

Monte Carlo based stochastic approach for the first order nonlinear ODE systems

Birinci mertebeden lineer olmayan adi diferansiyel denklem sistemleri için Monte Carlo temelli stokastik yaklaşım

Hande USLU^{1*}, Murat SARI²

^{1,2}Applied Mathematics, Faculty of Art and Sciences, Yıldız Technical University, İstanbul, Turkey.
usluh@yildiz.edu.tr, sarim@yildiz.edu.tr

Received/Geliş Tarihi: 19.10.2018, Accepted/Kabul Tarihi: 04.05.2019

* Corresponding author/Yazışılan Yazar

doi: 10.5505/pajes.2019.25493

Research Article/Araştırma Makalesi

Abstract

After the discovery of the effectiveness of the stochastic methods for solving real life problems, these methods have been applied to a wide range of problems in two types; deterministic problems and stochastic problems. The general opinion takes part in applying these methods to stochastic problems since it is preferable for realistic results. Moreover, those methods can also be used in dealing with deterministic models. This study aims to show how stochastic approaches can be applied to deterministic models. Thus, an algorithm based on the Monte Carlo simulation has been presented for solving some systems of nonlinear differential equations. To discuss the behavior of such models, the population equations have been taken into consideration. The considered approach has been seen to produce more accurate results than numerical techniques. A detailed discussion about the results has also been given in this work.

Keywords: Monte Carlo method, Stochastic method, Population models

Öz

Gerçek hayat problemlerini çözmek için stokastik yöntemlerin etkinliğinin keşfinden sonra bu yöntemler, deterministik ve stokastik olmak üzere iki tipteki geniş çaplı problemlere uygulanır oldular. Gerçekçi sonuçlar için tercih edilebilirliğinden dolayı bu yöntemleri stokastik problemlere uygulamak genel kanı olmuştur. Fakat bu yöntemler deterministik modellerle çalışmak için de kullanılabilir. Bu çalışma stokastik yöntemlerin deterministik modellere nasıl uygulanabileceğini göstermeyi amaçlamaktadır. Bu yüzden Monte Carlo simülasyonu temelli bir algoritma, lineer olmayan diferansiyel denklem sistemlerini çözmek için sunulmuştur. Bahsi geçen modellerin davranışlarını tartışmak için popülasyon denklemleri ele alınmıştır. Bu yaklaşımın sayısal tekniklerden daha doğru sonuçlar ürettiği görülmüştür. Bu çalışmada sonuçlar hakkında detaylı bir tartışma yapılmıştır.

Anahtar kelimeler: Monte Carlo yöntemi, Stokastik yöntem, Popülasyon modelleri

1 Introduction

Any physical event can be represented by a differential equation since it represents a rate of change by derivatives. The solutions of differential equations are important as well as differential equations themselves since they are needed to analyze the behavior of physical phenomena. Because of this importance, differential equations and their solutions have been considered in many scientific areas for many years.

There are various solution techniques for the differential equations. They can be mainly grouped under three headings; analytical, numerical and simulation techniques. Numerical and simulation techniques can be considered if the problems lack of having an exact solution or suffer from the way of calculations.

The Monte Carlo method is a stochastic simulation technique based on random sampling to model random behaviors that are too difficult to solve analytically [1],[2]. Although the introduction of the method came out in the late 1940s, a significant progress in the method has taken place in the last decades and the method has begun to be used in various fields of science; including statistics, engineering, computer science and so on. Developments in computer programming and fast computer tools led to enhance implementation of the Monte Carlo methods [3].

After the discovery of the effectiveness of the stochastic methods for solving real-life problems, these methods have been applied to a wide range of problems in two types;

deterministic and stochastic. The applicability of these methods to stochastic problems tends to be the first option coming to mind since it is preferable for getting all possible results. Also those methods can be properly used in dealing with deterministic models [4]-[6].

When a nonlinearity occurs in somewhere, unexpected behaviors can come out consecutively. Nonlinear cases need more effort than linear ones since the equations are fairly harder to solve than linear equations. If it is a system of nonlinear equations of interest, the effort spent needs to be increased.

The prey predator model is being used to represent a simple nonlinear system. There are various method to obtain accurate solutions by using numerical algorithms such that Adomian Decomposition Method (ADM), Fourth Order Runge Kutta Method (RK4) and so on [7]. On the other hand, the simulation techniques can be applied this type of problems as well as numerical ones. In this respect, some studies have been presented for finding the solutions of first order initial value problems of ordinary differential equations on the basis of Monte Carlo method [8]-[10]. Moreover, methods for solving nonlinear equations and systems of nonlinear equations have been created and improved day by day. But challenging parts are getting high stability, high accuracy and low computational costs [11],[12].

