Journal of Computational Physics 233 (2013) 175–191

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Spatial filter averaging approach of probabilistic method to linear second-order partial differential equations of the parabolic type

Sophie Loire *, Igor Mezić

Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106-5070, United States

ARTICLE INFO

Article history:
Received 27 July 2011
Received in revised form 10 July 2012
Accepted 14 August 2012
Available online 30 August 2012

Keywords:
Numerical methods
Probabilistic
Backward
Spatial filtering
Parabolic equation

ABSTRACT

A backward-in-time probabilistic method with spatial filter averaging is presented to solve linear second-order partial differential equations of the parabolic type. An advantage of this methodology is that while forward methods are subject to region with loss of density of particles and hence loss of spatial resolution of the solution, the solution given by backward methods is given on any desired grid. However, traditional backward time probabilistic method using Monte Carlo averaging are computationally expensive. We prove a convergence result and present several examples. The method leads to important improvement in computational efficiency and is expected to perform well to solve high dimensional problems where a solution is needed on a large grid.

1. Introduction

The strong interest in the study of second-order equations, and parabolic equations in particular, is motivated by their importance in mechanics [30], heat conduction [19], probability theory [35], and other fields of mathematics and physics. In microfluidics, for example, micron to nano-size particle transport is studied. Submicron-size particles have small diffusion coefficient but not always small enough compared to the other forces in the system to be negligible. This leads to the need of efficient numerical methods for advection–diffusion type equation with a high Péclet number.

Parabolic equations can be solved by finite-element analysis but although very accurate results are possible for diffusion dominated problems, at high Péclet number, traditional grid-based methods can suffer from numerical dispersion or oscillatory and even unstable solutions [41]. While extremely fine mesh refinement is one possible solution, it is, in some cases, not feasible due to excessive computational requirements. Thus, alternative numerical formulations are sought that would allow accurate solutions with reasonable computational effort.

Some Eulerian approaches, for example optimal spatial methods or Petrov–Galerkin methods [2,11,4,32], were developed to tackle convection dominated problems but those procedures can yield an upstream bias in the resulting approximation or do not completely eliminate issues of localized oscillations.

Another class of approaches, the Eulerian–Lagrangian methods have shown a great potential. In those methods the advective component is treated by a characteristic tracking algorithm (a Lagrangian frame of reference), and the diffusive step is treated separately using a more standard (Eulerian) spatial approximation. The principal drawbacks of many Eulerian–Lagrangian methods are their failure to conserve mass [6,15] and the difficulty of formulating them for general boundary conditions [1]. Moreover, the computational complexity of these classical techniques increases rapidly with increase in the dimension of the space and becomes prohibitive for problems with three or more dimensions.

* Corresponding author. Tel.: +1 805 893 5095.
E-mail addresses: sloire@engr.ucsb.edu (S. Loire), mezic@engr.ucsb.edu (I. Mezić).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.08.035

To avoid such problems, probabilistic particle methods have been frequently used. Stochastic methods have long been used to solve problems in heat transport because of the ease with which these methods can be adapted to complex interfaces [16]. They have also been applied to bulk-phase reaction diffusion systems and to the dispersion of tracers in a porous medium and many other physics problem [7,28]. The most straightforward probabilistic method is the forward random-walk method [21,26,14]. In forward random-walk method the concentration profile is represented by a set of moving particles, which are advected according to the velocity field from some initial time, while the diffusive displacements of the particles are sampled from a random distribution. The particle motion is hence described by a set of Stochastic Differential Equations (SDEs). SDE models play a relevant role in many other application areas including biology, epidemiology, investment finance and population dynamics, and this increased interest in implementing discretization and simulation methods for SDEs. There has been a growth in the development and implementation of effective high-order algorithms in the last thirty years [24,37,18,36,5,27,39,12,31]. Particle-tracking methods are stable, free of grid generation problems and easy to implement and parallelize.

However, forward methods have a disadvantage in terms of computational efficiency when only the density in a small region needs to be computed or if the density varies significantly from initial time to final time. Particles tend to move apart from each other in some directions, thereby decreasing the overlap parameter and making a remesh necessary to maintain a smooth scalar field. Under these circumstances, backward particle methods turn out to be a much better solution [22]. Backward particle methods track particles from the point of interest backward in time [10,23,38,34]. The clear advantage of backward particle methods is that particles are only needed in areas of interest as they do not suffer from the lack of coverage issues of forward methods. No valuable computational time is lost in the rest of the space.

The computational cost of the probabilistic algorithms grows with the number of SDEs to be solved and hence only linearly with respect to the dimension n of the state space $\mathbb{R}^n$, and are very useful in situations where the solution needs to be computed only at a small number of points. A major drawback of a random walk method is its stochastic nature so that the results include statistical errors proportional to $N^{-\frac{1}{2}}$ where N is the number of realizations of the random process in the simulation to evaluate the expected value. Multigrid [3] and multilevel Monte Carlo path simulation [12] methods have been developed to reduce this computational cost. However a large number of independent realizations is still necessary to obtain a reasonable solution accuracy. Here we aim to reduce that aspect of the problem.

