Int. J. Nonlinear Anal. Appl. In Press, 1–5
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2024.34368.5133

Solving time delay two-point boundary value problems with shooting continuous Runge-Kutta methods

Tahereh Jabbari Khanbehbin

Department of Mathematics and Statistics, Faculty of Basic Sciences, Gonbad Kavous University, Gonbad Kavous, Iran

(Communicated by Zakieh Avazzadeh)

Abstract

This paper presents an efficient method for solving time delay two-point boundary value problems. A combination of the Continuous Runge-Kutta method and the Newton shooting method has been used for solving this problem. The efficiency, high accuracy and the rate of convergence of the proposed method are illustrated by an example.

Keywords: time delay two-point boundary value problems, continuous Runge-Kutta methods, shooting methods
2020 MSC: 34K06, 34K28, 68T05

1 Introduction

Delay Differential Equations (DDEs) have played the important role in modeling the real issues, because the presence of time delay in many phenomena around us cannot be ignored. So to express the behavior of such phenomenawe need more complex systems. Some applications of these problems can be seen in chemistry, electronics, medicine, engineering, biology, economics, etc. Also, obtaining an analytical solution for these systems is associated with many complications. Therefore, from the distant past researchers are always trying finding approximate solutions to answer these problems. You can see some of this works in [2, 3, 4, 6, 7] and [8]A wide class of such equations is represented by the following general form:

$$\begin{cases} \dot{x}(t) = f(t, x(t), x(t-\tau(t))), & t > t_0, \\ x(t) = \phi(t), & t_0 - \tau(t_0) \leq t \leq t_0, \end{cases} \tag{1.1}$$

where $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n, \tau(t) \geq 0$ and $\phi \in C^0[t_0 - \tau(t_0), t_0]$ represents the initial information of the state variable x.

One of the most famous and efficient methods for solving DDEs, is the Continuous Runge-Kutta (CRK) methods [4]. CRK methods was originally designed to treat initial value problems for ODEs. For the first time, Zennaro [5] showed that for every discrete s-stage RK method of order p, there is a CRK method of degree d, if there exist s

Email address: jabbari.tahere@gmail.com (Tahereh Jabbari Khanbehbin)

Received: April 2024 *Accepted:* June 2024

polynomials $b_i(\theta), i = 1, ..., s$ of degree $\leq d$, independent of the f, as follows,

$$u(t_n + \theta h_{n+1}) = x_n + h_{n+1} \sum_{i=1}^{s} b_i(\theta) k_i, \quad 0 \leq \theta \leq 1,$$
$$k_i = f(t_n^i, x_n + h_{n+1} \sum_{j=1}^{s} a_{ij} k_j), \quad i = 1, \ldots, s,$$

where $c_i = \sum_{j=1}^{s} a_{ij}, t_n^i = t_n + c_i h_{n+1}, i = 1, \ldots, s$, and $h_{n+1} = t_{n+1} - t_n$ and $\{c_i, a_{ij}\}$'s are the same as the coefficients of the discrete Runge–Kutta method. This idea was then generalized to DDEs by Baker and Paul [1] for a general DDE of the form (1.1) as follows

$$u(t_n + \theta h_{n+1}) = x_n + h_{n+1} \sum_{i=1}^{s} b_i(\theta) k_i, \quad 0 \leq \theta \leq 1, \tag{1.2}$$
$$k_i = f(t_n^i, X_i, u(t_n^i - \tau(t_n^i))), \quad i = 1, \ldots, s, \tag{1.3}$$
$$X_i = x_n + h_{n+1} \sum_{j=1}^{s} a_{ij} k_j, \quad i = 1, \ldots, s. \tag{1.4}$$

When the delay is constant and $h_{n+1} \leq \tau$, then $u(t_n^i - \tau)$ is known from the previous steps for any i $(0 \leq c_i \leq 1)$, so this method is an explicit CRK method. The aim of our paper is to use the CRK methods to solve the delay Two-Point Boundary Value Problems (TPBVPs). For this pupose we combine Newton Shooting Method with CRK metods(NSCRK). We present the main results of our research in the next section and we show the efficiency of proposed method by an example in Section 3. In 4, conclusions and suggestions for future works are presented.

