RESEARCH ARTICLE

Qualitative behavior of stiff ODEs through a stochastic approach

Hande Uslu[a] , Murat Sari[a*], Tahir Cosgun[a,b]

[a] *Department of Mathematics, Yildiz Technical University, Turkey*
[b] *Department of Mathematics, Amasya University, Turkey*
usluh@yildiz.edu.tr, sarim@yildiz.edu.tr, tahir.coskun@amasya.edu.tr

ARTICLE INFO	ABSTRACT
	In the last few decades, stiff differential equations have attracted a great deal of interest from academic society, because much of the real life is covered by stiff behavior. In addition to importance of producing model equations, capturing an exact behavior of the problem by dealing with a solution method is also handling issue. Although there are many explicit and implicit numerical methods for solving them, those methods cannot be properly applied due to their computational time, computational error or effort spent for construction of a structure. Therefore, simulation techniques can be taken into account in capturing the stiff behavior. In this respect, this study aims at analyzing stiff processes through stochastic approaches. Thus, a Monte Carlo based algorithm has been presented for solving some stiff ordinary differential equations and system of stiff linear ordinary differential equations. The produced results have been qualitatively and quantitatively discussed.

1. Introduction

Differential equations are used to model real-life systems by conserving their physical structures. There are different types of differential equations which have been named by according to their characteristics. Stiff differential equations are one of those. While developing a model of a system, it is necessary to consider suddenly occurred reactions with small time steps without neglecting that the system continue to behave over the whole-time interval. Stiff equations represent unstable behaviors for very small values. In other words, a model contains a point which decays or grows very rapidly than others. Despite natural restrictions of physical systems represented by stiff Ordinary Differential Equations (ODEs), they are commonly used in modelling various problems, through chemical reactions, while creating electrical circuits or studying in control theory etc. Not only modelling a stiff behavior but also solving the model accurately play a key role for capturing real-life behavior.

Stiffness was firstly named by Curtiss and Hirschfelder [1] in 1952. Although this explanation leads to be realized that almost all real-life problems include stiff property, the first efficient algorithm for solving the model equations was suggested relatively late, in 1976 by Shampine and Gear [2]. Finding exact solution for stiff problems is generally limited to simple cases and conventional numerical methods have to be reconstructed with small time steps for these types of problems. However, the increased number of steps might possibly cause an accumulation of error. This fact gives rise to a necessity of alternative approaches for stiff equations. In the last few decades, various implicit and explicit methods related to stiffness have been developed.

The explicit methods find a solution by using the current time information to produce later time information. However, implicit ones use the current and later time information at the same time. While analyzing stiff behavior, it should be taken into consideration how much small changes in the current time information affects the later time. Explicit methods generally do not work efficiently for catching the changing behavior in small step sizes or if they do, it converges very slowly than expected [3]. If the initial conditions cause a divergence in the solution, an explicit method requires impractically small step sizes to control the convergence. Although the implicit ones need more computation and requires sensitive implementations, they are properly applied to many stiff problems.

Even though an application area of numerical methods has a broad range, they are occasionally suffering from their restrictions. They may be seen to be efficient for the aim of the solving the problems iteratively, but these methods cannot be a first choice considering their computational time, computational error or effort spent for construction of a stiff structure. At this point, new approaches such that simulation techniques emerge by paying attention to these corresponding issues [4-5]. The Monte Carlo Method (MCM) is one of the basic simulation techniques [6-8]. It has been generally defined as a random sampling method for solving any model. Since this method uses basically random variables to represent the behavior of physical processes, it is classified as a stochastic approach.

The MCMs can be applied to a wide range of problems in three different ways; sampling, estimation and optimization [9-10]. This classification depends on aim and a way of building algorithm. If a researcher wants to use simulation to mimic the nature of the system by creating objects or unreal systems, sampling methods are more useful than the rest. Therefore, random sampling and estimation techniques are used in this study to observe the behavior of the stiff differential equations.

