A Grid Free Monte Carlo Algorithm for Solving Elliptic Boundary Value Problems*

T. Gurov[1,2], P. Whitlock[2], and I. Dimov[1]

[1] Central Laboratory for Parallel Processing, Bulgarian Academy of Sciences
Acad. G. Bonchev St.,bl. 25 A, 1113 Sofia, Bulgaria
{dimov,gurov}@copern.bas.bg
[2] Dep. of Comp. and Inf. Sci., Brooklyn College – CUNY
2900 Bedford Avenue Brooklyn, NY 11210-2889
{whitlock,gurov}@sci.brooklyn.cuny.edu

Abstract. In this work a grid free Monte Carlo algorithm for solving elliptic boundary value problems is investigated. The proposed Monte Carlo approach leads to a random process called a ball process.
In order to generate random variables with the desired distribution, rejection techniques on two levels are used.
Varied numerical tests on a Sun Ultra Enterprise 4000 with 14 Ultra-SPARC processors were performed. The code which implemented the new algorithm was written in JAVA.
The numerical results show that the derived theoretical estimates can be used to predict the behavior of a wide class of elliptic boundary value problems.

1 Introduction

Consider the following three-dimensional elliptic boundary value problem:

$$Mu = -\phi(x),\ x \in \Omega,\ \Omega \subset \mathbb{R}^3 \quad \text{and} \quad u = \psi(x),\ x \in \partial\Omega, \tag{1}$$

where the differential operator M is equal to

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

We assume that the regularity conditions for the closed domain $\overline{\Omega}$ and the given functions $\mathbf{b}(x)$, $c(x) \leq 0$, $\phi(x)$ and $\psi(x)$ are satisfied. These conditions guarantee the existence and uniqueness of the solution $u(x)$ in $\mathbf{C}^2(\Omega) \cap \mathbf{C}(\overline{\Omega})$ of problem (1), (see [1,5]), as well as the possibility of its local integral representation (when $div\mathbf{b}(x) = \sum_{i=1}^{3} \frac{\partial b_i(x)}{\partial x_i^2} = 0$) by making use of the Green's function approach for standard domains lying inside the domain Ω (for example - a ball or an ellipsoid).

* Supported by ONR Grant N00014-96-1-1-1057 and by the National Science Fund of Bulgaria under Grant # I 811/1998.

L. Vulkov, J. Waśniewski, and P. Yalamov (Eds.): NAA 2000, LNCS 1988, pp. 359–367, 2001.

Denote by $B(x)$ the ball: $B(x) = B_R(x) = \{y : r =\mid y - x \mid\leq R(x)\}$, where $R(x)$ is the radius of the ball. Levy's function for the problem (1) is

$$L_p(y,x) = \mu_p(R) \int_r^R (1/r - 1/\rho)p(\rho)d\rho, \;\; r \leq R, \tag{2}$$

where the following notations are used: $p(\rho)$ is a density function;

$$r =\mid x - y \mid= \left(\sum_{i=1}^{3} (x_i - y_i)^2 \right)^{1/2}, \;\; \mu_p(R) = \left[4\pi q_p(R)\right]^{-1}, \;\; q_p(R) = \int_0^R p(\rho)d\rho.$$

It is readily seen that Levy's function $L_p(y,x)$, and the parameters $q_p(R)$ and $\mu_p(R)$ depend on the choice of the density function $p(\rho)$. In fact, the Eq.(2) defines a family of functions.

For the Levy's function the following representation holds (see [4]):

$$u(x) = \int_{B(x)} \left(u(y) M_y^* L_p(y,x) + L_p(y,x)\phi(y) \right) dy \tag{3}$$

$$+ \int_{\partial B(x)} \sum_{i=1}^{3} n_i \left[\left(\frac{L_p(y,x)\partial u(y)}{\partial y_i} - \frac{u(y)\partial L_p(y,x)}{\partial y_i} \right) - b_i(y)u(y)L_p(y,x) \right] d_y S \, ,$$

where $\mathbf{n} \equiv (n_1, n_2, n_3)$ is the exterior normal to the boundary $\partial B(x)$ and

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

is the adjoint operator to M.