In this study, a solution of systems of first order homogeneous and non-homogeneous nonlinear differential equations are

introduced with the Monte Carlo based stochastic algorithm and solutions are given both qualitatively and quantitatively. To understand more precisely behavior of the solutions, systems have been considered as population systems with two different populations; a prey and a predator, as named the Lotka-Volterra equation.

2 Implementation of the method

One of the most practical way for solving a differential equation is the usage of integration. Even if applying the integration to the system of equations may not be as easy as expected, the present algorithm is proposed to solve first order nonlinear ordinary differential equations (ODE) through the Monte Carlo approach [13],[14]. To be able to apply the algorithm, the system of differential equations is modified. The main steps of the implementation of the Monte Carlo method for producing simulated results are shown in Figure 1.

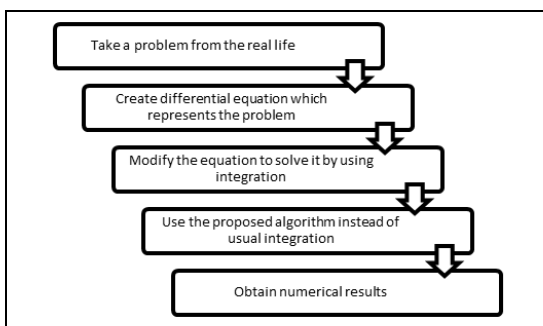


Figure 1: Implementation of the Monte Carlo method.

Let us now consider the first order differential equation in implicit form

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

where function F stands for an arbitrary function involving dependent and independent variables of the differential equation. By choosing the function F in a suitable way of the considered problem, a reference number is accepted to do comparison in the related steps of the algorithm. The reference number should be generated for each iteration. The first one can be created by using initial conditions X_0 and Y_0 and this creation keeps going on for each step. The step size is determined by dividing uniformly the interval to m points. Let us name this reference number as Classification Number (CN) defined as follows

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

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

Next step, determination of upper and lower bounds for generating random numbers is expected to lead to more accurate estimation. If an exact solution is known, then the bounds can be chosen according to the maximum and minimum values of the solution in the related interval. If not, the estimation can be made under the consideration of the physical realities of the problem.

Although it is possible to optimize the bounds as given in the literature [8], in reality it has not been standardized yet. The effect of determination a range for random variables is discussed in detail under the Example 1. Actually, this study

primarily focuses on the application of realistic problems rather than the detailed discussion of the optimum parameters.

After determining an upper and a lower bound, random numbers can be created according to these bounds for making a comparison with the CN. Create N positive random variables and N negative random variables for making a comparison with respect to the CN of the algorithm.

The algorithm works according to below strategy:

1. Define $\frac{dY}{dX} := F(X_n, Y_n)$,
2. Find the upper and lower boundaries, U and L , for making classification of random numbers,
3. Determine an interval and step size, then divide the interval m points with respect to step size,
4. Create N random numbers,
5. $CN = F(X_n, Y_n)$ for $n = 0, \dots, m$,
6. Initialize $X = X_0$ and $Y = Y_0$,
7. If $CN \geq 0$
Compare N positive random variables with CN Count the number of random variables smaller than or equal to CN and say S ,
8. Else
Compare N negative random variables with CN
Count number of random variable bigger than or equal to CN and say S ,
Calculate $Y_{k+1} = Y_k + U \frac{S}{N} \Delta X$

3 Systems of first order nonlinear ODEs

To analyze the behavior of the solution of the differential equations system, the well-known population model equation, the Lotka-Volterra equation, has been considered.

Let $x(t)$ represents the population of the prey species and $y(t)$ represents the population of the predator species, then the first order nonlinear ODE system is

$$\frac{dx}{dt} = \alpha x - \beta xy + h_x, \quad x(0) = x_0 \quad (3)$$

$$\frac{dy}{dt} = -\gamma y + \delta xy + h_y, \quad y(0) = y_0$$

with constant coefficients α, β, γ and δ , the initial conditions x_0 and y_0 and source terms $h_x = h_x(x, t)$ and $h_y = h_y(y, t)$ which represent the rate of external factors per unit time for prey population model and predator population model, respectively [15],[16].

A homogeneous system in real-life means that the physical environment of interest has been isolated from all possible external factors. This homogeneous system can be considered that two populations live in a closed area without affected by external factors in terms of behaviors of the population models. On the other hand, in the real-life problems, some external factors may occur and affect the number of the populations apart from the birth or death ratios. Since the existence of h_x and h_y in the problem is more realistic than the non-existence case, we also take into consideration the nonzero cases of the terms h_x and h_y . For the sake of being natural, all possible cases

of the terms h_x and h_y have been considered illustratively in examples.

