

An Interval Step Control for Continuation Methods

R. BAKER KEARFOTT

Department of Mathematics
University of Southwestern Louisiana

Abstract. We present a step control for continuation methods which is deterministic in the sense that (i) it will never cause the algorithm to jump across paths; (ii) it *computationally but rigorously* verifies that the corrector iteration will converge; and (iii) each predictor step is as large as possible, subject to assured convergence of the corrector iteration. The technique is general in the sense that it can be used with various choices of predictor direction and corrector manifold. We present performance data and comparisons for Brown's almost linear function and the Layne Watson exponential function in various dimensions.

1. MOTIVATION AND INTRODUCTION

Throughout, we will assume knowledge of both continuation methods and interval mathematics. An excellent introduction to continuation methods appears in [2]. Thorough introductions to interval methods appear in the books [13] or [1]. The proceedings [14] contains surveys and descriptions of applications of interval methods, while the recent survey [10] summarizes some of the basic facts.

Continuation methods are used to compute sequences of points on solution manifolds of systems of n equations in $n + 1$ unknowns; i.e., they are used to compute points on arcs in the set

$$(1.1) \quad \mathbf{Z} = \{Y \in \mathbf{R}^{n+1} \mid H(Y) = 0\},$$

where $H: \mathbf{R}^{n+1} \rightarrow \mathbf{R}^n$.

There is a wealth of literature on applications which involve finding arcs in the set \mathbf{Z} of (1.1). These applications divide naturally into two classes. In the first class, H models a parametrized physical system, which we wish to study as the parameter varies, such as in [12]. In the second class, we introduce an artificial parameter in order to solve (or find *all* solutions to) a system of n nonlinear equations in n unknowns for which locally convergent methods are unsuccessful. For problems in the second class, we may have mathematically rigorous guarantees that we will find all solutions to the original system of equations, provided the continuation method reliably tracks the arcs in \mathbf{Z} of (1.1); see [15], for an introductory explanation.

Many continuation methods are instances of the following general predictor-corrector algorithm.

ALGORITHM 1.1.

1. Input:

- (a) the initial point on the arc $Y \leftarrow Y_0$ and an arc-following stopping criterion \mathcal{PS} ;
- (b) the minimum predictor stepsize δ_{\min} , and the maximum predictor stepsize δ_{\max} ; and
- (c) a corrector stopping criterion \mathcal{CS} .

2. (Predictor-corrector iteration) Repeat the following until \mathcal{PS} is satisfied.¹

- (a) Compute a predictor step length δ ;
- (b) Compute a predictor step direction B (where $\|B\|_2 = 1$) and a predicted point on the arc

$$Z_0 = Y + \delta B.$$

- (c) Choose an n -dimensional corrector manifold \mathbf{S} such that $Z_0 \in \mathbf{S}$.
- (d) Compute a corrected point Z by using a locally convergent method whose starting point is Z_0 and whose iterates all lie in \mathbf{S} ; iterate the method until \mathcal{CS} is satisfied.
- (e) Replace Y by Z .

The direction B may be taken to be tangent to the arc, it may be a coordinate vector which is adaptively chosen on each corrector step, it may be a fixed coordinate vector, or it may only be implicitly given as $B = Y - Z_0$ when a higher-order method is used to determine Z_0 .

The corrector iteration manifold \mathbf{S} is often taken to be the affine space containing Z_0 and orthogonal to B . In that case, the corrector iteration may be defined by applying Newton's method or variations (such as quasi-Newton methods) to the augmented system $\tilde{H}(Y) = 0$, where

$$(1.2) \quad \tilde{H}(Z) = \begin{pmatrix} H(Z) \\ B^T(Z - Z_0) \end{pmatrix} = 0.$$

Alternately, for the same \mathbf{S} as in (1.2), the corrector iteration may consist of applying Newton's method to the system $\hat{H} = 0$, where

$$(1.3) \quad \hat{H}(P) = H\left(Z_0 + \sum_{j=0}^n p_j W_j\right) = 0,$$

where $P = (p_1, p_2, \dots, p_n)$ and where $\{W_j\}_{j=1}^n$ is a set of orthogonal vectors such that $W_j \circ B = 0$ for j between 1 and n .

¹The items in this step may be computed in an order different from that given here, depending on the details of how they are computed.

Step 2(a) of Algorithm 1.1, termed the *steplength algorithm*, has been much studied; see [5], [6], [7], and [20], among much such work. As is mentioned in [7], such step controls involve heuristics which, under certain circumstances, may lead to failure of Algorithm 1.1. In particular, if δ is chosen too large, then the corrector iteration may converge to a point which is not on the original arc; see Figure 1. Such failures would be difficult to detect automatically, and may lead to misleading conclusions about the problem. The failure can be corrected for specific problems by using a more conservative heuristic, but only provided we know it has occurred.

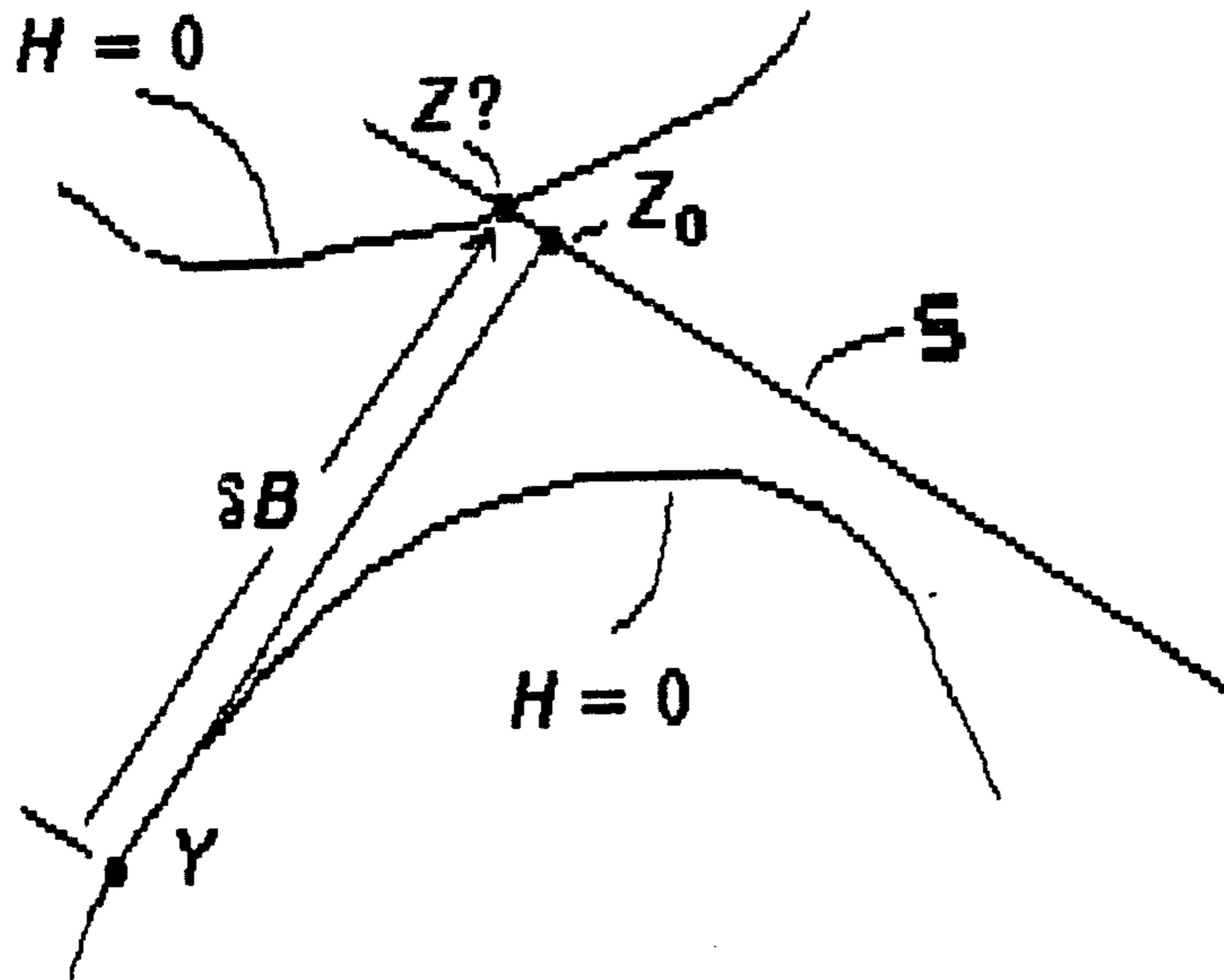


Fig. 1. An undetectable failure of the step control algorithm.

In this paper, we describe a step control for which it is mathematically impossible for failures as in Figure 1, even when the procedure is implemented on a finite precision computer. Our step control is based on interval arithmetic procedures which provide computational verification that there is a unique root in a given region, and that Newton's method will converge to that root. Such procedures generally give unambiguous results provided the size of the region is small enough, while, in our context, we may control the region's size by increasing or decreasing the stepsize.

An alternate interval method for describing the solution set \mathbf{Z} of (1.1)

is given in [18]. In that method, Neumaier obtains the entire solution set, as opposed to a set of points on a single arc. However, that technique involves (adaptive) subdivision of the entire space; for some problems, the amount of work may increase too rapidly as a function of n . In contrast, the method proposed here reliably solves a more modest problem, but may possibly be more easily modified to work practically for larger n .

In Section 2, we present the ideas underlying the interval step control. In Section 3, we give the interval step control algorithm, and we also present a simple but successful non-interval step control which we will use for comparisons. In Section 4, we present our numerical comparisons. In Section 5, we summarize and present conclusions.

2. THE UNDERLYING CONVERGENCE TEST

Throughout, we will denote points in \mathbf{R}^n or n -vectors by uppercase letters and scalars by lowercase letters. We will denote the corresponding interval quantities by boldface.

