ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 1, No. 2, 2006, *pp*. 85-92

Two Kinds of Important Numerical Methods for Calculating Periodic Solutions

Tiantian Qiao, Weiguo Li [+]

School of Mathematics and Computational Sciences, Petroleum University of China,
257061, Dongying, Shandong, P.R.China

(*Received December 26, 2005, Accepted March 1, 2006*)

Abstract. In this paper boundary value problem method and initial value problem method for periodic boundary value problem of ordinary differential equations are proposed. With use of these methods some numerical examples are computed by MATLAB .The numerical results show that these methods are very efficient.

Key words: ordinary differential equations, periodic boundary value problem, boundary value problem method, initial value problem method

1. Introduction

Ordinary differential equations (ODEs) describe phenomena that change continuously. They arise in models throughout mathematics, science, and engineering. By itself, a system of ODEs has many solutions. Commonly a solution of interest is determined by specifying the values of all its components at a single point $x = a$. This is an initial value problem (IVP). However, in many applications a solution is determined in a more complicated way. A boundary value problem (BVP) specifies values or equations for solution components at more than one x . Unlike IVPs, a boundary value problem may not have a solution, or may have a finite number, or may have infinite many. Because of this, programs for solving BVPs require users to provide a guess for the solution desired. Often there are parameters desired. We aim to solve a typical BVP as easy as possible.

The periodic boundary values problems is one kind of important BVP (PBVP). The periodic boundary conditions are a kind of non-separated boundary conditions. They are much harder to solve than IVPs and other BVPs. Any solver to PBVPs might fail, even with a good guesses for the solution and unknown parameters. The aim of this article is to give several numerical methods to solve PBVPs with MATLAB programs and shows how to formulate, solve, and plot the solution of a PBVP with the MATLAB programs.

We discuss the following PBVP:

$$\begin{cases} y^{(n)}(t) + f(t, y, y', \cdots, y^{(n-1)}) = 0 \\ y^{(i)}(0) = y^{(i)}(T), i = 0,1, \cdots, n-1 \end{cases} \tag{1.1}$$

where $1 < n \in N$, f is a continuous nonlinear function and periodic T is known or unknown.

2. Solve PBVPs with Boundary Value Problem Method

In MATLAB, there is a program, *bvp4c*, which is an effective solver to BVPs. But the underlying method and the computing environment are neither appropriate for high accuracies nor for problems with extremely sharp changes in their solutions. Although *bvp4c* accepts quite general BVPs, problems arise in the most diverse forms and they may require some preparation for their solutions. Sometimes, solving PBVPs may well involve an exploration of the unique existence of a model. This is quite different from solving IVPs.

Shooting method can be used to solve BVPs, but *bvp4c* is not a shooting code. *bvp4c* implements a collocation method for the solution of BVPs[1~4]. The general idea of the collocation method is that

[+] liwg20022004@yahoo.com.cn

approximate solution of question is denoted by $y_m(t)=\sum_{i=1}^{m}C_i\varphi_i(t)$, where $\varphi_i(t), i=1,\cdots,m$ is basic function and $C_i, i=1,\cdots,m$ is undetermined coefficient. For this approximate form, the initial equation becomes linear (nonlinear) equations about the unknown $C_i, i=1,\cdots,m$. The general basic functions are Lagrange polynomial, Legendre polynomial and so on. In MATLAB, *bvp4c* solve problems by collocation method. Suppose $S(t)$ is the approximate solution of boundary value problem which is continuous on $[0,T]$ and is cubic polynomial on every subinterval $[t_n,t_{n+1}]$, where $0=t_0<t_1<\cdots<t_N=T$. Then $S(t)$ satisfies at both ends and the midpoint of each subinterval:

$$\begin{cases} S'(t_n)=f(t_n,S(t_n)) \\ S'(t_{n+1})=f(t_{n+1},S(t_{n+1})) \\ S'((\frac{t_n+t_{n+1}}{2})=f((\frac{t_n+t_{n+1}}{2}),S((\frac{t_n+t_{n+1}}{2})) \end{cases}.$$

So one can solve the coefficient of polynomial on each subinterval $[t_n,t_{n+1}]$ by solving nonlinear equations and get the approximate solution of the initial problem. We can control the size of the residual $r(t)=S'(t_n)-f(t_n,S(t_n))$ very good using collocation method in *bvp4c.* For the failed initial guess, the procedure can regulate a little to get better approximate solution.

