AUT Journal of Mathematics and Computing

AUT J. Math. Comput., 6(1) (2025) 1-8
https://doi.org/10.22060/AJMC.2023.22596.1180

Original Article

An efficient computational approach for numerical solution of non-smooth dynamical systems

Mohammad Ali Mehrpouya*[a], Hossein Heidary[b]

[a]*Department of Mathematics, Tafresh University, P.O. Box 39518-79611, Tafresh, Iran*
[b]*Department of Mechanical Engineering, Tafresh University, P.O. Box 39518-79611, Tafresh, Iran*

ABSTRACT: In this paper, an efficient computational approach based on the fantastic Simpson integral formula is developed for the numerical solution of non-smooth dynamical equations. In the proposed approach, at first, the integral reformulation of the target problem is intended. Then, the Simpson formula is employed to discretize the obtained integral equation. It is mentioned that, the implementation of the method is simple, so, the method can be simply and quickly used to solve a wide variety of non-smooth dynamical systems arising in the various engineering models. Numerical experiments of two benchmark examples are presented at the end and the efficiency of the method is reported.

Review History:
Received:03 August 2023
Revised:17 November 2023
Accepted:18 November 2023
Available Online:01 December 2024

Keywords:
Non-smooth dynamical equation
Integral equation
Simpson formula
System of algebraic equations

MSC (2020):
49J52; 45D05; 65R20; 65H10

1. Introduction

One of the classical and challenging subjects in the dynamical systems theory is the non-smooth dynamical systems. The non-smooth dynamical systems whose right hand side of their dynamical systems may not be differentiable everywhere are utilized to model a wide variety of phenomenon, especially in electrical circuits, mechanical and control systems. For instance, we can refer to the mechanical systems with impact and friction [1], the electrical circuits with diodes and transistors [20] and control systems where the control functions jump from one boundary to another [15, 16]. These are some of the well-known applications in which the non-smooth dynamical systems play an important role. It is worthwhile to note that, in the literature, the non-smooth dynamical systems have been surveyed from two aspects. The first one is, those valuable researches that are related to the existence and uniqueness of the solution of the problem. In this regard, we owe a lot to the contribution provided by Filippov [7], Leine [9, 10] and Leine et al. [11, 12, 13]. The second aspect is, those few works which have only focused on problem solving. It is necessary to mention that, due to non-smoothness in the right hand side of their dynamical

**Corresponding author.*
E-mail addresses: mehrpouya@tafreshu.ac.ir, heidary@tafreshu.ac.ir

systems, the numerical solution of non-smooth dynamical systems is a difficult task. In the following, some of these well-known numerical methods are referred.

Erfanian et al. [6] proposed an approach for solving non-smooth continuous and discontinuous ordinary differential equations which is based on the Taylor expansion. Thus a generalized derivative was used for many cases where the points of non-differentiability of the function are not known.

Ghaznavi and Noori Skandari [8] developed a new approach for solving non-smooth dynamical systems by converting the non-smooth system to the smooth form, using Chebyshev interpolation. Then, the smooth system is solved using Chebyshev pseudo-spectral method.

In [19], a finite difference approach for the numerical solution of non-smooth problems for boundary value problems was investigated. The main idea proposed in [19] is to solve the non-smooth problem as a multi-point boundary value problem on the left and right-side intervals of the discontinuity point. The approximation of the regular solution on each sub-interval will be done by high order finite differences formulas.

Mahmoud and Chen [14] developed a verified inexact implicit Runge-Kutta method for numerical solution of non-smooth ordinary differential equations which computes an inexact solution with high accuracy by taking all possible rounding errors. They used the slanting Newton method to solve the systems of non-smooth equations, and interval method to compute the set of matrices of slopes for the enclosure of solution of the systems.

In [17], the trapezoidal method is proposed for the numerical solution of the non-smooth Tacoma Narrows Bridge equation. In this work, the target problem is discretized by the trapezoidal method and consequently the solution of the problem is reduced to the solution of a system of algebraic equations.