The method presented in this paper aims at increasing computational efficiency of the backward random-walk method by using a combination of usual averaging over realizations and spatial averaging where a spatial filter function, for example a Gaussian filter, is used as a point spread function by convolution. We will show that the expected convergence error is of order $\mathcal{O}\left(\varepsilon^2 + 1/\sqrt{N(2M+1)^n} + \Delta t^p\right)$ where ε is the spatial filter width, M is given by $M = \varepsilon/\Delta x$ with Δx the grid size, N is the number of independent simulations started on the full grid, Δt is the size of the time step and $p \geqslant 1$, (p depends on the order of accuracy of the method chosen to solve the SDE system). We also show that this backward spatial filtering method greatly decreases computational time to numerically calculate $u(x,t)$ on a grid of points compared to the traditional Monte-Carlo Averaging methods. This method is also very easy to implement and can be combined with other computational cost reducing methods, multigrid methods [33], stratified sampling [13], multilevel methods [12], for example. Combining those methods with the Backward Spatial Filtering method could reduce the number of independent simulations, N, and the spatial filter width, ε, needed to achieve a desired accuracy. In this paper, we will focus on the advantage of the spatial filtering method to reduce computational cost independently of the method chosen to solve the SDE system.

The paper is organized as follows. In Section 2, we introduce the unsteady linear parabolic problem of interest. In Section 3, we describe the numerical scheme. First we present the probabilistic solution of this problem. Then we propose the spatial filtering method. Finally, the convergence rates are analyzed in detail and the theoretical performances of four filters are discussed. In Section 4, numerical experiments are presented and analyzed. In Section 5, we compare the computational efficiency of the spatial filtering to the one from the Monte Carlo method.

2. Statement of problem

We are interested in the numerical approximation of a density field, $u(x,t) : H \to \mathbb{R}$, representing a solution of a boundary value problem for a linear second-order partial differential equation of the parabolic type:

$$\frac{\partial u}{\partial t} + \sum_{i=1}^{n} b_i(x,t)\frac{\partial u}{\partial x_i} + c(x,t)u + f(x,t) = \sum_{i,j=1}^{n} a_{ij}(x,t)\frac{\partial^2 u}{\partial x_i \partial x_j}. \tag{1}$$

By $\mathbb{R}^n$ we denote the Euclidean space of dimension n. An arbitrary point of an $(n+1)$-dimensional space $\mathbb{R}^{n+1} = \mathbb{R}^n \times \mathbb{R}$ is denoted (x,t) where the point x has coordinates $(x_1,\ldots,x_n)$. Let $T < \infty, H$ will denote a bounded domain in $\mathbb{R}^{n+1}$ lying between the planes $t = 0$ and $t = T$; $H = \{x \in D \subset \mathbb{R}^n,\ 0 \leqslant t \leqslant T < +\infty\}$ where D is a compact domain of $\mathbb{R}^n$. The functions a_{ij}, b_i, c, and f are real and bounded on $H, a_{ij} = a_{ji}$ and $\sum_{i,j=1}^{n} a_{ij}(x,t)\alpha_i\alpha_j > 0$ for $\sum_{i=1}^{n}\alpha_i^2 > 0$. The surface that consists of boundary points of D will be denoted by δD and the direction of the outer normal to the surface δD by ν.

On the boundary of the domain D, the density $u(x,t)$ satisfies the condition:

$$\frac{\partial u}{\partial \nu} + \beta(x,t)u(x,t) = \lambda(x,t), \tag{2}$$

where the functions β and λ are real, with finite values.

We shall investigate the solution of the Cauchy problem for (1)–(2) with initial time condition:

$$u(x, t = 0) = \varphi(x), \tag{3}$$

where $\varphi(x)$ is a given real function.

In what follows, we assume that the coefficients in (1)–(3) satisfy the conditions such that the solution $u(x,t)$ exists, is unique and belongs to $W^{2,p}(D) \forall t > 0$, (see for examples [17]) with:

$$W^{2,p}(D) := \left\{ f \in L^p(D);\ \partial^\alpha f = \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \dots x_n^{\alpha_n}} \in L^p(D),\ \forall \alpha \in \mathbb{N}^n,\ |\alpha| \leqslant 2 \right\}.$$

We provide $W^{2,p}(D)$ with the norm $\|f\|_{W^{2,p}(D)} := \sum_{|\alpha| \leqslant 2} \left\| \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \dots x_n^{\alpha_n}} \right\|_{L^p(D)}$.

We present a probabilistic method to solve this unsteady linear parabolic problem that uses a backward random walk scheme and a combination of usual Monte Carlo averaging and spatial filtering.

