Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Mathematics and Computers in Simulation 63 (2003) 587–604

MATHEMATICS AND COMPUTERS IN SIMULATION

www.elsevier.com/locate/matcom

# Green's function Monte Carlo algorithms for elliptic problems

I.T. Dimov*, R.Y. Papancheva

*Central Laboratory for Parallel Processing, Department of Parallel Algorithms, Bulgarian Academy of Sciences, Acad. G. Bonchev St., bl. 25A, 1113 Sofia, Bulgaria*

Received 2 September 2002; received in revised form 2 July 2003; accepted 28 July 2003

**Abstract**

In many large-scale problems, one is interested to obtain directly an approximate value of a functional of the solution. Here, we consider a special class of grid-free Monte Carlo algorithms for direct computing of linear functionals of the solution of an elliptic boundary-value problem. Such kind of problems appear in environmental sciences, computational physics and financial mathematics. To create the algorithms, we use the Green's function analysis and define the conditions under which the integral transformation kernel of the integral representation for the boundary-value problem under consideration is non-negative. This analysis is done for a possible set of densities, and it is used to generate three different grid-free Monte Carlo algorithms based on different choices of the density of the radius of the balls used in Monte Carlo simulation. Only one of the generated algorithms was known before. We shall call it Sipin's algorithm. It was proposed and studied by Sipin. The aim of this work is to study the two new algorithms proposed here and based on two other (than in Sipin's algorithm) possible choices of the densities. The algorithms are described and analyzed. The performed numerical tests show that the efficiency of one of the new algorithms, which is based on a constant density is higher than the efficiency of Sipin's algorithm.


*Keywords:* Monte Carlo method; Selection methods; Density function

## 1. Introduction

In this paper, new Monte Carlo algorithms for computing functionals of solutions of elliptic boundary-value problems are proposed and studied. There are many large-scale problems in computational physics, biology and environmental sciences, where it is important to be able to compute functionals of the solution without computing the solution itself. It is that because some important parameters which can be measured in physics, and some effects on the live matter are in fact functionals. For instance, the damage effect of

* Corresponding author. Tel.: +359-2-9796793; fax: +359-2-707273.
*E-mail addresses:* ivdimov@bas.bg (I.T. Dimov), rumi@cantor.bas.bg (R.Y. Papancheva).
*URL:* http://www.bas.bg/dpa.

0378-4754/$30.00 © 2003 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/S0378-4754(03)00094-6

danger pollution is measured by:

$$\int_{\Omega} f(x)u(x)\,\mathrm{d}x,$$

where $x \equiv (x_1, \ldots, x_d) \in \Omega \subset R^d$, $u(x)$ is the concentration of the pollutant and $f(x)$ is given sensitivity function. Another example could be given from the statistical physics, where parameters characterized a multi-particle system like the mean-velocity $V_{\mathrm{m}}$ and the energy $E$ are defined by:

$$V_{\mathrm{m}} = \int_{\Omega} f(x)v(x)\,\mathrm{d}x,$$

$$E = \int_{\Omega} f(x)v^2(x)\,\mathrm{d}x,$$

where $f(x)(x \in \Omega \subset R^d)$ is the distribution function.

The algorithms proposed in this paper can be efficiently implemented in calculating functionals of the solution. Consider the functional $J(u)$:

$$J(u) \equiv (g, u) = \int_{\Omega} g(x)u(x)\,\mathrm{d}x, \tag{1}$$

where $\Omega \subset R^3$ and $x = (x_1, x_2, x_3) \in \Omega$ is a point in the Euclidean space $R^3$. The functions $u(x)$ and $g(x)$ belong to the Banach space $X$ and to the adjoint space $X^*$, respectively, and $u(x)$ is the unique solution of the following Fredholm integral equation in an operator form:

$$u = K(u) + f. \tag{2}$$

The algorithms under consideration use a local integral representation of the solution by Green's function. The Monte Carlo approach with the use of Green's function is well known (see [8]). The basic problem is the estimation of the computational complexity and the construction of Monte Carlo algorithms with lower computational cost.

We also consider problems, in which a linear functional of the solution is searched for and a rough solution is acceptable. For practical computations, it means that the relative error is about 5–10%. For example, such kind of problems are the pollution problems in environmental sciences. It is known that the most efficient computational method for direct computing of functionals is the Monte Carlo method (see [2,6]).

This paper deals with an analysis of the choice of the density function $p(r)$ in Lévy's function $L_p(y, x)$. We use the Green's function analysis to create the algorithms and define the conditions under which the integral transformation kernel of the integral representation for the considered boundary-value problem is non-negative. This analysis is done for a possible set of densities. The result is used to generate different grid-free Monte Carlo algorithms based on different choices of the density of the radius of the balls used in Monte Carlo simulation. Only one of the generated algorithms was known before. We call it Sipin's algorithms. It was proposed by Sipin and studied in [9]. Since in the Monte Carlo simulation a selection technique is used, it is important to obtain an estimation for the efficiency of the algorithm.

The selection algorithm in general case works as follows. Let us assume that we have to calculate a random variable $\xi$ with density function $f_1(x)/F_1$ and an easily computed function $f_2(x)$ exists, with

$0 \leq f_1(x) \leq f_2(x)$ for all $x \in \Omega$. Here,

$$F_1 = \int_\Omega f_1(x)\,\mathrm{d}x \quad \text{and} \quad F_2 = \int_\Omega f_2(x)\,\mathrm{d}x.$$

Let $\xi$ be a random variable with density function $f_2(x)/F_2$, and $\gamma$ be an uniformly distributed random variable in the interval [0, 1]. It has been shown [8] that if:

$$\gamma f_2(\xi) \leq f_1(\xi), \tag{3}$$

then $\xi$ is the needed random variable with density function $f_1(x)/F_1$. This algorithm allows us to use the function $f_2(x)$ instead of the original function $f_1(x)$. However, some variables will be rejected, namely when condition (3) will not be fulfilled.

We use the following definition for the *efficiency*.

**Definition 1.1.** The efficiency of the selection algorithm is defined as the ratio:

$$E = \frac{F_1}{F_2}.$$

In 1984, Sipin proved that the efficiency of the selection grid-free Monte Carlo algorithm is greater than 1/2 [9]. An improved result for the efficiency of the Sipin's algorithm was obtained by Dimov in 1988 [3]:

$$E = \frac{1+\alpha}{2+\alpha},$$

for some explicit constant $\alpha$.

