ACADEMIC
PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 186 (2003) 527–544

JOURNAL OF
COMPUTATIONAL
PHYSICS

www.elsevier.com/locate/jcp

Generalized random walk algorithm for the numerical modeling of complex diffusion processes

Călin Vamoş [a], Nicolae Suciu [a,*], Harry Vereecken [b]

[a] *Romanian Academy, "T. Popoviciu" Institute of Numerical Analysis, Republicii Str. 37, P.O. Box 68, 3400 Cluj-Napoca 1, Romania*
[b] *Forschungszentrum Jülich GmbH, Institut für Agrosphäre (ICG-IV), D-52425, Jülich, Germany*

Received 11 December 2001; received in revised form 10 January 2003; accepted 25 January 2003

Abstract

A generalized form of the random walk algorithm to simulate diffusion processes is introduced. Unlike the usual approach, at a given time all the particles from a grid node are simultaneously scattered using the Bernoulli repartition. This procedure saves memory and computing time and no restrictions are imposed for the maximum number of particles to be used in simulations. We prove that for simple diffusion the method generalizes the finite difference scheme and gives the same precision for large enough number of particles. As an example, simulations of diffusion in random velocity field are performed and the main features of the stochastic mathematical model are numerically tested.

Keywords: Diffusion; Random walk; Groundwater; Contaminant transport

1. Introduction

It is well known that diffusion processes can be numerically simulated with random walk (RW) algorithm. For simple diffusion processes the RW algorithm is equivalent with the finite difference (FD) scheme [1] but, as we shall discuss in the following, this equivalence is not valid for more complex diffusion processes. The RW algorithm can be used to model the transport of arbitrary physical quantities if parts of the transported quantity are associated with fictitious particles obeying the RW law. To reduce the computational effort and to improve the smoothness of the numerical solution, the gradient of the transported quantity can be associated with the particles. Integrating the gradient transported by each particle over the computation domain the simulated field is obtained with a higher accuracy [8]. The "gradient random walk" algorithm was first developed by Chorin [7] for the simulation of turbulence, the transported quantity being the vorticity. In other applications, mainly for transport in porous media, to reduce memory and to avoid boundary effects, a grid free algorithm called "particle tracking" (PT) is used [22]. The

[*] Corresponding author.
E-mail addresses: cvamos@ictp.acad.ro (C. Vamoş), n.suciu@fz-juelich.de (N. Suciu), h.vereecken@fz-juelich.de (H. Vereecken).

0021-9991/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0021-9991(03)00073-1

algorithm consists in building trajectories in continuous space for each particle, by performing at discrete time steps an advection displacement and a random Gaussian one.

The application of RW algorithm to solve some practical transport problems is relatively limited. A good estimation of the concentration field requires a large number of particles at each grid point. For every jump of each particle a random number should be generated implying a certain number of numerical operations. Therefore, "if we require that the numerical solution should be identical to the analytic solution to three significant figures, the number of tracer particles simulated must be enormous" [19, p. 95]. In this paper we propose an improvement of the RW algorithm aiming to eliminate this limitation: all the particles from a given grid node are moved simultaneously, not individually. This procedure is possible since the number of the particles jumping from a given node to a single neighboring node obeys a Bernoulli distribution. In this way a great number of particles can be distributed generating only a single random number and the necessary random number generations are significantly reduced. We call this algorithm "Global Random Walk" (GRW). A more general form of the GRW algorithm is obtained when a part of the particles remain at the initial node and only the rest of them are scattered to the neighboring nodes according to the Bernoulli distribution.

The number of particles jumping from a given node to a single neighboring node according to GRW fluctuates about the mean value. These fluctuations can be eliminated if we allow the particles to be divided into parts and exactly half of them to jump to the left and the other half to the right. We show that for simple diffusion processes this "determinist" GRW algorithm (GRWD), without fluctuations, is identical with FD algorithm. Thus, GRW can be thought as a generalization of FD algorithm. If we do not intend to give up the particle indivisibility, then the fluctuations cannot be completely canceled. The minimum magnitude of the particles number fluctuations is equal to a single particle. In this case a form of GRW with reduced fluctuations, denoted GRWR, is obtained: if at a node there are an odd number of particles, then one of the particles is randomly distributed to the left or right and the rest of them is divided to half.

To show that the RW algorithm gives a numerical solution of the diffusion equation, let us consider an infinite lattice given by a grid with nodes at $x_i = i\delta x$, $i \in \mathbb{Z}$, where $\delta x > 0$ is the lattice constant or the space step. At time $t_k = k\delta t$, $k \in \mathbb{N}$, where $\delta t > 0$ is the time step, the particle jumps to the left or right neighboring node with equal probabilities. If we denote by $P(x_i, t_k)$ the probability distribution at t_k and by $P(x_i, t_k | x_j, t_l)$ the transition probability, then the consistency property of finite dimensional probabilities for discrete Markov processes gives

$$P(x_i, t_k) = \sum_j P(x_i, t_k | x_j, t_l) P(x_j, t_l), \tag{1.1}$$

where $t_k > t_l$. According to the RW law, the transition probabilities for successive time steps are given by

$$P(x_i, t_{l+1} | x_j, t_l) = \begin{cases} \frac{1}{2} & \text{for } i = j \pm 1, \\ 0 & \text{for } i \neq j \pm 1. \end{cases} \tag{1.2}$$

The Chapmann–Kolmogorov equation and (1.2) give the evolution equation for the transition probability

$$P(x_i, t_{k+1} | 0, 0) = \tfrac{1}{2}[P(x_{i-1}, t_k | 0, 0) + P(x_{i+1}, t_k | 0, 0)]. \tag{1.3}$$

One can prove (see for instance [5]) that if $\delta x \to 0$ and $\delta t \to 0$, and if the limit

$$\lim_{\delta x, \delta t \to 0} \frac{\delta x^2}{2\delta t} = D \tag{1.4}$$

is finite, then the limit of the solution of (1.3) is the Gaussian transition probability with the density

$$p(x,t|0,0) = (4\pi Dt)^{-1/2}\exp\left\{\frac{-x^2}{4Dt}\right\}. \tag{1.5}$$

The Gaussian process is a continuous Markov process and the probability density that a particle has the position x at time t,

$$p(x,t) = \int_{\mathbb{R}} p(x,t|x',0)p(x',0)\,\mathrm{d}x', \tag{1.6}$$

is the solution of the diffusion equation

$$\partial_t p = D\partial_x^2 p, \tag{1.7}$$

with the initial condition $p_0(x) = p(x,0)$ [5].

The connection between the RW process, described by (1.1) and (1.2), and the solution (1.6) of the diffusion equation is used to build up the RW algorithm to solve the diffusion equation (1.7). One considers N fictitious particles moving on the previously described infinite grid, their movement being governed by the RW law. At the initial time the distribution of the N particles approximates the values of the initial probability density $p_0(x)$. For each particle and each time step a random number taking the values -1 and $+1$ with a probability equal to 1/2 is generated. If the random number is -1, then the particle moves to the left and if it is $+1$, the particle moves to the right. In this way the distribution of the N particles at the time t_k approximately describes the solution $p(x,t_k)$ of the diffusion equation (1.7). This approximation can be improved by increasing N and reducing δx and δt.

In Section 2 we present the GRW algorithm and discuss the connections with FD algorithm. In Sections 3 and 4 we analyze GRW simulating the Gaussian solution of the one-dimensional diffusion equation far from the boundaries and with several boundary conditions. We show that for a large enough number of particles N, the numerical solution obtained with GRW coincides with the FD solution, for all spatial resolutions.