3. Numerical approximation of the solution

A stochastic process in the time interval $[0, \infty)$ is defined as a family of random variables $X_\omega(t), t \geqslant 0, \omega \in \Omega$, on a measurable space $(\Omega, \mathcal{F}, \mathcal{P})$. Let $\mathbb{E}_\omega[X]$ be the expected value of a stochastic value X with respect to the realizations $\omega \in \Omega$. Let $W_\omega(t) = (W_\omega^1(t), \cdots, W_\omega^n(t))$ be the collection of n independent one-dimensional Wiener processes. This means that if $t \geqslant 0$, for $i = 1, \dots, n, W_\omega^i(t), \omega \in \Omega$ is a continuous, with variance one, mean zero Gaussian process such that $\mathbb{E}_\omega[W(s)W(t)] = min(s,t)$.

3.1. Probabilistic solution of PDE

The solution of problem (1)–(3) has the probabilistic representation [9]:

$$u(x,t) = \mathbb{E}_\omega[\varphi(X_\omega(0,t,x))Y_\omega(0,t,x,1) + Z_\omega(0,t,x,1,0)], \tag{4}$$

where $(X_\omega(s,t,x), Y_\omega(s,t,x,y), Z_\omega(s,t,x,y,z)),\ 0 \leqslant s \leqslant t \leqslant T$, is the solution backward in time of the Cauchy problem of the following system of SDE:

$$\begin{cases} dX_\omega = -b(X_\omega, s)ds + \sigma(X_\omega, s)dW_\omega(s), & X_\omega(t,t,x) = x, \\ dY_\omega = -c(X_\omega, s)Y_\omega ds, & Y_\omega(t,t,x,y) = y, \\ dZ_\omega = -f(X_\omega, s)Y_\omega ds, & Z_\omega(t,t,x,y,z) = z. \end{cases} \tag{5}$$

Here $(x,t) \in \mathbb{R}^{n+1}, X_\omega \in D$ is a n-dimensional vector, Y_ω and Z_ω are scalars, $b(x,t)$ is the n-dimensional vector compounded from the coefficients $b_i(x,t)$, and the matrix $\sigma(x,t)$ of dimension $n \times n$ is derived from:

$$\sigma(x,t)\sigma^T(x,t) = a(x,t).$$

A typical problem that features a nonuniform advection field will lead to algebraic or exponential separation of nearby particles in time. The solution for $u(x,t)$ is defined only at the location x where the trajectories end at time t. This leads to issues of lack of coverage of the domain of interest when calculating $u(x,t)$ using methods based on forward time. The backward method presented here does not suffer from lack of coverage since the spatial location x at time t is chosen and the SDE is solved backward in time from time t to the initial time $t = 0$. The location of the trajectories at $t = 0$ is only used to evaluate the contribution of the initial condition, $\varphi(X_\omega(0,t,x))$, to the solution. The solution of $u(x,t)$ is hence well defined at each point x chosen to represent the domain of interest.

3.2. Spatial Filtering method

Traditionally the expected value, (4), that gives the local solution $u(x,t)$ is evaluated by applying the Monte-Carlo technique to N independent paths of $(X_{\omega(k)}(s,t,x), Y_{\omega(k)}(s,t,x,1), Z_{\omega(k)}(s,t,x,1,0)),\ k = 1, \dots, N$, solution of the SDE system in (5) by computing the average of the functional:

$$\Psi_{x,t}^{\omega(k)} := \varphi(X_{\omega(k)}(0,t,x))Y_{\omega(k)}(0,t,x,1) + Z_{\omega(k)}(0,t,x,1,0) \tag{6}$$

using:

$$\bar{u}^N(x,t) := \frac{1}{N}\sum_{k=1}^{N}[\Psi_{x,t}^{\omega(k)}].$$

By the law of large numbers, $\bar{u}^N(x,t)$ gives us an approximate value of $u(x,t)$. The quality of approximation depends only on the choice of N. This average converges generally as $N^{-1/2}$. Multigrid and multilevel ideas have been used to improve the

convergence rate of estimations of the expected value of a functional arising from a SDE system [12]. The total number of simulation needed with a traditional Monte Carlo method to solve the system (1)–(3) can rapidly become large as it is equal to the number of points on the grid multiplied by the number of independent realizations to obtain convergence of the averaging.

Here we propose to compute $u(x,t)$ using backward in time paths starting at (x,t) and starting at its spatial neighbouring points as well. We take neighbouring points into account by calculating the convolution of $u(x,t)$ with a spatial filter of width $\varepsilon, F_\varepsilon$.

A spatial filter is defined as a function, $F:\mathbb{R}^n \to \mathbb{R}^+$, such that:

$$\begin{cases} \text{The essential support of } F \text{ is } [-1;1]^n; \\ F(x) \geqslant 0, \quad \forall x \in \mathbb{R}^n; \\ \int_{\mathbb{R}^n} F(x)dx = 1; \\ \int_{\mathbb{R}^n} |x|^2 F(x)dx = K < \infty; \\ F(x_1,\ldots,x_i,\ldots,x_n) = F(x_1,\ldots,-x_i,\ldots,x_n), \quad \forall x \in \mathbb{R}^n, i = 1,\ldots,n. \end{cases}$$

