A Bisection-Like Algorithm for Branch Switching at a Simple Branch Point.

J. Hughes *   M. Friedman †

March 6, 2007

Abstract

At a Simple Branch Point x_0 of a parameter-dependent nonlinear system, two distinct solutions branches S_1 and S_2 intersect. A popular method of branch switching from S_1 to S_2 at x_0 is to use an orthogonal vector to the tangent vector to S_1at x_0 as an initial guess for the tangent vector to S_2 at x_0. We describe an improved version of this method using a bisection-like procedure, which is especially useful for large problems.

1 Introduction

Consider the problem of computing a parametrized solution branch $x^1(s) = (u(s), \alpha(s)) \in \mathbb{R}^n \times \mathbb{R}$, for some parametrization s, of a parameter-dependent nonlinear system

$$f(x) \equiv f(u, \alpha) = 0, \text{ where } u \in \mathbb{R}^n, \quad \alpha \in \mathbb{R}, \quad f(u, \alpha) \in \mathbb{R}^n. \tag{1}$$

The solution $x_0 = x(s_0)$ of (1) is called *a simple singular point* if $f_x^0 = f_x(x_0)$ has rank $n-1$. In the parameter formulation, where $f_x^0 = \left[f_u^0 \mid f_\alpha^0\right]$, we have $x_0 = (u_0, \alpha_0)$ is a simple singular point if, and only if (i) $\dim \mathcal{N}\left(f_u^0\right) = 1$, $f_\alpha^0 \in \mathcal{R}\left(f_u^0\right)$ or (ii) $\mathcal{N}\left(f_u^0\right) = 2$, $f_\alpha^0 \notin \mathcal{R}\left(f_u^0\right)$.

In this case, two distinct solutions branches x^1 and x^2, call them S_1 and S_2, correspondingly, of (1) pass through x_0. Then we must have

$$\mathcal{N}\left(f_x^0\right) = \text{Span}\left\{v_1, v_2\right\}, \quad \mathcal{N}\left(\left(f_x^0\right)^*\right) = \text{Span}\left\{\psi\right\}.$$

*Mathematical Sciences Department, University of Alabama in Huntsville, Huntsville, AL 35899, Hughes@email.uah.edu. The author was supported in part under NSF DMS-0209536 and NSF ATM-0417774.

†Mathematical Sciences Department, University of Alabama in Huntsville, Huntsville, AL 35899, friedman@math.uah.edu. The author was supported in part under NSF DMS-0209536 and NSF ATM-0417774.

There are several methods for locating a simple singular point x_0 of (1), see e.g. [3]. In particular, general and effective method for computing x_0 would be using a minimally augmented system ([7], [2], [1]). We assume that a predictor - corrector method, with tangent predictor and the Newton type corrector, is used for both computing a parametrized solution branch and for locating a simple singular point on that branch.

Once x_0 has been located, we want to switch branches. One method of branch switching is based on the *algebraic branching equation* (ABE) (see e.g. [3] and references there), which requires the computation of second order derivatives. Instead of using the ABE, one can use

$$f(x_1) = 0, \quad (x_1 - x_0)^* v_2 - \Delta s = 0, \tag{2}$$

where $v_2 = v_{orth}$ is the second null vector of f_x^0:

$$v_{orth} \in \mathcal{N}\left(f_x^0\right), \quad v_{orth} \perp v_1, \quad \|v_{orth}\| = 1, \tag{3}$$

and

$$v_1 = \frac{dx^1(s_0)}{ds}, \quad \|v_1\| = 1, \tag{4}$$

is the the tangent vector to S_1 at x_0. This method is implemented in AUTO [6] and works well in many applications, although there may be situations where it fails.

In this paper we describe an improved version of (2) using a bisection-like procedure.

2 Switching Branches

The idea of the method is as follows. Let v_1 and v_{orth} be defined by (4) and (3), correspondingly.