We denote the box in n -space described by

$$\{X = (x_1, x_2, \dots, x_n) \mid l_i < x_i < u_i \text{ for } 1 \leq i \leq n\}$$

by \mathbf{X} . We may then transform the nonlinear problem

$$(2.1) \quad \begin{aligned} &\text{Find all solutions of the nonlinear system} \\ &F(X) = (f_1(x_1, x_2, \dots, x_n), \dots, f_n(x_1, x_2, \dots, x_n)) = 0, \\ &\text{where bounds } l_i \text{ and } u_i \text{ are known such that} \\ &l_i \leq x_i \leq u_i \text{ for } 1 \leq i \leq n \end{aligned}$$

to the linear interval problem

$$(2.2) \quad \begin{aligned} &\text{Find a box } \overline{\mathbf{X}} \text{ which contains all solutions of the} \\ &\text{linear interval system} \\ &\mathbf{F}'(\mathbf{X})(\overline{\mathbf{X}} - X) = -F(X), \\ &\text{where } \mathbf{F}'(\mathbf{X}) \text{ is an elementwise interval extension} \\ &\text{of the Jacobian matrix,} \end{aligned}$$

and where $X \in \mathbf{X}$ is an approximation to a root of F . (See, eg. [13].)

Neumaier shows in [17] that, for many common ways of solving (2.2),

$$(2.3) \quad \begin{aligned} &\text{if } \overline{\mathbf{X}} \subset \mathbf{X}, \text{ then the system of equations in (2.1)} \\ &\text{has a unique solution in } \mathbf{X}, \text{ and Newton's method} \\ &\text{starting at } X \text{ will converge to that solution.} \end{aligned}$$

Furthermore, if directed roundings (*cf.* [13]) are used, then the condition in (2.3) may be checked with mathematical rigor on a computer.

Our interval step control will be based on (2.2) and (2.3). By including the interval value $\mathbf{d} = [0, \delta]$ in the computations, we will verify the condition (2.3) simultaneously for all corrector iterations corresponding to predictor steps between 0 and δ .

In the remainder of this section, we will first present the convergence test based on (2.2) and (2.3) in a general setting. We will then describe the specific version of this convergence test which we used in the numerical experiments.

Suppose that the corrector manifold \mathbf{S} corresponding to initial point Y and predictor step $\bar{\delta}$ is given by

$$(2.4) \quad \mathbf{S} = \{r(Y, \bar{\delta}, P) \mid P \in \mathbf{R}^n\}$$

Also choose a function $q(\bar{\delta})$ such that

- (1) $q(0) = 0$;
- (2) $q(\bar{\delta})$ is a monotonically increasing function of $\bar{\delta}$;

We make the following assumption in order to assure that the test will accept the predictor step for all sufficiently small δ .

ASSUMPTION 2.1. *The functions r and q are continuous, and r is one-to-one. Furthermore, for each c such that $0 < c \leq 1$ and each $\epsilon > 0$, there exists a $\delta_0 > 0$ such that, if $\bar{\delta} \leq \delta_0$, then, for every W with $\|W\|_2 = 1$ and $W \circ B \geq c$, there is a $V = \eta W$ for some η with $\eta \leq \epsilon$, with $V = r(Y, \bar{\delta}, \bar{X}) - Y$ for some \bar{X} in the interior of $\{X \in \mathbf{R}^n \mid \|X\|_\infty \leq \bar{\delta}q(\bar{\delta})\}$. Also, the Jacobian matrix $\partial r(Y, \delta, X)/\partial X$ is of full rank for every X such that $\|X\|_\infty \leq \bar{\delta}q(\bar{\delta})$, and each vector U with $U \circ (r(Y, \delta, X) - Y) = 0$ is in the range of $\partial r(Y, \delta, X)/\partial X$.*

The conditions on the Jacobian matrix $\partial r(Y, \delta, X)/\partial X$ preclude the surface $r(Y, \delta, *)$ from dipping tangentially towards Y . They also assure that the surfaces $r(Y, \delta, \mathbf{O})$ where \mathbf{O} is a fixed open set in the interior of \mathbf{X} with $0 \in \mathbf{O}$ will intersect, in the limit as $\delta \rightarrow 0$, all arcs emanating from Y at an acute angle with B . These conditions are convenient in the proofs, have not been proven to be necessary.

Assumption 2.1 would hold if \mathbf{S} is defined as in (1.3), if

$$r(Y, \delta, P) = Y + \delta B + \sum_{j=0}^n p_j W_j,$$

and if $q(\delta) = \delta^{-\eta}$ for some $\eta > 0$. This situation is illustrated in Figure 2.

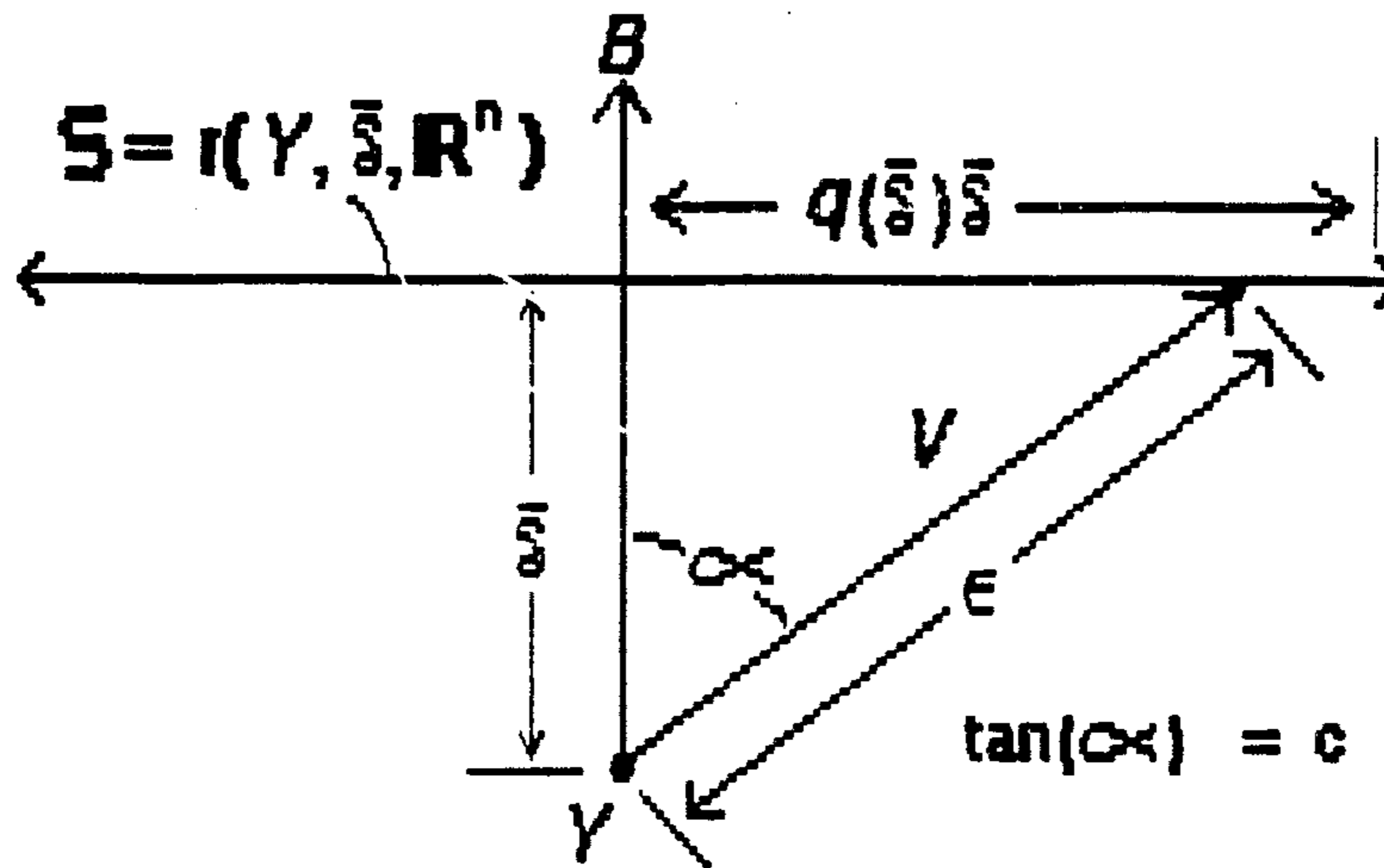


Fig. 2. Illustration of Assumption 2.1 when S is the hyperplane perpendicular to B .

Assumption 2.1 would also hold if the S corresponding to $\bar{\delta}$ were a sphere of radius $\bar{\delta}$ centered at Y , and if the parametrization P and $q(\delta)$ were such that $\{X \in \mathbf{R}^n \mid \|X\|_\infty \leq \bar{\delta}q(\bar{\delta})\}$ were the hemisphere centered at $Y + \bar{\delta}B$. (Such could be arranged easily with a parametrization in terms of spherical coordinates and with $\delta q(\delta)$ the appropriate constant.)

When we solve (2.2) for \bar{X} , the components of \bar{X} will generally have smaller widths than those of X provided each matrix A in the interval extension $F'(X)$ of the Jacobian matrix of F is well-conditioned and provided the widths of the entries of $F'(X)$ are sufficiently small. (Compare with Proposition 5 in [17].) For common interval extensions of components of continuous F' , the widths of $F'(X)$ tend to zero as the widths of X tend to 0. In other words, if there is an $X^* \in X$ with $F(X^*) = 0$, if the entries of F' are continuous, and if $F'(X^*)$ is nonsingular, then (2.3) will hold provided the widths of the components of X are sufficiently small. For this reason, we wish to have the following property.

ASSUMPTION 2.2. Assume $\lim_{\delta \rightarrow 0} \delta q(\delta) = 0$.

