



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Journal of Sound and Vibration 283 (2005) 495–506

JOURNAL OF
SOUND AND
VIBRATION

www.elsevier.com/locate/jsvi

Short Communication

On computing periodic orbits

M. Tadi*

*Department of Mechanical Engineering, University of Colorado at Denver, Campus Box 112, P.O. Box 173364,
Denver, CO 80217-3364, USA*

Received 2 June 2003; accepted 15 June 2004

Available online 11 November 2004

1. Introduction

This note is concerned with the computation of periodic orbits for nonlinear systems. Nonlinear differential equations appear frequently in modelling various physical phenomena including chemical kinetics [1], biology [2], and electronic devices [3]. Depending on the initial conditions and system parameters, the vector field in the phase space can be very rich and the response of the system can be very different. For autonomous systems periodic orbits are closed curves in the phase plane and their evaluation can help to map out the structure of the flow field in the phase plane. Also, for some nonlinear and otherwise chaotic plants, linear feedback controls have been used to stabilize the system around periodic orbits [4]. As a result the mathematical investigation of such systems often includes the question of existence and the evaluation of periodic orbits [5].

Periodic solutions are closed curves in the phase space, in other words, if the system trajectory is given by $x(t)$, $x \in R^n$, $t \geq 0$, then there exists a positive number T such that $x(t) = x(t + T)$ where T is the period. Therefore, one is led to look for a set of n unknown initial conditions $x(0)$, and an unknown period T by solving a system of n nonlinear algebraic equations given by

$$x(0) - x(T) = 0. \quad (1)$$

Most of the existing methods attempt to solve the above system of equations using various Newton methods [6,7]. A number of methods have also been proposed in which the above system is accompanied by an additional condition such as $s(x(0), T) = 0$, in order to eliminate the

*Tel.: +303-556-8311; fax: +303-556-6371.

E-mail address: mtadi@carbon.cudenver.edu (M. Tadi).

translational invariance of the periodic solution [8,9]. It is also shown that such a condition is not necessary and periodic solutions can be obtained, using Gauss–Newton iterations [7]. Multiple shooting methods [10], and approximate averaging techniques using elliptic functions [11] have also been used to solve for the periodic solutions for applications in circuit simulations and mechanical vibrations. Another method is also proposed in which external controls are used to guide the system towards a prespecified regular orbit [12], which requires a qualitative description of the periodic orbit.

The present work is based on a variational principle. It leads to an iterative algorithm which converges to a point on the orbit. The formulation allows to include higher-order conditions for periodic orbit. In Section 2, we present the first-order conditions and, using a numerical example, explain the algorithm in details. Then, higher-order conditions are included and a number of examples are used to discuss the applicability of the method. Section 3 is devoted to conclusions.

2. A two-point boundary value problem

Consider an autonomous system of first-order ordinary differential equations given by

$$\frac{dx}{dt} = f(x), \quad x \in R^n, \quad f: R^n \mapsto R^n, \quad t \geq 0. \tag{2}$$

We assume that the function $f(x)$ is differentiable as many times as needed. The problem of finding a periodic orbit for this system is to look for an initial condition, $x(0) = x_0$, for which $x(t) = x(t + T)$, where T is the period. The present approach is based on a variational method which seeks to minimize a cost functional given by

$$J = \frac{1}{2}(x_0 - x_T)^\top (x_0 - x_T), \tag{3}$$

where $x_T = x(T)$, and where we are using superscript \top to denote the transpose. The above minimization is subject to the equations of motion and therefore, using Lagrange multipliers, we can formulate a constrained minimization problem given by

$$J = \frac{1}{2}(x_0 - x_T)^\top (x_0 - x_T) + \int_0^T \lambda^\top (\dot{x} - f) dt, \tag{4}$$

where $\lambda(t) \in R^n$ is the Lagrange’s multiplier vector. Extremal conditions are obtained by taking the first variation of the cost functional which is given by

$$\delta J = \hat{y}^\top (\delta x_0 - \delta x_T) + \int_0^T [\delta \lambda^\top (\dot{x} - f) + \lambda^\top (\delta \dot{x} - f_x \delta x)] dt, \tag{5}$$

where $f_x = \partial f / \partial x$, and $\hat{y} = x_0 - x_T$ is the error. We can integrate the term $\lambda^\top \delta \dot{x}$ by parts to give

$$\int_0^T \lambda^\top \delta \dot{x} dt = \lambda^\top (T) (\delta x_T - \dot{x}(T) \delta T) - \lambda^\top (0) \delta x_0 - \int_0^T \dot{\lambda} \delta x dt. \tag{6}$$

Note that the upper limit of the integration, the period T , is also a variable and the variations of the term inside the integral at $t = T$ should be included. After grouping various terms in Eq. (6)

the first variation of the cost functional simplifies to

$$\delta J = (\hat{y} - \lambda(0))^\top \delta x_0 + (-\hat{y} + \lambda(T))^\top \delta x_T + \int_0^T [(-\dot{\lambda}^\top - \lambda^\top f_x) \delta x + \delta \lambda^\top (\dot{x} - f)] dt + (-\lambda(T)^\top f(x(T))) \delta T. \tag{7}$$