One continuation step to locate a point on the second branch S_2 is attempted, starting at x_0 and using $v_{guess} := v_{orth}$ as an initial guess for the tangent vector v_2 to S_2 at x_0. Then the resulting point (x_{corr}, v_{corr}) is tested to determine if $x_{corr} \in S_2$ or x_{corr} has fallen back onto the initial branch S_1. The test consists in determining if v_{corr}, which is interpreted as an approximation to v_2, lies between two vectors: v_{right}, close to v_1, and v_{left}, close to -v_1. If v_{corr} is found to lie between v_{right} and v_{left}, then it is concluded that $x_{corr} \in S_2$, and v_{guess} is accepted as an initial guess for v_2, see Fig 1. Otherwise, a bisection-like procedure is initiated to improve v_{guess}.

The resulting algorithm is as follows. Define the orthogonal projection P onto $\mathcal{N}\left(f_x^0\right)$ by

$$P = \left[\begin{array}{cc} v_1 & v_{orth} \end{array}\right] \cdot \left[\begin{array}{c} v_1^T \\ {v_{orth}}^T \end{array}\right]. \tag{5}$$

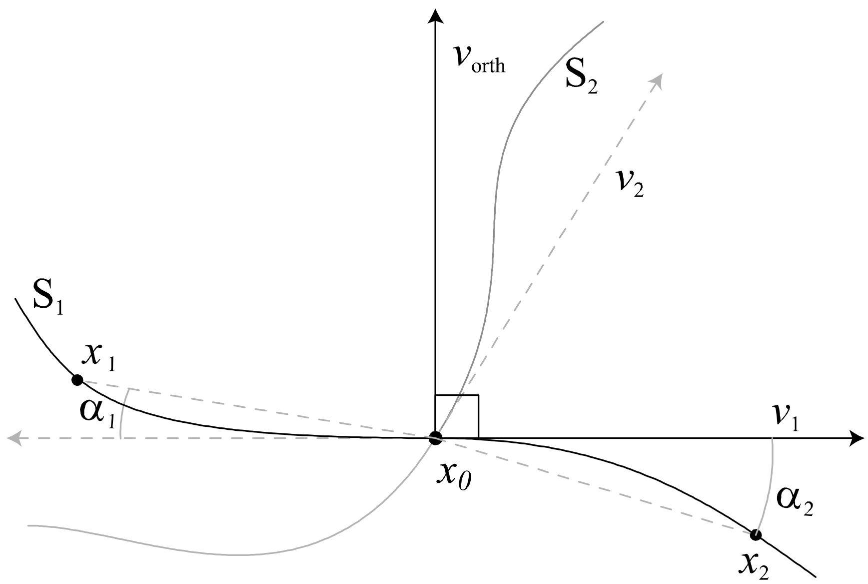

Figure 1: A simple branch point, x_0, and nearby points x_1 and x_2.

Algorithm 1 A bisection-like method for switching from a branch S_1 to the second branch S_2 at a simple branch point x_0.

Input: A simple branch point x_0 on the first branch S_1, an initial step size h, the points $x_1 = x^1(s_0 - h)$ and $x_2 = x^1(s_0 + h)$ on S_1, the tangent vector v_1 (4) to S_1 at x_0, v_{orth} (3), and a tolerance $\delta > 0$.

1. Initialization.

(a) Set $\alpha_1 :=$ the angle between v_1 and $P\left(x_1 - x_0\right)$, $\alpha_2 :=$ the angle between $-v_1$ and $P\left(x_2 - x_0\right)$, and $\theta_{\min} := \max\{\alpha_1, \alpha_2\}$, $0 \leq \theta_{\min} \leq \frac{\pi}{2}$, (a threshold for detection of a failed continuation attempt), see Fig. 1 and Fig. 2.

(b) If $\theta_{\min} \neq \dfrac{\pi}{2}$ set $\gamma := \tan(\theta_{\min})$, otherwise the method has failed.

