GLOBAL RANDOM WALK ALGORITHM FOR DIFFUSION PROCESSES

Nicolae Suciu*
Mathematics Department, Friedrich-Alexander University Erlangen-Nuremberg, Germany
Călin Vamoş†
Tiberiu Popoviciu Institute of Numerical Analysis, Romanian Academy, Romania

Key Words: random walk, probabilistic particle methods, diffusion processes

MSC2010 Subject Classification: 65C20, 82C31, 60J60, 60J65

*E-mail address: suciu@am.uni-erlangen.de
†E-mail address: cvamos@ictp.acad.ro

Abstract

The Global Random Walk algorithm (GRW) performs the simultaneous tracking on a fixed grid of large collections of particles, while the computational costs remain comparable to those of a single-trajectory simulation by the traditional Particle Tracking (PT) approach. Unlike the sequential PT procedure, GRW simulates diffusion processes by globally distributing all the particles lying at a lattice site. The global scattering is achieved by allowing the particles to execute unbiased spatial jumps with diffusive scaling, proportional to the square root of the simulation time step. When diffusion takes place in a velocity field, the diffusion step is preceded by a drift step which moves all the particles according to the velocity field at the lattice site. If the GRW procedure is applied to a single particle, it is equivalent to a PT procedure projected on a regular lattice. Thus, GRW can be thought as a superposition of PT procedures. Moreover, it has been shown that the GRW algorithm can also be implemented as a weak Euler scheme for the Ito equation governing the continuous diffusion process, which accurately reproduces the true probability distribution. The essential difference is that while the concentration field is estimated by post-processing the trajectories simulated sequentially in the PT approach, a single GRW simulation is required for concentration estimates. In fact, the output of the GRW simulation is not an ensemble of trajectories solving the Ito equation, rather it is a solution to the associated Fokker-Planck equation. The GRW algorithm saves memory and computing time and no restrictions are imposed to the total number of particles which ensures reliable concentration estimates.

For vanishing or constant drift coefficients, GRW is also equivalent with the stable finite-difference (FD) scheme and has the same convergence order for large enough numbers of particles. However, for space-variable drift this equivalence fails and GRW suffers from overshooting errors caused by particles jumping over lattice points with different velocities. Overshooting errors can be completely removed by allowing jumps only to first-neighbor lattice sites. This can be achieved with a biased-GRW, where the drift displacement is modeled as a bias in the jump probability. The new algorithm is no longer a weak Euler scheme for the Ito equation, instead it is equivalent to a stable FD scheme without numerical diffusion. Removing the overshooting with biased-GRW has the drawback of high computational costs, due to the fine grids required by the first-neighbor jumps constraint. Therefore, the biased algorithm is mainly useful as reference to assess the accuracy of the coarser, but faster, unbiased GRW algorithms which are more efficient in solving large scale diffusion problems.

This chapter includes a presentation of the GRW algorithm, with details on its derivation, implementation, equivalence with other schemes, sensitivity to parameters, and convergence properties. Since the GRW approach is highly recommended for large scale simulations of advection-dominated transport processes, where PT has limited accuracy and finite element/difference schemes suffer from numerical diffusion, some relevant applications to contaminant transport in groundwater are also presented for the purpose of illustration.

1. Introduction

It is well known that diffusion processes can be numerically simulated with random walk (RW) algorithms. For simple diffusion processes the RW algorithm is identical with the finite difference (FD) scheme [1] but, as we shall discuss in the following, this equivalence is not valid for complex diffusion processes [43]. 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 also 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 [15]. The "gradient random walk" algorithm was first developed by Chorin [4] for the simulation of turbulence, the transported quantity being the vorticity. In other applications, mainly for transport in porous media, to save memory and to avoid boundary effects, a grid free algorithm called "particle tracking" (PT) is used [21, 40, 39]. The PT algorithm consists of generating trajectories in continuous space for each particle, by performing at discrete time steps an advection displacement and a random Gaussian one.

The application of sequential RW algorithms to solve practical transport problems is relatively limited. A good estimation of the concentration field requires a large number of particles at each grid point. Sequential algorithms generate a random number, implying a certain number of numerical operations, for every jump of each particle. Therefore, if one requires that the numerical solution should be identical to the analytic solution to three significant figures, the number of tracer particles simulated must be enormous ([38], p. 95). In this chapter we propose an improvement of the RW algorithm aiming to eliminate this limitation. Towards this aim, all the particles from a given grid node are moved simultaneously, not individually. This is possible because the number of the particles jumping from a given node to a neighbor 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 rules fluctuates about the mean value. These fluctuations can be eliminated if we allow the particles to be divided in parts and exactly, half of them jumping to the left and the other half to the right. We show that for simple diffusion processes this deterministic GRW algorithm, without fluctuations, is identical with a 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 fluctuations is that corresponding to a single particle. In this case a form of GRW with reduced fluctuations is obtained: if at a node there is an odd number of particles, then one of the particles is randomly distributed to the left or to the right and the rest of them is divided to half. Such GRW algorithms have proved to be valuable tools for large scale simulations of environmental [30, 31, 34, 35] and life-science problems [33], as well as for theoretical investigations on passive sclar transport in random media [36, 37].

Models of passive scalar transport in highly heterogeneous media, such as groundwater systems, turbulent atmosphere, or plasmas, are often based on a stochastic partial differential equation for the concentration field $c(\mathbf{x}, t)$,

$$\partial_t c + \mathbf{V}\nabla c = D\nabla^2 c, \tag{1}$$

with space variable drift $\mathbf{V}(\mathbf{x})$ which is a sample of a random velocity field, and a local diffusion coefficient D which is assumed constant [30, 31, 36, 37, 25]. The normalized concentration solving (1) for the initial condition $c(\mathbf{x}, 0) = \delta(\mathbf{x} - \mathbf{x}_0)$ is the probability density function of the diffusion process described by the Itô stochastic ordinary differential equation

$$X_i(t) = x_{0i} + \int_0^t V_i[\mathbf{X}(t')]dt' + W_i(t), \tag{2}$$

where $i = 1, 2, 3$, $x_{0i} = X_i(0)$ are deterministic initial positions and W_i are the components of a Wiener process of mean zero and variance $2Dt$ [19].

In this chapter we shall present several applications to contaminant transport in saturated groundwater systems. The time-stationary random velocity field $\mathbf{V}(\mathbf{x})$ is in this case the solution of the continuity and Darcy equations

$$\nabla\mathbf{V} = 0,\ \mathbf{V} = -K\nabla h, \tag{3}$$

where $K(\mathbf{x})$ is the hydraulic conductivity of the medium and h is the piezometric head [25]. Dirichlet boundary conditions, consisting of constant heads at the inlet and outlet boundaries of the domain, ensure the stationarity in time of the velocity field $\mathbf{V}$. The hydraulic conductivity K is supplied by various interpretations of field-scale measurements in the form of a spatially distributed random parameter (random field) [6].

If the random velocity field, obtained by solving (3) for an ensemble of realizations of the K field, has a finite correlation range then it can be shown that, under certain conditions, the ensemble mean concentration is described asymptotically by an upscaled model of form (1), with drift coefficient given by the mean velocity and enhanced diffusion coefficients proportional with the velocity correlation lengths [20, 12]. Under less restrictive conditions, with the only assumption that the first two spatial moments of the concentration are finite at finite times, the mean concentration can still be described by an equivalent Gaussian distribution with time variable diffusion coefficients [37], referred to as the "macrodispersion" model in the hydrological literature [6]. Root-mean-square deviations of the solutions to (1), for fixed realizations of the velocity field, from the predictions of the upscaled model are often used to quantify the uncertainty in stochastic modeling of transport in random environments [30, 34, 35, 36]. When the estimated mean-square uncertainty is acceptably small, one considers that "ergodic conditions" are met and the macrodispersion model can be successfully used to describe the transport in a single realization of the groundwater formation [30]. Nevertheless, for contamination risk assessments mean-square uncertainty assessments are not enough and extreme values of the stochastic predictions are also required. Such a task can be carried out by assessing the correlations and the full probability distributions of the input/output parameters [3].

When solving advection-dominated transport problems associated to (1), like the one considered here, with Péclet number Pe$= U\lambda/D = 100$, where U is the amplitude of

the mean velocity and λ a correlation length, the challenge is to ensure the stability of the solutions and to avoid the numerical diffusion [25]. Therefore, numerical solutions to the Itô equation (2), implemented in PT algorithms, are often used to simulate trajectories of computational particles and to estimate concentrations by particles densities. PT methods are stable, free of numerical diffusion, thus suitable for advection-dominated transport problems. However, since the computational costs increase linearly with the number of particles, the estimated concentrations are too inaccurate for large-scale simulations of transport in groundwater. Overcoming the limitations of the sequential PT procedure, the GRW algorithm has no limitations as concerning the number of particles [30, 43]. As shown below in Sect. 2.2, GRW provides accurate simulations of the concentration field at costs comparable to those of a single-trajectory PT simulation.

The chapter is organized as follows. After recalling basic notions about Euler schemes and PT methods in Section 2.1, we introduce in Section 2.2 the GRW algorithm as a weak numerical scheme for the Itô equation and discuss its convergence and numerical efficiency in Section 2.3. Section 2.4 contains a digression on numerical diffusion, where it is shown that, for properly constrained space and time steps, RW methods are strictly free of numerical diffusion. The exact unbiased one-dimensional GRW algorithm and different implementation options are presented in Section 3.1, followed by numerical assessment of its convergence (Section 3.2) and numerical boundary conditions (Section 3.3). In Sections 3.4 and 3.5 we present two-dimensional unbiased GRW algorithms and some relevant applications. Section 4.1 introduces the two-dimensional biased-GRW algorithm and its usefulness in evaluating the unbiased GRW algorithms is presented in Section 4.2. Further, in Section 5 we demonstrate the ability of the GRW approach to produce a detailed sensitivity and uncertainty analysis of the macrodispersion model. Conclusions are presented in Section 6.

2. Random walk simulations of diffusion processes

2.1. Itô equation and Particle Tracking

Let us consider the one-dimensional Itô equation (2) and an equidistant time discretization $0 < \delta t < \cdots < k\delta t < \cdots < K\delta t = T$. In most of its implementations, the PT simulation of the particle's trajectory consists of an Euler approximation Y_t of the solution $X(t)$, which is a continuous time process satisfying the iterative scheme

$$Y_{k+1} = Y_k + V_k\delta t + \delta W_k, \tag{4}$$

where $Y_k = Y_{k\delta t}$, $V_k = V(Y_k)$, and $\delta W_k = W_{k+1} - W_k$ is the increment of the Wiener process. The Euler scheme (4) provides a grid-free PT algorithm which generates the trajectory of a computational particle in continuous space.

While the *strong convergence* of order $\beta > 0$ of the Euler scheme requires

$$\lim_{\delta t \longrightarrow 0} E\left(|X_t - Y_t|\right) \leq C\delta t^{\beta},$$

where E denotes the expectation, for the *weak convergence* of order $\beta > 0$, it suffices that

$$\lim_{\delta t \longrightarrow 0} |E\left(g(X_t)\right) - E\left(g(Y_t)\right)| \leq C\delta t^{\beta},$$

for some functionals $g(X_t)$ (e.g. moments $E(X_t^m)$, $m \geq 1$).

For strong pathwise convergence, the Euler scheme (4) has to consider the Wiener process specified in the Itô equation (2). For weak convergence, when only the probability distribution is approximated, the increments δW_k of the Wiener process can be replaced by random variables ξ with similar moments. For weak Euler scheme of order $\beta = 1$ the first three moments of ξ have to satisfy, for some constant M, the condition [19, Sect. 5.12]

$$|E(\xi)| + \left|E(\xi^3)\right| + \left|E(\xi^2) - \delta t\right| \leq M\delta t^2.$$

Easily generated noise increments satisfying the above condition are the two-states random variables

$$\xi : \Omega \longrightarrow \{-\sqrt{2D\delta t}, +\sqrt{2D\delta t}\},\ P\{\xi = \pm\sqrt{2D\delta t}\} = \frac{1}{2}. \tag{5}$$

2.2. Weak Euler scheme and global random walk

The weak Euler scheme for equation (2) without drift term is given by (4) with $V_k = 0$ and Wiener process increment δW_k replaced by (5),

$$Y_{k+1} = Y_k + \xi, \tag{6}$$

where the increment of Y_k has a constant amplitude,

$$|Y_{k+1} - Y_k| = \sqrt{2D\delta t} = \delta x. \tag{7}$$

A computational particle described by the PT algorithm (6) moves on a regular one-dimensional lattice with lattice constant δx. If one considers a superposition of N PT procedures (6), then at a given time step k the $n(i,k)$ particles lying at the lattice site i are spread according to

$$n(i,k) = \delta n(i-1,i,k) + \delta n(i+1,i,k), \tag{8}$$

where $\delta n(i-1,i,k)$ and $\delta n(i+1,i,k)$ are, respectively, the numbers of particles jumping to the left and to the right first-neighbor sites. Since (6) does not allow particles trapping at lattice sites, the evolution of the number of particles at a grid site j is governed by

$$n(j,k+1) = \delta n(j,j-1,k) + \delta n(j,j+1,k), \tag{9}$$

When the N particles are released at $t = 0$ from a single lattice site, then the equations (8-9) describe their evolution as a succession of independent "even and odd modes": if at given k even lattice sites are occupied by particles the odd sites are necessary empty; at $k+1$ all particles from even sites jump to odd sites that were previously empty, and the previously occupied even sites become empty (see Fig. 1).