Constant coefficients in Equation 3, α, β, γ and δ designate rate constants a birth rate of a prey population, a death rate of a prey population, a death rate of a predator population and a birth rate of a predator population respectively. For each prey-predator populations, these constants are different, i.e. they are unique for a system. The initial conditions x_0 and y_0 refer to the number of individuals of populations before the observation starts.

4 Illustrative simulations

In this section, nonlinear first order differential equation systems are solved by using the Monte Carlo based algorithm. For each example 100000 random samples are used. For comparison purpose, first order nonlinear differential equation systems are solved by the *ode45* function of MATLAB which is based on the Runge-Kutta method to solve ordinary differential equations numerically [15]. Evenly spaced interval with increments 0.01 has been used for systems. After the application of the algorithm to the problems, predicted results have been compared with the numerical *ode45* results, the qualitative and quantitative behaviors have been shown in detail. The two results are compared by using the formula $|p - y|$ where p is the predicted solution and y is the *ode45* solution of the given problem. 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.

In addition to the following simulated results, effects of the parameters are also discussed in this section.

4.1 Example 1

Let us consider a system of first order nonlinear differential equations in section 3 with constants $\alpha = 0.6, \beta = 0.3, \gamma = 0.4, \delta = 0.65$, the rates of external factors are $h_x = h(x, t) = 0$ and $h_y = h(y, t) = 0$ with initial conditions $x_0 = 2$ and $y_0 = 1$.

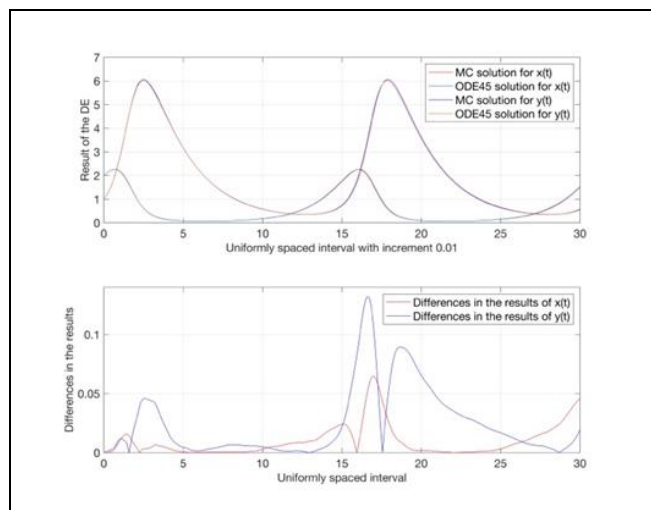


Figure 2: Comparison of the MC predictions and the numerical solutions of equations in Example 1 and differences between the results on the interval $[0, 30]$.

The Monte Carlo based algorithm is applied to the system by dividing the time interval uniformly with the increment 0.01.

Figures 2 and 3 and show the qualitative behavior of the algorithmic solutions compared with the numerical solutions obtained by the *ode45*. Quantitative results of the system have been given in a comparative way in Table 1.

When the qualitative behavior of the solution is considered, it can be easily seen that there is a rapid growth for the predator population after the observation starts. After a while, again a rapid decreasing occurs since the number of prey population is decreasing step by step. Then the prey population starts to grow because of the decreasing number of predators. However, this growth provides the food for the predator population causes again being hunted. When all these observations are considered, the predicted and numerical results are seen to be in relatively good agreement.

Since behavior of the populations is periodical when the external factors are isolated, different size of the time intervals has not been expected to affect the general behavior of the population in the real life situations. Therefore, the proposed algorithm is expected to exhibit the same behavior for various size of the intervals. Moreover, the algorithm and its restrictions should also be properly applied to different types of problems for producing physically suitable results.

To discuss effects of the parameters on the behavior of the populations for various intervals, the equation given in Example 1 has also been solved for interval $[0, 50]$ and then the agreement between the solutions are seen in Figures 4 and 5. The only difference between the two results for each one of Figures 4 and 5 is the amount of random numbers. The amount of random numbers are taken to be 100000 and 170000 for the corresponding figures, respectively. It can be concluded that the difference between the predicted and numerical results can be reduced when the amount of the random numbers increases reasonably.