Then we define a spatial filter of width $\varepsilon, F_\varepsilon$, for all $\varepsilon > 0$:

$$F_\varepsilon(x) = \frac{1}{\varepsilon^n} F\left(\frac{x}{\varepsilon}\right). \tag{7}$$

Let μ_j be the jth moment of a function $F:\mathbb{R}^n \to \mathbb{R}$:

$$\mu_j(F) := \int_{\mathbb{R}^n} \left(\sum_{i=1}^n x_i^j\right) F(x)dx, \quad j \in \mathbb{N}.$$

Then we have:

$$\begin{cases} \text{The essential support of } F_\varepsilon \text{ is } [-\varepsilon;\varepsilon]^n; \\ \mu_0(F_\varepsilon) = 1; \\ \mu_1(F_\varepsilon) = 0; \\ \mu_2(F_\varepsilon) = \varepsilon^2 \mu_2(F). \end{cases} \tag{8}$$

We define

$$\begin{aligned} u_\varepsilon(x,t) &:= F_\varepsilon(x) * u(x,t), \\ &= \int_{R^n} F_\varepsilon(y-x)u(y,t)dy, \\ &= \int_{R^n} F_\varepsilon(y-x)\mathbb{E}_\omega[\Psi^\omega_{y,t}]dy. \end{aligned}$$

The spatial filter F_ε, (7), has essential support $[-\varepsilon;\varepsilon]^n$. A neighbourhood of $x \in \mathbb{R}^n$ of width $\varepsilon > 0$ will be denoted by $I_{x,\varepsilon} = (x_1-\varepsilon; x_1+\varepsilon) \times \cdots \times (x_n-\varepsilon; x_n+\varepsilon)$. Let $k = (k_1,\ldots,k_n)$ be a vector of indices with $M \geqslant 0$, $k_i = -M,\ldots,M$ and $i = 1,\ldots,n$. The domain $I_{x,\varepsilon}$ is partitioned in a grid $I^g_{x,\varepsilon}$ with points located at coordinates $y_k = (x_1 + k_1\Delta x,\ldots,x_i + k_i\Delta x,\ldots,x_n + k_n\Delta x)$ with $\Delta x = \varepsilon/M$.

Let $\Psi^{\omega(y_k)}_{y_k,t}$ be defined by (6) using the exact solution of the SDE system in (5) where for each $y_k \in I^g_{x,\varepsilon}$ an independent path $\omega(y_k)$ is generated. Similarly, let $\Psi^{\omega(y_k)}_{y_k,t,\Delta t}$ be the approximate value of $\Psi^{\omega(y_k)}_{y_k,t}$ using the numerical solution given by a weak SDE solver of order $p \geqslant 1$ with a time step Δt. We define:

$$\begin{aligned} \tilde{u}^M_\varepsilon(x,t) &:= \sum_{y_k \in I^g_{x,\varepsilon}} F_\varepsilon(y_k - x)\Psi^{\omega(y_k)}_{y_k,t}\Delta x^n, \\ \tilde{u}^M_\varepsilon(x,t,\Delta t) &:= \sum_{y_k \in I^g_{x,\varepsilon}} F_\varepsilon(y_k - x)\Psi^{\omega(y_k)}_{y_k,t,\Delta t}\Delta x^n, \end{aligned} \tag{9}$$

as the approximations of $u(x,t)$ by a Backward Spatial Filtering method of width ε.

We now investigate how the solution $u(x,t)$ of (1)–(3) is approximated by $\tilde{u}^M_\varepsilon(x,t)$, (9). We will show how $\tilde{u}^M_\varepsilon(x,t)$ converges toward $u(x,t)$ when $M \to \infty$ and $\varepsilon \to 0$.

3.3. Convergence study

We state theoretical results concerning the theoretical error of our approximations and the rates of convergence of this method.

Theorem 1 (*Convergence theorem of Backward Probabilistic Method With Spatial Filtering*). *For large enough M and small enough ε, the error of the approximate solution $\tilde{u}^M_\varepsilon(x,t,\Delta t)$ is given by:*

$$\|u(x,t)-\tilde{u}_\varepsilon^M(x,t,\Delta t)\|_{L^p}=\mathcal{O}\left(\varepsilon^2+\frac{1}{\sqrt{(2M+1)^n}}+\Delta t^p\right), \tag{10}$$

except for an event of low probability ε_p, (11), *where* ε *is the spatial filter width,* M *is given by* $M=\frac{\varepsilon}{\Delta x}$ *with* Δx *the grid size,* Δt *is the size of the time step of SDE solver and* $p\geqslant 1$.

