



Convergence analysis of macrospreading in 3D heterogeneous porous media

Anthony Beaudoin, Jean-Raynald de Dreuzy, Jocelyne Erhel, Géraldine Pichot

► To cite this version:

Anthony Beaudoin, Jean-Raynald de Dreuzy, Jocelyne Erhel, Géraldine Pichot. Convergence analysis of macrospreading in 3D heterogeneous porous media. ESAIM: Proceedings, 2013, 41, pp.59-76. 10.1051/proc/201341003 . hal-00904436

HAL Id: hal-00904436

<https://inria.hal.science/hal-00904436>

Submitted on 5 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

CONVERGENCE ANALYSIS OF MACRO SPREADING IN 3D HETEROGENEOUS POROUS MEDIA^{*,**}

A. BEAUDOIN¹, J.-R. DE DREUZY², J. ERHEL³ AND G. PICHOT³

Abstract. Models of hydrogeology must deal with both heterogeneity and lack of data. We consider in this paper a flow and transport model for an inert solute. The conductivity is a random field following a stationary log normal distribution with an exponential or Gaussian covariance function, with a very small correlation length. The quantities of interest studied here are the expectation of the spatial mean velocity, the equivalent permeability and the macro spreading. In particular, the asymptotic behavior of the plume is characterized, leading to large simulation times, consequently to large physical domains. Uncertainty is dealt with a classical Monte Carlo method, which turns out to be very efficient, thanks to the ergodicity of the conductivity field and to the very large domain. These large scale simulations are achieved by means of high performance computing algorithms and tools.

1. INTRODUCTION

Groundwater resources are a key societal issue, both for quantity and quality of water. They must be managed in a sustainable way, with a fine tracking of contaminants. Numerical models are essential to address complex systems such as geological porous media, which are highly heterogeneous [16]. Underground media very often do not show any obvious homogenization scale and moreover display characteristic structures on a widely-scattered scale range, with a permeability spanning several orders of magnitude [25]. Moreover, the lack of measures implies uncertainty in the data, which must be introduced in the model. A common way is to consider input data as random fields, implying that output results are also random fields. Then the model should produce values of interest such as first moments or statistics of derived quantities.

In this paper, we consider flow and transport in highly heterogeneous porous media. Because of the widely-scattered medium heterogeneity, velocity values of groundwater span several orders of magnitudes. Particles are advected by the velocity field along the flow lines and can switch from flow lines because of diffusion and hydrodynamic dispersion and their spreading creates a so-called plume. The development of the plume has been analyzed for a finitely-correlated permeability field [13], [25], [38]. Thanks to the software PARADIS [20], the full development of the plume was simulated for the first time in large 2D domains and asymptotic dispersion was computed with no ambiguity [14], [15], [18].

Numerical models with random input data have been extensively studied recently. Stochastic finite element methods as well as collocation stochastic methods or multilevel Monte Carlo methods were developed, see for

* This work was partially supported by ANR-CIS, with the MICAS project and ANR-MN, with the H2MNO₄ project.

** Some computations were done with the Grid'5000 facility.

¹ University of Poitiers, UMR Pprime, Poitiers, France

² CNRS, UMR Géosciences, Rennes, France

³ Inria, Rennes, France

example [2], [3], [4], [9], [11], [12], [22], [24], [27], [31], [33], [40], [42]. In most of these methods, the conductivity field is first discretized in the probability space, usually with a truncated Karhunen-Loève expansion, then the physical space is discretized, by a classical finite element method for example. However, these approaches suffer from a curse of dimensionality, especially when the correlation length is small. In [14], [15], a classical Monte Carlo method is used, where physical space is first discretized, then a discrete random conductivity field is generated. It allows using very efficient methods such as spectral methods [17], [34], [41]. Recently, a similar approach was proposed in [28], along with a Quasi Monte Carlo method. In this approach, the numerical approximations are finely analyzed mathematically then the efficiency is illustrated for 2D domains and a conductivity field with a medium correlation length.

In this paper, we choose a classical Monte Carlo method and apply it to large 3D domains and a log normal random conductivity field with a very small correlation length. In order to characterize an asymptotic behavior, the transport must be simulated during many timesteps, thus simulations require a very large domain. We use a regular grid for space discretization and a mixed finite element method for the flow equation, which is here equivalent to a finite volume method. Periodic boundary conditions avoid border effects. The induced sparse linear system is solved by means of an algebraic multigrid method [39]. A Lagrangian method based on a random walker is used for the transport equation, in order to avoid numerical artefacts such as artificial diffusion [43]. The algorithm follows the flow lines for the advection process and simulates a brownian motion for the diffusion process.

This numerical model combining Monte Carlo, finite element and particle tracking is mathematically analyzed in [8], [10]. Since we use very large physical domains, ergodic conditions apply and convergence can be quite fast, as observed in [14]. The quantities of interest studied here are the spatial mean velocity and the macro spreading, in advection diffusion cases. The macro spreading is computed by means of the statistical moments of the particle trajectories. Here we report mainly an experimental analysis of convergence, whereas the characterization of the plume is discussed in [5].

The paper is organized as follows. In section 2, the physical model is given, and the main physical parameters are described. In section 3, the numerical model is explained and the main numerical parameters are highlighted. The computational cost is discussed as well as the choice of the numerical algorithms and the parallel implementation. Finally, section 4 presents numerical experiments. The choice of an algebraic multigrid linear solver and a particle tracker is justified by the experimental complexity analysis. Then convergence of the conductivity field, the spatial mean velocity and the macro spreading are analysed. It turns out that only a few Monte Carlo simulations are required to characterize the plume with sufficient accuracy.

2. PHYSICAL MODEL

In this paper, we consider a 3D porous medium defined by an open domain O . The position of a point is noted X with the three coordinates x, y, z in a given referential system. We consider here a single-phase flow and assume that the medium is saturated with a constant water density. The porous medium, assumed isotropic, is characterized by a hydraulic conductivity field K and a porosity ϵ . We also consider the transport of an inert solute, by advection and dispersion. Numerical simulations must compute the velocity field by solving flow equations, then the solute concentration by solving transport equations.

2.1. Random conductivity field

In groundwater studies, it is physically impossible to know the exact hydraulic conductivity at every point in the domain. One way to deal with uncertainty is to model conductivity as a random field, experimentally determined by the covariance structure and the expectation [13]. In our model, we introduce a probability space Ω , and the conductivity is given by a random field, which follows a second order stationary log-normal probability distribution $Y(\omega, X) = \log(K(\omega, X))$, where Y is a Gaussian field defined by an expectation μ , a variance σ^2 and a covariance function C , which is chosen either as exponential or as Gaussian, so that

$$C(r) = \sigma^2 \exp(-\frac{r}{\lambda}) \text{ or } C(r) = \sigma^2 \exp(-(\frac{r}{\lambda})^2),$$

where r represents the distance between two points and λ denotes the correlation length. This random field is thus ergodic and has mixing properties [1]. In particular, the expectation and the covariance can be approximated accurately by the spatial mean value and a variogram in the physical space, averaged over only a few realizations, provided that the domain is sufficiently large.

For example, let O be a cube with an edge length equal to L and let

$$\bar{K}(\omega) = \frac{1}{L^3} \int_O K(\omega, X) dX,$$

then $\lim_{L \rightarrow \infty} \bar{K}(\omega) = \mu$.

The porosity could also be modeled by a random field; however, in this paper, it is a deterministic function of the space variable.