2. Implementation of the method

The main intention of this study is to capture the exact behavior of stiff differential equations by using simulation techniques. To achieve this, differential equations are described by using integrals since the Monte Carlo integration is based on random sampling [11-13]. This randomness comes from uniformly distributed pseudorandom numbers selected by a sample space. The method is named by rejection sampling which is used for generating random variables X with density function ρ. The main advantage of using rejection sampling is that sampling can be used even if the density function cannot be integrated analytically.

Let us then consider any first order differential equation in an implicit form:

$$\frac{dy}{dx} = F(x,y) \tag{1}$$

where function F represent an arbitrary function with variables. After modifying the equation in this form, the algorithm needs a reference number which is chosen to do comparison in the related steps of the algorithm. The first reference number is generated by using initial conditions X_0 and Y_0 and this number should be revised for each iteration. The step size is determined by dividing uniformly the interval to m points. Let us call this reference number as Classification Number (*CN*) defined as follows

$$CN := \frac{dY}{dX} = F(X_n, Y_n) \tag{2}$$

where $n = 0,1,\ldots,m$.

Next step, determination of upper and lower bounds for generating random numbers is expected to lead to more accurate estimation. The estimation can be made under the consideration of the physical realities of the problem. These bounds are determined by initial conditions. After determining an upper and a lower bound, random numbers can be created according to these bounds for making a comparison with the *CN*. To create random numbers, *rand* function of MATLAB can be used. This function generates different pseudorandom numbers between 0 and 1. They are known as pseudorandom since even if they act as a random number they are generated according to some artificial algorithm by the function. These random numbers between 0 and 1 are extended to the interval which determined by upper and lower bounds.

Then the comparison starts with created N positive random variables and N negative random variables by using *rand* function with respect to the *CN* of the algorithm. The way of implementation is given in the following pseudocode.

Pseudocode: Monte Carlo Based Algorithm for ODEs
---
1. Consider the differential equation $\frac{dY}{dX}$ as a function $\boldsymbol{F(X,Y)}$ and initial conditions as $\boldsymbol{X_0}$ and $\boldsymbol{Y_0}$.
2. Define $:= \frac{dY}{dX} = F(X_n, Y_n)$.
3. Find the upper and lower boundaries, **U** and **L**, for the classifying random numbers.
4. Create **N** random numbers.
5. Initialize $\boldsymbol{X = X_0}$ and $\boldsymbol{Y = Y_0}$
6. while $(\boldsymbol{X < X_f})$
7. $CN = F(X,Y)$
8. if $\boldsymbol{CN} \geq 0$ then
        S ← the value of the random numbers $\leq \boldsymbol{CN}$
        $\boldsymbol{Y_{k+1} \leftarrow Y_k + U\frac{S}{N}\Delta X}$
9. Else
        S ← the value of the random numbers $\geq \boldsymbol{CN}$
        $\boldsymbol{Y_{k+1} \leftarrow Y_k - L\frac{S}{N}\Delta X}$
10. end if
11. end while
---

3. Illustrative examples

In this section, the predicted results of various differential equations by applying the current algorithm have been illustrated. In the examples, 100000 random samples are chosen for each iteration in the algorithm and the increment of time is taken to be 0.001. To justify the predicted results, comparison is made with both the *ode23s* based on a modified Rosenbrock formula of order 2 and the fourth order Runge-Kutta Method (RK4) as well as available analytical solutions. The two results are compared with each other by using the absolute error. Qualitative and quantitative behaviors have been exhibited by comparing with computational costs in detail. To compute the results, the codes have been produced in MATLAB 2018a

installed on a computer which has the properties of 2.3 GHz intel core i5 and 16 GB ram.