(c) Compute the 'boundary' vectors

$$v_{right} := \frac{\gamma v_{orth} + v_1}{\sqrt{\gamma^2 + 1}}, \tag{6}$$

$$v_{left} := \frac{\gamma v_{orth} - v_1}{\sqrt{\gamma^2 + 1}}$$

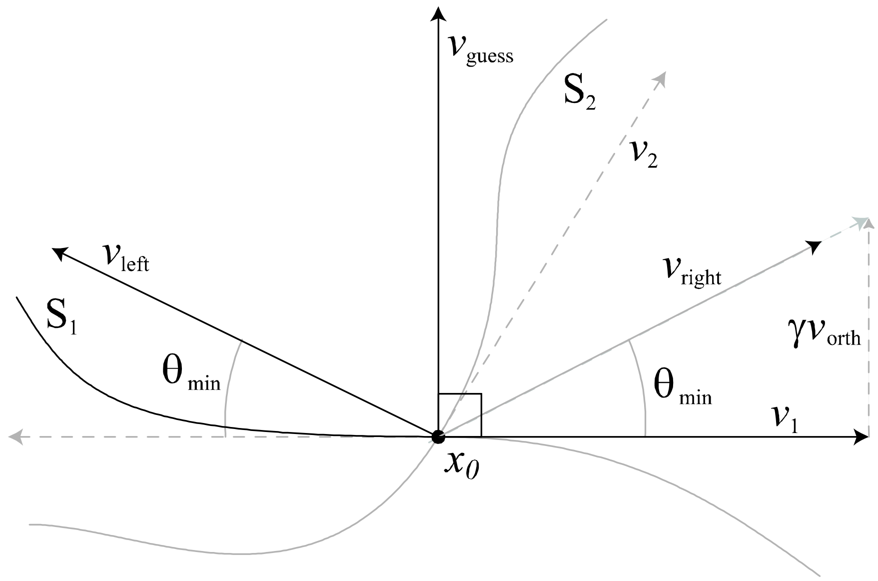

Figure 2: Initial setup of bisection method.

and set $v_{guess} := v_{orth}$, an initial guess for the tangent vector v_2 to S_2 at x_0.

2. The bisection-like procedure.

(a) Do one continuation step starting at (x_0,v_{guess}) to find a point (x_{corr}, v_{corr}).

(b) Set $\theta :=$ the angle between v_{corr} and $P(x_{corr} - x_0)$.

(c) If $\theta \leq \theta_{\min}$ (x_{corr} is judged to have corrected to S_1 and hence is rejected, see Fig. 3)

If $v_1^T P(x_{corr},x_0) \geq 0$ (the solution has corrected to the right), then set $v_{right} := v_{guess}$ (the initial guess should be chosen closer to v_1), else (the solution has corrected to the left) set $v_{left} := v_{guess}$ (the initial guess should be chosen closer to $-v_1$).

If $\|v_{left} - v_{right}\| \geq \delta$ then set $v_{guess} := (v_{left}+v_{right})/\left\|v_{left} + v_{right}\right\|$, else set $h := \frac{h}{2}$.

Go to (a).

(d) accept (x_{corr}, v_{corr}), where x_{corr} is the first point on S_2, and v_{corr} is an approximation to v_2 (see Fig. 4).