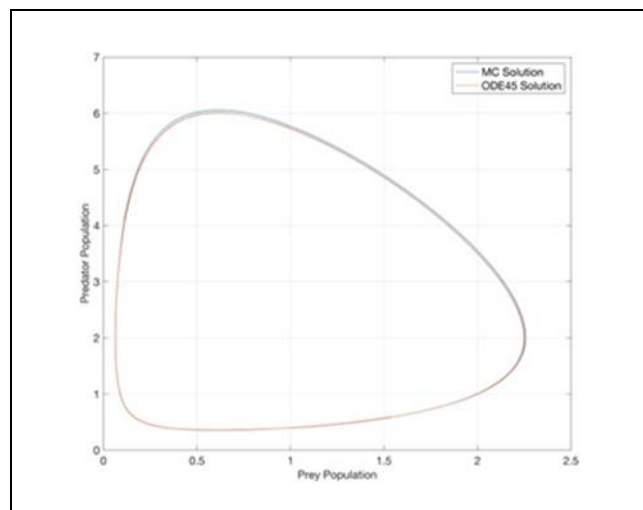


Figure 3: Behaviors of the population functions relative to each other for the equation system.

Table 1: Numerical results for the system of first order nonlinear differential equations in example 1.

Time t	The proposed algorithm solutions for $x(t)$	ode45 solutions for $x(t)$	Differences between the results for $x(t)$	The proposed algorithm solutions for $y(t)$	ode45 solutions for $y(t)$	Differences between the results for $y(t)$
0.0100	2.0059880	2.0059818	0.0000062	1.0090035	1.0090603	0.0000567
0.0500	2.0295352	2.0295256	0.0000096	1.0462560	1.0465326	0.0002766
0.1000	2.0580144	2.0580012	0.0000132	1.0957273	1.0962634	0.0005360
0.5000	2.2288696	2.2277177	0.0011519	1.6310536	1.6336544	0.0026008
1.0000	2.1846800	2.1746169	0.0100631	2.7572478	2.7686003	0.0113524
5.0000	0.0793320	0.0798507	0.0005187	3.1717523	3.1671811	0.0045712
10.0000	0.1687980	0.1717176	0.0029196	0.5789502	0.5739734	0.0049767
15.0000	1.7525588	1.7766046	0.0240458	0.7293483	0.7510186	0.0216702
20.0000	0.0922835	0.0903390	0.0019445	3.6255798	3.5601515	0.0654282
25.0000	0.1442134	0.1473222	0.0031088	0.6492148	0.6324381	0.0167766
30.0000	1.4971846	1.5431068	0.0459222	0.5658317	0.5850307	0.0191989

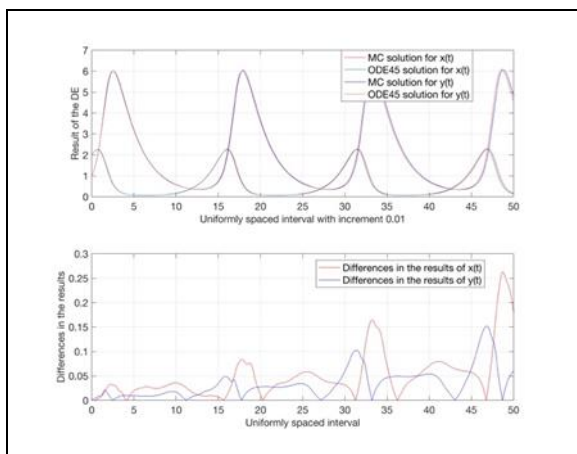


Figure 4: Results of a trial on the interval $[0, 50]$ with 100000 random numbers.

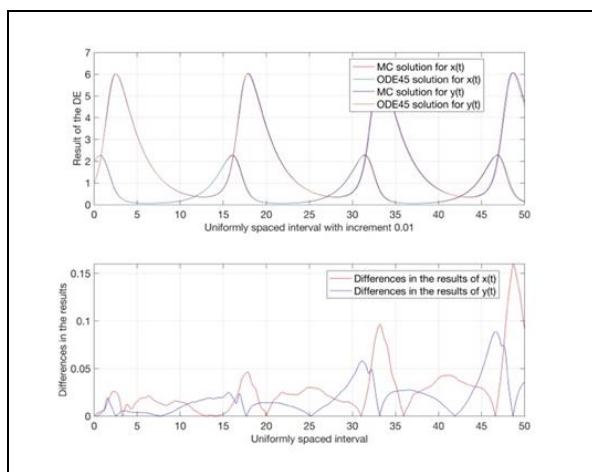


Figure 5: Results of a trial on the interval $[0, 50]$ with 170000 random numbers.

In principle, choices of the parameters; the bounds, number of random variables and initial conditions seem to be optional, in reality it is not the case. Choices of the parameters are realized to be effective on the accuracy of the algorithm as seen in Table 2 and Figure 6. Specifically the effect of the bounds is discussed here as given quantitatively and qualitatively in the

figure and the table. Although it is possible to optimize the bounds as given in the literature [8], in reality it has not been standardized yet. In a similar manner of the analysis of the bounds, effects of the rest of the parameters can be dealt with. Actually, this study primarily focuses on the application of such realistic problems rather than the detailed discussion of the optimum parameters.