Non-autonomous ordinary differential equation can be transformed into autonomous ordinary differential equation by adding one order. Therefore we just discuss the periodic boundary value problem of autonomous ordinary differential equation:

$$\begin{cases} y^{(n)}(t)+f(y,y',\cdots,y^{(n-1)})=0 \\ y^{(i)}(0)=y^{(i)}(T), i=0,1,\cdots,n-1 \end{cases} \tag{2.1}$$

where $1<n\in N$, f is a continuous nonlinear function and periodic T is known or unknown.

2.1. Computing Periodic Solution of ODEs with *bvp4c*

Solve problem (2.1) with *bvp4c* when periodic T is known, the program is very simple and just needs to construct some simple functions. Basic solving sentence:

SOLINIT = *bvpinit*(X,YINIT)

SOL=*bvp4c*(ODEFUN,BCFUN,SOLINIT,OPTIONS).

So we obtain SOL.Y which is periodic solution of problem (2.1) about periodic T . In order to use the above sentence we give the initial guesses of X and Y. They don't need to be very accurate. For example: X=LINSPACE(0,T,20),YINIT=[SIN(X),COS(X)] and so on. Furthermore, we also need to construct two functions ODEFUN and BCFUN. They are high order ordinary differential equation be written as a system of first order ODEs and residual values of boundary conditions, respectively.

Example1. Solve the following second order PBVP:

$$\begin{cases} x''+(x+0.5\pi)\sin(t)^2=e^{-1}\cos(t)\sin(2t)-\cos(t)^3 \\ x(0)=x(2\pi) \\ x'(0)=x'(2\pi) \end{cases}.$$

The computational results are recorded in Table2. According to the computational results the limit cycle of the model are drawn in Figure 1.

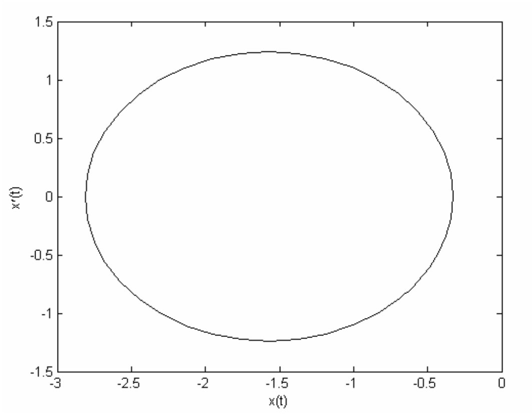

Fig.1 The limit cycle

Example2. Solve the following fourth order PBVP:

$$\begin{cases} x^{(4)} + x'''x'' - 16x'x - 16\sin(2t) = 0 \\ x(0) = x(\pi) \\ x'(0) = x'(\pi) \\ x''(0) = x''(\pi) \\ x'''(0) = x'''(\pi) \end{cases}.$$

The computational results are recorded in Table1. According to the computational results the periodic solution curve and the limit cycle of the model are drawn in Figure 2.