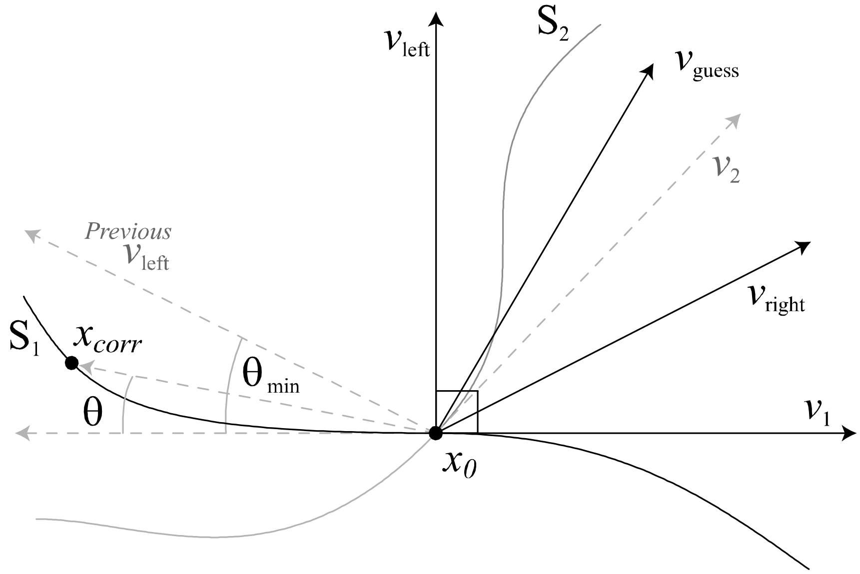

Figure 3: x_{corr} corrected to the left.

Theorem 1 Let S_1 and S_2 be two distinct solution branches of the system (1) that pass through a simple branch point x_0, and let v_1, given by (4), be the tangent vector to S_1 at x_0. Then the bisection-like procedure given by Algorithm 1 converges to the tangent vector v_2 to S_2 at x_0 whenever $h > 0$ is sufficiently small.

Proof. First note that the existence of v_2 is guaranteed by the existence of solution [3] to the *algebraic branching equation*. For convenience of notation, we identify v_{right}, v_{left}, v_{guess}, and v_2, correspondingly, with the angles α_r, α_l, α_g, and α_2, which they form, correspondingly, with v_1. Then according to Step 2(c) of Algorithm 1, at each step of Algorithm 1 (i) α_g, $\alpha_2 \in [\alpha_r,\, \alpha_l]$, and (ii) after each step of Algorithm 1 the length of the interval $[\alpha_r,\, \alpha_l]$ is reduced by a factor of 2. Hence α_g converges to α_2 whenever the Newton method converges to a point on S_2 from a starting point $x_0 + h \cdot v_2$. The latter convergence is, of course, guaranteed if we choose $h > 0$ sufficiently small. ■

Example 1 Consider the simple one dimensional ODE

$$x' = (e^{-x} - 1)(x - a) \tag{7}$$

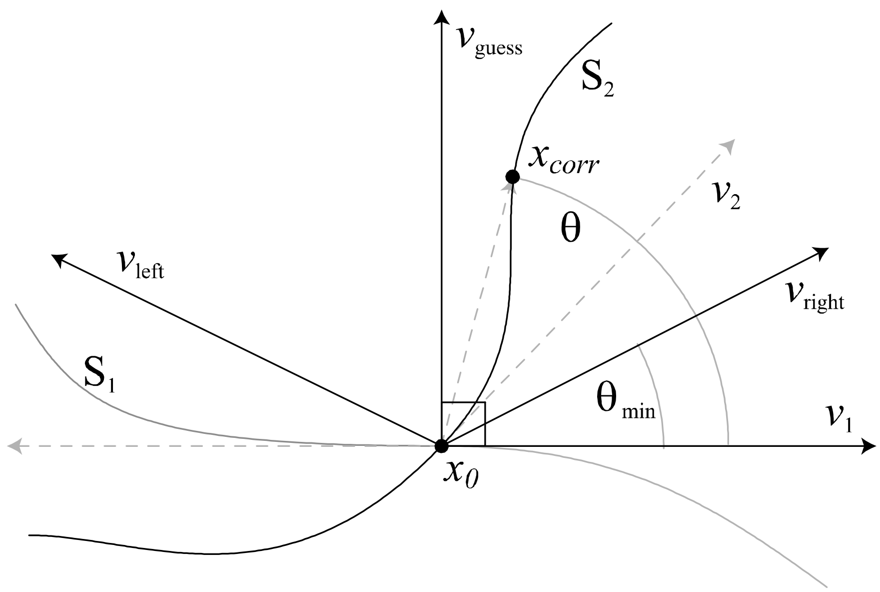

Figure 4: x_{corr} found on the second branch, S_2.