We may now state our main results. For $0 \leq \bar{\delta} \leq \delta$, let

$$\bar{X}_{\bar{\delta}} = \{X \in \mathbf{R}^n \mid \|X\|_\infty \leq \bar{\delta}q(\bar{\delta})\},$$

let

$$(2.5) \quad \mathbf{X}_\delta = \tilde{\mathbf{X}}_\delta = \bigcup_{0 \leq \bar{\delta} \leq \delta} \tilde{\mathbf{X}}_{\bar{\delta}}.$$

Define

$$\tilde{F}_\delta(X) = H(r(Y, \delta, X)),$$

so that

$$\tilde{F}'_\delta(X) = H'(r(Y, \delta, X)) \frac{\partial r(Y, \delta, X)}{\partial X}.$$

Furthermore, define the extension of $\tilde{F}_\delta(X)$ to \mathbf{d} by

$$\tilde{F}_{\mathbf{d}}(X) = H(r(Y, \mathbf{d}, X)),$$

and extend $\tilde{F}_{\mathbf{d}}$ to \mathbf{X}_δ by

$$\tilde{\mathbf{F}}_\delta(\mathbf{X}_\delta) = H(r(Y, \mathbf{d}, \mathbf{X}_\delta)).$$

Similarly extend \tilde{F}'_δ by

$$(2.6) \quad \tilde{\mathbf{F}}'_\delta(\mathbf{X}_\delta) = H'(r(Y, \mathbf{d}, \mathbf{X}_\delta)) \frac{\partial r}{\partial X}(Y, \mathbf{d}, \mathbf{X}_\delta).$$

where $\frac{\partial r}{\partial X}(Y, \mathbf{d}, \mathbf{X}_\delta)$ is the matrix of partial derivatives of r with respect to the variables X , formally evaluated with argument Y and interval arguments \mathbf{d} and \mathbf{X}_δ . We then have

THEOREM 2.3. *Assume $\check{X} \in \mathbf{X}_\delta$. Consider (2.2), with \mathbf{X}_δ replacing \mathbf{X} , $\tilde{F}_{\mathbf{d}}(\check{X})$ replacing $F(X)$ (where $\check{X} \in \mathbf{X}_\delta$ replaces X), and $\tilde{\mathbf{F}}'_\delta(\mathbf{X}_\delta)$ replacing $\mathbf{F}'(\mathbf{X})$. Also assume that the interval enclosure $\bar{\mathbf{X}}$ of the solution set to (2.2) is inclusion monotonic with respect to $\mathbf{F}'(\mathbf{X})$ and $F(X)$.² If $\bar{\mathbf{X}} \subset \mathbf{X}_\delta$, then, for each $0 \leq \bar{\delta} \leq \delta$, the system of equations $\tilde{F}_{\bar{\delta}}(X) = H(r(Y, \bar{\delta}, X)) = 0$ has a unique solution $X_{\bar{\delta}}$, and Newton's method applied to $\tilde{F}_{\bar{\delta}}(X)$ and with starting point \check{X} will converge to $X_{\bar{\delta}}$.*

PROOF OF THEOREM 2.3: As described in the statement of the theorem, (2.2) becomes

$$(2.7) \quad \left[H'(r(Y, \mathbf{d}, \mathbf{X}_\delta)) \frac{\partial r(Y, \delta, X)}{\partial X}(Y, \mathbf{d}, \mathbf{X}_\delta) \right] (\bar{\mathbf{X}} - \check{X}) = -\tilde{F}_{\mathbf{d}}(\check{X}).$$

On the other hand, the system corresponding to (2.2) for $\tilde{F}_{\bar{\delta}}(X) = 0$ is

²Such is the case for the methods of obtaining $\bar{\mathbf{X}}$ treated in [16] and [17].

$$(2.8) \quad \left[H' (r(Y, \bar{\delta}, \mathbf{X}_{\bar{\delta}})) \frac{\partial r(Y, \delta, X)}{\partial X} (Y, \bar{\delta}, \mathbf{X}_{\bar{\delta}}) \right] (\bar{\mathbf{X}}_{\bar{\delta}} - \check{X}) = -\tilde{F}_{\bar{\delta}}(\check{X}).$$

Comparing, we see

$$\begin{aligned} & \left[H' (r(Y, \bar{\delta}, \mathbf{X}_{\bar{\delta}})) \frac{\partial r(Y, \delta, X)}{\partial X} (Y, \bar{\delta}, \mathbf{X}_{\bar{\delta}}) \right] \\ & \subseteq \left[H' (r(Y, \mathbf{d}, \mathbf{X}_{\delta})) \frac{\partial r(Y, \delta, X)}{\partial X} (Y, \mathbf{d}, \mathbf{X}_{\delta}) \right], \end{aligned}$$

provided we have made inclusion monotonic interval extensions (cf. [13]). Since we also have $\tilde{F}_{\bar{\delta}}(\check{X}) \subseteq \tilde{F}_{\mathbf{d}}(\check{X})$ and since we are also assuming that $\bar{\mathbf{X}}$ in (2.2) is inclusion monotonic with respect to the interval function and Jacobian matrices, we obtain

$$\bar{\mathbf{X}}_{\bar{\delta}} \subseteq \bar{\mathbf{X}} \subset \mathbf{X}_{\delta},$$

where the second inclusion is true by hypothesis. Therefore, (2.3) is true for $\bar{\mathbf{X}}_{\bar{\delta}}$ replacing $\bar{\mathbf{X}}$, with \mathbf{X}_{δ} replacing \mathbf{X} , and with $\tilde{F}_{\bar{\delta}}$ replacing F in (2.1). That completes the proof.

Theorem 2.3 assures us that, if (2.7) holds, then there is a unique solution to the corrector equation (as in (1.2) or (1.3)) for each predictor step $\bar{\delta} \in [0, \delta]$, and corrector iteration with initial guess \check{X} will converge to that solution.

The next theorem will assert that the point $X_{\bar{\delta}}$ of Theorem 2.3 to which corrector iteration converges corresponds to a point $Z \in \mathbf{Z}$ of (1.1) on the same connected component of \mathbf{Z} as the previous point Y ; in other words, we are assured that the phenomenon of Figure 1 cannot occur. We make the following assumption in order to apply the implicit function theorem as in [4].

ASSUMPTION 2.4. *Assume that H , q , and r have continuous first partial derivatives.*

We make the following assumption so that, in combination with Assumption 2.1, we may conclude that $r(Y, \mathbf{d}, \mathbf{X}_{\delta})$ has an $(n+1)$ -dimensional interior.

ASSUMPTION 2.5. *If $H'(Y)V_0 = 0$, then $V_0 \circ B \neq 0$.*

THEOREM 2.6. *Suppose the hypotheses of Theorem 2.3 and Assumptions 2.1, 2.4, and 2.5 hold. Then the point $X_{\bar{\delta}}$ to which corrector*

iteration converges corresponds to a point Z on the same arc of \mathbf{Z} as Y . Furthermore, the Jacobian matrix is nonsingular at every point on the arc between Y and Z .

PROOF OF THEOREM 2.6: First, observe $\bar{\mathbf{X}}$ must contain all solutions to all systems of equations of the form $A(\bar{\mathbf{X}} - \check{X}) = \bar{F}_\delta(\check{X})$, where $A \in \bar{\mathbf{F}}'_\delta(\mathbf{X}_\delta)$; therefore, $\bar{F}'_\delta(X)$ is nonsingular for every $X \in \mathbf{X}_\delta$ and every $\bar{\delta}$ with $0 \leq \bar{\delta} \leq \delta$. Thus, $H'(r(Y, \delta, X))$ is of full rank and $\partial r(Y, \delta, X)/\partial X$ must be nonsingular for every $(\delta, X) \in \mathbf{d} \times \mathbf{X}_\delta$, so the implicit function theorem as stated on p. 20 of [4] is true at every point $Z_0 \in r(Y, \mathbf{d}, \mathbf{X}_\delta)$. Also, $r(Y, \mathbf{d}, \mathbf{X}_\delta)$ is compact and, by Assumption 2.1 and Assumption 2.4, must be $(n+1)$ -dimensional. These facts allow us to conclude that $\mathbf{Z} \cap r(Y, \mathbf{d}, \mathbf{X}_\delta)$ consists of a finite number of nonintersecting arcs and circles.

Let $Y_0(s)$ represent the arc such that $Y_0(0) = Y$ and such that s represents arclength, let $V(s) = Y_0(s) - Y$, and let V_0 denote the tangent to the arc at Y , i.e. $V_0 = Y_0'(0)$. Furthermore, orient V_0 so that $V_0 \circ B = 2c_0 > 0$. Also, since $\lim_{s \rightarrow 0} \{V(s)/\|V(s)\|_2\} = V_0$, there is an \bar{s} such that, for $s < \bar{s}$, $(V(s) \circ B)/\|V(s)\|_2 \geq c_0$.

The above and Assumption 2.1 allow us to conclude that there is a $\delta_0 \leq \delta$ such that, for each $\bar{\delta} \leq \delta_0$, there is an $s < \bar{s}$ such that $Y_0(s)$ corresponds to a solution of $\bar{F}'_{\bar{\delta}}(X) = 0$ in \mathbf{X}_δ . To see this, choose $\epsilon < \max_{s \leq \bar{s}} \|V(s)\|_2$, then use Assumption 2.1 to set a δ_0 so that, if $\bar{\delta} \leq \delta_0$, there are W and V as in Assumption 2.1 with $W \circ B \geq c_0$. Now note that, since $\lim_{s \rightarrow 0} \|V(s)\|_2 = 0$, there are an $s_1 \leq \bar{s}$ such that $\|V(s_1)\|_2 = \min_{W \in r(Y, \bar{\delta}, \mathbf{X}_\delta)} \|W\|_2$, and an s_2 with $s_1 \leq s_2 \leq \bar{s}$ with $\|V(s_2)\|_2 = \max_{W \in r(Y, \bar{\delta}, \mathbf{X}_\delta)} \|W\|_2$. Then, for each s with $s_1 \leq s \leq s_2$, define $W(s) \in r(Y, \bar{\delta}, \mathbf{X}_\delta)$ to be a vector in $r(Y, \bar{\delta}, \mathbf{X}_\delta)$ in the same direction as $V(s)$. ($W(s)$ must exist by Assumption 2.1.) We then set $h(s) = \|V(s)\|_2 - \|W(s)\|_2$. Then, from the portion of Assumption 2.1 dealing with $\partial r/\partial X$, $W(s)$ may be defined so that $h(s)$ is continuous. Furthermore, $h(s_1) \leq 0$ and $h(s_2) \geq 0$. Therefore, for some s_3 between s_1 and s_2 , $h(s_3) = 0$, so $V(s_3) = W(s_3)$. Thus, since Theorem 2.3 asserts that such solutions in \mathbf{X}_δ are unique, if $\delta = \delta_0$, the conclusion of Theorem 2.6 is true.

