Monte Carlo study of particle transport problem in air pollution *

R.J. Papancheva [1], T.V. Gurov [1], I.T. Dimov [1]

Abstract

The actual transport of the air pollutants is due to the wind. This normally called "advection of the air pollutants". Diffusion and deposition are other two major physical processes, which take place during the transport of pollutants in the atmosphere.

In this paper we study two classes of grid-free Monte Carlo (MC) algorithms for solving an elliptic boundary value problem, where the partial differential equation contains advection, diffusion and deposition parts. The grid-free MC approach to solve the above equation uses a local integral representation and leads to a stochastic process called a random "Walk on balls" (WOB).

In the first class of algorithms, the choice of a transition density function in the Markov chain depends on the radius of the maximal ball, lying inside the domain, in which the problem is defined, and on the parameters of the differential operator. While the choice of a transition density function in the second class of algorithms does not depend on the deposition part of the problem.

The computational complexity of both classes of grid-free MC algorithms was investigated using varied numerical tests on a PowerPC (G4 w/AltiVec) 450 MHz running YDL 2.0.

*Supported by Center of Excellence BIS-21 grant ICA1-2000-70016 and by the NSF of Bulgaria under Grants #I 811/98 and #MM 902/99

[1]Central Laboratory for Parallel Processing, Department of Parallel algorithms, Bulgarian Academy of Sciences, Acad. G. Bonchev St., bl. 25A, 1113 Sofia, Bulgaria. Email: rumi@cantor.bas.bg, Web site: http://www.bas.bg/dpa

1 Formulation of the problem

Consider the functional

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

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

$$u(x) = \int_{\Omega} k(x, y)u(y)dy + f(x). \tag{2}$$

We are interested to estimate the functional (1), using MC approach, where Eq. (2) is a local integral representation of the solution of the following boundary value problem:

$$Mu = -\Phi(x), \qquad x \in \Omega, \quad \Omega \subset R^3, \tag{3}$$

$$u = \psi(x), \qquad x \in \partial\Omega. \tag{4}$$

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

Such problems appear in environmental sciences. The functional (1) can be considered as a measure for the damaging effect of a danger pollution on the life in the nature. In this case $g(x)$ is a given sensitivity function, and $u(x)$ is the concentration of the pollutant, that can be described as a solution of the boundary value problem (3)-(4).

Here, we introduce the following notations:

$$b^* = \max_{x \in \Omega} \mid \mathbf{b}(x) \mid, \qquad c^* = \max_{x \in \Omega} \mid c(x) \mid,$$

and denote by R the radius of the maximal ball, lying inside Ω and by $R(x)$ the radius of the maximal ball, lying inside Ω with center in the point x .

We assume that the conditions for existing of an unique solution of the problem (3)-(4) are satisfied [1, 11], and $div\mathbf{b}(x) = 0$. Using the Levy's function $L_p(y, x)$ [3] as a Green's function we obtain following integral form:

$$u(x) = \int_{B(x)} M_y^* L_p(y, x)u(y)dy + \int_{B(x)} L_p(y, x)\Phi(y)dy. \tag{5}$$

In the Eq.(5) $B(x)$ is the ball inside Ω with center in the point x and radius $R(x)$. $M^* = \sum_{i=1}^{3} \left(\frac{\partial^2}{\partial x_i^2} - b_i(x) \frac{\partial}{\partial x_i} \right) + c(x)$ is the adjoint operator to M, and

$$k(x,y) = M_y^* L_p(y,x) = \mu_p(R) \frac{p(r)}{r^2} - \mu_p(R) c(y) \int_r^R \frac{p(\rho)}{\rho} d\rho$$
$$+ \frac{\mu_p(R)}{r^2} \left[c(y) r + \sum_{i=1}^{3} b_i(y) \frac{y_i - x_i}{r} \right] \int_r^R p(\rho) d\rho. \quad (6)$$

The representation of $u(x)$ in (5) is the basis for the proposed MC approach. Using it a biased estimator for the solution can be obtained.