It is noted that, in most of the existing algorithms, the accuracy of the obtained solution is usually poor, consequently, they need a large amount of points to achieve an accurate solution. The fact in the numerical solution of the non-smooth dynamical equations is that, this non-smoothness always causes a decrease in the accuracy of the numerical methods. So, to cope this difficulty, the alternative adaptive quadratures are usually used to deal with this non-smoothness. As we know, the adaptive methods change the number of points in each time steps by estimating the amount of error, which will increase the accuracy and speed of the method. Although an optimal choice to solve the non-smooth dynamical equations is to choose an adaptive method which can overcome the non-smoothness of the problem in the best way. It should be noted that the computer implementation of the adaptive methods is usually not an easy task. So, in this paper, we have come up to the simple and composite Simpson methods which have the same performance both in smooth intervals and in intervals including non-smoothness points.

It is worthwhile to note that, unlike the adaptive methods, our method utilizes the same number of points in each time step. However, considering the idea that we will use in solving the resulting system of algebraic equations, we have been able to increase the accuracy and speed of the proposed method. Consequently, increasing the number of points does not cause inefficiency of the method, and at the same time, the computer implementation of the method is also very simple.

So, the aim of this paper is to present a method for an efficient and robust numerical solution of the non-smooth dynamical equations. In particular, the non-smooth dynamical system is considered as an initial value problem of the form

$$\begin{cases} \dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t)), & t_0 \le t \le t_f, \\ \mathbf{y}(t_0) = \mathbf{y}_0, \end{cases} \tag{1}$$

where, $\mathbf{y} = [y_1, \dots, y_p]^T : [t_0, t_f] \to \mathbb{R}^p$ and $\mathbf{f} = [f_1, \dots, f_p]^T : [t_0, t_f] \times \mathbb{R}^p \to \mathbb{R}^p$. It is worthwhile to note that, in (1), the function $\mathbf{f}(t, \mathbf{y}(t))$ is a non-smooth function with respect to t or $\mathbf{y}$ and also this problem is assumed to have a unique solution, in this paper. Furthermore, in the problems (1), an equivalent reformulation of the problem as a system of Volterra integral equations [2] can be obtained. It is noted that, in this paper, we intend to discretize the obtained system of Volterra integral equations by utilizing the Simpson integral formula. Therefore, the solution of the non-smooth initial value problem (1) is converted to the solution of a system of algebraic equations.

This paper is organized as follows. In Section 2, the proposed method is applied to solve the non-smooth initial value problem (1). As a result, a set of algebraic equations are formed, and a solution of the considered problem is discussed. In this section, an iterative procedure for solving the obtained system of algebraic equations, which leads to an increase in the speed of the method, is introduced. In Section 3, the numerical examples are presented and a comparison is done with the existing works to demonstrate the effectiveness of the proposed method.

2. Solution of the non-smooth initial value problem

Let's go back to the non-smooth initial value problem (1) which is a system of first-order initial value problems. Integrating the dynamical equations in the Eq. (1) from t_0 to t, the equivalent system of Volterra integral equations is induced as

$$\mathbf{y}(t) = \mathbf{y}(t_0) + \int_{t_0}^{t} \mathbf{f}(\tau, \mathbf{y}(\tau)) d\tau = \mathbf{y}_0 + \int_{t_0}^{t} \mathbf{f}(\tau, \mathbf{y}(\tau)) d\tau, \qquad t_0 \leq t \leq t_f. \tag{2}$$

To utilize the Simpson method for the solution of the Volterra integral equations (2), at first, $(n+1)$ equally spaced points

$$t_0 = t_0 + 0h, \quad t_1 = t_0 + 1h, \quad t_2 = t_0 + 2h, \quad \ldots \quad t_f = t_n = t_0 + nh, \tag{3}$$

is considered. It is clear that, for a sample point $t_i,\ i = 0, 1, \ldots, n$, we get

$$\mathbf{y}(t_i) = \mathbf{y}_i = \mathbf{y}_0 + \int_{t_0}^{t_i} \mathbf{f}(\tau, \mathbf{y}(\tau)) d\tau. \tag{4}$$