For large N it is reasonable to assume that the probability of the surrogate increments of the Wiener process (5) can be approximated by the relative frequency of lef/right jumps,

$$P\{\xi = \pm\sqrt{2D\delta t}\} = \lim_{N\longrightarrow\infty} \frac{\delta n(i-1,i,k)}{N} = \lim_{N\longrightarrow\infty} \frac{\delta n(i+1,i,k)}{N} = \frac{1}{2}. \tag{10}$$

In the limit of large N, by inserting (10) into (8) one obtains

$$\delta n(i \pm 1, i, k) = \frac{1}{2} n(i, k), \tag{11}$$

and (9) becomes

$$n(j, k+1) = \frac{1}{2} \left[n(j-1, k) + n(j+1, k) \right] \tag{12}$$

Since $n(j,k)/N$ estimates, for large N, the probability distribution of the particles at site i and time k, (12) divided by N becomes a master equation for discrete time RW, the solution of which is the Bernoulli distribution [13]. Further, one defines the particles concentration

$$c(j, k) = n(j, k)/(l\delta x), \tag{13}$$

where $l = 1$ for nonsingular initial conditions (i.e., particles distributed over more than one lattice site) and, for singular initial conditions which, as shown above, generate independent even and odd modes, we can chose $l = 2$ to avoid discontinuities in the concentration field. With (13), the relation (12) leads to

$$c(j, k+1) - c(j, k) = \frac{D\delta t}{\delta x^2} \left[c(j-1, k) + c(j+1, k) - 2c(j, k) \right], \tag{14}$$

where, according to (7),

$$D = \frac{\delta x^2}{2\delta t}. \tag{15}$$

Equation (14) is the forward-time centered in space finite difference scheme, with stability parameter [1, 5] $D\delta t/\delta x^2 = 1/2$, for the one-dimensional partial differential diffusion equation $\partial_t c = D\partial_x^2 c$, which is the Fokker-Planck equation (1) associated to the Itô equation (2), particularized to one-dimensional diffusion without drift term, i.e., the one-dimensional Wiener process.

For finite N, the number of jumping particles $\delta n(i \pm 1, i, k)$ are obviously random. For large enough N however, the random variables $\delta n(i \pm 1, i, k)$ can be approximated as follows: if the number $n(i, k)$ of particles lying at the grid site i at time k is even, then half of them jump to the left and half to the right, according to (11); if $n(i, k)$ is odd, then one particle is allocated to either $\delta n(i-1, i, k)$ or to $\delta n(i+1, i, k)$ with unbiased probability of $1/2$. This simple rule achieves the implementation of the algorithm described by (8-9). Since this approach approximates the random walk equation (12) and the random walkers are distributed over lattice sites with the global procedure described above, it has been called "global random walk" (GRW) [43]. The GRW algorithm corresponds to a superposition of PT procedures described by the weak Euler schemes (6) and can be itself thought as a weak Euler scheme which provides the particles concentration and estimations of the moments of the process (6). In case of genuine diffusion, without drift, GRW is also equivalent with the FD scheme (14) for the one-dimensional diffusion equation. But this equivalence fails in presence of a space-variable drift term (see Section 3.1 below). Nevertheless, as we will see in Sections 3.1 and 3.2 below, in all unbiased GRW algorithms the distribution of particles jumping from a lattice site at every time step is still equivalent with a superposition of weak Euler schemes for the Itô equation (2).

2.3. Numerical example, convergence and computing time

Figure 1 illustrates the evolution of the number $n(i,k)$ of random walkers over the first three simulation steps, obtained with a straightforward MATLAB implementation of the one-dimensional GRW algorithm described above. One remarks the occurrence of independent even and odd modes, as a consequence of the singular initial distribution of particles, that were localized at a single lattice site.

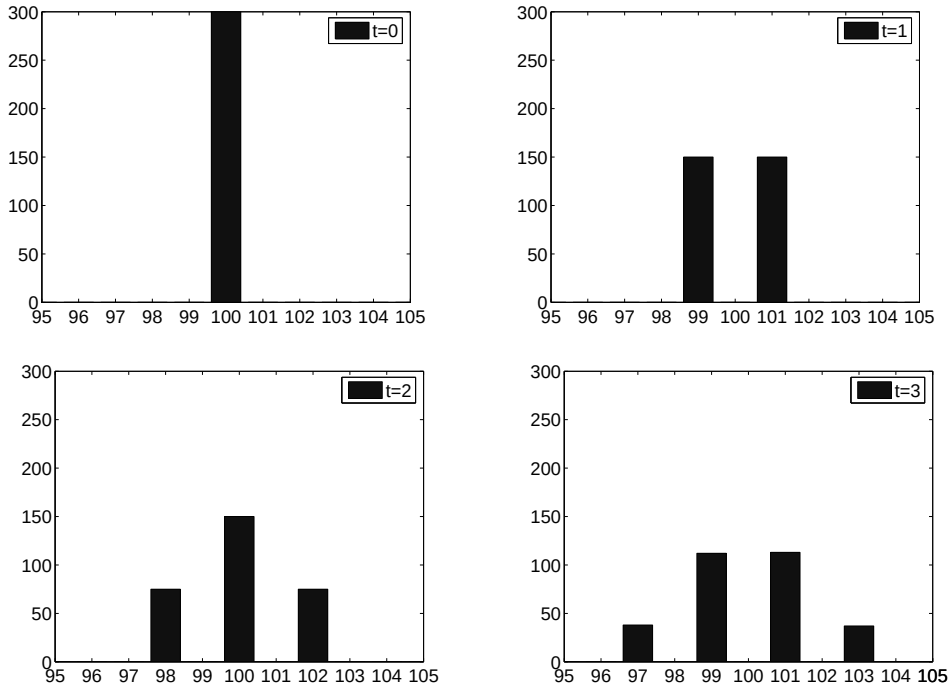

Figure 1. Distribution of $N = 300$ random walkers starting at $x = 100$ after the first three time steps of the GRW simulation.

The evaluation of the moments $E(X_t^m)$ within the numerical implementation of the weak Euler scheme (6) consists of an arithmetic average, over an ensemble of trajectories (6), of the position of the particles at a given time,

$$E(X_{k\delta t}^m) \approx \frac{1}{N}\sum_{i=1}^{N} Y_k^m,$$

which approximates the stochastic average with respect to the probability distribution, $E(X_t^m) = \int x^m p(x,t)dx$. But, as far as one approximates probability distributions and their moments, the trajectories of the weak Euler scheme are in fact not necessary. The probability density is given by the normalized concentration, $p(x,t) = c(x,t)/N$ and its GRW estimation by (13) yields $p(x,t) = n(i\delta x, k\delta t)/(Nl\delta x)$. With these, the discretization of the expectation integral giving the estimation of the m-th order moment of X_t reads

$$E(X_{k\delta t}^m) \approx \frac{1}{N}\sum_{i=1}^{L} (i\delta x)^m n(i, k\delta t), \tag{16}$$

where by L we denoted the length of the one-dimensional lattice. In particular, with (16) one obtains GRW estimates of the mean $M = E(X_t)$ and diffusion coefficient $D = [E(X_t^2) - E(X_t)^2]/(2t)$ of the Wiener process. GRW simulation results, for $\delta x = 1$ and $\delta t = 0.5$, are presented in Figs. 2-5. Figure 2 shows that the estimated mean and diffusion coefficient approximates quite well the nominal values $M = 0$ and $D = 1$ of the simulated Wiener process, even for a moderate number of particles $N = 300$. The distribution of the number of particles over the lattice, n_i, presented in Fig. 3 also shows a discrete Gaussian shape, with discontinuities typical for independent even and odd modes [43].

It is possible to further simplify the GRW algorithm by completely removing the randomness from the scheme. This can be done by setting $\delta n(i-1,i,k)$ and $\delta n(i+1,i,k)$ to the exact value of $n/2$. In this case N has no longer the meaning of a number of random walkers and can be taken as an arbitrary positive real number, for instance equal to 1. This deterministic GRW scheme is equivalent to the finite-difference scheme (14) for the one-dimensional diffusion equation and converges as δx^2 for $\delta x \to 0$ [43]. Since according to relation (15) $\delta x^2 \sim \delta t$, the deterministic GRW has the same order of convergence with the time step as the weak Euler scheme of order $\beta = 1$. The convergence of the stochastic GRW simulation reaches the same order only if the number of random walkers N is large enough to smooth out the random fluctuations of $n(i,k)$ in (8-9). Figure 4 shows the dependence on N of the absolute error $eD(t) = |D_{grw}(t) - D|$. Figure 5 illustrates the convergence of the error norm $\|D_{grw} - D\|$ defined by

$$\|D_{grw} - D\|^2 = \sum_{k=1}^{T/\delta t} [D_{grw}(k\delta t) - D]^2 .$$

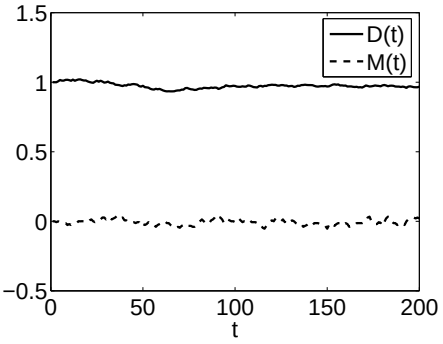

Figure 2. GRW estimates of the diffusion coefficient $D(t)$ and of the mean $M(t)$.

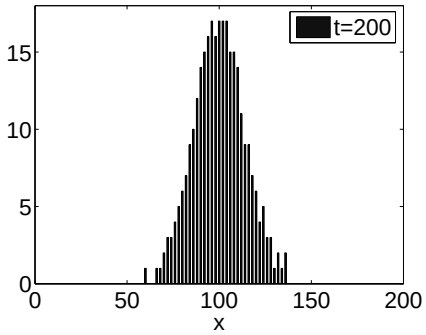

Figure 3. Distribution of $N = 300$ random walkers after 200 time steps.

Note that the GRW scheme described above is practically insensitive to the number of random walkers N. Assuming that all L grid points contain random walkers at all the computation time steps, one needs LT calls of a uniformly-distributed random-numbers generator for the entire simulation. Hence, the total computation time is of the order of that for the simulation of a single trajectory of the Itô process by the weak Euler scheme. Indeed, comparing LT with the computational costs of the order NT for a superposition of N PT simulations, we can see that the GRW algorithm achieves a speed-up of computations, with

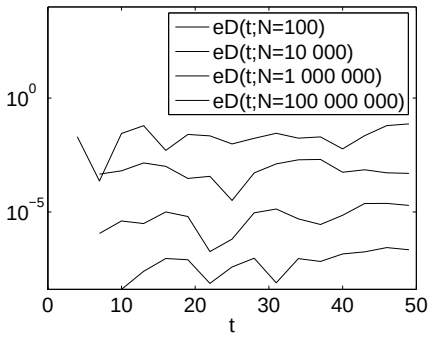

Figure 4. Absolute errors of the estimated diffusion coefficients for increasing numbers of particles N.

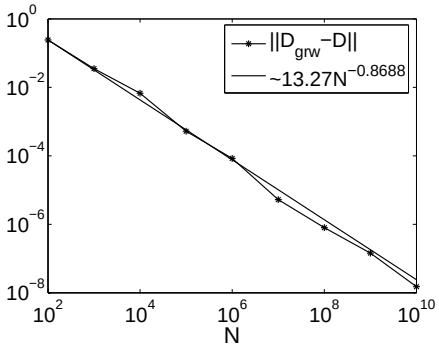

Figure 5. Convergence of the error norm of the estimated diffusion coefficients with the number of particles N.

respect to PT, of the order N/L. For example, while the convergence investigations with GRW presented in Figs. 4-5 were performed in about one second, similar investigations with the Euler scheme required several minutes on the same computer. In case of realistic simulations of diffusion processes, when very large numbers of particles should be considered, e.g. Avogadro's number $N = 10^{24}$, as well as large grids of the order of $L = 10^6$ nodes, a huge speed-up of computations by a factor of 10^{18} can be achieved by using the GRW algorithm.

2.4. Digression on numerical diffusion

The simplest way to estimate the numerical diffusion produced by a numerical scheme is to compare the diffusion coefficients computed from the corresponding numerical solution to their nominal values [25]. Such a test is presented in Fig. 2, which shows that the GRW algorithm accurately reproduces the diffusion coefficient of the Wiener process. In Section 2.3 we have shown that for genuine diffusion processes the GRW algorithm is equivalent with a FD scheme. Since FD schemes are not unconditionally free of numerical diffusion, one may be tempted to say that, by virtue of this equivalence, GRW also generates numerical diffusion (e.g. [9]). This assertion is not only wrong, but also in flagrant contradiction with the basic property of all RW methods, that of being strictly free of numerical diffusion.

Before proceeding with the discussion of the generic RW method we first note that the GRW algorithm (8-9), constrained by the relation (15), is equivalent to the FD scheme (14) with stability parameter $r = 1$. Since FD schemes with $r \leq 1$ are stable and free of numerical diffusion [5] GRW cannot generate numerical diffusion. GRW is also stable because, as shown by relation (16) for $m = 0$, the number of random walkers is conserved.

In general, a one-dimensional RW algorithm for genuine diffusion is a cumulative process consisting of a sum $X_k = \sum_{l=1}^{k} \delta X_l$ of independent increments δX_l, with $E(\delta X_l) = 0$ and $E(\delta X_l^2) = 2D\delta t$. The independence of increments implies

$$E(X_{k\delta t}^2) = \sum_{l=1}^{k} \delta X_l^2 = 2Dk\delta t, \tag{17}$$

hence the computed diffusion coefficient is $E(X^2_{k\delta t})/(2k\delta t) = D$, i.e., the RW scheme is unconditionally free of numerical diffusion. In particular, this is the case of strong and weak Euler schemes (4) and (6). Since GRW is a superposition of weak Euler schemes it is also unconditionally free of numerical diffusion. Note that the equivalence with a superposition of weak Euler schemes holds only if the amplitude of the increments is given by (7), i.e., if the time step, the space step, and the diffusion coefficient are related by (15). From (15) and (16) one obtains, for $m = 2$,

$$E(X^m_{k\delta t}) \approx \delta x^2 \sum_{i=1}^{L} i^2 \frac{n(i, k\delta t)}{N} = 2D\delta t \sum_{i=1}^{L} i^2 \frac{n(i, k\delta t)}{N}. \tag{18}$$

Equating (18) and (17) we find $\sum_{i=1}^{L} i^2 n(i, k\delta t)/N = k$. This relation is a consequence of the GRW rule for distributing particles over lattice sites. For instance, the GRW distribution of N particles, initially located at the origin of a symmetric one-dimensional lattice, yields $2(1^2 \cdot \frac{1}{2} + 0^2 \cdot \frac{1}{2}) = 1$ at $k = 1$, then $2(2^2 \cdot \frac{1}{4} + 1^2 \cdot 0 + 0^2 \cdot \frac{1}{2}) = 2$ at $k = 2$, etc.

