Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 190 (2007) 1593–1601
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

Monte Carlo method via a numerical algorithm to solve a parabolic problem

R. Farnoosh *, M. Ebrahimi

School of Mathematics, Iran University of Science and Technology, Narmak, Tehran 16844, Iran

Abstract

This paper is intended to provide a numerical algorithm consisted of the combined use of the finite difference method and Monte Carlo method to solve a one-dimensional parabolic partial differential equation. The numerical algorithm is based on the discretize governing equations by finite difference method. Due to the application of the finite difference method, a large sparse system of linear algebraic equations is obtained. An approach of Monte Carlo method is employed to solve the linear system. Numerical tests are performed in order to show the efficiency and accuracy of the present work.

Keywords: Monte Carlo method; Markov chain; Finite difference method; Parabolic partial differential equation; System of linear algebraic equations; Complexity; Efficiency

1. Introduction

This paper represents the numerical estimation of temperature distribution in a one-dimensional parabolic partial differential equation using a numerical algorithm based on Monte Carlo method. Monte Carlo methods for the parabolic partial differential equations are widespread. There are several basic advantages of Monte Carlo algorithms. It is clear that Monte Carlo algorithms are parallel algorithms. They have high parallel efficiency where parallel computers are used [1,2].

At the beginning of the present study, finite difference method is employed to discretize the problem domain. Owing to the application of the finite difference method, a large sparse system of linear algebraic equations is obtained. There are many classical numerical algorithms to solve large systems of linear algebraic equations

$$Ax = b, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ and $x, b \in \mathbb{R}^n$.

* Corresponding author.
E-mail addresses: rfarnoosh@iust.ac.ir (R. Farnoosh), mo.ebrahimi@iust.ac.ir (M. Ebrahimi).

0096-3003/$ - see front matter
doi:10.1016/j.amc.2007.02.102

It is well known that Monte Carlo methods are preferable for solving large sparse systems, such as those arising from approximations of partial differential equations [3]. Such methods are good for diagonally dominant systems in which the rate of the convergence is high, such as the system discussed in this article. One of the most important advantages of Monte Carlo methods is that they can be used to evaluate only one component of the solution or some linear form of the solution. This advantage is of great practical interest, since the most important problems in the applied sciences are formulated as problems of evaluating linear or nonlinear forms of the solution. In this case, it is not necessary to perform all computational work which is needed to obtain a complete solution [1]. In addition, even though Monte Carlo methods do not yield more accurate solutions than direct or iterative numerical methods for solving systems of linear algebraic equations, they are more efficient for large n [4]. The numerical experiments of the present work show that Monte Carlo method is preferable when one needs to have a coarse estimation of the solution.

1.1. The efficiency of Monte Carlo method

An important parameter of the algorithmic efficiency is the computational complexity or the time of the algorithm. For instance, in [5] direct methods such as the non-pivoting Gaussian elimination or Gauss–Jordan methods were applied to solve system (1) and the time was

$$T_{\mathrm{DIRECT}}(n) = \mathrm{O}(n^3). \tag{2}$$

While the iterative methods, such as Jacobi, Gauss–Sidel and various relaxation techniques, take time

$$T_{\mathrm{ITER}}(n,k) = \mathrm{O}(n^2 k), \tag{3}$$

if there are k iterations [1].

On the other hand, to compute the full solution vector using Monte Carlo method the total time required is

$$T_{\mathrm{MC}}(n) = \mathrm{O}(nkN), \tag{4}$$

or less, where N is the number of Markov chains and k is the length of the Markov chain. It should be mentioned that both of these quantities are independent of n and are bounded [4]. In comparison with iterative methods, n is replaced by N in Eq. (4). Thus, when $N < n$, Monte Carlo method is far more efficient than the classical methods.

2. Statement of the problem

In this paper we will consider a direct problem of determining an unknown function $U(x,t)$ in a one-dimensional linear parabolic equation. The problem is that one needs to find the temperature distribution $U(x,t)$ that satisfies initial boundary value problem

$$U_t = a(t)U_{xx}, \quad 0 < x < 1, \ \ t > 0, \tag{5}$$

$$U(x,0) = f(x), \quad 0 < x < 1, \tag{6}$$

$$U(0,t) = \varphi(t), \quad t > 0, \tag{7}$$

$$U(1,t) = \psi(t), \quad t > 0, \tag{8}$$

where $f(x)$, $\varphi(t)$ and $\psi(t)$ are continuous known functions. It is worth noting that $a(t) > 0$ [6] is known. Certain types of physical problems can be modeled by (5)–(8). The coefficient $a(t)$ can represent physical quantities, for example, the conductivity of a medium. The existence and uniqueness of the solutions to this problem and also some more applications are discussed in [7]. The numerical solution of the problem (5)–(8) has been discussed by several authors [6].