The variations in δx_T , $\delta \lambda$, and δx are arbitrary and we can obtain necessary conditions given by

$$\dot{x} = f, \quad x(0) = x_0, \quad \dot{\lambda} = -f_x^\top \lambda, \quad \lambda(T) = \hat{y}. \tag{8}$$

The updating equations are obtained by noting that for the extremal value of the cost functional, $\delta J / \delta x_0 = 0$ and $\delta J / \delta T = 0$. Therefore, starting from initial guesses for x_0 and T these gradients can be used to update their values according to

$$\Delta x_0 = \frac{\delta J}{\delta x_0} = \frac{1}{\alpha} (\lambda(0) - \hat{y}), \quad \Delta T = \frac{\delta J}{\delta T} = \frac{1}{\beta} \lambda(T)^\top f(x(T)), \tag{9}$$

where the parameters α and β are positive constants. This completes the two-point BVP that needs to be solved. We can now formulate an iterative algorithm as follows.

2.1. Algorithm

- (1) Choose an initial condition x_0 , an initial value for T and two positive constants α and β .
- (2) Use the given initial condition and solve the system of equations (1) forward until the time T .
- (3) Obtain the values of $x(T)$, and thereby, obtain the error $\hat{y} = x_0 - x_T$.
- (4) Solve the adjoint equation (8), backward in time, with the boundary condition at $t = T$ given in Eq. (8).
- (5) Update the assumed values of x_0 and T according to Eq. (9) and go to step (2).
- (6) Repeat the process until the error \hat{y} is arbitrary small and convergence is obtained.

The only nonzero component enters the Lagrange multipliers through the error \hat{y} at $t = T$. For vanishing error, the Lagrange multipliers are zero everywhere, and in particular at $t = 0$, i.e. λ_0 , which corresponds to convergence. In this note we start with large values of α and β and keep them constant. As the iterations proceed we gradually reduce their values. For large values of α and β the change is very gradual and the convergence is slow. This requires the algorithm to compute the gradients at every iterations.

We next consider a numerical example in order to explain the algorithm in detail.

2.2. Example 1

We consider the nerve membrane model [13]. This system is also considered in Ref. [7], and is given by

$$\dot{x}_1 = 3 \left(x_2 + x_1 - \frac{x_1^3}{3} - 1 \right), \tag{10}$$

$$\dot{x}_2 = -\frac{1}{3} (x_1 - 0.7 + 0.8x_2). \tag{11}$$

The associated two point BVP is given by

$$\begin{bmatrix} \dot{\lambda}_1 \\ \dot{\lambda}_2 \end{bmatrix} = - \begin{bmatrix} 3(1 - x_1^2) & -\frac{1}{3} \\ 3 & \frac{0.8}{3}x_3 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}, \tag{12}$$

$$\lambda(T) = \hat{y} = x_0 - x_T, \quad x_0 = x_0 + \frac{1}{\alpha}(\lambda(0) - \hat{y}), \quad T = T + \frac{1}{\beta} \lambda(T)^\top f(x(T)). \tag{13}$$

The procedure is to assume an initial guess for x_0 and T and solve Eqs. (10–11) forward from $t = 0$ to T , and thereby, compute the error \hat{y} . The adjoint equation can then be solved backward in time after which the assumed values for x_0 and T can be updated according to Eqs. (13). Fig. 1 shows the values of x_0 and the error \hat{y}^2 as a function of the number of iterations. The iteration is started from the initial guess $(-0.5, 0.5)$. The constant parameters are chosen as $\alpha = 500$ and $\beta = 20$. It also shows the convergence of the period, i.e. $0.1 \times T$. Fig. 2 shows the changes in the initial condition x_0 as it moves towards a point on the orbit, during which, the error is reduced. Fig. 2 also shows two other iterations which were started from $(-1.0, 1.0)$ and $(0.8, 0.7)$. For these cases the same values are used for the constants α and β . The algorithm converges to a different point on the orbit depending on the choice of the initial guess. The parameters α and β are chosen for the convergence of the scheme. It is best to start with a relatively large value and then reduce it for faster convergence. An appropriate value can be chosen after only a few trials.