2.2. Flow model

Classical laws governing the steady flow in a porous medium are mass conservation and Darcy's law:

$$\begin{cases} \epsilon(X)v(\omega, X) = -K(\omega, X)\nabla p(\omega, X), & \text{in } \Omega \times O \\ \nabla \cdot (\epsilon(X)v(\omega, X)) = 0, & \text{in } \Omega \times O, \end{cases} \quad (1)$$

where ϵ is the porosity, ϵv is the Darcy velocity and p is the hydraulic head. Boundary conditions complete the system. In the numerical experiments reported in this paper, the domain is a box, of size $L_x \times L_y \times L_z$. Dirichlet conditions $p = p_l$ and $p = p_r$ are prescribed respectively on the left and right sides, where p_l and p_r are constant values. Periodic boundary conditions are applied on the other four faces. This allows avoiding border effects and dealing with elongated domains. With these boundary conditions, the flow is mainly in the x direction.

We define the mean velocity over the physical domain by

$$\bar{v}(\omega) = \frac{1}{|O|} \int_O v(\omega, X) dX,$$

with $|O|$ the volume of the domain and we note $\bar{v}(\omega) = (\bar{v}_x(\omega), \bar{v}_y(\omega), \bar{v}_z(\omega))$. When the conductivity field is deterministic homogeneous and with other physical parameters of Table 1, the mean velocity would be equal to $(1, 0, 0)$.

For each realization ω , the equivalent conductivity is defined by

$$K_{eq}(\omega) = Q(\omega)L_x / (L_y L_z |p_l - p_r|),$$

where Q is the total volumetric flow through the inlet face of the domain. With the physical parameters of the numerical experiments, the equivalent conductivity is equal to the longitudinal mean velocity: $K_{eq}(\omega) = \bar{v}_x(\omega)$.

2.3. Transport model

We consider an inert solute, which does not exchange with the porous medium or any other species. The two mechanisms, governing the inert solute transport, are advection and dispersion and are described by the following advection - dispersion equation:

$$\frac{\partial(\epsilon(X)c(\omega, X, t))}{\partial t} + \nabla \cdot (\epsilon(X)c(\omega, X, t)v(\omega, X)) - \nabla \cdot (\epsilon(X)D(\omega, X)\nabla c(\omega, X, t)) = 0, \text{ in } \Omega \times O \times [0, T], \quad (2)$$

where c is the solute concentration in the aqueous phase and D is the dynamic dispersion tensor.

In a general setting, dispersion includes diffusion and velocity effects [43]:

$$D(\omega, X) = (\alpha_T(X) \|v(\omega, X)\| + D_m(X))I + (\alpha_L(X) - \alpha_T(X)) \frac{v(\omega, X)v(\omega, X)^T}{\|v(\omega, X)\|}, \quad (3)$$

where D_m is the molecular diffusion coefficient, α_L and α_T are respectively the longitudinal and transversal dispersion coefficients.

Boundary conditions are periodic in the four faces where periodic boundary flow conditions are applied. On the left side (inflow), an homogeneous Neumann boundary condition is used; on the right side (outflow), a Dirichlet boundary condition $c = 0$ is applied.

The initial condition is an injection of solute in a given window. In order to overcome border effects, the inert solute is injected at a distance of the left side. The initial condition is deterministic and is given by

$$c(\omega, X, 0) = c_0(X),$$

where c_0 is the normalized indicator function of the injection window.

2.4. Physical parameters

Physical data are of strong interest for phenomenological study. Indeed, one objective is to find out the impact of these parameters, such as the heterogeneity, the correlation length or the molecular diffusion, on the velocity field and the macro spreading. These parameters are also of great importance in the accuracy of the results and the CPU time required to run the simulations.

Geometrical parameters include the correlation length λ , the size of the domain defined by L_x/λ , L_y/λ and L_z/λ , where L_x , L_y and L_z are the length in each direction. Since the objective is to study the asymptotic behavior of the plume, transport simulations must involve large times, thus very large domains.

Heterogeneity is governed by the variance σ^2 of the normal distribution Y , ranging in the interval $[1 : 9]$.

The transport conditions of interest are pure advection or advection-dominated dispersion. Physical parameters are the molecular diffusion and the dispersion coefficients.

The physical parameters are summarized in Table 1. The last column indicates the values chosen in the numerical experiments of this paper.

In the numerical model of section 3 and in numerical experiments, we consider a constant porosity $\epsilon = 1$ and only an isotropic constant molecular diffusion D_m , without velocity dispersion. Thus

$$D(\omega, X) = D_m I. \quad (4)$$

2.5. Spreading and macro spreading

The objective of this model is to find out the longitudinal and transversal spreadings, which are the second moments of mass distribution. In the case of a random conductivity field, the estimation of macro spreading has to be made statistically because v and c are random fields.

The center of mass and the spreading are defined by

$$\begin{cases} G(\omega, t) = \int_O c(\omega, X, t) X dX, \\ S(\omega, t) = \int_O c(\omega, X, t) (X - G(\omega, t))(X - G(\omega, t))^T dX. \end{cases} \quad (5)$$

The quantity of physical interest studied here is the macro spreading, which is, at each time t , the expectation of the spreading:

$$S(t) = E_\omega[S(\omega, t)]. \quad (6)$$

Name	Description	Value or range
ϵ	porosity	1
μ	mean of log-conductivity	0
σ	variance of log-conductivity	[1 : 3]
L_x	domain size in x direction	[32 : 1024]
L_y	domain size in y direction	32 or 256
L_z	domain size in z direction	32 or 256
λ	correlation length	10
$C(r)$	covariance function	exponential or Gaussian
q	source term	0
p_l	head on the left face	L_x
p_r	head on the right face	0
D_m	molecular diffusion	10^{-3}
α_L	longitudinal dispersion	0
α_T	transversal dispersion	0

TABLE 1. Physical parameters in the PARADIS model. The values are those used in the numerical experiments.

2.6. Mathematical analysis

In this paper, we assume existence and uniqueness of solutions for the flow and transport equations.

3. NUMERICAL MODEL

3.1. Uncertainty quantification

The goal of uncertainty quantification (UQ) is to investigate the impact of random data and subsequently to provide reliable predictions [31], [42].

Since the systems (1) and (2) are stochastic partial differential equations, it is necessary to deal with specific methods. Stochastic finite element methods have been heavily studied for elliptic equations but they are intrusive and suffer from a curse of dimensionality [3], [24], [27]. Collocation methods are non intrusive but still very time consuming [2], [40]. Most of these methods rely on a decomposition of the conductivity field in the probability space, such as the Karhunen-Loève expansion. This decomposition is truncated in order to deal with a finite number of random variables, but this number must be quite high for covariance functions with a small correlation length and for conductivity fields which are not uniformly coercive [9], [11].

In this paper, we advocate the use of a classical Monte Carlo method. Each Monte Carlo simulation corresponds to a random draw of the conductivity field K and a system of deterministic flow and transport equations. For each Monte Carlo simulation, the numerical model is composed of three steps: generating the conductivity field, solving the flow equation, solving the transport equation. Solving flow and transport problems for each sample gives a set of the random field c used for estimating the spreading. Thus this method is clearly non intrusive and is embarrassingly parallel [21]. However, the number of samples required to approximate even the first moments is sometimes prohibitive. An alternative is to use a Quasi Monte Carlo method, where the samples are chosen in a deterministic way. This approach reduces drastically the number of samples required [28]. Another way of reducing the number of samples is the multilevel Monte Carlo method [4], [12]. Here, we restrict the study to the classical Monte Carlo method, because the random conductivity field is ergodic, and we use a large physical domain with a large injection window. Thus we can expect that the number of classical Monte Carlo simulations will remain small, as was observed for 2D domains [14].