Table 2: The comparison of the different upper and lower bounds for the algorithm of Example 1.

	Upper Bound	Lower Bound	Time t	Differences between the results for $x(t)$	Differences between the results for $y(t)$
(a)	85.00	60.00	0.10	0.0013542	0.0022267
			1.00	0.0511746	0.0391757
(b)	9.00	5.00	0.10	0.0000452	0.0015794
			1.00	0.0154289	0.0290938
(c)	7.00	3.00	0.10	0.0000325	0.0014233
			1.00	0.0146405	0.0254132

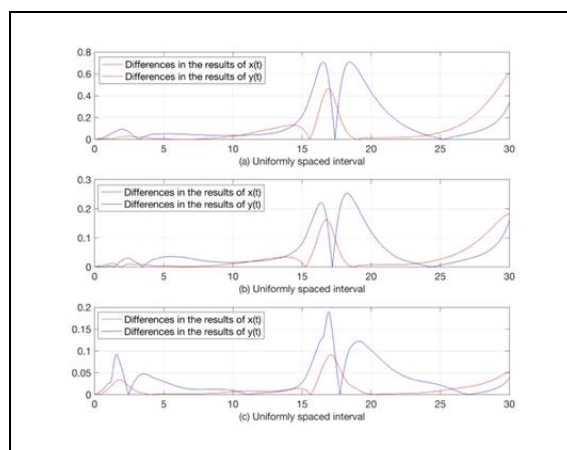


Figure 6: Differences between solutions of the functions for different bounds in Table 2 of Example 1.

4.2 Example 2

In Example 1, a system of first order nonlinear homogeneous differential equations is considered. This system can be considered that two population models live in a closed area

without affecting by external factors in terms of the behavior of population models. However, in the real-life examples, there can be external factors which affect populations positively or negatively apart from the birth or death ratios.

In the following examples, it is analyzed that what happens if external terms are included in equations. In this respect, the structure of a system changed as follows:

Let us consider a system of first order nonlinear differential equations in Equation 3 with constants $\alpha = 2, \beta = 1.2, \gamma = 0.9$ and $\delta = 0.4$ with initial conditions $x_0 = 1$ and $y_0 = 0.5$ and with different non-homogeneous terms $h_x = h(x, t) = 0$ and $h_y = h(y, t) = 0.55$.

Since the non-homogeneous terms are added to differential equations, the qualitative behavior is expected to change. Adding to the extra rate of change to the equation of predator population leads to the change in the growth rates of the populations and this change can be observed from

Figure 7 and 8. Also, it can be predicted that the red line is the qualitative behavior of the prey population since its additional term of a predator causes numerically lower growth rate to prey population. This result can also be observed by numerical results of Table 3. When there is no non-homogeneous term, the behavior of the solutions occurs like periodical oscillation. After adding terms to the equations, behavior becomes decaying oscillations.

As is the case in all simulation techniques, the Monte Carlo method has advantages such as flexibility and easy constructions of its algorithms as well as keeping realistic features of nature. In this study, the proposed algorithm has mostly considered the aspects of its accuracy level and implementation cost. Even if the computational time of the algorithm seems to be higher than the ode45 ones for various amounts of random numbers, the accuracy levels are relatively in good agreement with the referenced results as shown in the Table 4.

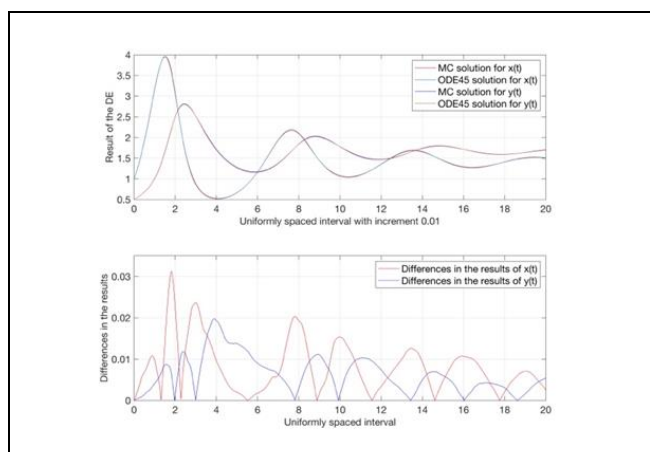


Figure 7: Comparison of the MC predictions and the numerical solutions of equations and differences between them.