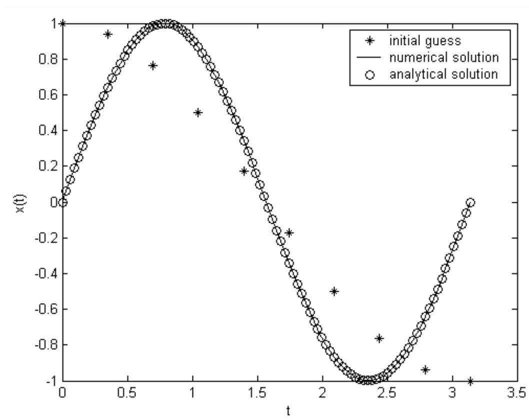

Fig.2(a) The numerical solution

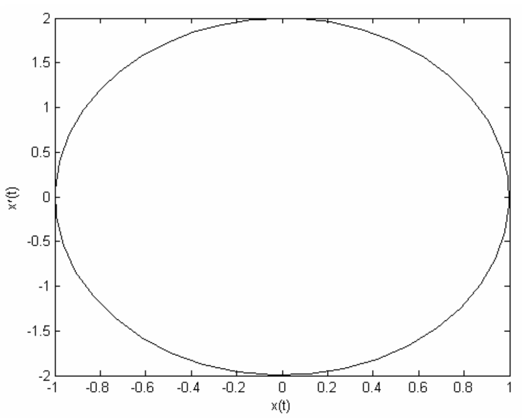

Fig.2(b) The limit cycle

2.2. Solve the Periodic Boundary Value Problems when Periodic is Unknown

When periodic T is unknown, solving the problem becomes complex and is sensitive to the guess of initial parameter. For the convenience of solving, we transform problem (2.1) and let $\tau = \frac{t}{T}$. So problem (2.1) becomes (2.2) which is as follows:

$$\begin{cases} y^{(n)}(\tau) + Tf(y, y', \cdots, y^{(n-1)}) = 0 \\ y^{(i)}(0) = y^{(i)}(1), i = 0,1,\cdots,n-1 \end{cases} \tag{2.2}$$

where $1 < n \in N$, f is a continuous nonlinear function, and periodic T is unknown.

Solving basic sentence as follows:

SOLINIT = *bvpinit*(X,YINIT,PARAMS)

SOL=*bvp4c*(ODEFUN,BCFUN,SOLINIT,OPTIONS).

Compared with the case of the known periodic T , it adds the term of parameter PARAMS. This parameter is initial guess of periodic T .

Example3. Solve the following second order PBVP when periodic is unknown

$$\begin{cases} y_1' = 3\left(y_1 + y_2 - \frac{1}{3} y_1^3 - 1.3 \right) \\ y_2' = -(y_1 - 0.7 + 0.8 y_2)/3 \\ y_1(0) = y_1(T) \\ y_2(0) = y_2(T) \end{cases}.$$

The computational results are recorded in Table2. According to the computational results the periodic solution curve and the limit cycle of the model are drawn in Figure 3.

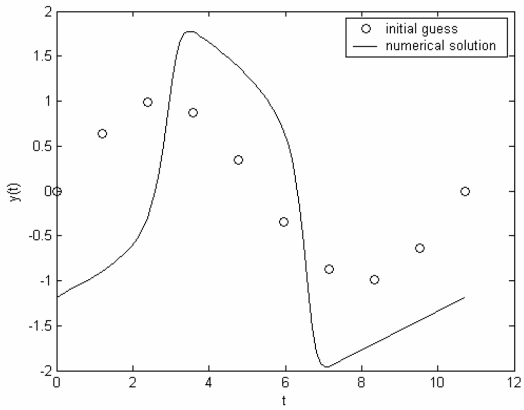

Fig.3(a) The numerical solution

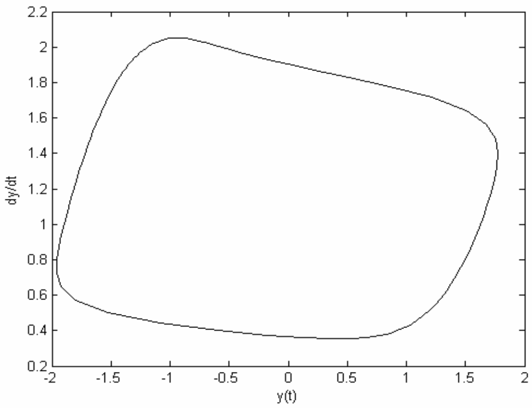

Fig.3(b) The limit cycle

Example4. Solve the following third order PBVP when periodic is unknown

$$\begin{cases} y''' + yy'' + y' - (y')^2 + 1 = 0 \\ y(0) = y(T) \\ y'(0) = y'(T) \\ y''(0) = y''(T) \end{cases}.$$

