

# Complexity of Monte Carlo Algorithms for a Class of Integral Equations\*

Ivan Dimov<sup>1,2</sup> and Rayna Georgieva<sup>2</sup>

<sup>1</sup> Centre for Advanced Computing and Emerging Technologies  
School of Systems Engineering, The University of Reading  
Whiteknights, PO Box 225, Reading, RG6 6AY, UK  
i.t.dimov@reading.ac.uk

<sup>2</sup> Institute for Parallel Processing, Bulgarian Academy of Sciences  
Acad. G. Bonchev 25 A, 1113 Sofia, Bulgaria  
ivdimov@bas.bg, rayna@parallel.bas.bg

**Abstract.** In this work we study the computational complexity of a class of *grid* Monte Carlo algorithms for integral equations. The idea of the algorithms consists in an approximation of the integral equation by a system of algebraic equations. Then the Markov chain iterative Monte Carlo is used to solve the system. The assumption here is that the corresponding Neumann series for the iterative matrix does not necessarily converge or converges slowly. We use a special technique to accelerate the convergence. An estimate of the computational complexity of Monte Carlo algorithm using the considered approach is obtained. The estimate of the complexity is compared with the corresponding quantity for the complexity of the *grid-free* Monte Carlo algorithm. The conditions under which the class of *grid* Monte Carlo algorithms is more efficient are given.

## 1 Introduction

Monte Carlo method (MCM) is established as a powerful numerical approach for investigation of various problems (evaluation of integrals, solving integral equations, boundary value problems) with the progress in modern computational systems. In this paper, a special class of integral equations obtained from boundary value problems for elliptic partial differential equations is considered. Many problems in the area of environmental modeling, radiation transport, semiconductor modeling, and remote geological sensing are described in terms of integral equations that appear as integral representation of elliptic boundary value problems. Especially, the approach presented in this paper is of great importance for studying environmental security. There are different Monte Carlo algorithms (MCAs) for solving integral equations. A class of *grid* Monte Carlo algorithms (GMCAs) falls into the range of the present research. The question: *Which Monte Carlo algorithm is preferable to solve a given problem?* is of great importance in computational mathematics. That is why the purpose of this paper is to study

---

\* Partially supported by NATO grant "Monte Carlo Sensitivity Studies of Environmental Security" (PDD(TC)-(ESP.EAP.CLG 982641), BIS-21++ project funded by the European Commission (INCO-CT-2005-016639) as well as by the Ministry of Education and Science of Bulgaria, under grant I-1405/2004.

the conditions under which the class of algorithms under consideration solves a given problem more efficiently with the same accuracy than other MCAs or is the only applicable. Here we compare the efficiency of *grid* MCAs with known **grid-free Monte Carlo** algorithms (GFMCAs), called spherical process (see [5]). A measure of the efficiency of an algorithm is its complexity (computational cost), which is defined as the mean number of operations (arithmetic and logical) necessary for computing the value of the random variable for a transition in a Markov chain.

## 2 Formulation of the Problem

We consider a special class of Fredholm integral equations that normally appears as an integral representation of some boundary-value problems for differential equations. As an example which has many interesting applications we consider an elliptic boundary value problem:

$$\begin{cases} \mathcal{M}u = -\phi(\mathbf{x}), & \mathbf{x} \in \Omega \subset \mathbb{R}^d, \\ u = \omega(\mathbf{x}) & \mathbf{x} \in \partial\Omega, \end{cases} \quad (1)$$

where  $\mathcal{M} = \sum_{i=1}^d \left( \frac{\partial^2}{\partial x^{(i)2}} + v_i(\mathbf{x}) \frac{\partial}{\partial x^{(i)}} \right) + w(\mathbf{x}), \quad \mathbf{x} = (x^{(1)}, x^{(2)}, \dots, x^{(d)}).$