In Section 5, we simulate the diffusion in random velocity fields. In this case GRW is no more equivalent with FD algorithm. Instead of (1.7), we consider advection–diffusion equations of the form

$$\partial_t c + V(x)\partial_x c = D\partial_x^2 c, \tag{1.8}$$

where the advection velocity $V(x)$ is a space function and the diffusion coefficient D is a constant. The relation between the concentration field c in (1.8) and the probability distribution p is given by

$$c(x,t) = N_0 p(x,t), \tag{1.9}$$

where N_0 is the total number of diffusing particles,

$$N_0 = \int_{-\infty}^{\infty} c(x,t)\,\mathrm{d}x. \tag{1.10}$$

If the advection velocity has large spatial variations then $V(x)$ is modeled as a random velocity field. Such problems occur in studies of transport processes in heterogeneous porous materials. Many authors agree that numerical simulations of diffusion in random fields are better achieved by RW algorithms than by usual FD or finite element algorithms [13,16,20–22]. We shall show that GRW allows the simulation of the two-dimensional diffusion in random velocity fields using moderate computing resources.

2. The GRW numerical algorithm

The GRW algorithm is identical with the RW algorithm in its mathematical principles, only the numerical implementation of the particles displacement is different. Hence, first we describe the RW algorithm for Eq. (1.8) using the same lattice as for (1.1)–(1.3). According to (1.4), we relate the lattice constant δx to the time step δt, for a given constant diffusion coefficient D, by

$$D = \frac{\delta x^2}{2\delta t}. \tag{2.1}$$

The particles in x_i jump either in $(x_i - \delta x)$ or in $(x_i + \delta x)$. Similarly, the realization of the random velocity field is described on the grid by a set of integers v_i, defined by

$$V(x_i) = \frac{\delta x}{\delta t} v_i, \tag{2.2}$$

so that the movement of particles in x_i due to advection field is given by $(x_i + v_i \delta x)$. The total displacement of particles is obtained, similarly to PT algorithm [20], as the sum of advective and diffusive displacements. After a time step δt, the particles starting from the node i reach either the node $(i + v_i - 1)$ or $(i + v_i + 1)$. The shortcoming of this approach is that $V(x_i)$ has only discrete values and if $V(x)$ has large variations, then $\delta x / \delta t$ must be small imposing a small space step. Therefore, in this case very large grids are needed. Also, the velocity $V(x_i)$ must be replaced by its average over a time step. For smooth variations, it was shown that $(V(x_i) + V(x_i + \delta t V(x_i)))/2$ approximates the average up to second order in δt [16].

At a given time $t_k = k\delta t$, the N particles are distributed on the grid so that $n(i,k)$ is the number of particles at the node $x_i = i\delta x$. Moving each particle according to the RW law, the distribution of the particles at the next time step $n(i, k+1)$ is obtained. Repeating S times the simulation under the same initial condition, we obtain for each node i and time k a sequence of values $n_s(i,k)$, $1 \leqslant s \leqslant S$. When S is large enough, we can approximate the solution $c(x,t)$ of the advection–diffusion equation (1.8) using the average over the S realizations of $n_s(i,k)$, denoted $\overline{n(i,k)}$. Assuming that the particles in the node x_i are assigned to an interval consisting of l space steps, the numerical solution of (1.8) is given by

$$c(x_i, t_k) = \frac{1}{l\delta x} \overline{n(i,k)}. \tag{2.3}$$

Usually $l = 1$, but it is possible that only half of the grid nodes can contain particles and then $l = 2$ (see Section 3).

The GRW algorithm moves the particles in large groups, not individually. We denote by $\delta n(i,j,k)$ the number of particles which at time t_k jump from x_j to x_i. Then the distribution of the particles at the next time is obtained from the relation

$$n(i, k+1) = \delta n(i,i,k) + \sum_{j \neq i} \delta n(i,j,k), \tag{2.4}$$

where $\delta n(i,i,k)$ is the number of particles remaining at t_{k+1} at the same node x_i. For every j and k, the scattering of the particles is given by the relation

$$n(j,k) = \delta n(j,j,k) + \delta n(j + v_j - d, j, k) + \delta n(j + v_j + d, j, k), \tag{2.5}$$

where the positive integer d describes the jumps of particles due to diffusion. From (2.5) it follows that the contributions $\delta n(i,j,k)$, $j \neq i$, in (2.4) come from all points j satisfying $j + v_j \pm d = i$. The GRW algorithm is defined if a procedure to calculate the values of the quantities in the right hand part of (2.5) is given.

We want that at a given time step only a fraction r of the number of particles jump in the neighboring nodes, the rest of them remaining at the same node. To avoid the division of particles, r must be a positive rational number $r \leqslant 1$, such that $(1 - r)N$ be an integer and equal to the total number of particles remaining at the same node at a time step. For increasing index j, we determine the number of particles remaining at a node x_j by means of the formula

$$\delta n(j,j,k) = \left[(1-r)\sum_{j'\leqslant j} n(j',k)\right] - \left[(1-r)\sum_{j'<j} n(j',k)\right], \tag{2.6}$$

where $[\cdot]$ is the integer part of the expression in the brackets. Taking the average over a great number of Monte Carlo realizations we obtain

$$\overline{\delta n(j,j,k)} = (1-r)\overline{n(j,k)}. \tag{2.7}$$

Since $\delta n(j,j,k)$ is known from (2.6), (2.5) relates the random variables $\delta n(j+v_j-d,j,k)$ and $\delta n(j+v_j+d,j,k)$ and only one of them has independent values.

As a consistency requirement, for a given diffusion process, the GRW algorithm must give the same mean square displacement as the RW algorithm. If δx_{RW} is the space step for the RW algorithm, then for a time step δt the mean square displacement of the particles in the node j is $\overline{n(j,k)}\delta x_{\mathrm{RW}}^2$, because all the particles jump at the first neighbors. For an equal time step, in GRW algorithm only the fraction r of the particles in the node j jump at the nodes $j \pm d$ and, from (2.5) and (2.7), the mean square displacement is $r\,\overline{n(j,k)}(d\delta x)^2$. Imposing the condition $\overline{n(j,k)}\delta x_{\mathrm{RW}}^2 = r\,\overline{n(j,k)}(d\delta x)^2$ and using (2.1) for δx_{RW}, the parameter r is given by

$$r = \frac{2D\delta t}{(d\delta x)^2}. \tag{2.8}$$

Usually the values of δx and δt are mainly determined by the necessity to describe enough accurately the spatial variations of the velocity field $V(x)$ and by the limitations of the available memory and computing speed. Then for any value of the diffusion coefficient D, even if it has spatial variations, the relation (2.8) allows us to choose the values of d and r. This is not possible using only the parameter d, because it takes discrete values. On the other hand r is a continuous parameter, but its value is restricted by the condition $r \leqslant 1$. Combining the unboundedness of d and the continuity of r we can obtain any value of D.

The GRW algorithm performs the evaluation of the random variables $\delta n(j+v_j \pm d,j,k)$ directly, not as a sum of the individual jumps of the $n \equiv n(j,k) - \delta n(j,j,k)$ particles. Since each of the n particles can reach the node $j+v_j-d$ with a probability equal to 1/2, it follows that the probability for $\delta n(j+v_j-d,j,k)$ to take the value m, $0 \leqslant m \leqslant n$, is given by the Bernoulli distribution $b_n(m) = 2^{-n}C_n^m$. To assign to $\delta n(j+v_j-d,j,k)$ a random value satisfying the Bernoulli distribution, at each time step, a random number η with a uniform distribution in the interval [0, 1] is generated. If we denote by $F_n(m) = \sum_{l=0}^{m} b_n(l)$, $0 < F_n(m) \leqslant 1$, the Bernoulli repartition, then $\delta n(j+v_j-d,j,k)$ takes the value m satisfying the condition $F_n(m-1) \leqslant \eta < F_n(m)$, where we use the convention $F_n(-1) = 0$.