The periodic orbit for the above system is asymptotically stable. In fact, if we simply start the system from an initial condition, for example $x_0 = (2.0, 2.0)$, then the response of the system approaches the above periodic orbit very quickly. For most dynamical systems the flow field in the phase plane can be very complicated and the above algorithm will require a large number of iterations, in addition to an initial guess which is close to the periodic orbit. This is also the case

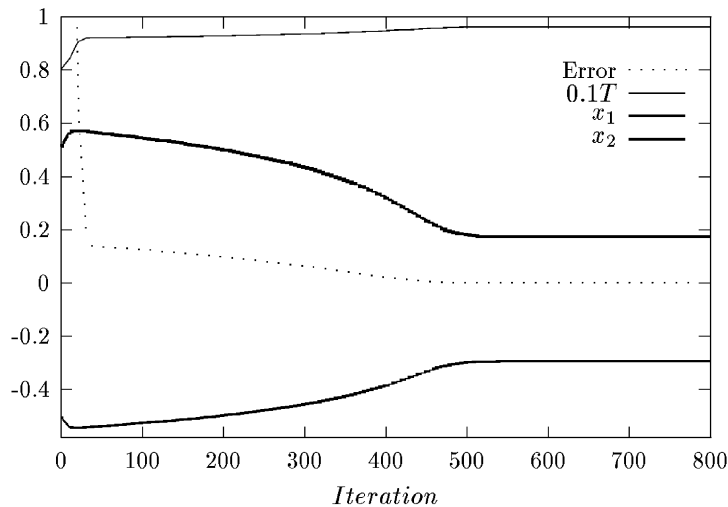


Fig. 1. The convergence of the initial condition to a point on the orbit as a function of the number of iterations for Example 1. It also shows the convergence of the period, $0.1 \times T$, and the reduction in the error. Convergence of x_1 , —; convergence of x_2 , - - -; convergence of $0.1 \times T$,; convergence of error, -

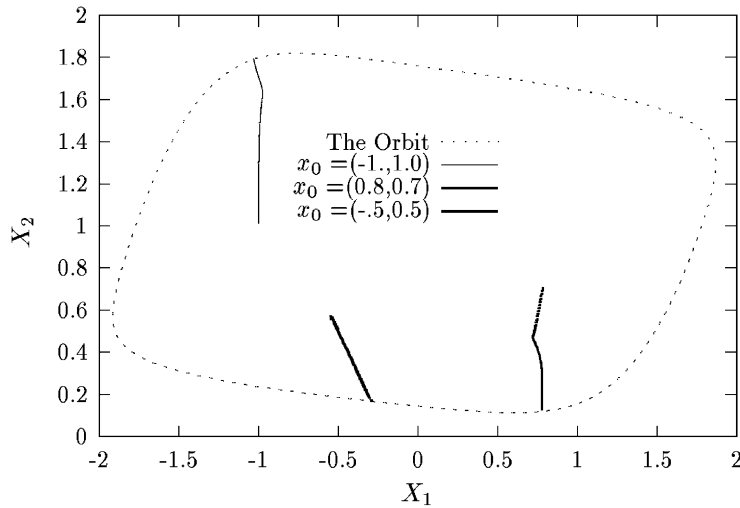


Fig. 2. Periodic orbit for Example 1, and the convergence to a point on the orbit for three different initial conditions. $x_0 = (-1., 1.0)$, ———; $x_0 = (0.8, 0.7)$, ———; $x_0 = (-.5, 0.5)$, ———; orbit,

with Newton-based algorithms. However, using this formulation one can easily include higher-order conditions for the periodic orbit which will greatly improve the convergence of the scheme.

For a dynamical system given by Eq. (2), if a point x_0 is on a periodic orbit with period T , then $x_0 = x_T$. This led to the cost functional given in Eq. (3). In addition to this condition we also note that if the point x_0 is on the periodic orbit with period T , then $\dot{x}_0 = \dot{x}_T$ and $\ddot{x}_0 = \ddot{x}_T, \dots$, assuming that the function f can be differentiated as many times as needed. We can then improve the convergence of the above algorithm by including these *higher-order* conditions in the cost functional. These terms are computed using

$$\dot{x} = f(x), \quad \ddot{x} = f_x f, \quad \ddot{\dot{x}} = (f_x f)_x f, \quad \text{etc.} \tag{14}$$

Now, we can simply add higher-order conditions to the original cost functional and consider the same minimization problem. The modified cost functional is given by

$$J = \frac{1}{2}(x_0 - x_T)^\top (x_0 - x_T) + \frac{1}{2}(\dot{x}_0 - \dot{x}_T)^\top (\dot{x}_0 - \dot{x}_T) + \frac{1}{2}(\ddot{x}_0 - \ddot{x}_T)^\top (\ddot{x}_0 - \ddot{x}_T) + \dots + \int_0^T \lambda^\top (\dot{x} - f) dt. \tag{15}$$

In a similar way, the necessary conditions are obtained by taking the first variation of the cost functional, integrating various terms by parts, and using the fact that the variables can have arbitrary variations. The associated two-point BVP is now given by

$$\dot{x} = f, \quad x(0) = x_0, \quad \dot{\lambda} = -f_x^\top \lambda, \tag{16}$$

$$\lambda(T) = \hat{y}_1 + [f_x|_T]^\top \hat{y}_2 + \left[\frac{\partial}{\partial x} (f_x f) |_T \right]^\top \hat{y}_3 + \dots, \tag{17}$$

where the symbol $*|_T$ is used to denote the evaluation of $*$ at $t = T$. After each iterations, the initial conditions are updated according to

$$x_0 = x_0 + \frac{1}{\alpha} \left(\lambda(0) - \hat{y}_1 - [f_x|_0]^\top \hat{y}_2 - \left[\frac{\partial}{\partial x} (f_x f)|_0 \right]^\top \hat{y}_3 + \dots \right), \tag{18}$$

$$T = T + \frac{1}{\beta} \lambda(T)^\top f(T), \tag{19}$$

where

$$\hat{y}_1 = x_0 - x_T, \quad \hat{y}_2 = f|_0 - f|_T, \quad \hat{y}_3 = (f_x f)|_0 - (f_x f)|_T, \quad \dots \tag{20}$$