3.1. Example 1

Let us consider a first order stiff ODE with an initial condition,

$$\frac{dy}{dt} = -1000y + 3000 - 2000e^{-t}, y(0) = 0. \quad (3)$$

The exact solution of the differential equation is:

$$y(t) = 3 - 0.998e^{-1000t} - 2.002e^{-t}. \quad (4)$$

The Monte Carlo based algorithm is applied to Equation (3) by dividing the time axis uniformly. Qualitative results including the solution produced by the proposed algorithm, the exact solution and the absolute errors have been illustrated in Figure 1. The corresponding numerical results can be seen in Table 1.

It can be easily observed in Figure 2 that the stiffness occurs between the points 0 and 0.006, near to initial value. The initial deviation dampens fast due to the large value of coefficients. Despite the fact that each trial uses different set of random numbers to predict the results, each trial indicates the common feature at this stiff point. So the quantitative results are also close to each other. Even though quantitative results have slightly little deflections, qualitative results can be seen in good agreement with the exact results.
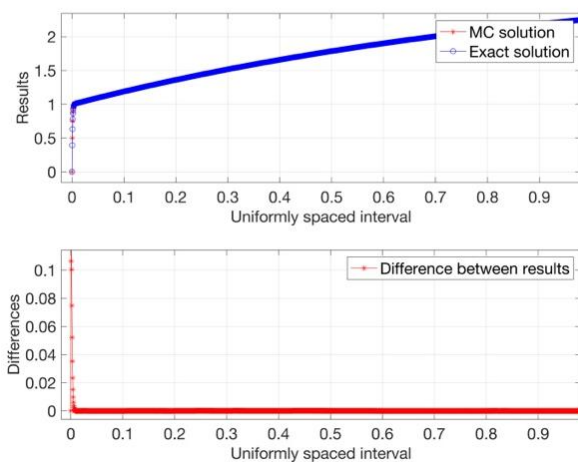

Figure 1. Comparison of the MC prediction and the exact solution of equation (3)

Computational time of the proposed algorithm is 0.5548 s for this set of trial. Moreover, the accuracy is expected to be improved by reasonably decreasing the step size of the interval. However, the small step size leads a large number of comparisons in the algorithm, so that increasing the computational cost. Even if there are higher computational costs for some complex problems, it is seen that the proposed algorithm is the accurate solver as one of the simulation techniques.

Table 1. Numerical results of Equation (3)

| Time t	Predicted Results	Exact Results	Absolute Errors
0.0005	0.49991771	0.39368315	0.10623456
0.0010	0.75242301	0.63285732	0.11956569
0.0015	0.87872659	0.77831685	0.10040974
0.0020	0.94166335	0.86693539	0.07472796
0.0025	0.97329583	0.92107792	0.05221791
0.0030	0.98957043	0.95430951	0.03526092
0.0035	0.99823965	0.97485776	0.02338189
0.0040	1.00283457	0.98771300	0.01512157
0.0045	1.00556209	0.99590198	0.00966011
0.0050	1.00740685	1.00126055	0.00614630
0.0100	1.01789819	1.01787492	0.00002327
0.0150	1.02777839	1.02780559	0.00002720
0.0200	1.03761332	1.03764225	0.00002893
0.0250	1.04739166	1.04742956	0.00003789
0.0500	1.09561564	1.09563869	0.00002305
0.2500	1.44096991	1.44084083	0.00012908
1.0000	2.26347104	2.26350536	0.00003432
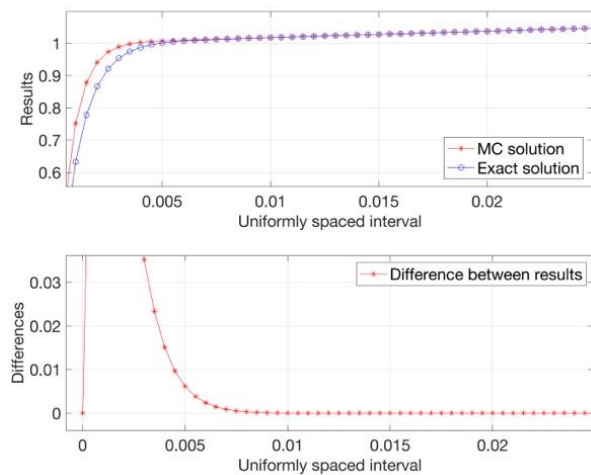
Figure 2. A closer view of Figure 1