Clearly, the sequential CPU time of Monte Carlo simulations is proportional to the number N_s of simulations.

3.2. Discrete conductivity generation

In Monte Carlo variants, it is not necessary to rely on the Karnhunen-Loève expansion. In our model, the covariance function is discretized in space and approximated by a covariance matrix. Then computing the conductivity field on the discrete spatial grid is done with a spectral density method, which is efficient for a small correlation length [34], [41]. A similar model is used in [28], but with a different method, derived from [17].

Since the domain is simple, we use a structured grid to discretize the space. Each cell i has dimensions $\Delta x, \Delta y, \Delta z$, so that the total number of cells N is

$$N = \frac{L_x L_y L_z}{\Delta x \Delta y \Delta z}. \quad (7)$$

Without loss of generality, we assume throughout the paper that $\Delta x = \Delta y = \Delta z$.

The random field K is discretized by computing an approximate value K_i in each cell. The covariance function is also discretized, yielding a covariance matrix, still called C , which has a block Toeplitz structure.

The conductivity K is generated as

$$K = \exp(\mu + \sigma Z)$$

where Z is a standard normal field (expectation 0 and variance 1) with covariance matrix C .

The spectral density method is chosen here, because the correlation length is very small compared to the domain size [17]. Then the discrete spectral density of Z is computed by using the DFT of the covariance matrix. Random phases are drawn with a uniform law, and an inverse FFT recovers the approximate normal field Z [34]. The covariance matrix C is expanded by doubling the number of points in each direction and by padding for parity, in order to guarantee real coefficients in Z when applying the inverse FFT. Also, the phases are expanded for respecting symmetry.

With sufficiently large domains and sufficiently small cells, this method is efficient and accurate [35].

The number of operations for generating the random conductivity field is roughly linear with $N \log(N)$.

3.3. Discrete flow equation

For each Monte Carlo simulation l , we get a realization of K . In what follows, we omit the variable ω , since we use a deterministic method for a given realization.

We then discretize the flow equation, using the same regular grid as for the conductivity generation. We choose a mixed finite element method with the usual RT0 element, which is equivalent here to a classical cell-centered finite volume method [30]. This approach has several nice properties. Indeed, it is well suited to flow equations, since it approximates both head and velocity; moreover, it ensures local and global mass conservation. Also, the resulting matrix is symmetric positive definite (SPD). A similar approach is used in [28], with another technique to get an SPD matrix.

In our model, we keep as unknowns the approximate hydraulic head p_i in each cell i . The discrete flow equation is then a linear system $Ap = q$, where A is a large sparse symmetric positive definite matrix. The matrix A has a regular structure, with nonzeros only for adjacent cells, thus with 7 nonzeros by row for 3D grids. Coefficients are computed by using harmonic means.

For two adjacent cells i and j , the coefficients are given by $A_{ij} = -\frac{1}{\Delta x} \text{mean}(K_i, K_j)$, where

$$\text{mean}(a, b) = \frac{2ab}{a + b}.$$

For cells which are not at a boundary, the sum of the coefficients is zero. The right-hand side q takes into account the boundary conditions and symmetry is preserved.

Once the linear system is solved, the normal velocity $v.n_{ij}$, where n_{ij} is the normal vector, is computed through each face between two adjacent cells i and j as follows:

$$v.n_{ij} = \frac{1}{\Delta x} \text{mean}(K_i, K_j)(p_i - p_j). \quad (8)$$

Inside a cell, the velocity is approximated by a bilinear interpolation, using the RT0 finite element.

The number of operations for building the matrix is linear with N .

3.4. Sparse linear solver

When solving flow equations, the most time and memory consuming part consists in solving the sparse linear system $Ap = q$. Thus the choice of an efficient sparse solver is very important [32]. The boundary conditions prevent from using easily fast solvers. Direct sparse solvers are very robust, but they are time and memory consuming. In [7], numerical experiments with 2D domains show clearly that sparse direct solvers are both CPU and memory time consuming. As predicted by the theory, the complexity is roughly in $O(N^{1.5})$ in 2D. The theoretical complexity for 3D problems (with the optimal Nested Dissection ordering) is in $O(N^2)$ [26]. Thus an iterative solver is preferred, and the objective is to get a linear complexity.

Krylov iterative methods are very efficient sparse linear solvers [19, 37]. Here, Preconditioned Conjugate Gradient (PCG) is the Krylov method of choice. In section 4.1, we show some results for PCG, where preconditioning is done with an incomplete Cholesky factorization. It appears that a multilevel preconditioner is required, in order to deal with a large system size and strong heterogeneities.

On the other hand, for such an elliptic problem, iterative multigrid methods are efficient, with a number of cycles almost independent of the size of the linear system, leading to a roughly linear complexity [39]. In some cases, multigrid may fail to converge or converge slowly. Then it can be combined with the Preconditioned Conjugate Gradient method, by using a cycle of multigrid for preconditioning PCG. This combination is more robust but can be slower than pure multigrid. Parallel multigrid solvers are available, for example within HyPre library [23].

In [7], numerical experiments show that geometric multigrid is very efficient for 2D domains, but is not adapted to highly variable coefficients, because the number of cycles becomes too high. In [20], numerical experiments compare the CPU time of geometric and algebraic multigrids methods, implemented as SMG and AMG in HyPre [23], still for 2D domains. As predicted, algebraic multigrid method can deal with highly variable coefficients without slowing down the convergence. Each cycle in AMG has roughly a linear complexity with N and the number of cycles is almost independent of the system size and the heterogeneity.

In section 4.1, we present numerical experiments with 2D domains and small 3D domains. It appears that AMG is robust and efficient also for 3D domains.

3.5. Discrete transport equation

Then, we have to discretize the transport equation for each Monte Carlo simulation l . Again, we omit the variable ω since we apply a deterministic numerical method. Because advection dominates in the transport conditions, the numerical method must not introduce artificial diffusion. We could use an Eulerian method such as a discontinuous finite element method; an operator splitting approach would allow to use tuned methods for advection and for dispersion [29]. ELLAM methods mixing Eulerian and Lagrangian descriptions are also promising [36].

Here, we choose to use a pure Lagrangian approach, which does not introduce artificial diffusion [43]. More precisely, we use a probabilistic random walker approach, which is very simple but which requires in general a high number of particles. Nevertheless, our 2D numerical experiments show that the CPU time for solving the transport equation remains smaller than for solving the flow equation [20].

We consider the physical parameters of Table 1. With a sufficiently large domain, only a small amount of solute will reach the boundary and the velocity can be extended by 0 outside of the domain [10]. The random walker is based on the Fokker-Planck equation written in \mathbb{R}^3

$$\begin{cases} \frac{\partial c(X,t)}{\partial t} + v(X) \nabla c(X,t) - D_m \Delta c(X,t) = 0 & \text{in } \mathbb{R}^3 \times [0, T] \\ c(X, 0) = c_0(X) & \text{in } \mathbb{R}^3. \end{cases} \quad (9)$$

The associated stochastic differential equation is

$$d\chi_t = v(\chi_t)dt + \sqrt{2D_m}dW_t, \quad (10)$$

where dW_t is a Wiener process in a probability space noted Θ (with random variable noted θ), and the law of χ_0 admits $c_0(X)$ as density, so that the law of χ_t admits $c(X, t)$ as density.