Obviously, the integral term in the Eq. (4) can be approximated using a numerical quadrature. Consequently, we have

$$\mathbf{y}_i \simeq \mathbf{y}_0 + \sum_{j=0}^{i} w_j \mathbf{f}(t_j, \mathbf{y}(t_j)), \qquad i = 1, 2, \ldots, n, \tag{5}$$

where, the quadrature $\sum_{j=0}^{i} w_j \mathbf{f}(t_j, \mathbf{y}(t_j))$ is used to approximate the integral term in the Eq. (4). It is noted that, the values of $\{\mathbf{y}_i,\ i = 1, 2, \ldots, n\}$ are implicitly defined in the Eq. (5), in which to find them, we have to solve a system of algebraic equations. In the following, we will say how we deal with this system of algebraic equations. But for now, let's focus on the Eq. (5). It is necessary to mention that, there are many quadratures for using in the Eq. (5). For instance, in [17], the trapezoidal quadrature has been proposed. In this paper, the Simpson's numerical quadrature is utilized where it's rule has the form

$$\int_{\alpha}^{\alpha+2h} F(s) ds \simeq \frac{h}{3}[F(\alpha) + 4F(\alpha+h) + F(\alpha+2h)].$$

Now, let us consider the case where, n is even. So, we have

$$\int_{t_0}^{t_n} \mathbf{f}(\tau, \mathbf{y}(\tau)) d\tau \simeq \frac{h}{3} \sum_{j=1}^{n/2} [\mathbf{f}(t_{2j-2}, \mathbf{y}_{2j-2}) + 4\mathbf{f}(t_{2j-1}, \mathbf{y}_{2j-1}) + \mathbf{f}(t_{2j}, \mathbf{y}_{2j})]. \tag{6}$$

Then, we consider the case that, n is odd and $n \geq 3$. Obviously in this case, the interval $[t_0, t_n]$ can not be divided into a union of subintervals $[t_{2j-2}, t_{2j}]$. Consequently, the Simpson's integration rule can not be utilized in the form of (6). In such a case, the Simpson's $\frac{3}{8}$'s formula over one subinterval of length $3h$ is used as follows

$$\int_{\alpha}^{\alpha+3h} F(s) ds \simeq \frac{3h}{8}[F(\alpha) + 3F(\alpha+h) + 3F(\alpha+2h) + F(\alpha+3h)],$$

and, the Simpson's rule is used over the remaining subintervals of length $2h$. More precisely, the interval $[t_0, t_n]$ is divided to the subintervals

$$[t_0, t_n] = [t_0, t_2] \cup \ldots \cup [t_{n-5}, t_{n-3}] \cup [t_{n-3}, t_n].$$

So, the Simpson's $\frac{3}{8}$'s formula is applied over $[t_{n-3}, t_n]$ and the Simpson's formula is applied over the remaining subintervals $[t_0, t_2] \cup \ldots \cup [t_{n-5}, t_{n-3}]$. As a result, we have

$$\begin{aligned} \int_{t_0}^{t_n} \mathbf{f}(\tau, \mathbf{y}(\tau)) d\tau \simeq\ & \frac{h}{3} \sum_{j=1}^{n/2} [\mathbf{f}(t_{2j-2}, \mathbf{y}_{2j-2}) + 4\mathbf{f}(t_{2j-1}, \mathbf{y}_{2j-1}) + \mathbf{f}(t_{2j}, \mathbf{y}_{2j})] \\ & + \frac{3h}{8}[\mathbf{f}(t_{n-3}, \mathbf{y}_{n-3}) + 3\mathbf{f}(t_{n-2}, \mathbf{y}_{n-2}) + 3\mathbf{f}(t_{n-1}, \mathbf{y}_{n-1}) + \mathbf{f}(t_n, \mathbf{y}_n)]. \end{aligned} \tag{7}$$