Assume that $\delta > \delta_0$, and that the conclusion to Theorem 2.6 is not true. We note that $\{Y_0(s) \mid s \geq 0\} \cap r(Y, \mathbf{d}, \mathbf{X}_\delta)$ is closed since \mathbf{X}_δ and \mathbf{d} are closed. But, if for each $\bar{\delta} \leq \delta$, there were an $s_{\bar{\delta}}$ with $Y_0(s_{\bar{\delta}}) \in r(Y, \bar{\delta}, \mathbf{X}_\delta)$, then Theorem 2.6 would be true by the uniqueness conclusion of Theorem 2.3. Therefore, we must conclude there are an s_4 and a $\delta_1 \geq \delta_0$ such that $Y_0(s_4) \in \partial r(Y, \delta_1, \mathbf{X}_\delta)$, and there is no

point of $Y_0(s)$ in $r(Y, \bar{\delta}, \mathbf{X}_\delta)$ for $\bar{\delta} > \delta_1$. (See Figure 3.) However, for each $\bar{\delta}$ with $\delta_1 < \bar{\delta} < \delta$, there must be an arc $Y_{\bar{\delta}}(s)$ and an $s_{\bar{\delta}}$ such that $Y_{\bar{\delta}}(s_{\bar{\delta}}) \in r(Y, \bar{\delta}, \mathbf{X}_\delta)$. But $Y_{\bar{\delta}}(s) \cap r(Y, \mathbf{d}, \mathbf{X}_\delta)$ must also be closed, so there is a δ_2 with $\delta_1 < \delta_2 < \bar{\delta}$ such that there is no point of $Y_{\bar{\delta}}$ in $r(Y, \delta_*, \mathbf{X}_\delta)$ for $\delta_* < \delta_2$. Therefore, there must be a third arc Y_{δ_*} which intersects $r(Y, \mathbf{d}, \mathbf{X}_\delta)$. We may continue this argument to conclude that there are an infinite number of arcs which intersect $r(Y, \mathbf{d}, \mathbf{X}_\delta)$; but that contradicts our previous conclusions. Therefore, Theorem 2.6 must be true. ■

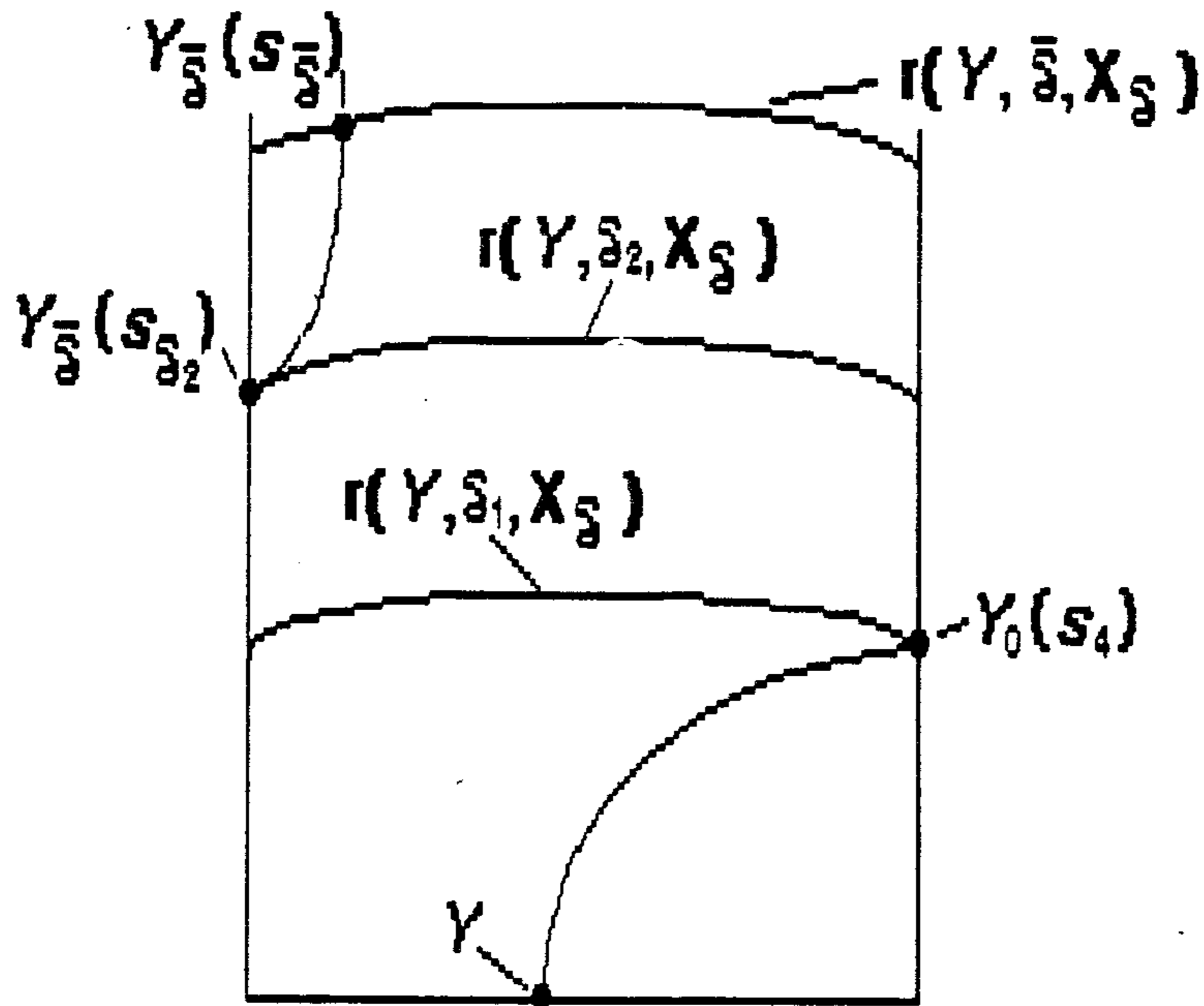


Fig. 3. Illustration of what must occur if convergence in Theorem 2.3 is to more than one arc.

The final theorem in this section assures us that $\bar{\mathbf{X}} \subset \mathbf{X}_\delta$ whenever we choose δ small enough.

THEOREM 2.7. *Suppose Assumptions 2.1, 2.4, and 2.5 hold, and \mathbf{X}_δ , \bar{F}_δ , and \bar{F}'_δ are as in Theorem 2.3. Furthermore, assume H' is of full rank in a neighborhood of Y . Finally, assume that the interval extension \bar{F}'_δ is such that, if $A \in \bar{F}'_\delta(\mathbf{X}_\delta)$, then $A \rightarrow \bar{F}'_\delta(X)$ as $(\mathbf{d}, \mathbf{X}_\delta) \rightarrow (\delta, X)$. Also, assume we solve (2.2) via one of the methods explained in [17]. Then, for \bar{X} sufficiently close to a solution of $\bar{F}_\delta(X) = 0$ and for δ sufficiently small, $\bar{\mathbf{X}} \subset \mathbf{X}_\delta$.*

We note that most common interval extensions obey the assumption on \tilde{F}'_δ in Theorem 2.7. We also note that most common predictor steps B and corrector manifolds r obey Assumptions 2.1, 2.2, 2.4, and 2.5, provided q is chosen as explained below Assumption 2.1.

PROOF OF THEOREM 2.7: The proof will be based on Proposition 5, p. 265 of [17].

By the assumption on the interval extension \tilde{F}'_δ , by the nonsingularity assumptions on H' and $\partial r/\partial X$, by Assumption 2.2, and by the portion of Assumption 2.1 dealing with the range of $\partial r/\partial X$, we may choose a δ_1 such that, if $\delta < \delta_1$, then each $A \in \tilde{F}'_\delta(\mathbf{X}_\delta)$ is nonsingular, so that the interval inverse $\tilde{F}'_\delta(\mathbf{X}_\delta)^I$ (in one of the senses in [17]) exists. Also, let X_δ^* be the solution to $\tilde{F}'_\delta(X) = 0$ which corresponds to the arc through Y . Then, as in the proof of Theorem 2.6, we may take $\delta_0 \leq \delta_1$ so that $X_\delta^* \in \text{int}(\mathbf{X}_\delta)$ for $\bar{\delta} \leq \delta < \delta_0$, where $\text{int}(\mathbf{X}_\delta)$ is the interior of \mathbf{X}_δ .

If $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$, where $\mathbf{x}_i = [a_i, b_i]$, then we define $\rho(\mathbf{X})$ to be the vector $\frac{1}{2}(b_1 - a_1, b_2 - a_2, \dots, b_n - a_n)$, and define $\rho(\mathbf{M})$ similarly, if \mathbf{M} is an interval matrix. Also, we define

$$|\mathbf{X}| = (\max\{|a_1|, |b_1|\}, \max\{|a_2|, |b_2|\}, \dots, \max\{|a_n|, |b_n|\}),$$

and define $|\mathbf{M}|$ similarly. (See [16] for details of these definitions.) Then a consequence of Proposition 5 of [17] is

$$(2.9) \quad \left\| |\bar{\mathbf{X}} - X_\delta^*| \right\|_\infty \leq \sigma \left\| \check{X} - X_\delta^* \right\|_\infty,$$

where

$$\sigma = \left\| \tilde{F}'_\delta(\mathbf{X}_\delta)^I \tilde{F}'_\delta(\mathbf{X}_\delta) \right\|_\infty.$$

On the other hand, since $X_\delta^* \in \text{int}(\mathbf{X}_\delta)$ and since $[0, \delta]$ is compact,

$$\tau = \min_{0 \leq \bar{\delta} \leq \delta} d(X_\delta^*, \partial \mathbf{X}_\delta) > 0,$$

where $d(P, \partial \mathbf{Q})$ is the distance of the point P to the boundary of the set \mathbf{Q} in the infinity norm. By this definition, it follows that

$$\left\| |\bar{\mathbf{X}} - X_\delta^*| \right\|_\infty < \tau \implies \bar{\mathbf{X}} \in \mathbf{X}_\delta,$$

so that it remains to show that, for sufficiently small δ , we have

$$\left\| |\bar{\mathbf{X}} - X_\delta^*| \right\|_\infty < \tau.$$

But (2.9) implies

$$\left\| \bar{\mathbf{X}} - X_{\bar{\delta}}^* \right\|_{\infty} < \tau \quad \text{if} \quad \left\| \check{X} - X_{\bar{\delta}}^* \right\|_{\infty} < \frac{\tau}{\sigma} \quad \text{for every} \quad \bar{\delta} \leq \delta.$$