2 NSCRK for problem

Consider the following basic form of a first order linear TPBVP with time-delay,

$$\begin{cases} \dot{x}(t) = f_1(t, x(t), y(t), x(t-\tau), y(t-\tau)), & t_0 \leq t \leq t_f \\ \dot{y}(t) = f_2(t, x(t), y(t), x(t-\tau), y(t-\tau)), & t_0 \leq t \leq t_f \\ x(t) = \phi(t), & t_0 - \tau \leq t \leq t_0, \\ y(t_f) = \beta. \end{cases} \tag{2.1}$$

For solving this problem we need to solve the following initial value problem,

$$\begin{cases} \dot{x}(t) = f_1(t, x(t), y(t), x(t-\tau), y(t-\tau)), & t_0 \leq t \leq t_f \\ \dot{y}(t) = f_2(t, x(t), y(t), x(t-\tau), y(t-\tau)), & t_0 \leq t \leq t_f \\ x(t) = \phi(t), & t_0 - \tau \leq t \leq t_0, \\ y(t_0) = z. \end{cases} \tag{2.2}$$

involving a parameter z. We do this by choosing the parameters $z = z_k$ so that

$$\lim_{k \to \infty} y(t_f, z_k) = y(t_f) = \beta,$$

where $y(t, z_k)$ denotes the solution to the IVP (2.2) and $y(t)$ denotes the solution to the BVP (2.1). We try to make the value of $y(t_f, z_k)$ as close to β as possible by adjusting the value of z_k. This technique is called Shooting method. For determining the parameters $z_k,\ k = 1, 2, ...$, we must solve this problem:

$$y(t_f, z) - \beta = 0,$$

To solve this equation, we can use the secant method and for more accurate appoximations in nonlinear systems, we can use the more powerful Newton's method to generate the sequence $\{z_k\}$, the iterations are generated by newton formula $z_{k+1} = z_k - \frac{y(t_f, z_k) - \beta}{\frac{\partial y}{\partial z}(t_f, z_k)},\ k = 2, 3, \cdots$. and requiers only one initial approximation z_1, but there isn't one explicit representation for $y(t_f, z)$. In order to solve this difficulty, by assuming that the solution depends on both t and z, we first take the partial derivative of (2.2) with respect to z and by simplify the notations as follow:

$$v(t) = \frac{\partial x(t,z)}{\partial z}, \quad w(t) = \frac{\partial y(t,z)}{\partial z},$$

and

$$v(t-\tau) = \frac{\partial x(t-\tau,z)}{\partial z}, \quad w(t-\tau) = \frac{\partial y(t-\tau,z)}{\partial z}.$$

We have

$$\begin{cases} \dot{v}(t) = \frac{\partial f_1}{\partial x(t,z)} \cdot v(t) + \frac{\partial f_1}{\partial y(t,z)} \cdot w(t) + \frac{\partial f_1}{\partial x(t-\tau,z)} \cdot v(t-\tau) + \frac{\partial f_1}{\partial y(t-\tau,z)} \cdot w(t-\tau), \\ \dot{w}(t) = \frac{\partial f_2}{\partial x(t,z)} \cdot v(t) + \frac{\partial f_2}{\partial y(t,z)} \cdot w(t) + \frac{\partial f_2}{\partial x(t-\tau,z)} \cdot v(t-\tau) + \frac{\partial f_2}{\partial y(t-\tau,z)} \cdot w(t-\tau), \\ v(0) = 0, \quad w(0) = 1. \end{cases} \tag{2.3}$$

for each iteration, IVP (2.3) should be solved with IVP (2.2). by replacing $w(t_f) = \frac{\partial y(t_f, z_{k-1})}{\partial z}$ in the newton's formula, we have

$$z_k = z_{k-1} - \frac{y(t_f, z_{k-1}) - \beta}{w(t_f)}, \tag{2.4}$$

Now, we present the following algorithm to solve (2.2) and (2.3) by shooting CRK method.