Using equation (10), the center of mass and spreading are now defined by

$$\begin{cases} G(\omega, t) = E_\theta[d\chi_t], \\ S(\omega, t) = E_\theta[(d\chi_t - G(\omega, t))(d\chi_t - G(\omega, t))^T]. \end{cases} \quad (11)$$

The random walker is based on the characteristics method for the drift (the advection) and on the Monte Carlo method for the brownian motion (molecular diffusion). Assuming that N_p particles are generated, the trajectory of each particle k is discretized by an explicit Euler scheme:

$$\begin{cases} \chi_k(t_{n+1}) = \chi_k(t_n) + v(\chi_k(t_n))\Delta t + \sqrt{2D_m\Delta t}\mathcal{N}_{k,n}, \\ \chi_k(0) = \chi_{k,0}. \end{cases} \quad (12)$$

where $t_{n+1} = t_n + \Delta t$ and $\mathcal{N}_{k,n}$ are independent standard normal vectors with a covariance identity. We use a local time step Δt for each particle, ensuring that the final position $\chi_k(t_{n+1})$ is either in the same cell as $\chi_k(t_n)$ or in an adjacent cell.

Since the trajectories of the particles are independent, the complexity is clearly linear with N_p . For a purely advective case, the method reduces to follow characteristic lines. If, moreover, the conductivity is homogeneous, the problem reduces to a 1D transport equation and the time for one particle is thus linear with $N_x = L_x/\Delta x$, leading to a global complexity proportional to $N_x \times N_p$. For heterogeneous or dispersive cases, the computing time is larger since trajectories are longer, so that the complexity is no longer linear with the longitudinal grid size.

3.6. Discrete spreading

For each realization, the computed trajectories are used to approximate the center of mass and the spreading defined in (11), by computing empirical Monte Carlo moments:

$$\begin{cases} \tilde{G}(\omega, t_n) = \frac{1}{N_p} \sum_{k=1}^{N_p} \chi_k(\omega, t_n), \\ \tilde{S}(\omega, t_n) = \frac{1}{N_p} \sum_{k=1}^{N_p} (\chi_k(\omega, t_n) - \tilde{G}(\omega, t_n))(\chi_k(\omega, t_n) - \tilde{G}(\omega, t_n))^T. \end{cases}$$

Now, to compute the macro spreading, we must approximate the expectation of the random spreading. For a Monte Carlo simulation l , we denote by $\tilde{S}(\omega_l, t_n)$ the computed spreading. We simply approximate the macro spreading defined in (6) by computing the empirical Monte Carlo mean $\tilde{S}(t_n)$ over N_s simulations:

$$\tilde{S}(t_n) = \frac{1}{N_s} \sum_{l=1}^{N_s} \tilde{S}(\omega_l, t_n). \quad (13)$$

An objective of this paper is to analyze experimentally the three diagonal elements, noted $\tilde{S}_x(t_n)$, $\tilde{S}_y(t_n)$, $\tilde{S}_z(t_n)$.

3.7. Numerical accuracy

Since we use approximation methods to solve the partial differential equations, we introduce several errors in the computations. Moreover, the Monte Carlo method gives only an approximation of the moments.

All these errors accumulate, with various constants which depend on the physical parameters. Moreover, the velocity in the advection phase is a random variable, which is approximated. Thus, a careful analysis must take

into account the impact of an approximation over another and the two probability spaces. A detailed numerical analysis of the whole numerical process is done in [8, 10]. Here, we briefly give the main results.

In general, the convergence of Monte Carlo simulations is driven by $1/\sqrt{N_s}$, and the convergence of the deterministic random walker (also a Monte Carlo method) is driven by $1/\sqrt{N_p}$. We assume here that the conductivity field is uniformly coercive and sufficiently smooth, with a parameter $0 < \alpha < 1$ of regularity. We also assume that the error of the finite volume method is driven by $\Delta x \log(\Delta x)$. The error in the numerical transport model is then driven by $\Delta t^{\frac{1+\alpha}{2}}$. Moreover, the global error at time T in the functional space $L^2(\Omega \times \Theta)$ can be estimated [8, 10]:

$$\|S(T) - \tilde{S}(T)\|_{L^2(\Omega \times \Theta)} \leq C(\Delta t^{\frac{1+\alpha}{2}} + \Delta x |\log(\Delta x)| + \frac{1}{\sqrt{N_p}} + \frac{1}{\sqrt{N_s}}),$$

where $S(T)$ is defined by (6) and $\tilde{S}(T)$ is defined by (13).

In this estimation, there is no assumption about ergodicity, so that the error bound contains the sum $(1/\sqrt{N_p} + 1/\sqrt{N_p})$. We expect that, thanks to the ergodic conditions, both the number of particles and the number of simulations will help in approximating the macro spreading. The underlying idea is that exploring the physical space with N_p or the probability space with N_s are more or less equivalent with ergodic conditions.

3.8. Numerical parameters

Name	Description	Value or range
N_s	number of simulations	[1 : 500]
Δx	dimension of one cubic cell	1
	size of injection window	$\Delta x \times 0.8L_y \times 0.8L_z$
N_p	number of particles	[500 : 50000]
Δt	timestep	dynamic local value

TABLE 2. Numerical parameters in the PARADIS model. Values are those used in numerical experiments.

The accuracy of simulations depends on various numerical parameters, which are summarized in Table 2.

The numerical parameter for uncertainty quantification is the number N_s of Monte Carlo simulations.

The main numerical discretization parameter for generating the conductivity field and computing the velocity field is the discrete structured grid. Here the grid is composed of cubic cells of size Δx .

When dealing with numerical complexity, the size N of the linear system, which is the total number of cells defined in (7), is the most important parameter.

In the transport equation, we use a local time step Δt , governed by the cell size Δx , the local velocity, and the molecular diffusion. Thus, the size of the injection window and the number of particles N_p are the most relevant parameters.

3.9. Parallel algorithms and numerical software

Because the domain is very large, simulations must rely on high performance computing. In order to cope with high memory and CPU requirements, parallel versions of the algorithms were designed and implemented in the parallel software PARADIS [20]. A distributed memory approach is used, with a message passing programming model.

The domain is decomposed into subdomains and ghost cells are used in order to communicate data at the interfaces. By using this domain decomposition and allocating one subdomain to each process, conductivity is generated in parallel and the matrix is built in parallel. Then, the distributed flow linear system is an input to a

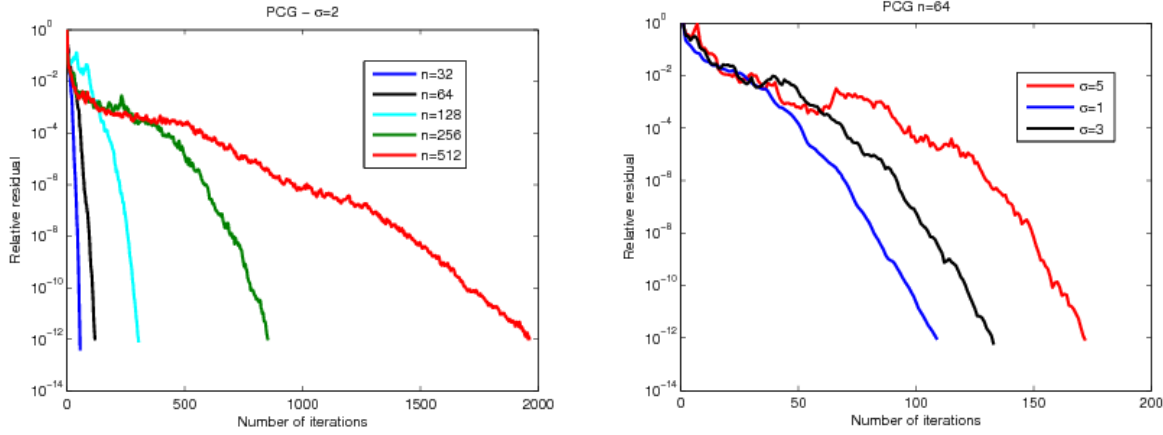


FIGURE 1. Relative residual versus the number of iterations using PCG for solving 2D flow problems. Left: fixed heterogeneity $\sigma = 2$, varying size $N = n^3$. Right: varying heterogeneity σ , fixed size $n = 64$.

parallel sparse linear solver, which may use internally another data distribution, but keeps the same distribution for output. The velocity is thus distributed among the processes, using the subdomains.