To complete the proof, we show that we can make $\left\| \check{X} - X_{\bar{\delta}}^* \right\|_{\infty}$ as small as we please by making δ small and by making $\left\| \check{X} - X_{\delta}^* \right\|_{\infty}$ small. But, for δ small enough, we may apply the implicit function theorem on p. 20 of [4] to \bar{F}_{δ} to obtain, for $0 \leq \bar{\delta} \leq \delta$,

$$(2.10) \quad \left\| X_{\bar{\delta}}^* - X_{\delta}^* \right\|_{\infty} = O(\delta).$$

We complete the proof by applying (2.10) in conjunction with the triangle inequality. ■

We conclude this section with some interpretive observations. First, Assumptions 2.1, 2.4, and 2.5 are not restrictive, but can usually be made to hold if the continuation method algorithm is constructed properly. Also, Assumption 2.2 deals with how we construct the step control algorithm, and can be made to hold. Finally, we note that we can obtain an interval extension to any function which is Lipschitz continuous. Hence, we can, in theory, devise a foolproof continuation method for any parameter-dependent system H the components of whose Jacobian matrix are Lipschitz continuous.

3. SPECIFIC ALGORITHMS

In this section, we give details of the algorithms in which we have implemented the ideas in Section 2, while in Section 4, we report numerical experiments based on these algorithms. Our algorithms have several parts, which we single out. In Section 3.1, we describe the construction of r and construction of the function \bar{F}_{δ} and interval Jacobian matrix $\tilde{\mathbf{F}}'_{\delta}$. In Section 3.2, we detail the interval Gauss-Seidel procedure we use to solve (2.2). In Section 3.3, we describe the algorithm we use to adjust δ so that the conditions of Theorem 2.3, Theorem 2.6, and Theorem 2.7 are satisfied. In Section 3.4, we outline the non-interval step control procedure we employ for comparisons. Finally, in Section 3.5, we indicate the three ways which we choose the predictor step B in the experiments.

3.1 The functions \bar{F}_{δ} and $\tilde{\mathbf{F}}'$.

As is indicated above (2.6), we must compute r in order to obtain values for \bar{F}_{δ} and $\tilde{\mathbf{F}}'_{\delta}$. In our experiments, we take r consistent with

(1.3), namely

$$(3.1) \quad r(Y, \delta, P) = Y + \delta B + \sum_{j=0}^n p_j W_j,$$

where $P = (p_1, p_2, \dots, p_n)$ and where $\{W_j\}_{j=1}^n$ is a set of orthogonal vectors such that $W_j \circ B = 0$ for j between 1 and n .

See Section 3.5 for the three ways in which we select B in the experiments.

Once we have computed B , we determine the orthogonal set $\{W_j\}_{j=1}^n$ by computing a QR -factorization of the Jacobian matrix of \tilde{H} defined in (1.2); the first n columns of Q are then vectors W_j which are orthogonal to B .

Once B , $\{W_j\}_{j=1}^n$, and δ (cf. Section 3.3 below) are available, we compute \tilde{F}_δ , \tilde{F}_d , and \tilde{F}_d directly from (3.1) and from the definitions of \tilde{F}_d and \tilde{F}_d above (2.6). We similarly compute $\tilde{F}'_\delta(\mathbf{X})$ and $\tilde{F}'(\tilde{X})$ directly from (3.1) and (2.6), once we have observed that j^{th} column of $\frac{\partial r(Y, \delta, X)}{\partial X}(Y, \mathbf{d}, \mathbf{X}_\delta)$ is simply W_j .

We also similarly compute \tilde{F}_δ and \tilde{F}'

The computations simplify considerably if B is one of the coordinate vectors. In that case, the W_j are also chosen to be coordinate vectors.

3.2 The interval Gauss-Seidel procedure.

The basic procedure for computing enclosures to the system (2.2) is similar to that in [10] and in [11], but is adapted to this context. The purpose of the algorithm in this context is to determine whether a given stepsize δ is acceptable according to the criterion $\bar{\mathbf{X}} \subset \mathbf{X}$ of Theorem 2.3, Theorem 2.6, and Theorem 2.7. We summarize in the following algorithm.

ALGORITHM 3.1.

- (1) *Input* Y , B , $\{W_j\}_{j=1}^n$, and δ .
- (2) *(Get hypothetical bounds for the new point on the arc.)*
 - (a) *Evaluate* $q(\delta) = \delta^{\alpha+1}$, where

$$\alpha = \begin{cases} \log(T)/\log(\delta_m) & \text{if } \log(T)/\log(\delta_m) > -1 \\ -0.9 & \text{if } \log(T)/\log(\delta_m) \leq -1 \end{cases},$$

where T is the maximum allowed tangent of an angle between the predictor step and the tangent to the arc at Y , and δ_m is the minimum predictor stepsize.³

³This sets it up so that the secant between the arc at Y and the point on r at δ_m is assumed to be at most T . In the experiments, we used $T = 50$ and $\delta_m = \sqrt{10\epsilon_m} \doteq 4.7 \times 10^{-8}$, where ϵ_m is our double precision machine epsilon.

- (b) Compute $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ by setting $\mathbf{x}_i = [-q(\delta), q(\delta)]$ for $i = 1, \dots, n$.
- (3) (Obtain the point approximation.)
- (a) If possible, compute an approximation $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$ to the solution of $\tilde{F}_\delta(X) = 0$ by applying the traditional Newton's method to this equation.⁴
- (b) If Newton's method in (a) does not seem to converge, then $\tilde{X} \leftarrow 0$.
- (4) (Check the interval function value.)
- (a) Compute $\tilde{\mathbf{F}}_d(\mathbf{X})$.
- (b) If $0 \notin \tilde{\mathbf{F}}_d(\mathbf{X})$, then return with an indication that the stepsize δ is too large.
- (5) Compute the interval Jacobian matrix $\tilde{\mathbf{F}}'_\delta$.
- (6) (Compute and check the preconditioner matrix.)
- (a) Compute the matrix Y , as in [10], etc.⁵
- (b) If Y is numerically singular, then return with an indication that the stepsize δ is too large.
- (7) Form $\mathbf{G} = (\mathbf{g}_{i,j})_{i,j=1}^n = Y\tilde{\mathbf{F}}'_\delta(\mathbf{X})$ and $\mathbf{K} = (\mathbf{k}_1, \mathbf{k}_2, \dots, \mathbf{k}_n)^T = Y\tilde{\mathbf{F}}_d(\tilde{X})$.
- (8) $INSIDE \leftarrow TRUE$.
- (9) (Do the actual interval Gauss-Seidel method; one pass through each substep represents the method applied to one coordinate.)
Do the following for $i = 1$ to n .
- (a) Compute

$$\bar{\mathbf{x}}_i = \tilde{x}_i - \left[\mathbf{k}_i - \sum_{\substack{j=1 \\ j \neq i}}^n \mathbf{g}_{i,j}(\mathbf{x}_j - \tilde{x}_j) \right] / \mathbf{g}_{i,i}$$

- in extended interval arithmetic.
- (b) If $\bar{\mathbf{x}}_i \cap \mathbf{x}_i = \emptyset$ or if $\bar{\mathbf{x}}_i$ consists of two intervals, then return with an indication that the stepsize δ is too large.
- (c) If $\bar{\mathbf{x}}_i \not\subset \mathbf{x}_i$, then $INSIDE \leftarrow FALSE$.
- (d) Replace \mathbf{x}_i by $\mathbf{x}_i \cap \bar{\mathbf{x}}_i$.
- (10) If $INSIDE = TRUE$, then return with an indication that the stepsize δ is acceptable.

⁴This value will actually be used as the corrected point Z in Step 2(d) of Algorithm 1.1, provided that the stepsize has been verified to be adequate.

⁵In the experiments, we chose Y to be the inverse of the midpoint matrix of $\tilde{\mathbf{F}}'_\delta(\mathbf{X})$; different choices could significantly affect this algorithm. Research on the choice of Y appears in [8] and [9].

- (11) Compute the ratio of volumes of the old \mathbf{X} (before execution of Step 9) to the present \mathbf{X} .⁶
- (12) If the volume ratio in Step 11 is too near to 1, or if $0 \notin \mathbf{X}$, then return with an indication that the stepsize δ is not acceptable.
- (13) If this step has been reached more than MAXITR times, then return with an indication that the stepsize δ is not acceptable. Otherwise, return to Step 8.

3.3 The step adjustment procedure for the interval criterion.

Algorithm 3.1 gives a criterion for acceptance or rejection of a predictor stepsize δ . Here, we give one possible scheme for adjusting δ based on the outcome of Algorithm 3.1. The *ansatz* for this scheme is the assumption that it is always advantageous in the overall algorithm to use the maximum predictor step δ which Algorithm 3.1 finds acceptable, regardless of the expense of determining that δ . This assumption may not be reasonable in practice, but in initial studies it should give us an indication of the limits of the method.

ALGORITHM 3.2.

- (1) Input an initial δ .
- (2) Execute Algorithm 3.1 initially.
- (3) If the initial stepsize is acceptable, then do the following.
 - (a) Compute MAXIT based on the present δ and the maximum allowable stepsize. (See Step (c) below.)
 - (b) $\delta \leftarrow 2\delta$.
 - (c) If this step has been reached more than MAXIT times, then return with an error message.
 - (d) Execute Algorithm 3.1. If the present δ is acceptable, then return to Step (b); otherwise, take $\delta/2$ as the present predictor stepsize.
- (4) If the initial stepsize is not acceptable, then do the following.
 - (a) Compute MAXIT based on the present δ and the minimum allowable stepsize.
 - (b) $\delta \leftarrow \delta/2$.
 - (c) If this step has been reached more than MAXIT times, then return with an error message.
 - (d) Execute Algorithm 3.1. If the present δ is not acceptable, then return to Step (b); otherwise, take δ as the present predictor stepsize.