Algorithm 1 shooting CRK method for time-delay TPBVP

Step 1. Set N, $h = \frac{t_f - t_0}{N}$, $K = 1$, M (the maximum number of iterations), and s (the number of stages of CRK method), choose z_1, and tolerance error bound ϵ.

Step 2. While $L \leq M$, do

- $x_0 = \alpha$, $y(t_0) = z_1, \nu_0 = 0$ and $w_0 = 1$,

  Step 3. For $k = 1, 2, \cdots$,

  - solve (2.2) and (2.3) using the delay CRK method (1.2), (1.3) and (1.4).

    Step 4. Check the stop condition,

    - If $|y_N - \beta| < \epsilon$, then the procedure is complete and jump to **Step 6**,
    - else, go to the step 5.

    Step 5. Calculate the next approximation for z_{k+1} from (2.4), set $y(t_0) = z_{k+1}$, and back to **Step 3**.

  - end for

  Step 6. Stop the algorithm and output (t_n, x_n, y_n).

- end while

Step 7. Output (maximum number of iterations exceeded) and stop.

3 Illustrative examples

In this section, we examine a delay problem of the first order.

Example 3.1. Consider the following first-order linear TPBVP with time delay

$$\begin{cases} x'(t) = 10x(t-\tau)y(t-\tau) - \frac{1}{2}x(t) - \frac{1}{10}x^2(t), \quad 0 \leqslant t \leqslant t_f, \\ y'(t) = y(t)\big(1 - y(t)\big) - x(t)y(t), \quad 0 \leqslant t \leqslant t_f, \\ x(t) = 1 \quad -\tau \leq t \leq 0, \\ y(t_f) = \beta. \end{cases} \tag{3.1}$$

We solve this problem by applying Algorithm 1. For this purpose, we use the explicit Runge–Kutta method of discrete order $p = 4$(RK4) with the following coefficients:

$$\begin{array}{c|cccc} 0 & 0 & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & & \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \\ \frac{1}{2} & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Moreover, for the CRK method, set

$$b_1(\theta) = \frac{1}{2}\theta^2 + \frac{2}{3}\theta, \qquad b_3(\theta) = \frac{1}{3}\theta,$$
$$b_2(\theta) = \frac{1}{3}\theta, \qquad b_4(\theta) = \frac{1}{2}\theta^2 - \frac{1}{3}\theta.$$

We solve this problem with $t_f = 100$ and we choose two delays $\tau = 0.25$ and $\tau = 5$. The approximate values of the variables $x(t)$ and $y(t)$ with two delays $\tau = 0.25$ and $\tau = 5$ can be seen in the figures 1, 2, 3 and 4, respectively.

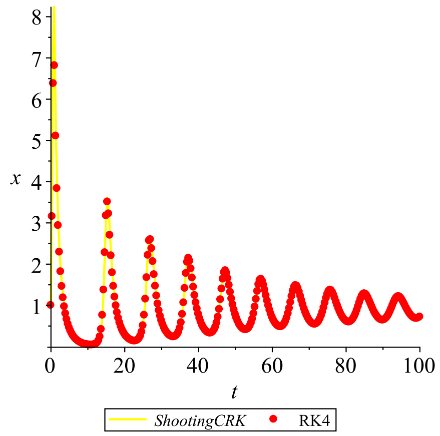

Figure 1: The approximate values of $x(t)$ with shooting continuose and discrete RK4 methods with $\tau = 0.25$ in Example 3.1