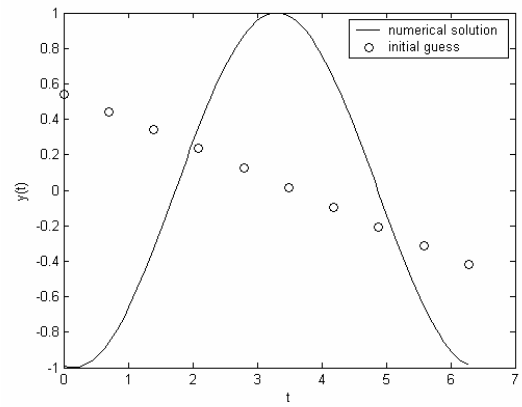

Fig.4(a) The numerical solution

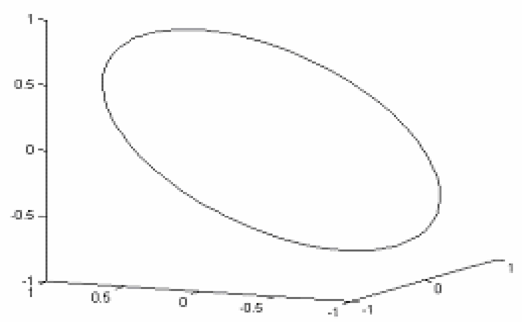

Fig.4(b) The limit cycle

The computational results are recorded in Table2. According to the computational results the periodic solution curve and the limit cycle of the model are drawn in Figure 4.

3. Solve PBVPs with Initial Value Problem Method

Applying the principle of shooting method, [5~6] described the initial value problem method to solve the periodic boundary value problem. If we solve the following differential equation with periodic boundary condition

$$\begin{cases} y^{(n)}(t) + f(t, y, y', \cdots, y^{(n-1)}) = 0 \\ y^{(i)}(0) = y^{(i)}(T), i = 0,1,\cdots, n-1 \end{cases} \tag{3.1}$$

where $1 < n \in N$, f is a continuous nonlinear function, and periodic T is known.

At first, we suppose α is n dimension vector and $\alpha_{i+1} = y^{(i)}(0), i = 0,1,\cdots, n-1$. Because $f(t, y, y', \cdots, y^{(n-1)})$ is continuous function, so we can get the solution of the initial value problem

$$\begin{cases} y^{(n)}(t) + f(t, y, y', \cdots, y^{(n-1)}) = 0 \\ y^{(i)}(0) = \alpha_{i+1}, i = 0,1,\cdots, n-1 \end{cases} . \tag{3.2}$$

Its solution is denoted by $y(t,\alpha)$. Furthermore, suppose $g(\alpha)$ is n dimension vector and $g^{i+1}(\alpha) = y^{(i)}(T,\alpha), i = 0,1,\cdots, n-1$. To satisfy periodic boundary condition of equation (3.1), the value of $y^{(i)}(t,\alpha), i = 0,1,\cdots, n-1$ at $t = 0$ and at $t = T$ is equal. Namely $y^{(i)}(0,\alpha) = y^{(i)}(T,\alpha), i = 0,1,\cdots, n-1$. So solving periodic boundary value problem becomes solving the fixed point of the following problem

$$g(\alpha) = \alpha \tag{3.3}$$

In fact, that is solve nonlinear equations

$$G(\alpha) = g(\alpha) - \alpha = 0 \tag{3.4}$$

Therefore, solving periodic boundary value problem of ordinary differential equation becomes solving nonlinear equations. There are many methods to solve nonlinear equations and we use quasi-Newton method with rank one correction. Let $A = Jacobi(G)$, iterative formula is denoted by

