \to \mathbb{R}$ by

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

where p is a shape parameter. $f = 0$ defines the simple hyperbolic curve. The curve has two disjoint branches, with vertices $(-p, 0.5)$ and $(p, 0.5)$. As p tends to 0, the two branches become closer and closer, and the curve degenerates into two straight lines, as in the left graph of Figure 4.3. The experiment in this figure was done by setting $p = 10^{-5}$ and using starting point $(0.5, 0)$, initial stepsize 0.01, minimum stepsize 0.0001, and maximum stepsize 0.01. Figure 4.3 shows that PITCON jumps from one branch to the other. In addition we tried $p = 10^{-15}$, with results given in Table 4.3. In this case, the distance between the two vertices is only $2p = 2 \times 10^{-15}$.

However, the interval step control is still successful in that case. We note that 10^{-16} is near the double precision machine epsilon on the IBM 3090.

CPU:	2.177
Aver. δ:	0.1856×10^{-2}
Total steps:	578

Table 4.3. Around a hyperbola's vertex with $p = 10^{-15}$.

Table 4.4 table gives point coordinates and the stepsize δ for selected points near the vertex $(p, 0.5)$, for $p = 10^{-15}$, for our interval step control. This table does not include all points (x, t) which were computed within this range, but is meant to show the way that the stepsize decreases harmonically as t tends to 0.5.

x	t	δ—stepsize
0.48381×10^{-12}	$0.4999999999995161 \times 10^{00}$	0.43695×10^{-14}
0.65608×10^{-13}	$0.4999999999999344 \times 10^{00}$	0.34467×10^{-14}
0.60961×10^{-14}	$0.4999999999999939 \times 10^{00}$	0.51861×10^{-15}
0.11914×10^{-14}	$0.5000000000000006 \times 10^{00}$	0.24149×10^{-15}
0.71785×10^{-14}	$0.5000000000000072 \times 10^{00}$	0.37734×10^{-14}
0.25215×10^{-13}	$0.5000000000000252 \times 10^{00}$	0.14394×10^{-14}
0.26241×10^{-12}	$0.5000000000002624 \times 10^{00}$	0.10982×10^{-12}

Table 4.4. Points near a hyperbola's vertex, with $p = 10^{-15}$

5. Summary, Conclusions and Future Work

We have presented algorithms for interval step control for continuation methods in $\mathbb{R}^2$, along with theoretical clarification that these algorithms are qualitatively more reliable than algorithms based on non interval step controls. We have presented several classes of numerical experiments that illustrate the immunity of these interval methods to certain types of failures. These experiments illustrate that these methods also can be practical on various problems.

State-of-the-art implementations of continuation methods based on non interval step controls function efficiently and fairly predictably, provided sufficient knowledge is known about the problem to set the algorithm tolerances appropriately, or provided it is possible to repeatedly redo the computation after human interaction. However, interval step controls should be considered in cases in which rigor is critical or in which human interaction with the method is not possible.

Methods to estimate the bounds M and second derivative bounds appearing in Theorems 3.2 through 3.3 will allow for an even more efficient algorithm that, besides not giving erroneous results, will always succeed in following the curve, without human intervention.