The above two-point BVP can now be solved in the iterative algorithm described in Section 2. We next consider a number of numerical examples.

2.3. Example 2

We consider the third-order Rossler system [14]. The dynamics is given by

$$\dot{x}_1 = 2 + x_1(x_2 - 4), \quad \dot{x}_2 = -x_1 - x_3, \quad \dot{x}_3 = x_2 + bx_3, \tag{21-23}$$

where $b = 0.42$. For this system we include four conditions namely, $x_0 - x_T, \dot{x}_0 - \dot{x}_T, \ddot{x}_0 - \ddot{x}_T$ and $\ddot{\dot{x}}_0 - \ddot{\dot{x}}_T$. Higher derivatives of the state are given by

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 2 + x_1(x_2 - 4) \\ -x_1 - x_3 \\ x_2 + bx_3 \end{bmatrix}, \quad \ddot{x} = f_x f = \begin{bmatrix} 2(x_2 - 4) + x_1(x_2 - 4)^2 - x_1(x_1 + x_3) \\ -2 - x_1(x_2 - 4) - x_2 - bx_3 \\ -x_1 - x_3 + b(x_2 + bx_3) \end{bmatrix}, \\ \ddot{x} &= \begin{bmatrix} -16x_2 + 2x_2^2 + 32 - 70x_1 + 16x_1^2 - 4x_3 + x_1x_2(x_2^2 - 12x_2 + 47 - 4x_1 - 3x_3) + (12 - b)x_1x_3 \\ -2x_2 - x_1x_2^2 + 8x_1x_2 + 8 - 15x_1 + x_1^2 + x_1x_3 - bx_2 + (1 - b^2)x_3 \\ -2 - x_1x_2 + (4 - b)x_1 + b(b^2 - 2)x_3 + (b^2 - 1)x_2 \end{bmatrix}. \end{aligned} \tag{24}$$

The coefficient matrices $[f_x]^\top, [\partial \ddot{x} / \partial x]^\top$, and $[\partial \dot{\ddot{x}} / \partial x]^\top$ can readily be obtained. We choose the constant parameters $\alpha = 6. \times 10^2$ and $\beta = 2. \times 10^2$. The iteration is started from an initial condition $x_0 = (1.0, -2.0, 3.0)$. The time step size is chosen as $\Delta t = 0.001$. Fig. 3 shows convergence of the initial condition as a function of the number of iterations. It also shows that the error is reduced and the iteration converges to a point on the periodic orbit. Figs. 4 and 5 show the periodic solution when the system is started from the calculated initial condition, i.e. $(0.37476359, -0.8575478, -3.5193117)$. This orbit is in fact an unstable orbit imbedded in a chaotic attractor. We next consider the same system and start the iterations from the initial condition $x_0 = (1.0, 0.1, -1.0)$. The algorithm reduces the error and as the iterations proceeds it converges to the point $(0.52943123, 0.22236117, -0.5294312)$. However, this is a critical point (equilibrium point) of the function, i.e., $f(x) = 0$. The algorithm converges to extremal points of the cost functional given in Eq. (15). This also includes the critical points of the system.

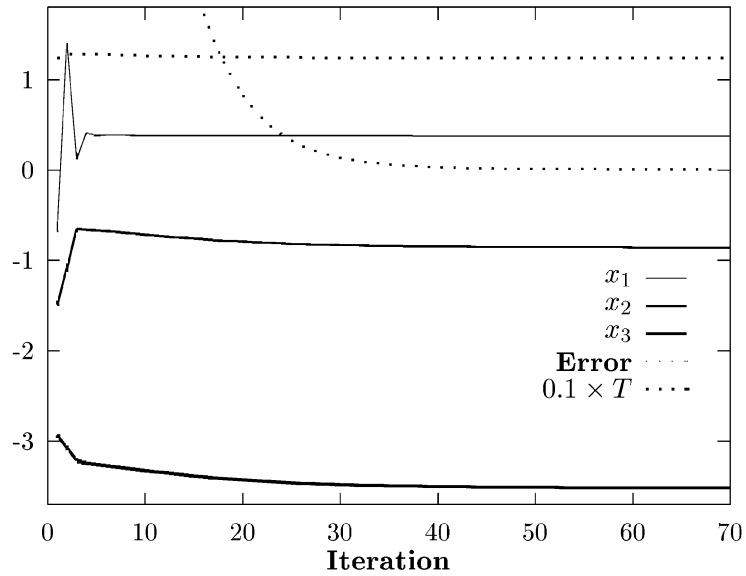


Fig. 3. The convergence of the initial condition to a point on the orbit as a function of the number of iterations for Example 2. It also shows the convergence of the period, $0.1 \times T$, and the reduction in the error. Convergence of x_1 , —; convergence of x_2 , —; convergence of x_3 , —; the error,; the convergence of $0.1 \times T$,