3.2. Example 2

Let us now take a first order stiff ODE with an initial condition,

$$\frac{dy}{dt} = -1000y + \sin t, y(0) = 1/1000001. \quad (5)$$

Exact solution of the differential equation is then

$$y(t) = \frac{1000\sin t + \cos t}{1000001}. \quad (6)$$

The proposed algorithm is applied to Equation (5) by dividing the time axis uniformly. The comparison of predicted results with the analytical solution of Equation (5) is given Figure 3 and the corresponding absolute errors are shown in Figure 4. Quantitative results of Equation (5) are exhibited in Table 2.

Unlike Example 1, from Figure 3 it can be seen that the solutions already computed are much closer to the exact solution for all points in the range. It can be seen from the absolute errors that the results are very accurately predicted by the current algorithm. Computational cost of the present algorithm is 0.2913 s for this set of trial. Even if the computational time of the algorithm seems to be higher than its rivals depending on random numbers, its accuracy level is relatively in good agreement with available analytical solution.
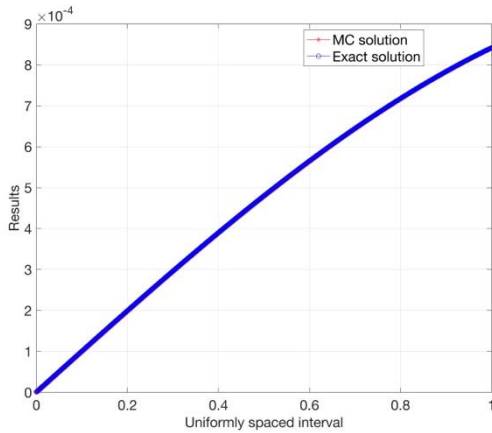
Figure 3. Comparison of the MC Prediction and the exact solution of equation (3)

Table 2. Numerical results of the first order stiff differential equation (5)

Time t	Predicted Results	Exact Results	Absolute Errors
0.0100	0.00000922	0.00001100	0.00000178
0.0500	0.00004920	0.00005098	0.00000178
0.1000	0.00009904	0.00010083	0.00000178
0.2000	0.00019788	0.00019965	0.00000177
0.3000	0.00029473	0.00029648	0.00000174
0.4000	0.00038866	0.00039034	0.00000168
0.5000	0.00047870	0.00048030	0.00000161
0.6000	0.00056397	0.00056547	0.00000150
0.7000	0.00064363	0.00064498	0.00000135
0.8000	0.00071679	0.00071805	0.00000126
0.9000	0.00078283	0.00078395	0.00000111
1.0000	0.00084103	0.00084201	0.00000098

3.3. Example 3

Now take a first order stiff differential equation system with initial conditions

$$\frac{dx}{dt} = -80.6x + 119.4y, \qquad x(0) = 1$$
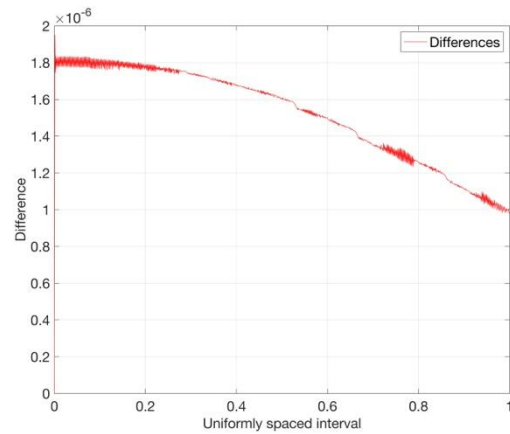$$\frac{dy}{dt} = 79.6x - 120.4y, \qquad y(0) = 2 \tag{7}$$

Figure 4. Absolute Errors of Equation (5)

The Monte Carlo based algorithm is applied to Equation (7) by dividing the time axis uniformly. The comparison of the results of the current algorithm with the ode23s results and a closer view can be seen in Figures 5 and 6, respectively. The corresponding differences are illustrated in Figure 7. Quantitative results of Equation (7) are exhibited in Table 3.

