Interval Computations
No 4, 1993

Boundary-Based Interval Newton's Method

L. Simcik and P. Linz

The boundary based method for approximating solutions to nonlinear systems of equations has a number of advantages over midpoint based algorithms such as Krawczyk's method and the Hansen-Sengupta method. Our research shows that for a certain class of problems the boundary-based method considerably reduces the need for bisection, which is a major source of difficulty for midpoint based methods.

Интервальный метод Ньютона, основанный на границе

Л. Симчик, П. Линц

Метод аппроксимации решений нелинейных систем уравнений, основанный на границе, имеет многие преимущества перед методами, основанными на средней точке, такими как алгоритм Кравчика и метод Хансена-Сенгупты. Наши исследования показывают, что для некоторого класса задач метод, основанный на границе, значительно сокращает число необходимых делений пополам, которые являются основным источником трудностей для методов, основанных на средней точке.

1 Introduction

1.1 Problem statement

We consider an n-dimensional nonlinear system:

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned} \tag{1}$$

where $f : \mathbf{R}^n \mapsto \mathbf{R}^n$ has an interval extension $F : \mathbf{I}^n \mapsto \mathbf{I}^n$, with the corresponding interval Jacobian matrix $J(X) : \mathbf{I}^n \mapsto \mathbf{I}^{n \times n}$. Furthermore, we restrict $f(x)$ so that $f \in \mathcal{C}^1$. Using self-validating methods, the basic problem is to bound the solution(s) to (1) starting with an arbitrary initial domain $X^0 \in \mathbf{I}^n$.

1.2 Krawczyk's method

For the purpose of illustration, we will use a simplified version of Krawczyk's method leaving out some details. We have the general form of the iteration:

$$\begin{aligned} X^{k+1} &= X^k \cap K(X^k) \\ K(X) &= m(X) - Yf\big(m(X)\big) + \big(I - YJ(X)\big)\big(X - m(X)\big). \end{aligned} \tag{2}$$

If X^0 is a "safe" starting region (for precise formulations, see, e.g., [5]) around a unique solution, then Krawczyk's method will converge quadratically. Most Newton-like methods behave in this manner, but unfortunately these methods lack a general theory that would allow a user to compute *a priori* a safe region X^0. For this reason, these schemes often rely upon some auxiliary self-validating globally convergent bounding method that typically has large storage requirements and a lower rate of convergence.

In (2), Y represents either an inverse Jacobian matrix, or its approximation. When $J(X)$ is singular, a bounding method (as mentioned above) must be used to split the domain into several sub-domain (thus splitting the problem into many sub-problems), and eliminate regions until a safe starting region is found.

There are many variations of (2) that use the trade-off between the computational complexity of a single iteration step and convergence rate.

2 Boundary based projection

Before describing the general method, the one-dimensional example will provide a good introduction to the concepts to follow.

2.1 One-dimensional example of a boundary-based method

Consider (1) with $n = 1$ and a simple solution in $X^0 = [a, b]$. Let's denote $F'(X^k) = \left[m_1^{(k)}, m_2^{(k)}\right]$. Without loss of generality, let $f(x)$ be strictly positive at the endpoints of X^0 (other cases are similar). Then

$$\begin{array}{rcl} \underline{X}^{k+1} & = & \underline{X}^k - F\left(\underline{X}^k\right)/m_1^{(k)} \\ \overline{X}^{k+1} & = & \overline{X}^k - F\left(\overline{X}^k\right)/m_2^{(k)} \\ X^{k+1} & = & \left[\underline{X}^{k+1}, \overline{X}^{k+1}\right]. \end{array} \tag{3}$$

This method converges quadratically to the solution in $[a, b]$ despite the possibility that $f'(\gamma) = 0$ for some $\gamma \in [a, b]$ such that $f(\gamma) \neq 0$ [6].