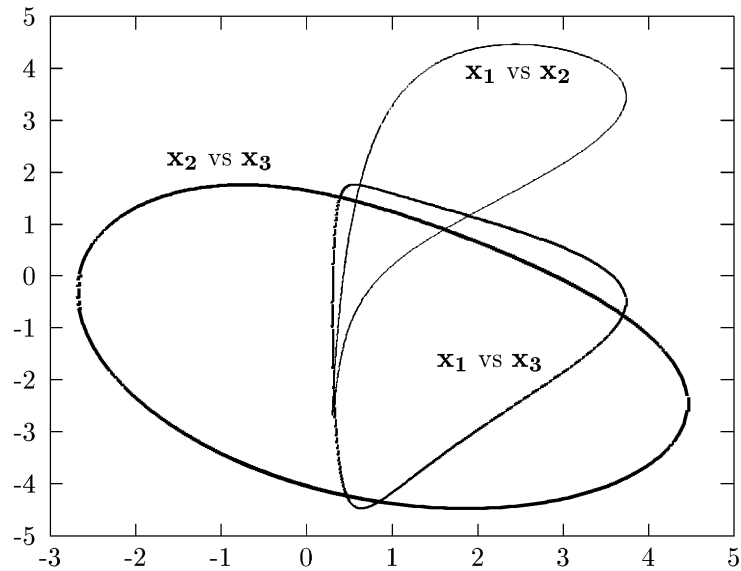


Fig. 4. The periodic orbit for the Rossler system in Example 2. x_1 -vs- x_2 , —; x_1 -vs- x_3 , —; x_2 -vs- x_3 , —.

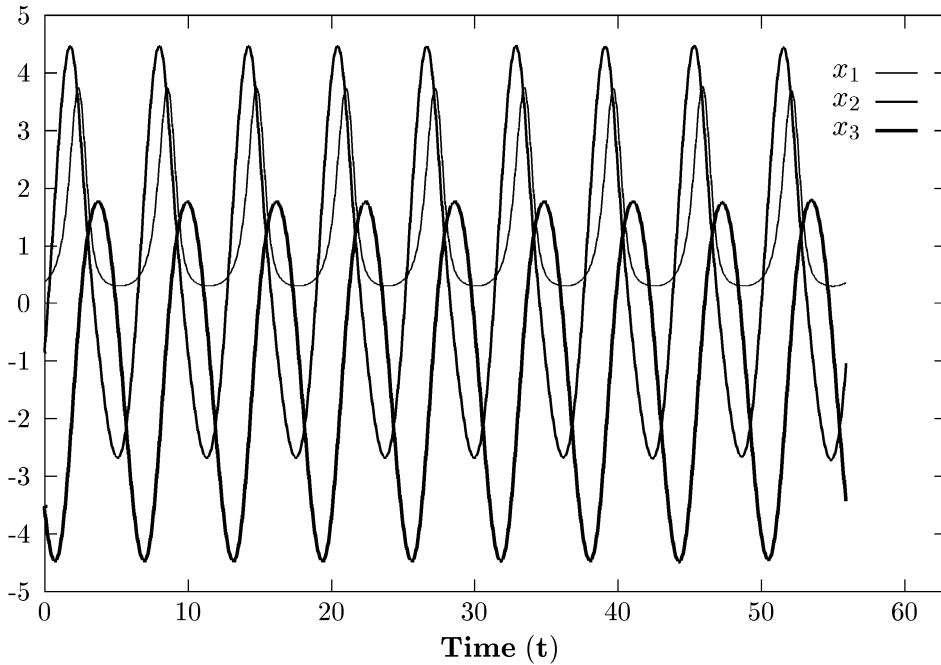


Fig. 5. The periodic solution for the Rossler system in Example 2. x_1 , —; x_2 , —; x_3 , —.

