Simulating diffusions with piecewise constant coefficients using a kinetic approximation

Antoine Lejay, Sylvain Maire

SIMULATING DIFFUSIONS WITH PIECEWISE CONSTANT COEFFICIENTS USING A KINETIC APPROXIMATION

ANTOINE LEJAY AND SYLVAIN MAIRE

Abstract. Using a kinetic approximation of a linear diffusion operator, we propose an algorithm that allows one to deal with the simulation of a multi-dimensional stochastic process in a media which is locally isotropic except on some surface where the diffusion coefficient presents some discontinuities. Basic numerical examples are given in dimensions one to three on PDEs or stochastic PDEs with or without source terms. Finally, we compute the hydrodynamic load in a porous media in the nuclear waste context.

1. Introduction

In this article, we address the problem of the simulation of a diffusion process in a discontinuous media. More precisely, we consider a divergence-form operator of type

$$L = \frac{1}{2}\nabla \cdot (a\nabla \cdot)$$

where a is a piecewise constant anisotropic diffusion coefficient, and we identify $a(x)$ with a scalar.

This kind of problem arises in a lot of modelling problems, such as molecular electrostatics [36], geophysics [40], ecology [7, 39], astrophysics [43], magneto/electro-encephalography (MEG, EEG) [19],...

One could be tempted to use a Monte Carlo method to solve linear parabolic or elliptic problems involving such an operator L, especially when one faces complicated and/or infinite domains and Dirac masses in the source terms as in the MEG/EEG problems or in molecular dynamics or electrostatics.

From a probabilistic point of view, this problem has been hardly treated, except in dimension one where the diffusion process generated by L is solution to some stochastic differential equation involving the local time of the process. See however, a recent article from N. Limic [31]. Several algorithms have then been proposed to simulate one-dimensional diffusion process with discontinuous coefficients [13, 16, 17, 18, 30, 34, 35].

Date: July 30, 2009.

Key words and phrases. Discontinuous media, divergence-form operator, stochastic process, kinetic approximation, Monte Carlo method.

Using the local property of the operator, the problem consists in finding a correct approximation of the process when it is close to a surface of discontinuity.

One knows that in general, the diffusion behavior of a physical model may be derived from a change of scale of a transport equation. This is the case for example for the Darcy law in geophysics [9], or for the displacement of bacterium such as *E. Coli* [3]. It is in general convenient to use the diffuse limit of transport equations, for example to characterize the criticality of the neutron transport equation, as it leads to simpler computations [2]. Rigorous development for this approximation may be found for example in [2, 10, 22]. Here, we propose to perform the opposite approach. In order to deal with this discontinuity, we use a small parameter ε approximation of the operator in divergence form by a transport operator. When the diffusion coefficient is discontinuous, there is no simple way to write the dynamic of the particle. Reducing the problem to a transport problem leads to a very simple simulation algorithm, since the particle moves in a given direction at a given speed until the next collision which changes its direction. The problem is then reduced to prescribe the manner the direction is changed at each collision and to prescribed the way to change the next collision time when the particle reaches a zone with a different diffusion coefficient.

First we make a global approximation of L by combining an exact simulation (See [32, 33, 41]) and a Romberg acceleration procedure. Then, we use only the approximation by the transport operator when one hits the surface of discontinuity of a and efficient simulations of the Brownian motion elsewhere. We describe the global approximation procedure in Section 2 and the local one in Section 3. In these two sections, we also make some comparisons with existing Monte Carlo methods. In the next three sections, we make various numerical tests from basic one dimensional problems in Section 4 to three dimensional stochastic PDEs in Section 6.

In section 7, we study a more realistic problem taken from the couplex exercises [4]. It consists in computing the hydro-dynamic load in a two dimensional porous media constituted of 4 very heterogeneous physical regions.

2. Global approximation of a diffusion by a transport operator

2.1. **The Feynman-Kac formula.** In this section, we consider the Dirichlet problem

$$\frac{1}{2}\nabla \cdot (a\nabla u) = f \text{ on } D \text{ with } u_{\partial D} = g \tag{1}$$

in a domain $D \subset \mathbb{R}^d$ with piecewise smooth boundary, and a is a piecewise smooth real-valued function which satisfies $0 < \lambda \leq a(x) \leq$

Λ, $x \in D$ for some positive constants λ and Λ. Let X denote the process generated by $\frac{1}{2}\nabla \cdot (a\nabla \cdot)$ (On the existence of such a process, see [27, 42]). As for the case of smooth coefficients, it can be shown that if g and f are continuous, the solution u to (1) is still given by the Feynman-Kac formula

$$u(x) = \mathbb{E}_x \left[g(X_\tau) + \int_0^\tau f(X_s)\,\mathrm{d}s \right],$$

where τ is the first exit time from τ. This follows simply for a regularization argument, since if $(a^n)_{n\in\mathbb{N}}$ is a family of smooth approximation of a, then the corresponding processes $(X^n)_{n\in\mathbb{N}}$ converges in distribution to X [42]. Besides, the corresponding solutions $(u^n)_{n\in\mathbb{N}}$ to (1) with a replaced by u^n also converges in $\mathrm{L}^2(D)$ and locally uniformly to u. This last statement follows from standard computations and the Harnack inequality which implies that u^n is locally Hölder continuous with a Hölder constant that depends only on the upper and lower bound on a [21].

Unlike the case of non-divergence form operators, for a general coefficient a, there is in general no simple formula describing the dynamics of the process X such as a Stochastic Differential Equation (SDE) which could be used for simulation purpose.

In our context, without loss of generality, we can assume that $d = 2$, that the domain can be divided in two pieces $D_+ = [0, L] \times [-\ell, \ell]$ and $D_- = [-L, 0] \times [-\ell, \ell]$ and that $a = a_1$ on D_+ and $a = a_2$ on D_-. The process X is then solution to the generalized SDE

$$\begin{cases} X_t^1 = x^1 + \int_0^t \sigma(X_s)\,\mathrm{d}B_s^1 + \frac{a_1 - a_2}{a_1 + a_2} L_t^0(X^1), \\ X_t^2 = x^2 + \int_0^t \sigma(X_s)\,\mathrm{d}B_s^2, \end{cases} \tag{2}$$

where (B^1, B^2) is a two-dimensional Brownian motion, $L_t^0(X^1)$ is the symmetric local time of X^1 around 0, $\sigma(x) = \sqrt{a_1(x)}$ if $x \geq 0$ and $\sigma(x) = \sqrt{a_2(x)}$ if $x < 0$.

The symmetric local time $L_t^0(X)$ is by definition

$$L_t^0(X) = \lim_{\epsilon \to 0} \frac{1}{2\epsilon} \int_0^t \mathbf{1}_{X_s \in [-\epsilon, \epsilon]} a(X_s)\,\mathrm{d}s$$

and characterizes the time spent by the process around 0. Note that the process is continuous, non-decreasing but almost everywhere constant! Such a process is difficult to simulate.

Eq. (2) is obtained by a simple generalization of the computations of [27] (See also [30] for example) which are themselves relying on arguments of J.-F. Le Gall to prove existence and uniqueness of solutions of SDE with local time [26]. The core argument is to use a regularization procedure by smoothing the coefficients and a clever change of variable to transform (2) into a SDE without local time.

2.2. **Approximation by a kinetic equation.** In many situations, diffusion equations are used to approximate transport equations. This corresponds to a change of scale and is helpful to reduce the number of variables of the unknown solution of the related PDEs. For the neutron transport problem, only the position is computed using a diffusion approximation while the original problem requires to compute both the velocity and the position of the particles.

Here, we use this approximation in the opposite way. The diffusion process is approximated by a transport process depending on a small parameter $\varepsilon > 0$. If $d \geq 2$, we denote by $\mathbb{V}$ the unit sphere of $\mathbb{R}^d$ and the interval $[-1,1]$ if $d = 1$. Let us consider the integral operator

$$Ku(x,v) = \frac{1}{\mathrm{Vol}(\mathbb{V})} \int_{\mathbb{V}} u(x,v')\mathrm{d}v' \text{ with } \mathrm{Vol}(\mathbb{V}) = \int_{\mathbb{V}} \mathrm{d}v$$

on functions defined on $D \times \mathbb{V}$. Let us note however that $Ku(x,v) = Ku(x,v')$ for any $v \neq v'$ but, for notational consistency, Ku is considered also as a function on $D \times \mathbb{V}$.