Consider the following equation:

$$f(x) = x^2\left(\frac{1}{3}x^2 + \sqrt{2}\sin x\right) - \sqrt{3}/19 \tag{4}$$

which has a simple solution in $[.1, 1]$. When (3) is applied with $X^0 = [.1, 1]$, the result is as follows:

X^k
$[.100000000000, 1.00000000000]$
$[.117520751584, .720303378869]$
$[.152064870794, .532597677896]$
$[.215187624078, .430345925997]$
$[\underline{.3}03082397292, \underline{.39}6393057590]$
$[\underline{.3}71179468697, \underline{.392}459492178]$
$[\underline{.39}1177285026, \underline{.39237971}8777]$
$[\underline{.3923}75589063, \underline{.3923795071}68]$
$[\underline{.392379507}094, \underline{.39237950713}8]$
$[\underline{.39237950713}5, \underline{.39237950713}8]$

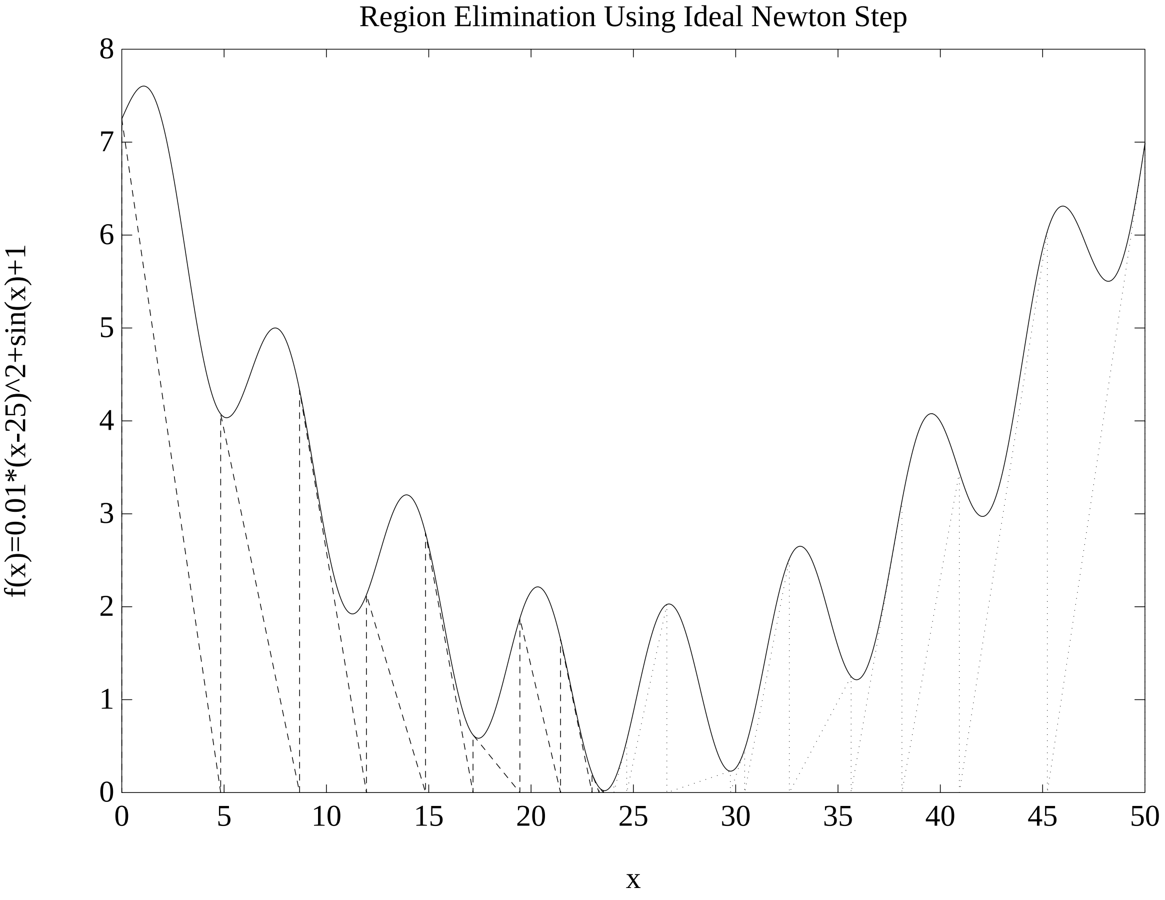

Figure 1: Region elimination

which bounds the real solution.

If there were two distinct solutions in X^0 then the method would converge to the convex hull containing both of them. By splitting the resulting interval and applying the method again to each piece, the solutions can be isolated.

Now consider the problem

$$f(x) \;=\; \frac{1}{100}(x^2 - 50x + 625) + \sin x + 1 \qquad (5)$$

which has no real solutions on $[0, 50]$. When (3) is applied to this problem, we get region elimination depicted in Figure 1. Figure 1 does not show the details of the region elimination at the function's global minimum, but the algorithm did eliminate the entire domain piece-by-piece (on the grounds that none of the eliminated regions could possibly contain a solution of the equation $f(x) = 0$).