The parallel particle tracker of PARADIS takes advantage of this partition and also of the independance of the particles [6]. The parallel algorithm injects packets of particles, which are distributed to the processes according to their position. When the particle attains an interface, it is handled by the neighboring process. In order to reduce communications, all particles in the subdomain are treated before communicating. Some global communication must occur in order to stop correctly the algorithm.

The parallel algorithms keep the same complexity as the sequential ones. The CPU time is ideally divided by the number of processes, but overhead due to the ratio of sequential part (Amdahl's law) and to communications reduces the speed-up.

Another level of parallelism is easily handled with Monte Carlo simulations. A tool was designed in PARADIS in order to generate in parallel several conductivity fields and to run simultaneously several simulations, ensuring the generation of random numbers and gathering statistics [21]. The tool is also fault-tolerant with the possibility to interrupt Monte Carlo simulations. There is almost no overhead here, thus this parallel level should be preferred. The number of processes is fixed according to the memory requirements of one simulation and to the architecture.

4. NUMERICAL EXPERIMENTS

4.1. Choice of the sparse linear solver

The most CPU time consuming part of the PARADIS software is the sparse linear solver.

We discuss first some numerical experiments with Preconditioned Conjugate Gradient, preconditioned by incomplete Cholesky, for 2D domains. In Figure 1, we plot the convergence history of PCG for varying system size and several values of σ . Clearly, the number of iterations increases quite rapidly with heterogeneity and very rapidly with the size of the domain. Thus a multilevel approach is necessary to handle the large system size.

We thus switch to multigrid methods, using Hypr library, with SMG (structured or geometric) and AMG (algebraic) multigrid solvers. Experiments were done on a parallel cluster of the Grid'5000 computing resource installed at Inria in Rennes. In Figure 2, we plot the CPU time, with 16 processors, of SMG and AMG versus the system size, for two values of σ . We observe that, for almost homogeneous media, SMG is more efficient

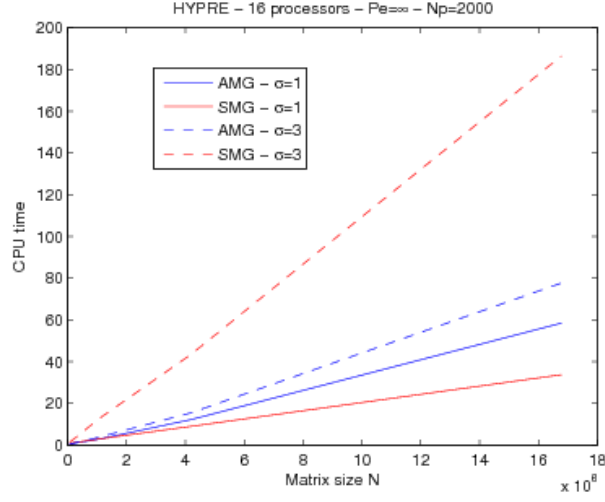
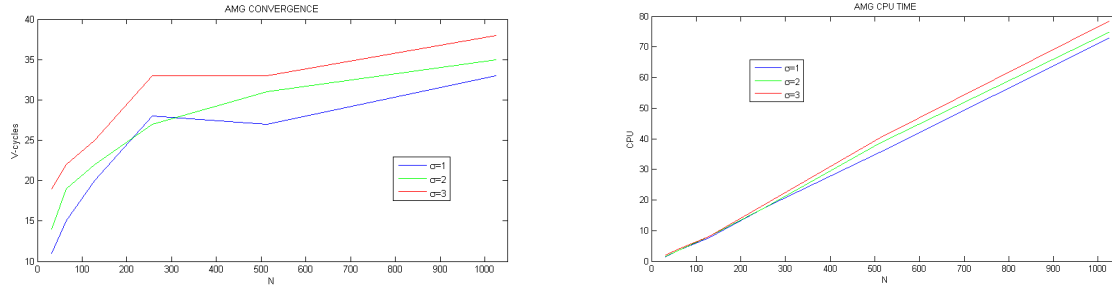


FIGURE 2. CPU time with 16 processors using SMG and AMG for solving 2D flow problems.

FIGURE 3. Number of cycles (left) and CPU time (right) versus the longitudinal grid size N_x , using AMG for solving small 3D flow problems.

than AMG; on the contrary, for highly heterogeneous media, AMG is more efficient than SMG. We also observe that the multigrid method is robust here and does not need to be combined with PCG.

Name	Value or range
σ	[1 : 3]
L_x	[32 : 1024]
L_y	32
L_z	32
$C(r)$	exponential
N_s	10

TABLE 3. Physical and numerical parameters used in numerical experiments for flow equations in small 3D domains.

In order to check if the same conclusions can be derived for 3D domains, numerical experiments are done with small 3D domains and AMG, on a sequential computer at Inria Rennes. The specific physical and numerical parameters are given in Table 3, and other parameters are those of Tables 1 and 2.

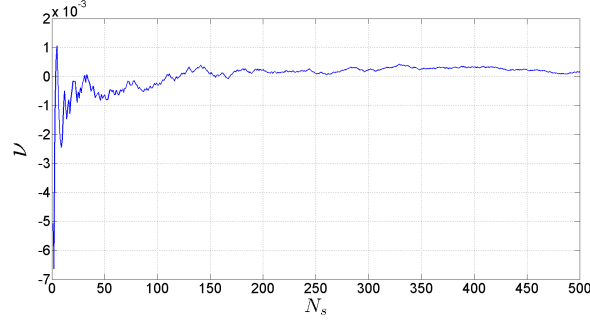


FIGURE 4. empirical mean ν of the conductivity field $Y = \log K$ (the exact mean is 0).

In Figure 3, we plot the CPU time and the number of cycles of AMG versus the longitudinal grid size N_x , for three values of σ . We check that, for 3D problems as for 2D problems, the CPU time is almost independent of the heterogeneity (the parameter σ), because the number of cycles is almost constant. We observe that the complexity is almost linear with the system size, as for 2D problems. Thus our choice of AMG is relevant for 3D problems.

4.2. Experiments with large 3D test cases: conductivity field

Name	Value or range
σ	1
L_x	1024
L_y	256
L_z	256
$C(r)$	Gaussian
N_s	[1 : 500]
N_p	[500 : 50000]

TABLE 4. Physical and numerical parameters used in numerical experiments with large 3D domains.

Now, we run numerical experiments with a large 3D domain. Physical and numerical parameters are given in Tables 1, 2, with specific values in Table 4. Thus $N = 2^{26} = 67108864$ for this large domain. Computations are done on a cluster located in Poitiers, using 128 cores. This cluster has 108 nodes of biprocessors AMD Opteron with 4 cores (2.3Ghz for the first group and 2.5Ghz for the second group), and with 16 Go of RAM. It runs a Linux system.