The last result was improved again by Dimov and Gurov in 1998 [4]:

$$E = \frac{1+\alpha}{2+\alpha-\varepsilon_R},$$

for some $\varepsilon_R$, with $0 < \varepsilon_R < 1$.

It seems that this improvement is exact, i.e. it could not be improved again (but the last statement is still not proved). In all these works [3,4,9], the density function is fixed—$p(r) = e^{-kr}$. Here, we should answer the question if a different function to be taken, how this affects the algorithm efficiency.

The aim of this work is to study the two new algorithms proposed here and based on two other (than in Sipin's algorithm) possible choices of the densities. The algorithms are described and analyzed.

The performed numerical tests show that the efficiency of one of the new algorithms which is based on a constant density is higher than the efficiency of Sipin's algorithm.

The paper is organized as follows. In Section 2, we describe the formulation of the problem and give the conditions for existing an unique solution $u(x) \in C^2(\Omega) \cap C(\bar{\Omega})$. In Section 3, the set of possible densities providing the conditions, under which the integral transformation kernel is non-negative is defined. Two important cases are considered: a case of a constant density and a case of an exponential density. We find a set of parameters of the problem for which both constant and exponential densities can be applied. In Section 4, we describe and analyze the algorithms generated by the chosen densities. Numerical tests are presented in Section 5. The results of the numerical simulation illustrate the efficiency of the proposed algorithms. In Section 6, we give some concluding remarks.

## 2. Formulation of the problem

Let $X = L_1(\Omega)$. Then, $X^* = L_\infty(\Omega)$ ([8], p. 148). Consider the case when $K$ is an ordinary linear integral transform:

$$K(u)(x) = \int_\Omega k(x, y)u(y)\,\mathrm{d}y.$$

In this case, (2) obtains the form:

$$u(x) = \int_\Omega k(x, y)u(y)\,\mathrm{d}y + f(x).$$

The corresponding Neumann series converges when the condition:

$$\|K(u)\|_{\mathcal{L}(L_1)} = \sup_{y\in\Omega} \int_\Omega |k(x, y)|\,\mathrm{d}x \le q < 1,$$

holds.

We have to calculate the functional $J(u)$, where $u(x)$ is the solution of the following boundary-value problem:

$$M(u)(x) = -\Phi(x), \quad x \in \Omega, \quad \Omega \subset R^3, \tag{4}$$

$$u(x) = \psi(x), \quad x \in \partial\Omega, \tag{5}$$

where the operator $M$ is defined by:

$$M = \sum_{i=1}^{3} \left( \frac{\partial^2}{\partial x_i^2} + b_i(x)\frac{\partial}{\partial x_i} \right) + c(x).$$

Here follows the definition for the class $A^{(k,\lambda)}$ of domains $\Omega$.

**Definition 2.1.** The domain $\Omega$ belongs to the class $A^{(k,\lambda)}$ if for any point $x \in \partial\Omega$ the boundary $\partial\Omega$ can be represented as a function $z_3 = \sigma(z_1, z_2)$ in the neighborhood of $x$ with $\partial^k\sigma \in C^{(0,\lambda)}(\Omega)$, i.e.

$$|\partial^k\sigma(y) - \partial^k\sigma(y')| \le Q|y - y'|^\lambda,$$

where the vectors $y \equiv (z_1, z_2)$ and $y' \equiv (z_1', z_2')$ are two-dimensional vectors, $Q$ is a constant, and $\lambda \in (0, 1]$.

The definition of the class $C^{(0,\lambda)}(\Omega)$ can be found in [12]. If in the closed domain $\bar{\Omega} \in A^{(1,\lambda)}$ the coefficients of the operator $M$ satisfy the conditions: $b_i\,(i = 1, 2, 3)$, $c \in C^{(0,\lambda)}(\bar{\Omega})$, $c \le 0$ and $\Phi \in C^{(0,\lambda)}(\Omega) \cap C(\bar{\Omega})$, $\psi \in C(\partial\Omega)$, the problem (4) and (5) has an unique solution $u(x) \in C^2(\Omega) \cap C(\bar{\Omega})$. The conditions for uniqueness of the solution can be found in [1,11].

The first step in studying the Monte Carlo approach is obtaining an integral representation of the solution in the form:

$$u(x) = \int_{B(x)} k(x, y)u(y)\,\mathrm{d}y + f(x).$$

From an algorithmic point of view, the domain $B(x)$ must be chosen in such a way that the coordinates of the boundary points $y \in \partial B(x)$ could be easily calculated. So we denote by $B(x)$ the ball:

$$B(x) = B_R(x) = \{y : r = |x - y| \leq R(x)\},$$

where $R(x)$ is the radius of the ball.

We seek a representation of the integral kernel $k(x, y)$, using Lévy's function and the adjoint operator $M^*$ for the initial differential operator $M$.

Here follows a short definition for the Lévy's function [12].

**Definition 2.2.** Let,

$$H(x, y) = \frac{1}{4\pi|x - y|}.$$

Every function $L(x, y)$ continuous in variables $x$ and $y$, for $x$ and $y$ in $\Omega$, and $x \neq y$, together with its first and second derivatives with respect to the $x_i$ is called a Lévy's function if it satisfies bounds of the following type for some $\lambda > 0$ and $r = |x - y|$:

$$L - H = \mathrm{O}(r^{\lambda-1}), \qquad \frac{\partial[L - H]}{\partial x_i} = \mathrm{O}(r^{\lambda-2}), \qquad \frac{\partial^2[L - H]}{\partial x_i \partial x_j} = \mathrm{O}(r^{\lambda-3}),$$

such bounds holding uniformly in every bounded domain in $\Omega$.