To analyze the relation between the GRW and the FD algorithms, we consider the centered differences and time explicit scheme for diffusion equation obtained from (1.8) when $V(x) \equiv 0$. Considering the approximation in the order $\mathcal{O}(\delta x^2)$ of $\partial_x^2 c$ in (1.8), using finite difference between $d\delta x$ space steps and the parameter r defined by (2.8), the explicit FD scheme can be written as

$$c(i,k+1) = \frac{r}{2}c(i+d,k) + (1-r)c(i,k) + \frac{r}{2}c(i-d,k). \tag{2.9}$$

The solution of (2.9) is stable if the von Neumann stability criterion, $r \leqslant 1$, is fulfilled. Since from (2.8) we also have $\delta t = \mathcal{O}(\delta x^2)$, the FD scheme (2.9) is a consistent approximation of the exact partial differential equation within the approximation order $\mathcal{O}(\delta x^2)$. The stability and consistency imply the convergence of the order $\mathcal{O}(\delta x^2)$ for the initial value problem attached to (1.8) with $V(x) \equiv 0$ [6]. For $v_i \equiv 0$, the relation (2.4) becomes

$$n(i,k+1) = \delta n(i,i,k) + \delta n(i,i+d,k) + \delta n(i,i-d,k).$$

Taking into account (2.3) and (2.7), the average of this relation is identical with (2.9) if

$$\overline{\delta n(j \pm d, j, k)} = \frac{1}{2} r \overline{n(j,k)}. \tag{2.10}$$

But this is the RW law stating that the average number of particles jumping in a direction is equal to half the total number of particles. This proves that the FD solution is identical with the ensemble average of the GRW solutions. The parameter (2.8) defining the particles fraction jumping to the neighboring nodes in (2.6) is the same as the stability parameter of the FD scheme (2.9).

One can define a modified GRW algorithm which is identical with the FD algorithm for $V(x) \equiv 0$, if the particles can be divided and $n(j,k)$ is a real number, not an integer. Instead of (2.6) we introduce

$$\delta n(j,j,k) = (1-r)n(j,k) \tag{2.11}$$

and in analogy with (2.10) we consider

$$\delta n(j + v_j \pm d, j, k) = \frac{1}{2} r n(j,k). \tag{2.12}$$

Then (2.5) is identically satisfied and all the quantities in (2.4) are defined. In this case, $\delta n(j + v_j - d, j, k)$ is not anymore a random variable but its value is uniquely determined by (2.12) and coincides with the mean value of the corresponding random variable in GRW. Therefore we call this modified algorithm as a "deterministic" GRW (GRWD).

Another form of the GRW algorithm can be obtained by both preforming a deterministic scattering and preserving the particles indivisibility. We use (2.6) and instead of (2.12) we introduce

$$\delta n(j + v_j - d, j, k) = \begin{cases} n/2 & \text{if } n \text{ is even,} \\ [n/2] + \theta & \text{if } n \text{ is odd,} \end{cases} \tag{2.13}$$

where $n = n(j,k) - \delta n(j,j,k)$, $[n/2]$ is the integer part of $n/2$ and θ is a random variable taking the values 0 and 1 with probability 1/2. The quantity $\delta n(j + v_j + d, j, k)$ is determined by (2.5). In comparison with GRW, this algorithm reduces the particles number fluctuations at only one particle and we call it GRWR. Since the fluctuations do not vanish, only the average of the GRWR solution is identical with the FD solution. The algorithms without fluctuations (GRWD) and with reduced fluctuations (GRWR) can be used to obtain numerical solutions for the advection–diffusion equation (1.8), as well as the stochastic algorithm GRW. The GRW algorithm is expected to be more accurate when the fluctuations significantly influence the simulated process [9].

The GRW algorithm and its modified forms GRWD and GRWR use the relation (2.4) where $\delta n(i,j,k)$ is nonvanishing for every j satisfying $j + v_j \pm d = i$. Therefore, if $V(x)$ varies in space, the evolution of the concentration in a node is obtained, unlike in (2.9), by contributions from more than the first neighboring nodes. The terms in (2.4) are not apriori known, because they depend on the value of V in x_j. In this case, the GRW algorithm is no more equivalent with a FD scheme.

The implementation of the GRW algorithm as a computer code, encounters some problems related to the computation of the Bernoulli distribution $b_n(m) = 2^{-n} C_n^m$ and of the corresponding repartition $F_n(m) = \sum_{i=0}^{m} b_n(i)$. When the number of particles is of order 10^6, the computation of $b_n(m)$ or $F_n(m)$ takes too much time to be performed at each computation step. Therefore the values of $F_n(m)$ are computed only once and stored in files for values $n = 2^k$ with $1 \leqslant k \leqslant 20$. Due to the symmetry of $F_n(m)$ with respect to $m = n/2$, only the values $F_n(m) \leqslant 0.5$ are stored. If $n < 2^{21}$, a binary representation $n = \sum_{l=0}^{20} a(l) 2^l$ is used. The 2^l particles of a group with $a(l) \neq 0$ are scattered in $\delta n_l(j + v_j - d, j, k)$ and $\delta n_l(j + v_j + d, j, k)$, as previously described, using a random number η uniformly generated in the interval $[0, 1]$. The final result is obtained from

$$\delta n(j+v_j-d,j,k)=\sum_{l=0}^{20}a(l)\delta n_l(j+v_j-d,j,k).$$

If $n \geqslant 2^{21}$, then there are several groups consisting of 2^{20} particles and for each group the procedure from above can be used. This method, referred to as GRW0 (first used in [23]), becomes time expensive for very large n (see the "GRW0" curve in Fig. 9). In this paper a different method is used. The reduced variable $\xi=(m-n/2)/\sqrt{n/4}$ and the repartitions $F_n(\xi)$ are introduced. For $n\to\infty$, the repartition $F_n(\xi)$ tends to the normal Gaussian repartition, according to De Moivre–Laplace theorem [15]. A good approximation is obtained when for every $n \geqslant 2^{21}$ one uses the repartition corresponding to $n=2^{20}$ as function of the reduced variable ξ. For instance, the relative error of the values δn obtained using $F_{2^{20}}$ instead of $F_{2^{30}}$ is of the order 10^{-9}. In this way, GRW can handle a number of particles equal to the maximum number of particles that can be represented in the internal memory of the computer. For two and three-dimensional problems, the GRW algorithms are implemented by performing the one-dimensional global scattering procedures described in this section on x_1, x_2 and x_3 space axes, according to the values of velocity components and diffusion coefficients.

3. Simulation of the one-dimensional Gaussian diffusion

We verify the GRW algorithm described in the previous section for the solution of Eq. (1.8) corresponding to $V(x)\equiv 0$, $D(x)\equiv 0.5$ and initial condition $\lim_{t\to 0+}c(x,t)=N_0\delta(x)$, where $\delta(x)$ is the Dirac function. In this case the solution has a Gaussian analytic form and, from (1.5) and (1.9), it is given by

$$c_{\text{Gauss}}(x,t)=N_0(2\pi t)^{-1/2}\exp\left\{\frac{-x^2}{2t}\right\}. \tag{3.1}$$

Different numerical solutions obtained by GRW are quantitatively compared with the analytical solution (3.1) in the space interval $x\in[-1,1]$. The comparison is achieved at the time t_{f}, when the number of particles which left the interval $[-1,1]$ is 1% of the total number of particles. From the condition

$$\frac{1}{N_0}\int_{-1}^{1}c_{\text{Gauss}}(x,t_{\text{f}})\,\mathrm{d}x=\operatorname{erf}\left(\frac{1}{\sqrt{2t_{\text{f}}}}\right)=0.99,$$

we have $t_{\text{f}}=0.15$.

The numerical solution is obtained using the GRW, GRWD and GRWR algorithms, for $v_i=0$ and $d=1$. Initially N fictitious particles are introduced in the origin of the space grid. We consider a sequence of grids with the space steps $\delta x=(10g)^{-1}$, where $g=1,2,\ldots,10$. Since $D=0.5$, from (2.8) it follows that the corresponding time steps are $\delta t=r\delta x^2=r(10g)^{-2}$ and the numerical simulation contains $k_{\text{f}}=t_{\text{f}}/\delta t=15g^2/r$ time steps. To eliminate the boundaries effect on the numerical solution we must choose a large enough grid so that no particles may reach the boundary at time t_{f}. A particle covers the maximum distance if it makes all the k_{f} jumps in the same direction. Therefore the space grid must contain at least k_{f} nodes on a side and on the other of the origin where all the N particles initially are located. Hence for different values of r and g (δx) we obtain different numerical algorithms for $D=0.5$ and $V(x)=0$.