The solution u of the PDE (1) may be approximated by the solution $u_\varepsilon(x,v)$ of the equation

(3)
$$\begin{cases} -\varepsilon^{-1} v \cdot \nabla_x u_\varepsilon(x,v) + \dfrac{1}{C_{\mathbb{V}} \varepsilon^2 a(x)} (K u_\varepsilon(x,v) - u_\varepsilon(x,v)) = f(x), \\ u_\varepsilon(x,v) = g(x) \text{ when } v \cdot \vec{n}(x) \leq 0, \ x \in \partial D \end{cases}$$

solved on $D \times \mathbb{V}$ where $C_{\mathbb{V}}$ is a constant that depends on $\mathbb{V}$ and $\vec{n}(x)$ is the inner unit vector at the boundary point $x \in \partial D$.

The value of the constant $C_{\mathbb{V}}$ is obtained by solving an integral equation [2, 10]. In the case of $L = \frac{1}{2}\triangle$, it is easy to prove that $C_{\mathbb{V}} = 3/2$ in dimension 1 and 3 and $C_{\mathbb{V}} = 1$ in dimension 2.

The heuristic idea of the approximation is the following [2]: We assume u_ε can be expanded as

$$u_\varepsilon(x,v) = u_0(x,v) + \varepsilon u_1(x,v) + \varepsilon^2 u_2(x,v) + \psi_\varepsilon(x,v) \tag{4}$$

where $\psi_\varepsilon(x,v) = \mathrm{O}(\varepsilon^3)$. Injecting this development in (3) leads to conclude that $u_0(x,v)$ depends only on x, and

$$C_{\mathbb{V}} a(x) v \cdot \nabla_x u_0(x) = K u_1(x,v) - u_1(x,v), \tag{5}$$

$$-v \cdot \nabla_x u_1(x,v) + \frac{1}{C_{\mathbb{V}} a(x)} (K u_2(x,v) - u_2(x,v)) = f(x). \tag{6}$$

Let us assume that x is in a zone where a is smooth. Thus differentiating (5) in the direction x_i, multiplying it by $D_i(v) = (v \cdot e_i) e_i$, where $\{e_1, \ldots, e_d\}$ is the canonical basis of $\mathbb{R}^d$, and summing in all the directions,

$$\sum_{i,j=1}^{d} (v \cdot e_i)(v \cdot e_j) C_{\mathbb{V}} \frac{\partial}{\partial x_i} \left(a(x) \frac{\partial u_0(x)}{\partial x_j} \right) = v K \nabla_x u_1(x,v) - v \nabla_x u_1(x,v).$$

Applying K to this equation leads to

$$\sum_{i,j=1}^{d}(v\cdot e_i)(v\cdot e_j)C_{\mathbb{V}}\frac{\partial}{\partial x_i}\left(a(x)\frac{\partial u_0(x)}{\partial x_j}\right)=-K(v\nabla_x u_1(x,v))$$

since $Ku_1(x,v)$ does not depend on v and $Kv=0$. Let us note that $K((v\cdot e_i)(v\cdot e_j))=0$ if $i\neq j$ and $\alpha=K((v\cdot e_i)^2)$ does not depend on $i\in\{1,\dots,d\}$ and $d\alpha=1$ for $d\geq 2$. On the other hand, applying K to the both sides of (6) leads to, since $K^2=K$ and $Kf(x)=f(x)$,

$$\alpha C_{\mathbb{V}}\nabla\cdot(a(x)\nabla u_0(x))=f(x). \tag{7}$$

One deduces from (7) that $u_0(x)$ is the solution to (1) in any subdomain where a is smooth if $C_{\mathbb{V}}=1/2\alpha$. It is then possible to compute $C_{\mathbb{V}}$ and to deduce that in the case of smooth coefficients [10, Theorem 2, p. 1238]:

$$\|u_\varepsilon-u\|_{\mathrm{L}^\infty(D\times\mathbb{V})}\leq\varepsilon C_{g,f} \tag{8}$$

for some constant $C_{g,f}$ that depends on g and f.

The problem of the approximation of the neutron equation by the diffusion seems to have never been treated, and in addition, one knows that $u(x,v)$ may be discontinuous. However, we give a formal reasoning on why it can be expected that the kinetic approximation also holds around discontinuities. We have seen that $u_0(x)$ is solution to (7) for any point x where $a(x)$ is smooth. It remains to prove that u_0 is a global solution to (1) on the whole domain D, which can be tackled by specifying the behavior of u on the boundary. Provided that f is continuous, it is also known that the solution u of (1) is a classical solution on open domains subsets of D where $a(x)$ is of smooth, and that at any surface of discontinuity, both u and its normal flux $\vec{n}\cdot a(x)\nabla u(x)$ are continuous [25], where $\vec{n}$ is the normal vector to the surface of discontinuity at the point x. As u_ε and $v\cdot\nabla u_\varepsilon(x,v)$ belongs to $\mathrm{L}^\infty(D\times\mathbb{V})$, one deduces that $t\in(-\eta,\eta)\mapsto u_\varepsilon(x+t\vec{n},v)$ has a continuous version. Then u_0 and u_1 are also locally continuous with respect to x on any segment which crosses a surface of discontinuity and (5) implies the continuity of the normal flux at any surface of discontinuity. It follows that $u_0(x)$ is solution to (1).

The solution u_ε can also be represented using a Feynman-Kac formula

$$u_\varepsilon(x,v)=\mathbb{E}_{x,v}\left[g(Y_\tau)+\int_0^\tau f(Y_s,V_s)\,\mathrm{d}s\right]$$

where τ is the first exit time from $D\times\mathbb{V}$ for the process (Y,V) whose dynamic is the following: Y is solution to $\frac{\mathrm{d}Y_t}{\mathrm{d}t}=V_t$ and the random velocity V of the particle is constant between random collision times.

When a collision occurs, we draw a new velocity $V=V_0/\varepsilon$ where V_0 is chosen uniformly in $\mathbb{V}$. The next collision time is also chosen randomly using an exponential random variable of parameter $a(y_0)\varepsilon^2C_{\mathbb{V}}$, where y_0

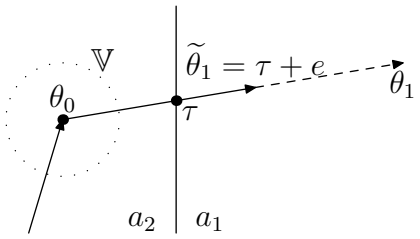

FIGURE 1. Changing the collision time.

is the current position of the particle. Hence, one can compute easily the position y_1 of the particle at the next collision time. If the position is outside D, we stop the particle when it crosses the boundary of D. If the particle remains in zone where a is constant, then a new position and time collision is drawn at the next collision time.

Because of the heterogeneity of a, the main difficulty is the simulation of the transport process when the neutron goes from one zone to another.

If between two successive collision times θ_0 and θ_1, the particle crosses at some time τ the boundary between two zones with different diffusion coefficients a_1 and a_2, then the collision time θ_1 is replaced by a new collision time $\widetilde{\theta}_1 = \tau + e$, where e is an exponential random variable with parameter $a_1\varepsilon^2 C_{\mathbb{V}}$ (See Figure 1). For more details, check [32, 33].

Remark 1. We could also proceed the following way: we choose V uniformly in $\mathbb{V}$ and θ as an exponential time θ of parameter $a(y_0)C_{\mathbb{V}}$. Then the new position is $y_0 + \varepsilon V\theta$ and the time is incremented by $\theta\varepsilon^2$.

2.3. **Romberg extrapolation.** We have seen in (8) that

$$\|u_\varepsilon - u\|_{L^\infty(D\times\mathbb{V})} \leq \varepsilon C_{g,f}$$

where $C_{g,f}$ depends on g and f but not on ε. As in addition u_ε has the asymptotic development (4), $u_\varepsilon = u_0 + \varepsilon u_1 + \varepsilon^2 u_2 + \psi_\varepsilon$ where u_0, u_1, u_2 and ψ_ε are solution of some other transport equations, it is possible to consider a Romberg extrapolation to get a better approximation of u. For this, we just write the two following approximations:

$$u_\varepsilon^{(2)} = 2u_{\frac{\varepsilon}{2}} - u_\varepsilon \text{ and } u_\varepsilon^{(3)} = \frac{1}{3}(8u_{\frac{\varepsilon}{4}} - 6u_{\frac{\varepsilon}{2}} + u_\varepsilon),$$

which are respectively of order 2 and 3. These two approximations are crucial in the numerical simulations because they allow not to take the parameter ε too small. Indeed the CPU times increase very quickly with this parameter.

