Journal of Computational Statistics and Modeling, Vol. 1, No. 2, 139-151, December 2021

Research Manuscript

New Adaptive Monte Carlo Algorithm to Solve Financial Option Pricing Problems

Mahboubeh Aalaei[1*]

[1] Assistant Professor, Insurance Research Center, Saadat Abad, Tehran, Iran.

Recieved: 29/04/2021 Accepted: 04/11/2021

Abstract:

In this paper, a new adaptive Monte Carlo algorithm is proposed to solve the systems of linear algebraic equations arising from the Black–Scholes model to price European and American options. The proposed algorithm, offers several advantages over the conventional and previous adaptive Monte Carlo algorithms. The corresponding properties of the algorithm and Convergence theories are discussed, and numerical experiments are presented, which demonstrate the computational efficiency of the proposed algorithm. The results are also compared with other methods.

Keywords: Adaptive Monte Carlo algorithm, Finite difference method, Black–Scholes model, European and American option.

Mathematics Subject Classification (2010): 65L05, 34K06, 34K28.

*Corresponding Author: aalaei@irc.ac.ir

1. Introduction

High dimensional systems of linear algebraic equations (SLAEs) are arising from real world problems (e.g., Alexandrov and *et al.* (2003, 2011); Dimov and *et al.* (1998)). The large linear systems can be obtained directly or after discretization of partial differential or integral equations (e.g., Baykus and Sezer (2010); Dehghan and Hajarian (2012)). Therefore the choice of an appropriate approach for solving large sparse SLAEs is a problem of unquestionable importance in many scientific and engineering applications.
Adaptive Monte Carlo algorithms are stochastic algorithms which are preferable for solving high dimensional SLAEs. Some advantages of adaptive Monte Carlo algorithms described in Rubinstein (1981); Alexandrov and *et al.* (2011); Farnoosh and Aalaei (2015) are:

- They are more efficient than direct or iterative numerical algorithms.
- They are good candidates for parallelization.
- They have much faster convergence and need fewer random paths than the conventional method does.

Adaptive Monte Carlo methods introduced by Halton to estimate the solution of SLAEs in Halton (1962). These methods improve the convergence of the conventional Monte Carlo method exponentially. Because of the advantages of these methods, several researchers have implemented them in different areas. Empirical studies have been discussed for two adaptive Monte Carlo methods with geometric convergence in Lai (2009). In Farnoosh and Aalaei (2015); Farnoosh and *et al.* (2015); Aalaei and Manteqipour (2021), adaptive Monte Carlo algorithms were proposed using the refinement method for solving SLAE with more accurate results, and the proposed algorithms were implemented to solve option pricing problems and two dimensional Fredholm integral equations. In Dimov and *et al.* (2015), a Monte Carlo algorithm was proposed for solving SLAEs, and it was combined with the sequential Monte Carlo method to improve the results. A new Monte Carlo method for solving SLAEs is presented in Vajargah and Hasanzadeh (2020) and the convergence of the method is discussed.

The advantages of adaptive Monte Carlo algorithms are author motivation for study on this work. In this paper, we propose an adaptive Monte Carlo algorithm that needs fewer random paths than adaptive algorithms presented in Farnoosh and Aalaei (2015) and Lai (2009). To confirm the efficiency of the proposed algorithm, the adaptive Monte Carlo algorithms are applied to approximate the value of European and American options. According to our best knowledge, Monte Carlo methods have been widely applied to option pricing and other financial problems (see e.g., Han and Lai (2010); Jasra and Moral (2011)). However evaluating option prices based on the proposed algorithm, has been investigated for the first time in this paper.

2. Option Pricing

The well known Black Scholes model for pricing European put option has been described in Farnoosh and Aalaei (2015) by the equation:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r-q)S\frac{\partial V}{\partial S} - rV = 0, \tag{2.1}$$

with the final condition $V(S,T) = max(E-S,0)$ and boundary conditions $V(0,t) = Ee^{-r(T-t)}$and $V(S,t) \approx 0$ as $S \to \infty$, where S, E, T, r are the current price of the asset, the strike price, the expiry time, the risk free interest rate, respectively. Also, S is assumed to behave $dS = (r-q)Sdt + \sigma SdW$, where dW is a Wiener process, r and σ are the drift rate and the volatility of the asset, respectively. In this case, there is the closed form solution. However, for more styles of options, there are not closed form solutions. Stochastic methods can be used to price these options. In this regard, the adaptive Monte Carlo algorithm can be used to value European options, and pricing formulas for these options can be checked using this method.