Proof. The approximation error can be broken into three terms, a spatial filtering term c_f, a sampling term c_s and a time splitting term c_t.

$$\|u(x,t)-\tilde{u}_\varepsilon^M(x,t,\Delta t)\|_{L^p}=\underbrace{\|u(x,t)-u_\varepsilon(x,t)\|_{L^p}}_{c_f}+\underbrace{\|u_\varepsilon(x,t)-\tilde{u}_\varepsilon^M(x,t)\|_{L^p}}_{c_s}+\underbrace{\|\tilde{u}_\varepsilon^M(x,t)-\tilde{u}_\varepsilon^M(x,t,\Delta t)\|_{L^p}}_{c_t}.$$

Assuming the SDE solver is of weak order p with $p\geqslant 1$, there exists $C>0$ such that:

$$|\Psi_{y_k,t}^{\omega(y_k)}-\Psi_{y_k,t,\Delta t}^{\omega(y_k)}|\leqslant C\Delta t^p.$$

Then the time splitting error is:

$$c_t=\mathcal{O}(\Delta t^p).$$

Lemma 2. *If* $f\in W^{2,p}(\mathbb{R}^n)$ *and* F_ε *is a spatial filter as defined in* (8), *then:*

$$\|F_\varepsilon * f-f\|_{L^p(\mathbb{R}^n)}\leqslant\frac{1}{2}\varepsilon^2\mu_2(F)\|f\|_{W^{2,p}(\mathbb{R}^n)}.$$

Proof. Let $f\in W^{2,p}(\mathbb{R}^n)$, using Taylor expansion with integral remainder, we have:

$$f(x-y)=f(x)+\sum_{i=1}^n y_i\frac{\partial f(x)}{\partial x_i}+\int_0^1(1-t)\sum_{i,j=1}^n y_iy_j\frac{\partial^2 f(x-ty)}{\partial x_i\partial x_j}dt.$$

Then

$$F_\varepsilon * f(x)=\int_{\mathbb{R}^n}f(x-y)F_\varepsilon(y)dy=\int_{\mathbb{R}^n}f(x)F_\varepsilon(y)dy+\int_{R^n}\sum_{i=1}^n y_i\frac{\partial f(x)}{\partial x_i}F_\varepsilon(y)dy+\int_{\mathbb{R}^n}\int_0^1(1-t)\sum_{i,j=1}^n y_iy_j\frac{\partial^2 f(x-ty)}{\partial x_i\partial x_j}F_\varepsilon(y)dt\,dy.$$

Using the properties of the spatial filter F_ε, we have:

$$\int_{R^n}f(x)F_\varepsilon(y)dy=f(x),$$

and

$$\int_{R^n}\sum_{i=1}^n y_i\frac{\partial f(x)}{\partial x_i}F_\varepsilon(y)dy=\sum_{i=1}^n\frac{\partial f(x)}{\partial x_i}\int_{R^n}y_iF_\varepsilon(y)dy=0.$$

Then

$$F_\varepsilon * f(x)=f(x)+\int_{\mathbb{R}^n}\int_0^1(1-t)\sum_{i,j=1}^n y_iy_j\frac{\partial^2 f(x-ty)}{\partial x_i\partial x_j}F_\varepsilon(y)dtdy.$$

Let $\tilde{F}_\varepsilon(y)=\frac{1}{t^{n+2}}F_\varepsilon(\frac{y}{t})$ and introduce the change of variable $z=ty$:

$$\mu_2(F_\varepsilon)=\int_{\mathbb{R}^n}x^2F_\varepsilon(x)dx=\frac{1}{t^{n+2}}\int_{\mathbb{R}^n}z^2F_\varepsilon\left(\frac{z}{t}\right)dz=\int_{\mathbb{R}^n}x^2\tilde{F}_\varepsilon(z)dz=\mu_2(\tilde{F}_\varepsilon).$$

Then,

$$\int_{\mathbb{R}^n}\sum_{i,j=1}^n y_iy_j\frac{\partial^2 f(x-ty)}{\partial x_i\partial x_j}F_\varepsilon(y)dy=\sum_{i,j=1}^n\int_{\mathbb{R}^n}\frac{\partial^2 f}{\partial x_i\partial x_j}(x-z)z_iz_j\tilde{F}_\varepsilon(z)dz=\sum_{i,j=1,i\neq j}^n 2\frac{\partial^2 f}{\partial x_ix_j}*(x_ix_j\tilde{F}_\varepsilon)+\sum_{i=1}^n\frac{\partial^2 f}{\partial x_i^2}*(x_i^2\tilde{F}_\varepsilon).$$

Thus, we have:

$$\left\|\int_{\mathbb{R}^n}y^TD^2f(x-ty)yF_\varepsilon(y)dy\right\|_{L^p}\leqslant 2\sum_{i,j=1,i\neq j}^n\left\|\frac{\partial^2 f}{\partial x_ix_j}*(x_ix_j\tilde{F}_\varepsilon)\right\|_{L^p}+\sum_{i=1}^n\left\|\frac{\partial^2 f}{\partial x_i^2}*(x_i^2\tilde{F}_\varepsilon)\right\|_{L^p}.$$

Using the fact that $||f*g||_{L^p} \leqslant ||f||_{L^p}||g||_{L^1}$ and that $\forall i,j=1,\dots,n,\ i\neq j,\ \|x_i x_j \tilde{F}_\varepsilon\|_{L^1}=0$ by the symmetry property of $F_\varepsilon(x)$, we get:

$$\left\|\int_{R^n} y^T D^2 f(x-ty) y F_\varepsilon(y)dy\right\|_{L^p} \leqslant \sum_{i=1}^n \left\|\frac{\partial^2 f}{\partial x_i^2}\right\|_{L^p} \|x_i^2 \tilde{F}_\varepsilon\|_{L^1}.$$