3. A local approximation by a kinetic equation

3.1. **The algorithm.** Using the Markov property of the stochastic process X, the kinetic approximation may be used only *locally* around

the discontinuity. Let us denote by $\{Z_i\}_i$ the family of open subsets of D on which the diffusion coefficient is constant. We write $a(x) = a_i$ on Z_i. The set of points on which a is discontinuous is denoted by S.

For one particle, the algorithm becomes then the following, where $\varepsilon > 0$ is a fixed real.

(1) *Initialization:* Set $F \leftarrow 0$, $t \leftarrow 0$.
(2) If the particle is at a position $x \in Z_i$ then simulate the first time τ at which X starting from x reaches $S \cup \partial D$ for the first time, as well as an approximation $\widetilde{F}$ of $\int_0^\tau f(X_s)\,\mathrm{d}s$. Set $F \leftarrow F + \widetilde{F}$ and $t \leftarrow t + \tau$.
(3) If the particle belongs to ∂D, then returns $g(X_\tau) + F$.
(4) If the particle $y = X_\tau$ belongs to S, then draw uniformly a direction v on $\mathbb{V}$. If $y + \eta v$ belongs to Z_j for $\eta > 0$ small enough, then draw an exponential time θ of parameter $a_i C_{\mathbb{V}}$.
(5) If $(y + \varepsilon v s)_{s\in[0,\theta]}$ does not remain in D, then compute the time τ at which $y+\varepsilon v\tau$ belongs to ∂D and return $g(y+\varepsilon v\tau)+F+\widetilde{F}$, where $\widetilde{F}$ is an approximation of $\int_0^\tau f(y + \epsilon v s)\,\mathrm{d}s$.
(6) If $(y+\varepsilon v s)_{s\in[0,\theta]}$ crosses another zone Z_i, then change θ according to the rule given in Section 2.2. Possibly, the particle may cross several zones, and which case, θ should be changed each time the particle enter a new zone. Check again if $(y+\varepsilon v s)_{s\in[0,\theta]}$ remains in D. If it is not the case, then go to Step 5. Otherwise, compute an approximation $\widetilde{F}$ of $\int_0^{\theta\varepsilon^2} f(y + vs)\,\mathrm{d}s$ and set $F \leftarrow F + \widetilde{F}$, $t \leftarrow t + \varepsilon^2\theta$. Go back to Step 2 with $x \leftarrow y + \varepsilon v\theta$.

For a particle, this algorithm returns an approximation G of the quantity $g(X_\tau) + \int_0^\tau f(X_s)\,\mathrm{d}s$ for the process X generated by L. In order to get an approximation of $\mathbb{E}_x\left[g(X_\tau) + \int_0^\tau f(X_s)\,\mathrm{d}s\right]$, where $\mathbb{P}_x$ is the distribution of the particle starting from x, one needs to run this algorithm N times with independent random variables in order to get a family $G_1, \ldots, G_N$ of approximations of realizations of $g(X_\tau) + \int_0^\tau f(X_s)\,\mathrm{d}s$ and then to compute $N^{-1}\sum_{i=1}^N G_i$.

In Step (2), one needs to compute the first exit time and position from a domain where the diffusion coefficient is constant. Several methods can be used to simulate the time the particle reaches the surface S or the boundary, among which the Euler scheme [24], the random walks on spheres [38] and its variants, the random walk on squares [20, 37] and on rectangles [11, 12]. The choice of one of the method depends on the geometry of the problem.

Remark 2. As presented, this algorithm is used to computed an approximation of $\mathbb{E}_x\left[g(X_\tau) + \int_0^\tau f(X_s)\,\mathrm{d}s\right]$. Note that however, we may return (τ, X_τ) instead, and then obtain an approximation of the density of the first exit time τ or of the exit position X_τ.

3.2. **An alternative method derived from finite differences.** In [36], M. Mascagni and N. Simonov proposed another way to solve a PDE of type (1). When the particle reaches a point x of S that separates two zones with diffusion coefficients a_1 and a_2, it is reinjected at a point $x+\varepsilon\vec{n}(x)$ with probability $p_+ = a_1/(a_1+a_2)$ and $x-\varepsilon\vec{n}(x)$ with probability $p_- = a_2/(a_1+a_2)$, where $\vec{n}(x)$ is the unit vector normal to S and points toward the direction where the diffusion coefficient is a_1. The parameter ε is fixed and arbitrary. This choice is justified by a finite-difference computation.

In dimension one, the probability p_+ corresponds to the probability that starting from x, the process X generated by $\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}x}\left(a\frac{\mathrm{d}}{\mathrm{d}x}\right)$ with $a = a_1\mathbf{1}_{\mathbb{R}_+} + a_2\mathbf{1}_{\mathbb{R}_-^*}$ reaches ε before reaching $-\varepsilon$. This can be easily computed using the *scale function* $S(x)$ of x (See [6] for example), since $S(x) = \int_0^x \mathrm{d}y/a(y)$ up to some multiplicative and additive constants.

In [18], we have considered the approximation of a diffusion process in a one-dimensional discontinuous media by a random walk on a fixed set of points, using the local properties of the process.

The approach above for a process $(X^1,\dots,X^d)$ living on $\mathbb{R}^d$ if $S = \{0\}\times\mathbb{R}^{d-1}$, and $a = a_1$ on $\mathbb{R}_+\pm\times\mathbb{R}^{d-1}$ and $a = a_2$ on $\mathbb{R}_-\times\mathbb{R}^{d-1}$ means that we neglect the behavior of $(X^2,\dots,X^d)$, since we have $\mathbb{E}_x[X^i_\tau] = 0$ for $i = 2,\dots,d$ when τ is small enough, where τ is the first time X reaches either $\vec{n}(x)\varepsilon + S$ or $-\vec{n}(x)\varepsilon + S$.

Using the computations in [18], we can also choose two constant ε_+ and ε_-, and let the particle jumps to $x+\varepsilon_+\vec{n}(x)$ and $x-\varepsilon_-\vec{n}(x)$ with some probabilities that are deduced from the behavior of one-dimensional process $Y_t = \vec{n}(x)X_t$ when X reaches S at the point x. Since we are interested only by the local properties of Y, we may assume that $S = \{0\}\times\mathbb{R}^{d-1}$, $a = a_1$ on $\mathbb{R}_+\times\mathbb{R}^{d-1}$ and $a = a_2$ on $\mathbb{R}_-\times\mathbb{R}^{d-1}$. The infinitesimal generator of Y then becomes

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}x}\left(b(x)\frac{\mathrm{d}}{\mathrm{d}x}\right) \text{ with } b(x) = \begin{cases} a_1 \text{ on } \mathbb{R}_+, \\ a_2 \text{ on } \mathbb{R}_-. \end{cases}$$

If $-\varepsilon_- < 0 < \varepsilon_+$ and τ is the first time Y reaches either $-\varepsilon_-$ or ε_+, we have

$$\mathbb{P}_0[Y_\tau = \varepsilon_+] = 1 - \mathbb{P}_0[Y_\tau = -\varepsilon_-] = u_+(0) \text{ with } \begin{cases} Au_+ = 0 \text{ on } (-\varepsilon_-,\varepsilon_+), \\ u_+(\varepsilon_-) = 0,\ u_-(\varepsilon_+) = 1, \end{cases}$$

$$\mathbb{E}_0[\tau|Y_\tau = \varepsilon_+] = T_+(0) \text{ with } \begin{cases} AT_+ = u_+ \text{ on } (-\varepsilon_-,\varepsilon_+), \\ T_+(\varepsilon_-) = T_+(\varepsilon_+) = 0, \end{cases}$$

$$\mathbb{E}_0[\tau|Y_\tau = \varepsilon_-] = T_-(0) \text{ with } \begin{cases} AT_- = 1-u_+ \text{ on } (\varepsilon_-,\varepsilon_+), \\ T_-(\varepsilon_-) = T_-(\varepsilon_+) = 0. \end{cases}$$

The functions u_+, v_+ and v_- are easily computed. Let us note that in addition, this approach gives us how to increment the time when the

particle is reinjected in the media. Thus,

$$u_+(0) = \frac{a_1\varepsilon_-}{a_1\varepsilon_- + a_2\varepsilon_+} \tag{9}$$

and

$$T_+(0) = \frac{a_2\varepsilon_+^3 + 3a_1\varepsilon_+^2\varepsilon_- + 2a_1\varepsilon_+\varepsilon_-^2}{3a_1(a_1\varepsilon_- + a_2\varepsilon_+)}, \tag{10}$$

$$T_-(0) = \frac{2a_2\varepsilon_+^2\varepsilon_- + 3a_2\varepsilon_+\varepsilon_-^2 + a_1\varepsilon_-^3}{3a_2(a_1\varepsilon_- + a_2\varepsilon_+)}. \tag{11}$$

Let us remark that if $\varepsilon_+ = \varepsilon_-$, which is natural choice in view of a "finite difference approximation" and $a_1 \gg a_2$, then $T_+(0) \approx 5/3a_1$ while $T_-(0) \approx 1/3a_2$. This means that the times are very different.