It is worthwhile to note that, in (7), the initial value $\mathbf{y}_1$ should be calculated by another method. In this paper, the trapezoidal method is utilized for generating $\mathbf{y}_1$ [3]. To this aim, and furthermore to find other $\{\mathbf{y}_i,\ i = 2, \ldots, n\}$ in the Eq. (5), the simple fixed point iteration method [5] for $k = 0, 1, \ldots, N$ is utilized in this paper, as following

$$\mathbf{y}_1^{(k+1)} \simeq \mathbf{y}_0 + \frac{h}{2}[\mathbf{f}(t_0, \mathbf{y}_0) + \mathbf{f}(t_1, \mathbf{y}_1^{(k)})], \tag{8}$$

$$\mathbf{y}_i^{(k+1)} \simeq \mathbf{y}_0 + \begin{cases} \frac{h}{3} \sum_{j=1}^{n/2} [\mathbf{f}(t_{2j-2}, \mathbf{y}_{2j-2}) + 4\mathbf{f}(t_{2j-1}, \mathbf{y}_{2j-1}) + \mathbf{f}(t_{2j}, \mathbf{y}_{2j}^{(k)})], \text{ if } n \text{ is even}, \\ \frac{h}{3} \sum_{j=1}^{n/2} [\mathbf{f}(t_{2j-2}, \mathbf{y}_{2j-2}) + 4\mathbf{f}(t_{2j-1}, \mathbf{y}_{2j-1}) + \mathbf{f}(t_{2j}, \mathbf{y}_{2j})] \\ + \frac{3h}{8}[\mathbf{f}(t_{n-3}, \mathbf{y}_{n-3}) + 3\mathbf{f}(t_{n-2}, \mathbf{y}_{n-2}) + 3\mathbf{f}(t_{n-1}, \mathbf{y}_{n-1}) + \mathbf{f}(t_n, \mathbf{y}_n^{(k)})], \text{ if } n \text{ is odd} \end{cases}. \tag{9}$$

As we can see in the numerical illustrations section, using the fixed point iterations greatly increased the speed of the method. Finally, looking at the Eq. (4), the values of $\{\mathbf{y}_i,\ i = 1, 2, \ldots, n\}$, are approximated respectively.

Remark 2.1. *It should be noted that, the Simpson's described method is extended to the composite Simpson method in this paper. To make this happen, after selecting the equally spaced grid (3), the proposed method can be utilized over each interval* $[t_i, t_{i+1}],\ i = 0, 1, \ldots, n-1$ *for* m*-equally spaced grid at each interval, so that, the solution obtained at the end of each interval is served as an initial value for the next interval and this continues until reaching the last interval.*

Remark 2.2. *The proposed method used in this paper can be also extended to solve non-smooth boundary value problems. In this case, the purpose is to solve the following equation*

$$\mathbf{y}(t_f) = \mathbf{y}_f = \mathbf{s} + \int_{t_0}^{t_f} \mathbf{f}(\tau, \mathbf{y}(\tau)) d\tau, \tag{10}$$

in which, the values of $\mathbf{y}(t_0) = \mathbf{s}$ *are considered to be unknown. Finally, based on using the Simpson formula for discretization, the non-smooth boundary value problem is converted to the following system of algebraic equations*

