

Numerical Continuation of Branch Points of Limit Cycles in MATCONT

Annick Dhooge¹, Willy Govaerts¹, and Yuri A. Kuznetsov²

¹ Department of Applied Mathematics and Computer Science, Gent University,
Krijgslaan 281-S9,B-9000 Gent, Belgium

{Annick.Dhooge,Willy.Govaerts}@UGent.be

² Mathematical Institute, Utrecht University,
Budapestlaan 6, 3584 CD Utrecht, The Netherlands

Yu.Kuznetsov@math.uu.nl

Abstract. MATCONT is a MATLAB continuation package for the interactive numerical study of a range of parameterized nonlinear problems. We discuss a recent addition to the package, namely the continuation of branch points of limit cycles in three parameters which is not available in any other package. This includes the exact location of the BPC points and branch switching. The algorithm is important in the numerical study of symmetry and we illustrate it in the case of the famous Lorenz model for the atmospheric circulation.

1 Introduction

Numerical continuation is a technique to compute a sequence of points which approximate a branch of solutions to $F(x) = 0$ where $F : \mathbb{R}^{N+1} \rightarrow \mathbb{R}^N$.

In particular, we consider a dynamical system of the form

$$\frac{dx}{dt} = f(x, \alpha) \tag{1}$$

with $x \in \mathbb{R}^n$, $f(x, \alpha) \in \mathbb{R}^n$, and α a vector of parameters. In this setting equilibria, limit points, limit cycles etcetera can be computed.

MATCONT provides a continuation toolbox for (1) which is compatible with the standard MATLAB representation of ODEs. The package is freely available at: <http://allserv.UGent.be/~ajdhooge/research.html>. It requires MATLAB 6.*. In [4] we describe the implementation in MATCONT of the continuation of the Fold bifurcation of limit cycles, using a minimal extended system, i.e. we only append a scalar equation to the definition of limit cycles [6]. Here we discuss the continuation in three parameters of branch points of limit cycles, an algorithm which is not available in any other package. For general background on dynamical systems we refer to [8,9]; for the algorithms that involve BPC we refer to [7].

2 Mathematical Background on Limit Cycles and Their Branch Points

2.1 Limit Cycles and Their Branch Points

A limit cycle is an isolated closed orbit that corresponds to a periodic solution of (1) with period T , i.e. $x(0) = x(T)$. Since T is not known in advance, it is customary (cf AUTO [5], CONTENT [10]) to use an equivalent system defined on the unit interval $[0, 1]$ by rescaling time:

$$\begin{cases} \frac{dx}{dt} - Tf(x, \alpha) = 0, \\ x(0) = x(1). \end{cases} \quad (2)$$

To obtain a unique solution the following integral constraint is often used [5,10]:

$$\int_0^1 \langle x(t), \dot{x}_{old}(t) \rangle dt = 0, \quad (3)$$

where $\dot{x}_{old}(t)$ is the derivative vector of a previously calculated limit cycle and is therefore known, $\langle x, v \rangle$ is just a different notation for $x^T v$. The left-hand side of (3) will be sometimes denoted by $Int_{\dot{x}_{old}}(x)$.

If, say, α_1 is the control parameter in (1) then a branch point of limit cycles (BPC) is a solution to (2)–(3) in $(x(t), T, \alpha_1)$ -space where the null space of the derivative operator of (2)–(3) with respect to $x(t), T, \alpha_1$ has dimension greater than one. Generically, it then has a two-dimensional null space and the solution to (2)–(3) has two intersecting branches in $(x(t), T, \alpha_1)$ -space. The complete BVP defining a BPC point using the minimal extended system is

$$\begin{cases} \frac{dx}{dt} - Tf(x, \alpha) & = 0 \\ x(0) - x(1) & = 0 \\ \int_0^1 \langle x(t), \dot{x}_{old}(t) \rangle dt & = 0 \\ G[x, T, \alpha] & = 0 \end{cases} \quad (4)$$

where $G = (G^{1T} \ G^{2T})^T$ is defined by requiring

$$L \begin{pmatrix} v^{11} & v^{21} \\ v^{12} & v^{22} \\ v^{13} & v^{23} \\ G^1 & G^2 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (5)$$