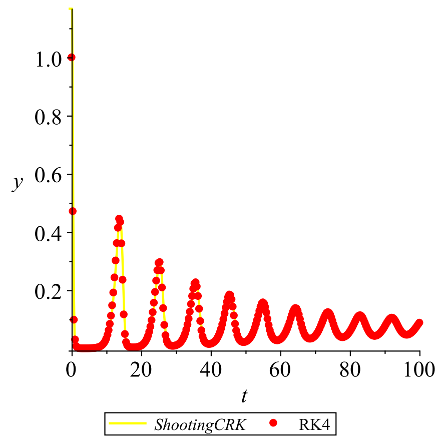

Figure 2: The approximate values of $y(t)$ with shooting continuose RK and discrete RK4 methods with $\tau = 0.25$ in Example 3.1

4 Conclusion

We employed a combination of CRK and shooting methods to solve delay TPBVPs. The original problem is transformed into a sequence of initial value problems with time delay then we solved these problems by NSCRK method. For the future works the time delay equations with time dependent delay can be evaluated with NSCRK method.

References

[1] Ch. Baker and Ch. Paul, *Parallel continuous Runge-Kutta methods and vanishing lag delay differential equations*, Adv. Comput. Math. **1** (1993), no. 3, 367–394.

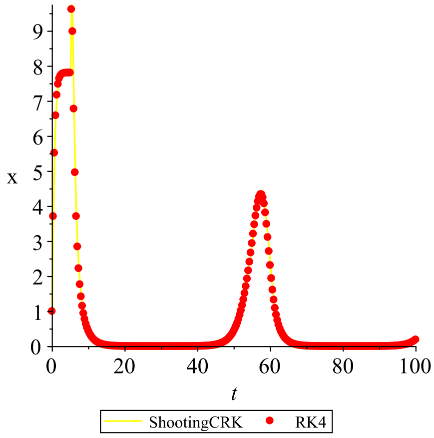

Figure 3: The approximate values of $x(t)$ with shooting continuose RK and discrete RK4 methods with $\tau = 5$ in Example 3.1

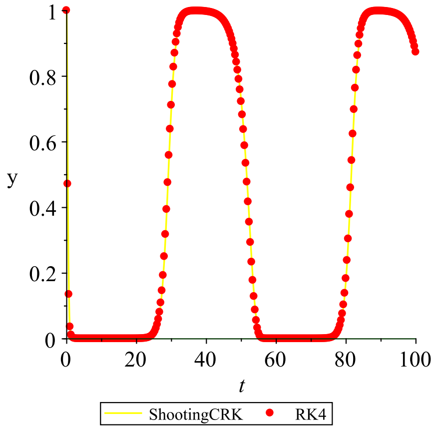

Figure 4: The approximate values of $y(t)$ with shooting continuose RK and discrete RK4 methods with $\tau = 5$ in Example 3.1

[2] C. Baker, C. Paul, and D. Wille, *A bibliography on the numerical solution of delay differential equations*, Technical Report 269, University of Manchester, 1995.

[3] M. Behroozifar and S.A. Yousefi, *Numerical solution of delay differential equations via operational matrices of hybrid of block-pulse functions and Bernstein polynomials*, Comput. Meth. Differ. Equ. **1** (2013), no. 2, 78–95.

[4] A. Bellen and M. Zennaro, *Numerical Methods for Delay Differential Equations*, Oxford University Press, 2013.

[5] M. Zennaro, *Natural continuous extensions of Runge-Kutta methods*, Math. Comput. **46** (1986), no. 173, 119–133.

[6] A. El-Safty,M.S. Salim, and M.A. El-Khatib, *Convergence of the spline function for delay dynamic system*, Int. J. Comput. Math. **80** (2003), no. 4, 509–518.

[7] D.J. Evans and K.R. Raslan, *The Adomian decomposition method for solving delay differential equation*, Int. J. Comput. Math. **82** (2005), no. 1, 49–54.

[8] E. Fridman, *Introduction to Time-Delay Systems: Analysis and Control*, Springer, 2014.