The explicit form of the Lévy's function for the problem (4) and (5) can be found in [12]:

$$L_p(y, x) = \mu_p(R) \int_r^R \left(\frac{1}{r} - \frac{1}{\rho}\right) p(\rho)\, \mathrm{d}\rho, \quad r \leq R, \tag{6}$$

where the following notation is used: $p(r)$ is a density function:

$$\mu_p(R) = [4\pi q_p(R)]^{-1}, \qquad q_p(R) = \int_0^R p(\rho)\, \mathrm{d}\rho.$$

It is known [4] that if the vector-function $\boldsymbol{b}(x)$ satisfies the conditions $b_j(x) \in C(\Omega)$ $(j = 1, 2, 3)$, and $\operatorname{div} \boldsymbol{b}(x) = 0$, then the adjoint operator $M^*$, applied on functions $v(x) \in C^2(\Omega)$, such that:

$$\frac{\partial v(x)}{\partial x_i} = v(x) = 0, \quad \text{for any } x \in \partial\Omega, \quad i = 1, 2, 3,$$

has the following form:

$$M^* = \sum_{i=1}^{3} \left(\frac{\partial^2}{\partial x_i^2} - b_i(x)\frac{\partial}{\partial x_i}\right) + c(x).$$

According the results of Miranda [12], when the Lévy's function is used, the following integral representation for the solution $u(x)$ of the boundary-value problem is true:

$$u(x) = \int_{T(x)} (u(y)M_y^* L_p(y,x) + L_p(y,x)\Phi(y))\,\mathrm{d}y + \int_{\partial T(x)} \sum_{i=1}^{3} n_i \left[ \left( \frac{L_p(y,x)\partial u(y)}{\partial y_i} - \frac{u(y)\partial L_p(y,x)}{\partial y_i} \right) - b_i(y)u(y)L_p(y,x) \right] \mathrm{d}y, \tag{7}$$

where $n = (n_1, n_2, n_3)$ is the exterior normal to the boundary $\partial T(x)$.

Formula (7) holds for any domain $T(x) \in A^{(1,\lambda)}$ and therefore it is remaining true for every ball $B(x)$, lying inside the domain $\Omega$.

It is shown that it is possible to construct the function $L_p(y,x)$ in such a way that $M_y^* L_p(y,x)$ is non-negative in $B(x)$, and such that $L_p(y,x)$ and its derivatives vanish on $\partial B(x)$.

Finally, the integral representation of the solution has the following form:

$$u(x) = \int_{B(x)} M_y^* L_p(y,x)u(y)\,\mathrm{d}y + \int_{B(x)} L_p(y,x)\Phi(y)\,\mathrm{d}y, \tag{8}$$

where

$$M_y^* L_p(y,x) = \mu_p(R)\frac{p(r)}{r^2} - \mu_p(R)c(y)\int_r^R \frac{p(\rho)}{\rho}\,\mathrm{d}\rho + \frac{\mu_p(R)}{r^2}\left[ c(y)r + \sum_{i=1}^{3} b_i(y)\frac{y_i - x_i}{r} \right] \int_r^R p(\rho)\,\mathrm{d}\rho.$$

For solving our problem, we use a Monte Carlo procedure that is called "Walk on the Balls" (WOB). To ensure the convergence of the process, we introduce the $\varepsilon$-strip of the boundary. This process is similar to the well-known spherical process [7,10,13,14].

The process starts at point $\xi_0 = x \in \Omega$, which is chosen correspondingly with some initial density function $\pi(x)$. In the particular case, when $\pi(x) = \delta(x - x_0)$ the Markov's chain starts at the point $x_0 \in \Omega$. The next random point is determined with transition density function $p(x, y)$, which is proportional to the kernel of the integral equation (2) inside maximal ball $B(x) \subset \bar{\Omega}$. This process terminates when the point falls into the $\varepsilon$-strip of the boundary.

In spherical coordinates, $p(x, y)$ is equal to: $p(r, \boldsymbol{w}) = p(r)\bar{p}_r(\boldsymbol{w})$. We aim at analyzing the opportunities for a choice of density function $p(r)$. Our goal is to consider such density functions, in which the choice of bigger radii of the next ball in the Markov chain is more likely. One may expect that in this case the length of the chains will be shorter and the process more effective.

## 3. Green's function analysis

Here, we introduce the following notations:

$$b^* = \sup_{x\in\Omega} |\boldsymbol{b}(x)|, \qquad c^* = \sup_{x\in\Omega} |c(x)|, \qquad \boldsymbol{b}(x) = (b_1(x), b_2(x), b_3(x)),$$

and denote by $R$ the radius of the maximal ball lying inside $\bar{\Omega}$.

In [5] was proved that the conditions:

$$M_y^* L_p(y,x) \geq 0, \quad \text{for any } y \in B(x), \tag{9}$$

and

$$L_p(y,x) = \frac{\partial L_p(y,x)}{\partial y_i} = 0, \quad \text{for any } y \in \partial B(x), \quad i = 1,2,3, \tag{10}$$

are satisfied for:

$$p(r) = e^{-kr}, \tag{11}$$

where

$$k \geq b^* + Rc^*.$$

The density function defined in (11) corresponds to the Sipin's algorithm. Our aim is to consider the opportunities and the conditions for a different choice of the density function $p(r)$.

The following problem can be formulated: find the class of possible densities for which the integral transformation kernel $M_y^* L_p(y,x)$ is non-negative and the conditions (10) are satisfied.

The condition:

$$L_p(y,x) = 0, \quad \text{for any } y \in \partial B(x),$$

is obvious and follows directly from (6). Since,

$$\frac{\partial L_p(y,x)}{\partial y_i} = \frac{\partial L_p}{\partial r}\frac{\partial r}{\partial y_i} = \frac{x_i - y_i}{r^3}\mu_p \int_r^R p(\rho)\,\mathrm{d}\rho,$$

we obtain:

$$\frac{\partial L_p(y,x)}{\partial y_i} = 0, \quad \text{for any } y \in \partial B(x).$$

We shall prove the following theorem.

**Theorem 3.1.** *If the density function $p(r)$ satisfies the inequality*:

$$p(r) \geq \left(b^* + \frac{R}{4}c^*\right)\int_r^R p(\rho)\,\mathrm{d}\rho, \tag{12}$$

*then the integral transformation kernel $M_y^* L_p(y,x)$ is non-negative for any $y \in B(x)$.*

**Proof.** $M_y^* L_p(y,x)$ can be written as follows [4]:

$$M_y^* L_p(y,x) = \frac{\mu_p(R)}{r^2}\Gamma_p(y,x),$$

where

$$\Gamma_p(y,x) = p(r) + c(y)r\left(\int_r^R p(\rho)\,\mathrm{d}\rho - \int_r^R \frac{rp(\rho)}{\rho}\,\mathrm{d}\rho\right) + \sum_{i=1}^{3} b_i(y)\frac{y_i - x_i}{r}\int_r^R p(\rho)\,\mathrm{d}\rho.$$

Since $\rho \geq r \geq 0$, it follows that $1 - r/\rho \geq 0$, and therefore,

$$\int_r^R p(\rho)\,\mathrm{d}\rho - \int_r^R \frac{rp(\rho)}{\rho}\,\mathrm{d}\rho \geq 0. \tag{13}$$