Hence,

$$\|F_\varepsilon * f - f\|_{L^p} \leqslant \sum_{i=1}^n \int_0^1 (1-t)\left\|\frac{\partial^2 f}{\partial x_i^2}\right\|_{L^p} \|x_i^2 \tilde{F}_\varepsilon\|_{L^1} dt \leqslant \frac{1}{2}\sum_{i=1}^n \left\|\frac{\partial^2 f}{\partial x_i^2}\right\|_{L^p} \|x_i^2 \tilde{F}_\varepsilon\|_{L^1}.$$

For each $i=1,\dots,n$:

$$\|x_i^2 \tilde{F}_\varepsilon\|_{L^1} \leqslant \sum_{i=1}^n \|x_i^2 \tilde{F}_\varepsilon\|_{L^1} = \mu_2(\tilde{F}_\varepsilon) = \mu_2(F_\varepsilon).$$

And,

$$\|F_\varepsilon * f - f\|_{L^p} \leqslant \frac{1}{2}\mu_2(F_\varepsilon)\|f\|_{W^{2,p}} \leqslant \frac{1}{2}\varepsilon^2 \mu_2(F)\|f\|_{W^{2,p}}. \qquad \square$$

Corollary 3. *The error between the convolution of $u(x,t)$ solution of (1)–(3) with $F_\varepsilon(x)$ and $u(x,t)$ is proportional to ε^2, where ε is the spatial filter width.*

Proof. The spatial filtering error, c_f, can be estimated using Lemma 2 which gives us the estimation:

$$c_f \leqslant \frac{1}{2}\varepsilon^2 \mu_2(F)||u||_{W^{2,n}} \leqslant K_f \varepsilon^2. \qquad \square$$

Now we study the sampling error:

$$c_s = ||u_\varepsilon(x,t) - \tilde{u}_\varepsilon^M(x,t)||_{L^p}.$$

Using a simple one-to one correspondence between elements of Ω and a given solution of the SDE (5), each $\omega(y)$ in Ω corresponds to one realization of the random walks starting at $y\in\mathbb{R}^n$, we get:

$$c_s = \left\|\int_{R^n} F_\varepsilon(y-x)\mathbf{E}_\omega[\Psi^\omega_{y,t}]dy - \sum_{y_k\in I^g_{x,\varepsilon}} F_\varepsilon(y_k-x)\Psi^{\omega(y_k)}_{y_k,t}\Delta x^n\right\|_{L^p} = \left\|\int_{R^n} F_\varepsilon(y-x)\mathbf{E}_\omega[\Psi^\omega_{y,t}]dy - \sum_{y_k\in I^g_{x,\varepsilon}} F_\varepsilon(y_k-x)\mathbf{E}_{\omega(y_k)}[\Psi^{\omega(y_k)}_{y_k,t}]\Delta x^n\right\|_{L^p}$$
$$+ \left\|\sum_{y_k\in I^g_{x,\varepsilon}} F_\varepsilon(y_k-x)\left(\mathbf{E}_{\omega(y_k)}[\Psi^\omega_{y_k,t}] - \Psi^{\omega(y_k)}_{y_k,t}\right)\Delta x^n\right\|_{L^p} := A + B.$$

The expression A is the discretization error of the integral:

$$A := \left\|\int_{R^n} F_\varepsilon(y-x)\mathbf{E}_\omega[\Psi^\omega_{y,t}]dy - \sum_{y_k\in I^g_{x,\varepsilon}} F_\varepsilon(y_k-x)\mathbf{E}_{\omega(y_k)}[\Psi^{\omega(y_k)}_{y_k,t}]\Delta x^n\right\|_{L^p}$$
$$= \left\|\int_{R^n} F_\varepsilon(y-x)u(y,t)dy - \sum_{y_k\in I^g_{x,\varepsilon}} F_\varepsilon(y_k-x)u(y_k,t)]\Delta x^n\right\|_{L^p}.$$

An estimation of A comes from the formula for the error of the trapezoidal rule for numerical integration for a function $f\in C^2[a,b]$, [8]. In 1D,

$$\left|\int_a^b f(x)dx - h\left[\frac{1}{2}f(a) + f(a+h) + \cdots + f(a+(M-1)h) + \frac{1}{2}f(b)\right]\right| = \frac{(b-a)^3}{12M^2}|f''(\xi)|,$$

with $a<\xi<b$.