$$\begin{aligned} &\mathbf{y}_0 - \mathbf{s} = \mathbf{0}, \\ &\mathbf{y}_1 - \mathbf{s} - \frac{h}{2}[\mathbf{f}(t_0, \mathbf{s}) + \mathbf{f}(t_1, \mathbf{y}_1)] = \mathbf{0}, \\ &\mathbf{0} = \mathbf{y}_f - \mathbf{s} - \begin{cases} \frac{h}{3}\sum_{j=1}^{n/2}[\mathbf{f}(t_{2j-2}, \mathbf{y}_{2j-2}) + 4\mathbf{f}(t_{2j-1}, \mathbf{y}_{2j-1}) + \mathbf{f}(t_{2j}, \mathbf{y}_{2j})], & \text{if } n \text{ is even}, \\ \frac{h}{3}\sum_{j=1}^{n/2}[\mathbf{f}(t_{2j-2}, \mathbf{y}_{2j-2}) + 4\mathbf{f}(t_{2j-1}, \mathbf{y}_{2j-1}) + \mathbf{f}(t_{2j}, \mathbf{y}_{2j})] & \\ + \frac{3h}{8}[\mathbf{f}(t_{n-3}, \mathbf{y}_{n-3}) + 3\mathbf{f}(t_{n-2}, \mathbf{y}_{n-2}) + 3\mathbf{f}(t_{n-1}, \mathbf{y}_{n-1}) + \mathbf{f}(t_n, \mathbf{y}_n)], & \text{if } n \text{ is odd} \end{cases} . \end{aligned}$$

It is noted that, the equation (10) has a solution as $\mathbf{y}(t_f, \mathbf{s})$ *and to evaluate it at a given* $\mathbf{s}$*, we need to solve the Eq. (4) to reach the solution* $\mathbf{y}(t_n)$*. If the equation (10) is not established by selecting that* $\mathbf{s}$*, then the new vector* $\mathbf{s}$ *should be evaluated and the above process will be resumed until the equation (10) is satisfied.*

3. Numerical illustrations

This section is devoted to the numerical examples. Two benchmark examples are considered to show the effectiveness of the presented method. In addition, all computations are performed on a 2.53 GHz Core i5 PC Laptop with 4 GB of RAM running in MATLAB R2016a. Furthermore, the simple fixed point iteration method for solving the resulting system of algebraic equations (8) and (9) sets to the $N = 5$ iterations.

Example 3.1. *Consider the following non-smooth initial value problem which is taken from [4]. The problem is*

$$\begin{cases} \dot{y}_1(t) = y_2(t), \\ \dot{y}_2(t) = \frac{1}{r}(\sin(4t) - q(y_1(t))), \\ y_1(0) = 0 \quad y_2(0) = 1, \end{cases}$$

where

$$q(y_1) = \begin{cases} 4y_1 & y_1 \geq 0, \\ y_1 & y_1 < 0 \end{cases} .$$

The analytical solution for $r = 1$ *in the time* $t \in [0, \frac{3\pi}{2}]$ *is*

$$y_1(t) = \begin{cases} \left(\frac{1}{2}(1 + \frac{1}{3}) - \frac{1}{6}\cos(2t)\right)\sin(2t), & 0 \leq t \leq \frac{\pi}{2} \\ \left((1 + \frac{2}{5}) - \frac{4}{15}\sin(t)\cos(2t)\right)\cos(t) & \frac{\pi}{2} \leq t \leq \frac{3\pi}{2} \end{cases} .$$

Now, we want to numerically solve this problem with the help of the composite Simpson method. In Figure 1, the numerical solution for $n = 100$ *and* $m = 250$ *discretization points is shown beside the exact solution, and the absolute error of the numerical solution on the interval* $0 \leq t \leq 3\pi/2$*. Also, the maximum absolute errors obtained from the implementation of the composite Simpson for various values of* n *and* m*, the simple Simpson method, the trapezoidal method used in [17] and 4th order Runge-Kutta method (RK4) which is one of the most common methods in solving dynamical equations,with their CPU times, are reported in Table 1. As we can see, the trapezoidal method is not a suitable method to achieve high accuracy. In other words, the CPU time of the trapezoidal method for a* *large number of points increases greatly, and therefore the method will not have the necessary efficiency to achieve high accuracy. In addition, it can be seen from the Table 1 that, the proposed composite Simpson method has a higher efficiency compared to the simple Simpson method, so that it has managed to achieve high accuracy in a much shorter time. Furthermore in comparison of the RK4 and the proposed composite Simpson method, we can see that, for the same number of points, the accuracy of both methods are almost equal. However, considering the fact that, the RK4 is an explicit method and the composite Simpson method is an implicit method, the speed of the RK4 is higher than the proposed method.*