For any $y$:

$$0 \geq c(y) \geq -c^*. \tag{14}$$

As $|x - y| = r$, we obtain that:

$$\sum_{i=1}^{3} b_i(y)\frac{y_i - x_i}{r} \geq -b^*. \tag{15}$$

Taking in account (13)–(15), we obtain:

$$\Gamma_p(y, x) \geq p(r) - c^* r\left(\int_r^R p(\rho)\,\mathrm{d}\rho - \int_r^R \frac{rp(\rho)}{\rho}\,\mathrm{d}\rho\right) - b^*\int_r^R p(\rho)\,\mathrm{d}\rho. \tag{16}$$

Since $0 \leq r \leq \rho \leq R$, it follows:

$$1 - \frac{r}{\rho} \leq 1 - \frac{r}{R},$$

and therefore,

$$-r\left(\int_r^R p(\rho)\,\mathrm{d}\rho - \int_r^R \frac{rp(\rho)}{\rho}\,\mathrm{d}\rho\right) \geq -r\left(1 - \frac{r}{R}\right)\int_r^R p(\rho)\,\mathrm{d}\rho. \tag{17}$$

For $0 \leq r \leq R$, one has:

$$r\left(1 - \frac{r}{R}\right) \leq \frac{R}{4}.$$

Finally, using the obtained extremum and the inequalities (16) and (17) we can write:

$$\Gamma_p(y, x) \geq p(r) - c^*\frac{R}{4}\int_r^R p(\rho)\,\mathrm{d}\rho - b^*\int_r^R p(\rho)\,\mathrm{d}\rho = p(r) - \left(b^* + c^*\frac{R}{4}\right)\int_r^R p(\rho)\,\mathrm{d}\rho.$$

Hence the result. □

The inequality (12) defines a class of possible densities $p(r)$, which can be used in a Monte Carlo simulation. We shall consider two possible densities: $p(r) = p = \text{const}$ for a given set of parameters of the problem (this case is described in Section 3.1) and $p(r) = e^{mr}$, $m > 0$ (described in Section 3.2).

### 3.1. Case A: a constant density

In this subsection, we describe the conditions which have to be fulfilled in order to use a constant density $p(r) = p = \text{const}$.

For this choice of the function $p(r)$ is true that:

$$p(r) - \left(b^* + \frac{R}{4}c^*\right)\int_r^R p(\rho)\,\mathrm{d}\rho \geq p\left[1 - \left(b^* + \frac{R}{4}c^*\right)R\right].$$

Consequently, if $(b^* + (R/4)c^*)R \leq 1$, then $M_y^* L_p(y, x) \geq 0$.

**Lemma 3.1.** *If the parameters of the problem satisfy the inequality*:

$$1 \geq (b^* + \tfrac{1}{4}Rc^*)R,$$

*and the density function* $p(r)$ *is chosen to be*:

$$p(r) = p = \text{const},$$

*then the condition* (9) *is true*.

**Remark.** If the radius of the maximal ball, lying inside the domain $\Omega$ satisfies the inequality:

$$R \leq \frac{2}{c^*}(\sqrt{b^{*2} + c^*} - b^*),$$

then the condition (9) is true.

### 3.2. Case B: an exponential density

While in Section 3.1, the radius $r$ is uniformly distributed in $[0, R]$, here, since the function $p(r) = e^{mr}$, $m > 0$ is strictly increasing, the choice of bigger radii will be more probable.

Let us introduce a parameter $k$, defined by the expression:

$$k = b^* + \tfrac{1}{4}Rc^*,$$

and consider the density function $p(r) = e^{kr}$.

From the inequality (12) one can obtain that:

$$k(R - r) \leq \ln 2.$$

Consequently if the condition $kR \leq \ln 2$ is fulfilled, then the inequality (12) is true for any $r$.

**Lemma 3.2.** *If the parameters of the problem satisfy the inequality*:

$$kR \leq \ln 2,$$

*and the density function* $p(r)$ *is chosen to be*:

$$p(r) = e^{kr}, \quad k = b^* + \tfrac{1}{4}Rc^*,$$

*then the condition* (9) *is fulfilled*.

## 4. Monte Carlo algorithms

Consider the problem of evaluating the functional:

$$J(u) \equiv (g, u) = \int_{\Omega} g(x)u(x)\,\mathrm{d}x,$$

where $u(x)$ is the solution of the integral equation (8).

The Monte Carlo algorithm for solving this problem can be defined as a 'ball process'. To ensure the convergence of the process, we introduce the $\varepsilon$-strip of the boundary, i.e.

$$\partial\Omega_{\varepsilon} = \{x \in \Omega : B(x) = B_{\varepsilon}(x)\},$$

where

$$B_{\varepsilon}(x) = \{y \in \Omega : |x - y| \leq \varepsilon\}.$$

The random variable, whose mathematical expectation coincides with $J(u)$ is:

$$\Theta[g] = \frac{g(\xi_0)}{\pi(\xi_0)} \sum_{j=0}^{\infty} Q_j f(\xi_i),$$

where

$$Q_0 = 1, \qquad Q_j = Q_{j-1}\frac{k(\xi_{j-1}, \xi_j)}{p(\xi_{j-1}, \xi_j)}, \quad j = 1, 2, 3, \ldots,$$

and $\xi_0, \xi_1, \ldots$ is a Markov chain in $\Omega$ with initial density function $\pi(x)$ and transition densities $p(x, y)$, which are tolerant to $g(x)$ and $k(x, y)$, respectively [2,8,15]. We use the following definition of the *tolerant density function*.

**Definition 4.1.** It is said that the density $p(x)$ is tolerant to the function $f(x)$ if $p(x) > 0$ in all points $x \in \Omega$, with $f(x) \neq 0$.

In [4] was proved that the integral transformation kernel in local integral representation can be used as a transition density function in the Markov process.