Experiments described in [6] showed that in general, for functions f of one variable, the number of iterations required for region elimination is $\mathcal{O}\big(\eta\ \log(\frac{1}{|\alpha|})\big)$, with η being the number of extrema over $[a, b]$, and α be the height of the extrema closest to the x-axis.

Using (3) means using a pessimistic bound on $f'(x)$ near the endpoints, unless $f(x)$ is monotonic over all of X^0. The monotonicity of the inclusion $F'(X)$ implies that for all $X \subset X^0$ we have $F'(X) \subset F'(X^0)$. Thus using some X at the boundary of X^0 will provide a sharper bound on the range of slopes near that particular boundary. Since this is a self-validating method, for a given choice of $X \subset X^0$ we can at most eliminate X as a region with no solutions. In [6], we proved that if a region X^k contains a solution, then at each boundary, there exists the largest interval X that can be reliably eliminated in this manner. These largest intervals X at the left and right endpoints are called the *ideal steps*. They represent the largest portion of the region that can be eliminated at each end during one iteration if we use the boundary-based method.

The program creating the data for Figure 1 used the ideal step boundary-based interval method, finding the ideal step at the left and right side of the interval during each iteration. The ideal steps can be found through a binary search or estimated by using an interval Taylor expansion of $F'(X)$.

2.2 Algorithm description

The principal difference between an n-dimensional problem and the one-dimensional case is that in order to eliminate a region in an n-dimensional space, an algorithm must find a boundary in $\mathbf{I}^n$ where $f(x)$ is guaranteed to be non-zero.

The n-dimensional problem takes the form of (1) with an n-dimensional rectangle X^0. We define a *face* of an n-rectangle by fixing the value of one variable (say, x_j) either at its upper bound or at its lower bound (in both cases, x_j will thus be a constant). Let us denote this face by $X_{\mathcal{F}}$. Thus, for each region X^k, there are $2n$ faces for which we have to evaluate $F_i(X_{\mathcal{F}})$, $i = 1 \ldots n$.

Note that we do not have to apply all n functions F_i to all the faces. E.g., if for some i, the expression for F_i does not contain x_j at all, then we do not need to apply F_i to the corresponding face. Indeed, in this case, the range $F_i(X_{\mathcal{F}})$ of F_i for this face coincides with the range $F_i(X^k)$ of F_i for the entire region X^k, and thus (unless the initial problem has no solutions at all) this range does contain 0.

If after searching over all the faces, we find i such that $0 \notin F_i(X_{\mathcal{F}})$, then we can eliminate a region that is adjacent to this face by using $\frac{\partial F_i}{\partial X_j}$ (i.e., by using the corresponding entry in $J(X)$). This partial derivative evaluated over the entire region can be used to bound the behavior of $F(X)$ in the X_j direction. Thus, in the same manner as the one-dimensional case, we can eliminate a region that is adjacent to the boundaries.

Similar to the one-dimensional case, we can now prove the existence of an ideal step at each face. To find an ideal step, we can either apply a binary search, or we can use an interval Taylor expansion. Since a region is eliminated in only one dimension at a time, we need only a one-dimensional expansion: e.g., if we restrict ourselves to quadratic terms only, then we only need to compute $\frac{\partial^2 F_i}{\partial X_j^2}$, and the desired estimate for an ideal step follows from solving a quadratic equation.

If $0 \notin F(X_{\mathcal{F}})$ at some face $X_{\mathcal{F}}$, and there is no ideal step from this face, this means that the entire domain has no solutions (the proof of this fact is rather simple). This fact demonstrates that the boundary-based method with the ideal step is inherently a non-existence test.

Algorithm pseudo-code:
Start with n-dimensional vector X^0.

1. Create a stack of vectors denoting each face of X^k, say $X_{\mathcal{F}_p}$, $p = 1{:}2n$.

2. For each entry of the interval Jacobian matrix that is not identically zero, check the corresponding two faces for $0 \notin F_i(X_{\mathcal{F}_p})$, and if this condition is true, raise a flag and link this face with the corresponding Jacobian entry. If no faces are “flagged” during this check, then the region X^k must be subdivided before any region elimination can begin (a prudent choice of subdivision may be all that is necessary, rather than a time-consuming complete bisection in each dimension).

3. At each face $x_j = \text{const}$ that was flagged and linked to a Jacobian entry, a region can be eliminated in the j^{th} dimension using the i^{th} function F_i analogously to the way it is eliminated in the one dimen-

sional case. The ideal step can be found through binary search or estimated using $\frac{\partial^2 F_i}{\partial X_j^2}$.