**Definition 1.** *The domain  $\Omega$  belongs to the class  $\mathbf{A}^{(n,\nu)}$  if it is possible to associate a hypersphere  $\Gamma(\mathbf{x})$  with each point  $\mathbf{x} \in \partial\Omega$ , so that the boundary  $\partial\Omega$  can be presented as a function  $z^{(d)} = \zeta(z^{(1)}, \dots, z^{(d-1)})$  in the neighborhood of  $\mathbf{x}$  for which  $\zeta^{(n)}(z^{(1)}, z^{(2)}, \dots, z^{(d-1)}) \in \mathbf{C}^{(0,\nu)}$ , i.e.  $|\zeta^{(n)}(\mathbf{z}_1) - \zeta^{(n)}(\mathbf{z}_2)| \leq \text{const} |\mathbf{z}_1 - \mathbf{z}_2|^\nu$ , where the vectors  $\mathbf{z}_1 = (z_1^{(1)}, z_1^{(2)}, \dots, z_1^{(d-1)})$  and  $\mathbf{z}_2 = (z_2^{(1)}, z_2^{(2)}, \dots, z_2^{(d-1)})$  are  $(d - 1)$ -dimensional vectors and  $\nu \in (0, 1]$ .*

If in the bounded domain  $\bar{\Omega} \in \mathbf{A}^{(1,\nu)}$  the coefficients of the operator  $\mathcal{M}$  satisfy the conditions  $v_j, w(\mathbf{x}) \in \mathbf{C}^{(0,\nu)}(\bar{\Omega}), w(\mathbf{x}) \leq 0$  and  $\phi \in \mathbf{C}^{(0,\nu)}(\Omega) \cap \mathbf{C}(\bar{\Omega}), \omega \in \mathbf{C}(\partial\Omega)$ , the problem (1) has a unique solution  $u(\mathbf{x}) \in \mathbf{C}^2(\Omega) \cap \mathbf{C}(\bar{\Omega})$ . The conditions for uniqueness of the solution can be found in [9].

An integral representation of the solution is obtained using the Green’s function for standard domains  $B(\mathbf{x}), \mathbf{x} \in \Omega$  (for example - sphere, ball, ellipsoid), lying inside the domain  $\Omega$  taking into account that  $B(\mathbf{x})$  satisfies required conditions (see [9]). Therefore, the initial problem for solving an elliptic differential task (1) is transformed into the following Fredholm integral equation of the second kind with a spectral parameter  $\lambda$  ( $\mathcal{K}$  is an integral operator,  $\mathcal{K} : L_p \mapsto L_p$ ):

$$u(\mathbf{x}) = \lambda \int_{B(\mathbf{x})} k(\mathbf{x}, \mathbf{t}) u(\mathbf{t}) \, d\mathbf{t} + f(\mathbf{x}), \quad \mathbf{x} \in \Omega \quad (\text{or} \quad u = \lambda \mathcal{K}u + f), \quad (2)$$

where  $k(\mathbf{x}, \mathbf{t})$  and  $f(\mathbf{x})$  are obtained using Levy’s function and satisfy:

$$k(\mathbf{x}, \mathbf{t}) \in L_p^{\mathbf{x}}(\Omega) \cap L_q^{\mathbf{t}}(B(\mathbf{x})), \quad f(\mathbf{x}) \in L_p(\Omega), \quad p, q \in \mathbb{Z}, p, q \geq 0, \quad \frac{1}{p} + \frac{1}{q} = 1.$$

The unknown function is denoted by  $u(\mathbf{x}) \in L_p(\Omega), \mathbf{x} \in \Omega, t \in B(\mathbf{x})$ .

We are interested in Monte Carlo method for evaluation with a *priori* given error  $\varepsilon$  of linear functionals of the solution of the integral equation (2) of the following type:

$$J(u) = \int_{\Omega} \varphi(\mathbf{x}) u(\mathbf{x}) \, d\mathbf{x} = (\varphi, u) \quad \text{for } \lambda = \lambda_*. \tag{3}$$

It is assumed that  $\varphi(\mathbf{x}) \in L_q(\Omega)$ ,  $q \geq 0$ ,  $q \in \mathbb{Z}$ .