In our case:

$$p(x, y) = k(x, y) = \begin{cases} M_y^* L_p(y, x), & \text{when } x \in \Omega\backslash\partial\Omega, \\ 0, & \text{when } x \in \partial\Omega, \end{cases}$$

and

$$f(x) = \begin{cases} \int_{B(x)} L_p(y, x)\Phi(y)\,\mathrm{d}y, & \text{when } x \in \Omega\backslash\partial\Omega, \\ \psi(y), & \text{when } x \in \partial\Omega, \end{cases}$$

The function $p(x, y)$ can be expressed in spherical coordinates as [4]:

$$p(r, \boldsymbol{w}) = \frac{\sin\theta}{4\pi}\frac{p(r)}{q_p(R)}\bar{p}_r(\boldsymbol{w}),$$

where

$$\bar{p}_r(\boldsymbol{w}) = 1 + \left[\frac{|\boldsymbol{b}(x+r\boldsymbol{w})|\cos(\boldsymbol{b},\boldsymbol{w}) + c(x+r\boldsymbol{w})r}{p(r)}\right]\int_r^R p(\rho)\,\mathrm{d}\rho - \frac{c(x+r\boldsymbol{w})r^2}{p(r)}\int_r^R \frac{p(\rho)}{\rho}\,\mathrm{d}\rho.$$

For simulating random variable with density function $\bar{p}_r(\boldsymbol{w})$ we use the selection algorithm [4]. Since,

$$\bar{p}_r(\boldsymbol{w}) \le 1 + \frac{b^*}{p(r)}\int_r^R p(\rho)\,\mathrm{d}\rho = h(r),$$

the function $h(r)$ can be used as a majorant.

Here follows the algorithm for one random walk [4]:

1. *Calculate* the radius $R(x)$ of the maximal ball of center $x$ lying inside $\bar{\Omega}$.
2. *Compute* a realization $r$ of the random variable $\tau$ with the density:

$$\frac{p(r)}{q_p(R)}.$$

3. *Calculate* the function:

$$h(r) = 1 + \frac{b^*}{p(r)}\int_r^R p(\rho)\,\mathrm{d}\rho.$$

4. *Compute* independent realizations $\boldsymbol{w}_j$ of an unit isotropic vector in $\boldsymbol{R}^3$.
5. *Construct* independent realizations $\gamma_j$ of an uniformly distributed random variable in the interval $[0, 1]$.
6. *Repeat* steps 4 and 5 until find the parameter $j_0$ given by:

$$j_0 = \min\{j : h(r)\gamma_j \le \bar{p}_r(\boldsymbol{w}_j)\}.$$

The random vector $\boldsymbol{w}_{j_0}$ has the density $\bar{p}_r(\boldsymbol{w})$.

7. *Calculate* the random point $y$, using the following formula:

$$y = x + r\boldsymbol{w}_{j_0}.$$

The value $r = |y - x|$ is the radius of the ball lying inside $\Omega$ and having a center at $x$.

8. *Stop* the algorithm when the random process reaches the $\varepsilon$-strip—$\partial\Omega_\varepsilon$, i.e. $y \in \partial\Omega_\varepsilon$. If $y \notin \partial\Omega_\varepsilon$ then the algorithm has to be repeated for $x = y$.

### 4.1. Case A: $p(r) = p = const$

Here,

$$\frac{p(r)}{q_p(R)} = \frac{1}{R},$$

and consequently for sampling the jump $r$ in WOB process one can use the formula:

$$r = R\gamma,$$

where $\gamma$ is an uniformly distributed random variable in the interval $[0, 1]$.

For majorant function $h(r)$, we have:

$$h(r) = 1 + b^*(R - r),$$

and for $\bar{p}_r(\boldsymbol{w})$ we obtain:

$$\bar{p}_r(\boldsymbol{w}) = 1 + \left[\sum_{i=1}^{3} b_i(x + r\boldsymbol{w})w_i + c(x + r\boldsymbol{w})r\right](R - r) - c(x + r\boldsymbol{w})r^2 \ln\left(\frac{R}{r}\right).$$

*4.2. Case B:* $p(r) = e^{kr}$, $k = b^* + (R/4)c^*$

In this case:

$$\frac{p(r)}{q_p(R)} = \frac{ke^{kr}}{e^{kR} - 1},$$

and consequently,

$$r = \frac{1}{k}\ln(1 + (e^{kR} - 1)\gamma),$$

where $\gamma$ is an uniformly distributed random variable in the interval [0, 1].

For majorant function $h(r)$, we have:

$$h(r) = 1 + \frac{b^*}{k}(e^{k(R-r)} - 1)$$

and for $\bar{p}_r(\boldsymbol{w})$ we obtain:

$$\bar{p}_r(\boldsymbol{w}) = 1 + \left[\sum_{i=1}^{3} b_i(x + r\boldsymbol{w})w_i + c(x + r\boldsymbol{w})r\right]\frac{1}{k}(e^{k(R-r)} - 1) - \frac{c(x + r\boldsymbol{w})r^2}{e^{kr}}\int_r^R \frac{e^{k\rho}}{\rho}\,\mathrm{d}\rho.$$

## 5. Numerical tests

Numerical tests are performed in order to compare the proposed algorithms on a problem, which is similar to some real-life problems. The examples we consider here deal with the following boundary-value problem:

$$\sum_{i=1}^{3}\left(\frac{\partial^2 u(x)}{\partial x_i^2} + b_i(x)\frac{\partial u(x)}{\partial x_i}\right) + c(x)u(x) = 0 \quad \text{in } \Omega = [0, 1]^3,$$

$$u(x_1, x_2, x_3) = e^{a_1x_1 + a_2x_2 + a_3x_3}, \quad (x_1, x_2, x_3) \in \partial\Omega.$$

In our tests:

$$b_1(x) = a_2a_3(x_2 - x_3), \qquad b_2(x) = a_3a_1(x_3 - x_1), \qquad b_3(x) = a_1a_2(x_1 - x_2), \tag{18}$$

Table 1
CPU time in seconds for the case when $\varepsilon = 0.1$ and $a_i = 0.25\,(i = 1, 2, 3)$ for different values of the numbers of the Markov chains $N$

| $N$ | $2 \times 10^3$ | $5 \times 10^3$ | $10^4$ | $5 \times 10^4$ | $10^5$ | $2 \times 10^5$ | $5 \times 10^5$ |
|---|---|---|---|---|---|---|---|
| $p(r) = \text{const}$ | 0.05 | 0.12 | 0.26 | 1.28 | 2.55 | 5.11 | 12.76 |
| $p(r) = e^{kr}$ | 0.11 | 0.28 | 0.56 | 2.77 | 5.53 | 11.07 | 27.63 |
| $p(r) = e^{-kr}$ | 0.12 | 0.29 | 0.58 | 2.88 | 5.76 | 11.51 | 28.68 |

and

$$c(x) = -(a_1^2 + a_2^2 + a_3^2),$$

where $a_1, a_2, a_3$ are parameters of the problem.