$$\begin{cases} \alpha_{k+1} = \alpha_k - A_k^{-1} G_k \\ \Delta\alpha_k = \alpha_{k+1} - \alpha_k \\ \Delta G_k = G_{k+1} - G_k \\ \Delta A_k = \dfrac{(\Delta G_k - A_k \Delta\alpha_k)\Delta\alpha_k^T}{\Delta\alpha_k^T \Delta\alpha_k} \\ A_{k+1} = A_k + \Delta A_k \end{cases} \tag{3.5}$$

Thus, solution α^* obtained is the approximate initial value of problem (3.1).

The algorithm:

Step 0: Given initial vector $\beta_0 \in R^n$ and a small positive number $0 < \bar{\delta} < 1$;

Step 1: Compute initial matrix $A_0 = Jacobi(G)$;

Step 1.1: Given a perturbation matrix $\delta \in R^{n\times n}$ and it is hold $\delta_{i,j} = \begin{cases} \bar{\delta} & i = j \\ 0 & i \neq j \end{cases}$, let a systems of vector $\beta_1, \beta_2, \cdots, \beta_n$, it is hold $\beta_k = \delta_k + \beta_0$, where δ_k is k-th column vector of matrix δ;

Step 1.2: Compute initial vector $\beta_0, \beta_1, \beta_2, \cdots, \beta_n$ with *ode45* and obtain its values at $t = T$ which are denoted by $g0_i, i = 0,1,\cdots, n$;

Step 1.3: Compute $A_0 = Jacobi(G) = g' - I$ by replacing derivative g' by finite-difference approximation, $g'_{i,j} = \frac{\partial g_i}{\partial \beta_j} = \frac{g0_j(i) - g0_0(i)}{\beta_j(j) - \beta_0(j)} = \frac{g0_j(i) - g0_0(i)}{\bar{\delta}}$ for $i, j = 1, \cdots, n$;

Step 2: Solve nonlinear equations and solution α^* is obtained

Step2.1: Let $\alpha_0 = \beta_0, G_0 = g0_0 - \alpha_0$, a small positive number $0 < \varepsilon << 1$ and $k := 0$;

Step2.2: Compute first formula of (3.5), let obtained α_{k+1} is initial value of IVP, then compute (3.2) with *ode45* and obtain its solution $Y(\alpha_{k+1})$, vector $g_{k+1} = Y(T, \alpha_{k+1})$, via (3.4) G_{k+1} is obtained, then compute other formulae in (3.5) and obtain $\Delta\alpha_k, \Delta G_k, \Delta A_k, A_{k+1}$, let $k = k + 1$. While $\|G_{k+1}\| \le \varepsilon$, stop;

Step3: The solution α^* obtained by **Step2** is approximate initial value of (3.1), then solve initial value problem (3.2) with *ode45* and we can obtain numerical solution $y^*(t)$ of (3.1).

Example5:

With use of the initial value problem method example 1 is computed, the computational results are recorded in Table3. According to the computational results the limit cycle of the model are drawn in Figure 5.

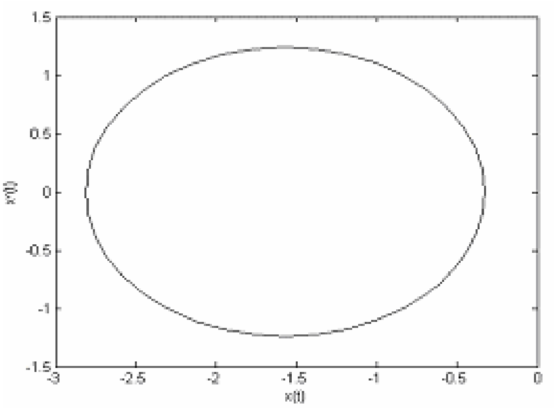

Fig.5 The limit cycle

Example6:

With use of the initial value problem method example 2 is computed, the computational results are recorded in Table3. According to the computational results the periodic solution curve and the limit cycle of the model are drawn in Figure 6.