It is proved (see [3]) that the conditions $M_y^* L_p(y,x) \geq 0$ (for any $y \in B(x)$) and $L_p(y,x) = \partial L_p(y,x)/\partial y_i = 0, \; i = 1,2,3$ (for any $y \in \partial B(x)$) are satisfied for $p(r) = e^{-kr}$, where

$$k \geq b^* + Rc^*, \quad b^* = \max_{x \in \Omega} \mid \mathbf{b}(x) \mid, \quad c^* = \max_{x \in \Omega} \mid c(x) \mid$$

and R is the radius of the maximal ball $B(x) \subset \overline{\Omega}$.

This statement shows that it is possible to construct the Levy's function choosing the density $p(\rho)$ such that kernel $M_y^* L_p(y,x)$ is non-negative in $B(x)$ and such that $L_p(y,x)$ and its derivatives vanish on $\partial B(x)$.

It follows that the representation (3) can be written in the form:

$$u(x) = \int_{B(x)} M_y^* L_p(y,x) u(y) dy + \int_{B(x)} L_p(y,x)\phi(y)dy, \tag{4}$$

where

$$M_y^* L_p(y,x) = \mu_p(R) \frac{p(r)}{r^2} - \mu_p(R) c(y) \int_r^R \frac{p(\rho)}{\rho} d\rho$$

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

The representation of $u(x)$ in (4) is the basis for the proposed Monte Carlo method. Using it, a biased estimator for the solution can be obtained.

2 Monte Carlo Method

The Monte Carlo procedure for solving Eq.(4) can be defined as a "ball process" or "walk on small spheres". Consider a transition density function

$$p(x, y) \geq 0 \text{ and } \int_{B(x)} p(x, y)dy = 1. \tag{5}$$

and define a Markov chain $\xi_0, \xi_1, \ldots,$ such that every point $\xi_j, j = 1, 2, \ldots,$ is chosen in the maximal ball $B(\xi_{j-1})$ lying in Ω in accordance with the density (5).

Generally, the "walk on small spheres" process can be written as following (see [8]):

$$\xi_j = \xi_{j-1} + \mathbf{w}^j \alpha R(\xi_{j-1}), \;\; j = 1, 2, \ldots, \;\; \alpha \in (0, 1],$$

where $\mathbf{w}^j$ are independent unit isotropic vectors in $\mathbb{R}^3$. In particular, when $\alpha = 1$ the process is called "walk on spheres" (see [6,8]).

To ensure the convergence of the process under consideration we introduce the ε-strip of the boundary, i.e.

$$\partial\Omega_\varepsilon = \{y \in \Omega : \exists x \in \partial\Omega \text{ for which } \mid y - x \mid \leq \varepsilon\}.$$

Thus the Markov chain terminates when it reaches $\partial\Omega_\varepsilon$ and the final point is $\xi_{l_\varepsilon} \in \partial\Omega_\varepsilon$.

Consider the biased estimate Θ_{l_ε} for the solution of Eq.(4) at the point ξ_0 (see [2]):

$$\Theta_{l_\varepsilon}(\xi_0) = \sum_{j=0}^{l_\varepsilon - 1} W_j \int_{B(\xi_j)} L_p(y, \xi_j)\phi(y)dy + W_{l_\varepsilon}\psi(\xi_{l_\varepsilon}), \tag{6}$$

where
$$W_0 = 1, \;\; W_j = W_{j-1}\frac{M_y^* L_p(\xi_j, \xi_{j-1})}{p(\xi_{j-1}, \xi_j)}, \;\; j = 1, 2, \ldots, l_\varepsilon.$$

If the first derivatives of the solution are bounded in Ω then the following inequality holds (see [6]):

$$|E\Theta_{l_\varepsilon}(\xi_0) - u(\xi_0)|^2 \leq c_1\varepsilon^2 \, . \tag{7}$$

Using N independent samples we construct a random estimate of the form

$$\overline{\Theta}_{l_\varepsilon}(\xi_0) = \frac{1}{N}\sum_{i=0}^{N} \Theta_{l_\varepsilon}^{(i)}(\xi_0) \approx u(\xi_0).$$

The root mean square deviation is defined by the relation

$$E(\Theta_{l_\varepsilon}(\xi_0) - u(\xi_0))^2 = Var(\Theta_{l_\varepsilon}(\xi_0)) + (u(\xi_0) - E\Theta_{l_\varepsilon}(\xi_0))^2.$$

Hence

$$E(\overline{\Theta}_{l_\varepsilon}(\xi_0) - u(\xi_0))^2 = \frac{Var(\Theta_{l_\varepsilon}(\xi_0))}{N} + (u(\xi_0) - E\Theta_{l_\varepsilon}(\xi_0))^2 \leq \frac{d_0}{N} + c_1\varepsilon^2 = \mu^2, \tag{8}$$

where μ is the desired error, d_0 is upper boundary of the variance and c_1 is the constant from Eq. (7).