As seen in the corresponding figures, the deviations originating near to the initial conditions have arisen rapidly. Two different equations behave separately; one is increasing while the other one is decreasing. Though the solution remains close to the referenced solution curves in a large scale of vertical axis, the deviations may occur. However, the qualitative and quantitative results can be seen in good agreement with the *ode23s* results.

Even though *ode23s* is commonly accepted as one of the most suitable methods for properly capturing stiff behavior, the current method is seen to be as suitable as the *ode23s*. To support this, another suitable method, RK4 can be applied to this example. The difference between results of the simulation technique and the RK4 results are seen to be relatively small. Therefore, it has been claimed that the approach has ability to capture the stiff behavior. The predicted results have reasonable agreement with the results of *ode23s* and RK4 according to the Figure 8 and Table 3 and the differences between the predicted and *ode23s* solutions.

The computational costs of the current algorithm, *ode23s* and RK4 are 0.4214, 0.0180 and 0.1835 s, respectively. Despite the relatively higher computational cost of the proposed algorithm, the rest of its advantages is taken us to see attractiveness of the approach. In this respect, the computational cost can be sacrificed in simulation techniques in case of especially discrete and continuous methods have serious lack of accuracy or not existing solution for intricate problems.
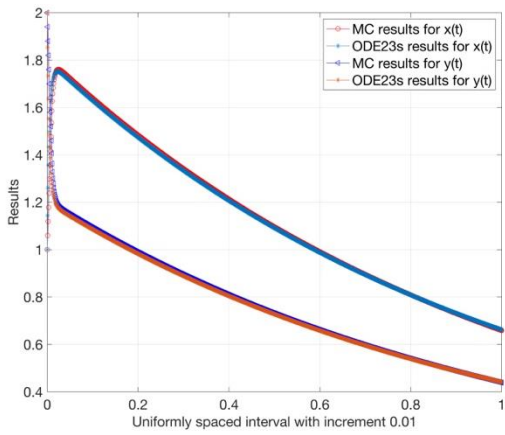
Figure 5. Comparison of the MC prediction results and the ode23s results of the first order stiff differential equation system (7)
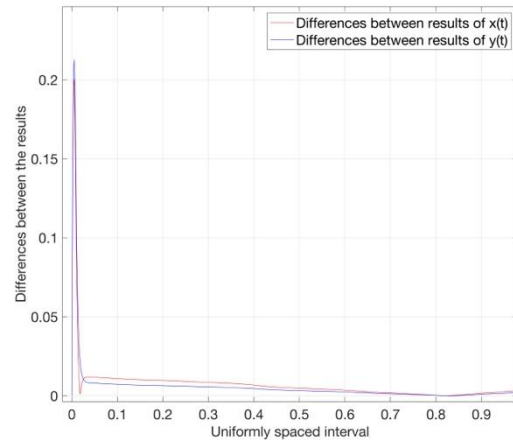