2 Monte Carlo algorithms

The MC estimator, whose mathematical expectation coincides with $J(u)$ is

$$\Theta[g] = \frac{g(\xi_0)}{\pi(\xi_0)} \sum_{j=0}^{\infty} Q_j f(\xi_j), \quad (7)$$

where $Q_0 = 1, Q_j = Q_{j-1} \frac{k(\xi_{j-1}, \xi_j)}{p(\xi_{j-1}, \xi_j)}, j = 1, 2, 3, ...$ and $\xi_0, \xi_1, ...$ is a Markov chain in Ω with initial density function $\pi(x)$ and transition density $p(x,y)$, which are tolerant to $g(x)$ and $k(x,y)$, respectively [2, 5, 11].

To ensure the convergence of the process, we introduce an ε - strip of the boundary. The process starts at point $\xi_0 = x \in \Omega$, chosen with the initial density function $\pi(x)$. The next random point y is determined by a transition density function $p(x,y)$. This process terminates when the point falls into the ε - strip of the boundary.

In the first class of algorithms (class A), the transition density is taken to be proportional to the kernel (6), i.e.

$$p(x,y) = \frac{k(x,y)}{\int_{B(x)} k(x,y) dy}. \quad (8)$$

In this case we need to know when the integral transformation kernel (6) is non-negative. It is proved in [4] that $p(x,y) \geq 0$, when the following inequality

$$p(r) \geq \left(b^* + \frac{R}{4}c^*\right) \int_r^R p(\rho)d\rho \tag{9}$$

is true.

In the second class (class B) the transition density function in the Markov chain is taken to be the kernel, where the deposition parameter is equal to zero, i.e.

$$p(x,y) = k(x,y) = M_y^* L_p(y,x) \quad \text{with} \quad c(y) \equiv 0. \tag{10}$$

It is proved in [10] that when the condition:

$$p(r) \geq b^* \int_r^R p(\rho)d\rho \tag{11}$$

is fulfilled, $p(x,y)$ can be used as transition density function in Markov chain.

Here we consider three cases for $p(r)$:

- **case I** $\quad p(r) = e^{-kr}$,
- **case II** $\quad p(r) = p = const, \qquad \text{if} \quad kR \leq 1$,
- **case III** $\quad p(r) = e^{kr}, \qquad \text{if} \quad kR \leq \ln 2$,

where

$$k = \begin{cases} b^* + \frac{R}{4}c^* & \text{for class A,} \\ \\ k = b^* & \text{for class B.} \end{cases}$$

The function $p(x,y)$ can be written in spherical coordinates as:

$$p(r,\mathbf{w}) = \frac{\sin\theta}{4\pi}\frac{p(r)}{q_p(R)}p(\mathbf{w}|r), \tag{12}$$

where for the class A we have:

$$\begin{aligned} p(\mathbf{w}|r) \quad &= \quad 1 + \left[\frac{|\mathbf{b}(x+r\mathbf{w})|\cos(\mathbf{b},\mathbf{w}) + c(x+r\mathbf{w})r}{p(r)}\right] \int_r^R p(\rho)d\rho \\ &\quad - \frac{c(x+r\mathbf{w})r^2}{p(r)} \int_r^R \frac{p(\rho)}{\rho}d\rho, \end{aligned} \tag{13}$$

and for algorithms from class B the conditional probability is:

$$p(\mathbf{w}|r) = 1 + \frac{|\mathbf{b}(x + r\mathbf{w})| \cos(\mathbf{b}, \mathbf{w})}{p(r)} \int\limits_r^R p(\rho) d\rho. \tag{14}$$

The next random point y in the Markov chain depends on both the jump r into the maximal ball, and on the direction $\mathbf{w}$. We sample the jump r with a density function $p(r)/q_p(R)$, using an inverse-transformation rule. The random direction $\mathbf{w}$ with density function $\widetilde{p}(\mathbf{w}|r) = \frac{\sin\theta}{4\pi} p(\mathbf{w}|r)$ is computed, using an acceptance-rejection (AR) technique [4, 10].