Here v^{1k} and v^{2k} are vector functions, G^1 and G^2 are scalars and

$$L = \begin{bmatrix} D - Tf_x(x(\cdot), \alpha) & -f(x(\cdot), \alpha) & -Tf_{\alpha_1}(x(\cdot), \alpha) & w_0^1 \\ \delta_1 - \delta_0 & 0 & 0 & w_0^2 \\ Int_{\dot{x}_{old}(\cdot)} & 0 & 0 & w_0^3 \\ v_0^{11} & v_0^{12} & v_0^{13} & 0 \\ v_0^{21} & v_0^{22} & v_0^{23} & 0 \end{bmatrix} \quad (6)$$

4 Continuation of BPC Cycles

4.1 Discretization of the BPC Equations

The last equation in (4) expresses that the operator

$$\begin{bmatrix} D - Tf_x(x(\cdot), \alpha) & -f(x(\cdot), \alpha) & -Tf_{\alpha_1}(x(\cdot), \alpha) \\ \delta_1 - \delta_0 & 0 & 0 \\ Int_{\tilde{x}_{old}(\cdot)} & 0 & 0 \end{bmatrix} \quad (8)$$

that appears as a block in (6) is rank deficient. In the numerical implementation in MATCONT and CL-MATCONT we replace this by the condition that the discretized operator of (8) is rank deficient: $G_d^1 = G_d^2 = 0$. To find G_d^i , we solve

$$L^d \begin{pmatrix} v_d^{11} & v_d^{21} \\ v_d^{12} & v_d^{22} \\ v_d^{13} & v_d^{23} \\ G_d^1 & G_d^2 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (9)$$

where $L^d =$

$$\begin{bmatrix} [D - Tf_x(x(\cdot), \alpha)]_d & [-f(x(\cdot), \alpha)]_d & [-Tf_{\alpha_1}(x(\cdot), \alpha)]_d & w_{0d}^1 \\ I_{\dim(x)} & 0_{n \times (Nm-1)\dim(x)} & -I_{\dim(x)} & w_{0d}^2 \\ & Int_{[\tilde{x}_{old}(\cdot)]_d} & & w_{0d}^3 \\ & v_{0d}^{11T} & v_{0d}^{12} & 0 \\ & v_{0d}^{21T} & v_{0d}^{22} & 0 \\ & v_{0d}^{23} & v_{0d}^{23} & 0 \end{bmatrix}, \quad (10)$$

where the bordering vectors $v_{0d}^{11}, v_{0d}^{21}, w_{0d}^1$ and w_{0d}^2 and scalars $v_{0d}^{12}, v_{0d}^{22}, v_{0d}^{13}, v_{0d}^{23}$ and w_{0d}^3 are chosen so that L^d is nonsingular. The structure is similar to that of (7); however, the bordering rows and columns have a different meaning.

To continue a solution branch of the discretized equations (4), the Jacobi matrix of the system is needed, which means that the derivatives of G_d with respect to the unknowns of the system, i.e., with respect to $x^{i,j}, T$, and the control parameters α , have to be calculated.

The derivative with respect to z (being a component of $x^{i,j}, T$ or α) is obtained from

$$L^d \begin{pmatrix} v_{dz}^{11} & v_{dz}^{21} \\ v_{dz}^{12} & v_{dz}^{22} \\ v_{dz}^{13} & v_{dz}^{23} \\ G_{dz}^1 & G_{dz}^2 \end{pmatrix} + J_z \begin{pmatrix} v_d^{11} & v_d^{21} \\ v_d^{12} & v_d^{22} \\ v_d^{13} & v_d^{23} \\ G_d^1 & G_d^2 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Simplifying gives

$$L^d \begin{pmatrix} v_{dz}^{11} & v_{dz}^{21} \\ v_{dz}^{12} & v_{dz}^{22} \\ v_{dz}^{13} & v_{dz}^{23} \\ G_{dz}^1 & G_{dz}^2 \end{pmatrix} = \begin{pmatrix} J_1 & J_2 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

where $J_i = [Tf_x(x(t), \alpha)]_{dz} v_d^{i1} + [f(x(t), \alpha)]_{dz} v_d^{i2} + [Tf_{\alpha_1}(x(t), \alpha)]_{dz} v_d^{i3}$, $i = 1, 2$. Instead of solving this for every z we solve the transposed equations

$$(w_d^{1T}, w_d^{2T}, w_d^3, w_d^4, w_d^5) L^d = (0, 0, 0, 1) \quad (11)$$

where w_d^1 is a $\dim(x) \times Nm$ vector, w_d^2 a $\dim(x)$ vector and w_d^3, w_d^4 and w_d^5 are scalars. Combining (9) and (11) we find

$$G_{dz}^i = w_d^{1T} ([Tf_x(x(t), \alpha)]_{dz} v_d^{i1} + [f(x(t), \alpha)]_{dz} v_d^{i2} + Tf_{\alpha_1}(x(t), \alpha)]_{dz} v_d^{i3}). \quad (12)$$

So in each iteration step we solve three systems with the structure of (7) or its transpose.