The general discretization (finite difference) method can be used to approximate the solution of (2.1), Wilmott and *et al.* (1995) where $\theta \in (0,1)$ is the parameter of discretization. Assume that $V_{ij} = V(i\Delta_S, j\Delta_t), 0 < i < N, 0 \leq j \leq M$. The Black Scholes model can be formulated as the following linear systems:

$$CV^{j+1} = DV^j + b^j, \tag{2.2}$$

where

$$C = \begin{bmatrix} 1-\theta m_1 & -\theta u_1 & 0 & \cdots & 0 \\ -\theta d_2 & 1-\theta m_2 & -\theta u_2 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -\theta d_{N-1} & 1-\theta m_{N-1} \end{bmatrix},$$

$$b^j = \begin{bmatrix} \theta d_1 V_{0j} + (1-\theta) d_1 V_{0j+1} \\ 0 \\ \vdots \\ 0 \\ \theta u_{N-1} V_{Nj} + (1-\theta) u_{N-1} V_{Nj+1} \end{bmatrix},$$

$$D = \begin{bmatrix} 1+(1-\theta)m_1 & (1-\theta)u_1 & 0 & \cdots & 0 \\ (1-\theta)d_2 & 1+(1-\theta)m_2 & (1-\theta)u_2 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & (1-\theta)d_{N-1} & 1+(1-\theta)m_{N-1} \end{bmatrix},$$

where $d_i = \frac{\Delta_t \sigma^2 S_i^2}{2\Delta_S^2} - \frac{\Delta_t (r-q) S_i}{2\Delta_S}$, $m_i = -\frac{\Delta_t \sigma^2 {S_i}^2}{\Delta_S^2} - \Delta_t r$, $u_i = \frac{\Delta_t \sigma^2 {S_i}^2}{2\Delta_S^2} + \frac{\Delta_t (r-q) S_i}{2\Delta_S}$. The linear system obtained in each time step will be approximated using the adaptive Monte Carlo algorithm, and at the end, the solution vector will be the price of the European option. To describe an American put option using the Black Scholes model, we assume $V(S,t) \geq max(E-S,0)$, final condition $V(S,T) = max(E-S,0)$ and boundary conditions $V(0,t) = Ee^{-r(T-t)}$ and $V(S,t) \approx 0$ as $S \to \infty$.

For American option pricing, the linear system (2.2) should be solved in each time step, and the solution vector should be compared with the final condition, and the result will be the solution vector in that time step. The solution vector will be calculated for $j = M, \ldots, 0$ and at the end, the solution vector will be the price of the American option.

We solve the linear system (2.2) using the following adaptive Monte Carlo algorithms.

3. Adaptive Monte Carlo algorithms

Monte Carlo algorithms have proved to be a valuable and flexible computational tool in modern finance and have been developed within the past years to price the options. Also, It is well known that Monte Carlo methods are more effective and more preferable than direct and iterative numerical methods for solving large sparse systems. In this chapter, we present, propose and analyze adaptive Monte Carlo algorithms for solving the linear systems obtained using finite difference for option pricing. We then proceed to analyze the convergence of the algorithm, and discuss the corresponding properties of the algorithms. Consider that we are going to solve the system of linear equations

$$Bx = F, \tag{3.3}$$

using Monte Carlo algorithms. Introducing $A = A_ij_{i,j=1}^n = I - B$, where I is an identity matrix, we have $x = Ax + F$ and therefore using recursive formula

$$x^{(k+1)} = Ax^{(k)} + F, \tag{3.4}$$

We have an estimator for x under the assumption $max_i \sum_{j=1}^n |A_ij| < 1$ and the following Monte Carlo algorithms converge. In all following algorithms, independent random paths of Markov chain will be simulated with initial distribution $p = (p_1, \cdots, p_n)$ and transition matrix P.
In this section, for better understanding the differences between algorithms, we discuss the conventional Monte Carlo method and the Halton adaptive Monte Carlo algorithm presented in Halton (1962). Also, the adaptive Monte Carlo algorithm presented in Farnoosh and Aalaei (2015) is reviewed, and a new adaptive Monte Carlo algorithm is proposed.