Figure 7. Differences between the MC prediction results and the ode23s results of Equation (7)
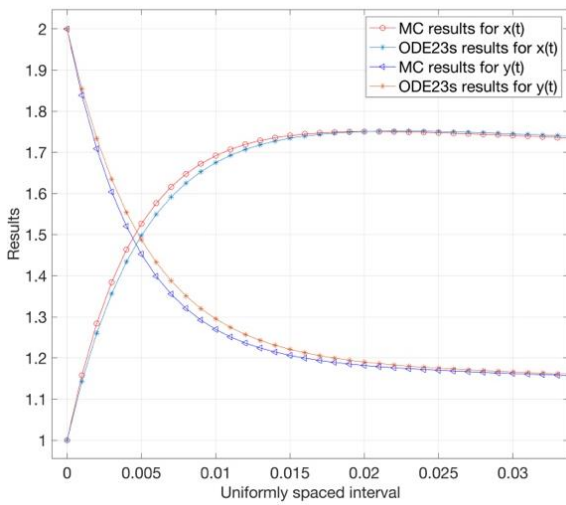

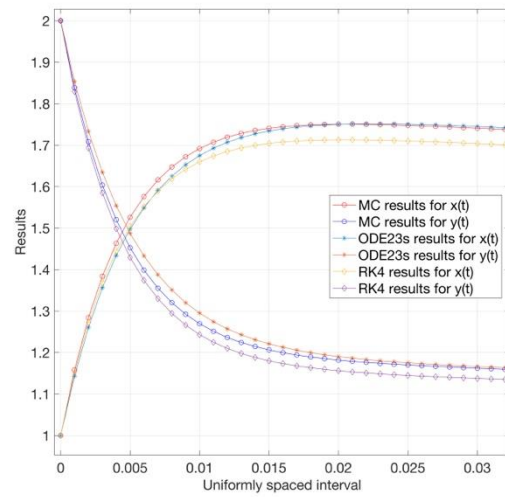**Figure 6.** A closer view of Figure 5

Figure 8. Comparison of the MC results with the ode23s and RK4 results for the system of differential equations (7)

Table 3. Comparison of the predicted results with the *ode23s* and RK4 results

Time t	Predicted results for $x(t)$	*ode23s* solutions for $x(t)$	RK4 solutions for $x(t)$	Predicted results for $y(t)$	*ode23s* solutions for $y(t)$	RK4 solutions for $y(t)$
0.0010	1.15799020	1.14321714	1.15199238	1.83863000	1.85378436	1.82952707
0.0050	1.52630360	1.49780400	1.50433322	1.45274000	1.48723344	1.42945587
0.0100	1.69183400	1.67488571	1.66003934	1.26965000	1.29526379	1.24329043
0.0500	1.70742880	1.71222798	1.67279687	1.13853200	1.14145986	1.11337548
0.1000	1.62454282	1.62871297	1.59577436	1.08315800	1.08581070	1.06410479
0.2000	1.46892028	1.47366156	1.45219073	0.97910000	0.98244072	0.96835941
0.3000	1.32774952	1.33336946	1.32152638	0.88512200	0.88891302	0.88122894
0.4000	1.20073072	1.20643333	1.20261887	0.80063000	0.80428888	0.80193825
0.5000	1.08589966	1.09158143	1.09441035	0.72404000	0.72772096	0.72978193
0.6000	0.98283910	0.98766339	0.99593815	0.65544800	0.65844226	0.66411805
0.7000	0.88987552	0.89363829	0.90632622	0.59349800	0.59575886	0.60436243
0.8000	0.80615050	0.80856434	0.82477734	0.53775800	0.53904289	0.54998347
0.9000	0.73031200	0.73158939	0.75056602	0.48726200	0.48772626	0.50049739
1.0000	0.66218674	0.66193829	0.68303204	0.44193800	0.44129220	0.45546393

4. Conclusions and recommendations

In this study, a Monte Carlo based stochastic algorithm has been developed to discover the behavior of real-world processes governed by stiff differential equations. All qualitative and quantitative results produced by the present algorithm have been seen to be in good agreement with the real environment. Despite the effect of randomness to error, the current procedure has been seen to produce highly acceptable results. Even if reconstructing the conventional methods with small time steps for stiff problems is affordable, simulation techniques can be better choices for challenging problems. In real-life problems, when there are sudden deviations in the consequences of random movements, it is necessary to consider the current stochastic approach that can handle the rapidly changing behavior.