Since F_ε has compact support $[-\varepsilon,\varepsilon]^n$, we use the multidimensional version of the above formula and obtain:

$$A = \mathcal{O}(\varepsilon\Delta x^2).$$

$$B := \left\|\sum_{y_k\in I^g_{x,\varepsilon}} F_\varepsilon(y_k-x)\left(\mathbf{E}_\omega[\Psi^\omega_{y_k,t}] - \Psi^{\omega(y_k)}_{y_k,t}\right)\Delta x^n\right\|_{L^p}.$$

The Law of Large Numbers implies that $\sum_{y_k \in I^g_{x,\varepsilon}} F_\varepsilon(y_k - x)\mathbf{E}_\omega[\Psi^{\omega(y_k)}_{y_k,t}]\Delta x^n - \sum_{y_k \in I^g_{x,\varepsilon}} F_\varepsilon(y_k - x)\Psi^{\omega(y_k)}_{y_k,t}\Delta x^n \to 0$ a.s. with a rate of convergence proportional to one over the square root of the number of points in the sum, i.e. $\frac{1}{\sqrt{(2M+1)^n}}$.

In fact,

Lemma 4 (Bennett's inequality). *Let Y_k, be independent bounded random variables with mean zero, variances σ_k^2 and $|Y_k| \leqslant C$. Let $S = \sum_k Y_k, V \geqslant \sum_i \sigma_k^2$. Then for all $v > 0$,*

$$Pr\{|S| \geqslant v\} \leqslant 2exp\left[-\frac{1}{2}v^2V^{-1}B(CvV^{-1})\right],$$

where $B(\lambda) = 2\lambda^{-2}[(1+\lambda)ln(1+\lambda) - \lambda], \lambda > 0$.

See proof of Lemma 4 in [25], Appendix B.

Let $Y_k = F_\varepsilon(y_k - x)\left[\mathbf{E}_\omega[\Psi^{\omega(y_k)}_{y_k,t}] - \Psi^{\omega(y_k)}_{y_k,t}\right]\Delta x^n$. We have $E[Y_k] = 0$ and

$$|Y_k| \leqslant K_1 \frac{1}{\varepsilon^n}\max_{|x|\leqslant 1}F(x)\left(\frac{\varepsilon}{M}\right)^n = K_1 \max_x F(x)\frac{1}{M^n} = C.$$

Similarly,

$$\sum_{y_k \in I^g_{x,\varepsilon}} Var[Y_k] \leqslant K_2^2(2M+1)^n\left(\max_x F(x)\frac{1}{M^n}\right)^2 = V,$$

where K_1 and K_2 are bounds on the values and the variance of $\mathbf{E}_\omega[\Psi^{\omega(y_k)}_{y_k,t}] - \Psi^{\omega(y_k)}_{y_k,t}$ and depend only on T, and bounds of $a_{ij}(x,t), b_i(x,t), c(x,t)$ and $f(x,t)$.

Let K be a positive constant and M large enough such that $v = \frac{KK_2}{\sqrt{(2M+1)^n}}$ and CvV^{-1} can be considered small. Then,

$$Pr\left[|S| \geqslant \frac{KK_2}{\sqrt{(2M+1)^n}}\right] \leqslant 2\left(1 + \mathcal{O}\left(CvV^{-1}\right)\right)exp\left(-\frac{1}{2}v^2V^{-1}\right).$$

We have:

$$\frac{1}{2}v^2V^{-1} = \frac{K^2}{2\max_x F(x)(2+1/M)^{2n}}.$$

Let

$$\varepsilon_p := 2exp\left(-\frac{K^2}{2\max_x F(x)(2+1/M)^{2n}}\right). \tag{11}$$

The constant K can be chosen such that $K \geqslant \max_x F(x)2^n\sqrt{2ln(2/\varepsilon_p)}$ then,

$$|S| \leqslant \frac{KK_2}{\sqrt{(2M+1)^n}},$$

except for an event of probability less or of the order of ε_p. The value of the constant K to obtain a desired ε_p depends only on the filter choice and on n, the dimension of the Eucledian space of coordinates x.

Thus:

$$E[c_s] = \mathcal{O}\left(\frac{1}{\sqrt{(2M+1)^n}}\right),$$

with high probability. □

Theorem 5 (*Convergence theorem of Backward Probabilistic Method With combination of Spatial Filtering and Monte Carlo averaging*). *For large enough M and small enough ε, the error of the approximate solution $\sum_{i=1}^N \bar{u}^M_\varepsilon(x,t,\Delta t)$ is given by:*

$$\left\|u(x,t) - \sum_{i=1}^N \bar{u}^M_\varepsilon(x,t,\Delta t)\right\|_{L^p} = \mathcal{O}\left(\varepsilon^2 + \frac{1}{\sqrt{N(2M+1)^n}} + \Delta t^p\right),$$

except for an event of low probability ε_p, (11), where ε is the spatial filter width, M is given by $M = \frac{\varepsilon}{\Delta x}$with Δx the grid size, N is the number of independent simulations, ω, started on the full grid and Δt is the size of the time step and $p \geqslant 1$.

Proof. The proof of this theorem is similar to proof of Theorem 1. □

The tuning parameters to control accuracy of the Backward Spatial Filtering method are hence ε, the width of the spatial filter, $\Delta x = \varepsilon/M$, the grid step size, Δt, the time step used to solve the SDE system and N the number of random independent simulations on the grid.