We consider three cases for the coefficients:

- the first case, when $a_i = 0.25\,(i = 1, 2, 3)$;
- the second case, when $a_i = 0.5\,(i = 1, 2, 3)$;
- the third case, when $a_i = -1\,(i = 1, 2, 3)$.

Such kind of problems appear in environmental mathematics and describe the pollution transport due to advection and diffusion and take into account the deposition of pollution in areas free of emission sources (for more details, see [16]).

We also use different values to define the $\varepsilon$-strips:

- $\varepsilon = 0.01$;
- $\varepsilon = 0.05$;
- $\varepsilon = 0.1$.

The numerical experiments estimate the solution at the point:

$$x_0 = (0.5, 0.5, 0.5), \quad \text{i.e. } g(x) = \delta(x - x_0).$$

Some numerical results are presented in Table 1. We compare the computational time in cases when $p(r) = \text{const}$ (algorithm A), $p(r) = e^{kr}$ (algorithm B) and $p(r) = e^{-kr}$. The last case is used as a basic algorithm. One can see, that the algorithm A is much faster than the basic algorithm. It can be also seen, that the algorithm B is even a little bit faster than the basic one. The reason for such a result could be that the number of operations of the proposed algorithms is less, or the average length of the Markov chains is less.

Table 2 shows the average length of the Markov chain for all three algorithms. One can see, that the length of the Markov chains is practically the same (for the both of proposed algorithms the length of the Markov chains is a little bit less). It means that the reason for decreasing the computational time for the

Table 2
Average lengths of the Markov chains in case when $\varepsilon = 0.1$ and $a_i = 0.25\,(i = 1, 2, 3)$ for different values of $N$

| $N$ | $2 \times 10^3$ | $5 \times 10^3$ | $10^4$ | $5 \times 10^4$ | $10^5$ | $2 \times 10^5$ | $5 \times 10^5$ |
|---|---|---|---|---|---|---|---|
| $p(r) = \text{const}$ | 9.42 | 9.34 | 9.34 | 9.33 | 9.30 | 9.31 | 9.29 |
| $p(r) = e^{kr}$ | 9.18 | 9.15 | 9.17 | 9.16 | 9.15 | 9.15 | 9.14 |
| $p(r) = e^{-kr}$ | 9.60 | 9.57 | 9.51 | 9.50 | 9.49 | 9.48 | 9.45 |

Table 3
Average lengths of the Markov chains in case when $N = 200{,}000$

| $\varepsilon$ | $a_i$ $(i = 1, 2, 3)$ | $p(r) = \text{const}$ | $p(r) = e^{kr}$ | $p(r) = e^{-kr}$ |
|---|---|---|---|---|
| 0.1 | 0.25 | 9.304724 | 9.147489 | 9.462249 |
| 0.01 | 0.25 | 36.163536 | 35.889938 | 36.423023 |
| 0.1 | 0.5 | 9.302123 | 8.701044 | 9.963235 |

Table 4
Maximal lengths of the Markov chains in case when $N = 200{,}000$

| $\varepsilon$ | $a_i$ $(i = 1, 2, 3)$ | $p(r) = \text{const}$ | $p(r) = e^{kr}$ | $p(r) = e^{-kr}$ |
|---|---|---|---|---|
| 0.1 | 0.25 | 95 | 92 | 93 |
| 0.01 | 0.25 | 294 | 291 | 301 |

algorithm A is mostly the fact that this algorithm is less time consuming. Our numerical tests show that the length of the Markov chains (the number of Monte Carlo iterations) does not depend on the number $N$ of the Markov chains. Such a result can be expected, because the number of the Monte Carlo iterations depends on the radius of convergence of the iterative operator and the value of the $\varepsilon$-strip. As smaller is the radius of convergence of the iterative operator, smaller is the number of iterations (that is the length of the Markov chain for the algorithms under consideration). As smaller is the $\varepsilon$-strip of the domain, larger

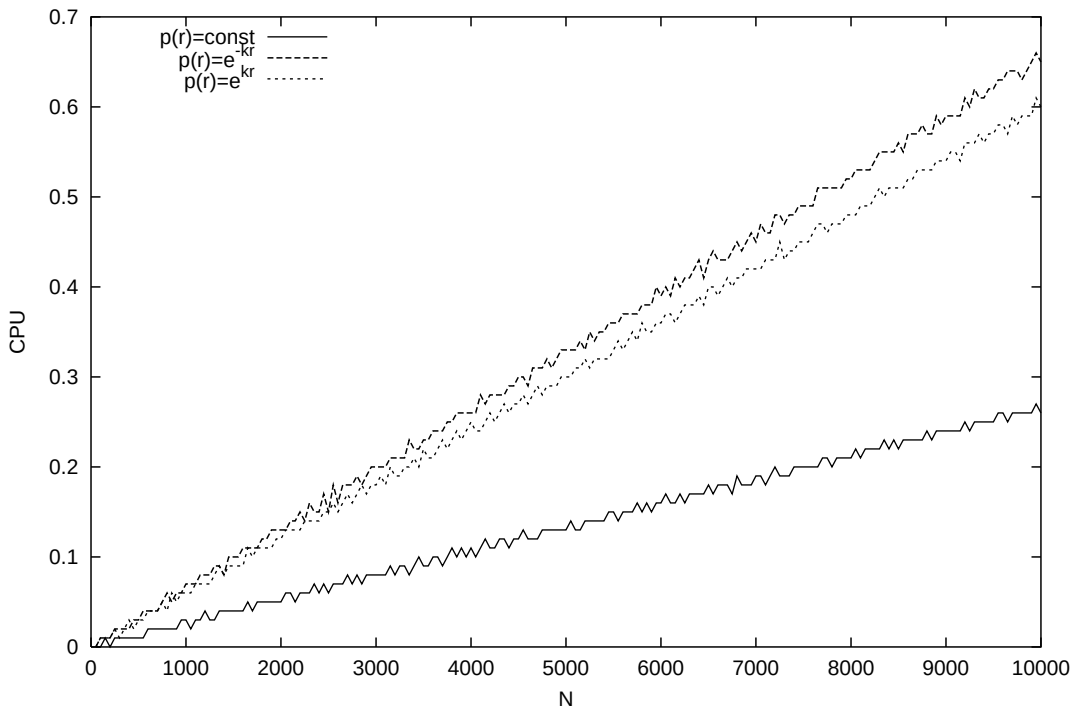


Fig. 1. Dependence of the CPU time from the number of random trajectories in case when $\varepsilon = 0.1$, $a = 0.25$.

is the number of iteration (the average length of the Markov chain). Some of our numerical tests presented in Table 3 illustrate these statements. The first line presents the results of the average length of the Markov chain (which corresponds to the average number of the Monte Carlo iterations) in the case when $\varepsilon = 0.1$ and $a_i = 0.25$ ($i = 1, 2, 3$). The length of the chain is averaged on a large number of Markov chains ($N =$ 200,000). Tests are performed for different initial value of the pseudo-random number generator. The average length of the Markov chains practically does not depend on the initial value of the pseudo-random number generator. Table 3 contains the same parameter obtained for the smaller $\varepsilon$-strip ($\varepsilon = 0.01$). In this case, one can observe a significant increasing of the average number of Monte Carlo iterations.