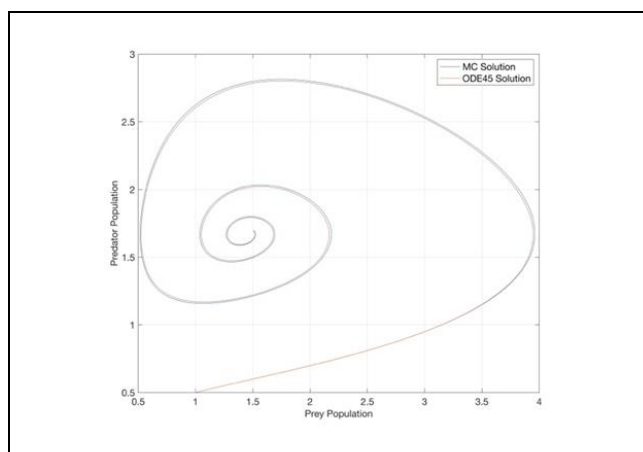


Figure 8: Behaviors of the population functions relative to each other for the equation system.

Table 3: Numerical results for the system of first order nonlinear differential equations in Example 2.

Time t	The proposed algorithm solutions for $x(t)$	ode45 solutions for $x(t)$	Differences between the results for $x(t)$	The proposed algorithm solutions for $y(t)$	ode45 solutions for $y(t)$	Differences between the results for $y(t)$
0.0100	1.0139582	1.0140802	0.0001220	0.5029789	0.5030066	0.0000277
0.0500	1.0713965	1.0720220	0.0006255	0.5150428	0.5151750	0.0001322
0.1000	1.1468563	1.1481711	0.0013148	0.5304795	0.5307518	0.0002723
0.5000	1.9067179	1.9135219	0.0068040	0.6796044	0.6810196	0.0014152
1.0000	3.1620285	3.1718218	0.0097933	1.0048308	1.0089151	0.0040844
2.0000	3.0779638	3.0553538	0.0226100	2.5032768	2.5019906	0.0012861
3.0000	0.8607238	0.8843588	0.0236350	2.5137366	2.5139670	0.0002304
4.0000	0.5163658	0.5259647	0.0095989	1.7004291	1.7197061	0.0192770
5.0000	0.6651553	0.6671276	0.0019723	1.2662871	1.2800386	0.0137516
6.0000	1.1679079	1.1661515	0.0017564	1.1630987	1.1720983	0.0089996
7.0000	1.9396527	1.9334824	0.0061703	1.3708880	1.3772233	0.0063353
8.0000	2.1051190	2.0857946	0.0193244	1.8413106	1.8384128	0.0028978
9.0000	1.4413240	1.4429302	0.0016062	2.0234778	2.0124287	0.0110491
10.0000	1.0647100	1.0800615	0.0153515	1.7776293	1.7790701	0.0014408
15.0000	1.4417665	1.4464625	0.0046960	1.7969120	1.7909153	0.0059967
20.0000	1.5062519	1.5036929	0.0025590	1.7003323	1.6948991	0.0054332

Table 4: Numerical results for the Example 2 with different size of random variables where t_1 is the computational time for proposed algorithm and t_2 is the computational time for ode45 where N is the number of the random variables.

	N=1000	N=10000	N=100000
t_1	0.048049 sec	0.182024 sec	0.896560 sec
t_2	0.029796 sec	0.011975 sec	0.015925 sec

Time t	Differences between the results for $x(t)$	Differences between the results for $y(t)$	Differences between the results for $x(t)$	Differences between the results for $y(t)$	Differences between the results for $x(t)$	Differences between the results for $y(t)$
0.0100	0.0003158	0.0011266	0.0000323	0.0000926	0.0001220	0.0000277
0.0500	0.0004890	0.0056340	0.0003783	0.0004076	0.0006255	0.0001322
0.1000	0.0002729	0.0115288	0.0009130	0.0007517	0.0013148	0.0002723
0.5000	0.0409211	0.0456126	0.0010076	0.0050892	0.0068040	0.0014152
1.0000	0.1829862	0.0705931	0.0264533	0.0131826	0.0097933	0.0040844
2.0000	0.3106042	0.1433714	0.1575567	0.0098115	0.0226100	0.0012861
3.0000	0.0199508	0.1624380	0.0383667	0.1099879	0.0236350	0.0002304
4.0000	0.0701567	0.0800989	0.0280392	0.0861938	0.0095989	0.0192770
5.0000	0.1454776	0.0571664	0.0955737	0.0417063	0.0019723	0.0137516
6.0000	0.2686125	0.0540433	0.2005779	0.0346734	0.0017564	0.0089996
7.0000	0.2123034	0.1992653	0.1867842	0.1170028	0.0061703	0.0063353
8.0000	0.4141254	0.2130148	0.1508918	0.1353254	0.0193244	0.0028978
9.0000	0.4497838	0.1112163	0.2410662	0.0488645	0.0016062	0.0110491
10.0000	0.0895025	0.2241559	0.0565099	0.1312488	0.0153515	0.0014408
15.0000	0.4467215	0.0431173	0.1423845	0.0041636	0.0046960	0.0059967
20.0000	0.0035601	0.1132821	0.0194581	0.0273567	0.0025590	0.0054332