We want to compare c defined by (2.3) with the analytic solution (3.1), at time t_{f} and over the spatial interval $[-1,1]$. For $r=1$ some additional problems occur. Since initially all the N particles are at a single node, if $r=1$ the simulation takes place on a single numerical mode, i.e., at even time steps the particles are distributed only on the even nodes $2p\delta x$ and the odd nodes $(2p+1)\delta x$ are empty, and vice versa. So, at a given time only half of the grid nodes contains particles and in (2.3) we must consider $l=2$, i.e., $c=\bar{n}/(2\delta x)$. The nodes corresponding to $x=\pm 1$ are always even, $\pm 1/\delta x=\pm 10g$, and we introduce the quantity

$$i_{\mathrm{f}} = \begin{cases} 10g & \text{if } k_{\mathrm{f}} \text{ is even,} \\ 10g - 1 & \text{if } k_{\mathrm{f}} \text{ is odd,} \end{cases}$$

such that $x_{\mathrm{f}} = \pm i_{\mathrm{f}}\delta x$ should be the node in the interval $[-1, 1]$ which contains particles at t_{f} and is the closest to $x = \pm 1$. Then for $r = 1$ we characterize the accuracy of the solution c with respect to the analytical solution c_{Gauss} by the norm $\|c - c_{\mathrm{Gauss}}\|$, defined as

$$\|c - c_{\mathrm{Gauss}}\|^2 = \frac{1}{i_{\mathrm{f}} + 1} \sum_{i=0}^{i_{\mathrm{f}}} \left[\frac{1}{N} c((2i - i_{\mathrm{f}})\delta x, t_{\mathrm{f}}) - \frac{1}{N_0} c_{\mathrm{Gauss}}((2i - i_{\mathrm{f}})\delta x, t_{\mathrm{f}}) \right]^2, \tag{3.2}$$

where only the nodes containing particles at t_{f} are taken into account. If $r < 1$, then the two numerical modes are mixed by the particles remaining at the same node. In this case one can use a formula analogous to (3.2) with $i_{\mathrm{f}} = 10g$, summing over all $(2i_{\mathrm{f}} + 1)$ nodes of the grid and with $c = \bar{n}/\delta x$, defined by (2.3) for $l = 1$.

First we analyze the numerical solution obtained with the GRWD algorithm described by (2.11) and (2.12) with $n(i, k)$ real number. In the previous section it was proved that, for constant D and $V \equiv 0$, GRWD is identical with the explicit FD scheme (2.9), hence we denote its solution by $c_{\mathrm{FD}}(x_i, t_k)$. In Fig. 1, the norm $\|c_{\mathrm{FD}} - c_{\mathrm{Gauss}}\|$ is represented as function of δx^2 for several values of the parameter r. The linear behavior corresponds to convergence in δx^2 order of the FD scheme. From Fig. 2 it follows that the maximum precision of the finite difference scheme is obtained for $r \approx 0.3$.

In the following, besides $d = 1$ and $D = 0.5$, we also fix the parameter $r = 1$. We denote by $c(x_i, t_k)$ the numerical solution (2.3) obtained with the GRW algorithm described in the previous section. In fact this solution also depends on the spatial resolution $\delta x = (10g)^{-1}$, the total number of the fictitious particles N,

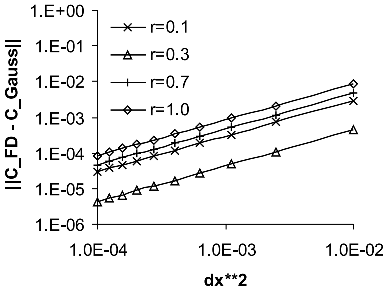

Fig. 1. Convergence of order δx^2.

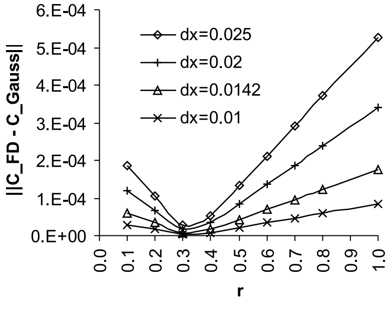

Fig. 2. Dependence on r.

and the number of simulations S, i.e., $c = c(x_i, t_k; \delta x, N, S)$. We successively analyze the dependence of the solution accuracy on these parameters.

First for $S = 1$ fixed, $\delta x = 0.1$ (Fig. 3(a)) and $\delta x = 0.01$ (Fig. 3(b)) we represent the norm (3.2) for the numerical solution obtained with GRW and GRWR algorithms. Because $S = 1$, there is a single simulation, $\bar{n}$ is identical with n and because $r = 1$, from (2.3), we have $c = n/(2\delta x)$. For a large enough number of particles N, both GRW and GRWR approximates the analytical solution as well as the FD scheme. Since the fluctuations in GRWR are reduced to minimum, this algorithm becomes equivalent to the FD scheme for a smaller value of N. For this simple diffusion process the FD scheme is always more efficient, but for complex diffusion processes as those presented in Section 5 other algorithms are needed.

To study the dependence of the precision on N, in Fig. 4 we represent for $\delta x = 0.1$ the norm $\|c - c_{\mathrm{FD}}\|$, defined similarly to (3.2), where c corresponds to GRW, as a function of $1/\sqrt{N}$. One notices that the dependence is linear. Then, Figs. 1 and 4 show that GRW converges to the analytical solution as $\mathcal{O}(\delta x^2) + \mathcal{O}(1/\sqrt{N})$. For large N, when the condition $1/\sqrt{N} = \mathcal{O}(\delta x^2)$ is satisfied, the convergence order of GRW is $\mathcal{O}(\delta x^2)$, the same as for the finite differences scheme.

In Fig. 5 we analyze the dependence of norm (3.2) on N and S, for $\delta x = 0.1$. When $N = 1$ (the points represented by ◇ in Fig. 5), then the value $\|c - c_{\mathrm{Gauss}}\| \cong 0.01$, corresponding to FD scheme, is reached only if $S \cong 10^6$ repetitions of the simulation are performed. In this case the GRW simulation is similar to those produced by PT method, where the trajectory of one particle is generated S times. The same result is

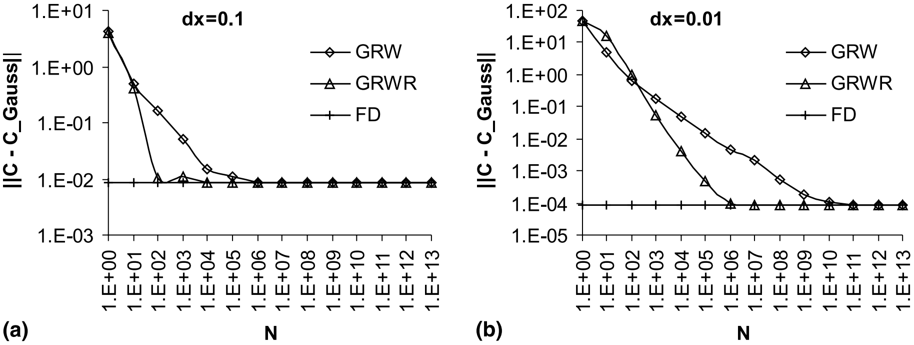

Fig. 3. Dependence on N, for $\delta x = 0.1$ (a) and $\delta x = 0.01$ (b).