Some numerical tests were performed in order to study how the average length of the Markov chain depends of the parameters of the problem. In our tests we vary the value of the coefficients $a_i$ ($i = 1, 2, 3$). It corresponds to variation of the advection part of the problem, because these coefficients define the advection coefficients $b_i$ ($i = 1, 2, 3$) (see (18)). The numerical results show that such kind of changes in the operator (respectively, in the Monte Carlo iterative operator) does not change a lot the number of the Monte Carlo iterations. From the results presented in Table 3, one can see that the average length of the Markov chain does not change significantly when the values of $a_i$ are changed from 0.25 to 0.5.

For these algorithms it is also important to know the maximal length of the Markov chain. Table 4 presents the results of length of Markov chains with the maximum states, which corresponds to the maximal number of Monte Carlo iterations. The Markov chain with the maximal length is chosen among a large number of chains ($N = 200{,}000$). The results are obtained for $\varepsilon = 0.1$ and $a_i = 0.25$ ($i = 1, 2, 3$).

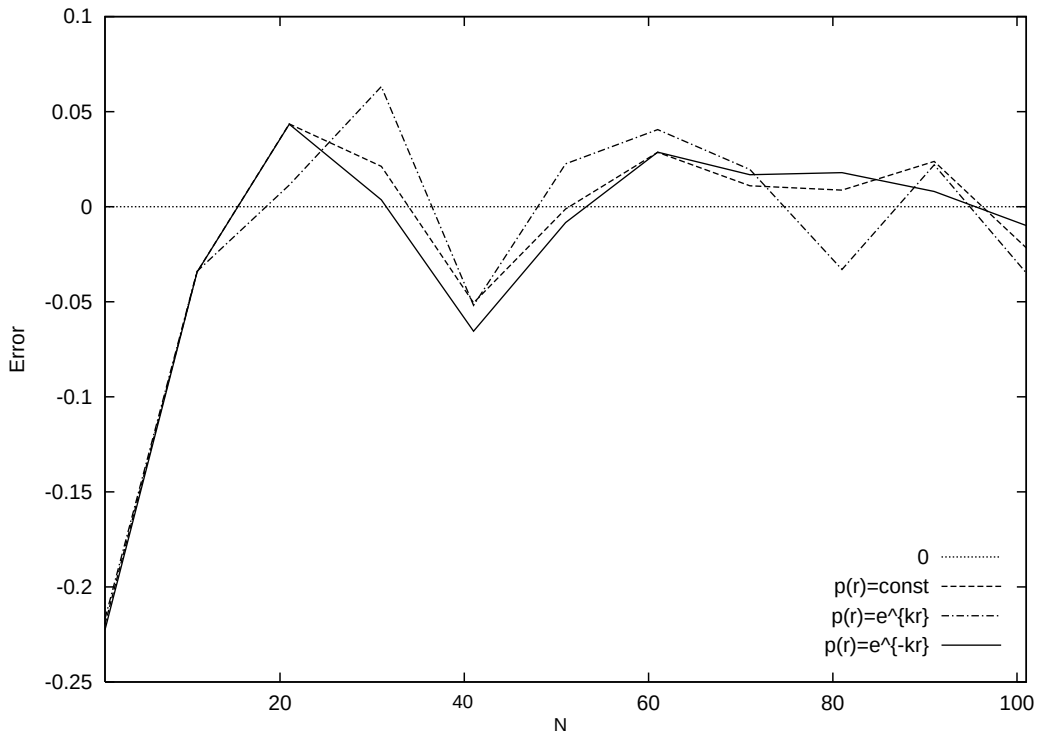


Fig. 2. Convergence of the algorithms for "small" values of $N$, $\varepsilon = 0.1$, $a = 0.25$, $u(x_0) = 1.454991$.

Similar results are obtained for $\varepsilon = 0.01$. One can see, that there is significant dependence of the maximal length of the Markov chain from the value of the $\varepsilon$-strip of the domain.

The dependence of the CPU time from the number of random trajectories (number of the Markov chains) is shown in Fig. 1. One can see, that the dependence is linear. It is because the computational work increases linearly with the increasing of the number of random trajectories $N$.

The difference between the exact and the approximate solutions is presented in Figs. 2 and 3 as a parameter, called "Error". The convergence of all three algorithms is practically the same. One can expect such a result, because the convergence depends on the radius of convergence of the iterative operator of the Monte Carlo method, which is the same for all three algorithms.

Some results of the convergence of the algorithms under consideration are presented in Fig. 3. One can see, that for values of $N$ larger than 1000 all algorithms converge. The speed of convergence is practically the same for all three algorithms. As the exact solution $u(x)$ is greater than 1 from Figs. 2 and 3, one can also see that the relative error is always less than 5%, which satisfies the requirements for the needed accuracy of the algorithms for solving this kind of problems. Under relative error, we assume the ratio among absolute error and the exact solution, i.e.

$$\text{Error} = \frac{|u(x_0) - \tilde{u}(x_0)|}{u(x_0)},$$

where $\tilde{u}(x_0)$ is the approximate value of the solution $u(x)$ at the point $x_0 \in \Omega$.