3.1 Conventional Monte Carlo algorithm

The base of the conventional Monte Carlo method described in Rubinstein (1981) is to express each component of the solution vector as the expectation of some random variable. To estimate the inner product $\langle h, x^{(k+1)} \rangle$, we generate Z random paths $i_0^{(s)} \to i_1^{(s)} \to \cdots \to i_k^{(s)}$ and calculate θ_k via $\theta_k(h) = \frac{1}{Z} \sum_{s=1}^Z \eta_k^{(s)}(h)$, where $\eta_k^{(s)}(h) = \frac{h_{i_0^{(s)}}}{p_{i_0^{(s)}}} \sum m = 0^k w_m^{(s)} F_{i_m^{(s)})}$ and $w_m^{(s)} = w_{m-1}^{(s)} \frac{A_{i_{m-1}^{(s)} i_m^{(s)}}}{P_{i_{m-1}^{(s)} i_m^{(s)}}}, w_0^{(s)} = 1$.

3.2 Halton adaptive Monte Carlo algorithm

For Halton adaptive Monte Carlo algorithm described in Halton (1962), Consider $F^{(0)} = F, \theta_k^{(0)} = 0, F^{(d)} = F^{(d-1)} - B\theta_k^{(d-1)}, d = 1, \cdots, r$, where r is the number of stages and $\theta_k^{(d)}$ is the approximate solution of

$$B\Delta^d x = F^{(d)}, \tag{3.5}$$

using described conventional Monte Carlo method which random paths are generated through a fixed transition matrix P. Then

$$\varphi_k^{(d)}(h) = \varphi_k^{(d-1)}(h) + \theta_k^{(d)}(h),$$

is the approximated solution of SLAE (3.3). It is shown in Halton (1962) that

$$\lim_{r\to\infty} F^{(r)} = 0,\ \lim_{r\to\infty} \theta_k^{(r)} = 0,\ \lim_{r\to\infty} \varphi_k^{(r)} = x_j,$$

where x_j is the j th component of the exact solution to SLAE (3.3). Note that if $r = 1$, we have the conventional Monte Carlo method.

3.3 Adaptive Monte Carlo algorithm in Farnoosh and Aalaei (2015)

In the adaptive Monte Carlo algorithm proposed in Farnoosh and Aalaei (2015), which we call AMC1, the transition matrix P is fixed for all stages. Therefore we can use the same random paths generated through the transition matrix P for all of the stages. It means that, we do not need to calculate $w_m^{(s)}$ for each stage because they are fixed for all stages.

3.4 Proposed Adaptive Monte Carlo algorithm

An idea proposed by Spanier to improve the efficiency of adaptive Monte Carlo methods is to use a branching process in which many correlated random walks are processed in parallel. As described in Lai (2009), in a branching process, random walks are generated using the same rules as before, but instead of computing one component at a time corresponding to each discrete source index, each component whose index is visited by a random walk receives contributions. A simple example is used to illustrate this idea. Suppose that A is a 3×3 matrix and, the random path is $1 \to 3 \to 1 \to 2 \to 3 \to 1$. Then for estimating x_1, the path $1 \to 3 \to 1 \to 2 \to 3 \to 1$ is used, and the paths $2 \to 3 \to 1$ and $3 \to 1 \to 2 \to 3 \to 1$ are used to estimate x_2 and x_3, respectively.

In the new adaptive Monte Carlo method proposed in this article, we use this idea and the assumption that the initial distribution p and the transition matrix P of the Markov chain are fixed for all components and all stages. Therefore, the same random paths are used for all components of the solution vector and all stages. Then the total number of random paths with length $k(> n)$ to estimate the solution vector in our proposed algorithm, is Z which the length of random paths should be at least n. So the total number of random variables in the proposed algorithm is at least nZ. We note that the total number of random variables in our proposed algorithm in Farnoosh and Aalaei (2015) is knZ. Also, the total number of random variables for the presented algorithm in Lai (2009) is at least

rnZ. Therefore the proposed algorithm in this paper needs less random variables comparative to both of mentioned algorithms, and it can be less time consuming. Furthermore, the proposed algorithm has a simple structure, low cost, desirable speed, and accuracy and is easy to be parallelized.

3.5 Convergence

To examine the convergence of the proposed algorithm, we should define some notations as follows. Consider $F^{(0)} = F, \Delta^0 x = x$ and Eq. (3.3) for stage r as

$$B\Delta^r x = F^{(r)}, \tag{3.6}$$

where $\Delta^r x$ and $F^{(r)}$ are obtained by the following recursive equations

$$\Delta^r x = \Delta^{r-1} x - \Delta_k^{r-1} x,$$

$$F^{(r)} = F^{(r-1)} - B\Delta_k^{r-1} x,$$

and $\Delta_k^r x$ is the approximate solution of SLAE (3.3) obtained by using Eq. (3.4), k times. Considering $S_0 = \Delta_k^0 x$ and $S_r = S_{r-1} + \Delta_k^r x$, clearly, we have

$$x = S_r + \Delta^{r+1} x, \tag{3.7}$$

and the following theorem will be proven.