Theorem 1. *The problem (5)–(8) has a unique solution if $f(x)$, $\varphi(t)$ and $\psi(t)$ are continuous functions and $a(t) > 0$* [7].

3. Numerical algorithm

The numerical algorithm applied in this study is discussed in the following.

3.1. Finite difference method for discretizing

At first we use fully implicit finite difference approximation [8] for discretizing problem (5)–(8). Therefore, Eq. (5) is approximated at the point (p,q) by the difference equation

$$F_{p,q}(u)=\frac{u_{p,q+1}-u_{p,q}}{v}-a_{q+1}\frac{u_{p-1,q+1}-2u_{p,q+1}+u_{p+1,q+1}}{\mu^2}=0, \tag{9}$$

and as a result, from Eqs. (5)–(8) we obtain

$$u_{p,q}=-ra_{q+1}u_{p-1,q+1}+(1+2ra_{q+1})u_{p,q+1}-ra_{q+1}u_{p+1,q+1}, \tag{10}$$
$$u_{p,0}=f(p\mu),\quad q=0, \tag{11}$$
$$u_{0,q}=\varphi(qv),\quad p=0, \tag{12}$$
$$u_{n,q}=\psi(qv),\quad n\mu=1, \tag{13}$$

where $a_{q+1}=a(qv+v)$, $r=\frac{v}{\mu^2}$, $x=p\mu$, $t=qv$, and $p=1,\ldots,n-1$. Problem (5)–(8) may be written in the following matrix form:

$$AU=b, \tag{14}$$

where

$$A=ra_{q+1}\begin{pmatrix}2&-1&&&&\\-1&2&-1&&&\\&&\ddots&\ddots&\ddots&\\&&&-1&2&-1\\&&&&-1&2\end{pmatrix}+\begin{pmatrix}1&&&&&\\&1&&&&\\&&\ddots&\ddots&\ddots&\\&&&&1&\\&&&&&1\end{pmatrix}, \tag{15}$$

$$U^{t}=(\,u_{1,q+1}\quad u_{2,q+1}\quad\ldots\quad u_{n-1,q+1}\,), \tag{16}$$

and

$$b^{t}=(\,u_{1,q}\quad\cdots\quad u_{n-1,q}\,)-(\,ra_{q+1}\varphi(qv+v)\quad\cdots\quad ra_{q+1}\psi(qv+v)\,). \tag{17}$$

It is necessary to remember that Eq. (14) is a linear system of algebraic equations.

Theorem 2. *The finite difference scheme (9) is unconditionally stable.*

Proof. To provide stability by the Fourier series method, Von Neumanns method, we assume the error function

$$E_{p,q}=e^{i\beta p\mu}\xi^{q}, \tag{18}$$

where $\xi=e^{\alpha v}$, and α, in general, is a complex constant. The error will not increase as t increases provided that

$$|\xi|\leqslant 1. \tag{19}$$

Substituting $E_{p,q}$ into Eq. (9) gives

$$e^{i\beta p\mu}\xi^{q+1}-e^{i\beta p\mu}\xi^{q}=ra_{q+1}(e^{i\beta(p-1)\mu}\xi^{q+1}-2e^{i\beta p\mu}\xi^{q+1}+e^{i\beta(p+1)\mu}\xi^{q+1}). \tag{20}$$

Divided by $e^{i\beta p\mu}\xi^{q}$ leads to

$$\xi-1=ra_{q+1}\xi(e^{-i\beta\mu}-2+e^{i\beta\mu})=-4ra_{q+1}\xi\sin^{2}(\beta\mu/2). \tag{21}$$

Hence

$$\xi = \frac{1}{1 + 4ra_{q+1}\sin^2(\beta\mu/2)}. \tag{22}$$

From Eq. (22) we conclude that for all positive values of r the condition (19) is true. Therefore, the finite difference scheme (9) is unconditionally stable. □

Here the exact solution of the partial differential equation (5) and the exact solution of the finite difference equation (9) are shown by U and u, respectively.