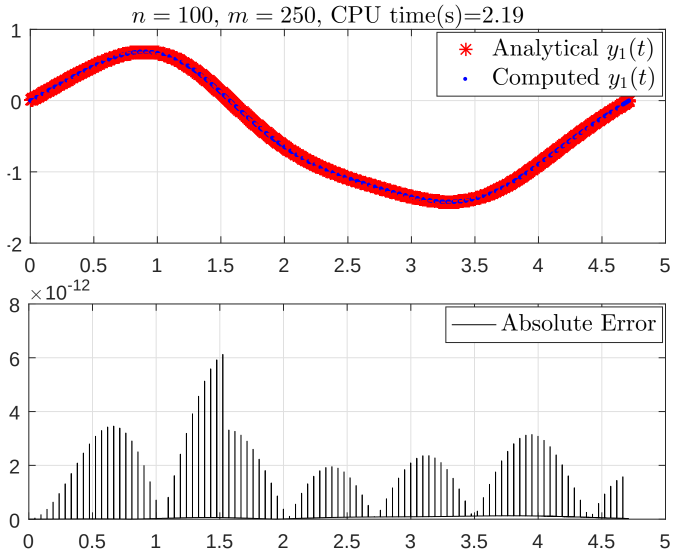

Figure 1: Solution of Example 3.1 by using the composite Simpson method.

Example 3.2. *The next example is the model of water level which is taken from [18, 4]. The problem is*

$$\begin{cases} \dot{y}_1(t) = y_2(t), \\ \dot{y}_2(t) = 0.5u(y_1(t)), \\ y_1(0) = 5, \quad y_2(0) = 1, \end{cases}$$

in which the function u is defined by

$$u(y_1) = \begin{cases} +1 & y_1 < 3, \\ -1 & y_1 > 7, \\ \ \ 0 & otherwise \end{cases}.$$

We consider this problem on the interval $[0, 18]$. It is noted that, the problem has the following analytical solution

$$y_1(t) = \begin{cases} t+5 & 0 \le t \le 2, \\ -\frac{t^2}{4} + 2t + 4 & 2 \le t \le 6, \\ -t+13 & 6 \le t \le 10, \\ \frac{t^2}{4} - 6t + 38 & 10 \le t \le 14, \\ t-11, & 14 \le t \le 18 \end{cases}.$$

As we can see, we are faced with a large interval problem which leads to the trouble in numerical methods.Now, we want to approximately solve this problem with the help of the composite Simpson method. The numerical solution for $n = 101$ and $m = 250$ discretization points is shown in Figure 2 beside the exact solution, and the absolute error of the numerical solution on the interval $0 \le t \le 18$. Furthermore, the maximum absolute errors obtained from the implementation of the composite Simpson for various values of n and m, the simple Simpson, the trapezoidal method used in [17] and RK4 which is one of the most common methods in solving dynamical equations,with their CPU times, are reported in Table 2. It is important to mention one point here.

As we can see in the Table 2, the accuracy of the proposed method has decreased compared to the Example 3.1. By examining this issue more closely, we find that the degree of smoothness of the analytical solution in Example 3.2 is

Table 1: The maximum absolute errors $E(y_1)$ for various values of n and m, in Example 3.1.

n	m	E(y_1)	CPU time(s)
4	70	5.26e-06	0.02
4	190	2.66e-07	0.05
4	1000	1.83e-09	0.58
4	5500	1.10e-11	9.62
9	30	9.01e-06	0.02
9	95	2.74e-07	0.06
9	450	2.66e-09	0.43
9	2200	1.24e-11	4.66
32	10	3.82e-06	0.03
32	40	5.11e-08	0.11
32	170	6.36e-10	0.47
32	700	9.02e-12	3.21
50	100	8.05e-10	0.42
50	470	7.61e-12	2.71
100	50	7.95e-10	0.41
100	250	6.12e-12	2.19
The simple Simpson method with $n = 4000$		2.02e-10	1.64
The simple Simpson method with $n = 13000$		5.88e-12	13.69
The simple Simpson method with $n = 25000$		8.27e-13	48.60
The trapezoidal method used in [17] with $n = 4000$		1.51e-06	1.53
The trapezoidal method used in [17] with $n = 13000$		1.43e-07	12.96
The trapezoidal method used in [17] with $n = 25000$		3.87e-08	47.99
The 4th order Runge-Kutta method (RK4) with $n = 4000$		4.99e-11	0.13
The 4th order Runge-Kutta method (RK4) with $n = 13000$		1.47e-12	0.36
The 4th order Runge-Kutta method (RK4) with $n = 25000$		2.08e-13	0.89