4.2 Initialization and Adaptation of the Borders

The bordering vectors in (10) must be such that the matrix L^d is nonsingular. We choose them in such a way that L^d is as well conditioned as possible. This involves an initialization of the borders when the continuation is started and a subsequent adaptation during the continuation.

During the initialization the borders must be chosen so that the extension L^d of $O =$

$$\begin{bmatrix} [D - Tf_x(x(\cdot), \alpha)]_d & [-f(x(\cdot), \alpha)]_d & [-Tf_{\alpha_1}(x(\cdot), \alpha)]_d \\ I_{\dim(x)} & 0_{\dim(x) \times (Nm-1)\dim(x)} & -I_{\dim(x)} \\ & Int_{f(x(\cdot), \alpha)_d} & 0 \end{bmatrix}$$

has full rank. We first perform an QR orthogonal-triangular decomposition with column pivoting. The MATLAB command $[Q, R, E] = QR(full(O))$ produces a permutation matrix E , an upper triangular matrix R of the same dimension as O and an unitary matrix Q so that $OE = QR$. The column pivoting guarantees that the QR decomposition is rank revealing and in particular that $\text{abs}(\text{diag}(R))$ is decreasing. Since O has rank defect 1, the last element on the diagonal and the bottom right element of R should be zero (up to approximation). The borders $[v_{0d}^{11T} v_{0d}^{12} v_{0d}^{13}]$ and $[v_{0d}^{21T} v_{0d}^{22} v_{0d}^{23}]$ in (10) are chosen as an orthogonal base for the null space of O . If p is a two-column matrix that spans this null space, then from $Op = 0$ follows that $RE^T p = 0$. Setting the bottom right element and the last element on the diagonal of R to zero, we obtain

$$\left(\begin{array}{cccc|cc} * & * & * & \dots & * & * \\ 0 & * & * & \dots & * & * \\ 0 & 0 & * & \dots & * & * \\ & & & \dots & \dots & * \\ 0 & 0 & 0 & \dots & * & * \\ \hline 0 & 0 & 0 & \dots & 0 & 0 \end{array} \right) E^T p = \begin{pmatrix} 0 \\ 0 \\ \hline 0 \\ 0 \end{pmatrix}$$

By imposing some structure on $E^T p$ we get

$$\left(\begin{array}{c|cc} R_1 & b_1 & b_2 \\ \hline 0 & 0 & 0 \end{array} \right) \left(\begin{array}{c|c} p_1 & p_2 \\ \hline 1 & 0 \\ 0 & 1 \end{array} \right) = \begin{pmatrix} 0 \\ 0 \\ \hline 0 \\ 0 \end{pmatrix}.$$

or

$$R_1(p_1 \ p_2) = -(b_1 \ b_2)$$

where R_1 is a nonsingular square upper triangular matrix. So $[v_{0d}^{11T} v_{0d}^{12} v_{0d}^{13}]$ and $[v_{0d}^{21T} v_{0d}^{22} v_{0d}^{23}]$ in (10) are initially chosen as the normalization and orthogonalization of $E[(R_1 \setminus [-b_1, -b_2]); eye(2)]$ where $eye(2)$ is the 2-by-2 identity matrix. We choose this column as the bordering column $[w_{0d}^{1T} w_{0d}^{2T} w_{0d}^{3T}]^T$ in (10). This choice of the borders in (10) makes the bordered matrix nonsingular.

The borders $[v_{0d}^{11T} v_{0d}^{12} v_{0d}^{13}]$ and $[v_{0d}^{21T} v_{0d}^{22} v_{0d}^{23}]$ are adapted by replacing them by the normalized and orthogonalized $[v_d^{11T} v_d^{12} v_d^{13}]$ and $[v_d^{21T} v_d^{22} v_d^{23}]$ in (9). The borders w_{0d}^1, w_{0d}^2 and w_{0d}^3 in (10) are adapted by solving the transposed equations and replacing them respectively by the normalized and orthogonalized w_d^1, w_d^2 and w_d^3 in (11).