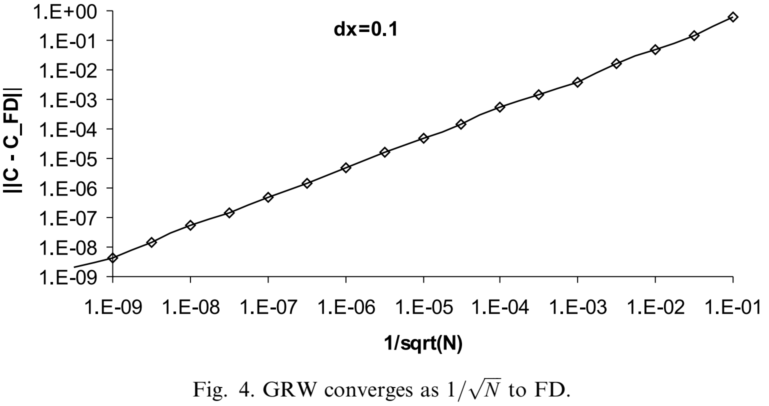

Fig. 4. GRW converges as $1/\sqrt{N}$ to FD.

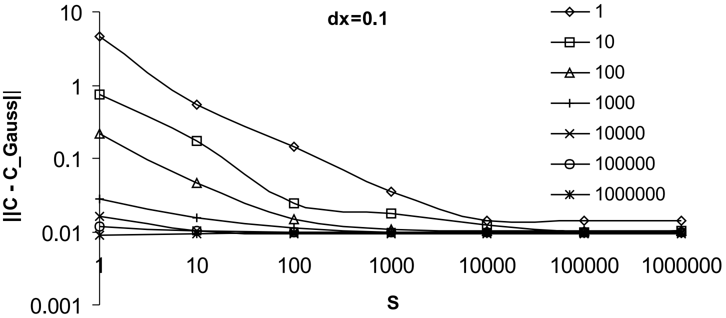

Fig. 5. Dependence on S.

obtained in a single realization, $S = 1$, for $N \cong 10^6$ particles handled with GRW (points represented by $*$ in Fig. 5). This took only 0.5 s while for $N = 1$ and $S = 10^6$ (PT procedure) the computation time was of about 17 min. PT may be thought as an "analogical Monte Carlo method" where the diffusion is modeled by averages over the diffusive behavior of individual particles. Since GRW algorithm can use large number of particles, Monte Carlo repetitions are not necessary to obtain the required precision, and the computation time is thousands times smaller.

4. Numerical boundary conditions

The boundary conditions for GRW algorithm depend on the values of d and v_i. In this section we discuss only the simplest case $d = 1$ and $v_i = 0$ analyzed in the previous section. In more complicated cases the boundary conditions can be similarly derived by means of the methods presented in the following.

To formulate the boundary conditions we use the numerical flux of particles $J(x, t)$ defined as the number of particles crossing at time t the coordinate x. We evaluate the numerical flux during a time step δt, so that the value obtained should be assigned to the middle of the time step. For $d = 1$ and $v_i = 0$ the particles jump only between the neighboring nodes, so that the numerical flux should be assigned to the middle of the space step,

$$J(i + 1/2, k + 1/2) = \frac{1}{\delta t}[\delta n(i + 1, i, k) - \delta n(i, i + 1, k)]. \tag{4.1}$$

For the GRWD algorithm, using (2.12) and (2.3) with $l = 1$, the relation (4.1) becomes

$$J(i + 1/2, k + 1/2) = r\frac{\delta x}{2\delta t}[c(x_i, t_k) - c(x_{i+1}, t_k)].$$

From (2.8) it follows that this is the usual FD form of the Fick law

$$J(x, t) = -D\partial_x c(x, t). \tag{4.2}$$

Let us consider a finite grid with $2L + 1$ nodes, $i = -L, -L + 1, \ldots, L$. Since $v_i = 0$ and $d = 1$, the boundary conditions imply only the nodes $i = \pm L$. We discuss only the boundary $i = L$, the case $i = -L$ being similar. A Dirichlet boundary condition can be formulated fixing the number of particles at the boundary $n(L, k) = n_b(k)$, with $n_b(k)$ a given function of time. For other boundary conditions, including those of von Neumann type, we must evaluate the boundary flux

$$J(L + 1/2, k + 1/2) = \frac{1}{\delta t}[\delta n(L + 1, L, k) - \delta n(L, L + 1, k)]. \tag{4.3}$$

In this formula $\delta n(L+1, L, k)$ is determined by means of the GRW algorithm from $n(L, k)$, but the number of particles $\delta n(L, L+1, k)$ jumping from outside in the node $i = L$ is unknown. Therefore the boundary condition can be formulated by calculating $\delta n(L, L+1, k)$ such that the flux (4.3) would have the requested value. Consider a von Neumann boundary condition

$$J(L+1/2, k+1/2) = J_{\mathrm{b}}(k+1/2), \tag{4.4}$$

where J_{b} is a given function of time. From (4.3) it follows the boundary condition

$$\delta n(L, L+1, k) = \delta n(L+1, L, k) - \delta t J_{\mathrm{b}}(k+1/2). \tag{4.5}$$

For $J_{\mathrm{b}}(k+1/2) = 0$, from (4.5) we obtain the boundary condition for impermeable walls. For numerical simulations with $r = 1$ which use a single numerical mode, we must take into account that (4.5) mixes the numerical modes. One can avoid the mixing of the numerical modes summing up (4.5) on two time steps.

Absorbing boundary condition corresponds to the case when all the particles leaving the grid are removed, no particles being introduced from the exterior of the grid. In this case,

$$\delta n(L, L+1, k) = 0 \tag{4.6}$$

and the flux from the grid towards its exterior has the maximum value. The stationary boundary condition imposes the equality of the fluxes on a side and the other of the boundary

$$J(L+1/2, k+1/2) = J(L-1/2, k+1/2).$$

Using (4.1) and (2.5) we obtain the boundary condition

$$\delta n(L, L+1, k) = n(L, k) - \delta n(L, L, k) - \delta n(L, L-1, k). \tag{4.7}$$

Applied to GRWD (which is equivalent to the FD algorithm), this condition becomes, by means of (2.11) and (2.12), the "transmission boundary condition" [12]

$$n(L+1, k) = 2n(L, k) - n(L-1, k).$$

If we perform a numerical simulation on a finite grid of a nonstationary diffusion process in an unbounded domain after the particles reached the grid boundary, then we must formulate special boundary conditions. A method to obtain such nonstationary conditions is to express the time derivative of the concentration at the boundary by means of the time derivatives of the inside neighboring nodes. For $V(x) = 0$, it follows from (1.8)

$$\begin{aligned}\partial_t c(x-\delta x, t-\delta t) &= D\partial_x^2 c(x-\delta x, t-\delta t) = D\partial_x^2[c(x,t) - \partial_x c(x,t)\delta x - \partial_t c(x,t)\delta t + \mathcal{O}(\delta x^2)]\\ &= \partial_t c(x,t) - D\partial_x^3 c(x,t)\delta x + \mathcal{O}(\delta x^2),\end{aligned}$$

where we used the relation $\delta t = \mathcal{O}(\delta x^2)$ derived from (2.8). Then we have

$$\partial_t c(x,t) = \partial_t c(x-\delta x, t-\delta t) + \mathcal{O}(\delta x). \tag{4.8}$$

Repeating the same argument for $\partial_t c(x-2\delta x, t-\delta t)$ we also obtain

$$\partial_t c(x,t) = 2\partial_t c(x-\delta x, t-\delta t) - \partial_t c(x-2\delta x, t-\delta t) + \mathcal{O}(\delta x^2). \tag{4.9}$$

For $x = L\delta x$ and $t = (k+1)\delta t$, these relations written in finite difference, using (2.3), give the nonstationary boundary condition

$$n(L, k+1) = n(L-1, k+1) - n(L-1, k) + n(L, k) \tag{4.10}$$

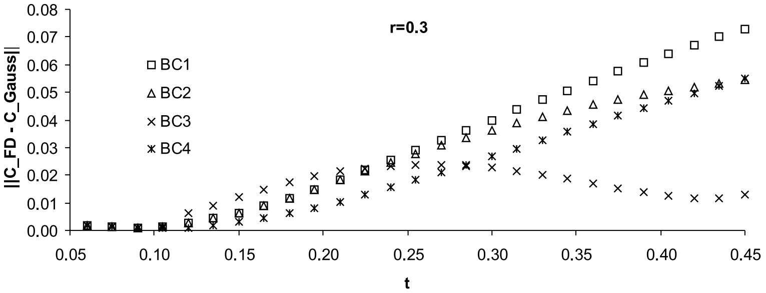

Fig. 6. Accuracy comparison.