lower than the degree of smoothness of the analytical solution in Example 3.1. In other words, the analytical solution in Example 3.2 is a piecewise smooth function with the degree 1*. However, the analytical solution in Example 3.1 is a piecewise smooth function with the degree* 2*.*

4. Conclusion

In this paper, the simple and composite Simpson numerical formulas were used along with the simple fixed point iteration method in solving root finding problem obtained from discretizing the target non-smooth initial value problems. The main advantages of the present work are that, good results are obtained and the time to implement the method is appropriate. So, the efficient proposed method is reasonably accurate, simple and fast for solving non-smooth initial value problems. Also, by comparing the composite Simpson method with other methods such as simple Simpson and trapezoidal methods, the superiority of the composite Simpson method over the simple Simpson and trapezoidal methods is observed. Moreover, by comparing the composite Simpson method with RK4, which is considered one of the most common methods for solving dynamical equations, it was concluded that the proposed method can be a suitable substitute for the mentioned method in solving dynamical equations. Comparing the stability of these two methods with each other, by considering the implicitness of the proposed method and the explicitness of RK4, will be interesting.

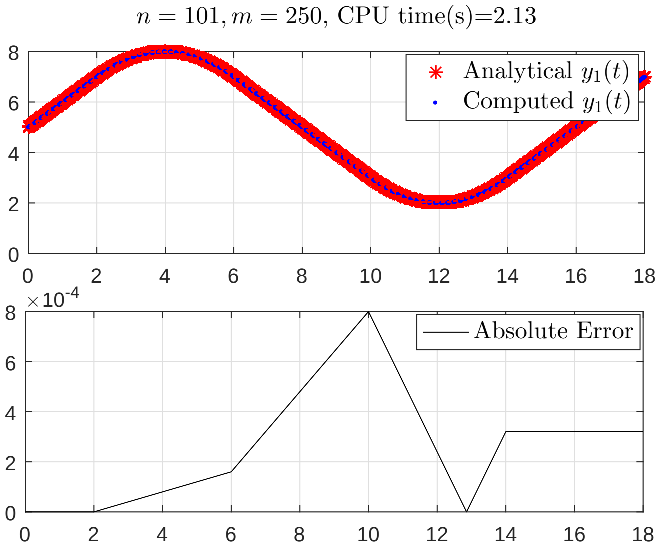

Figure 2: Solution of Example 3.2 by using the composite Simpson method.

Table 2: The maximum absolute errors $E(y_1)$ for various values of n and m, in Example 3.2.

n	m	E(y_1)	CPU time(s)
5	70	7.08e-02	0.03
5	190	2.61e-02	0.07
5	1000	4.99e-03	0.59
5	5500	9.09e-04	10.08
32	10	6.30e-02	0.03
32	40	1.61e-02	0.11
32	170	3.79e-03	0.44
32	700	9.22e-04	2.57
50	100	4.08e-03	0.39
50	470	8.68e-04	2.31
101	50	4.00e-03	0.38
101	250	8.00e-04	2.13
851	250	9.41e-05	17.65
The simple Simpson method with $n = 25250$		7.92e-04	45.91
The trapezoidal method used in [17] with $n = 25250$		6.34e-04	44.50
The 4th order Runge-Kutta method (RK4) with $n = 25250$		3.96e-04	1.01

References

[1] V. Acary and B. Brogliato, *Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics*, Lecture Notes in Applied and Computational Mechanics, Springer-Verlag, Berlin, Heidelberg, 2008.