Another approach for moving groups of particles, similar with GRW, was described by Delay et al. [7, 8]. In their approach, to the center of the square cells of a regular lattice one associates a variable that represents the number of particle which are then uniformly distributed, proportional with the area of the intersection of a "dispersion rectangle" with the neighboring cells. In the one-dimensional case the dispersion interval is the segment $[-\sqrt{2D\delta t}, \sqrt{2D\delta t}]$ [7], so that the particles are distributed according to a deterministic rule which, for genuine diffusion is identical to the GRW rule (12). Yet, this approach is different from GRW in that the space and time steps are not related with the diffusion coefficient, being instead adjusted by trial and error. As stated by the authors, their algorithm necessarily generates some numerical diffusion [7, 8, 9]. Since their approach uses the GRW distribution rule and a δx independent of D and δt, one generates an artificial diffusion which can be exactly quantified by (18) as $\delta x^2/(2k\delta t) - D$. Hence, the numerical diffusion can be completely removed in the one-dimensional algorithm of Delay et al. [7] by imposing the restriction (15) on space and time steps, which transforms their approach into a deterministic GRW.

Coarse-graining by the definition of the concentration field also may induce numerical diffusion. For instance, if we chose $l = 2$ to define the concentration with (13) we also have to chose $dx \approx 2\delta x$ to approximate the expectation and to obtain the correct form of (16). If $dx = \delta x$ were chosen, then all the moments of X_t would be divided by 2. Similarly, wrong results may be obtained if the concentration is defined as space average over more lattice sites. This issue, specific to continuous modeling of discrete systems, can be resolved by defining continuous fields as averages over symmetrical space-time "measurement" domains of the physical properties of the "microscopic" constituents of the system. The only requirement is that the particles move along piecewise analytical trajectories [46, 44]. This is for instance the case of a system of random walkers with piecewise constant velocities $\delta x/\delta t$ and $-\delta x/\delta t$. Using this approach to continuous modeling, we proved that there is a space-time continuum scale where the gradient of the concentration and the flux of random walkers are related by the Fick's law [45]. We have thus an independent proof that RW algorithms are free of numerical diffusion. Consequently, when numerical diffusion occurs in RW methods it can only be produced by inappropriate implementation options.

3. Unbiased GRW algorithms

3.1. Exact one-dimensional GRW algorithm

The one-dimensional GRW algorithm, which generalizes the algorithm (8-9) presented in Section 2.2 to account for advective displacements, describes the scattering of the $n(i,k)$ particles from (x_i, t_k) by

$$n(j,k) = \delta n(j+v_j, j, k) + \delta n(j+v_j-d, j, k) + \delta n(j+v_j+d, j, k), \tag{19}$$

where $v_j = V_j \delta t/\delta x$ are discrete displacements produced by the velocity field and d describes the diffusive jumps. The quantities $\delta n(j+v_j \pm d, j, k)$ in (19) are Bernoulli random variables describing the number of particles jumping to the left and to the right of the advected position $j+v_j$ and $\delta n(j+v_j, j, k)$ gives the number of particles which remain at the lattice site reached after an advective displacement. Unlike in the simple algorithm from Section 2.2, now only the group of particles $\delta n(j+v_j-d, j, k)+\delta n(j+v_j+d, j, k)$ evolves like a superposition of weak Euler schemes, and only over a time step. If $\delta n(j+v_j, j, k) > 0$ at all times, then the even and odd modes are mixed and the particles distribution shown in Fig. 3 will be smoothed out. The distribution of the particles at the next time $(k+1)\delta t$ is given by

$$n(i,k+1) = \delta n(i,i,k) + \sum_{j \neq i} \delta n(i,j,k). \tag{20}$$

As follows from (19), the contributions $\delta n(i,j,k)$, $j \neq i$ come from all lattice sites and $\delta n(i,i,k)$ from those satisfying $j + v_j = i$.

The average number of particles undergoing diffusive jumps and the average number of particles remaining at the same node after the displacement v_j are given by the relations

$$\overline{\delta n(j+v_j \pm d, j, k)} = \frac{1}{2} r\, \overline{n(j,k)}, \tag{21}$$

$$\overline{\delta n(j, j+v_j, k)} = (1-r)\, \overline{n(j,k)}, \tag{22}$$

where $0 \leq r \leq 1$ is a rational number. A consistency requirement is that, for the same time step δt, the algorithm described by (19-22) reproduces the mean square displacement given by the algorithm (8-9), which is strictly equivalent with a superposition of weak Euler schemes. If $\delta x'$ is the space step used in (8-9), then the mean square displacement of the particles jumping from j to first-neighbor sites is $\overline{n(j,k)}\delta x'^2$. In the new algorithm, where in the mean only a fraction r of the particles reaching $j+v_j$ undergo jumps of amplitude $d\delta x$, using (21) one obtains $r\overline{n(j,k)}(d\delta x)^2$. Equating the two expressions of the mean square displacement and using (15) one obtains the following relation between the parameters of the GRW algorithm

$$D = r\frac{(d\delta x)^2}{2\delta t}, \tag{23}$$

which generalizes (15) and ensures that the scheme does not produce numerical diffusion. For given δx and δt, using (23) one obtains combinations of parameters d and r for any possible value of the diffusion coefficient D.

Particularizing the above one-dimensional GRW algorithm for genuine diffusion, i.e., letting $v_j = 0$ in (19), one can easily see that the evolution of the mean number of particles is described by

$$\overline{n(i,k+1)} = \frac{r}{2}\overline{n(i+d,k)} + (1-r)\overline{n(i,k)} + \frac{r}{2}\overline{n(i-d,k)}. \tag{24}$$

According to (13), $\overline{n(i,k)}$ is proportional with the concentration $c(i,k)$. Hence, the relation (24) has the form of the explicit FD scheme for the one-dimensional diffusion equation $\partial c = D\partial_x^2 c$. Since from (23) we have $\delta t = \mathcal{O}(\delta x^2)$, the FD scheme (24) is a consistent approximation of the partial differential diffusion equation and, by the condition $r \leq 1$ (equivalent with von Neumann's criterion), it is also stable. The stability and consistency imply the convergence of the order $\mathcal{O}(\delta x^2)$ to the exact solution of the initial value problem for the diffusion equation [14]. Equation (14), corresponding to the algorithm presented in Section 2.2, is the particular case of (24) for $r = 1$.

The exact GRW algorithm is implemented by specifying the procedure to calculate the fluctuating quantities in the right hand side of (19). The fraction r of the number of particles undergoing jumps to neighboring nodes must be chosen as a positive rational number $r \leq 1$, such that $(1-r)N$ is an integer equal to the total number of particles remaining at the same site after an advective step. The latter are calculated recursively, at each site j as follow. We proceed by increasing the index j and if j_1 is the first site with $n(j_1,k) > 0$, then $\delta n(j_1 + v_{j_1}, j_1, k) = [(1-r)n(j_1,k)]$, where $[\cdot]$ is the integer part of the expression in the brackets. Further, we compute and store the rest $R_{j_1} = (1-r)n(j_1,k) - \delta n(j_1+v_{j_1}, j_1, k)$. At the next site containing particles, $j_2 > j_1$, we compute $\delta n(j_2 + v_{j_2}, j_2, k) = [(1-r)n(j_2,k) + R_{j_1}]$. Repeating these operations for increasing index j, we obtain

$$\delta n(j+v_j, j, k) = \left[(1-r)\sum_{j'\leq j} n(j',k)\right] - \left[(1-r)\sum_{j'<j} n(j',k)\right]. \tag{25}$$

Averaging (25) over a large number of GRW simulations one obtains the relation (22) for the mean number of particles. Since $\delta n(j,j,k)$ is known, (19) 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.

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, $0 \leq m \leq n$, 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 $F_n(m) = \sum_{l=0}^{m} b_n(l)$, $0 < F_n(m) \leq 1$, the Bernoulli repartition, then $\delta n(j+v_j-d,j,k)$ takes the value m satisfying the condition $F_n(m-1) \leq \eta < F_n(m)$, where we use the convention $F_n(-1) = 0$.

The implementation of the GRW algorithm as a computer code encounters some difficulties 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 $1 \leq k \leq 20$. Due to the symmetry of $F_n(m)$ with respect to $m = n/2$, only the values $F_n(m) \leq 0.5$ 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 $n \geq 2^{21}$, 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 [42]), becomes time expensive for very large n (see the "GRW0" curve in Fig. 6). Therefore, to increase the efficiency of the GRW algorithm, we introduce an approximation based on reduced variables $\xi = (m - n/2)/\sqrt{n/4}$ and the corresponding repartitions $F_n(\xi)$. If $n \geq 2^{21}$ we approximate $F_n(\xi)$ by the repartition corresponding to $n = 2^{20}$ as function of the reduced variable ξ. We found that the results obtained in this way are fully satisfactory. 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. We also note that an even simpler approximation uses the fact that for $n \to \infty$, the repartition $F_n(\xi)$ tends to the normal Gaussian repartition, according to De Moivre-Laplace theorem [23]. Then, the number of particles jumping to the left, $\delta n_l(j + v_j - d, j, k)$, can be readily determined by using the error-function the instead repartition (see GRW-erf curve in Fig. 7). This procedure is a bit faster if $n < 10^{12}$ but attenuates the fluctuations of the number of particles.

In case of constant diffusion coefficients, two- and three-dimensional GRW algorithms can still be implemented by performing the 1-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. But for variable coefficients this extension of the one-dimensional procedure doesn't work and the two-dimensional algorithm described below in Section 3.2 should be used.

One can also 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 (21) we introduce

$$\delta n(j + v_j \pm d, j, k) = \frac{1}{2} r\, n(j, k), \tag{26}$$

and in analogy with (22) we consider

$$\delta n(j, j, k) = (1 - r)n(j, k) \tag{27}$$

Then (19) is identical satisfied and all the quantities in (20) 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 (26) and coincides with the mean value of the corresponding random variable in GRW. Therefore we refer to this modified algorithm, which generalizes the deterministic algorithm without advection and without parameters d and r, 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. For this purpose, we use (20) and instead of (26) we introduce

$$\delta n(j+v_j-d,j,k)=\begin{cases} n/2 & \text{if } n \text{ is even} \\ [n/2]+\theta & \text{if } n \text{ is odd,} \end{cases} \tag{28}$$

where $n=n(j,k)-\delta n(j+v_j,j,k)$, $[n/2]$ is the integer part of $n/2$ and θ is a random variable taking the values 0 and 1 with probability $1/2$. Further, the number of particles jumping in the opposite direction, $\delta n(j+v_j+d,j,k)$, is determined by (19). In comparison with GRW, this algorithm reduces the fluctuations of the number of particles to those of a sigle random walker and we call it "reduced fluctuations" GRW (GRWR). Since the fluctuations do not vanish, only the average of the GRWR solution is identical with the FD solution. Solutions for the advection-diffusion equation (1) can be obtained with either GRWD, GRWR, or with the stochastic algorithm GRW. The latter is expected to be more accurate when the fluctuations significantly influence the simulated process [16, 10, 17]. Instead, in large scale, advection-dominated transport problems where the fluctuations of the velocity field are more important than the fluctuations of the number of particles, GRWD and GRWR are more efficient. In this case, the use of GRWR requires reduced computing resources with respect to either GRW and GRWD algorithms. Unlike in the exact GRW algorithm, in GRWR only one random number has to be generated, at every time step and only when at a given site there is an odd number of particles. Because the indivisibility of particles is preserved, the diffusion front has a smaller extension than in the case of GRWD algorithm and, consequently, smaller grids are necessary. An efficient implementation of GRWR can be obtained if instead of using (28) and generating random numbers one redistributes the rests R_j, appearing in the computation of $\delta n(j+v_j,j,k)$ and the rests $R_{j+v_j}=n/2-[n/2]$ of the division by 2 of the number of jumping particles, and, similarly to (25), one calculates $\delta n(j+v_j-d,j,k)=[n/2+R_j+R_{j+v_j}]$. To ensure the strict conservation of the number of particles, besides the condition requiring that $(1-r)N$ be an integer, related to the use of parameter r, N should also be a power of 2. Nevertheless, if $N\geq 10^8$ the possible truncation errors are negligible and these precautions are no longer necessary.

The GRW algorithms are much more faster and accurate than the sequential PT procedure. 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 and post-processed to obtain concentrations. Moreover, the large numbers of particles necessary to have good concentration estimations may be prohibitive for PT simulations. 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 processes. 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 [22]). The same problem was solved on a Cray T3E parallel computer [43]. In Fig. 6 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

increase 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. 6 corresponds to GRW0 algorithm described above, where no approximations of binary repartitions are used. 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. The comparison from In Fig. 7, done for the same problem, shows that for more than 10^{12} particles the GRW algorithm and its approximation using the error function (GRW-erf) need almost the same cpu-time, which is twice the cpu-time for the deterministic algorithm GRWD. The reduced fluctuations algorithm GRWR (not shown in the picture) needs practically the same cpu-time as GRWD but, for larger total simulation times GRWR is faster because, as noted above, it preserves the indivisibility of particles and the diffusion fronts have smaller spatial extension.

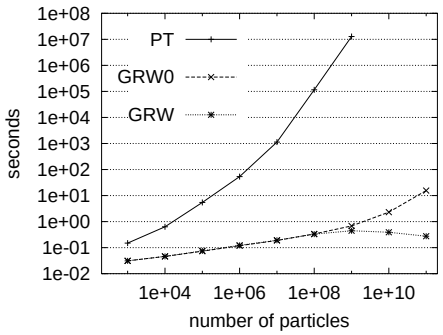

Figure 6. CPU times for PT, the exact GRW algorithm, denoted by GRW0, and the reduced fluctuations GRW. The comparison was done for simulations over ten time-steps of the three-dimensional Gaussian diffusion with constant coefficient.

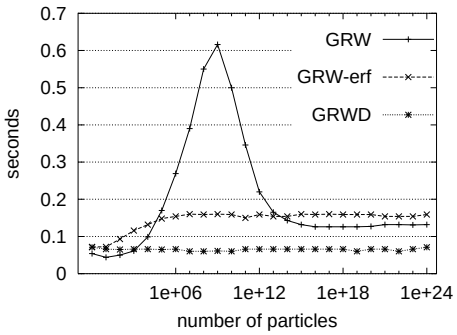

Figure 7. CPU times for the GRW algorithm that uses the Bernoulli repartition for $n = 10^{20}$ particles to approximate repartitions for larger numbers of particles, the error-function approximation GRW-erf and for the deterministic algorithm GRWD.