Generating the conductivity field is very fast and takes less than one second.

In order to validate the generated conductivity field, we analyze the convergence of the expectation and the variance versus the number of Monte Carlo simulations. Recall that the exact expectation of $Y = \log K$ is $\mu = 0$ and the exact variance is $\sigma = 1$.

We compute for each realization l , the discrete mean value of Y over the physical domain:

$$\bar{\mu}_l = \frac{1}{N} \sum_{i=1}^N Y_{i,l},$$

and the empirical Monte Carlo mean:

$$\nu = \frac{1}{N_s} \sum_{l=1}^{N_s} \bar{\mu}_l,$$

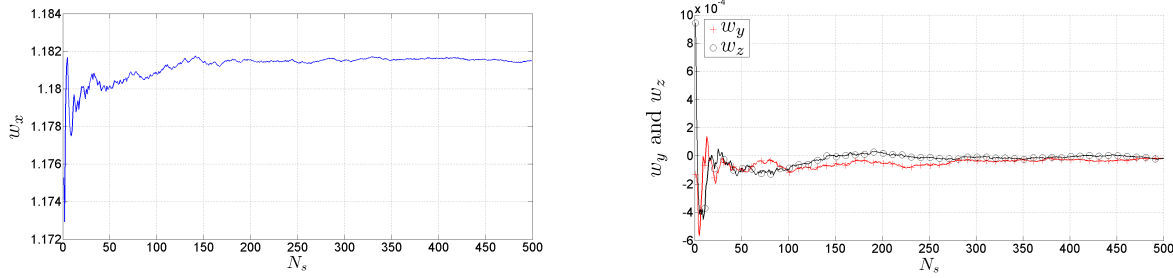


FIGURE 5. empirical mean of the spatial mean velocity field. Left: longitudinal mean velocity w_x . Right: transversal mean velocities w_y and w_z .

We check the property that $\lim_{N_s \rightarrow \infty} \nu = \mu$ and that this convergence is fast thanks to the ergodic properties of Y and to the large domain. In Figure 4, we plot ν versus the number N_s of simulations. Clearly, only a few Monte Carlo iterations are required to get an accurate value, close to 0.

We also compute the discrete variance of Y over the physical domain:

$$\bar{\sigma}_l = \frac{1}{N} \sum_{i=1}^N (Y_{i,l}^2 - \bar{\mu}_l^2).$$

With our method for generating the conductivity field, this value is independent of l [35]: $\bar{\sigma}_l = \bar{\sigma}$. Moreover, we get a small relative error $\frac{|\bar{\sigma} - \sigma|}{\sigma} = 4 \cdot 10^{-5}$, thanks to the large domain.

4.3. Experiments with large 3D test cases: flow

The residual in the multigrid algorithm is smaller than $tol = 10^{-12}$. We check that all systems are solved with a number of cycles in the range $[25 : 30]$ and that the CPU time for the flow computation is almost constant for all the random simulations, roughly equal to 130 seconds.

For each computed flow, we check mass conservation. The sum of fluxes through a cell is very small, of the order of tol , as well as the sum of fluxes through two opposite boundaries of the domain, thus our simulations ensure local and global mass conservation.

For each random realization l , we also compute the discrete equivalent conductivity $K_{eq,l}$ and the discrete mean velocity \bar{v}_l over the domain. We get as predicted $K_{eq,l} = \bar{v}_l$.

We compute the empirical Monte Carlo mean

$$w = \frac{1}{N_s} \sum_{l=1}^{N_s} \bar{v}_l,$$

and check convergence towards an expected value. In Figure 5, we plot the three components (w_x, w_y, w_z) versus the number of simulations. We observe convergence of both transversal mean velocities towards 0 and convergence of the longitudinal velocity towards about 1.182. Moreover, this convergence is quite fast.

4.4. Experiments with large 3D domains: spreading of pollutant

Another quantity of interest for physical studies is the macro spreading $S(t)$, defined in (6) and approximated in (13). Our objective is to get an asymptotic behavior when the simulated time becomes very large.

As illustrated in Figure 6, the CPU time for computing the macro-spreading is proportional to the number of particles and is almost independent of the random sample, as expected. Thus, for a grid of fixed size, the total time for N_s simulations can be roughly estimated as $N_s(\tau_1 + \tau_2 N_p)$, where τ_1 is the time for initializing/ending

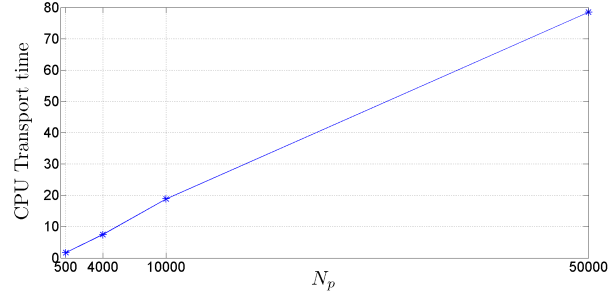
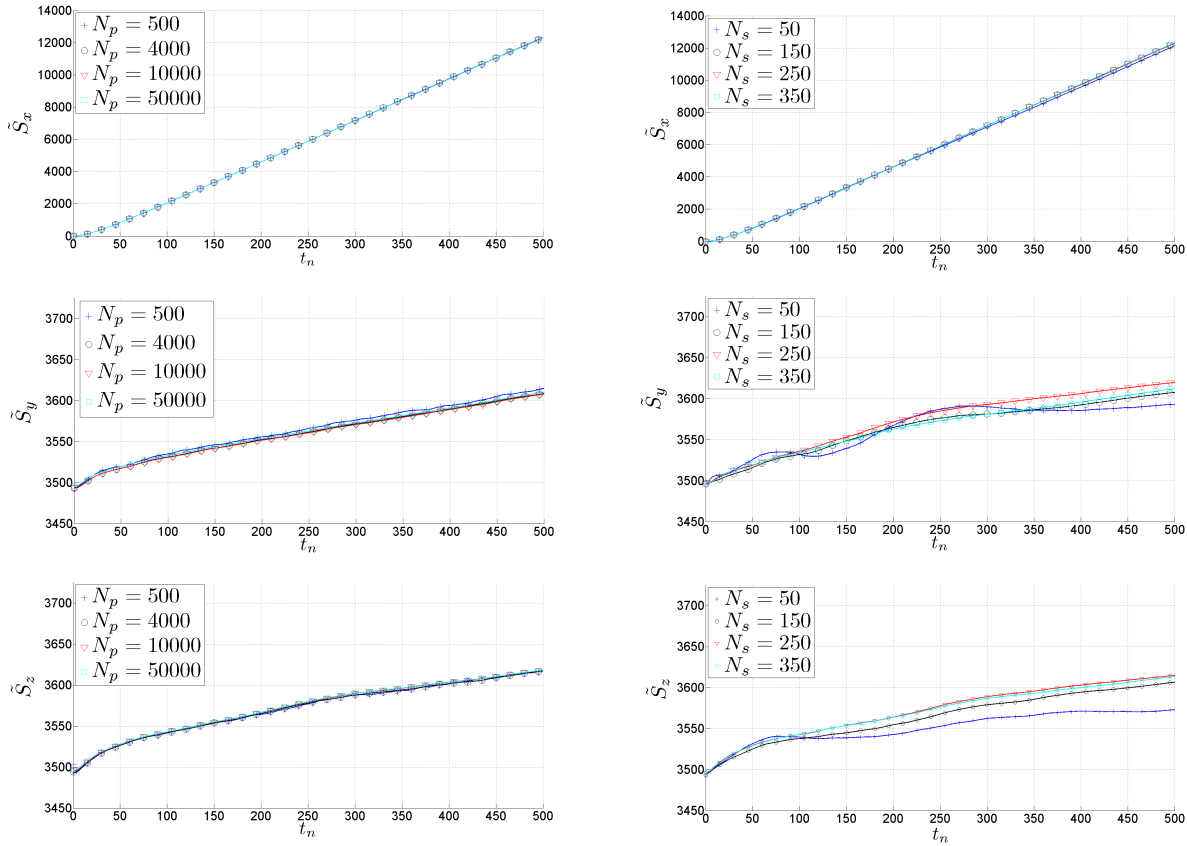


FIGURE 6. CPU time versus number of particles for the transport simulation.