Theorem 3.1. *Under the assumption $\|A\| < 1$ and $\Delta_0^r x = 0$, as k, Z and r tend to infinity, $\varphi_k^{(r)}$ converges to x.*

Proof. From Eq. (3.7), we have $x = \sum_{d=1}^{r} \Delta_k^d x + \Delta^{r+1} x$. We prove the theorem in two parts. At First, we prove that $\lim_{r\to\infty} \Delta^r x = 0$, and then we prove $\theta_k^{(r)}$ converges to $\Delta_k^r x$. Therefore, the theorem will be concluded.

From Eq. (3.4) and (3.5), we have

$$\Delta^r x = A\Delta^r x + F^{(r)},$$

$$\Delta_k^r x = A\Delta_{k-1}^r x + F^{(r)}.$$

Then we can obtain

$$\Delta^r x = \Delta^{r-1} x - \Delta_{k-1}^{r-1} x = A(\Delta^{r-1} x - \Delta_{k-1}^{r-1} x) = \cdots = A^k \Delta^{r-1} x,$$

and

$$\Delta^r x = A^k \Delta^{r-1} x = A^{2k} \Delta^{r-2} x = A^{3k} \Delta^{r-3} x = \cdots = A^{(r-1)k} x.$$

Therefore

$$\|\Delta^r x\| \leq \|A^{(r-1)k}\|.\|x\|, \tag{3.8}$$

Since $\|A\| < 1$, then $B = I - A$ is invertible and has a unique solution. Therefore $\|x\|$ is finite. As r tends to infinity, taking the limit of Eq. (3.8) will conclude $\lim_{r\to\infty} \Delta^r x = 0$ and the first part of the proof is completed.

Since the random variable $\eta^{(r,s)}_{k-j_t}(h)$ is defined along the path, $i^{(s)}_{j_t} \to i^{(s)}_{j_t+1} \to \ldots \to i^{(s)}_k$ we have

$$E[\eta^{(r,s)}_{k-j_t}(h)] = \sum_{i^{(s)}_k}^{n} \cdots \sum_{j_t}^{n} \eta^{(r,s)}_{k-j_t}(h) P_{i^{(s)}_{j_t} i^{(s)}_{j_t+1}} \cdots P_{i^{(s)}_{k-1} i^{(s)}_k},$$

therefore

$$\begin{aligned} E[\eta^{(r,s)}_{k-j_t}(h)] &= E[\sum_{m=j_t}^{k} w^{(s)}_m F^{(r)}_{i^{(s)}_m}] \\ &= \sum_{i^{(s)}_{j_t}=1}^{n} \cdots \sum_{i^{(s)}_k=1}^{n} \sum_{m=j_t}^{k} A_{i^{(s)}_{j_t} i_{j^{(s)}_{t+1}}} \cdots A_{i^{(s)}_{m-1} i^{(s)}_m} F^{(r)}_{i^{(s)}_m} P_{i^{(s)}_m i^{(s)}_{m+1}} \cdots P_{i^{(s)}_{k-1} i^{(s)}_k} \\ &= \sum_{m=j_t}^{k} \sum_{i^{(s)}_{j_t}=1}^{n} \cdots \sum_{i^{(s)}_m=1}^{n} A_{i^{(s)}_{j_t} i^{(s)}_{j_t+1}} \cdots A_{i^{(s)}_{m-1} i^{(s)}_m} F^{(r)}_{i^{(s)}_m}. \end{aligned}$$

The last equation is obtained using the property $\sum_{j=1}^{n} P_{ij} = 1$ and we immediately obtain

$$E[\eta^{(r,s)}_{k-j_t}(h)] = \langle h, \sum_{m=0}^{k-j_t} A^m F^{(r)} \rangle = \langle h, \Delta^r_{k-j_t} x \rangle.$$

And therefore as Z tends to infinity, $\theta^{(r)}_k = \frac{1}{Z} \sum_{s=1}^{Z} \eta^{(r,s)}_k$ converges to $\Delta^r_k x$. So, the second part of the proof is completed.