Definition 1. Let $F_{p,q+1}(u) = 0$ represents the difference equation at the $(p, q+1)$th mesh point. If u is replaced by U at the mesh points of the difference equation, then the value of $T_{p,q+1} = F_{p,q+1}(U)$ is called the local truncation error at the $(p, q+1)$th mesh point. If $T_{p,q+1}$ tends to zero as the mesh lengths μ and v tend to zero, the difference equation (9) is said to be consistent with the partial differential equation (5) [9].

Theorem 3. *The finite difference scheme (9) is consistent with the parabolic partial differential equation (5).*

Proof. By Taylorś expansion

$$U_{p,q+1} = U_{p,q} + v(U_t)_{p,q} + \tfrac{1}{2}v^2(U_{tt})_{p,q} + \tfrac{1}{6}v^3(U_{ttt})_{p,q} + \cdots, \tag{23}$$

$$U_{p+1,q+1} = U_{p,q+1} + \mu(U_x)_{p,q+1} + \tfrac{1}{2}\mu^2(U_{xx})_{p,q+1} + \tfrac{1}{6}\mu^3(U_{xxx})_{p,q+1} + \cdots, \tag{24}$$

$$U_{p-1,q+1} = U_{p,q+1} - \mu(U_x)_{p,q+1} + \tfrac{1}{2}\mu^2(U_{xx})_{p,q+1} - \tfrac{1}{6}\mu^3(U_{xxx})_{p,q+1} + \cdots. \tag{25}$$

Hence

$$T_{p,q+1} = (U_t - a(t)U_{xx})_{p,q+1} - \tfrac{1}{2}v(U_{tt})_{p,q+1} - \tfrac{1}{12}\mu^2(a(t)U_{xxxx})_{p,q+1} + \cdots. \tag{26}$$

But U is the solution of differential equation

$$(U_t - a(t)U_{xx})_{p,q+1} = 0, \tag{27}$$

therefore

$$T_{p,q+1} = \mathrm{O}(v) + \mathrm{O}(\mu^2). \tag{28}$$

Now as $\mu \to 0$ and $v \to 0$ then

$$T_{p,q+1} \to 0. \tag{29}$$

Consequently, the finite difference scheme (9) is consistent with the parabolic equation (5). □

Theorem 4 (Laxś equivalence theorem). *Given that we have a well-posed linear initial value problem and a linear finite difference approximation, and that the consistency condition exists, therefore stability is the necessary and sufficient condition for convergence* [10].

Theorem 5. *If U is the exact solution of the problem (5)–(8) and u is the exact solution of the finite difference equation (9), then u converges to U as μ tends to zero.*

Proof. From Theorems 2–4, it obviously follows that u converges to U as μ tends to zero. □

3.2. The solution of the linear system of algebraic equations

To solve the linear system (14), we consider the following iterative method:

$$u_i^{(k)} = (1-\gamma)u_i^{(k-1)} + \frac{\gamma}{a_{ii}}\left\{b_i - \sum_{j=1}^{i-1} a_{ij}u_j^{(k-1)} - \sum_{j=i+1}^{n-1} a_{ij}u_j^{(k-1)}\right\}, \tag{30}$$

where $i = 1, \ldots, n-1$ and $\gamma \in (0, 1]$. Which is called the Jacobi overrelaxation iterative method with relaxation parameter $\gamma \in (0, 1]$ in [1].

Eq. (30) may be written in the following matrix form:

$$U^{(k)} = LU^{(k-1)} + f, \quad k = 1, 2, \ldots, \tag{31}$$

where $U^k = (u_1^{(k)}, u_2^{(k)}, \ldots, u_{n-1}^{(k)})^T$, $L = I - DA$, $f = Db$ and

$$D = \operatorname{diag}\left(\frac{\gamma}{a_{11}}, \ldots, \frac{\gamma}{a_{n-1,n-1}}\right) \tag{32}$$

is a diagonal matrix. In fact, we convert the system (14) into an equivalent system of the following form:

$$U = LU + f. \tag{33}$$

Therefore, the sequence of approximate solution vectors of system (33) is generated by applying recursive equation (31). From (31) we obtain

$$U^{(k)} = f + Lf + \cdots + L^{k-1}f + L^k U^{(0)}, \quad k = 1, 2, \ldots. \tag{34}$$

If $U^{(0)} = 0$, then

$$U^{(k)} = (I + L + \cdots + L^{k-1})f = \sum_{m=0}^{k-1} L^m f, \quad k = 1, 2, \ldots. \tag{35}$$

Since the eigenvalues of matrix L are

$$\lambda_z = 1 - \gamma + \left(\frac{2r\gamma a_{q+1}}{1 + 2ra_{q+1}}\right) \cos\left(\frac{z\pi}{n}\right), \quad z = 1, \ldots, n-1, \tag{36}$$

then

$$\rho(L) < 1, \tag{37}$$

where $\rho(L)$ stands for the spectral radius of the matrix L.