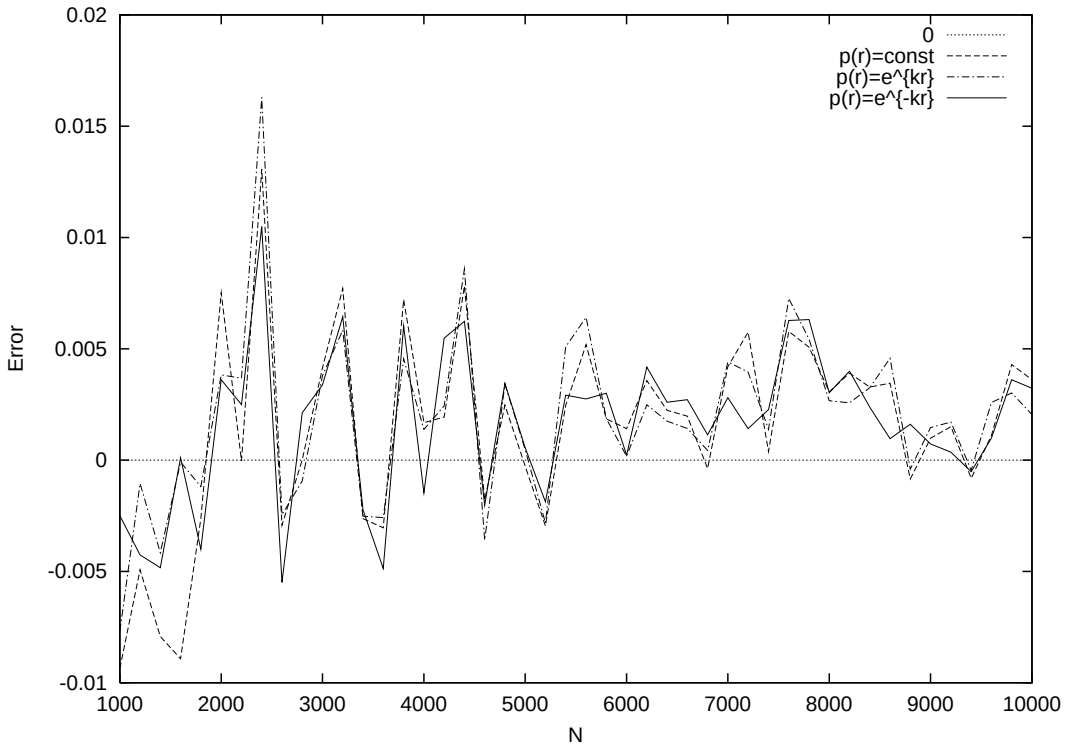


Fig. 3. Variation of the error for "large" values of $N$, $\varepsilon = 0.1$, $a = 0.25$, $u(x_0) = 1.454991$.

## 6. Concluding remarks

- We study two new grid-free Monte Carlo algorithms proposed in this work. The algorithms under consideration are based on two different choices of the density of the radius of the balls used in Monte Carlo simulations. The algorithms are described and analyzed.
- We obtain integral representations for constant and exponential densities and prove that the integral transformation kernels of the corresponding representations can be used as density functions of the grid-free Monte Carlo algorithms.
- The performed numerical tests show that:
  - The number of the Monte Carlo iterations (the length of the Markov chain) depends mainly of the value of the $\varepsilon$-strip of the domain.
  - The average length of the Markov chains practically does not depend of the number of chains. For the both of proposed algorithms this variation is smaller than the variation for the Sipin's algorithm considered as a basic algorithm.
  - The algorithms under consideration converge; the speed of convergence is practically the same as the speed of convergence of the Sipin's algorithm. One can also see, that the relative error is always less than 5%, which satisfies the requirements for the needed accuracy of the algorithms for solving this kind of problems.
  - The efficiency of one of the new algorithms which is based on a constant density is higher than the efficiency of Sipin's algorithm. The main reason for this is the fact that the algorithm based on the constant density has the lowest computational complexity.

One can conclude that algorithm A is preferable, because of its high algorithmic efficiency and the fact that the speed of convergence of this algorithm is practically the same as the speed of convergence of algorithm B or the Sipin's algorithm.

## Acknowledgements

Supported by BIS Center of Excellence (Contract No. ICA1-CT-2000-70016) and by the Ministry of Education and Science of Bulgaria under Grant #MM 902/99.

## References

[1] A.V. Bitzadze, Equations of the Mathematical Physics, Nauka, Moscow, 1982.
[2] J.H. Curtiss, Monte Carlo methods for the iteration of the linear operators, J. Math. Phys. 32 (4) (1954) 209–232.
[3] I.T. Dimov, Efficiency estimators the Monte Carlo algorithms, in: Bl. Sendov, R. Lazarov, I.T. Dimov (Eds.), Proceedings of the International Conference on Numerical Methods and Applications, Publishing House of the Bulgarian Academy of Sciences, Sofia, 1989, pp. 111–115.
[4] I.T. Dimov, T. Gurov, Estimates of the computational complexity of iterative Monte Carlo algorithm based on Green's function approach, Math. Comput. Simul. 47 (1998) 183–199.
[5] I.T. Dimov, A. Karaivanova, H. Kuchen, H. Stoltze, Monte Carlo algorithms for elliptic differential equations—a data parallel functional approach, J. Parallel Algorithms Appl. 9 (1996) 39–65.
[6] I.T. Dimov, O. Tonev, Monte Carlo algorithms: performance analysis for some computer architectures, J. Comput. Appl. Math. 48 (1993) 253–277.

[7] B.S. Elepov, A.A. Kronberg, G.A. Mikhailov, K.K. Sabelfeld, Solving the Boundary Value Problems by the Monte Carlo Method, Nauka, Novosibirsk, 1980.
[8] S.M. Ermakov, G.A. Mikhailov, Statistical Simulation, Nauka, Moscow, 1982.
[9] S.M. Ermakov, V.V. Nekrutkin, A.S. Sipin, Random Processes for Solving Classical Equations of the Mathematical Physics, Nauka, Moscow, 1984.
[10] G.A. Mikhailov, New Monte Carlo Methods with Estimating Derivatives, VSP BV, Utrecht, The Netherlands, 1995.
[11] V.P. Mikhailov, Partial Differential Equations, Nauka, Moscow, 1983.
[12] C. Miranda, Partial Differential Equations of Elliptic Type, Springer-Verlag, Berlin, 1970.
[13] M. Muller, Some continuous Monte Carlo methods for the Dirichlet problem, Ann. Math. Stat. 27 (3) (1956) 569–589.
[14] K.K. Sabelfeld, Monte Carlo Methods in Boundary Value Problems, Springer-Verlag, Berlin, 1991.
[15] I.M. Sobol, Monte Carlo Numerical Methods, Nauka, Moscow, 1973.
[16] Z. Zlatev, I.T. Dimov, K. Georgiev, Three-dimensional version of the Danish Eulerian model, Z. Angew. Math. Mech. 76 (S4) (1996) 473–476.