$\square$

4. Numerical Test Results

4.1 Linear Equations

In this subsection, we obtain solutions of linear equations using the proposed adaptive Monte Carlo algorithm and compare the results with Farnoosh and Aalaei (2015). If x is the exact solution of the linear system and $x^{(r)}$ is the approximate solution using the adaptive Monte Carlo method at stage r , the L_2 absolute estimate will be

$$\|x - x^{(r)}\| = (\sum_{i=1}^{n}(x_i - {x_i}^{(r)})^2)^{\frac{1}{2}}.$$

But the exact solution x is not known, then we use the following formula as the absolute error of estimation,

$$(\sum_{i=1}^{n}({x_i}^{(r)} - \sum_{j=1}^{n} A_{ij}{x_j}^{(r)} - F_i)^2)^{\frac{1}{2}}.$$

Example 4.1. *Consider a linear system with*

$$A_{ij} = \frac{\rho_{ij} r_i}{\sum_{k=1}^{n} \rho_{ik}},$$

where $r_i = c + \rho_i(d - c)$ *and* $c = min_i \sum_{j=1}^{n} A_{ij}$, $d = max_i \sum_{j=1}^{n} A_{ij} = \|A\|$ *and* ρ_i *and* ρ_{ij} *are pseudo-random numbers uniformly distributed in (0, 1) and* $F_i = i$, *Lai (2009). We consider* $c = 0.25$, $d = 0.75$ *and* $Z = 100$.

The results are shown in Table 1 and 2. It is clear that the proposed adaptive Monte Carlo algorithm called Proposed AMC and algorithm in Farnoosh and Aalaei (2015) called AMC1, converges exponentially. However, as we described in the previous section, in the new algorithm, the number of generated random numbers and, therefore, the corresponding calculations is reduced. Furthermore, the Algorithm in Lai (2009) called AMC2 diverged.

4.2 Option Pricing

Example 4.2. *Consider an European put option with* $E = 12, T = 0.5, r = 0.05, \sigma = 0.2, q = 0, S_{min} = 0, S_{max} = 100, N = 300, M = 100$.

We obtained the following numerical results. The values obtained by the proposed algorithm, those obtained by the algorithm proposed in Farnoosh and Aalaei (2015) and the Black Scholes formula are shown in Table 3. Also, the difference

Table 1: The absolute error of estimation for algorithms in Farnoosh and Aalaei (2015); Lai (2009) and the proposed algorithm for $n = 1000$

Stage	$k = 15000$	$k = 20$	
	Proposed AMC	AMC1	AMC2
1	2.4704×10^{3}	4.9526×10^{3}	4.995×10^{4}
5	9.8124×10^{-2}	2.2295×10^{-1}	2.381×10^{5}
9	2.4401×10^{-6}	4.8934×10^{-6}	5.047×10^{5}
13	5.2894×10^{-11}	1.0559×10^{-10}	8.502×10^{5}
17	4.8355×10^{-12}	1.9700×10^{-11}	1.233×10^{6}

Table 2: The absolute error of estimation for algorithms in Farnoosh and Aalaei (2015); Lai (2009) and the proposed algorithm for $n = 3000$

Stage	$k = 45000$	$k = 20$	
	Proposed AMC	AMC1	AMC2
1	1.1784×10^{4}	2.9863×10^{4}	1.4247×10^{5}
5	4.3762×10^{0}	1.4266×10^{1}	3.2243×10^{5}
9	7.5563×10^{-4}	6.0000×10^{-3}	5.2573×10^{5}
13	9.7764×10^{-7}	2.5320×10^{-6}	7.4809×10^{5}
17	7.5704×10^{-10}	1.0993×10^{-9}	9.7569×10^{5}

between the solution obtained proposed algorithm and the Black Scholes model as the error.

Example 4.3. *Consider an American put option with* $E = 50, T = 3, r = 0.05, \sigma = 0.25, q = 0, S_{min} = 0, S_{max} = 2 \times S, N = 100.$

The values obtained by the proposed algorithm and those obtained by the algorithm considered in Richardson (2009) are shown in Table 4.