FIGURE 7. Computed macro spreadings for various numbers of particles and Monte Carlo simulations. Top: $\tilde{S}_x(t_n)$, middle $\tilde{S}_y(t_n)$, bottom $\tilde{S}_z(t_n)$. Left: $N_s = 499$, right: $N_p = 50000$.

the simulation and for flow computation, and τ_2 is the time by particle for transport computation. For a given total time, it is thus possible to run either many simulations with a few particles or a few simulations with many particles.

In theory, as stated in [8], convergence of the macro spreading is governed by the sum $(1/\sqrt{N_p} + 1/\sqrt{N_s})$. Here, we analyze numerically convergence with the Monte Carlo simulations and the number of particles.

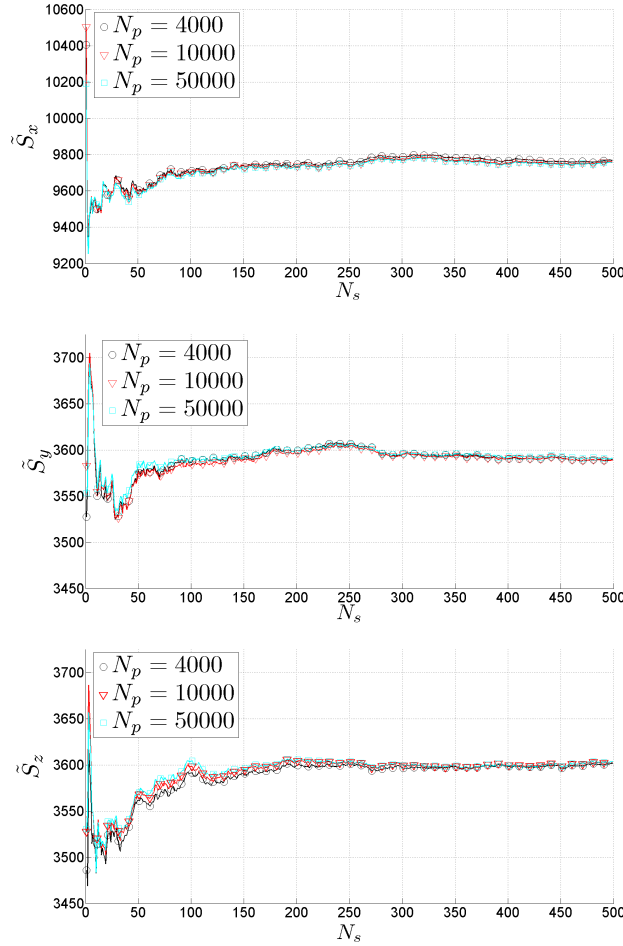


FIGURE 8. Computed macro spreadings versus the number of random simulations at the given time $t = 400$. Top: $\tilde{S}_x(t_n)$; middle: $\tilde{S}_y(t_n)$; bottom: $\tilde{S}_z(t_n)$.

In Figure 7, we plot the computed macro spreadings $\tilde{S}_x(t_n)$, $\tilde{S}_y(t_n)$, $\tilde{S}_z(t_n)$, for a various number of particles N_p and a various number of simulations N_s . On the left, we choose $N_s = 499$ and plot four curves for $N_p = 500, 4000, 10000, 50000$. On the right, we fix $N_p = 50000$ and plot four curves for $N_s = 50, 150, 250, 350$.

We observe an almost linear behavior for $t \geq 300$, showing an asymptotic limit of the macro dispersion (defined as the time derivative of the macro spreading). In order to analyze the asymptotic behavior, we plot, in Figure 8, the computed macro spreadings $\tilde{S}_x(t_n)$, $\tilde{S}_y(t_n)$, $\tilde{S}_z(t_n)$, at the given time $t_n = 400$, versus the number of Monte Carlo simulations. In both Figures 7 and 8, we observe that the macro spreadings converge quite rapidly when increasing the number of simulations, provided that the number of particles is large enough. This saves CPU time since each new sample requires computing a velocity field, thus solving a large linear system.

Thus, it seems that convergence of macro spreading is governed by N_p and N_s in a more efficient way than predicted by the theory. We assume that this behavior is related to the ergodic properties of the physical data.

Thanks to the software PARADIS, it is possible to study in 3D domains the macro dispersion and its asymptotic behavior. The impact of the heterogeneity can be well shown, by varying the variance σ and the numerical results can be compared with those from a perturbation theory [5].

5. CONCLUSION

In this paper, we consider the flow and the transport of an inert solute in highly heterogeneous porous media, where the conductivity is given by a random field for dealing with the lack of data. We consider a log normal probability distribution with an exponential or Gaussian covariance function. The numerical model combines a Monte Carlo method for uncertainty quantification, a discrete random field generation by a spectral method, a mixed finite element method for the flow equation and a random walker method for the transport equation. Numerical experiments are done for the first time with large 3D domains, and a Gaussian covariance function with a small correlation length. Simulations show a fast convergence of Monte Carlo simulations, with a relatively large grid size and a relatively large number of particles. Large scale parallel simulations run on multicore systems with a reasonable computational cost and allow characterizing the asymptotic behavior of the plume. We plan to further analyze the convergence of macro spreading and to highlight theoretically the effect of ergodicity.