5 BPC Cycles on a Curve of Limit Cycles

Generically, i.e. if no symmetry is present, then BPC are not expected on curves of limit cycles. However, they are common if the system has symmetry. The location and processing of BPC in that case requires a special treatment.

5.1 Branch Point Locator

Location of BPC points in as zeros of some test functions causes numerical difficulties because no local quadratic convergence can be guaranteed (see [3] in the case of equilibria). This difficulty is avoided by introducing an additional unknown $\beta \in \mathbb{R}$ and considering the minimally extended system:

$$\begin{cases} \frac{dx}{dt} - Tf(x, \alpha) + \beta p_1 & = 0 \\ x(0) - x(1) + \beta p_2 & = 0 \\ \int_0^1 \langle x(t), \dot{x}_{old}(t) \rangle dt + \beta p_3 & = 0 \\ G[x, T, \alpha] & = 0 \end{cases} \quad (13)$$

where G is defined as in (5) and $[p_1^T p_2^T p_3^T]^T$ is the bordering vector $[w_{0d}^{1T}; w_{0d}^{2T}; w_{0d}^{3T}]^T$ in (10) calculated as in §4.2. We solve this system with respect to x, T, α_1 and β by Newton's method with initial $\beta = 0$. A branch point (x, T, α_1) corresponds to a regular solution $(x, T, \alpha_1, 0)$ of system (13) (see [1], p. 165).

5.2 Processing of the BPC

The tangent vector v_{1st} at the BPC singularity is approximated as $v_{1st} = \frac{v_1 + v_2}{2}$ where v_1 is the tangent vector in the continuation point previous to the BPC and v_2 is the one in the next point. To start the continuation of the secondary cycle branch passing through the BPC point, we need an approximation of the tangent vector of the secondary branch. We choose the vector which is in the space spanned by $p = [p_1 p_2]$ obtained in §4.2 as an orthogonal base for the null space of O and orthogonal to the tangent vector to the primary branch.

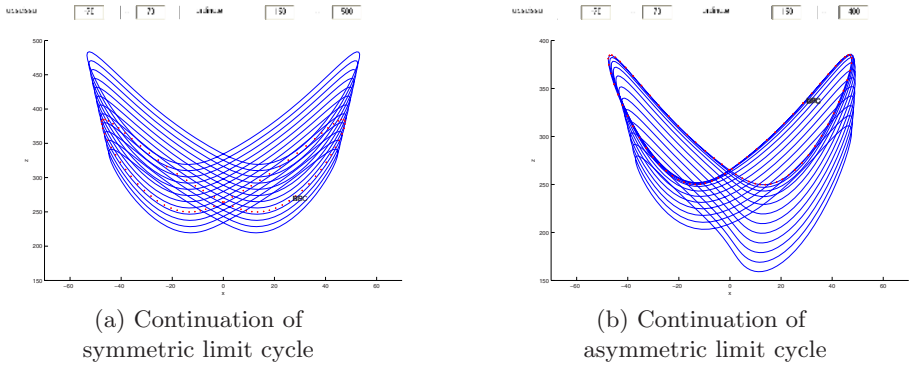


Fig. 1.

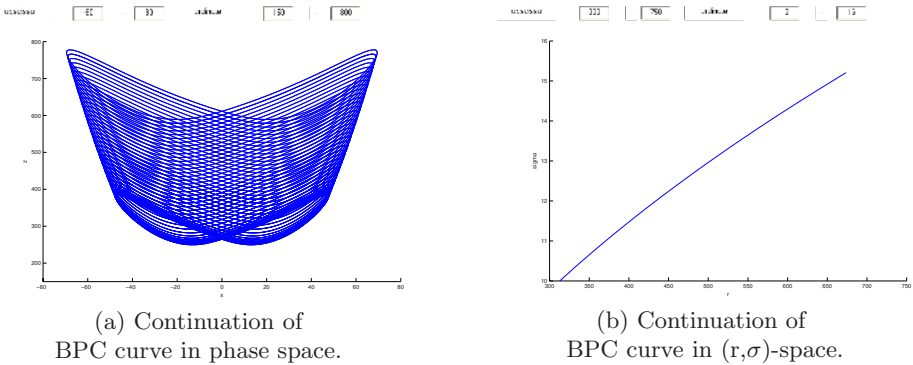


Fig. 2.