### 3 A Class of Grid Monte Carlo Algorithms for Integral Equations

The investigated *grid* Monte Carlo approach for approximate evaluating of the linear functional (3) is based on the approximation of the given integral equation (2) by a system of linear algebraic equations (SLAE). This transformation represents the initial step of the considered class of *grid* MCAs. It is obtained using some approximate cubature rule (cubature method, Nystrom method, [1,7]). The next step is to apply the resolvent MCA [2,3] for solving linear systems of equations.

#### 3.1 Cubature Method

Let the set  $\{A_j\}_{j=1}^m$  be the weights and the points  $\{\mathbf{x}_j\}_{j=1}^m \in \Omega$  be the nodes of the chosen cubature formula. Thus, the initial problem for evaluating of  $(\varphi, u)$  is transformed into the problem for evaluating of the bilinear form  $(h, y)$  of the solution  $y$  of the obtained SLAE:

$$y = \lambda L y + b, \quad L = \{l_{ij}\} \in \mathbb{R}^{m \times m}, \quad y = \{y_i\}, \quad b = \{b_i\}, \quad h = \{h_i\} \in \mathbb{R}^{m \times 1} \tag{4}$$

with the vector  $h \in \mathbb{R}^{m \times 1}$ . The following notation is used:

$$l_{ij} = A_j k(\mathbf{x}_i, \mathbf{x}_j), \quad y_i = u(\mathbf{x}_i), \quad b_i = f(\mathbf{x}_i), \quad h_i = A_i \varphi(\mathbf{x}_i), \quad i, j = 1, \dots, m.$$

The error in the approximation on the first step is equal to:

$$\lambda \sum_{i=1}^m h_i \rho_1(\mathbf{x}_i; m, k, u) + \rho_2(m, \varphi, u),$$

where  $\rho_1(\mathbf{x}_j; m, k, u)$  and  $\rho_2(m, \varphi, u)$  are the approximation errors for the integral in equation (2) at the node  $\mathbf{x}_i$  and linear functional (3), respectively.

Some estimations for the obtained errors  $\rho_1, \rho_2$  from the approximation with some quadrature formula in the case when  $\Omega$  is an interval  $[a, b] \subset \mathbb{R}$  are given below. The errors depend on derivatives of some order of the functions  $k(x, t) u(t)$  and  $\varphi(x) u(x)$ . Estimations for these quantities obtained after differentiation of the integral equation and using Leibnitz's rule are given in the works of Kantorovich and Krylov [7]. Analogous estimations are given below:

$$\left| \frac{\partial^j}{\partial t^j} [k(x_i, t) u(t)] \right| \leq \sum_{l=0}^j \binom{j}{l} F^{(l)} K_t^{(j-l)} + |\lambda|(b-a) \sum_{l=0}^j \binom{j}{l} K_x^{(l)} K_t^{(j-l)} U^{(0)},$$

$$|(\varphi u)^{(j)}| \leq \sum_{l=0}^j \binom{j}{l} F^{(l)} \Phi^{(j-l)} + |\lambda|(b-a) \sum_{l=0}^j \binom{j}{l} K_x^{(l)} \Phi^{(j-l)} U^{(0)},$$

where 
$$K_x^{(j)} = \max_{t \in B(x)} \left| \frac{\partial^j k(x, t)}{\partial x^j} \right|_{x=x_i}, \quad K_t^{(j)} = \max_{t \in B(x)} \left| \frac{\partial^j k(x_i, t)}{\partial t^j} \right|,$$

$$F^{(j)} = \max_{x \in \Omega} |f^{(j)}(x)|, \quad U^{(j)} = \max_{x \in \Omega} |u^{(j)}(x)|, \quad \Phi^{(j)} = \max_{x \in \Omega} |\varphi^{(j)}(x)|.$$

The quantity  $U^{(0)}$ , which represents the maximum of the solution  $u$  in the interval  $\Omega = [a, b]$ , is unknown. We estimate it using the original integral equation, where the maximum of the solution in the right-hand side is estimated by the maximum of the initial approximation:  $U^{(0)} \leq (1 + |\lambda|(b - a)K^{(0)}) F^{(0)}$ .

### 3.2 Resolvent Monte Carlo Method for SLAE

Iterative Monte Carlo algorithm is used for evaluating a bilinear form  $(h, y)$  of the solution of the SLAE (4), obtained after the discretization of the given integral equation (2).