The GRW algorithm and its modified forms GRWD and GRWR use the relation (20) where $\delta n(i, j, k)$ is non-vanishing 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 (24), by contributions from more than the first neighboring nodes. The terms in (19) 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. All these GRW versions remain equivalent with a weak Euler scheme, but only over single time steps, because of the new parameter r. Therefore for space-variable drift $V(x)$ the convergence properties cannot be inferred theoretically, as for the simple GRW from Section 2.2. However, evaluations of the unbiased GRW algorithm can be done by comparisons with the biased algorithm that will be presented in Section 4.1.

3.2. Simulation of the one-dimensional Gaussian diffusion

We verify the GRW algorithm described in the previous section for the solution of the equation (1), in one dimension, with $V(x) \equiv 0$, $D(x) \equiv 0.5$ and initial condition $\lim_{t \longrightarrow 0+} c(x, t) = N_0 \delta(x)$, where $\delta(x)$ is the Dirac function. In this case the solution

has a Gaussian analytic form and is given by

$$c_{Gauss}(x,t) = N_0(2\pi t)^{-1/2}\exp\left\{\frac{-x^2}{2t}\right\}. \tag{29}$$

Different numerical solutions obtained by GRW are quantitatively compared with the analytical solution (29) 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

$$\frac{1}{N_0}\int_{-1}^{1} c_{Gauss}(x,t_f)\,dx = \text{erf}\left(\frac{1}{\sqrt{2t_f}}\right) = 0.99,$$

we have $t_f = 0.15$.

The numerical solution is obtained using the GRW, GRWD and GRWR algorithms, for $v_i = 0$ and $d = 1$. Initially N 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 $g = 1, 2, ..., 10$. Since $D = 0.5$, from (23) it follows that the corresponding time steps are $\delta t = r\,\delta x^2 = r\,(10g)^{-2}$ and the numerical simulation contains $k_f = t_f/\delta t = 15g^2/r$ time steps. To eliminate the boundaries effect on the numerical solution we must choose a large enough grid so that no particles 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.

We want to compare c defined by (13) with the analytic solution (29), at time t_f and over the spatial interval $[-1,1]$. For $r = 1$, because of the singular initial condition , there are independent even and odd modes and we have to chose $l = 2$ in the concentration definition (13), 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

$$i_f = \begin{cases} 10g & \text{if } k_f \text{ is even} \\ 10g-1 & \text{if } k_f \text{ is odd} \end{cases},$$

such that $x_f = \pm i_f\,\delta x$ 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 $\|c - c_{Gauss}\|$, defined as

$$\|c - c_{Gauss}\|^2 = \frac{1}{i_f+1}\sum_{i=0}^{i_f}\left[\frac{1}{N}c((2i-i_f)\delta x, t_f) - \frac{1}{N_0}c_{Gauss}((2i-i_f)\delta x, t_f)\right]^2, \tag{30}$$

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 (30) with $i_f = 10g$, summing over all $(2i_f + 1)$ nodes of the grid and with $c = \bar{n}/\delta x$, defined by (13) for $l = 1$.

The analysis of the GRWD algorithm, equivalent to a FD scheme, indicates a linear behavior of the norm $\|c_{FD} - c_{Gauss}\|$ as function of δx^2 for several values of the parameter

r, hence the convergence of order δx^2 of the FD scheme. The same investigation shows that the maximum precision of the finite difference scheme is obtained for $r \approx 0.3$ [43].

In the following we fix the parameter $r = 1$. By $c(x_i, t_k)$ we denote the numerical solution 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 particles N, and the number of simulations used to compute the mean number of particles $\bar{n}$ which, introduced in (13), give the concentration estimation. To investigate the convergence of as function of δx and N, we consider here single GRW simulations, for which $\bar{n} = n$ and, because $r = 1$, from (13) we have $c = n/(2\delta x)$. We computed the norm (29) for the numerical solution for increasing N obtained with GRW and GRWR algorithms for $\delta x = 0.1$ (Fig. 8) and $\delta x = 0.01$ (Fig. 9). The results shown in Figs. 8 and 9 indicate that 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.

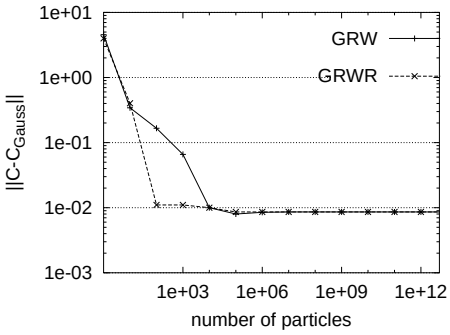

Figure 8. Convergence of GRW and reduced fluctuations algorithm GRWR towards the analytical Gaussian solution as function of number of particles for $\delta x = 0.1$.

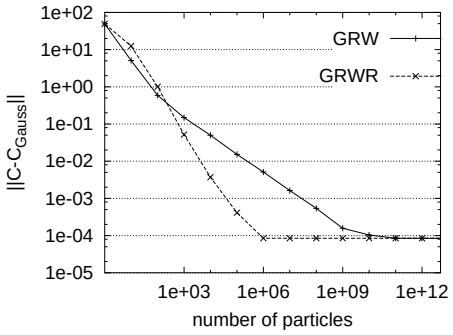

Figure 9. Convergence of GRW and reduced fluctuations algorithm GRWR towards the analytical Gaussian solution as function of number of particles for $\delta x = 0.01$.

We also remark the decrease $\sim 1/\sqrt{N}$ of the norm in case of GRW solutions. Thus this numerical investigations suggests a converges to the analytical solution as $\mathcal{O}(\delta x^2)$ $+\mathcal{O}(1/\sqrt{N})$, at moderate N, and a convergence of the order $\mathcal{O}(\delta x^2)$ as for the finite differences scheme, for larger N, when the condition $1/\sqrt{N} = \mathcal{O}(\delta x^2)$ is met. The GRWR curves from Figs. 8 and 9 indicate a faster convergence with N, as already suggested by the results from Fig. 5.

Since GRW algorithm can use large numbers of particles, it is "self-averaging", in the sense that Monte Carlo repetitions are not necessary and the required precision can be achieved in a single GRW simulation. This is the essential difference with respect to the sequential PT procedures, which may be thought as "analogical Monte Carlo method" where the solution of the diffusion equation is obtained by averages over individual trajectories of individual particles (e.g. [19]).

3.3. 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,

$$J(i+1/2,k+1/2) = \frac{1}{\delta t}\left[\delta n(i+1,i,k) - \delta n(i,i+1,k)\right] . \tag{31}$$

For the GRWD algorithm, using (26) and (13) with $l = 1$, the relation (31) becomes

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

From (23) it follows that this is the usual FD form of the Fick's law

$$J(x,t) = -D\partial_x c(x,t)\,. \tag{32}$$

Let us consider a finite grid with $2L+1$ nodes, $i = -L, -L+1, ..., L$. 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

$$J(L+1/2,k+1/2) = \frac{1}{\delta t}\left[\delta n(L+1,L,k) - \delta n(L,L+1,k)\right] \tag{33}$$

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 (31) would have the requested value. Consider a von Neumann boundary condition

$$J(L+1/2,k+1/2) = J_b(k+1/2)\,, \tag{34}$$

where J_b is a given function of time. From (33) it follows the boundary condition

$$\delta n(L,L+1,k) = \delta n(L+1,L,k) - \delta t J_b(k+1/2). \tag{35}$$

For $J_b(k+1/2) = 0$, from (35) 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 (35) mixes the numerical modes. The mixing of the numerical modes can be avoided by summing up (35) over 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,

$$\delta n(L, L+1, k) = 0\,, \tag{36}$$

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 (31) and (19) we obtain the boundary condition

$$\delta n(L, L+1, k) = n(L, k) - \delta n(L, L, k) - \delta n(L, L-1, k). \tag{37}$$

Applied to GRWD (which is equivalent to the FD algorithm), this condition becomes, by means of (26) and (27), the "transmission boundary condition" [18]

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

If we perform a numerical simulation on a finite grid of a nonstationary diffusion process in an unbounded domain after the particles reach 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. From the one-dimensional version of (1) with $V(x) = 0$,

$$\partial_t c = D\partial_x^2 c.$$

one obtains

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

where we used the relation $\delta t = \mathcal{O}(\delta x^2)$ derived from (23). Then we have

$$\partial_t c(x,t) = \partial_t c(x-\delta x, t-\delta t) + \mathcal{O}(\delta x). \tag{38}$$

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

$$\partial_t c(x,t) = 2\partial_t c(x-\delta x, t-\delta t) - \partial_t c(x-2\delta x, t-\delta t) + \mathcal{O}(\delta x^2). \tag{39}$$

For $x = L\,\delta x$ and $t = (k+1)\,\delta t$, these relations written in finite difference, using (13), give the nonstationary boundary condition

$$n(L, k+1) = n(L-1, k+1) - n(L-1, k) + n(L, k) \tag{40}$$

and

$$\begin{aligned} n(L, k+1) = 2n(L-1, k+1) - 2n(L-1, k) - n(L-2, k+1) + \\ n(L-2, k) + n(L, k)\,. \end{aligned} \tag{41}$$

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 (40) and (41) 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 (35) with $J_b = 0$.

As an illustration of these boundary conditions we continue the simulation of the previous section for a temporal interval three times larger than t_f . We use the GRWD algorithm defined by (26) and (27) with the parameter $r = 0.3$, In these simulations $\delta x = 0.1$ and the computational interval contains the spatial interval $[-1, 1]$. In Fig. 10 we represent the time evolution of the boundary concentration for four different boundary conditions. The points BC1 correspond to the impermeable wall boundary condition given by (35) with $J_b = 0$, BC2 and BC3 to the nonstationary boundary conditions (40) and (41), and BC4 to the absorbing condition (36). The nonstationary condition (40) do not improve the accuracy with respect to (35) and (36). We have found that the condition BC3 given by (41) keeps the norm (30) at values smaller than 0.02, which proves that it is suitable to be used in simulation of nonstationary diffusion. The same conclusion can be drawn from the spatial variation of the concentration at the final time shown in Fig. 11.

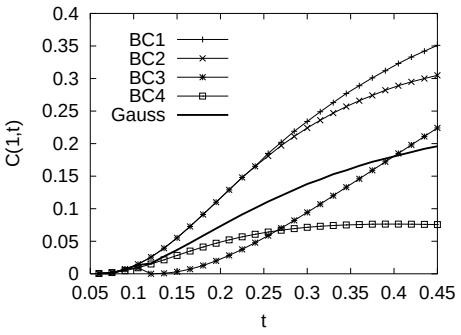

Figure 10. Concentrations at the right boundary as function of time for different boundary conditions.

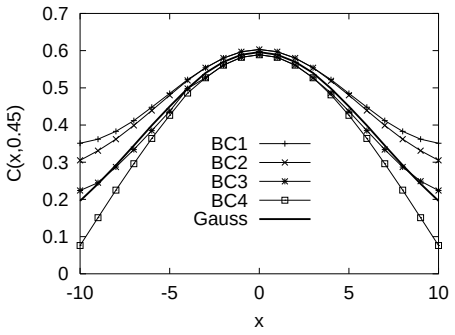

Figure 11. Spatial concentration profiles at the final simulation time t_f for different boundary conditions.

3.4. Two-dimensional GRW algorithms

As already mentioned in Section 3.1, for constant diffusion coefficients the two and three-dimensional algorithms can be simply built by repeating the one-dimensional procedure for all space directions. Figure 12 illustrates such a two-dimensional GRW algorithm where, after the advective step the particles execute jumps on x_2-direction, then jumps on x_1-direction, both according to the one-dimensional rule (19). When the diffusion coefficients vary in space two different parameters r are needed to describe the ratios of particles undergoing jumps (see Section 3.5 below) and the GRW algorithm follows the rules illustrated in Fig. 13 [33].

The two-dimensional algorithm from Fig. 12 is mainly useful in simulations of advective-diffusive transport, described by (1), in heterogeneous media characterized by

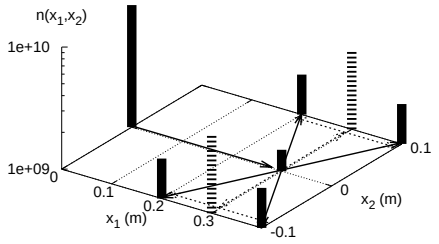

Figure 12. Two-dimensional GRW algorithm as a superposition of one-dimensional GRW procedures.

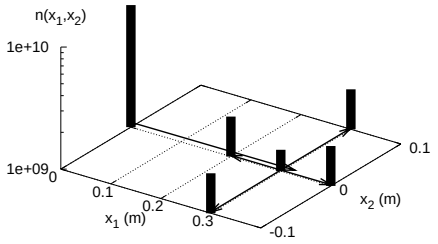

Figure 13. Two-dimensional GRW algorithm for independent longitudinal and transverse random walks.

highly variable velocity fields dominating the diffusion process, which therefore can be sufficiently well approximated as isotropic and characterized by a single diffusion coefficient [28, 30, 31, 33, 34, 36, 37]. The values of the normalized concentration at a given time $t = k\delta t$ and at a lattice site $(x_1, x_2) = (i_1\delta x_1, i_2\delta x_2)$, where δt is the time step and δx_1, and δx_2 the space steps, can be estimated from the particles distribution produced by GRW simulations with the formula

$$c(x_1, x_2, t) = \frac{1}{N}\frac{1}{4\delta x_1 \delta x_2}\sum_{i_1'=i_1-1}^{i_1+1}\sum_{i_2'=i_2-1}^{i_2+1} n(i_1', i_2', k\delta t). \tag{42}$$

Figure 14 shows the concentration field at successive times given by (42) for a GRW simulation of isotropic diffusion in a velocity field generated as a sample of a random space function (see Section 5.1), in case of a singular initial distribution of particles. Figure 15 shows concentration fields for a non-singular initial condition.