2.4. Example 3

We next study the dynamical system generated by the potential x^2y^2 . This system was considered to be ergodic until it was shown that it has at least one family of stable periodic orbits [15], (also references therein). For this example, we were not able to obtain convergence to a periodic orbit without the inclusion of the higher-order terms using the present scheme. The above potential generates a system which is given by

$$\dot{x}_1 = x_3, \quad \dot{x}_2 = x_4, \quad \dot{x}_3 = -x_1x_2^2, \quad \dot{x}_4 = -x_1^2x_2. \tag{25–28}$$

For this system, we need to include six conditions, namely, $(x_0 - x_T), \dots, (\ddot{x}_0 - \ddot{x}_T)$. Higher derivatives of the state and the corresponding gradient matrices can readily be computed. Fig. 6 shows the convergence of the initial condition x_0 as a function of the number of iterations. The iteration is started from $x_0 = (0.17, 0.00001, 0.0, 0.0)^T$. The constant parameters are chosen as $\alpha = 1. \times 10^3$ and $\beta = 1. \times 10^9$, and the initial guess for the period is $T = 30$. The iteration converges to $(0.170000000125, 0.98101038E - 5, 0.147458E - 9, -0.2168999E - 6)$ with the period converging to 36.957. Note that we are scaling the data in Fig. 6. Fig. 7 shows the periodic orbit as a function of time when the system is started from the calculated initial condition. By varying the initial guess for $x_1(0)$ we can obtain a number of periodic solutions in this region of the phase space. If the initial guess is chosen as $x_0 = (1., 0.00001, 0.0, 0.0)^T$, then the iterations converges to $(1.00000000003, 0.988722206E - 5, 0.7677829E - 9, -0.16914E - 14)$ with the period converging to 31.416. Fig. 8 shows the periodic solution for the calculated initial condition.

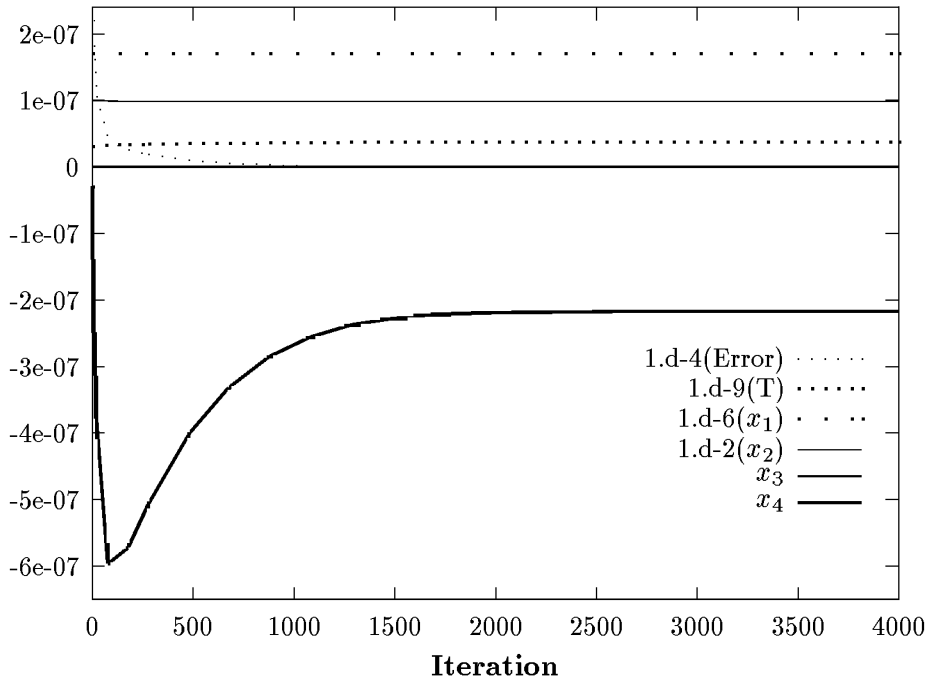


Fig. 6. The convergence of the initial condition to a point on the orbit as a function of the number of iterations for $x_2 y^2$ potential in Example 3. It also shows the convergence of the period, and the reduction in the error. Convergence of $(1.E - 4) \times \text{Error}$,; convergence of $(1.E - 9) \times T$,; convergence of $(1.E - 6) \times x_1$,; convergence of $(1.E - 2) \times x_2$, ———; convergence of x_3 , ———; convergence of x_4 , ———.

Note that the period for this orbit is $T = 6.283$ and the algorithm has converged to $T = 31.416$ which is equal to $5 \times T$. Fig. 9 shows a number of periodic orbits around this region. The calculated values for the initial conditions are given by

$$\begin{bmatrix} x_1 & 0.170000000125E + 0 & 0.229999999963E + 0 & 0.600000000006E + 0 & 0.100000000003E + 1 \\ x_2 & 0.981010380230E - 5 & 0.996545022230E - 5 & 0.999110789116E - 5 & 0.988722206005E - 5 \\ x_3 & 0.147458666332E - 9 & 0.159586800885E - 9 & 0.470383618896E - 9 & 0.767782938779E - 9 \\ x_4 & -0.2168999542E - 6 & 0.139189741819E - 6 & -0.2132584063E - 8 & -0.169144005E - 14 \end{bmatrix}.$$

By including higher-order conditions we are essentially reducing the space of the extremal points for the cost functional that corresponds to a periodic orbit.