3 Numerical tests

As an example, the following boundary value problem was solved, using two classes of MC algorithms:

$$\sum_{i=1}^{3} \left(\frac{\partial^2 u}{\partial x_i^2} + b_i(x) \frac{\partial u}{\partial x_i} \right) + c(x) u = 0 \quad \text{in} \quad \Omega = [0, 1]^3, \tag{15}$$

$$u = \psi(x), \qquad\qquad x \in \partial\Omega_\varepsilon. \tag{16}$$

Two test variants are considered for $\psi(x)$, $\mathbf{b}(x)$ and $c(x)$:

Test 1:

$$\psi(x_1, x_2, x_3) = e^{a_1 x_1 + a_2 x_2 + a_3 x_3}, \qquad (x_1, x_2, x_3) \in \partial\Omega_\varepsilon,$$

$$b_1(x) = a_2 a_3 (x_2 - x_3), \quad b_2(x) = a_3 a_1 (x_3 - x_1), \quad b_3(x) = a_1 a_2 (x_1 - x_2),$$
$$c(x) = -(a_1^2 + a_2^2 + a_3^2);$$

Test 2:

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

$$b_1(x) = a_2 x_1 (x_2 - x_3), \quad b_2(x) = a_2 x_2 (x_3 - x_1), \quad b_3(x) = a_2 x_3 (x_1 - x_2),$$
$$c(x) = -3 a_1^2.$$

Here a_1, a_2, a_3 are parameters.

It is easy to see that $div \mathbf{b}(x) = 0$. This condition guarantees the possibility to use the local integral representation by Green's function.

Different values for the coefficients $a_i, i = 1, 2, 3$ are considered. In our tests the following ε-strips are used: $\varepsilon = 0.01, \quad \varepsilon = 0.05, \quad \varepsilon = 0.1$.

Figures 1 - 3 show the CPU times for algorithms of classes A and B, with different values for coefficients a_i. One can see that the algorithm B is faster than the algorithm A, with increasing of b^*. This behavior can be explained with the less number of the moves in the Markov chains (see Table 1). When the maximum of the advection vector $\mathbf{b}(x)$ increases the average length of Markov chain $(El_{(\varepsilon,B)})$ for the algorithm B is shorter than $El_{(\varepsilon,A)}$ for algorithm A. Figure 3 shows that the algorithm B is about 1.75 times faster than the algorithm A in case I when $a_i = -3, \quad b^* = 15.588457$.

The relative errors of the algorithms are practically the same for both classes A and B, and they depend on b^* (see Figures 4 - 7).

Figure 8 compares CPU times of the MC algorithms from the class B with a different choice of the density function $p(r)$. One can see that the algorithm in case II is faster than the others. The less CPU time of this case is due to the computational simplicity. On the other hand the algorithm in case III is a little bit faster than the algorithm in a case I. This can be explained with results for the average moves in the WOB that are presented in Table 2.

We conclude that the numerical results show that the algorithms of class B are more efficient. Also we recommend to use a case II or a case III, if the parameters of the problem allow this.

a_i	$El_{(\varepsilon,A)}$	$El_{(\varepsilon,B)}$	b^*
0.25	36.2899	36.2610	0.1083
-1	39.2195	38.6290	1.7321
-2	53.0822	49.2457	6.9282
-3	97.5206	79.9681	15.5885
-4	218.8564	159.5063	27.7128
	$p(r) = e^{-kr}$	$p(r) = e^{-b^* r}$	

Table 1: The average lengths of Markov chains with $\varepsilon = 0.01$.