This system has equilibria at $x = 0$ and $x = a$: When $a = 0$ there is a transcritical branch point bifurcation. Using continuation starting at $u = (x, a) = (0, -1)$ a continuation using AUTO detects the bifurcation at $x_{BP} \simeq (0, 0)$ but is unable to continue the second branch of equilibrium points when starting from the x_{BP}. Using the algorithm above with MATCONT one step of the bisection loop is completed and the remaining branch vector is found to be $v_2 = (0.707, 0.707) \simeq (\sqrt{2}, \sqrt{2})$ Continuation is attempted using v_2 and successfully computes the first few points of the second branch.

Example 2 The *1D Brusselator* [8] is a well known model system for autocatalytic chemical reactions with diffusion:

$$\begin{array}{ll} \frac{d_1}{l^2}u'' - (b+1)u + u^2v + a = 0, & \frac{d_2}{l^2}v'' + bu - u^2v = 0, \quad \text{in } \Omega = (0,1), \\ u(0) = u(1) = a, & v(0) = v(1) = \frac{b}{a}. \end{array} \tag{8}$$

This problem exhibits a rich bifurcation scenario and has been used in the literature as a standard model for bifurcation analysis, see e.g. [4, 9]. We utilize the second order central difference discretization

$$f'' \approx \frac{1}{h^2}(f_{i-1} - 2f_i + f_{i+1}), \quad h = (N+1)^{-1},$$

with uniform grid of N grid points. Since there are two unknowns per grid point, the resulting discrete problem, which has dimension $n = 2N$, can be written in the form (1). This discretization of the Brusselator is used in a CL_MATCONT example [5]. Setting $N = 256$ and using MATCONTL continuation over parameter l, from a non-constant equilibrium solution, we locate a brach point x_{BP} for $l \approx 0.250$. Using the method from AUTO we are unable to perform continuation of a second branch from x_{BP}. Using the bisection method, we are able to perform continuation of the second branch to a closed curve as seen in figure 5.

Remark 1 This method has a couple of major advantages. First, one need not calculate second derivatives. Second, the method will always succeed in finding an approximate tangent vector given a small enough step size. This allows for good possibility for continuation of branches for large dimensional problems. However, this method can fail if the step size is chosen too large and the actual angle between the tangent vectors associated with the branch vectors is less than the calculated value of $\theta_{\min}$. If this happens, the step size used for continuation can be reduced.

Remark 2 (Implementation Issues) In the case of a near-linear initial branch $\theta_{\min} \simeq 0$, a tolerance must be defined in some other way. If the