Let us analyze in more detail the particles distribution in case of the initial condition consisting of N particles uniformly distributed over N_{X_0} grid sites. By $n(i_1, i_2, k; x_{0,1}, x_{0,2})$ we denote the distribution of particles at the time step k given by the GRW procedure for a diffusion process staring at $(x_{0,1}, x_{0,2})$. Since the distribution of the particles at time k can be written as

$$n(i_1, i_2, k) = \sum_{x_{0,1}, x_{0,2}} n(i_1, i_2, k; x_{0,1}, x_{0,2}),$$

it follows that

$$\begin{aligned}&\frac{1}{N}\sum_{i_1,i_2} n(i_1, i_2, k)\\ &= \frac{1}{N_{X_0}}\sum_{x_{0,1},x_{0,2}} \frac{N_{X_0}}{N}\sum_{i_1,i_2} n(i_1, i_2, k; x_{0,1}, x_{0,2}).\end{aligned} \tag{43}$$

Thus, according to (43), the concentration (42), as well as its spatial moments, are averages over the trajectories of the diffusion process starting at given initial positions and over the

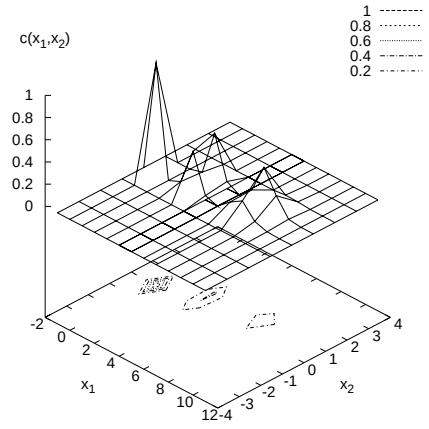

Figure 14. The concentration field, at $t = 0$, $t = 5\delta t$ and $t = 10\delta t$, for a singular initial distribution of 10^{10} particles released at the origin of the lattice.

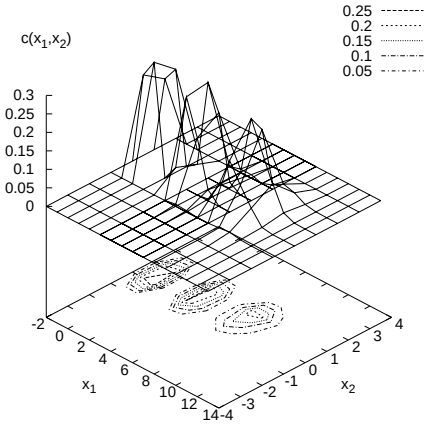

Figure 15. The concentration field at the same times as in Fig. 14, for a uniform initial distribution of 10^{10} particles in the square of sides δx_1 and δx_2.

distribution of the initial positions. The ability of the GRW to distinguish between the two kind of averages has useful applications in investigations on dependence on initial conditions and memory effects of diffusion in random velocity fields [34, 36, 37].

3.5. GRW algorithm for two-dimensional diffusion in non-homogeneous and anisotropic media

In the following we consider a two-dimensional diffusion process described by the diagonal diffusion tensor

$$\left\| \begin{array}{cc} D_x(x,y) & 0 \\ 0 & D_y(x,y) \end{array} \right\|$$

and the diffusion equation

$$\partial_t c = \partial_x(D_x \partial_x c) + \partial_y(D_y \partial_y c).$$

The GRW solution of the diffusion equation with spatially dependent diffusion coefficients should use correction terms associated with the drift term $(-\partial_x D_x, -\partial_y D_y)$ [32]. However, for the purpose of illustration, we consider here only the particular case of slowly variable diffusion coefficients, i.e., $\partial_x D_x \approx 0$ and $\partial_y D_y \approx 0$, for which the diffusion equation can be approximated by

$$\partial_t c = D_x \partial_x^2 c + D_y \partial_y^2 c. \tag{44}$$

The equation (44) has no drift terms and the diffusion coefficients vary in space. In this case, simulations using the superposition of one-dimensional GRW rules (19) illustrated in Fig. 12 are no longer accurate. Therefore, we use the new algorithm from Fig. 13, where

the particles lying at a lattice site are globally spread according to the rule

$$\begin{aligned} n(i,j,k) &= \delta n(i,j \mid i,j,k) \\ &+ \delta n(i-d_x,j \mid i,j,k) + \delta n(i+d_x,j \mid i,j,k) \\ &+ \delta n(i,j-d_y \mid i,j,k) + \delta n(i,j+d_y \mid i,j,k) \end{aligned} \tag{45}$$

where $n(i,j,k)$ is the number of particles at the site $(x_i,y_j)=(i\delta x,j\delta y)$ at the time $k\delta t$. Unlike the two-dimensional algorithm presented in Fig. 12, where the one-dimensional procedure is applied successively for longitudinal and transversal directions, the procedure (45) moves the particles simultaneously along the principal directions of the diffusion tensor as illustrated in Fig. 13 (for the general case of a non vanishing drift).

The anisotropy is taken into account by two different parameters d_x and d_y which describe the diffusive jumps along the coordinates axes. The spatial variation of the diffusion coefficients is described through the variable parameters

$$r_x=\frac{2D_x\delta t}{(d_x\delta x)^2},\, r_y=\frac{2D_y\delta t}{(d_y\delta y)^2}. \tag{46}$$

After the global change of state of the lattice by the procedure (45) applied at every site containing particles, the new numbers of particles at sites are obtained similarly to (20) by summation over two spatial indices.

The average over an ensemble of simulations of the terms in (45) are related by

$$\begin{aligned} \overline{\delta n(i\pm d_x \mid j,k)} &= \frac{1}{2}r_x(i,j)\,\overline{n(i,j,k)}, \\ \overline{\delta n(i \mid j\pm d_y,k)} &= \frac{1}{2}r_y(i,j)\,\overline{n(i,j,k)}, \\ \overline{\delta n(i,j \mid i,j,k)} &= [1-r_x(i,j)-r_y(i,j)]\,\overline{n(j,k)}. \end{aligned}$$

These relations can be used to show that for slowly variable diffusion coefficients this two-dimensional GRW algorithm approximates the finite difference scheme for the diffusion equation (44).

The lattice steps δx and δy are chosen accordingly to the desired resolution of the concentration field. The time step δt is inferred from the condition $r_x+r_y\leq 1$, which states that the numbers of diffusing particles are limited by the numbers of particles at the lattice sites. Using (46) one obtains

$$\delta t\leq\left[\frac{2D_x^{\max}}{(d_x\delta x)^2}+\frac{2D_y^{\max}}{(d_y\delta y)^2}\right]^{-1}, \tag{47}$$

where $D_x^{\max}$ and $D_y^{\max}$ are the upper bounds of the diffusion coefficients $D_x(x,y)$ and $D_y(x,y)$.

The two dimensional algorithm (45) has been used to simulate the diffusion through the human skin [33]. A two dimensional geometry with the x-axis parallel and the y-axis perpendicular to the surface of the skin was used. A thin film consisting of $N=10^{20}$ water molecules with $\Delta x=10\ mm$ was considered to be applied on the surface of the skin. Since the skin structure is stratified, the diffusion can be described by the two-dimensional

equation (44). To describe the diffusion in the horizontal direction the lattice dimension on the x axis was $3\Delta x$. An acceptable resolution was obtained with $\delta x = 0.1\ mm$, (300 nodes per horizontal lattice length). A simplified two layers structure of the skin was considered, with thickness $y_1 = 0.1\ mm$ and $y_2 = 0.5\ mm$ respectively. With a resolution of $\delta y = 0.01\ mm$ the lattice extended over 10 nodes in the first layer and over 50 nodes in the second.

Because of the nonhomogeneous structure of the skin the diffusion coefficients show spatial fluctuations about the mean value. At every lattice site the coefficients D_x and D_y were generated as normal random variables with the mean value equal to half the maximum values $D_x^{\max}$ and $D_y^{\max}$ and variance equal to a fraction $p = 0.1$ of the corresponding maximum. Since the cells are rather flat, the coefficient $D_x^{\max} = 2D_y^{\max} = 5.8810^{-7}\ m2/s$ was considered for the first layer. In the second layer an isotropic diffusion coefficient ten times larger than in the superior layer was chosen. Between the layers a transition zone of thickness $3\delta y$ was placed, where diffusion coefficients vary linearly [33].

Two different boundary conditions were used. At the surface of the skin, the molecules which jump in the exterior of the domain were blocked on the boundary. At the inferior boundary a "transmission boundary condition" in the first order of approximation (41) was imposed. This last condition has the property to not disturb the diffusion front at the boundary. The time step was chosen using the condition (47). Some simulation results are presented in Figs. 16 and 17. The high resolution GRW simulations were further used to estimate probability distributions of water molecules and fluxes through the superior and inferior layers of the skin model [33].

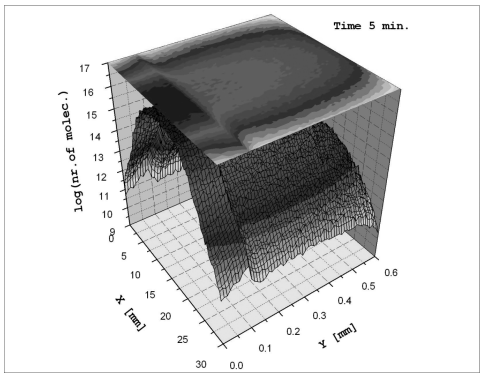

Figure 16. The distribution of the water molecules in the skin after 5 minutes.

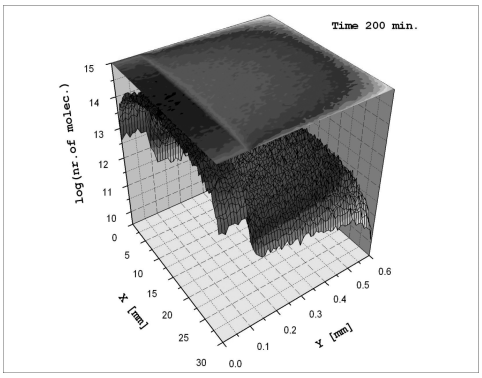

Figure 17. The distribution of the water molecules in the skin after 200 minutes.

4. Two dimensional biased-GRW algorithm

4.1. The BGRW algorithm

The GRW algorithm was introduced in Section 2.2 as a superposition of many PT procedures, implemented as weak Euler schemes for the Itô equation (2). Further, in Section 3.1, GRW was generalized by allowing groups of particles to be captured at a lattice site after the advection step, so that it remains equivalent with the Weak Euler scheme (i.e., it moves particles by advection displacements followed by unbiased random walk jumps) only for a

time step. Both GRW algorithms were shown to be also equivalent, in case of vanishing or constant drift coefficients, with the FD scheme for the diffusion equation. Since the global spreading of the particles is given by rules which update the state of the lattice sites, e.g. (8-9), (19-20), (45), GRW is a particular cellular automaton (CA), i.e., it is a stochastic process in the space of configurations, defined at a given time by the occupation numbers at each lattice site [29]. What makes the GRW algorithm different from a typical CA approach [17] is that the number of particles per grid site is not limited by an "exclusion principle" and there are no limitations for the total number of particles. Therefore, as shown in Figs. 8-9, GRW is "self-averaging" in the sense that the solution given by a single simulation is practically the same as that obtained after averaging over large ensembles of simulations. By working with integers, GRW is free of round-of errors, avoids numerical diffusion and it is inherently stable (see Sections 2.2 and 3.1). However, for variable drift and diffusion coefficients overshooting errors occur when the particles jump over more than one lattice site (see Fig. 18). This is mainly the case of diffusion in space variable velocity fields, when the velocity at the sites between the initial and final position of the group of particles during the advection step may have sharp variations. Since by playing only with the parameters δx, δt, d, and r of the unbiased GRW algorithm it is very difficult to reduce the overshooting errors [28], a better solution is to modify the CA rule of moving groups of particles.

To get rid of overshooting errors, we impose that particles jump only to the nearest sites (Fig. 19). In this procedure the advection will be simulated by a bias in the random walk jumps. Therefore, we call it "biased global random walk" (BGRW) algorithm. Since BGRW moves all the particles lying at a lattice site in a single numerical procedure, N can be as large as necessary to ensure the self-averaging, which is the main difference with respect to other CA for diffusion without exclusion principle (e.g. [17]).

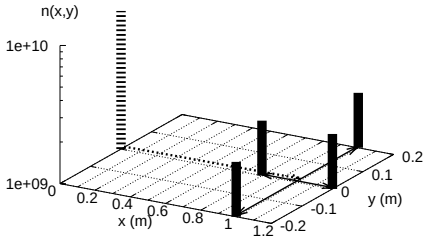

Figure 18. Change of state in unbiased GRW algorithm over a time step $\delta t = 0.5$.

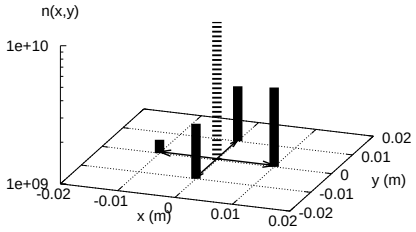

Figure 19. Change of state in BGRW algorithm over a time step $\delta t = 0.0025$.

The 2-dimensional BGRW is defined by the CA rule

$$\begin{aligned} n(i,j,k) &= \delta n(i,j \mid i,j,k) \\ &+ \delta n(i+1,j \mid i,j,k) + \delta n(i-1,j \mid i,j,k) \\ &+ \delta n(i,j+1 \mid i,j,k) + \delta n(i,j-1 \mid i,j,k), \end{aligned} \tag{48}$$

where $n(i,j,k)$ is the number of particles at the site $(x,y) = (i\delta x, j\delta y)$ at the time $t = k\delta t$. Corresponding to the components of the drift (velocity) and diffusion coefficients of

the transport problem, $V_x(x,y,t)$, $V_y(x,y,t)$, $D_x(x,y,t)$ and $D_y(x,y,t)$, we define the dimensionless parameters

$$v_x = V_x\frac{\delta t}{\delta x},\ v_y = V_y\frac{\delta t}{\delta y},\ r_x = D_x\frac{2\delta t}{\delta x^2},\ r_y = 2D_y\frac{2\delta t}{\delta y^2}. \tag{49}$$

The averages of the terms in (48) over an ensemble of simulations are related by

$$\begin{aligned}
&\overline{\delta n(i,j \mid i,j,k)} = (1-r_x-r_y)\,\overline{n(i,j,k)},\\
&\overline{\delta n(i\pm 1,j \mid i,j,k)} = \frac{1}{2}(r_x \pm v_x)\overline{n(i,j,k)},\\
&\overline{\delta n(i,j\pm 1 \mid i,j,k)} = \frac{1}{2}(r_y \pm v_y)\overline{n(i,j,k)}.
\end{aligned} \tag{50}$$

The last four terms in (48) are Bernoulli random variables which, for the simulations presented in the following, were approximated as in the reduced fluctuations algorithm GRWR presented in Section 3.1.