⁶We actually only compute the volumes of the projections into the subspace of \mathbf{R}^n containing those coordinate vectors whose corresponding components of the old \mathbf{X} have acceptably large widths.

3.4 The non-interval step control.

It is difficult to computationally compare an interval step control with a non-interval step control, since non-interval step controls involve heuristics and parameters which can be used to tune the step control to specific problems. For some values of these tuning parameters, the heuristic step control will cause Algorithm 1.1 to follow many arcs safely but inefficiently, while for other values, Algorithm 1.1 will follow some arcs efficiently, but jump across other arcs. In contrast, the only issue in our interval step control is efficiency.

Nonetheless, we have chosen the following simple but effective step control algorithm for comparison purposes.

ALGORITHM 3.3.

- (1) *(Input)*
 - (a) *Input the present predictor direction B , the previous predictor direction \bar{B} , the upper target dot product C_{\max} , and the lower target dot product C_{\min} .*
 - (b) *Input the number of times I that Step 2(d) of Algorithm 1.1 has been executed before CS was satisfied. Also input the upper target value I_{\max} and the lower target value I_{\min} for I .*
- (2) *If $B \circ \bar{B} \geq C_{\max}$ and $I \leq I_{\min}$ then*
 - (a) $\delta \leftarrow 2\delta$.
 - (b) *If $\delta > \delta_{\max}$ then $\delta \leftarrow \delta_{\max}$.*
- (3) *If $B \circ \bar{B} < C_{\min}$ or $I > I_{\max}$ then*
 - (a) $\delta \leftarrow \delta/2$.
 - (b) *If $\delta < \delta_{\min}$ then $\delta \leftarrow \delta_{\min}$.*

The above algorithm is used when B is taken to be the tangent to the arc. In the cases where we use the elevator predictor or the simplified Rheinboldt/den Heijer local coordinate scheme (see Section 3.5), we delete all references to C_{\min} , C_{\max} , and $B \circ \bar{B}$ (and delete the corresponding conditions in Steps 2(a) and 3(a)).

In our experiments, we took $I_{\min} = 3$ and $I_{\max} = 6$. When applicable, we also took $C_{\max} = .9$ and $C_{\min} = .1$.

Since Algorithm 3.3 is heuristic, it is possible that the corrector iteration (Step 2(d) of Algorithm 1.1; Newton's method applied to (1.2)) will fail. In such instances, we must adjust δ , then restart the corrector iteration. The following algorithm (unnecessary for the interval step control) checks for divergence at the beginning of each corrector step.

ALGORITHM 3.4. (*Adjustment of δ when corrector iteration diverges*)

- (1) *(Input)*

- (a) Input the magnification factor M_Z by which the Euclidean norm of the corrector step can increase from one corrector iteration to the next.
 - (b) Input the magnification factor M_H by which the Euclidean norm of $\tilde{H}(Z)$ can increase from one corrector iteration to the next.
 - (c) Input the maximum allowable number M_{it} of corrector iterations.
- (2) Let \bar{s} denote the Euclidean norm of the present corrector step, let s denote the Euclidean norm of the previous corrector step, let \bar{u} denote the Euclidean norm of the present $\tilde{H}(Z)$, and let u denote the Euclidean norm of the previous $\tilde{H}(Z)$. Also let I denote the number of corrector steps which have been executed to this point in the present predictor step. If
- (i) $\bar{s} \geq M_Z s$, or
 - (ii) $\bar{u} \geq M_H u$, or
 - (iii) $I > M_{it}$,⁷
- then
- (a) $\delta \leftarrow \delta/2$;
 - (b) Compute a new predicted point Z_0 in Step 1(b) of Algorithm 1.1 using the new δ .
 - (c) Restart the corrector iteration with $I \leftarrow 0$.

In the experiments of the next section, we chose $M_Z = M_H = 10.0$ and $M_{it} = 15$.

We wish to emphasize here that in the non-interval case, the corrector iteration is executed by applying Newton's method to (1.2), whereas, in the interval case, the corrector iteration is executed in Step 2 of Algorithm 3.1, and consists of applying Newton's method to (1.3).

3.5 The predictor direction B .

In the experiments, we select B to be one of the following three predictor steps:

- (1) the tangent to the arc (i.e. the appropriately oriented vector B such that $H'(Y)B = 0$);
- (2) a coordinate vector chosen via a simplified version of the scheme in [19].
- (3) the "elevator" predictor, i.e. the coordinate vector $e_{n+1} = (0, 0, \dots, 0, 1)^T$.

⁷We may also check $\|\tilde{H}(Z)\|$ and the condition number of the Jacobian matrix of $\tilde{H}(Z)$ in order to prevent overflow when we compute the corrector step. However, that condition seldom occurs.

There are a number of satisfactory ways to obtain the tangent vector B as in (1); we computed it from the QR-decomposition for the Jacobian matrix for H . The elevator predictor in (3) requires no computation. The following algorithm, used in our experiments, is a simplified version of that in [19].

ALGORITHM 3.5 (CHOICE OF COORDINATE VECTOR).

- (1) Compute the tangent vector $T = (t_1, t_2, \dots, t_n)$ to the arc at Y .
- (2) Possibly change the sign of T so that its dot product with the previous predictor step is positive.
- (3) Take as predictor step that coordinate vector $B = e_l$ such that $|t_l| = \max_{1 \leq i \leq n} |t_i|$.
- (4) Possibly change the sign of B so that $B \circ T > 0$

4. THE NUMERICAL EXPERIMENTS

We describe here the test problems and experimental results.

Our goal here is not exhaustive comparison, but an initial exploration of the interval step control. With this in mind, we chose the following two test functions. We use the notation

$$H(Y) = (h_1(Y), h_2(Y), \dots, h_n(Y))^T,$$

and $Y = (y_1, y_2, \dots, y_{n+1})^T$.

TEST FUNCTION 1. *Brown's almost linear function.*

$$h_i(Y) = \begin{cases} y_i + y_{n+1} \left[\sum_{j=1}^n y_j - (n+1) \right] & \text{for } 1 \leq i \leq n-1 \\ (1 - y_{n+1})y_n + y_{n+1} \left[\prod_{i=1}^n y_i - 1 \right] & \text{for } i = n \end{cases}$$

The coordinate y_{n+1} increases monotonically for this function along a single path from $y_{n+1} = 0$ to $y_{n+1} = 1$. However, the Jacobian matrix becomes ill-conditioned at $y_{n+1} = 1$, especially for large n .

TEST FUNCTION 2. *The Layne Watson exponential cosine function.*

$$h_i(Y) = y_i - y_{n+1} \exp \left(\cos \left(i \sum_{j=1}^n y_j \right) \right), 1 \leq i \leq n.$$

This function has a path from $Y = 0$ to $y_{n+1} = 1$. However, this path has sharp changes in curvature and numerous turning points in y_{n+1} , especially for n greater than about five or six.

In the tests, we started the algorithm at $Y = 0$ in the direction of increasing y_{n+1} , and completed the computation when y_{n+1} attained or crossed the value 1.

In the tests, we used the simple, reasonably portable Fortran-77 routines in [11] for the interval arithmetic; we specially wrote interval routines for the sine, cosine, and exponential which use that package. We computed analytic Jacobian matrices manually, but we interfaced our package with the Augment precompiler (*cf.* [3]) to ease the burden of programming the interval operations.

Behavior of the non-interval step control algorithm (Algorithm 3.3) depends strongly on the tolerance with which we follow the arcs. In particular, we select ν_1 and ν_2 such that the corrector stopping criterion \mathcal{CS} is satisfied if either the norm of the most recent corrector step is less than ν_1 or the norm of the most recent value of \tilde{H} is less than ν_2 . In the experiments, we took $\nu_1 = \nu_2 = 10^{-5}$.⁸

Additionally, an initial predictor stepsize δ_0 must be heuristically chosen for the non-interval step control. We selected $\delta_0 = 10^{-3}$ in the experiments. (This value is also used initially in Algorithm 3.1, but its size has little effect there on subsequent values of δ .)

Because we did not wish at this point to be concerned with optimization of the interval arithmetic for our particular machine and test problems, because the philosophy behind Algorithm 3.2 is to maximize the predictor steplength at each iteration, regardless of cost, and because of technical questions concerning CPU time, our main comparisons do not involve CPU time. (However, we give *some* CPU times at the end of this section for a rough idea.) In our tables, we report the total number of predictor iterations N_{pred} to traverse the arc from $y_{n+1} = 0$ to $y_{n+1} = 1$, the minimum attained predictor steplength δ_1 , the maximum attained predictor steplength δ_2 , the average predictor steplength δ_a , the minimum number of corrector iterations C_1 , the maximum number of corrector iterations C_2 , and the average number of corrector iterations C_a .

⁸The interval step control algorithm in theory also depends on these tolerances, since we use Newton's method to obtain a point approximation to the next point on the arc in Step 3(a) of Algorithm 3.1. However, Algorithm 3.1 is in practice insensitive to the accuracy of this point approximation.