On the other hand, it is also possible to "tune" ε_+ and ε_- in order to get $T_+(0) = T_-(0)$.

Thus, using the results of [18], we get a flexible way of choosing the future localization of the particle, as well as an approximation of the average time it takes to go there. We now call this approach the random walk approximation.

3.3. **A comparison between the two methods.** We can now give a heuristic and rough justification of the kinetic approximation, based on probabilistic comparison between the two methods. We still consider the case of a media where $a(x) = a_1$ on $\mathbb{R}_+^* \times \mathbb{R}^{d-1}$ and $a(x) = a_2$ on $\mathbb{R}_-^* \times \mathbb{R}^{d-1}$. We denote by $\vec{n} = (1, 0, \dots, 0)$ the vector normal to S.

With the kinetic approximation or the random walk approximation, the particle has the same behavior until it reaches the surface of discontinuity.

With the kinetic approximation, when the particle is at a point x on the hyperplane $S = \{0\} \times \mathbb{R}^{d-1}$, its direction $v \in \mathbb{V}$ is chosen uniformly on $\mathbb{V}$, so that the probability that $v \cdot \vec{n} > 0$ is equal to $1/2$. If it is the case, then its average position is $x + \varepsilon C_{\mathbb{V}}(a_1/2)\vec{n}$ and the time is incremented in average by $\varepsilon^2 C_{\mathbb{V}}(a_1/2)\vec{n}$. Otherwise, its average position is $x - \varepsilon C_{\mathbb{V}}(a_2/2)\vec{n}$ and the time is incremented in average by $\varepsilon^2 C_{\mathbb{V}}(a_2/2)\vec{n}$.

With the random walk approximation, we deduce from (9) that the probability that when at $x \in S$, the particle reaches $x + \varepsilon C_{\mathbb{V}}(a_1/2)\vec{n}$ before $x - \varepsilon C_{\mathbb{V}}(a_2/2)\vec{n}$ is equal to $1/2$. In addition, using (10), we get that the average time t_+ (resp. t_-) to reach $x + \varepsilon C_{\mathbb{V}}(a_1/2)\vec{n}$ (resp. $x - \varepsilon C_{\mathbb{V}}(a_2/2)\vec{n}$) before $x - \varepsilon C_{\mathbb{V}}(a_2/2)\vec{n}$ (resp. $x + \varepsilon C_{\mathbb{V}}(a_1/2)\vec{n}$) is equal to

$$t_+ = C_{\mathbb{V}}\varepsilon^2 \frac{2a_1^2a_2 + a_1a_2^2}{3(a_1^2 + a_2^2)}, \text{ resp. } t_- = C_{\mathbb{V}}\varepsilon^2 \frac{2a_2^2a_1 + a_2a_1^2}{3(a_1^2 + a_2^2)}.$$

We then obtain that the two methods give in average comparable ways of re-injecting the particle in the media.

4. Basic one-dimensional examples

4.1. **Description of the models.** The aim of this section is to compare the global and the local approximation described in Sections 2 and 3 on two basic examples in dimension one. In the domain $D = (-1, 1)$, we consider the two equations

$$-\frac{1}{2}\nabla \cdot (a(x)\nabla u(x)) = 0 \tag{12}$$

with Dirichlet boundary conditions $u(-1) = 0$, $u(1) = 1$ and

$$-\frac{1}{2}\nabla \cdot (a(x)\nabla v(x)) = 1 \tag{13}$$

with Dirichlet boundary conditions $v(-1) = 0$, $v(1) = 0$ where $a(x) = a_1$ if $x > 0$ and $a(x) = a_2$ if $x < 0$. Using the continuity of the solutions and of the flux at the origin, the exact solution of (12) is easily computed and is given by

$$u(x) = \frac{a_1 + a_2 x}{a_1 + a_2} 1_{[0,1]}(x) + \frac{a_1 + a_1 x}{a_1 + a_2} 1_{[-1,0]}(x)$$

while the solution to (13) is

$$v(x) = \frac{2}{a_1 + a_2} + \left(\frac{(a_1 - a_2)x}{a_1(a_1 + a_2)} - \frac{x^2}{a_1} \right) 1_{[0,1]}(x) + \left(\frac{(a_1 - a_2)x}{a_2(a_1 + a_2)} - \frac{x^2}{a_2} \right) 1_{[-1,0]}(x).$$

Let us denote the stochastic process X_t^x generated by the operator $\frac{1}{2}\nabla \cdot (a(x)\nabla \cdot)$. The solution $u(x)$ of the first equation is the probability that the process X_t^x reaches 1 before -1. The solution $v(x)$ of the second is the mean exit time of the process of the domain D. We approximate the operator $\frac{1}{2}\nabla \cdot (a(x)\nabla)u$ by the transport operator

$$T_\varepsilon u(x, v) = -\frac{v}{\varepsilon}\frac{\partial u}{\partial x}(x, v) + \frac{2}{3\varepsilon^2 a(x)} \left(\frac{1}{2}\int_{-1}^{1} u(x, v')dv' - u(x, v) \right)$$

depending on the parameter ε in the domain $D \times [-1, 1]$, with boundary conditions $u(1, v) = 1$ and $u(-1, v) = 0$ on Γ_-. In the case of the global approximation, we compute numerically an averaged solution on the velocities at a given point x. This simply means that the initial velocity is picked uniformly at random on the velocity space $[-1, 1]$.

4.2. **Global approximation.** We first look at an example where the jump in the diffusion coefficient is small that is $a_1 = 2, a_2 = 1$. We compute the solution at three reference points $-0.5, 0$ and 0.5 using $N_1 = 10^5$ simulations. We study the accuracy on the solutions u and v as a function of the parameter ε for the estimators based on a direct approximation and for the ones based on the Romberg extrapolation. We start with the point $x = 0.5$ for which the simulations should be

the fastest as this point is close to the boundary and in the zone with biggest diffusion coefficient. The exact values are $u(0.5) \simeq 0.83333$ and $v(0.5) \simeq 0.45833$. In the following table, we compute direct approximations for values of ε from 0.1 to 0.025 and Romberg extrapolations with $\varepsilon = 0.1$ and 0.01. We denote by u_{dir} and v_{dir} the direct approximations and by u_{rom} and v_{rom} the Romberg type approximations.

ε	0.1	0.05	0.025	0.01	0.005	0.0025
u_{dir}	0.7908	0.8103	0.8223	0.8289	0.8307	0.8322
v_{dir}		0.8297	0.8359		0.8326	0.8340
u_{rom}	0.657	0.556	0.509	0.478	0.4687	0.4655
v_{rom}		0.4561	0.4633		0.4591	0.463

These simulations and the following are performed on a standard 1.66 Ghz laptop. The simulations have taken 14 seconds for the first Romberg extrapolations (that is for the total time with parameter ε equal to successively 0.1, 0.05 and 0.025). They have taken 1550 seconds for the second Romberg extrapolations (that is for the total time with parameter ε equal to successively 0.01, 0.005 and 0.0025). We can notice that the direct approximations require ε to be equal to 0.0025 to be acceptable in terms of bias. The same accuracy is obtained with the the Romberg extrapolations of level one and two for $\varepsilon = 0.1$. This shows that the Romberg extrapolation is very satisfactory and that the direct method using a small value of the parameter should be discarded because of too large computational times. We now look if this Romberg extrapolation is still acceptable for the other reference points. For the point $x = 0$, the exact values are $u(0) \simeq 0.66666$ and $v(0) \simeq 0.66666$ and for the point $x = -0.5$, the exact values are $u(-0.5) \simeq 0.33333$ and $v(-0.5) \simeq 0.58333$.

ε	0.1	0.05	0.025	0.1	0.05	0.025
u_{dir}	0.644	0.654	0.662	0.355	0.342	0.340
v_{dir}		0.6647	0.671		0.331	0.340
u_{rom}	0.842	0.747	0.705	0.753	0.667	0.626
v_{rom}		0.6529	0.6663		0.5812	0.5859

We observe that for both points, the approximations v_{dir} and v_{rom} are acceptable and that the approximation using only $\varepsilon = 0.1$ and 0.05 may be more accurate than the one using the 3 values of the parameter. This means that the Romberg extrapolation using two parameters is sufficient in these cases and that the variance of the estimator using the three parameters is bigger. The direct approximation is not accurate enough especially for the computation of the mean exit time. The computational times are 22 seconds for both points. We can conclude on this first example that the Romberg extrapolation works well and that it provides an approximation accurate up to about 3 digits in a computational time of roughly 20 seconds. We now study the same example using the local approximation to check its efficiency and whether or

not it is necessary to study more difficult examples based on the global approximation.