4.3 Example 3

Consider Example 2 with different non-homogeneous terms non-homogeneous terms $h_x = h(x, t) = 0.34$ and $h_y = h(y, t) = 0.55$. These source terms are chosen to observe what happens if predators increase their number of individuals with higher growth rate than preys. The proposed algorithm is applied to the equation with these parameters by dividing the time interval uniformly with the increment 0.01. The comparison of the solutions of the proposed algorithm and the numerical solutions obtained by ode45 have been indicated at Figure 9.

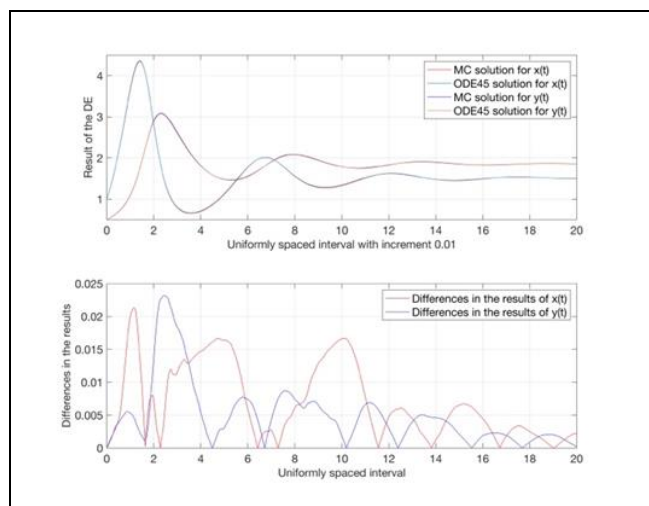


Figure 9: Comparison of the MC predictions and the numerical solutions of equations and differences between the compared results.

The behaviors of population models relative to each other are illustrated in Figure 10. Table 5 shows quantitative results for corresponding trial.

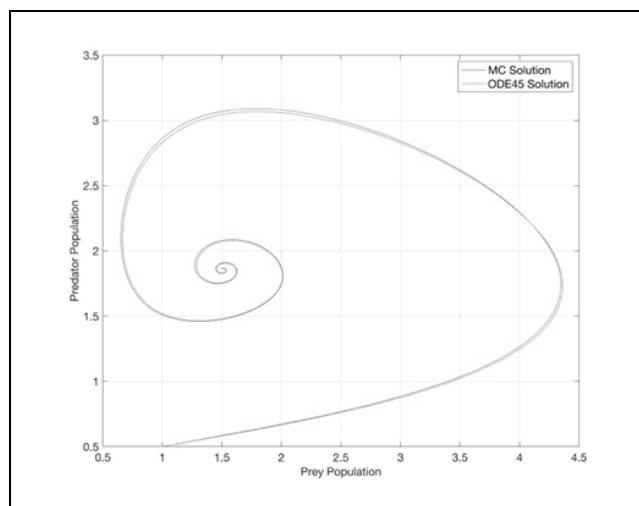


Figure 10: Behaviors of the population functions relative to each other for the equation system.

Since the non-homogeneous terms are added to the differential equations, the results change. Since the higher rate of change added to the equation of predator population, its growth rate increases and this change can be observed from Figure 9. It can be seen that the red line is the qualitative behavior of the prey population since its additional term is lower than predator one and this causes numerically lower growth rate than predator population. This result can also be observed by numerical results of Table 5.

When there is no non-homogeneous term, the behavior of the solutions is seen to be a periodical oscillation. However, after adding the terms to the equations, behaviors transform decaying oscillations. This decaying oscillation can also be observed in Figure 9. The figure includes spiral lines showing behaviors of populations relative to each other. Moreover, the differences between two results seem to behave periodically, as well.

Table 5: Numerical results for the system of first order nonlinear differential equations with nonhomogeneous terms in Example 3.