3. Conclusion

In this note, we presented a method for the computation of the period orbit for dynamical systems. The approach is based on a variational principle which seeks to minimize a cost

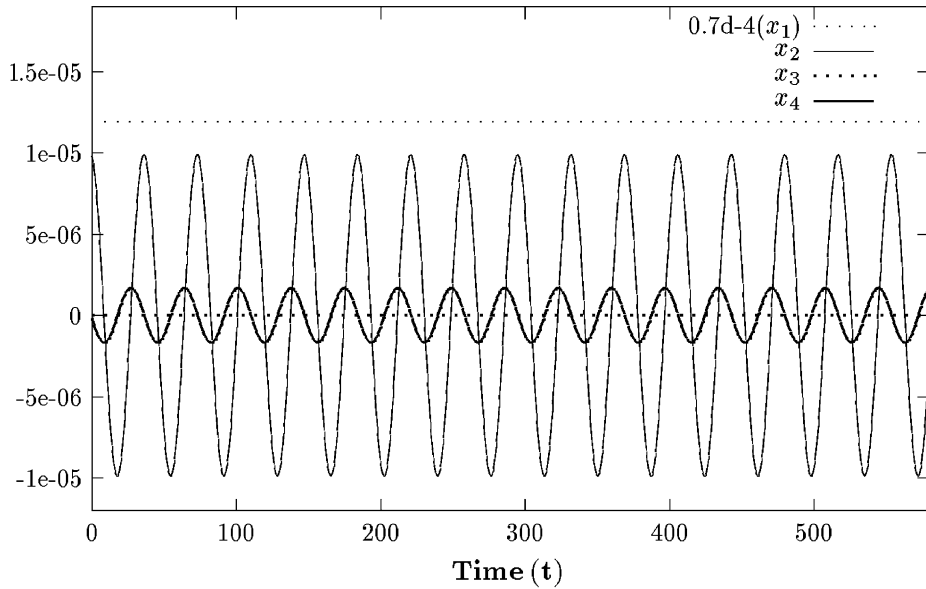


Fig. 7. A periodic solution for the x_2y^2 potential in Example 3 starting from the initial condition $(0.170000000125, 0.98101038E - 5, 0.147458E - 9, -0.2168999E - 6)$. Convergence of $(7.E - 5) \times x_1, \dots$; convergence of $x_2, \text{---}$; convergence of x_3, \dots ; convergence of $x_4, \text{---}$.

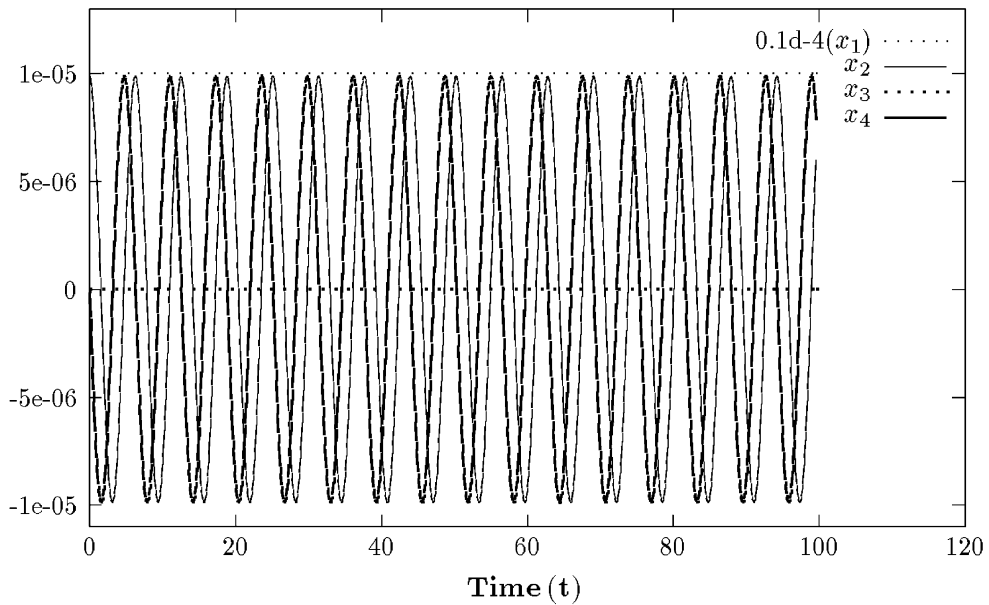


Fig. 8. A periodic solution for the x_2y^2 potential in Example 3 starting from the initial condition $(1.00000000003, 0.988722206E - 5, 0.7677829E - 9, -0.169141E - 14)$. Convergence of $(1.E - 5) \times x_1, \dots$; convergence of $x_2, \text{---}$; convergence of x_3, \dots ; convergence of $x_4, \text{---}$.