4.3. **Local approximation.**

4.3.1. *Description of the algorithm.* The local approximation leads to a very simple and very fast algorithm based on the properties of the Brownian motion on the interval [-1,1]. We still denote by u_ε and v_ε the approximations of equations (12) and (13) and we describe now the motion of a particle used to compute $u_\varepsilon(x)$ and $v_\varepsilon(x)$ simultaneously.

The motion of the particle starts at some point x in $(-1,1)$. We denote by y the position of a particle that is still alive during the walk. If $y > 0$ then with probability $1-y$ it goes to 0 and with probability y it goes to 1. The time to add the total time of the walk is in both cases $\frac{y(1-y)}{a_1}$. If $y < 0$ then with probability $1+y$ it goes to 0 and with probability $-y$ it goes to -1. The time to add the total time is in both cases $\frac{-y(1+y)}{a_2}$. If finally $y = 0$, we pick a velocity v uniformly in $[-1,1]$ and an exponential random variable t of mean 1. If $v > 0$, the time to add to the total time is $\min(\frac{\varepsilon}{v}, 1.5a_1\varepsilon^2 t)$ and the position is $\min(1, 1.5a_1 t\frac{\varepsilon}{v})$. If $v < 0$, the time to add to the total time is $\min(\frac{-\varepsilon}{v}, 1.5a_2\varepsilon^2 t)$ and the position is $\min(-1, 1.5a_2 t\frac{\varepsilon}{v})$. The motion continues until the particle hits -1 or 1.

4.3.2. *Previous example.* We experiment this method on the previous example starting with $x = 0.5$ using $N_1 = 10^5$ and $N_2 = 10^6$ simulations.

	N_1			N_2		
ε	0.1	0.05	0.025	0.1	0.05	0.025
$v_\varepsilon(0.5)$	0.4621	0.4567	0.4568	0.4637	0.4588	0.4574
$u_\varepsilon(0.5)$	0.8328	0.8349	0.8342	0.8323	0.8334	0.8337

The CPU times for N_1 simulations are respectively 0.2, 0.4 and 0.6 seconds for $\varepsilon = 0.1$,0.05 and 0.025. We can notice that we obtain similar values for both u_ε and v_ε for $\varepsilon = 0.05$ and 0.025. This is confirmed when we use N_2 simulations where both approximations are accurate up to 3 or 4 digits. On the contrary, the approximations based on $\varepsilon = 0.1$ still have a bias which is larger than the Monte Carlo error. We now compute the solutions at the two other reference points using N_2 simulations to focus on the bias.

ε	0.1	0.05	0.025	ε	0.1	0.05	0.025
$u_\varepsilon(0)$	0.664	0.6666	0.6665	$u_\varepsilon(-0.5)$	0.332	0.3337	0.3336
$v_\varepsilon(0)$	0.6791	0.6665	0.6665	$v_\varepsilon(-0.5)$	0.5887	0.5840	0.5823

We observe that we still obtain an accuracy of 4 digits taking $\varepsilon = 0.05$. The CPU times for this value are 6 seconds at point 0 and 3 seconds at point -0.5. We can conclude that the new approximation can be used in this example with a parameter $\varepsilon = 0.05$ and that it provides an

accuracy of about 3 digits using $N_1 = 10^5$ simulations in no more than 0.6 seconds. For the same accuracy, the local approximation is about 40 times quicker than the global one.

4.3.3. *More difficult situations.* The question now is to figure out if our method is efficient when the ratio $\frac{a_1}{a_2}$ is big and how to calibrate the value of ε in such situations. We have made some numerical tests for different values of this ratio. In fact the choice of ε depends more on $\max(a_1, a_2)$ than on the ratio $\frac{a_1}{a_2}$. A good choice is to take $\varepsilon = \frac{0.1}{\max(a_1,a_2)}$ which gives a good accuracy and low CPU times. We give a numerical example in the case $a_1 = 1000$ and $a_2 = 1$ where the physical constants are different from several degrees of magnitude. We use $N = 10^5$ simulations with $\varepsilon = 0.0005$. Numerical results are summarized in the following table.

x	$u(x)$	$u_\varepsilon(x)$	$v(x)$	$v_\varepsilon(x)$	CPU
0.5	0.001249	0.00124783	0.9995	0.99951	0.58
0	0.001998	0.00199742	0.999	0.99911	1.1
-0.5	0.250999	0.2509978	0.4995	0.4993	0.56

The values of u_ε at the reference points are accurate up to 6 digits and the values of v_ε up to 4. These computations require no more than 1.1 seconds. We have obtained the same kind of results in all the tests we have experimented.

5. Two dimensional examples

5.1. **Description of the models.** We now study two equations in dimension 2 using only the local approximation which appeared as the most efficient in the previous section. The spatial domain D is the square $[-1, 1]^2$ divided in the two rectangles $D_2 = [-1, 0] \times [-1, 1]$ and $D_1 = [0, 1] \times [-1, 1]$ in which the diffusion coefficient $a(x, y)$ is constant and equal to a_2 in D_2 and a_1 in D_1.

The first equation is

$$-\frac{1}{2}\nabla \cdot (a(x, y)\nabla u(x, y)) = 0$$

with Dirichlet boundary conditions $u(x, y) = (1 + \frac{x}{a_1})(1 + y)$ if $x \geq 0$ and $u(x, y) = (1 + \frac{x}{a_1})(1 + y)$ if $x \leq 0$. The exact solution is

$$u(x, y) = \left(1 + \frac{x}{a_2}\right)(1 + y)1_{[-1,0]}(x) + \left(1 + \frac{x}{a_1}\right)(1 + y)1_{[0,1]}(x).$$

It verifies the continuity and flux conditions at the interface $x = 0$.

The second equation is

$$-\frac{1}{2}\nabla \cdot (a(x, y)\nabla w(x, y)) = 1$$

with Dirichlet boundary conditions $w(x, y) = 0$ on ∂D. Let us denote the stochastic process $Z_t^{x,y}$ generated by the operator $\frac{1}{2}\nabla \cdot (a(x, y)\nabla)w$.

The solution $w(x,y)$ of this second equation is the mean exit time of $Z_t^{x,y}$ from the domain D. As the exact solution is unknown, we will use as a reference solution, a numerical solution using a finite element method taken from the `pdetool` package of Matlab with a very fine mesh.

We approximate the operator $\frac{1}{2}\nabla\cdot(a(x,y)\nabla u)$ by the transport operator

$$T_\varepsilon u(x,y,\theta) = -\frac{\cos(\theta)}{\varepsilon}\nabla_x u(x,y,\theta) - \frac{\sin(\theta)}{\varepsilon}\nabla_y u(x,y,\theta) + \frac{1}{\varepsilon^2 a(x,y)}\left(\frac{1}{2\pi}\int_0^{2\pi} u(x,y,\theta)d\theta - u(x,y,\theta)\right)$$

depending on the parameter ε in the domain $D\times\mathbb{V}$, with boundary conditions of absorption type (no incoming neutrons). We use the following algorithm to compute the solution at point (x,y). This algorithm is a combination of the walk on spheres method with a η absorbing boundary layer for each of the subdomains and of a local approximation based on T_ε at the interface. Obviously, we could have used the walk on rectangles method [11] but it would be maybe too favourable on this particular example. The walk starts at point (x,y). If for example $x>0$, we walk until we are close to η from the boundary of D_1. Along with this walk, we compute the contribution of the source term for each of the n circles of radius r_i until we hit the boundary. As the source term is equal to one, this contribution is equal to

$$\sum_{i=1}^{n}\frac{r_i^2}{2a_1}.$$

In the case of a non-constant source term, the modified walk on spheres could be used to compute this contribution [23]. If the walk hits ∂D it stops. If it hits the interface, we pick an angle θ uniformly in $[0,2\pi]$ and an exponential random variable to choose the new position. As in dimension one, we have to study carefully if there is an intersection or not with the boundary. According to the sign of $v_x=\cos(\theta)$, the particle goes into D_1 or D_2 and the motion continues until a boundary of D is reached.