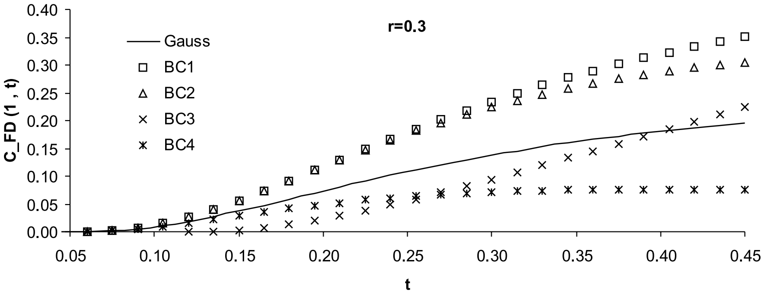

Fig. 7. Concentrations at the boundary.

and

$$n(L, k+1) = 2n(L-1, k+1) - 2n(L-1, k) - n(L-2, k+1) + n(L-2, k) + n(L, k). \tag{4.11}$$

These conditions are expected to give an useful approximation when the particles distribution near the boundary is a good approximation of the solution. But when the first particles approach the boundary, there are significant fluctuations of the particles number. Therefore at the value obtained from (4.10) and (4.11) supplementary conditions are imposed: (a) $n(L, k+1) > 0$; (b) $n(L, k+1)$ must be smaller than the value obtained from the impermeable wall condition (4.5) with $J_b = 0$.

As an illustration of these boundary conditions we continue the simulation of the previous section for temporal interval three times larger than t_f. We use the GRWD algorithm defined by (2.11) and (2.12) with the parameter $r = 0.3$. In Fig. 6 we represent the time evolution of the norm (3.2) for four different

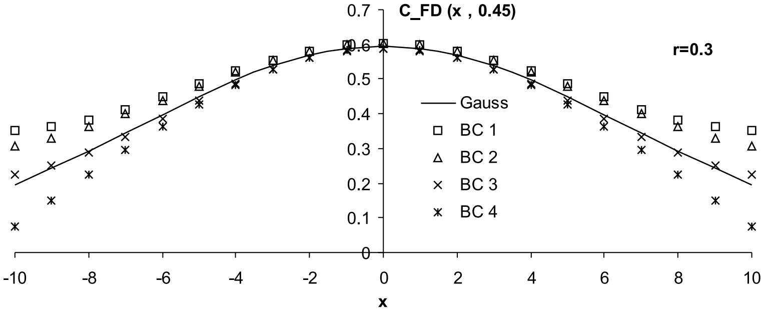

Fig. 8. Concentrations at t_f.

boundary conditions. In these simulations $\delta x = 0.1$ and the computational interval contains the spatial interval $[-1, 1]$. The points BC1 correspond to the impermeable wall boundary condition given by (4.5) with $J_{\mathrm{b}} = 0$, BC2 and BC3 to the nonstationary boundary conditions (4.10) and (4.11), and BC4 to the absorbing condition (4.6). The nonstationary condition (4.10) do not improve the accuracy with respect to (4.5) and (4.6). The condition (4.11) keeps the norm under 0.02, which proves that it is suitable to be used in simulation of nonstationary diffusion. The same conclusion can be drawn from the evolution of the boundary concentration (Fig. 7) and from the spatial variation of the concentration at the final time (Fig. 8).

5. Numerical simulation of diffusion in random velocity field

The transport in a random velocity field is a complex process consisting in diffusive movements of particles and their transport along the stream lines of the velocity field. A mathematical description of this process is given by Eq. (1.8) where D is a local diffusion coefficient and $V(x)$ a random field. Such models are used to describe the transport of pollutants in natural porous media. In this section we consider the numerical simulation of the transport of a contaminant substance in a saturated aquifer, for a punctual injection case. The heterogeneity of the advection velocities is described by realizations of a random field. Under these conditions, traditional methods (finite difference/element) are restricted at simplified aquifer models [21]. Better results in simulation of field experiments were obtained by means of the stochastic models based on PT method [20,22]. For instance, when the simulations of the contaminant transport at field scale are performed with PT, the numerical diffusion and dispersion problems occurring in finite element/difference methods are completely eliminated [13].

In PT algorithm, the diffusion process is described by the movement of an ensemble of fictitious particles in continuous space. For each particle the change of the position x in the time interval $(t_{k+1} - t_k)$, due to the realization $V(x)$ of the random velocity field and the local diffusion with coefficient D, is described, in one-dimensional case by the discrete form of the Itô equation

$$x(t_{k+1}) - x(t_k) = V(x(t_k), t_k)\delta t + G\sqrt{2D\delta t}, \tag{5.1}$$

where G is a Gaussian random variable with mean zero and unit variance. For large number of particles one expects that their number density gives an approximation of the concentration field $c(x, t_k)$ satisfying an advection–diffusion equation [22]. This assertion is based on the relation between the Itô equation and the Fokker–Planck equation for the probability density of particle position [5]. According to (1.9) the probability density is proportional to the concentration, so that (1.8) is in fact the Fokker–Planck equation. The accuracy of the solution strongly depends on the number of tracked particles. As it is mentioned in literature [19], and we also have seen in Fig. 5, where the first curve ($N = 1$) corresponds to PT procedure, a trade off should be made to reduce the computation time without affecting the accuracy. Although improvements of the algorithm have been made, the high computational costs of PT simulations is still an open problem [8]. That is why sometimes a "semianalytical" evaluation of diffusive movement is used [17]. In [24] for instance, on the basis of some numerical tests, the last term in (5.1) was taken as $4\sqrt{D\delta t}$.

The computational effort in the PT method is due to the fact that every particle is separately displaced and all the trajectories must be stored. That is why the GRW method, where groups of particles are simultaneously displaced, saves time and memory. In the following we show that GRW allows a faster and more complete simulation of the diffusion in random fields. To illustrate the advantage of our method when large numbers of particles are necessary, the computing time for GRW was compared with the computing time for a PT method ("ParTrace" code described in [14]). The same problem was solved using the same computing platform (Cray T3E). In Fig. 9 we present the simulation of an isotropic diffusion, with $D = 0.5$, into a cube the side of which consisted of 21 nodes, for 10 time steps and for different number of particles injected at the initial moment into the center of the cube.

GRW needs less than one second and only one Cray computing node while the computing time for "ParTrace" linearly increases with the number of particles and more Cray nodes are required (we stopped the computation at 10^9 particles and 256 Cray nodes). The middle curve in Fig. 9 corresponds to GRW0 algorithm, where no approximations of binary repartitions are used (see Section 2). In this case the computing time still remains orders of magnitude smaller than that of PT but, for $N > 10^9$ the time increases with N.

For the simulation of two-dimensional diffusion in random fields we consider an advection–diffusion equation similar to (1.8) and use the same procedure to generate the field as in [18]. The velocity field is divergence free and has the form $\mathbf{V}(x_1, x_2) = \mathbf{U} + \mathbf{u}(x_1, x_2)$, where $\mathbf{U} = (U, 0)$ is the mean velocity and $\mathbf{u}(x_1, x_2)$ the fluctuation. The correlation coefficients

$$R_{ll} = \langle u_l(x_1, x_2) u_l(x_1 + a, x_2) \rangle / \langle u_l(x_1, x_2) u_l(x_1, x_2) \rangle,$$

where $l = 1, 2$ and $\langle \cdot \rangle$ denotes the average over the realizations of the random field, decay as shown in Figs. 10 and 11, corresponding to an exponential correlated logarithm of the hydraulic conductivity, with correlation length $\lambda = 1$ m. For our simulation, the parameters are the same as those used in simulations represented in Figs. 11 and 14 of [18], i.e., $U = 1$ m/day, and local diffusion coefficient $D = 0.01$ m^2/day.