It is evident that property (37) is a necessary and sufficient condition for convergence, i.e.

$$\lim_{k \to \infty} U^{(k)} = U. \tag{38}$$

In the next section we compute the iterations $U^{(k)}$ using Monte Carlo method where k is a finite number.

3.2.1. Monte Carlo method to solve linear system of algebraic equations

The application of the present Monte Carlo method to find a solution of linear system (33) is as follows: Consider the Markov chain

$$x_0 \to x_1 \to x_2 \to \cdots \to x_k \to \cdots, \tag{39}$$

with state space $\{1, 2, \ldots, n-1\}$ and transition matrix $P = \{p_{ij}\}$, $i, j = 1(1)n-1$. Let

$$P(x_0 = i) = p_i, \qquad P(x_{n-1} = j \mid x_{n-2} = i) = p_{ij}, \tag{40}$$

where p_i and p_{ij} are, respectively, the initial distribution and the transition probabilities of the Markov chain. The weight function W_m, for Markov chain (39) with $n-1$ states, is defined by using the recursion formula

$$W_0 = 1; \qquad W_m = W_{m-1}\frac{l_{x_{m-1}x_m}}{p_{x_{m-1}x_m}}, \quad m = 1, 2, \ldots. \tag{41}$$

Now the following random variable is defined:

$$\Gamma_k[H] = \frac{H_{x_0}}{p_{x_0}} \sum_{m=0}^{k} W_m b_{x_m}, \tag{42}$$

which is associated with the sample path

$$x_0 \to x_1 \to x_2 \to \cdots \to x_k, \tag{43}$$

where k is a given integer number and $H \in \mathbb{R}^{n-1}$ is a given vector. We also consider the problem of finding the inner product

$$\langle H, U\rangle = h_1u_1 + \cdots + h_{n-1}u_{n-1}, \tag{44}$$

where $H^{\mathrm{t}} = (h_1, \ldots, h_{n-1})$ is a given vector and $U^{\mathrm{t}} = (u_1, \ldots, u_{n-1})$ is a solution of (33). If we assume that

$$U^{(k)} = \begin{pmatrix} u_1^{(k)} \\ u_2^{(k)} \\ \cdot \\ \cdot \\ \cdot \\ u_{n-1}^{(k)} \end{pmatrix}, \tag{45}$$

and that it is the kth iterative solution of (31), then the following statement holds.

Theorem 6. *The mathematical expectation value of the random variable $\Gamma_k[H]$ is equal to the inner product $\langle H, U^{(k)}\rangle$, i.e.,*

$$E(\Gamma_k[H]) = \langle H, U^{(k)}\rangle \quad [11]. \tag{46}$$

To estimate

$$\langle H, U^{(k)}\rangle = h_1u_1^{(k)} + h_2u_2^{(k)} + \cdots + h_{n-1}u_{n-1}^{(k)}, \tag{47}$$

we simulate N random paths

$$x_0^{(s)} \to x_1^{(s)} \to x_2^{(s)} \to \cdots \to x_k^{(s)}, \quad s = 1(1)N, \tag{48}$$

each with the length of k, and evaluate the sample mean

$$\Theta_k[H] = \frac{1}{N}\sum_{s=1}^{N} \Gamma_k^{(s)}[H] \approx \langle H, U^{(k)}\rangle. \tag{49}$$

In fact, from Theorem 6 we conclude that $\Gamma_k[H]$ is an unbiased estimator of the inner product $\langle H, U^{(k)}\rangle$. It is readily seen that by setting

$$H^{\mathrm{t}} = (\underbrace{0, \ldots, 0, 1}_{j}, 0, \ldots, 0), \tag{50}$$

we obtain

$$\langle H, U^{(k)}\rangle = u_j^{(k)}, \quad j = 1, \ldots, n-1. \tag{51}$$

Hence $\Gamma_k[H]$ is an unbiased estimator of the $u_j^{(k)}$.