Table 1(a). B is the tangent vector.
Brown's Almost Linear Function

Problem	N_{pred}	δ_1	δ_2	δ_a	C_1	C_2	C_a
$n = 2$							
Interval	76	6.2×10^{-3}	2.0×10^{-1}	2.4×10^{-2}	1	2	1.41
Non-int.	117	4.0×10^{-3}	1.6×10^{-2}	1.6×10^{-2}	1	2	1.96
$n = 4$							
Interval	496	7.7×10^{-4}	4.9×10^{-2}	4.9×10^{-3}	1	2	1.05
Non-int.	156	4.0×10^{-3}	1.6×10^{-2}	1.6×10^{-2}	1	2	1.97
$n = 5$							
Interval	857	3.9×10^{-4}	4.9×10^{-2}	3.2×10^{-3}	1	1	1.00
Non-int.	172	4.0×10^{-3}	1.6×10^{-2}	1.6×10^{-2}	1	2	1.97
$n = 10$							
Interval	4854	2.4×10^{-5}	2.5×10^{-2}	7.7×10^{-4}	1	1	1.00
Non-int.	120	4.0×10^{-3}	3.2×10^{-2}	3.1×10^{-2}	1	2	1.95

Table 1(b). B is the tangent vector.
Watson's Exponential Cosine Function

Problem	N_{pred}	δ_1	δ_2	δ_a	C_1	C_2	C_a
$n = 2$							
Interval	56	2.5×10^{-2}	4.9×10^{-2}	2.9×10^{-2}	1	2	1.84
Non-int.	55	4.0×10^{-3}	3.2×10^{-2}	3.0×10^{-2}	1	2	1.89
$n = 4$							
Interval	7708	1.9×10^{-4}	1.2×10^{-2}	8.5×10^{-4}	1	1	1.00
Non-int.	460	4.0×10^{-3}	3.2×10^{-2}	1.4×10^{-2}	1	3	1.98
$n = 5$							
Interval	71867	4.8×10^{-5}	6.1×10^{-3}	2.1×10^{-4}	1	1	1.00
Non-int.	915	2.0×10^{-3}	6.4×10^{-2}	1.6×10^{-2}	1	3	1.98

Table 2(a). B is by Algorithm 3.5.
Brown's Almost Linear Function

Problem	N_{pred}	δ_1	δ_2	δ_a	C_1	C_2	C_a
$n = 2$							
Interval	21	2.5×10^{-2}	2.0×10^{-1}	6.4×10^{-2}	1	3	1.95
Non-int.	160	4.0×10^{-3}	1.6×10^{-2}	8.3×10^{-3}	1	2	1.97
$n = 4$							
Interval	94	1.5×10^{-3}	9.9×10^{-2}	1.6×10^{-2}	1	3	1.66
Non-int.	316	4.0×10^{-3}	1.6×10^{-2}	4.6×10^{-3}	1	2	1.98
$n = 5$							
Interval	155	7.7×10^{-4}	9.9×10^{-2}	9.7×10^{-3}	1	3	1.59
Non-int.	330	4.0×10^{-3}	1.6×10^{-2}	4.5×10^{-3}	1	2	1.99
$n = 10$							
Interval	872	2.4×10^{-5}	9.9×10^{-2}	1.9×10^{-3}	1	3	1.30
Non-int.	384	4.0×10^{-3}	1.6×10^{-2}	4.3×10^{-3}	1	2	1.98

Table 2(b). B is by Algorithm 3.5.
Watson's Exponential Cosine Function

Problem	N_{pred}	δ_1	δ_2	δ_a	C_1	C_2	C_a
$n = 2$							
Interval	24	4.9×10^{-2}	9.9×10^{-2}	5.6×10^{-2}	1	3	2.58
Non-int.	121	4.0×10^{-3}	1.6×10^{-2}	1.1×10^{-2}	1	2	1.96
$n = 4$							
Interval	2152	7.7×10^{-4}	4.9×10^{-2}	2.4×10^{-3}	1	3	1.49
Non-int.	411	4.0×10^{-3}	3.2×10^{-2}	1.3×10^{-2}	1	2	1.98
$n = 5$							
Interval	17766	1.9×10^{-4}	2.5×10^{-2}	6.3×10^{-4}	1	2	1.00
Non-int.	2768	2.0×10^{-3}	3.2×10^{-2}	4.1×10^{-3}	1	3	2.00

The computation time was excessive for the interval step control and Watson's function for higher values of n .

The interval step control algorithm does better *vis à vis* Algorithm 3.3 when the predictor direction B is chosen to be a coordinate vector. This may be due to less widening of intervals during the computation when one of the coordinate intervals has width zero during evaluation of H .

Even better results could probably be obtained if we took additional care with the form in which we program H . Also, the interval step control should generally work relatively better for H which are algebraically simple, such as low degree polynomials or, perhaps, systems arising from low-order discretizations of partial differential equations.

We note that the degree of the last component of Brown's function increases with the dimension. This could account for the more rapid increase with n in the number of predictor steps required when the interval step control algorithm is used. Also, our additional work on the interval Newton algorithm, as in [8] and [9], would help on systems like Brown's almost linear function.

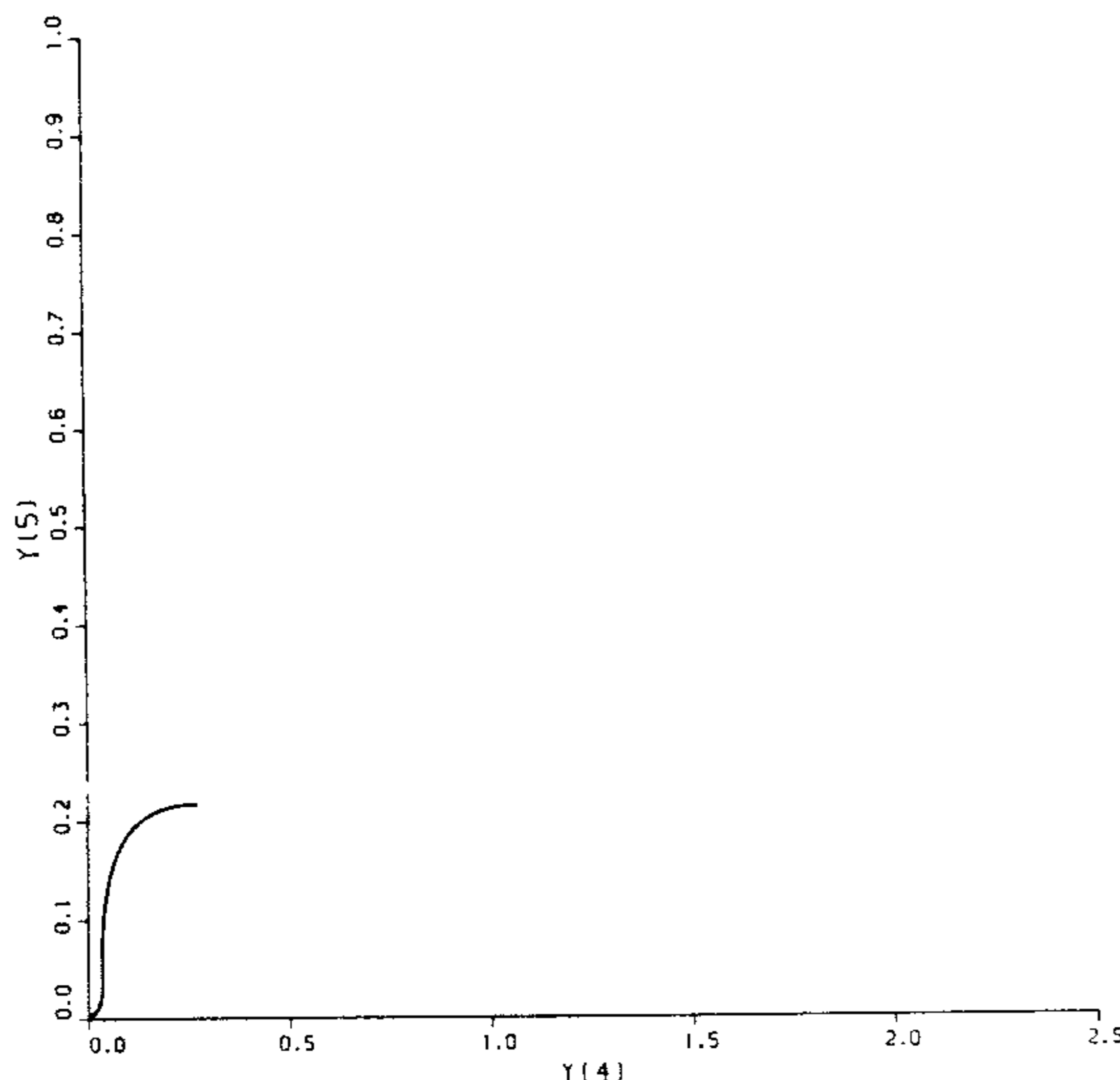


Fig. 4. Iterates for the Watson exponential cosine function with $n = 5$, the elevator predictor, and interval step control.

Except for those H for which theory justifies its use, the elevator predictor (i.e. always using e_{n+1} for B) generally gives inferior results *vis à vis* the predictor chosen via Algorithm 3.5. We used it mainly to test the interval step control, since, if it is properly programmed, it will never allow a step which jumps beyond a turning point. Our experiments confirmed this. In fact, for Watson's function the interval step control stalled at a turning point, as it should, whereas when Algorithm 3.3 and Algorithm 3.4 were used, the iterates jumped across large loops of the