3 A Grid Free Monte Carlo Algorithm

Using spherical coordinates [2] we can express the kernel $k(x,y) = M_y^* L_p(y,x)$ as follows:

$$k(r,\mathbf{w}) = \frac{p(r)\sin\theta}{q_p(R)4\pi}\times$$
$$\times\left[1 + \frac{\sum_{i=1}^{3} b_i(x+r\mathbf{w})w_i + c(x+r\mathbf{w})r}{p(r)}\int_r^R p(\rho)d\rho - \frac{c(x+r\mathbf{w})r^2}{p(r)}\int_r^R \frac{p(\rho)}{\rho}d\rho\right].$$

Here $\mathbf{w}\equiv(w_1,w_2,w_3)$ is an unit isotropic vector in $\mathbb{R}^3$, where $w_1 = \sin\theta\cos\varphi$, $w_2 = \sin\theta\sin\varphi$ and $w_3 = \cos\theta$ ($\theta\in[0,\,\pi)$ and $\varphi\in[0,\,2\pi)$).

Let us consider the following two non-negative functions

$$p_0(r,\mathbf{w}) = \frac{p(r)\sin\theta}{q_p(R)4\pi}\left[1 + \frac{\sum_{i=1}^{3} b_i(x+r\mathbf{w})w_i}{p(r)}\right]\int_r^R p(\rho)d\rho,$$

when $c(x+r\mathbf{w})\equiv 0$ and $p(r,\mathbf{w}) = k(r,\mathbf{w})$, when $c(x+r\mathbf{w})\leq 0$.

The following inequalities hold:

$$p(r,\mathbf{w}) \leq p_0(r,\mathbf{w}) \leq \frac{p(r)\sin\theta}{q_p(R)4\pi}\left[1 + \frac{b^*}{p(r)}\int_r^R p(\rho)d\rho\right]. \qquad (9)$$

We note that function $p_0(r,\mathbf{w})$ satisfies the condition (5) (see [2]).

Denote by $\overline{p}(r,\mathbf{w})$ the following function:

$$\overline{p}(r,\mathbf{w}) = \frac{p(r,\mathbf{w})}{V}, \text{ where } \int_0^\pi\int_0^{2\pi}\int_0^R p(r,\mathbf{w})drd\theta\varphi = V < 1. \qquad (10)$$

Introduce the functions:

$$\overline{p}(\mathbf{w}/r) = 1 + \frac{\sum_{i=1}^{3} b_i(x+r\mathbf{w})w_i + c(x+r\mathbf{w})r}{p(r)}\int_r^R p(\rho)d\rho - \frac{c(x+r\mathbf{w})r^2}{p(r)}\int_r^R \frac{p(\rho)}{\rho}d\rho;$$

$$\overline{p}_0(\mathbf{w}/r) = 1 + \frac{\sum_{i=1}^{3} b_i(x+r\mathbf{w})w_i}{p(r)}\int_r^R p(\rho)d\rho.$$

Using inequalities (9) we obtain:

$$\overline{p}(\mathbf{w}/r) \leq \overline{p}_0(\mathbf{w}/r) \leq 1 + \frac{b^*}{p(r)}\int_r^R p(\rho)d\rho. \qquad (11)$$

Now we can describe the grid free algorithm for simulating the Markov chain with transition density function (10). The Markov chain is started at the fixed point ξ_0. The inequalities in (11) are used to sample the next point ξ_1 by applying a two level acceptance-rejection sampling (ARS) rule.

The ARS rule or the Neumann rule can be used if another density function $v_2(x)$ exists such that $c_2v_2(x)$ is everywhere a maximum of the density function $v_1(x)$, that is, $c_2v_2(x)\geq v_1(x)$ for all values x (see for details [2]). The efficiency of this rule depends upon $c_2v_2(x)$ and how closely it envelopes $v_1(x)$. A two level ARS rule is preferable when $v_1(x)$ is a complex function. In this case a second majorant function must be found which envelopes very closely our density function.