REFERENCES

- [1] D. Allard. Statistiques spatiales : introduction à la géostatistique. Lecture notes, University of Montpellier, 2012.
- [2] I. Babuska, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Review*, 52(2):317–355, 2010.
- [3] I. Babuska, R. Tempone, and G. Zouraris. Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. *Computer methods in applied mechanics and engineering*, 194:1251–1294, 2005.
- [4] A. Barth, C. Schwab, and N. Zollinger. Multi-level monte carlo finite element method for elliptic pdes with stochastic coefficients. *Numer. Math.*, 119(1):123–161, 2011.
- [5] A. Beaudoin and J.-R. de Dreuzy. Numerical assessment of 3D macrodispersion in heterogeneous porous media. *WRR*, 49:2489–2496, 2013.
- [6] A. Beaudoin, J.-R. de Dreuzy, and J. Erhel. An efficient parallel particle tracker for advection-diffusion simulations in heterogeneous porous media. In A.-M. Kermarrec, L. Bougé, and T. Priol, editors, *Euro-Par 2007, LNCS 4641*, pages 717–726. Springer-Verlag, Berlin, Heidelberg, 2007.
- [7] A. Beaudoin, J. Erhel, and J.-R. de Dreuzy. A comparison between a direct and a multigrid sparse linear solvers for highly heterogeneous flux computations. In *Eccomas CFD 2006*, volume CD, 2006.
- [8] J. Charrier. Numerical analysis of the advection-diffusion of a solute in random media. Research Report RR-7585, INRIA, March 2011.
- [9] J. Charrier. Strong and weak error estimates for elliptic partial differential equations with random coefficients. *SIAM Journal on numerical analysis*, 50(1):216–246, 2012.
- [10] J. Charrier. Numerical analysis of the advection-diffusion of a solute in porous media with uncertainty. Technical report, University of Aix-Marseille, 2013.
- [11] J. Charrier and A. Debussche. Weak truncation error estimates for elliptic pdes with lognormal coefficients. *Stochastic Partial Differential Equations : Analysis and Computations*, 1:63–93, 2013.
- [12] J. Charrier, R. Scheichl, and A. Teckentrup. Finite element error analysis for elliptic pdes with random coefficients and applications. *SIAM Journal on Numerical Analysis*, 51:322–352, 2013.
- [13] G. Dagan, A. Fiori, and I. Jankovic. Flow and transport in highly heterogeneous formations: 1. conceptual framework and validity of first-order approximations. *Water Resources Research*, 9, 2003.
- [14] J.-R. de Dreuzy, A. Beaudoin, and J. Erhel. Asymptotic dispersion in 2D heterogeneous porous media determined by parallel numerical simulations. *Water Resource Research*, 43(W10439, doi:10.1029/2006WR005394), 2007.
- [15] J.-R. de Dreuzy, A. Beaudoin, and J. Erhel. Reply to comment by a. fiori et al. on "asymptotic dispersion in 2d heterogeneous porous media determined by parallel numerical simulations". *Water Resources Research*, 44(W06604, doi:10.1029/2008WR007010), 2008.
- [16] Ghislain de Marsily, Frédéric Delay, J. Gonçalves, Philippe Renard, Vanessa Teles, and S. Violette. Dealing with spatial heterogeneity. *Hydrogeology Journal*, 13:161–183, 2005.
- [17] C.R. Dietrich and G.N. Newsam. Fast and exact simulation of stationary gaussian processes through circulant embedding of the covariance matrix. *SIAM J. Sci. Comput.*, 18:1088–1107, 1997.
- [18] J. Erhel. Stochastic groundwater simulations for highly heterogeneous porous media. In B. Amaziane, D. Barrera, M. Fortes, M. Ibanez, M. Odunlami, A. Palomares, M. Pasadas, M. Rodriguez, and D. Sbibi, editors, *Proceedings of the third international conference on approximation methods and numerical modelling in environment and natural resources, MAMERN'09*, volume 1, pages 419–422. EUG, 2009. invited plenary talk.
- [19] J. Erhel. *Computational Technology Reviews*, volume 3, chapter Some Properties of Krylov Projection Methods for Large Linear Systems, pages 41–70. Saxe-Coburg Publications, 2011.
- [20] J. Erhel, J.-R. de Dreuzy, A. Beaudoin, E. Bresciani, and D. Tromeur-Dervout. A parallel scientific software for heterogeneous hydrogeology. In Ismail H. Tuncer, Ulgen Gulcat, David R. Emerson, and Kenichi Matsuno, editors, *Parallel Computational Fluid Dynamics 2007*, volume 67 of *Lecture Notes in Computational Science and Engineering*, pages 39–48. Springer, 2009. invited plenary talk.
- [21] J. Erhel, J.-R. de Dreuzy, and E. Bresciani. Multi-parametric intensive stochastic simulations for hydrogeology on a computational grid. In D. Tromeur-Dervout, G. Brenner, D. Emerson, and J. Erhel, editors, *Parallel Computational Fluid Dynamics 2008*, Lecture Notes in Computational Science and Engineering (LNCSE), Lyon, May 2010. accepted contribution.
- [22] J. Erhel, Z. Mghazli, and M. Oumouni. Calcul de l'espérance de la solution d'une edp stochastique unidimensionnelle à l'aide d'une base réduite. *Comptes Rendus de l'Académie des Sciences de Paris (CRAS), série I*, 349:861–865, 2011.
- [23] R.D. Falgout, J.E. Jones, and U.M. Yang. *Numerical Solution of Partial Differential Equations on Parallel Computers*, chapter The Design and Implementation of Hypre, a Library of Parallel High Performance Preconditioners, pages 267–294. Springer-Verlag, 2006.
- [24] P. Frauenfelder, C. Schwab, and R.A. Todor. Finite elements for elliptic problems with stochastic coefficients. *Comput. Methods Appl. Mech. Engrg.*, 194:205–228, 2005.
- [25] L. Gelhar. *Stochastic Subsurface Hydrology*. Engelwood Cliffs, New Jersey, 1993.
- [26] A. George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10:345–363, 1973.

- [27] R. G. Ghanem and P. D. Spanos. *Stochastic finite elements : a spectral approach*. Springer, 1991.
- [28] I. Graham, F. Kuo, D. Nuyens, R. Scheichl, and I. Sloan. Quasi-monte carlo methods for elliptic pdes with random coefficients and applications. *Journal of Computational Physics*, 230:3668–3694, 2011.
- [29] H. Hoteit, P. Ackerer, R. Mosé, J. Erhel, and B. Philippe. New two-dimensional slope limiters for discontinuous galerkin methods on arbitrary meshes. *International Journal of Numerical Methods in Engineering*, 61:2566–2593, 2004.
- [30] H. Hoteit, J. Erhel, R. Mosé, B. Philippe, and P. Ackerer. Numerical reliability for mixed methods applied to flow problems in porous media. *Computational Geosciences*, 6:161–194, 2002.
- [31] O.P. Le Maître and O.M. Knio. *Spectral Methods for Uncertainty Quantification With Applications to Computational Fluid Dynamics*. Scientific Computation. Springer, 2010.
- [32] G. Meurant. *Computer solution of large linear systems*. North Holland, Amsterdam, 1999.
- [33] M. Oumouni. *Analyse numérique de methodes performantes pour les EDP stochastiques modélisant l'écoulement et le transport en milieux poreux*. PhD thesis, University of Rennes 1 and University of Kenitra, 2013.
- [34] E. Pardo-Iguzquiza and M. Chica-Olmo. The Fourier integral method: an efficient spectral method for simulation of random fields. *Mathematical Geology*, 25(2):177–217, 1993.
- [35] G. Pichot. About the generation of a log-normal correlated field using spectral simulation. Technical report, Inria, 2012.
- [36] F. Ramasomanana and A. Younes. Efficiency of the eulerian lagrangian localized adjoint method for solving advection-dispersion equation on highly heterogeneous media. *International Journal for Numerical Methods in Fluids.*, 69:639–652, 2012.
- [37] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2003.
- [38] P. Salandin and V. Fiorotto. Solute transport in highly heterogeneous aquifers. *Water Resources Research*, 34:949–961, 1998.
- [39] P. Wesseling. *An Introduction to Multigrid Methods*. Edwards, 2004.
- [40] D. Xiu. Fast numerical methods for stochastic computations:a review. *Communications in Computational Physics*, 5:242–272, 2009.
- [41] T. Yao. Reproduction of the mean, variance, and variogram model in spectral simulation. *Mathematical Geology*, 36:487–506, 2004.
- [42] D. Zhang. *Stochastic Methods for Flow in Porous Media: Coping with Uncertainties*. Academic Press, San Diego, 2002.
- [43] C. Zheng and G. D. Bennett. *Applied Contaminant Transport Modeling; second edition*. John Wiley & Sons, New-York, 2002.