5.2. **Numerical results.** As we did in dimension one, we first consider the case $a_1=2$ and $a_2=1$. We use $N=10^4$ simulations with $\varepsilon=0.02$ and $\eta=10^{-5}$. The three reference points are mesh points of the `pdetool` mesh. We take $(x_0,y_0)=(0,0.002674)$, $(x_1,y_1)=(-0.5027,y_0)$ and $(x_2,y_2)=(-x_1,y_0)$. Numerical results are summarized in the following table.

	u	u_ε	v	v_ε	CPU
(x_0, y_0)	1.00267	1.00127	0.39290	0.38500	1.8
(x_1, y_1)	0.49862	0.49180	0.38040	0.37812	0.8
(x_2, y_2)	1.25469	1.2468	0.26676	0.26500	0.8

The values of u_ε and v_ε at the reference points are accurate from 2 to 3 digits. We now study the case $a_1 = 80$ and $a_2 = 4$ where the physical constants are taken from a problem on the computation of molecule physical properties [36] and where now $\varepsilon = 0.0005$ using the same scaling than in dimension one. The values of u_ε and v_ε at the reference points are as accurate as previously and the CPU times similar which indicates that our scaling is efficient.

	u	u_ε	v	v_ε	CPU
(x_0, y_0)	1.00267	0.98900	0.01401	0.01392	2.3
(x_1, y_1)	0.87670	0.87260	0.06232	0.06276	1
(x_2, y_2)	1.00897	1.00033	0.00830	0.00826	0.9

On this last case, we have also tested the method based on the finite differences approximation at the boundary. When the particle hits the interface at point $(0, y)$ it is replaced at the position (h, y) with probability $80/(4+80)$ and at position $(-h, y)$ with probability $4/(4+80)$. Taking $h = 0.01$, we have obtained an accuracy and CPU times similar to the ones obtained with our method.

6. PDE and stochastic PDE in dimension 3

6.1. **The physical model.** We study the same equation than in dimension 2 with no source term. The spatial domain D is the square $[-1, 1]^3$ divided in the two hyper-rectangles $D_2 = [-1, 0] \times [-1, 1]^2$ and $D_1 = [0, 1] \times [-1, 1]^2$ in which the diffusion coefficient a is constant and equal to a_2 in D_2 and a_1 in D_1. The equation is

$$-\frac{1}{2}\nabla \cdot (a\nabla u) = 0$$

with Dirichlet boundary conditions $u(x, y, z) = (1 + \frac{x}{a_1})(1 + y)(1 + z)$ if $x \geq 0$ and $u(x, y, z) = (1 + \frac{x}{a_2})(1 + y)(1 + z)$ if $x \leq 0$. The exact solution is

$$u(x, y, z) = (1+\frac{x}{a_2})(1+y)(1+z)1_{[-1,0]}(x)+(1+\frac{x}{a_1})(1+y)(1+z)1_{[0,1]}(x).$$

We approximate the operator $\frac{1}{2}\nabla \cdot (a\nabla \cdot)$ by a transport operator on the interface depending on the parameter ε in the domain $D \times \mathbb{V}$, with boundary conditions of absorption type (no incoming neutrons) where the velocity space is the unit sphere. The walk on spheres method is used elsewhere with $\eta = 10^{-4}$ in the numerical simulations.

6.2. **Numerical results.** We perform $N = 10^4$ simulations at the three reference points $M_0 = (0,0,0)$, $M_1 = (-0.5,0,0)$ and $M_2 = (0.5,0,0)$ for the same two sets of diffusion coefficients used in the test case in 2D. We take respectively $\varepsilon_1 = 0.02$ for the first one and $\varepsilon_2 = 0.0005$ for the second one. We obtain the same accuracy on the solution of two or three digits for CPU times about 5 times bigger compared to the problems in dimension 2.

	u	u_{ε_1}	CPU	u	u_{ε_2}	CPU
M_0	1.0	1.0060	11	1.0	0.9989	13
M_1	0.5	0.5053	4.9	0.8750	0.8742	5.4
M_2	1.2500	1.2417	4.8	1.00625	1.0091	5.5

6.3. **A stochastic PDE.**

6.3.1. *Introduction.* In some physical situations, the diffusion coefficient is not known exactly and is considered as a random variable which modelizes the error on its measure. More precisely, the domain is divided into different zones in which the diffusion coefficient is constant and has a given distribution independent of the other zones. In general one is interested in the mean value of the solution. It is obtained by solving many times the PDE using a deterministic method with a random distribution of the diffusion coefficient in each zone [1, 14, 15].

It is possible to use our approach to build a double Monte Carlo method to compute this mean value at a fixed point. We pick at random according to their distributions the diffusion coefficients and we make only one simulation of the diffusion process in the heterogeneous medium to approximate the Feynman-Kac representation. Moreover, we can also compute the integral of the solution in the domain by just picking in addition the starting point uniformly at random in the domain.

6.3.2. *Numerical results.* We assume that a_1 and a_2 are uniform and independent random variables on $[0.5, 1.5]$. The mean value of the solution is

$$v(x,y,z) = (1 + x\ln(3))(1+y)(1+z)$$

and its integral is equal to 8. The parameter ε is adapted to the values of a_1 and a_2 to ensure low CPU times. For each simulation, we take $\varepsilon = \frac{0.03}{\max(a_1,a_2)}$ using once again a scaling argument. We compute the mean value of the solution at point M_0, $M_1, M_3 = (-\frac{1}{\ln(3)}, 0, 0)$ and its integral using respectively $N_1 = 10^4$ simulations and $N_2 = 10^5$ simulations.

	u	u_{N_1}	CPU	u_{N_2}
D	8	8.067	3.3	8.021
M_0	1	0.979	13.6	1.0026
M_1	0.4507	0.444	5.4	0.4509
M_3	0	-0.0047	1.3	0.001

The accuracy and the CPU times are similar to the ones obtained when solving the deterministic PDE of Section 6.2. This means that the cost of the resolution of the stochastic PDE is about the same than the one of the deterministic PDE, when one is interested on the solution at just one point of the domain D. This is really a big advantage compared to the resolution by means of deterministic methods.

7. Computation of the hydro-dynamic load in a porous media

In order to deal with a realistic case, we present some results on the computation of the hydro-dynamic load in a porous media. For this, we consider the COUPLEX exercises provided in 2001 by the French agency ANDRA (Agence Nationale pour la Gestion des Déchets Radioactifs – French National Radioactive Waste Management Agency) regarding the disposal of nuclear waste. The exercises are presented in [4]. A special issue of *Computational Geoscience* (vol. 8:2, 2004) is dedicated to them.

7.1. **Description of the model.** We aim at solving at some point the equation

$$\nabla \cdot (K \nabla H) = 0 \text{ in } D$$

where D is a two dimensional rectangle $(0, 25000) \times (0, 695)$. The domain is decomposed in several zones according to Figure 2, which corresponds to different types of rocks. The permeabilities are

	Marl	Limestone	Clay	Dogger
K (m/year)	$3.1536 \cdot 10^{-5}$	6.3072	$3.1536 \cdot 10^{-6}$	25.2288

The boundary conditions are

$$\begin{aligned}
&H = 289 \text{ on } \{25000\} \times (0, 200) \text{ (right Dogger)},\\
&H = 310 \text{ on } \{25000\} \times (350, 595) \text{ (right limestone)},\\
&H = 180 + \frac{160x}{25000} \text{ on } (0, 25000) \times \{695\} \text{ (top)},\\
&H = 200 \text{ on } \{0\} \times (295, 595) \text{ (left limestone)},\\
&H = 286 \text{ on } \{0\} \times (0, 200) \text{ (left Dogger)},\\
&\frac{\partial H}{\partial n} = 0 \text{ elsewhere.}
\end{aligned}$$

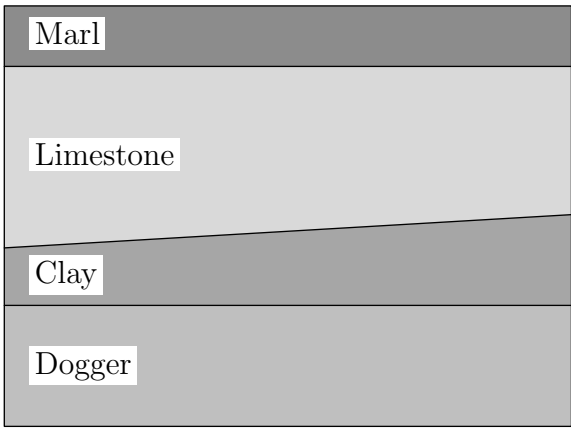

FIGURE 2. The geometry of the domain.

7.2. **Approximate analytical solution.** We propose here to give an approximate analytical solution in each of the physical zones based on probabilistic arguments. This has been already done in [8] for two of the zones using physical arguments.