Algorithm 3.1

1. **Compute** the radius $R(\xi_0)$ of the maximal ball lying inside Ω and having center ξ_0.
2. **Generate** a random value r of the random variable τ with the density

$$\frac{p(r)}{q_p(R)} = \frac{ke^{-kr}}{1 - e^{-kR}}. \tag{12}$$

3. **Calculate** the function

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

4. **Generate** the independent random values $\mathbf{w}$ of a unit isotropic vector in $\mathbb{R}^3$.
5. **Generate** the independent random value γ of an uniformly distributed random variable in the interval $[0, 1]$.
6. **Go to** the step 8 **if** the inequality holds: $\gamma h(r) \le \overline{p}_0(\mathbf{w}/r)\}$.
7. **Go to** the step 4 **otherwise**.
8. **Generate** the independent random value γ of a uniformly distributed random variable in the interval $[0, 1]$.
9. **Go to** the step 11 **if** the inequality holds: $\gamma \overline{p}_0(\mathbf{w}/r) \le \overline{p}(\mathbf{w}/r)$.
10. **Go to** the step 4 **otherwise**.
11. **Compute** the random point ξ_1, with a density $\overline{p}(\mathbf{w}/r)$ using the following formula: $\xi_1 = \xi_0 + r\mathbf{w}$.

The value $r = \mid \xi_1 - \xi_0 \mid$ is the radius of the sphere lying inside Ω and having center at ξ.

12. **Repeate** Algorithm 3.1 for new point ξ_1 **if** $\xi_1 \overline{\in} \partial\Omega_\varepsilon$.
13. **Stop** Algorithm 3.1 **if** $\xi_1 \in \partial\Omega_\varepsilon$.

The random variable $\Theta_{l_\varepsilon}(\xi_0)$ is calculated using formula (6).

The computational cost of the algorithm under consideration is measured by quantity

$$S = N\, t_0\, El_\varepsilon\,,$$

where N is the number of the trajectories performed; El_ε is the average number of balls on a single trajectory; t_0 is the time of modeling a point into the maximal ball lying inside Ω and of computing the weight W which corresponds to this point.

We note that for a wide class of boundaries Ω, (see [8,6]), the following estimate has been obtained on the basis of the restoration theory, $El_\varepsilon = c_2 |\ln \varepsilon|$. If the radius $r = r_0$ is fixed and $r_0/R = \alpha \in (0, 1]$ then the following estimate holds (see [8]):

$$El_\varepsilon = \frac{4R^2 |\ln \varepsilon|}{r_0^2} + O(r_0^4),$$

where R is the radius of the maximal ball lying inside Ω.

It is clear that the algorithmic efficiency of the Algorithm 3.1 depends on the position of the points in the Markov chain. They must be located "not far from the boundary of the ball". Thus, the location of every point depends on the random variable τ with a density Eq.(12).

The following assertion holds (see [2]):

Lemma 1. *Let* $\alpha_0 \in (0, 0.5)$. *Then* $E\tau \in (\alpha_0 R, 0.5R)$, *if and only if the radius* R *of the maximal ball and the parameters* b^* *and* c^* *satisfy the inequality*

$$R(b^* + Rc^*) \leq \beta_0, \tag{13}$$

where β_0 *is the solution of the equation* $g_1(z) = \frac{1}{z} + \frac{1}{1-e^z} = \alpha_0$.

Therefore, after substitution $r_0 = \alpha_0 R$, where α_0 is the parameter from Lemma 1, the average number of balls get

$$El_\varepsilon \asymp \frac{4}{\alpha_0^2} |\ln \varepsilon| \,. \tag{14}$$

In order to obtain the error of order μ, (see Eq. (8)), the optimal order of the quantities N and ε must be

$$N = O(\mu^{-2}), \quad \varepsilon = O(\mu), \quad S \asymp \frac{4t_0}{\alpha_0^2} \frac{|\ln \mu|}{\mu^2}.$$

Note that this estimate of computational cost is optimal as to the order of magnitude of μ only. It does not take into account the values of the constants in (8).