Consider the discrete Markov chain  $T : k_0 \rightarrow k_1 \rightarrow \dots \rightarrow k_i$  with  $m$  states  $1, 2, \dots, m$ . The chain is constructed according to initial probability  $\pi = \{\pi_i\}_{i=1}^m$  and transition probability  $P = \{p_{ij}\}_{i,j=1}^m$ . The mentioned probabilities have to be normalized and tolerant to the vector  $h$  and the matrix  $L$  respectively.

It is known (see, for example, [5,11]) that the mathematical expectation of the random variable, defined by the formula

$$\theta[h] = \frac{h_{k_0}}{\pi_{k_0}} \sum_{j=0}^{\infty} W_j b_{k_j}, \quad \text{where } W_0 = 1, \quad W_j = W_{j-1} \frac{l_{k_{j-1}k_j}}{p_{k_{j-1}k_j}},$$

is equal to the unknown bilinear form, i.e.  $\mathbf{E}\theta[h] = (h, y)$ .

Iterative MCM is characterized by two types of errors:

- *systematic* error  $r_i, i \geq 1$  (obtained from truncation of Markov chain) which depends on the number of iterations  $i$  of the used iterative process:

$$|r_i| \leq \alpha^{i+1} \|b\|_2 / (1 - \alpha), \quad \alpha = |\lambda| \|L\|_2, \quad b = \{b_j\}_{j=1}^m, \quad b_j = f(x_j)$$

- *statistical* error  $r_N$ , which depends on the number of samples  $N$  of Markov chain:  $r_N = c_\beta \sigma^2(\theta[h])N^{-1/2}, \quad 0 < \beta < 1, \beta \in \mathbb{R}$ .

The constant  $c_\beta$  (and therefore also the complexity estimates of algorithms) depends on the confidence level  $\beta$ . Probable error is often used, which corresponds to a 50% confidence level.

The problem to achieve a good balance between the systematic and statistical error has a great practical importance.

## 4 Estimate of the Computational Complexity

In this section, computational complexity of two approaches for solving integral equations is analysed. These approaches are related to iterative Monte Carlo methods and they have similar order of computational cost. That is why, our main goal is to compare the coefficients of leading terms in the expressions for complexity of algorithms under consideration. The values of these coefficients (depending on the number of operations necessary for every move in Markov chain) allow to determine the conditions when the considered *grid* MCA has higher computational efficiency than the mentioned *grid-free* MCA.

### 4.1 A Grid Monte Carlo Algorithm

To estimate the performance of MCAs, one has to consider the mathematical expectation  $\mathbf{E}T(\mathcal{A})$  of the time required for solving the problem using an algorithm  $\mathcal{A}$  (see [4]). Let  $l_A$  and  $l_L$  be the number of suboperations of the arithmetic and logical operations, respectively. The time required to complete a suboperation is denoted by  $\tau$  (for real computers this is usually the clock period).

**Cubature Algorithm.** The computational complexity is estimated for a given cubature rule:

$$T(\mathcal{CA}) > \tau \left[ c^s (p_k + 1) \varepsilon^{-s} + c^{-s/2} (p_f + p_\varphi + p_{node}) \varepsilon^{-s/2} + p_{coef} \right] l_A,$$

where the constant  $c$  depends on the following quantities

$$c = c(\lambda, K_x^{(r)}, K_t^{(r)}, F^{(r)}, \Phi^{(r)}), \quad r = 1, \dots, ADA + 1, \quad s = s(ADA) \quad (5)$$

( $ADA$  is the algebraic degree of accuracy of the chosen cubature formula). The number of arithmetic operations required to compute one value of the functions  $k(\mathbf{x}, \mathbf{t})$ ,  $f(\mathbf{x})$  and  $\varphi(\mathbf{x})$  and one node (coefficient) is denoted by  $p_k, p_f$  and  $p_\varphi$ , respectively and by  $p_{node}(p_{coef})$ .