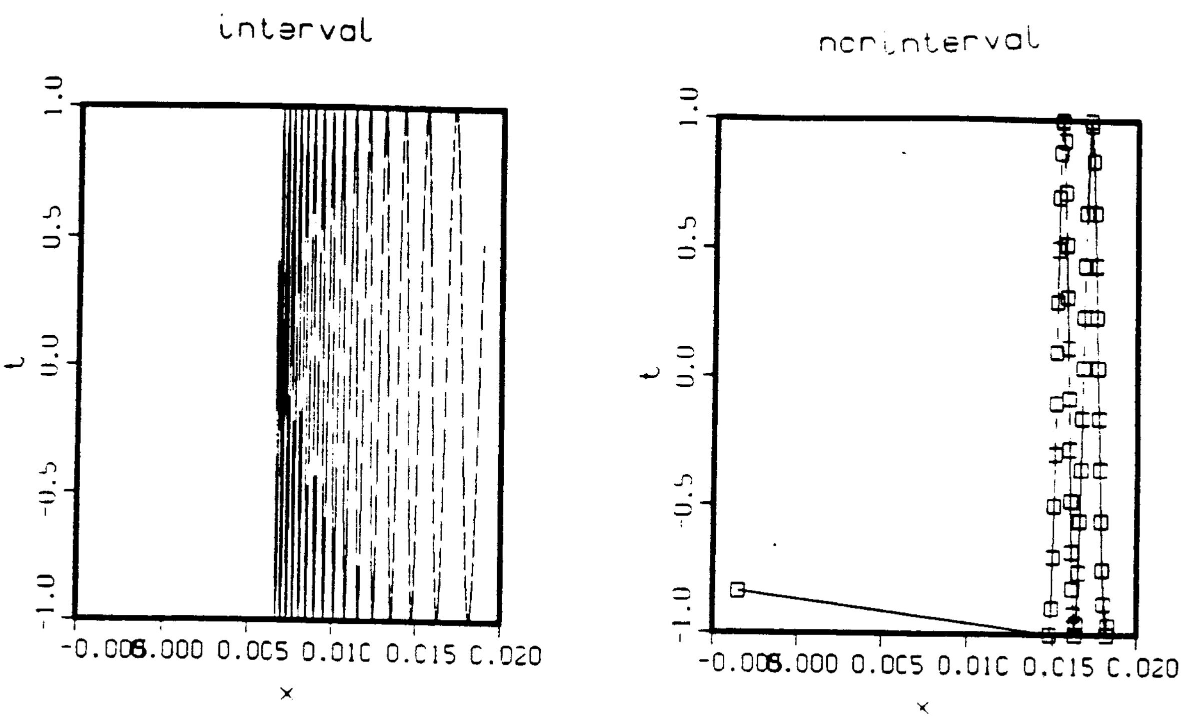

Figure 4.2. TOPOLOGIST'S SINE CURVE

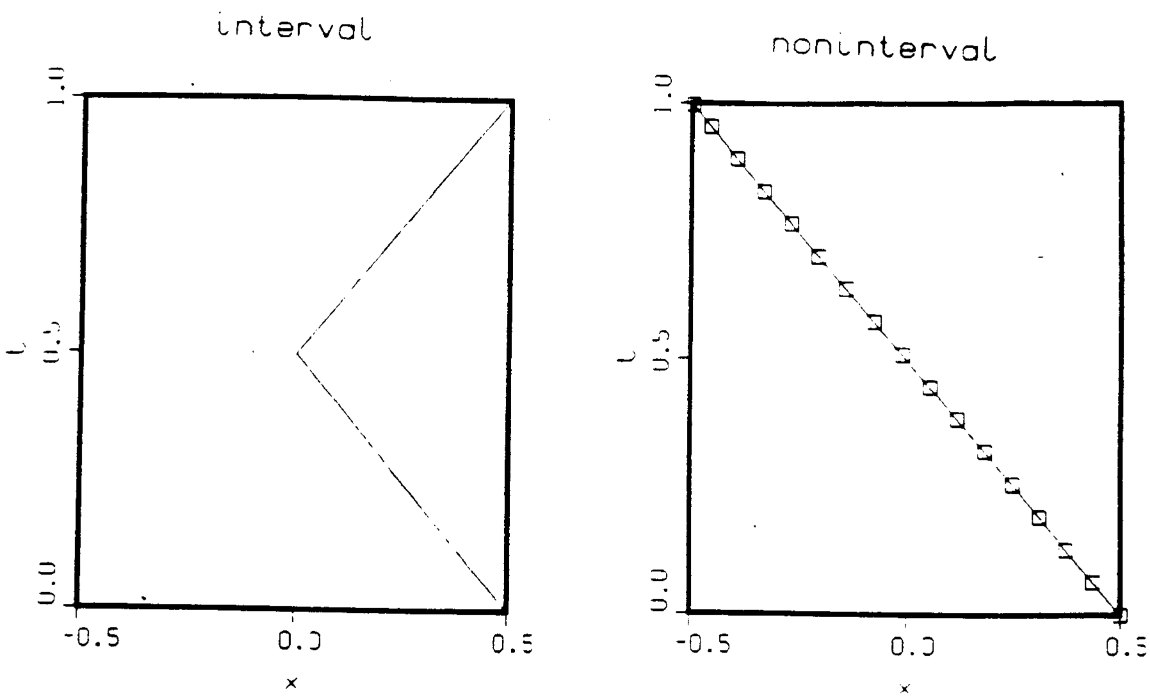

Figure 4.3. ALMOST DEGENERATE HYPERBOLA.

REFERENCES

1. Alefeld, Götz, and Herzberger, Jürgen, *Introduction to Interval Computations*, Academic Press, New York, etc., 1983.

2. Allgower, E., and Georg, K., *Numerical Continuation Methods: An Introduction*, Springer-Verlag, New York, 1990.
3. Bleher, J. H., Rump, S. M., Kulisch, U., Metzger, M., Ullrich, C., and Walter, W., *Fortran-SC – A study of a Fortran extension for engineering scientifi c computation with access to ACRITH*, Computing **39** (1987), no. 2, 93–110.
4. Heijer, C. den, and Rheinboldt, W. C., *On Steplength Algorithms for a Class of Continuation Methods*, SIAM J. Numer. Anal. **18** (1981), no. 5, 925–948.
5. Moore, Ramon E., *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.
6. Neumaier, A., *The Enclosure of Solutions of Parameter-Dependent Systems of Equations*, Reliability in Computing: The Role of Interval Methods in Scientific Computing, Academic Press, New York, etc., 1988, pp. 269–286.
7. Neumaier, A., *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, England, 1990.
8. Rheinboldt, W. C., Burkardt, J. V., *A Locally Parameterized Continuation Process*, ACM Trans. Math. Software **9** (1983), no. 2, 215–235.
9. Rheinboldt, W. C., and Burkardt, J. V., *A Program for a Locally-Parametrized Continuation Process*, ACM Trans. Math. Software **9** (1983), no. 2, 215–35.

U.S.L. Box 4-1010, Lafayette, LA 70504-1010