Defining the particle density $\rho(x,y,t) = \overline{n}(i,j,k)$ and summing the contributions from the first neighbors to a lattice site, from (48-50) one obtains

$$\begin{aligned}
&\frac{\rho(x,y,t+\delta t)-\rho(x,y,t)}{\delta t} +\\
&\frac{V_x\rho(x+\delta x,y,t)-V_x\rho(x-\delta x,y,t)}{2\delta x} + \frac{V_y\rho(x,y+\delta y,t)-V_y\rho(x,y-\delta y,t)}{2\delta y} =\\
&\frac{D_x\rho(x+\delta x,y,t)-2D_x\rho(x,y,t)+D_x\rho(x-\delta x,y,t)}{\delta x^2} +\\
&\frac{D_y\rho(x,y+\delta y,t)-2D_y\rho(x,y,t)+D_y\rho(x,y-\delta y,t)}{\delta y^2},
\end{aligned} \tag{51}$$

which is just the forward in time centered in space FD scheme for the Fokker-Plank equation

$$\partial_t\rho + \partial_x(V_x\rho) + \partial_y(V_y\rho) = \partial_x^2(D_x\rho) + \partial_y^2(D_y\rho). \tag{52}$$

The advection-diffusion equation which corresponds to Fick's law

$$\partial_t\rho + \partial_x(V_x^*\rho) + \partial_y(V_y^*\rho) = \partial_x D_x\partial_x\rho + \partial_y D_y\partial_y\rho,$$

is equivalent to (52) if the new drift coefficients are given by the relations $V_x^* = V_x - \partial_x D_x$ and $V_y^* = V_y - \partial_y D_y$ [19] and the corresponding BGRW algorithm can be easily derived.

As it follows from (50), BGRW is subject to the restrictions

$$r_x + r_y \le 1,\ \ |v_x| \le r_x,\ \ |v_y| \le r_y. \tag{53}$$

Adding the conditions $r_x \le 0.5$ and $r_y \le 0.5$, the von Neumann criterion for stability is satisfied, implying that there is no numerical diffusion. The last two inequalities in (53) ensure that the Courant numbers $V_x\delta t/\delta x$ and $V_y\delta t/\delta y$ are sub-unitary, thus the algorithm also avoids the overshooting errors. As shown by (51), the BGRW algorithm is equivalent with a FD scheme even in case of space-time variable velocity fields, unlike the unbiased

GRW, for which the equivalence holds only for constant velocity. Instead, because the advective displacement is accounted for by biased jump probabilities, BGRW is not equivalent with an Euler scheme.

As a direct consequence of (53), we can see that removing the overshooting errors requires high computational costs. Let us consider an isotropic two-dimensional diffusion in groundwater ($D_x = D_y = D = 0.01$ m^2/day) in a mean flow of $U = 1$ m/day oriented along the x axis and with a standard deviation $\sigma = 0.2$ m/day. The velocity field is generated as a realization of a periodic random field, consisting of a superposition of 64 sinmodes which approximates a Gaussian field (see Section 5.1 and Equation (60)). Assuming that the maximum velocity can be as large as $V^{\max} = U + 5\sigma = 2$ m/day, from (49) and the second condition (53) it follows that $\delta x \leq 2D/V_x^{\max} = 0.01$ m. Since this space step also fulfils the third condition (53), in the following we take $\delta y = \delta x$. Correspondingly, from (49), $\delta t = 0.0025$ day (the case represented in Fig. 19). The simulation of the transport over 100 days, for a point instantaneous injection at the origin of the lattice, required about 15 cpu hours. For the same problem and consuming the same cpu time, the unbiased GRW algorithm with $\delta x = 0.1$ m and $\delta t = 0.5$ day (Fig. 18) was able to perform a transport simulation over 4000 days, on the same computer (IBM Regatta-Power 4) [32]. Nevertheless, the BGRW simulations are very helpful for the evaluation of other numerical methods [32], mainly, as in the case presented here, when no analytical solutions are available.

We computed the first and second centered spatial moments of the density ρ, defined by

$$\mu_\alpha(t) = \int\int \alpha\rho(x,y,t)dxdy,\ \mu_{\alpha\alpha}(t) = \int\int (\alpha - \mu_\alpha)^2\rho(x,y,t)dxdy, \qquad (54)$$

where α stands for x or y and the integrals are computed over the support of ρ. Further, using (54), we computed the derivatives of the 1-st moments $V_\alpha^{cm} = d\mu_\alpha/dt$, which represent the velocity components of the center of mass of the solute body, and the half-rates of increase with time of the 2-nd moments $D_{\alpha\alpha}^{eff} = \mu_{\alpha\alpha}/(2t)$, which in the large time limit define the effective diffusion coefficients for the transport process [2].

The self-averaging of the GRW simulations for the transport problem considered here is ensured if the total number of particles is of the order $N = 10^{10}$ [28]. Using this value of N in all cases, the numerical solution $\rho = \overline{n}$ was estimated by the actual number of particles n at the lattice sites.

The moments (54) were computed with BGRW for the parameters $\delta x = 0.01$ m and $\delta t = 0.0025$ day (case b1) and for a finer discretization, $\delta x = 0.005$ m and $\delta t = 0.000625$ day (case b2), with $r_x = r_y = r = 0.5$ in both cases. The errors of BGRW simulation for the case (b1) are estimated by

$$\varepsilon(V_\alpha^{cm}) = \sqrt{\frac{1}{T}\sum\nolimits_{k=0}^{k=T}(\Delta V_\alpha^{cm})^2(k)},\ \varepsilon(D_{\alpha\alpha}^{eff}) = \sqrt{\frac{1}{T}\sum\nolimits_{k=0}^{k=T}(\Delta D_{\alpha\alpha}^{eff})^2(k)} \qquad (55)$$

where ΔV_α^{cm} and $\Delta D_{\alpha\alpha}^{eff}$ are the deviations of the corresponding quantities computed in case (b1) with respect to those obtained in case (b2) and T is the simulation duration. The error estimates presented in Table 1 are orders of magnitude smaller than the fluctuations of the first two moments of the density ρ (governed by the physical parameters $D = 0.01$ m^2/day and $\sigma = 0.2$ m/day). A numerical investigation on the convergence of BGRW by comparisons with analytical solutions has not yet been done. However, since there are no

$\varepsilon(V_x^{cm})$	$\varepsilon(V_y^{cm})$	$\varepsilon(D_{xx}^{eff})$	$\varepsilon(D_{yy}^{eff})$
0.00033 m/day	0.00026 m/day	0.00075 m^2/day	0.00002 m^2/day

Table 1. Errors for the BGRW simulation case (b1) estimated by comparison with the BGRW simulation case (b2).

overshooting errors, it is expected that, for the large number of particles used in simulations, the convergence order for BGRW is the same as that for GRW simulations of genuine diffusion (which was shown in Section 3.2 to be $\mathcal{O}(\delta x^2)$). Since, due to conditions (53), this order is much smaller than for the particles methods with overshooting, BGRW solutions can serve as reference to evaluate the coarser (but faster) unbiased GRW algorithm.

As an illustration, we compare in Figs. 20 and 21 the deviations ΔV_α^{cm} and $\Delta D_{\alpha\alpha}^{eff}$ with respect to BGRW (case b1) of the results given by unbiased GRW for the sets of parameters $\delta x = 0.1$ m, $\delta t = 0.5$ day, $r = 0.25$ (case u1) and $\delta x = 0.01$ m, $\delta t = 0.1$ day, $r = 0.408$ (case u2). The corresponding error estimations via (55) are given in Table 2. Estimates like those given in Table 2 can be used to check whether the numerical setup of

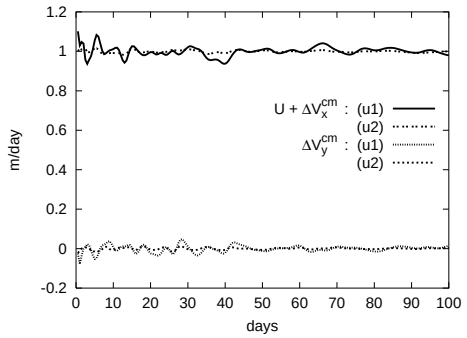

Figure 20. Deviations ΔV_α^{cm} from BGRW solution of unbiased-GRW results for simulations (u1) and (u2).

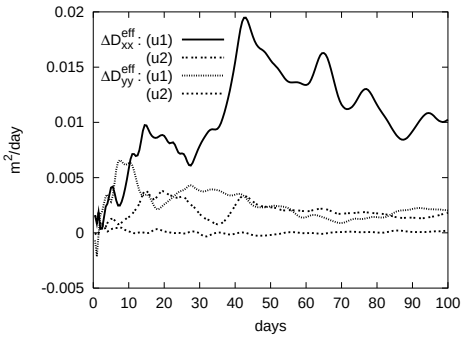

Figure 21. Deviations $\Delta D_{\alpha\alpha}^{eff}$ from BGRW solution of unbiased-GRW results for simulations (u1) and (u2).

	$\varepsilon(V_x^{cm})$	$\varepsilon(V_y^{cm})$	$\varepsilon(D_{xx}^{eff})$	$\varepsilon(D_{yy}^{eff})$
(u1)	0.02359 m/day	0.01716 m/day	0.01317 m^2/day	0.00257 m^2/day
(u2)	0.00612 m/day	0.00524 m/day	0.00312 m^2/day	0.00039 m^2/day

Table 2. Errors for the unbiased GRW simulations case (u1) and case (u2) estimated by comparisons with the BGRW simulation case (b1).

the unbiased GRW simulation fulfils the accuracy requirements for specific investigations. For instance, we conclude that even if the coarser discretization (u1) yields errors $\varepsilon(D_{xx}^{eff})$ of the order of the local diffusion coefficient D it is still accurate enough to reproduce the behavior of the expectations (averages over ensembles of velocity fields) of the longitudinal effective coefficients D_{xx}^{eff}, which are one order of magnitude larger than D [30, 31]. In

this case the unbiased GRW can be successfully used in investigations on the large time behavior and self-averaging properties of the transport process [31, 11]. Since for (u2) the errors are one order of magnitude smaller, in this case the unbiased GRW can be used to simulate the behavior in single realizations. BGRW should be used when higher accuracy is necessary (e.g. smaller time scales, transport in multi-phase systems, complex reaction-diffusion processes).

BGRW is similar to the CA method for reaction-diffusion processes proposed by Karapiperis and Blankleider [17] and their modeling strategy for the reaction step can be directly implemented in our algorithm. The essential difference is that, instead of the sequential procedure of Karapiperis and Blankleider [17], BGRW performs a global spreading of particles over lattice sites and, moreover, benefits from the flexibility of choosing the global spreading method (Bernoulli distributions, reduced fluctuations GRW, erf-GRW, presented in Section 3.1). Since BGRW is anyway equivalent to a FD method, the question naturally arising is why don't use just the FD scheme? For two reasons. First, by using particles BGRW yields a physical meaningful smallest particles density, that of one particle per lattice site, and diffusion fronts have a smaller extent than for FD schemes, which reduces the computational effort. The second reason is that BGRW mimics the molecular chaos by the fluctuations of the numbers of particles. Therefore, BGRW is able to describe phenomena that are not captured by FD numerical approaches. For instance, coherent structures produced by the interplay of reaction dynamics and molecular chaos, in absence of concentration gradients, when the diffusion term of FD vanish, already captured by particles-CA [17], can be efficiently simulated by a BGRW algorithm.

4.2. Evaluation of unbiased GRW by comparison with BGRW

We consider the generic problem for transport in groundwater, the numerical setup (b1) for BGRW, and the setup (u1) for the unbiased GRW, described in the previous section.

To assess the reliability of the faster, but coarser, unbiased GRW simulations we shall use the overshooting-free (and more expensive in terms of cpu time) BGRW algorithm. Because of the statistical nature of the predictions for groundwater contamination, we go beyond the single realization comparisons done in the previous section, and proceed with an evaluation of GRW solutions for the means and fluctuations of the observables of the transport process, computed by averaging over an ensemble of 256 velocity realizations. As observables we consider the spatial moments of the solute concentration and the space average of the concentration over the cross-section of the solute plume.

The two-dimensional version of the equation (1) was solved by GRW and BGRW, for identical realizations of the velocity, point instantaneous injection at the origin of the computational domain. The grid dimensions were fixed to $L_1 = 150$ m and $L_2 = 20$ m, so that during the total simulation time, $T = 100$ days, no particle reached the boundary. We evaluated, from both GRW and BGRW simulations, the first and the 2-nd spatial moments of the concentration (54), the velocity components of the center of mass $V_l^{cm}(t) = d\mu_l(t)/dt$, and the effective diffusion coefficients $D_{ll}^{eff}(t) = \mu_{ll}(t)/(2t)$, where $l = 1, 2$. The cross-section concentration was computed as mean concentration in a narrow slab of dimensions

1 m ($= 10\delta x$) times 20 m ($= L_2$) with the formula [30]

$$C(x_1,t) = \frac{1}{N}\frac{1}{10\delta x L_2}\sum_{i_2=1}^{L_2/\delta x}\sum_{i_1'=-5}^{5} n(i_1+i_1', i_2, k\delta t). \tag{56}$$

For comparisons we used the cross-section concentration at the center of mass of the plume, i.e., (56) evaluated at $x_1 = \mu_1(t)$.