Because the movement due to velocity fluctuations is much larger than diffusive motion, only the spreading of the plume over regions with different velocities is important and not the fluctuations of the number of particles. That is why the simulations were performed with the reduced fluctuations algorithm

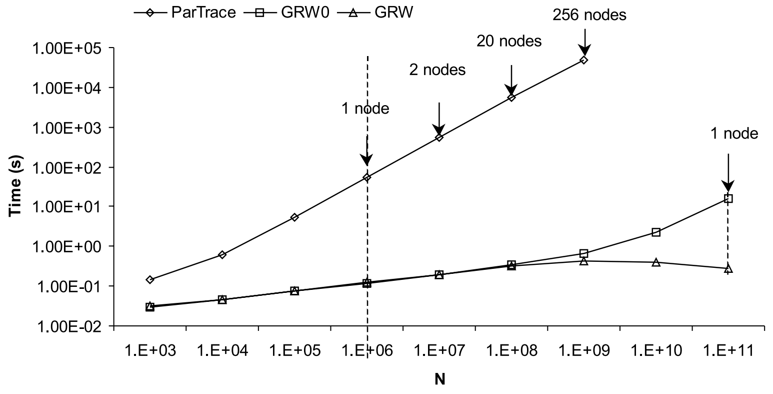

Fig. 9. Comparison of computing times

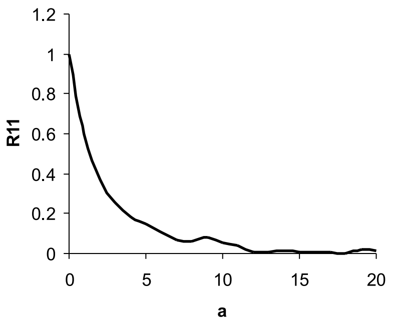

Fig. 10. Longitudinal correlation.

GRWR with $r = 1$. The GRWR algorithm requires reduced computing resources with respect to GRW and GRWD algorithms. Unlike in GRW algorithm, only one random number will be generated (at every time step and when at a given node there is an odd number of particles). Because the indivisibility of particles is preserved, the plume has a smaller extension than in the case of GRWD algorithm and, consequently, smaller grids are necessary.

The computation domain is a rectangular grid with 10^7 nodes, $0 \leqslant i \leqslant 10\,000$ and $0 \leqslant j \leqslant 1000$, the space step is $\delta x = \delta y = 0.1$ m and the time step $\delta t = 0.5$ day. In two different realizations of the velocity field, $N = 10^{10}$ particles were released at the point $(i_0, j_0) = (50, 500)$. We checked that for greater values of N the simulation results remain unchanged. A period of 800 days was simulated, so that in this interval the particles travel a mean distance of 800 correlation lengths. The computations were performed with a PC (Pentium III, 600 MHz, 64 Mb RAM) and lasted about 3 h for each simulation. In Fig. 12 we represent, at different times, the number of particles summed up over transversal sections of the plume, $n(x_1, t)$, for one realization of the velocity field.

In PT approach the diffusion in random fields is analyzed indirectly by means of the so called "effective diffusion coefficient". For a given realization of the random field, the average of $(x_l - \bar{x}_l)^2$ over the realizations of the local diffusion process, described by the second term from (5.1), defines the variance of displacements $\sigma_{ll}^2(t) = \overline{x^2}_l - \overline{\bar{x}_l}^2$. Let $\langle \sigma_{ll}^2(t) \rangle$ be its average over the realizations of the random field. The process has *asymptotic diffusive behavior* if there is a finite limit called *effective diffusion coefficient*

$$\lim_{t\to\infty} \frac{\langle \sigma_{ll}^2(t) \rangle}{t} = 2D_{ll}^*, \tag{5.2}$$

where $D < D_{ll}^* < +\infty$. When the limit (5.2) is infinite one says that the asymptotic behavior is *superdiffusive.* Processes with diffusive behavior also have the *self-averaging* property

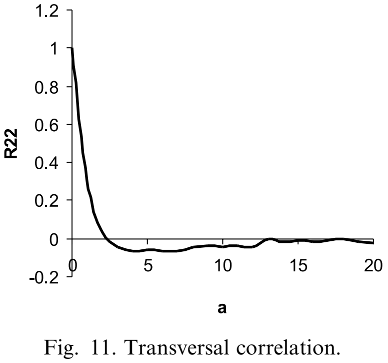

Fig. 11. Transversal correlation.

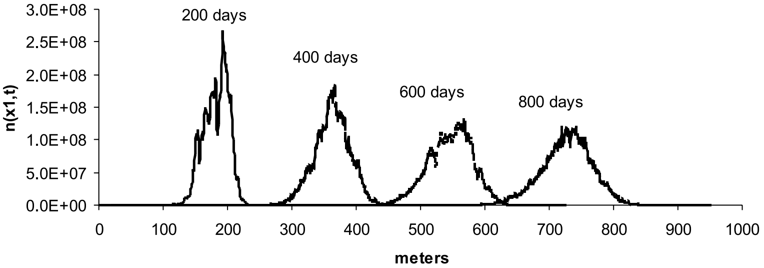

Fig. 12. Distribution of particles across the plume.

$$\lim_{t\to\infty}\frac{\sigma_{ll}^2(t)}{t}=2D_{ll}^*, \tag{5.3}$$

i.e., the coefficient defined by a single realization is the same as that defined as ensemble average [3].

In Figs. 13 and 14 the evolution of $D_{11}^*(t)$ and $D_{22}^*(t)$, for the two-dimensional GRWR simulations, are represented for the two realizations of the random field. The coefficients D_{11}^* and D_{22}^* are closed to the constant values predicted by theory presented in [18].

The basic assumption of stochastic approach of diffusion in random media is that for large times the process behaves asymptotically diffusive and, correspondingly, the self-averaging property (5.3) holds. Then one expects that the stochastic model may have predictive force for individual realizations of the porous medium [2]. "A major motivation of stochastic analysis of transport in a heterogeneous porous medium has been to derive an effective equation for the concentration that may be used to make decision in real life contamination events" [10]. First, the existence of effective diffusion coefficients (5.2) and self-averaging property (5.3) must be checked. Further, the fluctuations of concentration in individual realizations should be studied, in order to find the superior bound which delimits the "unsafe zone" [11]. Both problems are difficult to be solved and they imply high computing resources. In [18], to obtain the approach to asymptotic value of longitudinal effective diffusion coefficient, 3200 particle trajectories were simulated with PT for times corresponding at 5000 correlation lengths. In other studies [17], the reliability of the computations is achieved using 20,000 trajectories (500 realizations of the field and 40 particles in each realization), up to 20 correlations lengths but the asymptotic regime is not reached. The computational effort needed to obtain statistically significant results "has serious implication" when the behavior in a single realization is studied [4]. The limitations imposed to maximum number of particles also make it difficult to obtain simulations of concentration fields and concentration fluctuations. Although the use of GRW

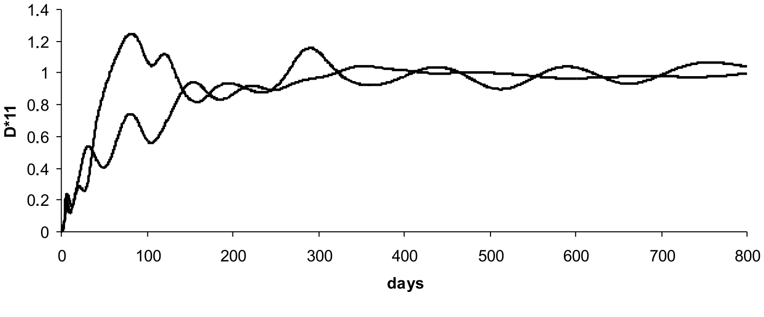

Fig. 13. Longitudinal effective diffusion coefficient.