In order to minimize the computational cost we should solve the conditional minimum problem (see [6]):

$$S = N\, t_0\, El_\varepsilon \to \min_{N,\varepsilon}, \ \ \frac{d_0}{N} + c_1\varepsilon^2 = \mu^2 \ \text{ or } \ S = \frac{d_0\, t_0}{\mu^2 - c_1\varepsilon^2} El_\varepsilon \to \min_{\varepsilon}, \ \ c_1\varepsilon^2 \leq \mu^2.$$

Having solved this problem we obtain the optimal values of the quantities N, S and ε:

$$N^* = \frac{d_0}{2c_1\varepsilon_*^2 |\ln \varepsilon_*|}, \quad S^* \asymp \frac{2d_0}{c_1\alpha_0^2} \frac{t_0}{\varepsilon_*^2},$$

where ε_* is a solution of the equation

$$c_1\varepsilon^2 + 2c_1\varepsilon^2 |\ln \varepsilon| = \mu^2 \,. \tag{15}$$

It is not difficult to estimate the variance $Var(\Theta_{l_\varepsilon}(\xi_0))$ when the function $\phi(x) = 0$. In this case we have $\Theta_{l_\varepsilon}(\xi_0) = W_{l_\varepsilon}\psi(\xi_{l_\varepsilon})$. Thus

$$Var(\Theta_{l_\varepsilon}(\xi_0)) \leq E(\Theta^2_{l_\varepsilon}(\xi_0)) =$$

$$= \int_{B(\xi_0)} \ldots \int_{B(\xi_{l_\varepsilon})} \frac{(M_y^* L_p(\xi_1, \xi_0))^2}{\overline{p}(\xi_0, \xi_1)} \ldots \frac{(M_y^* L_p(\xi_{l_\varepsilon}, \xi_{l_\varepsilon - 1}))^2}{\overline{p}(\xi_{l_\varepsilon - 1}, \xi_{l_\varepsilon})} \psi(\xi_{l_\varepsilon})^2 d\xi_{l_\varepsilon} \ldots d\xi_0.$$

Denote by

$$V = \max_{x \in \Omega} \int_{B(x)} M_y^* L_p(y,x) dy \ \text{ and } \ \psi_*^2 = \max_{x \in \Omega} \int_{B(x)} |\psi(y)|^2 dy$$

Now we obtain

$$Var(\Theta_{l_\varepsilon}(\xi_0)) \leq V^{2l_\varepsilon} \psi_*^2 \leq \psi_*^2.$$

Thus in this case, the optimal values of the quantities N, S get:

$$N^* = \frac{\psi_*^2}{2c_1 \varepsilon_*^2 |\ln \varepsilon_*|}, \quad S^* \asymp \frac{2\psi_*^2}{c_1 \alpha_0^2} \frac{t_0}{\varepsilon_*^2},$$

where the constant c_1 depends on the condition the first derivatives of the solution shall be bounded in Ω and α_0 depends on Eq.(13).

4 Numerical Result

As an example the following boundary value problem was solved in the cube $\Omega = [0,1]^3$:

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

$$u(x_1,x_2,x_3) = e^{a_1(x_1+x_2+x_3)}, \ (x_1,x_2,x_3) \in \partial\Omega_\varepsilon.$$

In our tests $b_1(x) = a_2 x_1 (x_2 - x_3)$, $b_2(x) = a_2 x_2 (x_3 - x_1)$, $b_3(x) = a_2 x_3 (x_1 - x_2)$, and $c(x) = -3a_1^2$, where a_1 and a_2 are parameters.

We note that the condition $\operatorname{div} \mathbf{b}(x) = 0$ is satisfied.

The code which implemented the algorithm under consideration was written in JAVA. The multiplicative linear-congruential generator, which was used to obtain a sequence of random numbers distributed uniformly between 0 and 1, is $x_n = 7^5 x_{n-1} \mathrm{mod}(2^{31} - 1)$. It was highly recommended by Park and Miller [7] and they called it the "minimal standard".

Numerical tests on a Sun Ultra Enterprise 4000 with 14 UltraSPARC processors were performed for different values of the parameters a_1 and a_2 (see Tables 1,2). The solution was estimated at the point with coordinates $x = (0.5, 0.5, 0.5)$. In the tables, $u(x)$ is the exact solution, $u_{l_\varepsilon}(x)$ is the estimate of the solution, μ_ε is the estimate of the corresponding mean square error, σ^2 is the estimate $Var(\Theta_{l_\varepsilon}(x))$. The results presented in Table 1 are in good agreement with theoretical one (see Eq's. 14,15). Moreover, the results presented in Table 2 show how important it is to have a good balancing between the stochastic and systematic error. When $N^* = 50533$, the time of estimating solution is: $t_1 = 51m14.50s$ and when $N = 10^6$ the time is: $t_2 = 19h17m44.25s$. Thus, the computational effort in the first case is about twenty times better than second one, while Monte Carlo solutions are approximately equal. On the other hand the numerical tests show that the variance does not depend on the vector-function $b(x)$.