4. Update X^k to take into consideration the region elimination accomplished at each flagged face, and increment k.
5. Repeat steps 1–4 until a convergence criterion is met.

Steps 2 and 3 above can be implemented in a parallel processing environment, since each check is an independent task; similarly, region eliminations at different faces can be done in parallel.

2.3 Two-dimensional example

Consider the following nonlinear system:

$$\begin{aligned} g(x,y) &= x^2 + 6x + y^2 - 6y + 17, \\ h(x,y) &= y^2 - 6y - x^2 - 6x - y\sin 2x + 3\sin 2x + \frac{1}{4}\sin^2 2x. \end{aligned} \tag{6}$$

All solutions to (6) lie in

$$\{(x,y) : -10 \le x \le 10, -10 \le y \le 10\}.$$

Figure 2 shows the contour lines of $g(x,y) = 0$ (circle) and $h(x,y) = 0$ (wobbly cross). The dashed rectangles outline the outer convergence of the algorithm from $X^0 = \big([-10,10], [-10,10]\big)$ to the minimal rectangular hull containing all four solutions. In this example, *the interval Jacobian matrix is singular* over most of the region except for the squares that contain each solution and whose width is approximately equal to $\frac{1}{5}$. The next step is to perform one bisection in each dimension to create four new problems. To each of the resulting domains, we then apply the same boundary-based algorithm. Figure 3 shows inner convergence of the algorithm to each solution.

3 Strategies for using boundary-based methods

When a region is eliminated at a face $X_{\mathcal{F}}$ the question arises concerning whether or not a larger region could have been eliminated if we used a