The evaluation of the center of mass velocity was done using the absolute errors

$$\delta(\Phi) = \Phi(\text{GRW}) - \Phi(\text{BGRW}), \tag{57}$$

where Φ stands, respectively, for the expectation $E(V_l^{cm})$, computed by arithmetic means over the 256 velocity realizations, and for the corresponding standard deviations $SD(V_l^{cm}) = \{E[(V_l^{cm})^2] - [E(V_l^{cm})]^2\}^{1/2}$. The results for $l = 1$ and $l = 2$ are presented in Figs 22 and 23. In these figures we also plotted by horizontal lines the mean values of the errors (57) calculated by

$$\|\delta(\Phi)\| = \left\{\frac{1}{T-T_1}\int_{T_1}^{T}[\delta(\Phi)(t)]^2 dt\right\}^{1/2}, \tag{58}$$

where $T_1 = 1$ day.

The evaluation for the effective coefficients D_{ll}^{eff} and cross-section concentration C were achieved by using the percentage relative errors

$$\varepsilon(\Phi) = 100\frac{\Phi(\text{GRW}) - \Phi(\text{BGRW})}{\Phi(\text{BGRW})}, \tag{59}$$

where Φ stands, again, for the corresponding expectations, $E(\cdot)$, and standard deviations, $SD(\cdot)$. The results for the longitudinal and transverse effective coefficients are given in Figs 24 and 25, respectively, and those for the cross-section concentration in figure 26. The horizontal lines in these figures correspond to the mean errors (59), calculated similarly to (58), as $\|\varepsilon(\Phi)\|$.

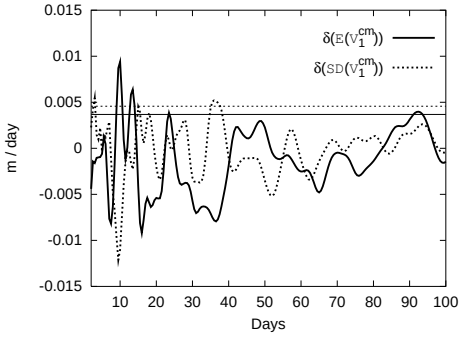

Figure 22. Evaluation of the longitudinal component of the center of mass velocity.

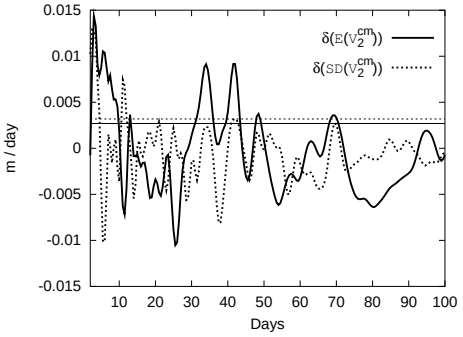

Figure 23. Evaluation of the transverse component of the center of mass velocity.

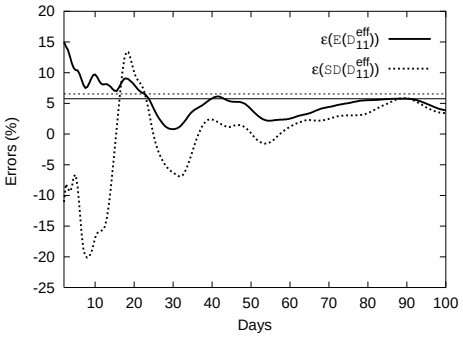

Figure 24. Evaluation of the longitudinal effective diffusion coefficient.

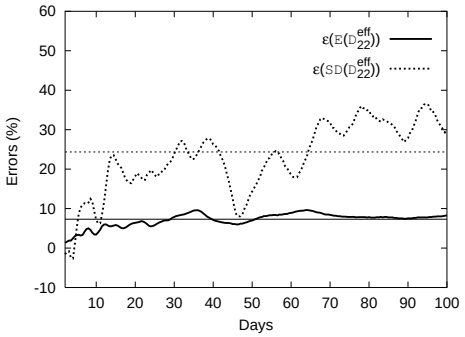

Figure 25. Evaluation of the transverse effective diffusion coefficient.

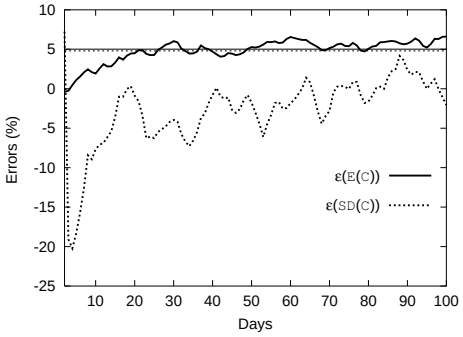

Figure 26. Evaluation of the cross-section concentration at the plume center of mass.

A comparison between Figs. 22 and 23 and the theoretical expectation values $E(V_1^{cm}) = 1$ m/day and $E(V_2^{cm}) = 0$ m/day [37], shows that GRW reproduces the mean and the fluctuations of the velocity of the plume center of mass with a very good precision of the order of a few cm/day. Figures 25 and 27 also show that the GRW estimations of the longitudinal effective coefficient and cross-section concentration are obtained with a satisfactory precision of about 5%. The errors for the transverse effective coefficient are larger, mainly those for the standard deviation (Fig. 25). However, since limit theorems results for the long time behavior of the transport process considered here predict up-scaled diffusion coefficients given by asymptotic expansions truncated at the order of local diffusion coefficient D [37], absolute errors of the GRW estimates for the transverse coefficient smaller than D are also acceptable. The GRW algorithm, validated in this way by comparisons with BGRW results, has been used to evaluate the classical "first-order approximation" of the longitudinal effective diffusion coefficients [31].

5. Sensitivity/uncertainty analysis of macrodispersion model

5.1. Monte Carlo simulations

To enable the simulation of large ensembles of transport realizations, a linearization of the flow equation (3) was considered and the velocity samples were generated, for given statistics of the hydraulic conductivity K, by the Kraichnan's randomization method [26], which has been successfully used in numerical investigations on large scale behavior of the passive transport in aquifers [11, 30, 31]. We considered a log-normally distributed conductivity K, i.e., a normal $\ln K$ field with variance σ^2 and exponential isotropic correlation $\rho(|\mathbf{x}_1 - \mathbf{x}_2|) = \sigma^2 \exp(-|\mathbf{x}_1 - \mathbf{x}_2|/\lambda)$, where λ is the correlation length. For a given pressure gradient between the inlet and outlet boundaries, which fixes the value of the ensemble mean velocity $U = |\langle \mathbf{V} \rangle|$, the incompressible Darcy flow, solution of equations (3), was approximated by a superposition of N_p periodic modes [27]

$$V_i(\mathbf{x}) = U\delta_{i1} + U\sigma\sqrt{\frac{2}{N_p}}\sum_{l=1}^{N_p} p_i(\mathbf{q}_l)\sin(\mathbf{q}_l \cdot \mathbf{x} + \alpha_l). \tag{60}$$

The wave vectors $\mathbf{q}_l$ are mutually independent random variables, with probability distribution proportional with the spectral density of the $\ln K$ field, and the phases α_l are random variables uniformly distributed in the interval $[0, 2\pi]$. The functions p_l are projectors which ensure the incompressibility of the flow. It has been shown that V_i tends to a Gaussian random field when $N_p \to \infty$ [26]. It was also found that $N_p = 6400$, which we fix in the following, provides reliable approximations of the velocity field at the problem's spatial scale considered here [30, 11].

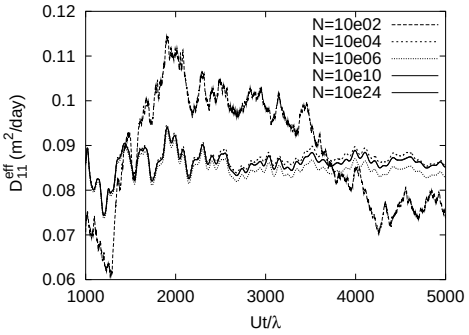

Figure 27. Single realization GRW estimates of longitudinal diffusion coefficient for increasing N show that $N = 10^{10}$ ensures the self-averaging of the estimate.

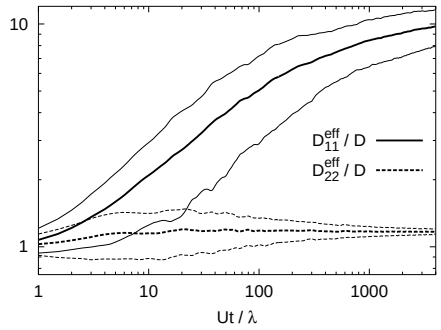

Figure 28. Monte Carlo estimates of mean value and standard deviations (thin lines) of the longitudinal and transverse diffusion coefficients for a singular initial condition.

The mean velocity occurring in (60), which can be freely chosen, was set to a typical value of $U = 1$ m/day. We also have chosen a typical local-scale diffusion coefficient in (1), $D = 0.01$ m^2/day, and $\lambda = 1$ m for the correlation length of the $\ln K$ field, so that the Péclet number was set to Pe$= U\lambda/D = 100$. Transport simulations were carried out with the two-dimensional unbiased GRW algorithm with reduced fluctuations, constructed as a

superposition of one-dimensional GRW procedures (see Section 3.4 and Fig. 12), within the numerical setup (u1) described in Section 4.1. The total number of particles was $N = 10^{10}$, a value which ensures the self averaging of the single GRW simulations and, as shown in Fig. 27, yields estimates of effective diffusion coefficients that are indistinguishable from those for $N > 10^{10}$ (in fact we found that they coincide in the limit of double precision [28]). In this way, we excluded a possible source of errors, that was important in traditional Monte Carlo simulations based on PT procedures using thousands of particles (e.g. [27]). By averaging over ensembles of 1024 realizations we estimated mean values and standard deviations of the effective diffusion coefficients (see e.g. Fig. 28) with Monte Carlo errors smaller than half the local coefficient D [34].

We conducted Monte Carlo simulations for two cases, corresponding to two extreme degrees of heterogeneity: $\sigma^2 = 0.1$, for which the approximation (60) of the velocity field is accurate and the macrodispersion model is expected to provide a reliable description of the mean behavior of the transport process, and $\sigma^2 = 6$, an extremely large value, for which (60) is no longer close to the true solution of flow equations (3) but can however serve to illustrate the situation when the macrodispersion model might be inadequate.

The behavior of a passive tracer, initially uniformly distributed in slabs of dimensions $100\lambda \times \lambda$ perpendicular to the mean flow direction, was simulated over 2000 days for the low heterogeneity case $\sigma^2 = 0.1$, in 1024 realizations of the random field (60), and over 300 days, in 100 realizations in the highly heterogeneous case $\sigma^2 = 6$. The contours of the solute plumes in the two extreme cases are compared in Figs. 29 and 30. (Note that the spatial simulation domain was, in all cases, large enough to avoid the influence of the boundaries.)

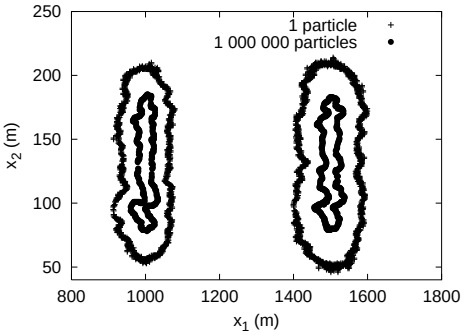

Figure 29. Plume contours for $\sigma^2 = 0.1$ at $t = 1000$ days and at $t = 1500$ days.

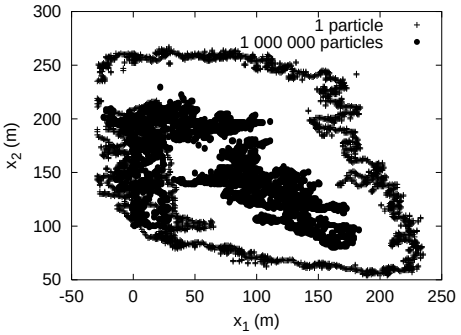

Figure 30. Plume contours for $\sigma^2 = 6.0$ at $t = 10$ days and at $t = 100$ days.

5.2. Sensitivity and uncertainty analysis

Monte Carlo estimates, by equal-weight (arithmetic) averages over the corresponding ensembles of realizations, hereafter denoted by $\langle \cdots \rangle$, were computed for the set of input parameters of the macrodispersion model, consisting of longitudinal $u = dE(X_1)/dt$ and transverse $v = dE(X_2)/dt$ components of the center of mass velocity, longitudinal $D_x = D_{11}^{eff}$ and transverse $D_y = D_{22}^{eff}$ effective diffusion coefficients, for the only output

parameter considered here, consisting of the cross-section space average concentration at the center of mass (hereafter denoted by c), as well as for their cross-correlations, $\langle uv \rangle$, $\langle uD_x \rangle$, $\langle uD_y \rangle$, $\langle vD_x \rangle$, $\langle vD_y \rangle$, $\langle D_x D_y \rangle$, $\langle uc \rangle$, $\langle vc \rangle$, $\langle D_x c \rangle$, and $\langle D_y c \rangle$. Probability densities of the parameters, approximated by histograms, were summed-up to estimate cumulative probability distributions.

Figure 31 shows that for low heterogeneity ($\sigma^2 = 0.1$) the only input-input relevant correlation is that between the longitudinal velocity of the center of mass and the transverse effective diffusion coefficient. The sensitivity to the longitudinal mean flow velocity of the transverse effective diffusion coefficient indicates its increased role in case of small mean flow velocity. The results for the highly heterogeneous case ($\sigma^2 = 6$) presented in Fig. 5 show stronger correlations between the input parameters, which are expected to facilitate the uncertainty propagation and to reduce the reliability of the macrodispersion model.

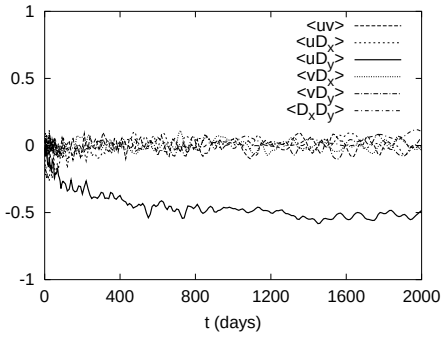

Figure 31. Correlations between the input parameters of the macrodispersion model (velocity components of center of mass, u and v, and effective diffusion coefficients, D_x and D_y) for $\sigma^2 = 0.1$.