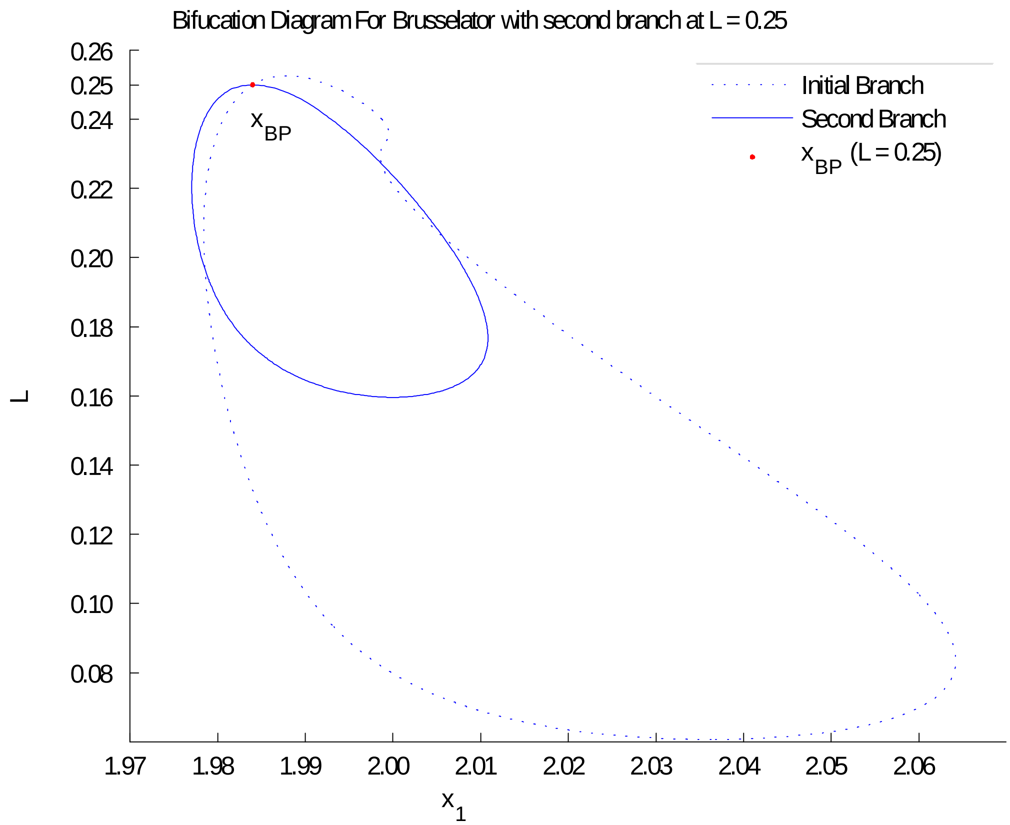

Figure 5: MatcontL output for Brussalator

actual angle between branches is less than the tolerance, the method will fail completely; this can be resolved by either reducing the tolerance or re-scaling the problem. Although Moore-Penrose continuation is used in the implementation with MATCONT, it is not required. The method only requires Newton correction of the predictor point not the tangent vector.

References

[1] E. Allgower and K. Georg, *Numerical path following*, in Handbook of Numerical Analysis, volume 5, P. G. Ciarlet and J. L. Lions, eds., North-Holland, 1997, pp. 3–207.

[2] E. Allgower and H. Schwetlick, *A general view of minimally extended systems for simple bifurcation points*, Z. angew. Math. Mech., 77 (1997), pp. 83–98.

[3] W.-J. Beyn, A. Champneys, E. J. Doedel, Yu.A. Kuznetsov, B. Sandstede, and W. Govaerts, *Numerical continuation and computation of normal forms*, in Handbook of Dynamical Systems III: Towards Applications, B. Fiedler, ed., Elsevier, 2001, ch. 4.

[4] C. S. Chien, Z. Mei, and C. L. Shen, *Numerical continuation at double bifurcation points of a reaction-diffusion problem*, Int. J. Bifur. and Chaos, 8 (1997), pp. 117–139.

[5] A. Dhooge, W. Govaerts, Yu.A. Kuznetsov, W. Mestrom, and A. M. Riet, *MATLAB continuation software package CL_MATCONT*, Jan. 2003. `http://www.math.uu.nl/people/kuznet/cm/`.

[6] E. Doedel, A. Champneys, T. Fairgrieve, Y. Kuznetsov, B. Sandstede, and X.-J. Wang, *AUTO97: continuation and bifurcation software for ordinary differential equations (with homcont)*, 1997. `http://indy.cs.concordia.ca/`.

[7] A. Griewank and G. Reddien, *Characterization and computation of generalized turning points*, SIAM J. Numer. Anal., 21 (1984), pp. 176–185.

[8] R. Lefever and I. Prigogine, *Symmetry-breaking instabilities in dissipative systems II*, J. Chem. Phys., 48 (1968), pp. 1695–1700.

[9] Z. Mei, *Numerical bifurcation analysis for reaction-diffusion equations*, PhD thesis, University of Marburg, 1997.