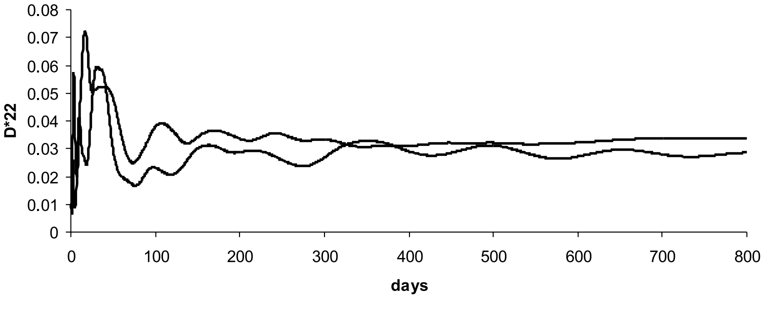

Fig. 14. Transversal effective diffusion coefficient.

method requires large grids, unlike PT which is a grid free method, such simulations can be obtained using large number of particles. The results from Figs. 12–14 indicates that GRW could be an alternative tool in modeling contaminant transport in groundwater.

6. Conclusions

The GRW algorithm moves groups of particles, unlike the usual random walk algorithm where for each particle a random number generation is necessary. Hence, important saving of memory and computing time are achieved. The number of particles is limited only by the maximum number that can be represented on each computing machine. For simple diffusion processes, GRW generalizes the FD scheme and for great enough number of particles their precision is the same. In this case, the determinist algorithm GRWD is identical with FD algorithm.

The GRWR algorithm, where the fluctuations of the number of particles are reduced to a single particle, can be used, as well as the GRWD algorithm, when the fluctuations do not influence the process essentially. A version of GRWR also can be used to avoid the approximation of the binary repartitions for $n > 2^{20}$, described in Section 2, if only groups with $n \leqslant 2^{20}$ particles are scattered using binary repartitions while the group exceeding this number is scattered by the determinist rule given by (2.12). Another GRW algorithm can be obtained when the Gaussian repartition and reduced variable is used instead of binary repartitions.

The GRW method is useful for complex diffusion processes, when the usual finite difference/element methods become inefficient. In this paper we have presented the simulation of diffusion in random velocity field. While the usual simulations uses tens of thousand of particles, the GRWR presented in this paper contains 10^{10} particles. This enables us to get informations about the asymptotic diffusive behavior and concentration fluctuations. When no effective diffusion coefficients are computed and we are mainly interested in concentration simulations, the boundary conditions discussed in Section 4 make it possible to save more computing time and memory, because smaller grids can be used.

GRW can describe other complex diffusion processes. As an example, the simulation presented in Section 5 can be completed for 3-dimensions, random space variable coefficient $D(x)$, chemical reactions between several species of particles and radioactive decay. The modeling of chemical reactions and radioactive decay can be achieved if, at each time step, the variation of the number of particles is introduced as described, respectively, by the kinetic formula for chemical reaction and the law of radioactive decay.

Acknowledgements

The authors are grateful to Dr. Jan Vanderborght and Dr. Olaf Nitzsche for fruitful discussions. C.V. and N.S. are also grateful to Prof. M. Peculea for the initial impulse in the numerical simulation of diffusion.

References

[1] W.F. Ames, Numerical Methods for Partial Differential Equations, second ed., Academic Press, New York, 1977.

[2] S. Attinger, M. Dentz, H. Kinzelbach, W. Kinzelbach, Temporal behavior of a solute cloud in a chemical heterogeneous porous medium, J. Fluid Mech. 386 (1999) 77.

[3] M. Avellaneda, A.J. Majda, Superdiffusion in nearly stratified flows, J. Stat. Phys. 69 (3/4) (1992) 689.

[4] A. Bellin, P. Salandin, A. Rinaldo, Simulation of dispersion in heterogeneous porous formations: statistics, first-order theories, convergence of computations, Water Resour. Res. 28 (9) (1992) 2211.

[5] C.W. Gardiner, Handbook of Stochastic Methods, Springer, New York, 1983.

[6] S.K. Godunov, V.S. Ryabenkii, Difference Schemes: An Introduction to the Underlying Theory, North-Holland, Amsterdam, 1987.

[7] A.J. Chorin, Vortex sheet approximation of boundary layers, J. Comput. Phys. 27 (1978) 428.

[8] A.F. Ghoniem, F.S. Sherman, Grid-free simulation of diffusion using random walk methods, J. Comput. Phys. 61 (1985) 1.

[9] W. Horsthemke, R. Lefever, Noise-induced Transitions. Theory and Applications in Physics, Chemistry and Biology, Springer, Berlin, 1984.

[10] C. Kapoor, L.W. Gelhar, Transport in three-dimensionallity heterogeneous aquifers 1. Dynamics of concentration fluctuations, Water Resour. Res. 30 (6) (1994) 1775.

[11] C. Kapoor, L.W. Gelhar, Transport in three-dimensionallity heterogeneous aquifers 2. Prediction and observations of concentration fluctuations, Water Resour. Res. 30 (6) (1994) 1789.

[12] W. Kinzelbach, Numerische Methoden zur Modellierung des Transports von Schadstoffen im Grundwasser, Oldenbourg Verlag, München, 1992.

[13] G.L. Moltyaner, M.H. Klukas, C.A. Willis, R.W.D. Killey, Numerical simulations of twin lake natural-gradient tracer tests: A comparison of methods, Water Resour. Res. 29 (10) (1992) 3443.

[14] O. Neuendorf, Numerische 3D-Simulation des Stofftransport in einem heterogenen Aquifer. Ph.D. Thesis, Jül-3421 Forschungszentrum Jülich, 1997.

[15] A. Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill, New York, 1991.

[16] K. Roth, K. Hammel, Transport of conservative chemical through an unsaturated two-dimensional Miller-similar medium with steady state flow, Water Resour. Res. 32 (6) (1996) 1653.

[17] P. Salandin, V. Fiorotto, Solute transport in highly heterogeneous aquifers, Water Resour. Res. 34 (5) (1998) 949.

[18] H. Schwarze, U. Jaekel, H. Vereecken, Estimation of macrodispersivity by different approximation methods for flow and transport in randomly heterogeneous media, Transp. Porous Media 43 (2001) 265.

[19] Ne.-Z. Sun, Mathematical Modeling in Groundwater Pollution, Springer, New York, 1996.

[20] A.F.B. Tompson, L.W. Gelhar, Numerical simulation of solute transport in three-dimensional, randomly heterogeneous porous media, Water Resour. Res. 26 (10) (1990) 2541.

[21] A.F.B. Tompson, R.B. Knapp, Reactive Geochemical Transport Problems in Nuclear Waste Analyses, Preprint UCRL-00552, Lawrence Livermore National Laboratory, Livermore, 1989.

[22] A.F.B. Tompson, R.D. Falgout, S.G. Smith, W.J. Bosl, S.F. Asby, Analysis of subsurface contaminant migration and remediation using high performance computing, Adv. Water Resour. 22 (3) (1998) 203.

[23] C. Vamoş, N. Suciu, H. Vereecken, O. Nitzsche, H. Hardelauf, Global random walk simulations of diffusion, in: W. Kramer, J.W.V. Gudenberg (Eds.), Scientific Computing, Validated Numerics, Interval Methods, 343, Kluwer Academic Publishers, Dordrecht, 2001.

[24] R. Zhang, K. Huang, M.T. van Genuchten, An efficient Eulerian–Lagrangian method for solving solute transport problems in steady and transient flow fields, Water Resour. Res. 29 (12) (1993) 4131.