4. The comparative analysis of complexities of iterative and Monte Carlo methods to solve linear system

We show that Monte Carlo method asymptotically (by dimension of system) has a better order of complexity than that of the iterative method for linear system (33). In this section, the complexity of the algorithms is determined in terms of the number of basic arithmetic operations (addition, subtraction, multiplication and division) which are required to find a solution with a prescribed accuracy.

4.1. The complexity of the iterative method

Let us consider iterative method (31) with $U^{(0)} = 0$. Suppose that solving linear system (33) with desired accuracy ϵ requires at least $M(\epsilon, L, f) < \infty$ iterations. Then

$$U^{(M)} = (I + L + \cdots + L^{M-1})f = \sum_{m=0}^{M-1} L^m f \tag{52}$$

gives a solution of linear system (33) with accuracy ϵ. The number of arithmetic operations required for the calculation of the sum (52) is

$$T_{\mathrm{ITER}}(\epsilon, L) \geq 2M(\epsilon, L, f)(2N - 4). \tag{53}$$

4.2. The complexity of Monte Carlo method

We define the complexity of Monte Carlo method for solving system of linear algebraic equations (33) as the average number of arithmetic operations required to get a confidence interval of the desired length ϵ. The complexity of Monte Carlo method is denoted by $E[T(L, \epsilon)]$ [12]. An average relative efficiency of Monte Carlo method in comparison with the iterative method for linear system (33) is determined by

$$R(L, \epsilon) = \frac{E[T(L, \epsilon)]}{T_{\mathrm{ITER}}(\epsilon, L)}. \tag{54}$$

On the basis of (54) we conclude that when $R(L, \epsilon)$ becomes smaller Monte Carlo method is preferable to the iterative method.

Theorem 7. *Consider the linear system (33), with dimension $n - 1 \geq 2$, Then the asymptotic order of $R(L, \epsilon)$ is $\mathrm{O}\left(\frac{\log(n-1)}{\epsilon^2(n-1)}\right)$ as $n \to \infty$ [12].*

From Theorem 7 we conclude that if $\epsilon(n-1) = \omega\left(\sqrt{\frac{\log(n-1)}{(n-1)}}\right)$ [12], then $R(L, \epsilon) = \mathrm{o}(1)$ as $n \to \infty$, i.e.

$$\lim_{n\to\infty} \frac{R(L, \epsilon)}{1} = \lim_{n\to\infty} \frac{E[T(L, \epsilon)]}{T_{\mathrm{ITER}}(\epsilon, L)} = 0. \tag{55}$$

Hence we say that $T_{\mathrm{ITER}}(\epsilon, L)$ grows unboundedly faster than $E[T(L, \epsilon)]$. Therefore, it can be concluded that when the length of confidence interval $\epsilon(n-1) = \omega\left(\sqrt{\frac{\log(n-1)}{(n-1)}}\right)$, Monte Carlo method is preferred to the iterative method, as $n \to \infty$. Although for linear systems of small dimension Monte Carlo method yields a less accurate result in comparison with the iterative method, when dimension of system increases Monte Carlo method is more appropriate. There is a consensus among experts that Monte Carlo method is applicable for solving linear systems with relatively low accuracy.

5. Discussion of the numerical results

To give a clear overview of the numerical algorithm, the following examples are considered and the results are obtained.

Example 1. Consider (5)–(8) with

$$a(t) = 2t, \tag{56}$$
$$U(x, 0) = \sin(\pi x), \quad 0 < x < 1, \tag{57}$$
$$U(0, t) = 0, \quad t > 0, \tag{58}$$
$$U(1, t) = 0, \quad t > 0 \tag{59}$$

for which the exact solution is

$$U(x, t) = \exp(-\pi^2 t)\sin(\pi x). \tag{60}$$

The results obtained for $U(x, t)$ with $\gamma = 0.1011001$, $\mu = 0.005$, $v = 0.005$, $k = 5$ and $N = 1000$ are presented in Table 1.

Now the following example from the literature [6] is considered and the solution is obtained.