Time t	The proposed algorithm solutions for $x(t)$	ode45 solutions for $x(t)$	Differences between the results for $x(t)$	The proposed algorithm solutions for $y(t)$	ode45 solutions for $y(t)$	Differences between the results for $y(t)$
0.0100	1.0174345	1.0175040	0.0000695	0.5031001	0.5030100	0.0000901
0.0500	1.0892782	1.0896238	0.0003456	0.5157248	0.5152631	0.0004617
0.1000	1.1838269	1.1846034	0.0007765	0.5320051	0.5311173	0.0008878
0.5000	2.1411886	2.1453872	0.0041986	0.6976956	0.6938275	0.0038681
1.0000	3.6654637	3.6854659	0.0200023	1.0983932	1.0930748	0.0053184
2.0000	2.8166322	2.8094390	0.0071932	2.9053176	2.8916546	0.0136630
3.0000	0.7896117	0.8011591	0.0115474	2.6188652	2.6004778	0.0183874
4.0000	0.7008756	0.7155807	0.0147051	1.8115697	1.8061867	0.0053830
5.0000	1.1174976	1.1339312	0.0164336	1.4771927	1.4802482	0.0030555
6.0000	1.7729546	1.7788439	0.0058893	1.5526819	1.5600778	0.0073959
7.0000	1.9834693	1.9807358	0.0027335	1.9077857	1.9035250	0.0042607
8.0000	1.5543493	1.5604023	0.0060531	2.0856559	2.0782381	0.0074178
9.0000	1.2872653	1.3001082	0.0128429	1.9470829	1.9405551	0.0065277
10.0000	1.3282333	1.3448332	0.0165999	1.7890159	1.7871891	0.0018268
15.0000	1.4498878	1.4564152	0.0065274	1.8535635	1.8514698	0.0020937
20.0000	1.4937019	1.4959447	0.0022428	1.8578928	1.8576953	0.0001975

5 Conclusions and recommendation

In this study, a Monte Carlo based stochastic algorithm has been developed to solve deterministic models.

All qualitative and the quantitative results of the problems of interest found by the proposed algorithm have been seen to be consistent with the behavior of population models in nature. Despite the randomness in error interval, the proposed algorithm has been seen to produce accurate results. This study showed that the algorithm can be effectively used for reducing the computational cost. The qualitative results created by the proposed algorithm revealed that the structure of the algorithm is appropriate for physical processes represented by these types of differential equations.

6 Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions for improving the paper. The first author also would like to thank Scientific and Technological Research Council of Turkey (TUBITAK), under the 2211-E Program which supports the author.

7 References

- [1] Sadiku MNO. *Monte Carlo Methods for Electromagnetics*. 1st ed. Taylor and Francis Group, 2009.
- [2] Göncü A. Monte Carlo and Quasi-Monte Carlo Methods in Financial Derivative Pricing. PhD Thesis, Florida State University, Florida, USA, 2009.
- [3] Sobol IM. *A Primer for the Monte Carlo Method*. 1st ed. Boca Raton, CRC Press, 1994.
- [4] Taylor HM, Karlin S. *An Introduction to Stochastic Modeling*. 3rd ed. London, UK, Wiley, 1998.
- [5] Resnick S. *Adventures in Stochastic Processes*. 4th ed. New York, USA, Birkhauser Basel, 1992.
- [6] Kroese DP, Taimre T, Botev ZI. *Handbook of Monte Carlo Methods*. 1st ed. New Jersey, USA, Wiley, 2011.
- [7] Paul S, Mondal SP, Bhattacharya P. "Numerical solution of lotka volterra prey predator model by using runge-kutta-fehlberg method and laplace adomian decomposition method". *Alexandria Engineering Journal*, 55, 613-617, 2015.
- [8] Akhtar MN, Durad MH, Ahmed A. "Solving initial value ordinary differential equations by Monte Carlo method". *Proceedings of IAM*, 4(2), 149-174, 2015.
- [9] Zhong W, Tian Z. "Solving initial value problem of ordinary differential equations by Monte Carlo method". *2011 International Conference on Multimedia Technology (ICMT)*, Hangzhou, China, 26-28 July 2011.
- [10] Fishman G. *Monte Carlo: Concepts, Algorithms and Applications*. 1st ed. Stanford, USA, Springer, 1999.
- [11] Rubinstein RY, Kroese DP. *Simulation and The Monte Carlo Method*. 3rd ed. New Jersey, USA, Wiley, 2017.
- [12] Ross SM. *A First Course in Probability*. 8th ed. New Jersey, USA, Pearson, 2010.
- [13] Uslu H. Behavior of First Order Differential Equations Through a Monte Carlo Based Algorithm. MSc Thesis, Yildiz Technical University, Istanbul, Turkey, 2018.
- [14] Sari M, Uslu H. "Monte Carlo Based Stochastic Approach for Some Nonlinear ODE Systems". *9th International Conference on Numerical Methods and Applications (NM&A'18)*, Borovets, Bulgaria, 20-24 August 2018.
- [15] Chapra CS. *Applied Numerical Methods with MATLAB for Engineers and Scientists*. 4th ed. New York, USA, McGraw Hill, 2017.
- [16] Murray JD. *Mathematical Biology*. 2nd ed. Seattle, USA, Springer, 2002.