References

[1] Curtis, C.F. & Hirschfelder, J.O. (1950). *Integration of Stiff Equations.* Proceedings of the National Academy of Sciences of the United States of America*,* 38(3), 235-243.

[2] Shampine, L.F. & Gear, C.W. (1976). *A User's View of Solving Ordinary Differential Equations.* Department of Computer Science University of Illinois at Urbana.

[3] Hairer, E. & Wanner, G. (2000). *Solving Ordinary Differential Equations II.* Springer*,* Second Edition.

[4] Birge, J.R. & Louveaux, F. (2011). *Introduction to Stochastic Programming*. Springer Science and Business Media.

[5] Kroese, D.P. & Chan, J.C. (2016). *Statistical Modeling and Computation*. Springer.

[6] Rubinstein, R.Y. & Kroese, D.P. (2016). *Simulation and the Monte Carlo Method* (Vol. 10). John Wiley & Sons.

[7] Fishman, G. (2013). *Monte Carlo: Concepts, Algorithms, and Applications*. Springer Science and Business Media.

[8] Kroese, D.P., Taimre, T., & Botev, Z.I. (2013). *Handbook of Monte Carlo Methods* (Vol. 706). John Wiley and Sons.

[9] Chapra, C.S. (2017). *Applied Numerical Methods with MATLAB for Engineers and Scientists*. McGraw Hill.

[10] Dekking, F.M. (2005). *A Modern Introduction to Probability and Statistics: Understanding why and how*. Springer Science and Business Media.

[11] Uslu, H. (2018). *Behavior of First Order Differential Equations Through a Monte Carlo Based Algorithm*. MSc Thesis, Yildiz Technical University, Istanbul, Turkey.

[12] Sari, M., Uslu, H. & Cosgun, T. (2018). The Qualitative Behavior of Some Stiff ODEs Through Stochastic Methods. *The International Conference on Applied Mathematics in Engineering (ICAME'18),* Balikesir, Turkey.

[13] Uslu, H. & Sari, M. (2019). Monte Carlo based stochastic approach for first order nonlinear ODE systems. *Pamukkale Journal of Engineering Sciences*, 26(1), 133-139.

Hande Uslu is a Research Assistant in the Department of Mathematics at Yildiz Technical University, Istanbul Turkey. She received her BSc degree in Mathematics from Bogazici University in 2015, and MSc degree in Applied Mathematics from Yildiz Technical University, in 2018. She is currently continuing her PhD studies in Department of Mathematics at Yildiz Technical University. Her research interests are numerical solutions of differential equations, mathematical modelling, simulation techniques and stochastic processes.
https://orcid.org/0000-0002-1642-1120

Murat Sari is a Professor of Mathematics at Yildiz Technical University. He received the BSc degree from the Ondokuzmayis University in 1991, and the MPhil (transferred) and PhD degrees in Mathematics from University of South Wales, UK, in 1997 and 2000, respectively. His current research interests include numerical solutions of differential equations, simulation and computational methods, computational fluid dynamics, modelling of nonlinear behavior, economical modelling, and biomechanical/biomedical modelling. He has over 50 high-quality scientific papers, over 40 conference proceedings, and written/translated various chapters in some books. He is a reviewer for many international high-quality journals.
https://orcid.org/0000-0003-0508-2917

Tahir Cosgun has been working as a Research Assistant at Amasya University since 2012. He received BSc degree in Mathematics at Bogazici University (2012), and MSc degree in Mathematics at Amasya University (2015). Currently, he is doing his Ph.D. in Applied Mathematics at Yildiz Technical University. His main research interests are dynamical systems, stiff differential equations, numerical and analytical solutions of partial differential equations and integral equations.
https://orcid.org/0000-0003-2970-0863

An International Journal of Optimization and Control: Theories & Applications (http://ijocta.balikesir.edu.tr)