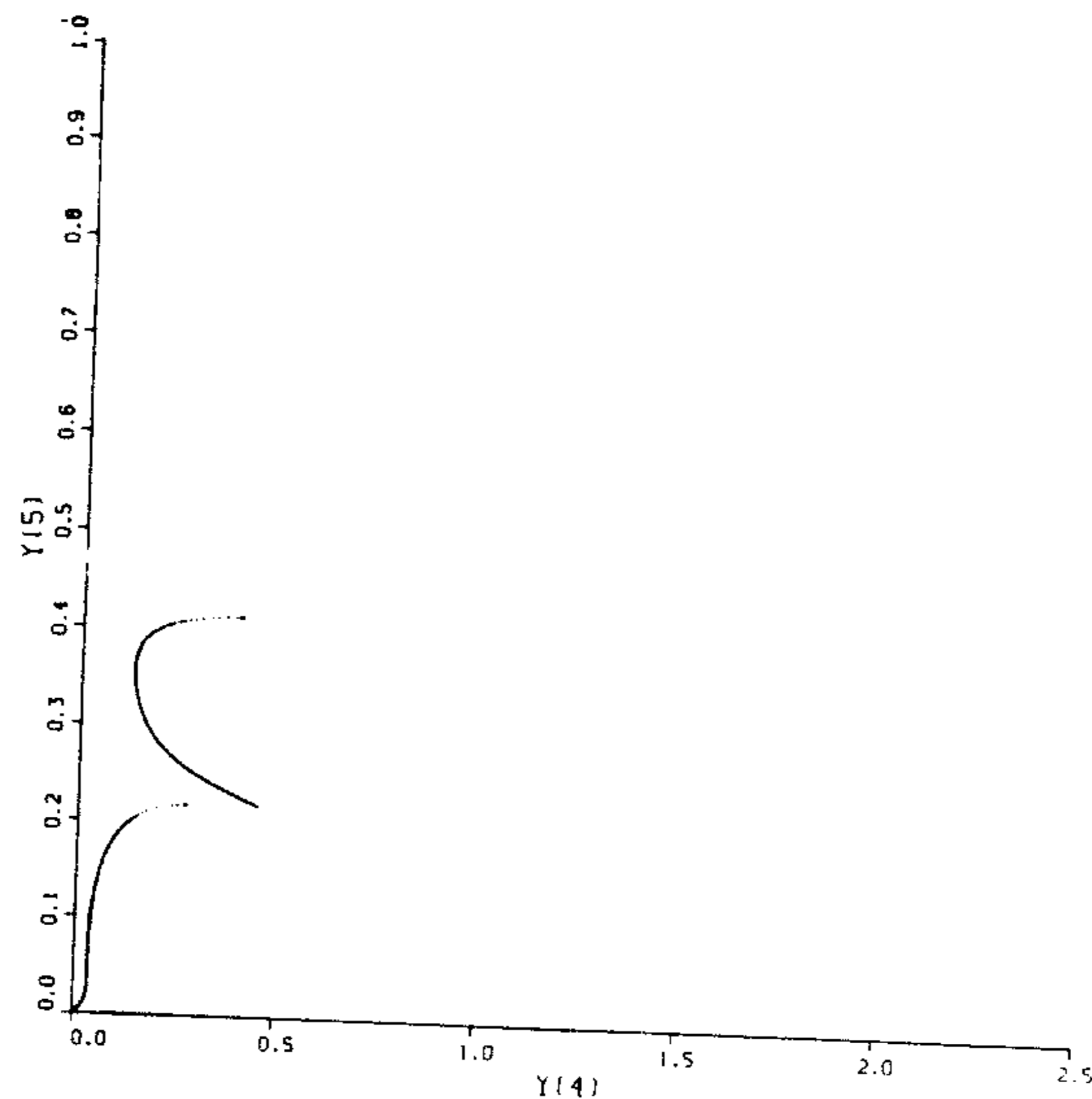


Fig. 5. Iterates for the Watson exponential cosine function with $n = 5$, the elevator predictor, and non-interval step control.

curve without indicating trouble. Figures 4, 5, and 6 illustrate this.

The CPU time on an IBM 3090 for Brown's function with $n = 5$, using Algorithm 3.5 for B , was approximately 16 seconds for the interval step control and 4 seconds for the non-interval step control.⁹

REFERENCES

1. Alefeld, G., and Herzberger, J., "Introduction to Interval Computations," Academic Press, New York, 1983.
2. Allgower, E. L., and Georg, K., *Simplicial and Continuation Methods for Approximating Fixed Points and Solutions to Systems of Equations*, SIAM Rev. **22** 1 (1980), 28-85.
3. Crary, F., *The AUGMENT precompiler*, Mathematics Research Center Technical Summary Report no. 1470, The University of Wisconsin at Madison.
4. Garcia, C. B., and Zangwill, W. I., "Pathways to Solutions, Fixed Points, and Equilibria," Prentice-Hall, Englewood Cliffs, NJ, 1981.
5. Georg, K., *A Note on Stepsize Control for Numerical Curve Following*, in "Homotopy Methods and Global Convergence," Nato Conference Series II: Systems Science 13, Plenum, New York and London, 1983, pp. 145-154.

⁹Our implementation of interval arithmetic is probably a factor of between 100 and 200 slower than the fastest possible implementation. We could not gain this much in the non-interval step control with a more efficient implementation.

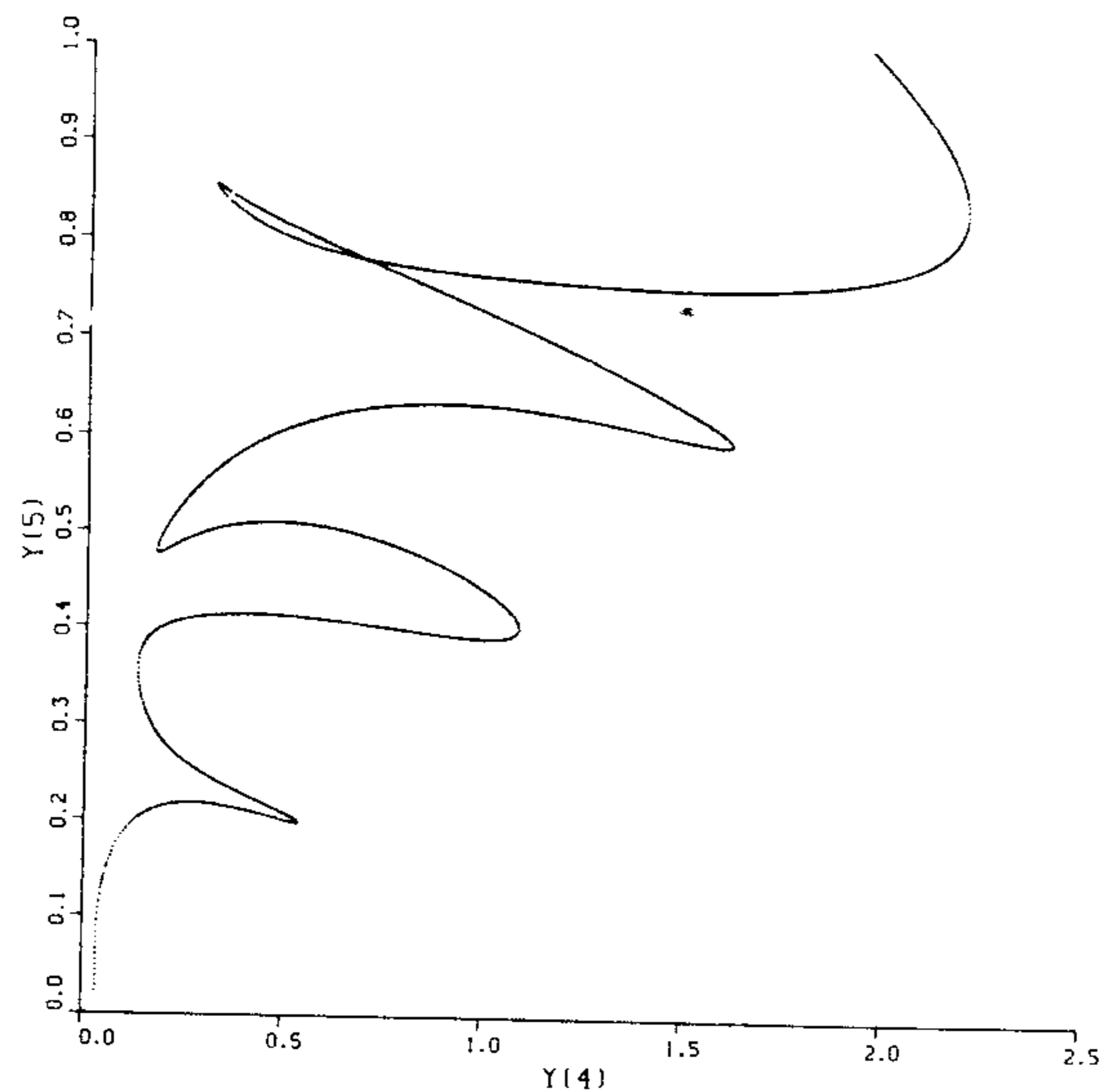


Fig. 6. Iterates for the Watson exponential cosine function with $n = 5$, the B as in Algorithm 3.5 and interval step control.

6. Hackl, J., Wacker, H. J., and Zulehner, W., *An Efficient Step Size Control for Continuation Methods*, BIT **20** (1980), 475-485.
7. Heijer, C. den, and Rheinboldt, W. C., *On Steplength Algorithms for a Class of Continuation Methods*, SIAM J. Numer. Anal **18** 5 (1981), 925-948.
8. Hu, C.-Y., and Kearfott, R. B., *An Optimal Preconditioner for the Interval Gauss-Seidel Method*, submitted to Math. Comp..
9. Kearfott, R. B., *Preconditioners for the Interval Gauss-Seidel Method*, submitted to SIAM J. Numer. Anal..
10. Kearfott, R. B., *Interval Interval Arithmetic Techniques in the Computational Solution of Nonlinear Systems of Equations: Introduction, Examples, and Comparisons*, in "Proceedings of the 1988 AMS-SIAM Summer Seminar in Applied Mathematics," (Colorado State University, July 18-29, 1988) (to appear).
11. Kearfott, R. B., and Novoa, M., *A Program for Generalized Bisection*, submitted to ACM Trans. Math. Software.
12. Keller, H. B., *Continuation Methods in Computational Fluid Dynamics*, in "Numerical and Physical Aspects of Aerodynamic Flows," Springer Verlag, New York, 1982, pp. 3-13.
13. Moore, R. E., "Methods and Applications of Interval Analysis," SIAM, Philadelphia, 1979.
14. Moore, R. E., ed., "Reliability in Computing: The Role of Interval Methods in Scientific Computing," Academic Press, New York, etc., 1988.
15. Morgan, A. P., "Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems," Prentice-Hall, Englewood Cliffs, NJ, 1987.

16. Neumaier, A., *New Techniques of the Analysis of Linear Interval Equations*, Linear Algebra and its Applications **58** (1984), 273-325.
17. Neumaier, A., *Interval Iteration for Zeros of Systems of Equations*, BIT **25** 1 (1985), 256-273.
18. Neumaier, A., *The enclosure of solutions of parameter-dependent systems of equations*, in "Reliability in Computing: The Role of Interval Methods in Scientific Computing," Academic Press, New York, 1988, pp. 269-286.
19. Rheinboldt, W. C., and Burkardt, J. V., *A Locally Parametrized Continuation Process*, ACM Trans. Math. Software **9** 2 (1983), 215-235.
20. Schwetlick, H., *On the Choice of Steplength in Path Following Methods*, Z. Angew. Math. Mech. **64** 9 (1984), 391-396.