In Dogger, a Brownian motion starting at (x, z) cannot cross the barrier $z = 0$ because of the reflexion due the Neumann boundary conditions and the barrier $y = 200$ can also be replaced approximatively by Neumann boundary conditions because K is very small in clay and large in dogger. Only the horizontal motion is important in this case and this motion is just a one dimensional Brownian motion because K is constant in dogger. So we can approximate $H(x, z)$ by the solution $H_a(x)$ of the one dimensional Laplace equation with boundary conditions $H_a(0) = 286$ and $H_a(25000) = 289$ that is

$$H_a(x) = 286 + \frac{3x}{25000}.$$

In limestone, we can use the same reasoning by replacing the two barriers constituted of marl and clay by Neumann boundary conditions at the interfaces between limestone and marl or limestone and clay. If we also consider that the slope of the barrier separating clay and limestone is negligible, we obtain the approximate solution

$$H_b(x) = 200 + \frac{110x}{25000}.$$

The approximation proposed in [8] which takes into account the slope of the barrier is

$$H_c(x) = 200 + \frac{110}{\ln(\frac{245}{300})} \ln\left(1 - \frac{55x}{300 \times 25000}\right).$$

The approximation $H_b(x)$ can be obtained using a first order expansion of the two terms in logarithm in $H_c(x)$. An easy computation shows

that $H_b(x) - H_c(x) \geq 0$ and that

$$\max_{x\epsilon(0,25000)} H_b(x) - H_c(x) = 2.783$$

at point $x = 12,921$. It has also been noticed in [8] that the difference between the numerical solution obtained using the finite volume method and the approximate analytical one never exceeded 1.5 in dogger and limestone.

In marl, the probability to reach the top of the domain before the bottom starting at the point (x, z) can be approximated by $p(z) = \frac{z-595}{100}$ using the previous arguments this time for the vertical motion. Then, the Dirichlet boundary conditions on the top of the marl region are modelled by the linear function $g(x) = 180 + \frac{160x}{25000}$ and we can also use linear or close to linear Dirichlet boundary conditions on the bottom boundary by taking $H_b(x)$ or $H_c(x)$ on it. Now the conditional law $f_{x,z}(y)$ of the hitting position of the top of the domain is close to a symmetric function with a fast decay away from x so we have

$$\int_0^{25000} g(y) f_{x,z}(y) dy \simeq \int_0^{25000} (g(x) + g(t)) f_{x,z}(x+t) dt \simeq g(x).$$

Using the same arguments for the bottom boundary, we can give the analytical approximation

$$H_d(x, z) = p(z)g(x) + (1 - p(z))H_c(x)$$

in the marl region.

In clay, we can use the same method assuming furthermore that the slope of the interface between limestone and clay is negligible. This leads to the approximation

$$H_e(x, z) = p_1(x, z)H_c(x) + (1 - p_1(x, z))H_a(x)$$

with

$$p_1(x, z) = \frac{z - 200}{z - 200 + 295 + \frac{55}{25000}x - z} = \frac{z - 200}{95 + \frac{55}{25000}x}.$$

7.3. **Numerical results.** For each of the physical regions, we use the walk on spheres method inside the regions with an absorption parameter $\eta = 10^{-8}$. When the particle hits an interface between two regions, we use the local approximation with parameter $\varepsilon = 0.2$. When the particle hits the boundary at a point with Neumann boundary conditions, we still use the local approximation by always replacing the particle inside the domain. For each of the regions, we give the numerical approximations $\widetilde{H}$ at 4 significant points and compare it with the approximate analytical solutions of the previous subsection. We perform $N = 100$ Monte Carlo simulations on a standard laptop. We also give the CPU times in second, the variance σ^2 and $\frac{\sigma}{\sqrt{N}}$ which is an indicator of the expected accuracy.

Dogger	$\widetilde{H}(x,z)$	$H_a(x)$	σ^2	$\frac{\sigma}{\sqrt{N}}$	CPU
$(x,z)=(1000,50)$	286.2	286.1	0.59	0.07	15
$(x,z)=(6000,150)$	286.8	286.7	1.79	0.13	71
$(x,z)=(12500,100)$	287.3	287.5	2.2	0.15	103
$(x,z)=(18440,200)$	288.1	288.2	1.94	0.14	72

In dogger, we observe that the variance is very small and that for all the points where the numerical solution is computed the error between the approximate numerical solution and the approximate analytical one is always less than 0.2. The numerical solution seems not to depend on z even for the point $(x,z)=(18440,200)$ which is located on the interface. The CPU times are large, one second for a single trajectory when starting from the center of the domain. In this case, the trajectory has hit $240,000$ times in average the interface between dogger and clay. None of the 100 trajectories reached the limestone region.

Limestone	$\widetilde{H}(x,z)$	$H_c(x)$	σ^2	$\frac{\sigma}{\sqrt{N}}$	CPU
$(x,z)=(1000,320)$	202.2	204	121	1.1	59
$(x,z)=(6000,570)$	224.2	224.4	2097	4.6	236
$(x,z)=(12500,460)$	251.7	252.2	3044	5.5	344
$(x,z)=(21680,342.7)$	291	294	1804	4.2	187

In limestone, the variance is a lot bigger than in clay. Nevertheless we achieve a good accuracy, only the error at point $(x,z)=(21680,342.7)$ is greater then 3 which is bigger than the acceptable error given in [8]. The CPU times are bigger than in dogger because the limestone region is larger. Starting from point $(x,z)=(12500,460)$, the trajectory has hit in average 1.6 million times one or the other interface. One single trajectory among the hundred has reached the top of the marl region and none the dogger region.

Clay	$\widetilde{H}(x,z)$	$H_e(x,z)$	σ^2	$\frac{\sigma}{\sqrt{N}}$	CPU
$(x,z)=(1000,220)$	270.6	269.2	1104	3.3	15
$(x,z)=(6000,280)$	240.3	240.7	2321	4.8	261
$(x,z)=(12500,260)$	268.9	270.2	1948	4.4	250
$(x,z)=(20560,247)$	286.1	288.5	705	2.7	99

In clay and marl, we observe a good agreement between the analytical and the numerical approximate solutions. The CPU times and variances are between the one of the limestone and dogger regions.

Marl	$\widetilde{H}(x,z)$	$H_d(x,z)$	σ^2	$\frac{\sigma}{\sqrt{N}}$	CPU
$(x,z)=(1000,670)$	191	190.8	179	1.3	10
$(x,z)=(6000,620)$	220.4	223	1384	3.7	200
$(x,z)=(12500,645)$	253.8	256.1	1448	3.8	179
$(x,z)=(20000,675)$	301.8	303.6	665	2.5	58

We can conclude that our method is efficient and that it treats well the discontinuity of the diffusion coefficients. This example is especially

a good test for our method because the trajectory hits a huge number of times the interface between the regions.

8. Conclusion

We have introduced two kinetic schemes to approximate the solutions of partial differential equations in divergence form with piecewise constant diffusion coefficients.

Both schemes have provided accurate approximations but the scheme based on a local approximation is a lot more efficient as it uses the kinetic approximation only at the interface and a fast simulation of the Brownian motion elsewhere. On the two dimensional basic examples, the `pdetool` solver of Matlab was more efficient than our method as it gives a global solution in lower CPU times. The kinetic approximation has also given a good approximation in the case of the pointwise computation of the hydro-dynamic load in a very heterogeneous physical domain. For problems in dimension three and especially for stochastic PDEs, our approach seems really efficient as it cost increases slowly compared to the ones of two dimensional problems.

Our method has several advantages compared to the finite differences method of Mascagni and Simonov. With Remark 2, it can provide the law of the exit time from the domain, which is crucial in the computation of the principal eigenelements of the operator by means of Monte Carlo methods [28, 29]. In a recent work [5] it has also been used successfully for solving the Poisson-Boltzmann equation of molecular electrostatics for which the finite differences method was originally designed. It has been proved for example that in the case of a molecule constituted of one sphere our method is of order two in ε instead of order one for the finite differences method. Moreover, the numerical performances of our method were very satisfactory.

Our modified finite difference scheme also allows one to overcome the problem of the precision of the time simulation. Finally the kinetic approximation allows one to deal more easily when several discontinuity surfaces are close to each other.

Acknowledgment. The first author wishes to thank to *Groupement MOMAS* (ANDRA, BRGM, CEA, EDF, IRSN) for its financial support.

References

[1] I. Babuška, F. Nobile, and R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.*, **45**(3):1005–1034, 2007.