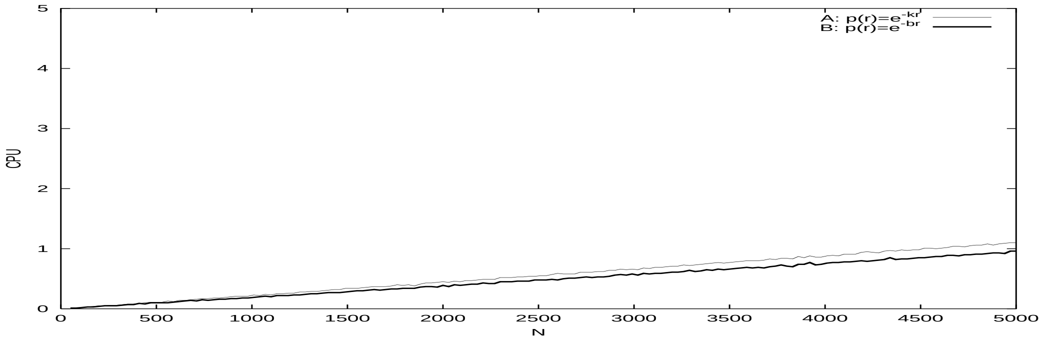

Figure 1: Comparison of the CPU times among algorithms A and B in case when $\varepsilon = 0.01$, $a = 0.25$ for Test 1.

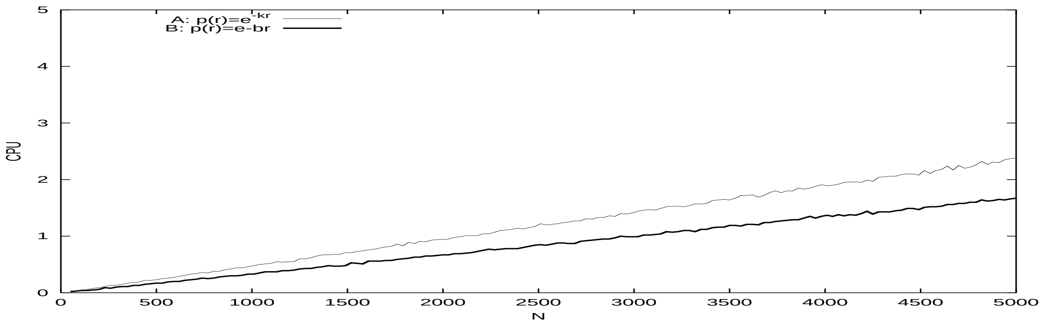

Figure 2: Comparison of the CPU times among algorithms A and B in case when $\varepsilon = 0.01$, $a = -2$ for Test 1.

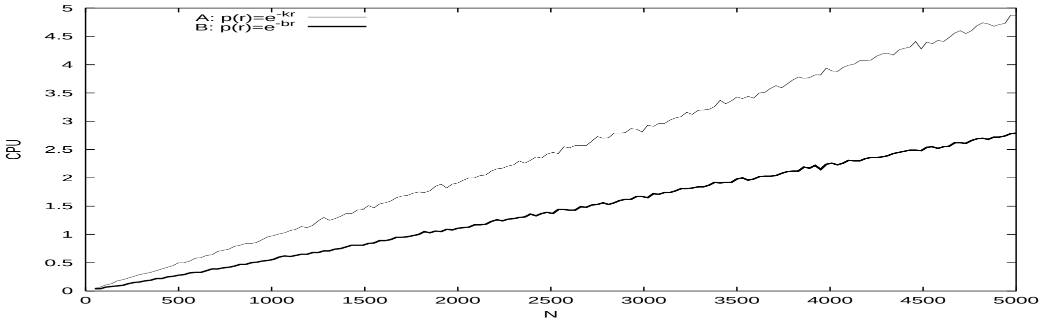

Figure 3: Comparison of the CPU times among algorithms A and B in case when $\varepsilon = 0.01$, $a = -3$ for Test 1.