Table 1
Results for U with $\gamma = 0.1011001$, $\mu = 0.005$, $v = 0.005$, $k = 5$ and $N = 1000$

q	$u_{1,q}$		$u_{2,q}$		$u_{3,q}$	
	Numerical	Exact	Numerical	Exact	Numerical	Exact
1	0.098212	0.098200	0.138902	0.138900	0.098223	0.098200
2	0.013623	0.013600	0.019305	0.019300	0.013603	0.013600
3	0.001904	0.001900	0.002711	0.002700	0.001911	0.001900
4	0.000241	0.000263	0.000361	0.000372	0.000253	0.000263
5	0.000027	0.000036	0.000051	0.000051	0.000036	0.000036

Table 2
Results for U with $\gamma = 0.1011001$, $\mu = 0.01$, $v = 0.01$, $k = 4$ and $N = 34000$

q	$u_{1,q}$		$u_{5,q}$		$u_{9,q}$	
	Numerical	Exact	Numerical	Exact	Numerical	Exact
10	2.211323	2.211450	3.299283	3.299090	4.921614	4.921660
20	2.219317	2.219110	3.310467	3.310530	4.938711	4.938730
30	2.239387	2.239400	3.340768	3.340790	4.983859	4.983870
40	2.276758	2.276820	3.396595	3.396610	5.067143	5.067150
50	2.333134	2.333140	3.480623	3.480630	5.192508	5.192500

Example 2. Consider (5)–(8) with

$$a(t) = \frac{3t^2}{2 + 5t^3 + 3t^6}, \tag{61}$$

$$f(x) = 2\exp(x), \tag{62}$$

$$\varphi(t) = 1 + \frac{1 + 2t^3}{1 + t^3}, \tag{63}$$

$$\psi(t) = \exp(1) + \frac{\exp(1)(1 + 2t^3)}{1 + t^3} \tag{64}$$

for which the exact solution is

$$U(x,t) = \exp(x) + \frac{\exp(x)(1 + 2t^3)}{1 + t^3}. \tag{65}$$

The results obtained for $U(x,t)$ with $\gamma = 0.1011001$, $\mu = 0.01$, $v = 0.01$, $k = 4$ and $N = 34000$ are presented in Table 2.

The parameter γ is chosen such that it minimizes the norm of U in order to accelerate the convergence. In our algorithm we use the Euclidean norm and the maximum norm of vector $U \in \mathbb{R}^{n-1}$.

6. Conclusion

The present study successfully applied the numerical algorithm involving the finite difference method in conjunction with Monte Carlo method to solve a linear parabolic problem. From the numerical examples, it can be seen that the proposed numerical method is efficient and accurate to estimate the temperature distribution $U(x,t)$ in a one-dimensional linear parabolic partial differential equation. We also applied other different values of the mesh length μ and number of independent random paths N such as $\mu = 0.01$, $\mu = 0.001$, $N = 100$ and $N = 5000$. The results show that the present Monte Carlo method is also very efficient when the linear system is too large or too intricate for other treatment. Furthermore the results indicate that the present Monte Carlo method is preferable when one needs to have a rough estimation.

References

[1] I.T. Dimov, T.T. Dimov, T.V. Gurov, A new iterative Monte Carlo approach for inverse matrix problem, Journal of Computational and Applied Mathematics 92 (1998) 15–35.
[2] I.T. Dimov, O. Tonev, Monte Carlo algorithms: performance analysis for some computer architectures, Journal of Computational and Applied Mathematics 48 (1993) 253–277.
[3] J.H. Curtiss, Monte Carlo methods for the iteration of linear operators, Journal of Mathematical Physics 32 (4) (1954) 209–232.
[4] C.J.K. Tan, Solving systems of linear equations with relaxed Monte Carlo method, The Journal of Supercomputing 22 (2002) 113–123.
[5] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Athena Scientific, 1997.
[6] M. Dehghan, Identification of a time-dependent coefficient in a partial differential equation subject to an extra measurement, Numerical Methods for Partial Differential Equations 21 (3) (2005) 611–622.
[7] J.R. Cannon, The One-Dimensional Heat Equation, Addison-Wesley, Menlo Park, CA, 1984.
[8] A. Shidfar, R. Pourgholi, M. Ebrahimi, A numerical method for solving of a nonlinear inverse diffusion problem, Computers and Mathematics with Applications 52 (6–7) (2006) 1021–1030.
[9] G.D. Smith, Numerical Solution of Partial Differential Equations, second ed., Clarendon Press, Oxford, 1978.
[10] R.D. Richtmyer, K.W. Morton, Difference methods for initial-value problems, Interscience Publishers, 1967.
[11] R.Y. Rubinstein, Simulation and the Monte Carlo method, Wiley, New York, 1981.
[12] D.L. Danilov, S.M. Ermakov, J.H. Halton, Asymptotic complexity of Monte Carlo methods for solving linear systems, Journal of Statistical Planning and Inference 85 (2000) 5–18.