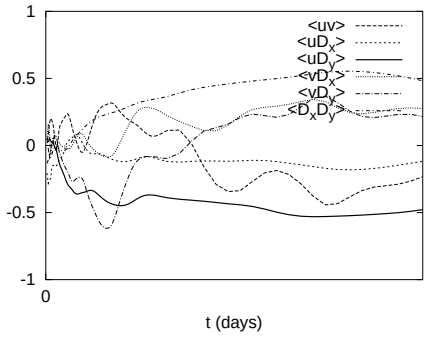

Figure 32. Correlations between the input parameters of the macrodispersion model (velocity components of center of mass, u and v, and effective diffusion coefficients, D_x and D_y) for $\sigma^2 = 6.0$.

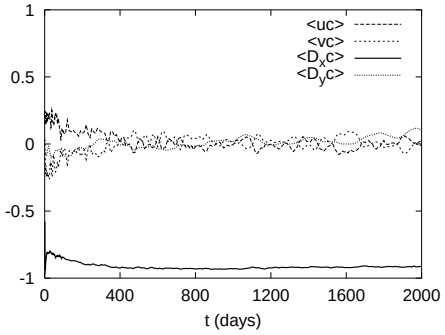

Figure 33. Correlations between input parameters u, v, D_x, and D_y, and the output parameter c (the cross-section space average concentration at the center of mass) for $\sigma^2 = 0.1$.

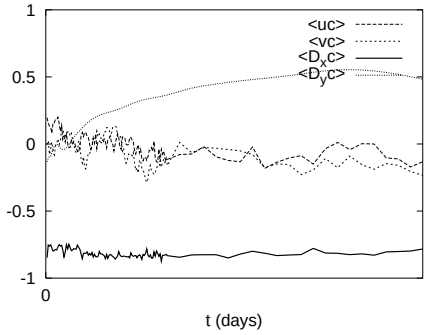

Figure 34. Correlations between input parameters u, v, D_x, and D_y, and the output parameter c (the cross-section space average concentration at the center of mass) for $\sigma^2 = 6.0$.

As expected, for low heterogeneity (Fig. 33) there is a strong correlation between the

longitudinal effective diffusion coefficient and the cross-section averaged concentration. This suggests that, when the only output parameter of interest is the cross-section concentration, the macrodispersion model can be trusted as reliable for single-realizations of the transport process, in agreement with other observations that the cross-section concentration can be modeled as an one-dimensional advection-diffusion process governed by the longitudinal effective diffusion coefficient [30]. The situation is different for high heterogeneity (Fig. 34), where the cross-section concentration is also strongly correlated with the transverse effective diffusion coefficient. Again, this result renders questionable the applicability of the macrodispersion model to highly heterogeneous media. To illustrate the capability of

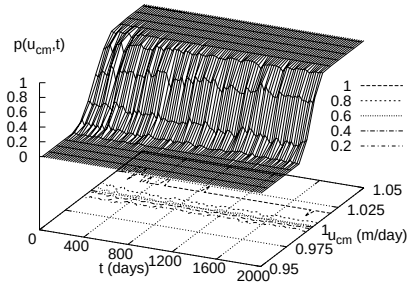

Figure 35. Probability distributions of the longitudinal component of the velocity of the center of mass as function of time $u_{cm}(t)$ for $\sigma^2 = 0.1$.

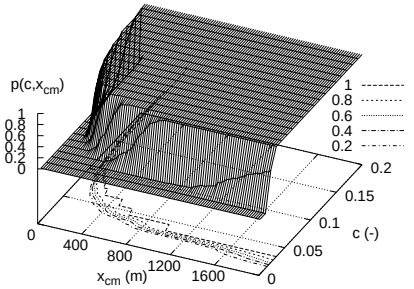

Figure 36. Probability distributions of the concentration c estimated along the longitudinal component of the center of mass x_{cm} for $\sigma^2 = 0.1$.

the Monte Carlo approach based on GRW simulations to produce a full statistical description of the transport process, we present in Fig. 35 the estimated cumulative probability distribution of the longitudinal velocity of the center of mass and in Fig. 36 the distribution of the cross section concentration at the plume's center of mass.

6. Conclusion

The GRW algorithm moves groups of particles according to the random walk rule, generalizing the usual sequential algorithms which generate trajectories of the individual particles. Practically, GRW simulates the collective diffusive behavior of large systems of particles at costs comparable with those of moving a single particle by sequential algorithms. This results in a significant saving of memory and computing time. The number of particles is limited only by the maximum number that can be represented on a computer. As a random walk method, GRW is stable and free of numerical diffusion [43].

The unbiased GRW algorithm can be thought of as a weak Euler scheme for the Itô equation which, instead of simulating individual trajectories, provides a numerical solution for the corresponding probability density governed by the associated Fokker-Planck equation. For simple diffusion processes with constant drift coefficients, GRW generalizes the FD scheme and, for large enough number of particles, has the same precision and reaches the same order of convergence. The biased algorithm BGRW models the drift as a bias in

the jump probability, and, therefore, it is no longer equivalent to an Euler scheme for the Itô equation. Instead, BGRW is equivalent to a stable FD scheme free of numerical diffusion for the Fokker Planck equation, even in the general case of space-time variable drift and diffusion coefficients. Even though BGRW requires finer grids, hence higher computational resources, it is very useful as a reference method in evaluating the overshooting errors of the unbiased GRW codes [32, 31]. By working with indivisible particles, both BGRW and unbiased GRW avoid extending diffusion fronts beyond the minimum concentration of one particle per lattice site, saving time, and mimic the molecular chaos by fluctuations of the number of particles, accounting for subtle phenomena such as coherent structures at equilibrium, which cannot be described by FD methods [17].

GRW algorithms are suitable for modeling complex diffusion processes such as advection-dominated transport with random space-variable coefficients, with chemical reactions between several molecular species, as well as radioactive decay processes. The numerical modeling may benefit of the simple cellular automaton structure of the GRW algorithms and of their ability to consider systems of particles as large as the real number of species molecules involved in chemical reactions. Owing to the formal equivalence of Fick's and Darcy's laws, GRW solutions of flow in porous media described by Richards equation [25] can also be implemented. Another promising application is the multidimensional GRW simulation for the evolution of the probability distributions of the concentrations in reactive turbulent transport, modeled as diffusive process in a multidimensional space constructed as cartesian product between the physical space and the ranges of the species concentrations [3, 24, 41]. In principle, the GRW algorithm can be adapted to solve problems for parabolic differential equations or ordinary stochastic differential equations whenever they describe normal diffusion processes and are compatible with random walk approximations. Anomalous diffusion processes do not belong to this class but some anomalous transport problems can yet be solved similarly with the problem of diffusion in random fields presented in this chapter. For instance, if the drift coefficients are samples of a fractional Gaussian noise in space, ensembles of GRW simulations may be used to assess the statistics of the resulting process, consisting of a superposition of normal diffusion and fractional Brownian motion [37].

Acknowledgement. The development of the GRW algorithms was supported by the Romanian Academy research program II "Connection between differentiable and associated discrete dynamical systems" at the Tiberiu Popoviciu Institute of Numerical Analysis, the Project No. JICG41 at Jülich Supercomputing Centre, and the Deutsche Forschungsgemeinschaft under Grant SU 415/1-2.

References

[1] Ames, W. F. *Numrical Methods for Partial Differential Equations*; Academic Press: New York, 1977.

[2] Attinger, S.; Dentz, M.; Kinzelbach, H.; Kinzelbach, H. *J. Fluid Mech.* 1999, *386*, 77-104.

[3] Colucci, P. J.; Jaberi, F. A.; Givi, P. *Phys. Fluids* 1998, *10*, 499-515.

[4] Chorin, A. J. *J. Comput. Phys.* 1978, *27*, 428-442.

[5] Crank, J. *The Mathematics of Diffusion*; Oxford Univ. Press: Oxford, 1975.

[6] Dagan, G. *Flow and Transport in Porous Formations*; Springer: Berlin, 1989.

[7] Delay, F. ; Dzikowski, M.; de Marsily, G. *Mathematical Geology* 1993, *25*, 689-712.

[8] Delay, F.; Housset-Resche, H.; Porei, G.; de Marsily, G. *Mathematical Geology* 1996, *28*, 45-71.

[9] Delay, F.; Ackerer, P.; Danquigny, C. *Vadose Zone Journal* 2006, *4*, 360-379.

[10] Doering, C. R.; Horsthemke, W.; Riordan, J. *Phys. Rev. Lett.* 1994, *72*, 2984-2987.

[11] Eberhard, J.; Suciu, N.; Vamoş, C. *J. Phys. A: Math. Theor.* 2007, *40*, 597-610, doi: 10.1088/1751-8113/40/4/002.

[12] Fannjiang, A.; Komorowski, T. *Ann. Appl. Probab.* 1999, *9*, 591-610.

[13] Gardiner, C. W. *Handbook of Stochastic Methods*; Springer: New York, 1983.

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

[15] Ghoniem, A. F.; Sherman, F. S. *J. Comput. Phys.* 1985, *61*, 1-37.

[16] Horsthemke, W.; Lefever, R. *Noise-Induced Transitions. Theory and applications in Physics, Chemistry and Biology*; Springer: Berlin, 1984.

[17] Karapiperis, T.; Blankleider, B. *Physica D* 1994, *78*, 30-64.

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

[19] Kloeden, P. E.; Platen, E. *Numerical Solutions of Stochastic Differential Equations*; Springer: Berlin, 1999.

[20] Komorowski, T.; Papanicolaou, G. *Ann. Appl. Probab.* 1997, *7*, 229-264.

[21] Moltyaner, G. L.; Klukas, M. H.; Willis, C. A.; Killey, R. W. D. *Water Resour. Res.* 1993, *29*, 3433-3452.

[22] Neuendorf, O.; *Numerische 3D-Simulation des Stofftransport in einem heterogenen Aquifer*; Ph.D. thesis Jül-3421; Forschungszentrum Jülich: Jülich, 1997.

[23] Papoulis, A.; Pillai, S. U. *Probability, Random Variables and Stochastic Processes*; McGraw-Hill: New York, 1991.

[24] Pope, S. B. *Annu. Rev. Fluid Mech.* 1994, *26*, 23-63.

[25] Radu, F. A.; Suciu, N.; Hoffmann, J.; Vogel, A.; Kolditz, O.; Park, C.-H.; Attinger, S. *Adv. Water Resour.* 2011, *34*, 47-61, doi:10.1016/j.advwatres.2010.09.012, 2010.

[26] Sabelfeld, K. *Monte Carlo methods in boundary value problems*; Springer: Berlin, 1991.

[27] Schwarze, H.; Jaekel, U.; Vereecken, H. *Transport in Porous Media* 2001, *43*, 265-287.

[28] Suciu N.; Vamoş, C.; Vanderborght, J.; Hardelauf, H.; Vereecken, H. *Monte Carlo Methods and Appl.* 2004, *10*, 155-179.

[29] Suciu, N.; Vamoş, C.; Knabner, P.; Rüde, U. In *Simulationstechnique,* 18^{th} *Symposium in Erlangen, September 2005*; Hülsemann, F.; Kowarschik, M.; Rüde, U.; SCS Publishing House e. V.: Erlangen, 2005; 562-567.

[30] Suciu, N.; Vamoş, C.; Vanderborght, J.; Hardelauf, H.; Vereecken, H. *Water Resour. Res.* 2006, *42*, W04409, doi:10.1029/2005WR004546.

[31] Suciu, N.; Vamoş, C.; Eberhard, J. *Water Resour. Res.* 2006, *42*, W11504, doi:10.1029/2005WR004714.

[32] Suciu, N.; Vamoş, C. *Revue d'Analyse Numérique et de Théorie de l'Approximation* 2006, *35*, 119-126.

[33] Suciu, N.; Vamoş, C.; Turcu, I.; Pop, C.V.L.; Ciortea, L. I. *Comput. Visual. Sci.* 2007, *12*, 77-85, doi:10.1007/s00791-007-0077-6.

[34] Suciu, N.; Vamoş, C.; Vereecken, H.; Sabelfeld, K.; Knabner, P. *Water Resour. Res.* 2008, *44*, W08501, doi:10.1029/2007WR006740.

[35] Suciu, N.; Knabner, P. *Water Resour. Res.* 2009, *45*, W05601, doi:10.1029/2008WR007498.

[36] Suciu, N.; Vamos, C.; Radu, F. A.; Vereecken, H.; Knabner, P. *Phys. Rev. E* 2009, *80*, 061134, doi:10.1103/PhysRevE.80.061134.

[37] Suciu, N. *Phys. Rev. E* 2010, *81*, 056301, doi:10.1103/PhysRevE.81.056301.

[38] Sun, Ne-Z. *Mathematical Modeling in Groundwater Pollution*; Springer: New York, 1996.

[39] Tompson, A. F. B.; Gelhar, L. W. *Water Resour. Res.* 1990, *26*, 2541-2562.

[40] Tompson, A. F. B.; Falgout, R. D.; Smith, S. G.; Bosl, W. J.; Asby, S. F. *Adv. Water Resour.* 1998, *22*, 203-221.

[41] Valiño, L.; Dopazo, C. *Phys. Fluids A* 1991 *3*, 3034-3037.

[42] Vamoş, C.; Suciu, N.; Vereecken, H.; Nitzsche, O.; Hardelauf, H. In *Proceedings of Scan2000 conference, September 18 - 22, 2000, Karlsruhe, Germany*, Kramer, W.; von Gudenberg, J. W.; Kluwer: Dordrecht, 2001; 343-354.

[43] Vamoş, C.; Suciu, N.; Vereecken, H. *J. Comput. Phys.* 2003, *186*, 527-544, doi:10.1016/S0021-9991(03)00073-1.

[44] Vamoş, C.; Suciu, N.; Georgescu, A. *Phys. Rev. E* 1997, *55*, 6277-6280.

[45] Vamoş, C.; Suciu, N.; Peculea, M. *Revue d'Analyse Numérique et de Théorie de l'Approximation* 1997, *26*, 237-247.

[46] Vamoş, C.; Georgescu, A.; Suciu, N.; Turcu, I. *Physica A* 1996, *227*, 81-92.