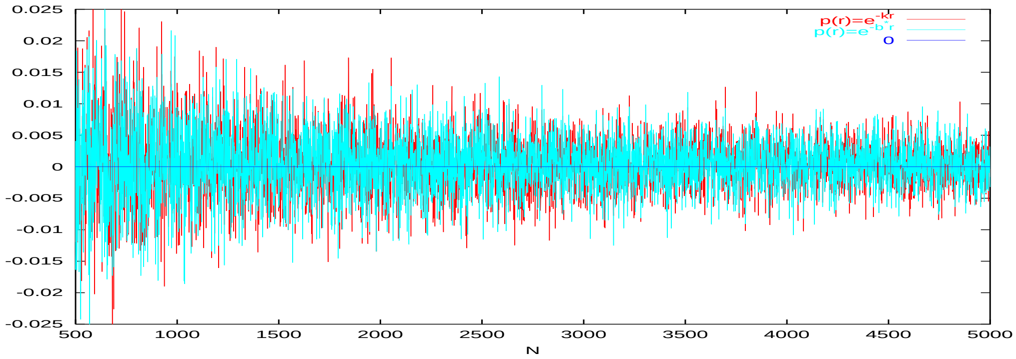

Figure 4: Variation of the relative error in case, when $\varepsilon = 0.01$, $a = 0.25$.

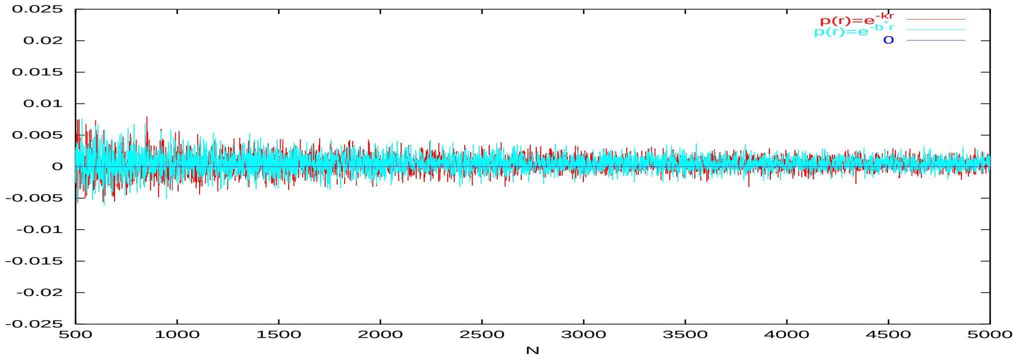

Figure 5: Variation of the relative error in case, when $\varepsilon = 0.01$, $a = -2$.

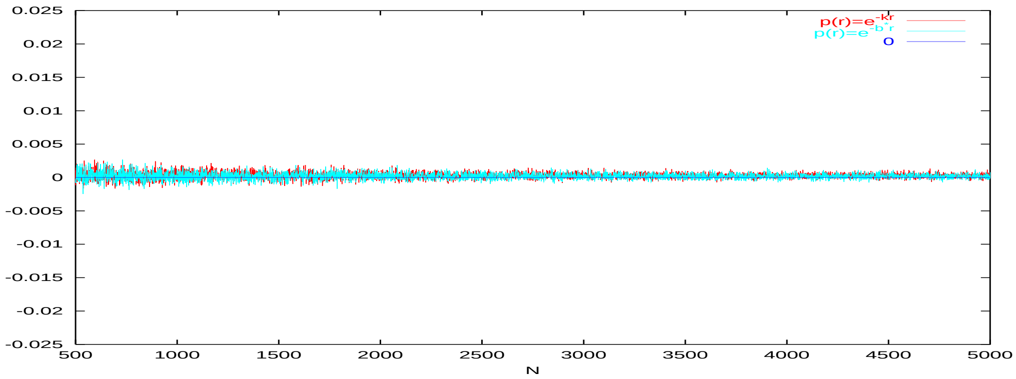

Figure 6: Variation of the relative error in case, when $\varepsilon = 0.01$, $a = -3$.

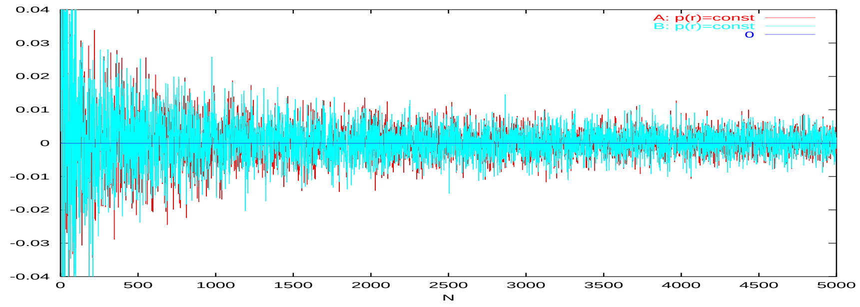

Figure 7: Variation of the relative error in case, when $\varepsilon = 0.01$, $a = 0.25$ for Test 2.