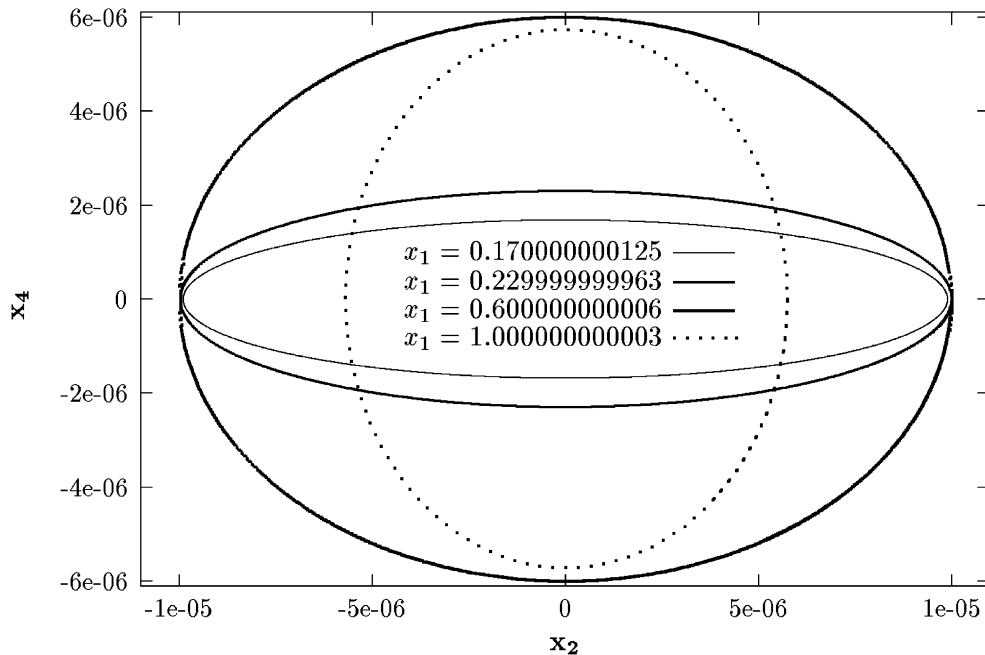


Fig. 9. A number of periodic solutions for the x_2y^2 potential in Example 3. $x_1 = 0.170000000125$, —; $x_1 = 0.229999999963$, —; $x_1 = 0.600000000006$, —; $x_1 = 1.000000000003$,

functional. The method is based on the definition of a periodic orbit and can be applied to a general dynamical system. We also presented a number of numerical examples. In all the cases the inclusion of the higher-order terms greatly improves the convergence of the algorithm to a solution.

References

- [1] J. Milstein, The inverse problem: estimation of kinetic parameters, in: K.H. Ebert, P. Deuflhard, W. Jäger (Eds.), *Modeling of Chemical Reaction Systems*, Springer, New York, 1981.
- [2] J.D. Murray, *Mathematical Biology*, Springer, New York, 1989.
- [3] L.O. Chua (Ed.), Special issue on nonlinear waves, patterns, and spatiotemporal chaos in dynamic arrays, *IEEE Transactions on Circuits and Systems* 42 (10) (1995).
- [4] M. Paskota, A.I. Mees, K.L. Teo, On control of chaos: higher periodic orbits, *Dynamics and Control* 5 (4) (1995) 365–388.
- [5] F. Verhulst, *Nonlinear Differential Equations and Dynamical Systems*, Springer, Berlin, Germany, 1996.
- [6] M. Kubicek, I. Schreiber, Computational analysis of nonlinear dynamical systems, *Zeitschrift für Angewandte Mathematik und Mechanik* 77 (S2) (1997) 603–604.
- [7] P. Deuflhard, Computation of periodic solutions of nonlinear odes, *BIT Numerical Mathematics* 24 (1984) 456–466.
- [8] S. Skelboe, Computation of the periodic steady-state response of nonlinear networks by extrapolation methods, *IEEE Transactions on Circuits and Systems CAS-27* 27 (1980) 161–175.

- [9] K. Lust, D. Roose, A. Spence, A.R. Champneys, An adaptive Newton–Picard algorithm with subspace iteration for computing periodic solutions, *SIAM Journal on Scientific Computing* 19 (1998) 1188–1209.
- [10] W. Schmidt, Limit cycle computation of oscillating electric circuits, in: R.E. Bank, R. Bulirsch, H. Gajewski, K. Merten (Eds.), *Mathematical Modelling and Simulation of Electrical Circuits and Semiconductor Devices*, Birkhauser, Basel, 1994, pp. 91–101.
- [11] V.T. Coppola, R.H. Rand, Averaging using elliptic functions: approximation of limit cycles, *Acta Mechanica* 81 (1990) 125–142.
- [12] J. Botina, H. Rabitz, Finding regular orbit, *Physical Review Letters* 75 (16) (1995) 2948–2951.
- [13] R. Fitzhugh, Impulse and physiological states in theoretical models of nerve membrane, *Biophysical Journal* 1 (1961) 445–466.
- [14] U. Parlitz, L. Junge, Subharmonic entrainment of unstable periodic orbits and generalized synchronization, *Physical Review Letters* 79 (17) (1997) 3158–3161.
- [15] P. Dahlqvist, G. Russberg, Existence of stable orbits in the x^2y^2 potential, *Physical Review Letters* 65 (23) (1990) 2837–2838.