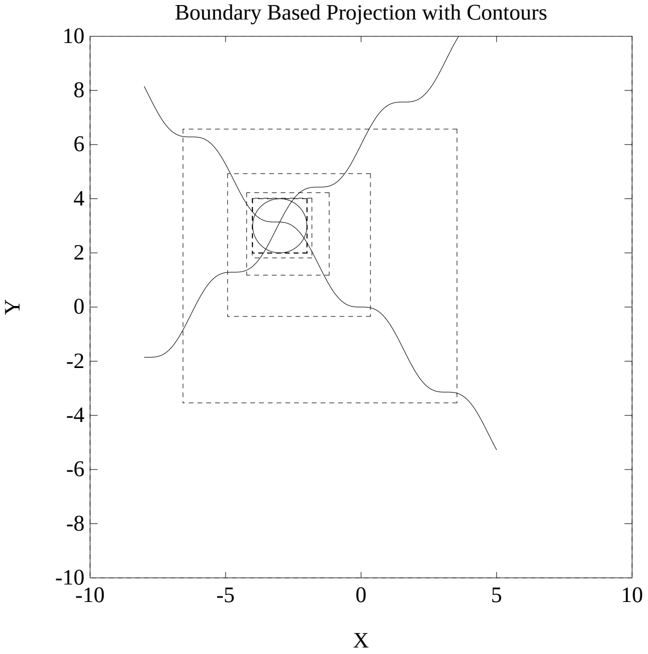

Figure 2: Contours and outer convergence

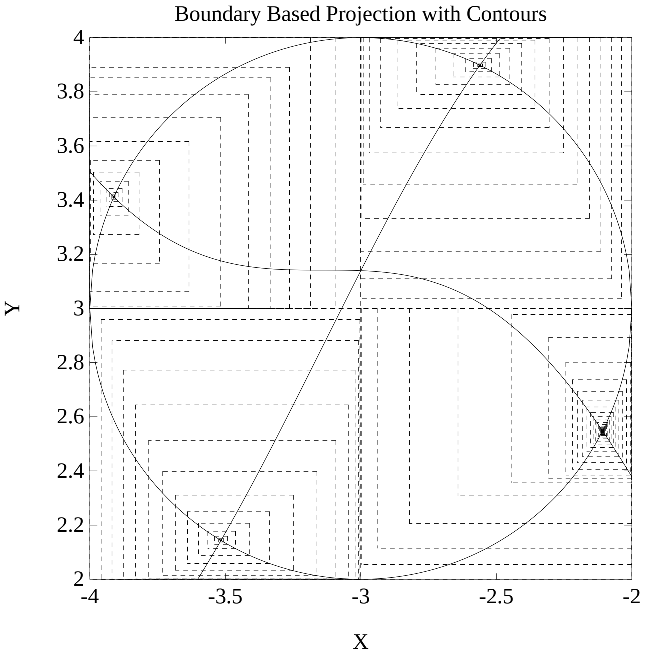

Figure 3: Contours and inner convergence

boundary based interval bisection technique instead. Actually, bisection here is a misnomer for what more accurately should be called boundary-based region elimination using only information from $F(X)$. For our general class of functions, there is no general theory that would recommend the choice of one method over the other for region elimination. However, the boundary based methods can be implemented in an adaptive scheme that dynamically tests for every non-zero entry of $J(X)$, which method will eliminate a larger region per unit of work in each dimension with respect to each non-zero entry of $J(X)$.

For example, let's consider the case when for a k^{th} iterate of the boundary based Newton method, in some dimension, $X_j^k = [\underline{X}_j^k, \overline{X}_j^k]$, using $F_i(X)$, and $\frac{\partial F_i}{\partial X_j}$, for one of the faces we eliminate a region γ_j. If $0 \notin F(X_1^k, X_2^k, \ldots, \gamma_j, \ldots, X_n^k)$, this means that in this case, the boundary-based 'bisection' would have eliminated a larger region in one iteration.

There is also a simple check that enables us to find out when bisection method is worse: it is sufficient to check for the existence of an ideal step in the region eliminated by bisection method. If there is no ideal step, then the boundary-based interval Newton method would have eliminated a larger region in one iteration.

The number of floating point operations for one iteration of Newton's method and a bisection method are known *a priori* allowing the computer to decide which method is more computationally efficient (both methods use the same amount of computer memory, so when deciding which of the methods to use, we do not have to take memory into consideration).

The above tests can be used in a decision strategy that can adaptively choose one of the boundary-based methods. Thus, an algorithm might be using boundary-based Newton's method in one dimension using one F_i, and bisection in the same dimension using a different F_i. There are many possibilities for a decision process based upon local region elimination. Additionally, the boundary-based methods can be adaptively combined with Krawczyk's method using known tests for safe-starting regions.

4 Advantages and disadvantages

The main advantage is that boundary-based methods do not require the inversion of a Jacobian matrix, and do not even require that we know an approximation to this inverse matrix. Additionally, the methods tend to rely upon bisection far less than Krawczyk's method; this fact reduces the storage requirements and the need for 'rejoining' regions that were previously split up. The boundary-based methods have inherent adaptive strategies that allow dynamic choices between Newton's, bisection, and Krawczyk's method. Finally, there appears to be the potential for levels of parallelism, since region elimination at each face can be considered a separate problem.

The main disadvantage of the boundary based Newton's method and bisection is the need to find the faces where $0 \notin F_i(X_{\mathcal{F}})$ and the subsequent reliance upon interval bisection when there are no suitable faces available.

5 Future research

The boundary-based interval Newton's method needs a generalized convergence theory for the multi-dimensional problem. To make a significant improvement in current methods, the boundary-based techniques must eliminate regions more efficiently than traditional bisection and the Hansen-Sengupta method [3]. If this cannot be proven, then we must at least demonstrate adaptive schemes with the previously mentioned algorithms to augment their performance. Additionally, we need to demonstrate parallelism and quantify its contribution to the rate of convergence. Also, we need to uncover which classes of nonlinear systems and possibly sparse unpatterned linear systems are best suited for the boundary based methods.

References

[1] Moore, R. E. *Methods and applications of interval analysis.* SIAM, Philadelphia, 1979.

[2] Alefeld, G. and Herzberger, J. *Introduction to interval computations.* Academic Press, New York, 1983.

[3] Neumaier, A. *Interval methods for systems of equations.* Cambridge University Press, Cambridge, 1990.

[4] Rump, S. M. *Reliability in computing: the role of interval methods in scientific computing.* In: Moore, R. E. (ed.) "Algorithms for verified inclusions: theory and practice". Academic Press, New York, 1988.

[5] Moore, R. E. and Jones, S. T. *Safe starting regions for iterative methods.* SIAM Journal of Numerical Analysis **14** (6) (1977), pp. 1051–1065.

[6] Simcik, L. J. *Ideal step interval Newton method.* Master's Thesis, Department of Mathematics, University of California, Davis, 1992.

L. Simcik
University of California
Department of Mathematics
Davis, California 95616
USA

P. Linz
University of California
Department of Computer Science
Davis, California 95616
USA