[2] K. E. Atkinson, *The numerical solution of integral equations of the second kind*, vol. 4 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 1997.

[3] K. E. Atkinson, W. Han, and D. E. Stewart, *Numerical Solution of Ordinary Differential Equations*, John Wiley & Sons, Ltd, 2009.

[4] E. Auer and S. Kiel, *Uses of verified methods for solving non-smooth initial value problems*, Reliable Computing, 19 (2013), pp. 156–175.

[5] R. L. Burden, J. D. Faires, and A. M. Burden, *Numerical Analysis*, Cengage Learning, 10th ed., 2016.

[6] H. R. Erfanian, M. N. Skandari, and A. Kamyad, *A numerical approach for nonsmooth ordinary differential equations*, Journal of Vibration and Control, 19 (2013), pp. 2124–2136.

[7] A. F. Filippov, *Differential equations with discontinuous righthand sides: control systems*, vol. 18, Springer Science & Business Media, 2013.

[8] M. Ghaznavi and M. N. Skandari, *An efficient pseudo-spectral method for nonsmooth dynamical systems*, Iranian Journal of Science and Technology, Transactions A: Science, 42 (2018), pp. 635–646.

[9] R. Leine, *Bifurcations in discontinuous mechanical systems of Filippov's type*, PhD thesis, Technische Universiteit Eindhoven, Eindhoven, Netherlands, 2000.

[10] ——, *Non-smooth stability analysis of the parametrically excited impact oscillator*, International Journal of Non-Linear Mechanics, 47 (2012), pp. 1020–1032.

[11] R. Leine and D. V. Campen, *Bifurcation phenomena in non-smooth dynamical systems*, European Journal of Mechanics-A/Solids, 25 (2006), pp. 595–616.

[12] R. Leine, D. V. Campen, and C. Glocker, *Nonlinear dynamics and modeling of various wooden toys with impact and friction*, Journal of Vibration and Control, 9 (2003), pp. 25–78.

[13] R. I. Leine and H. Nijmeijer, *Dynamics and bifurcations of non-smooth mechanical systems*, vol. 18, Springer Science & Business Media, 2013.

[14] S. Mahmoud and X. Chen, *A verified inexact implicit Runge-Kutta method for nonsmooth ODEs*, Numer. Algorithms, 47 (2008), pp. 275–290.

[15] M. A. Mehrpouya and M. K. e Oshagh, *An efficient numerical solution for time switching optimal control problems*, Computational Methods for Differential Equations, 9 (2021), pp. 225–243.

[16] M. A. Mehrpouya and S. Fallahi, *A modified control parametrization method for the numerical solution of bang-bang optimal control problems*, Journal of Vibration and Control, 21 (2015), pp. 2407–2415.

[17] M. A. Mehrpouya and M. Shahini, *The use of the trapezoidal method for solving the tacoma narrows bridge model*, International Journal of Nonlinear Analysis and Applications, 14 (2023), pp. 67–72.

[18] N. S. Nedialkov and M. V. Mohrenschildt, *Rigorous simulation of hybrid dynamic systems with symbolic and interval methods*, in Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301), vol. 1, IEEE, 2002, pp. 140–147.

[19] I. Sgura, *A finite difference approach for the numerical solution of non-smooth problems for boundary value odes*, Mathematics and Computers in Simulation, 95 (2014), pp. 146–162.

[20] Z. T. Zhusubaliyev and E. Mosekilde, *Bifurcations and chaos in piecewise-smooth dynamical systems*, vol. 44, World Scientific, Singapore, 2003.

Please cite this article using:

Mohammad Ali Mehrpouya, Hossein Heidary, An efficient computational approach for numerical solution of non-smooth dynamical systems, AUT J. Math. Comput., 6(1) (2025) 1-8
https://doi.org/10.22060/AJMC.2023.22596.1180