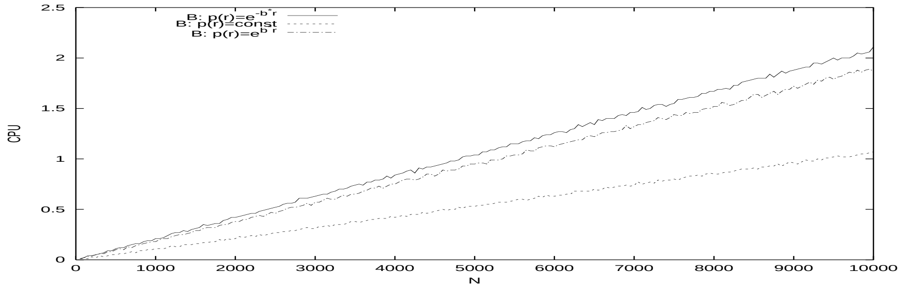

Figure 8: Comparison of the CPU times among algorithms B in case, when $\varepsilon = 0.01$, $a = -1$ for Test 2.

	Test 1		Test 2	
	$El_{(\varepsilon,A)}$	$El_{(\varepsilon,B)}$	$El_{(\varepsilon,A)}$	$El_{(\varepsilon,B)}$
case I				
$\varepsilon = 0.01$	36.3416	36.2685	36.6713	36.6653
$\varepsilon = 0.05$	16.5512	16.5516	16.8629	16.8077
$\varepsilon = 0.1$	9.4304	9.3719	9.5875	9.5780
case III				
$\varepsilon = 0.01$	35.9657	36.0139	35.6345	35.5995
$\varepsilon = 0.05$	16.2966	16.2906	15.9906	16.0484
$\varepsilon = 0.1$	9.1904	9.2176	9.0142	9.0186

Table 2: The average lengths of Markov chains in case when $a = 0.25$.

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), pp. 209 - 232.

[3] I.T. Dimov, T.V. Gurov, Estimates of the computational complexity of iterative Monte Carlo algorithm based on Green's function approach. **Mathematics and Computers in Simulation.** Vol. 47 (2-5), 1998, pp. 183-199.

[4] I.T. Dimov, R.J. Papancheva, Green's function Monte Carlo algorithms for elliptic problems, submitted to journal of Mathematics and Computers in simulations.

[5] S.M. Ermakov, G.A. Mikhailov, Statistical Simulation, **Nauka**, Moscow, 1982.

[6] S.M. Ermakov, V.V. Nekrutkin, A.S. Sipin, Random Processes for Solving Classical Equations of the Mathematical Physics, **Nauka**, Moscow, 1984.

[7] T.V. Gurov, P.A. Withlock, I.T. Dimov, A grid free Monte Carlo algorithm for solving elliptic boundary value problems, L.Vulkov et al. eds., Lecture notes in Comp. Sci.,Vol. 1998, **Springer-Verlag**, 2001, 359-367.

[8] V.P. Mikhailov, Partial Differential Equations, **Nauka**, Moscow, 1983.

[9] C. Miranda, Equasioni alle dirivate parziali di tipo ellipttico. **Springer-Verlag**, Berlin, 1955.

[10] R.J. Papancheva, I.T. Dimov, T.V. Gurov, A new class grid-free Monte Carlo algorithms for elliptic boundary value problems, submitted to the proceeding of the NMA'02 Conference 20-24.08.2002, Borovets, Bulgaria.

[11] I.M. Sobol, Monte Carlo Numerical Methods, **Nauka**, Moscow, 1973.