Conclusion

In this paper, we have proposed an adaptive Monte Carlo algorithm to solve large linear systems of algebraic equations with less random number generation and therefore more efficient than adaptive Monte Carlo algorithms, which have been introduced before in Farnoosh and Aalaei (2015); Lai (2009). We have analyzed the convergence and efficiency of the algorithm in the case of dealing with large random matrices with sizes 1000 and 3000. It is clear that the proposed adaptive

Table 3: A comparison with the Black-Scholes price for an European put option using the proposed algorithm

Asset price	Proposed AMC	AMC1	Black Scholes	Error
4	7.703718942	7.703718940	7.703718944	1.5199×10^{-9}
6	5.703719949	5.703719970	5.703719211	7.3804×10^{-7}
8	3.705425339	3.705517843	3.705213181	2.1215×10^{-4}
10	1.805367502	1.805287951	1.805980466	6.1296×10^{-4}
12	0.527094565	0.526912877	0.530366373	3.2718×10^{-3}
14	0.087240205	0.087149526	0.088259744	1.0195×10^{-3}

Table 4: Comparison of our algorithm with other methods in Richardson (2009) for an American put option price

Asset price	Proposed AMC $M = 500$	Proposed AMC $M = 1000$	Binomial Tree	Implicit Euler
30	20.0000	20.0000	20.0000	20.0000
35	15.0000	15.0000	15.0147	15.0291
40	10.9212	10.9228	10.9440	10.9492
45	7.9862	7.9924	7.9999	7.9904
50	5.8468	5.8476	5.8547	5.8527
55	4.2811	4.2817	4.2955	4.2891
60	3.1353	3.1358	3.1541	3.1456

algorithm and algorithm in Farnoosh and Aalaei (2015) converges exponentially, but the number of generated random numbers and, therefore, the corresponding calculations are reduced in the proposed algorithm. Furthermore, the proposed algorithm has been implemented to solve sparse matrices which are arising from the discretization of parabolic partial differential equation arising from option pricing. The results show the efficiency and accuracy of the proposed adaptive Monte Carlo algorithm for pricing European and American options pricing. The examples have been compared with other methods which demonstrate the efficiency of the proposed algorithm.

References

Aalaei M. and Manteqipour M. (2021), An adaptive Monte Carlo algorithm for European and American options, *Computational Methods for Differential Equations,* In Press.

Alexandrov V. N., Dimov I. T., Karaivanova A. and Tan C.J.K. (2003), Parallel Monte Carlo algorithms for information retrieval, *Mathematics and Computers in Simulation,* **62**, 289-295.

Alexandrov V. N., Martel C. G. and Straßburg J. (2011), Monte Carlo scalable algorithms for computational finance, *Procedia Computer Science,* **4**, 1708-1715.

Baykus N. and Sezer M. (2010), Solution of high-order linear Fredholm integro-differential equations with piecewise intervals, *Numerical Methods for Partial Differential Equations,* **27(5)**, 1327-1339.

Dehghan M. and Hajarian M. (2012), SSHI methods for solving general linear matrix equations, *Engineering Computations,* **28(8)**, 1028-1043.

Dimov I. T., Dimov T. T. and Gurov T. V. (1998), A new iterative Monte Carlo approach for inverse matrix problem, *Journal of Computational and Applied Mathematics,* **92**, 15-35.

Dimov I., Maire S., Sellier J. M. (2015), A new walk on equations Monte Carlo methods for linear algebraic problems, *Applied Mathematics Modelling,* **39**, 4494-4510.

Farnoosh R. and Aalaei M. (2015), New adaptive Monte Carlo algorithm for parallel solution of large linear systems with applications, *Proceedings of the Romanian Academy Series A,* **16(1)**, 11-19.

Farnoosh R., Aalaei M. and Ebrahimi M. (2015), Combined probabilistic algorithm for solving high dimensional problems, *Stochastics: An International Journal of Probability and Stochastic Processes,* **87(1)**, 30-47.

Halton J. (1962), Sequential Monte Carlo, *Proceedings of the Cambridge Philosophical Society,* **58**, 57-78.

Han C. H. and Lai Y. (2010), A smooth estimator for MC/QMC methods in finance, *Mathematics and Computers in simulation,* **81(3)**, 536-550.

Jasra A. and Moral P.D. (2011), Sequential Monte Carlo methods for option pricing, *Stochastic analysis and applications,* **29**, 292-316.

Lai Y. (2009), Adaptive Monte Carlo methods for matrix equations with applications, *Journal of Computational and Applied Mathematics,* **231**, 705-714.

Richardson M. (2009), *Numerical Methods for Option Pricing*, University of Oxford, Special topic.

Rubinstein R. Y. (1981), *Simulation and the Monte Carlo method*, Wiley, New York.

Vajargah B. F., Hasanzadeh Z., (2021). A new Monte Carlo method for solving system of linear algebraic equations, *Computational Methods for Diffrential Equations,***9(1)**, 159-179.

Wilmott P., Dewynne J. and Howison S. (1995), *Option pricing: mathematical models and computation*, Oxford University Press.