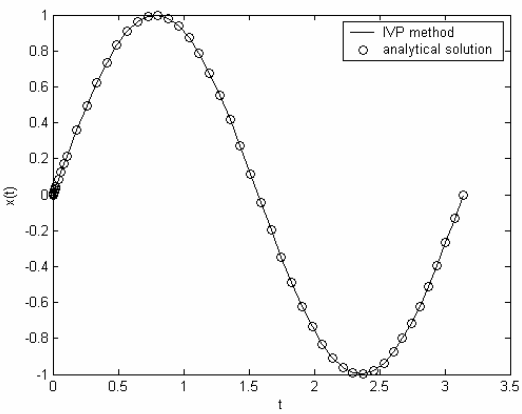

Fig.6(a) The numerical solution

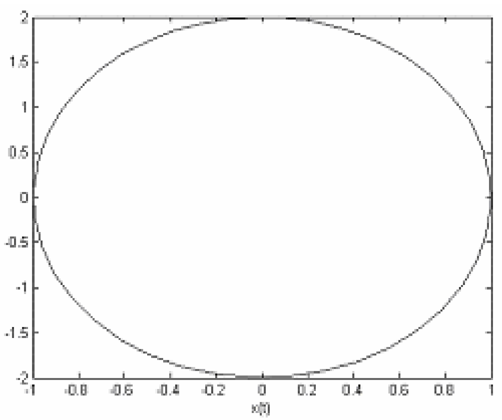

Fig.6(b) The limit cycle

4. Conclusion

Numerical examples show that these methods are very efficient for solving PBVPs, and they have their own characteristic.

(a) The boundary value problem method can solve complex case in which the number of unknown is more than the number of equations because periodic is unknown by introducing parameter, but initial value problem method can't solve this case.

(b) For initial value problem method, we solve the periodic boundary value problem with the initial value problem method. This is not only a good idea but also an efficient method solving the hard periodic boundary value problem.

(c) Generally speaking, though accuracy of initial value problem method and boundary value problem method isn't very high, we can accept it. During the computation of initial value problem method, storage and computational time decrease greatly by replacing derivative by finite-difference approximation in quasi-Newton method.

All in all, solving problem (1.1) using initial value problem method and boundary value problem method, we should combine the problem itself and the characteristic of the methods.

Table 1 Computational results

	Example 1	**Example 2**
Computational time	0.35100000000000	0.94200000000000
Error	1e-6	1e-6

Table 2 Computational results

	Example 3	**Example 4**
Computational time	4.69700000000000	0.92200000000000
Error	1e-13	1e-3
Periodic T	10.71080854662410	6.28318532363360

Table 3 Computational results

	Example 5	**Example 6**
Computational time	0.18100000000000	0.28000000000000
Error	1.0e-015	1.0e-13
Iterative count	k =13	k =14

5. References

[1] J. Kierzenka, Studies in the Numerical Solution of Ordinary Differential Equations, Southern Methodist University, Dallas, TX, 1998.

[2] U. Ascher, R. Mattheij, and R. Russell, Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, SIAM, Philadelphia, PA, 1995.

[3] H. Keller, Numerical Methods for Two-Point Boundary Value Problems, Dover, New York, 1992.

[4] The MATHWORKS, INC., Using MATLAB, 24 Prime Park Way, Natick, MA, 1996.

[5] Weiguo Li, Solving the Periodic Boundary Value Problem with the Initial Value Problem Method, Journal of

Mathematical Analysis and Applications, 226(1998)1, 259-270.

[6] Weiguo Li and Zuhe Shen, A Constructive Proof of Existence and Uniqueness of 2*pi-Periodic Solution to Duffing Equation, Nonlinear Analysis, Ser. A : Theory Methods, 42(2000)7, 1209-1220.