Table 1. $u(x) = 1.4549915$, $a_1 = 0.25$, $c^* = 0.1875$, $R_{max} = 0.5$

	$\varepsilon^* = 0.01,\quad N^* = 3032$			$\varepsilon^* = 0.001,\quad N^* = 202130$		
$b^* = a_2\sqrt{3}$	$u_{l_\varepsilon}(x)\quad \mu_\varepsilon$	σ^2	El_ε	$u_{l_\varepsilon}(x)\quad \mu_\varepsilon$	σ^2	El_ε
$8\sqrt{3}$	1.465218 ± 0.029	0.0437	63.73	1.456395 ± 0.0035	0.04455	95.75
$4\sqrt{3}$	1.460257 ± 0.029	0.0427	43.62	1.456704 ± 0.0035	0.04448	74.92
$2\sqrt{3}$	1.465602 ± 0.029	0.0434	38.85	1.457545 ± 0.0035	0.04466	69.46
$\sqrt{3}$	1.456592 ± 0.029	0.0423	36.96	1.456211 ± 0.0035	0.04462	67.95
$\sqrt{3}/4$	1.461289 ± 0.029	0.0432	36.46	1.456149 ± 0.0035	0.04455	67.32
$\sqrt{3}/16$	1.456079 ± 0.029	0.0428	36.72	1.455545 ± 0.0035	0.04450	67.12

Table 2. $u(x) = 2.117$, $a_1 = 0.5$, $c^* = 0.75$, $R_{max} = 0.5$

	$\varepsilon^* = 0.001,\quad N^* = 50533$			$\varepsilon = 0.001,\quad N = 1000000$		
$b^* = a_2\sqrt{3}$	$u_{l_\varepsilon}(x)\quad \mu_\varepsilon$	σ^2	El_ε	$u_{l_\varepsilon}(x)\quad \mu_\varepsilon$	σ^2	El_ε
$8\sqrt{3}$	2.12829 ± 0.015	0.374	97.41	2.12749 ± 0.015	0.3770	97.00
$4\sqrt{3}$	2.13151 ± 0.015	0.3782	75.93	2.12972 ± 0.015	0.3787	75.68
$2\sqrt{3}$	2.12878 ± 0.015	0.3775	70.17	2.12832 ± 0.015	0.3790	70.05
$\sqrt{3}$	2.12898 ± 0.015	0.3781	68.39	2.12547 ± 0.015	0.3774	67.95
$\sqrt{3}/4$	2.12227 ± 0.015	0.3771	67.71	2.12125 ± 0.015	0.3760	67.63
$\sqrt{3}/16$	2.12020 ± 0.015	0.3753	67.61	2.11869 ± 0.015	0.3750	67.53

5 Conclusion

In this work it is shown that a grid free Monte Carlo algorithm under consideration can be successfully applied for solving elliptic boundary value problems. An estimate for minimization of computational cost is obtained. The balancing of errors (both systematic and stochastic) either reduces the computational complexity when the desired error is fixed or increases the accuracy of the solution when the desired computational complexity is fixed.

The studied algorithm is easily programmable and parallelizable and can be efficiently implemented on MIMD-machines.

References

1. Bitzadze, A. V.: Equations of the Mathematical Physics. Nauka, Moscow, (1982). 359
2. Dimov, I., Gurov, T.: Estimates of the computational complexity of iterative Monte Carlo algorithm based on Green's function approach. Mathematics and Computers in Simulation. **47** (2-5) (1998) 183–199. 361, 362, 364
3. Ermakov, S. M., Nekrutkin V. V., Sipin, A. S.: Random processes for solving classical equations of the mathematical physics. Nauka, Moscow, (1984). 360
4. Miranda, C.: Equasioni alle dirivate parziali di tipo ellipttico. Springer-Verlag, Berlin, (1955). 360
5. Mikhailov, V. P.: Partial differential equations. Nauka, Moscow, (1983). 359

6. Mikhailov, G. A.: New Monte Carlo Methods with Estimating Derivatives. Utrecht, The Netherlands, (1995). 361, 363, 364
7. Park, S. K., Miller, K. W.: Random Number Generators: Good Ones Are Hard to Find, Communications of the ACM, **31** (10) (1988) 1192–1201. 365
8. Sabelfeld, K. K.: Monte Carlo Methods in Boundary Value Problems. Springer Verlag, Berlin - Heidelberg - New York - London, (1991). 361, 363