6 An Example

Consider the Lorenz model [11] where x, y and z are state variables and r, σ and b are parameters:

$$\begin{cases} \dot{x} = \sigma(-x + y) \\ \dot{y} = rx - y - xz \\ \dot{z} = -bz + xy. \end{cases} \quad (14)$$

This problem satisfies the equivariance relation with respect to a group of two transformations, i.e. $\{I_3, S\}$, where $S = \text{Diag}(-1, -1, 1)$. As in the Tutorial to CONTENT[10], we compute an orbit starting from the point $(0, 50, 600)$ at $\sigma = 10$, $r = 400$ and $b = \frac{8}{3}$ and start a limit cycle continuation with respect to the control parameter r from the converged closed orbit. This is clearly a branch of S -symmetric periodic solutions of (14), see Fig. 1(a). We detect a BPC at $r = 312.9735\dots$ We continue in r the secondary cycle branch passing through the BPC point. From Fig. 1(b) it is clear that for the secondary cycle the S -symmetry is broken. To compute the branch of BPC points with respect to r through the BPC point with control parameters r, σ we need to introduce

an additional free parameter that breaks the symmetry. We choose to introduce a parameter ϵ and extend the system (14) by simply adding the term ϵ to the right-hand side of the first equation in (14). For $\epsilon = 0$ this reduces to (14) while for $\epsilon \neq 0$ the symmetry is broken. Using the code for the continuation of generic BPC points with three free parameters r, σ, ϵ we continue the curve of non-generic BPC points, where ϵ remains close to zero ($1e - 8$). The picture in Fig. 2 clearly shows that the symmetry is preserved.

References

1. Beyn, W.J., Champneys, A.R., Doedel, E., Govaerts, W., Kuznetsov, Yu.A., Sandstede, B.: Numerical continuation and computation of normal forms. In: B. Fiedler, ed. "Handbook of Dynamical Systems, Vol 2", Elsevier 2002, 149–219.
2. Dhooge, A., Govaerts, W., Kuznetsov Yu.A.: MATCONT: A MATLAB package for numerical bifurcation analysis of ODEs, *ACM TOMS* **29**(2) (2003), 141–164.
3. Dhooge, A., Govaerts, W., Kuznetsov, Yu.A., Mestrom, W., Riet, A.M. : A Continuation Toolbox in MATLAB, Manual (2003): http://allserv.UGent.be/~ajdhooge/doc_c1_matcont.zip
4. Dhooge, A., Govaerts, W., Kuznetsov, Yu.A.: Numerical continuation of fold bifurcations of limit cycles in MATCONT, Proceedings of the ICCS 2003, Part I. Springer Verlag Lecture Notes in Computer Science, Vol. 2657 (May 2003) (eds. P.M.A. Sloom, D. Abramson, A.V. Bogdanov, J.J. Dongarra, A.Y. Zomaya and Y.E. Gorbachev), 701–710.
5. Doedel, E.J., Champneys, A.R., Fairgrieve, T.F., Kuznetsov, Yu.A., Sandstede, B., Wang, X.J., AUTO97-AUTO2000 : Continuation and Bifurcation Software for Ordinary Differential Equations (with HomCont), User's Guide, Concordia University, Montreal, Canada (1997–2000): <http://indy.cs.concordia.ca>.
6. Doedel, E.J., Govaerts W., Kuznetsov, Yu.A.: Computation of periodic solution bifurcations in ODEs using bordered systems, *SIAM J. Numer. Anal.* **41**(2) (2003), 401–435.
7. Doedel, E.J., Govaerts, W., Kuznetsov, Yu.A., Dhooge A.: Numerical continuation of branch points of equilibria and periodic orbits, Preprint 1280, Department of Mathematics, Utrecht University, The Netherlands (2003)
8. Govaerts, W.: Numerical Methods for Bifurcations of Dynamical Equilibria, SIAM, Philadelphia (2000).
9. Kuznetsov, Yu.A.: Elements of Applied Bifurcation Theory, 2nd edition, Springer-Verlag, New York (1998)
10. Kuznetsov, Yu.A., Levitin, V.V.: CONTENT: Integrated Environment for Analysis of Dynamical Systems. CWI, Amsterdam (1997): <ftp://ftp.cwi.nl/pub/CONTENT>
11. Lorenz, E.: Deterministic non-periodic flow, *J. Atmos. Science* **20** (1963), 130–141.