[2] C. Bardos. Équation de transport. Théorie spectrale et approximation de la diffusion. *Journées Équations aux dérivées partielles*, 1–10, 1982.

[3] H. Berg. *Random walk in biology.* Princeton University Press, 1993.

[4] A. Bourgeat, M. Kern, S. Schumacher and J. Talandier. The COUPLEX Test Cases: Nuclear Waste Disposal Simulation. *Comput. Geosciences*, **8**(2):83–89, 2004.
[5] M. Bossy, N. Champagnat, S. Maire and D. Talay. Probabilistic interpretation and random walk on spheres algorithms for the Poisson-Boltzmann equation in molecular dynamics. Preprint INRIA, 2010.
[6] L. Breiman. *Probability*. Addison-Wesley, 1981.
[7] R.S. Cantrell and C. Cosner. Diffusion models for population dynamics incorporating individual behavior at boundaries: Applications to refuge design. *Theoretical Population Biology*, **55**(2):189–207, 1999.
[8] E. Chénier, R. Eymard and M. Kern. A Finite Volume Scheme for the Transport of Radionucleides in Porous Media: Simulation of Transport Around a Nuclear Waste Disposal Site: The COUPLEX Test Cases. *Comput. Geosciences*, **8**(2):163–172, 2004.
[9] G. Dagan. *Flow and Transport in Porous Formations*. Springer-Verlag, 1989.
[10] R. Dautray and J.-L. Lions. *Évolution : numérique, transport*, volume 9 of *Analyse Mathématique et Calcul Numérique pour les Sciences et Techniques.* Masson, Paris, 1987.
[11] M. Deaconu and A. Lejay. A random walk on rectangles algorithm. *Methodol. Comput. Appl. Probab.*, **8**(1):135–151, 2006.
[12] M. Deaconu and A. Lejay. Simulation of diffusions by means of importance sampling paradigm, *Ann. Appl. Probab.* (to appear), 2010.
[13] M. Decamps, A. de Schepper, M. Goovaerts, and W. Schoutens. A note on some new perpetuities. *Scand. Actuar. J.*, **2005**(4):261–270, 2005.
[14] D. Estep, A Målqvist and S. Tavener. Nonparametric density estimation for randomly perturbed elliptic problems I: computational method, a posteriori analysis, and adaptive error control. *SIAM J. Sci. Comput.*, **31**:4, 2935–2995, 2009.
[15] D. Estep, A Målqvist and S. Tavener. Nonparametric density estimation for randomly perturbed elliptic problems II: applications and adaptive modeling, *Int. J. Numer. Methods Engrg.*, **80**(6):846–867, 2009. DOI:10.1002/nme.2547
[16] P. Étoré. *Approximation de processus de diffusion à coefficients discontinus en dimension un et applications à la simulation*. PhD thesis, Université Nancy 1, 2006.
[17] P. Étoré. On random walk simulation of one-dimensional diffusion processes with discontinuous coefficients. *Electron. J. Probab.*, **11**(9):249–275, 2006.
[18] P. Étoré and A. Lejay. A Donsker theorem to simulate one-dimensional diffusion processes with measurable coefficients. *ESAIM Probab. Stat.*, **11**:301–326, 2007.
[19] O. Faugeras, F. Clément, R. Deriche, R. Keriven, T. Papadopoulo, J. Roberts, T. Viéville, F. Devernay, J. Gomes, G. Hermosillo, P. Kornprobst, and D. Lingrand. The inverse EEG and MEG problems: The adjoint state approach I: The continuous case. Research report RR-3673, INRIA, 1999.
[20] O. Faure. *Simulation du mouvement brownien et des diffusions*. PhD thesis, École nationale des ponts et chaussées, 1992.
[21] D. Gilbarg and N. Trudinger. *Elliptic Partial Differential Equations of Second Order*. Springer-Verlag, 1998.
[22] T. Hillen and G. Othmer. The diffusion limit of transport equations derived from velocity-jump processes. *SIAM J. Appl. Math.*, **61**(3):751–775, 2000.
[23] C.-O. Hwang, M. Mascagni and J. A. Given. A Feynman-Kac path-integral implementation for Poisson's equation using an h-conditioned Green's function,

In 3rd IMACS Seminar on Monte Carlo Methods—MCM 2001 (Salzburg), *Math. Comput. Simulation*, **62**(3–6):347–355, 2003.

[24] P. E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*. Springer-Verlag, Berlin, 1992.

[25] O.A. Ladyženskaja, V. Ja. Rivkind and N. N. Ural'ceva. Solvability of diffraction problems in the classical sense. *Trudy Mat. Inst. Steklov*, **92**:116–146, 1966.

[26] J.-F. Le Gall. One-Dimensional Stochastic Differential Equations Involving the Local Times of the Unknown Process. In *Stochastic Analysis and Applications*, 51–82, Lecture Notes in Math. **1095**, Springer-Verlag, 1985.

[27] A. Lejay. *Méthodes probabilistes pour l'homogénéisation des opérateurs sous forme-divergence : cas linéaires et semi-linéaires*. PhD thesis, Université de Provence, Marseille, France, 2000.

[28] A. Lejay and S. Maire. Computing the principal eigenvalue of the Laplace operator by a stochastic method. *Math. Comput. Simulation*, **73**(3):351–363, 2006.

[29] A. Lejay and S. Maire. Computing the first eigenelements of some linear operators using a branching Monte Carlo method. *J. Comput. Phys.*, **227**:9794–9806, 2008.

[30] A. Lejay and M. Martinez. A scheme for simulating one-dimensional diffusion processes with discontinuous coefficients. *Ann. Appl. Probab.*, **16**(1):107–139, 2006.

[31] N. Limić. Markov Jump Processes Approximating a Nonsymmetric Generalized Diffusion. Preprint, arXiv:`0804.0848`, 2008.

[32] S. Maire. *Réduction de variance pour l'intégration numérique et pour le calcul critique en transport neutronique*.Ph.D. thesis, Université de Toulon et du Var, 2001.

[33] S. Maire and D. Talay. On a Monte Carlo method for neutron transport criticality computations, *IMA J. Numer. Anal.*, **26**(4):657–685, 2006.

[34] M. Martinez. *Interprétations probabilistes d'opérateurs sous forme divergence et analyse de méthodes numériques associées*. PhD thesis, Université de Provence / INRIA Sophia-Antipolis, 2004.

[35] M. Martinez and D. Talay. Discrétisation d'équations différentielles stochastiques unidimensionnelles à générateur sous forme divergence avec coefficient discontinu. *C. R. Acad. Sci. Paris Sér. I Math.*, **342**:51–56, 2006.

[36] M. Mascagni and N. A. Simonov. Monte Carlo methods for calculating some physical properties of large molecules. *SIAM J. Sci. Comput.*, **26**(1):339–357 (electronic), 2004.

[37] G.N. Milstein and M.V. Tretyakov. Simulation of a space-time bounded diffusion. *Ann. Appl. Probab.*, **9**(3):732–779, 1999.

[38] M. E. Muller. Some continuous Monte Carlo methods for the Dirichlet problem. *Ann. Math. Statist.*, **27**:569–589, 1956.

[39] O. Ovaskainen and S. J. Cornell. Biased movement at a boundary and conditional occupancy times for diffusion processes. *J. Appl. Probab.*, **40**(3):557–580, 2003.

[40] J.M. Ramirez, E.A. Thomann, E.C. Waymire, Haggerty R., and Wood B. A generalized Taylor-Aris formula and Skew Diffusion. *Multiscale Model. Simul.*, **5**(3):786–801, 2006.

[41] P. Seumen Tonou. *Méthodes Numériques Probabilistes pour la Résolution d'Équations du Transport et pour l'Évaluation d'Options Exotiques*, Ph.D. thesis, université de Provence, 1997.

[42] D.W. Stroock. Diffusion semigroups corresponding to uniformly elliptic divergence form operator. In *Séminaire de Probabilités XXII*, 316–347, Lecture Notes in Math. **1321**, Springer-Verlag, 1988.
[43] M. Zhang. Calculation of diffusive shock acceleration of charged particles by skew Brownian motion. *Astrophys. J.*, 541:428–435, 2000.

Projet TOSCA, Institut Élie Cartan de Nancy, (Nancy-Université, CNRS, INRIA), Campus scientifique, BP 239, 54506 Vandoeuvre-lès-Nancy CEDEX, France
E-mail address: Antoine.Lejay@iecn.u-nancy.fr

ISITV, Université de Toulon et du Var, Avenue G. Pompidou, BP 56, 832 62 La Valette du Var CEDEX, France
E-mail address: maire@univ-tln.fr