The degree  $s$  and the constants  $p_{node}$  and  $p_{coef}$  depend on the applied formula. For instance:

$$s = \begin{cases} 1 & \text{for rectangular and Trapezoidal rule;} \\ 1/2 & \text{for Simpson's rule.} \end{cases} \quad (6)$$

**Resolvent Monte Carlo Algorithm.** Firstly, the case when the corresponding Neumann series converges (the supposition for slow convergence is allowed) is considered.

The following number of operations is necessary for one random walk:

- generation of one random number :  $k_A$  arithmetic and  $k_L$  logical operations;
- modeling the initial probability  $\pi$  to determine the initial **or** next point in the Markov chain:  $\mu_A$  arithmetic and  $\mu_L$  logical operations ( $\mathbf{E} \mu_A + 1 = \mathbf{E} \mu_L = \mu, 1 \leq \mu \leq m - 1$ );
- computing one value of the random variable: 4 arithmetic operations.

To calculate in advance the initial  $\pi$  and transition  $P$  probabilities (a vector and a square matrix, respectively), it is necessary a number of arithmetic operations, proportional to the matrix dimension  $m$ :  $2m(1 + m)$ .

To ensure a statistical error  $\varepsilon$ , it is necessary to perform  $i$  transitions in the Markov process, where  $i$  is chosen from the inequality

$$i > \ln^{-1} \alpha (\ln \varepsilon + \ln(1 - \alpha) - \ln \|b\|_2) - 1 \quad (\text{assuming } \|b\|_2 > \varepsilon(1 - \alpha)),$$

where  $\alpha = |\lambda| \|L\|_2$  and the initial approximation is chosen to be the right-hand side  $b$ .

To achieve a probable error  $\varepsilon$ , it is necessary to do  $N$  samples depending on the inequality  $N > c_{0.5} \sigma^2(\theta) \varepsilon^{-2}$ ,  $c_{0.5} \approx 0.6745$ , where  $\theta$  is the random variable, whose mathematical expectation coincides with the desired linear functional (3).

Therefore, the following estimate holds for the mathematical expectation of the time required to obtain an approximation with accuracy  $\varepsilon$  using the considered *grid* MCA:

$$\mathbf{E}T(\mathcal{RMCA}) > \tau [(k_A + \mu + 3)l_A + (k_L + \mu)l_L] \frac{[c_\beta \sigma(\xi_j^{\mathcal{R}}[h])]^2}{\ln^3 \alpha} \frac{(\ln^3 \varepsilon + a)}{\varepsilon^2} + 2\tau m(m + 1)l_A,$$

where  $a = \ln(1 - \alpha) - \ln \|b\|_2$ ,  $m > \sqrt{c^s} \varepsilon^{-s/2}$  (the constants are given by (5) and (6)), and  $\xi_j^{\mathcal{R}}$  is the unbiased estimate of the  $j$ -th iteration of the matrix  $L$ , obtained using the resolvent MCA.

Consider the case when the corresponding Neumann series does not converge. The convergence of the Monte Carlo algorithm for solving the SLAE (4) can be ensured (or accelerated) by application of an analytical continuation of the Neumann series by substituting of the spectral parameter  $\lambda$  (mapping) (see [2,6,7,8,10]). The main advantage of this approach for acceleration of convergence of an iterative process is its inessential influence over the computational complexity of the algorithm. The computational complexity on every walk is increased only with one arithmetic operation required for multiplication by the coefficients  $g_j, j \geq 0$ , that ensures convergence (on the supposition that these coefficients are calculated with a high precision in advance). To obtain the computational complexity of the modified algorithm, it is necessary to estimate the

variation of the new random variable:  $\theta[h] = \frac{h_{k_0}}{\pi_{k_0}} \sum_{j=0}^{\infty} \mathbf{g}_j W_j b_{k_j}.$

We will use the following statement for a class of mappings proved in [10]: *The conformal mapping  $\lambda = \psi(\eta) = a_1\eta + a_2\eta + \dots$  has only simple poles on its boundary of convergence  $|\eta| = 1$ . If  $Var \xi_k^{\mathcal{R}} \leq \sigma^2$  and  $q = \bar{a} |\eta_*| / (1 - |\eta_*|) < 1$ , then the complexity estimate of the algorithm has an order  $O(|\ln \varepsilon|^4 / \varepsilon^2)$ , where  $\bar{a}$  is such a constant that  $|a_i| \leq \bar{a}, i = 1, 2, \dots, \lambda_*$  is the value of the spectral parameter in the integral equation (2) (respectively SLAE (4)) and  $\eta_* = \psi^{-1}(\lambda_*)$ .*

In general, a computational estimate of this class of *grid* MCAs can be obtained if the behavior of  $g_j$  and  $Var \xi_j^{\mathcal{R}}$  is known.

#### 4.2 A Grid-Free Monte Carlo Algorithm

The computational complexity of the *grid* MCA under consideration is compared with the computational complexity of a *grid-free* Monte Carlo approach. This approach is based on the use of a local integral representation (assuming that such a representation exists, [9,12]) of the solution of an elliptic boundary value problem. Existence of this representation allows to construct a Monte Carlo algorithm, called spherical process (in the simplest case) for computing of the corresponding linear functional. As a first step of this algorithm an  $\varepsilon$ -strip  $\partial\Omega_\varepsilon$  of the boundary  $\partial\Omega$  is chosen (on the supposition that the solution is known on the boundary) to ensure the convergence of the constructed iterative process. The following number of operation is necessary for one random walk:

- generation of  $n$  (this number depends on initial probability  $\pi$ ) random numbers to determine the initial point in the Markov chain:  $n(k_A + k_L)$  operations ( $k_A$  and  $k_L$  are the arithmetic and logical operations necessary for the generation of one random

- number) **or** modeling of an isotropic vector that needs of the order of  $R*n(k_A+k_L)$  operations (the constant  $R$  depends on the efficiency of the modeling method and transition probability);
- calculating the coordinates of the initial **or** next point:  $p_{next}$  (depends on the modeling method and the dimension  $d$  of the domain  $B(x)$ );
  - calculating one value of functions:  $p_f$ ;  $p_\pi, p_\varphi$  **or**  $p_k, p_P$ ;
  - calculating one sample of the random variable: 4 arithmetic operations;
  - calculating the distance from the current point to the boundary  $\partial\Omega$ :  $\gamma_A$  arithmetic and  $\gamma_L$  logical operations (depends on the dimension  $d$  of the domain  $\Omega$ );
  - verification if the current point belongs to the chosen  $\delta$ -strip  $\partial\Omega_\delta$ .

The following logarithmic estimate for the average number  $\mathbf{E}i$  of spheres on a single trajectory holds for a wide class of boundaries [5]:

$$\mathbf{E}i \leq const |\ln \delta|, \quad const > 0, \quad (7)$$

where  $const$  depends on the boundary  $\partial\Omega$ .

Calculating the linear functional with a preliminary given accuracy  $\varepsilon$  and attainment of a good balance between the *statistical* and the *systematic* error is a problem of interest to us.

Let us to restrict our investigation of the statistical error to the domain  $\Omega \equiv [a, b]$ . To ensure a statistical error  $\varepsilon$ , it is necessary to do  $i$  transitions in the Markov process, where  $i$  is chosen from the inequality:

$$i > \ln^{-1} \alpha (\ln \varepsilon + \ln(1 - \alpha) - \ln F^{(0)}) - 1 \quad (\text{assuming } F^{(0)} > \varepsilon(1 - \alpha)),$$

where  $\alpha = |\lambda| V_{B(x)} K$ ,  $K = \max_{x,t} |k(x, t)|$  and the initial approximation is chosen to be the right-hand side  $f(x)$ . On the other hand, the estimate (7) depending on the chosen  $\varepsilon$ -strip of the boundary is done. Then, an expression for  $\delta$  according to the number of transition  $i$  is obtained from these two estimates:  $\delta \approx e^{-i/const}$ . Therefore, the following estimate holds for the mathematical expectation of the time required to obtain an approximation with accuracy  $\varepsilon$  using the considered *grid-free* MCA:

$$\begin{aligned} \mathbf{E}T(\mathcal{GFMCA}) &> \tau [(n k_A + p_{next} + p_f + p_\pi + p_\varphi + \gamma_A + 4) l_A \\ &+ (n k_L + \gamma_L + 1) l_L + ((R n k_A + p_{next} + p_f + p_k + p_P + 4 + \gamma_A) l_A \\ &+ (R n k_L + \gamma_L + 1) l_L) \times \frac{(\ln \varepsilon + \ln(1 - \alpha) - \ln^3 F^{(0)})}{\ln^3 \alpha} \frac{[c_\beta \sigma(\xi_j^S[h])]^2}{\varepsilon^2}. \end{aligned}$$

Obtained expressions for coefficients in computational complexity for MCAs under consideration allow us to define some conditions when the GMCA is preferable to the GFMCA:

- the functions that define the integral equation (2) ( $k(x, t), f(x), \varphi(x)$ ) have comparatively small maximum norm in the corresponding domain and their values can be calculated with a low complexity

- the initial and transition probability are complicated for modeling (acceptance-rejection method)
- large dimension of the integration domain

It has to be noted the fact that the *grid* MCAs under consideration are admissible only for integral equations with smooth functions, but some techniques of avoiding singularities of this kind exist (see [1]).

## 5 Concluding Discussion

In this paper we deal with performance analysis of a class of Markov chain *grid* Monte Carlo algorithms for solving Fredholm integral equations of second kind. We compare this class of algorithms with a class of *grid-free* algorithms. *Grid-free* Monte Carlo uses the so-called spherical process for computing of the corresponding linear functional. Obviously, the *grid* approach assumes higher regularity of the input data since it includes an approximation procedure described in Section 4.1. The (*grid-free*) approach does not need additional approximation procedure and directly produces a bias approximation to the solution. But the *grid-free* algorithm is more complicated and its implementation needs more routine operations (like checking the distance from a given point to the boundary) that decrease the efficiency of the algorithm. Analyzing the regularity of the problem one may chose either *grid* or *grid-free* algorithm is preferable. Especially, if the input data has higher regularity ( $k(x, t)$ ,  $f(x)$ ,  $\varphi(x)$  have comparatively small maximum norm) than the *grid* algorithm should be preferred.

## References

1. Bahvalov, N.S., Zhidkov, N.P., Kobelkov, G.M.: Numerical Methods. Nauka, Moscow (1987)
2. Dimov, I.T., Alexandrov, V.N., Karaivanova, A.N.: Parallel Resolvent Monte Carlo Algorithms for Linear Algebra Problems. Mathematics and Computers in Simulation **55** (2001) 25–35
3. Dimov, I.T., Karaivanova, A.N.: Iterative Monte Carlo Algorithms for Linear Algebra Problems. In: Vulkov, L., Wasniewski, J., Yalamov, P. (eds.): Lecture Notes in Computer Science, Vol. 1196. Springer-Verlag, Berlin (1996) 150–160
4. Dimov, I.T., Tonev, O.I.: Monte Carlo Algorithms: Performance Analysis for Some Computer Architectures. J. Comput. Appl. Math. **48** (1993) 253–277
5. Ermakov, S.M., Mikhailov, G.A.: Statistical Modeling. Nauka, Moscow (1982)
6. Kantorovich, L.V., Akilov, G.P.: Functional Analysis in Normed Spaces. Pergamon Press, Oxford (1964)
7. Kantorovich, L.V., Krylov, V.I.: Approximate Methods of Higher Analysis. Physical and Mathematical State Publishing House, Leningrad (1962)
8. Kublanovskaya, V.N.: Application of Analytical Continuation by Substitution of Variables in Numerical Analysis. In: Proceedings of the Steklov Institute of Mathematics (1959)
9. Miranda, C.: Partial Differential Equations of Elliptic Type. Springer-Verlag, Berlin Heidelberg New York (1970)
10. Sabelfeld, K.K.: Monte Carlo Methods in Boundary Value Problems. Springer-Verlag, Berlin Heidelberg New York London (1991)
11. Sobol', I.M.: The Monte Carlo Method. The University of Chicago Press, Chicago (1974)
12. Vladimirov, S.V.: Equations of Mathematical Physics. Nauka, Moscow (1976)