3.4. Comparison of filters

We compare the performances of four different filters, a Gaussian filter, a Triangular filter, a Square filter and a Cos^2 filter (see Fig. 1(a)). The method presented here is not restricted to these four filters. In certain cases, other spatial filters such as Savitzky–Golay smoothing filters [29] could be better suited.

A Square filter is defined by:

$$\forall x \in \mathbb{R}^n, \quad \begin{cases} F(x) = \frac{1}{2^n}, & \text{if } \forall i\, |x_i| \leqslant 1, \\ F(x) = 0, & \text{otherwise.} \end{cases} \tag{12}$$

A Triangular filter is defined by:

$$\forall x \in \mathbb{R}^n, \quad \begin{cases} F(x) = \prod_{i=1}^{n} (1 - |x_i|), & \text{if } \forall i\, |x_i| \leqslant 1, \\ F(x) = 0, & \text{otherwise.} \end{cases} \tag{13}$$

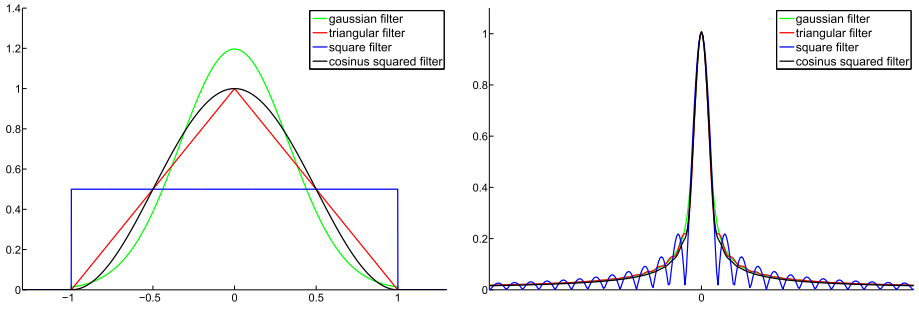
A Gaussian filter is defined by:

$$\forall x \in \mathbb{R}^n, \quad \begin{cases} F(x) = \prod_{i=1}^{n} \left(\frac{3}{\sqrt{2\pi}}\right) e^{-\frac{(3x_i)^2}{2}}, & \text{if } \forall i\, |x_i| \leqslant 1, \\ F(x) = 0, & \text{otherwise.} \end{cases} \tag{14}$$

A Cos^2 filter is defined by:

$$\forall x \in \mathbb{R}^n, \quad \begin{cases} F(x) = \prod_{i=1}^{n} cos^2\left(\frac{\pi |x_i|}{2}\right), & \text{if } \forall i\, |x_i| \leqslant 1, \\ F(x) = 0, & \text{otherwise.} \end{cases} \tag{15}$$

A convolution filter is essentially a weighted average to estimate the value at a point conditioned by the value of its neighbours. Since the solution $u(x,t) \in W^{2,p}(D)$ then $u(x,t)$ is smooth, but the probabilistic method results in $\Psi^{\omega}_{x,t}(k)$ being a very noisy signal unless Monte-Carlo averaging is applied. A justification for the choice of spatial filters presented above is their frequency response. The frequency response of a convolution filter, i.e. its effect on different spatial frequencies, can be seen by taking the Fourier Transform of the filter (see Fig. 1(b)). We wish to uncover the low frequency solution from the simulation data, $\Psi^{\omega}_{x,t}(k)$. All four filters presented above, (12)–(15), attenuate high frequencies more than low frequencies and are called low-pass filters (see Fig. 1(b)). A narrow main peak in the frequency response is desirable. The Square filter is the best in this sense but it exhibits the most oscillations in its frequency response. The Triangular filter has a better oscillation attenuation. The Gaussian and Cos^2 filters on the other hand show no oscillations. Indeed the Gaussian and Cos^2 filter have a good balance between side oscillation attenuation and main peak width. So by choosing an appropriately sized Gaussian or Cos^2 filter we can be fairly confident about what range of spatial frequencies are still present in the solution after filtering, which is not the case of the Square filter.



(a) 4 types of filters (b) FFT

Fig. 1. The 4 types of filters: in blue a continuous Gaussian filter, (14), in green a Triangular filter, (13), in red a Square filter, (12), in black a Cos^2 filter, (15). (a) Continuous representation. (b) Frequency Response. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4. Numerical investigation: illustrative numerical experiments

We can test the implementation of the methods, using problems for which precise numerical solutions can be independently calculated. In the following examples we will use the Stochastic Runge–Kutta method described in [27] to solve the SDE systems.

4.1. 1D example

To illustrate our new method with spatial filtering we will show results first for a 1D test example, a 1D Fokker–Planck equation:

$$\frac{\partial}{\partial t}u(x,t)+\frac{\partial}{\partial x}(v(x)u(x,t))=D\frac{\partial^2}{\partial x^2}u(x,t) \tag{16}$$

in $H=[-1,1]\times[0,T_f],\ T_f>0$, with advective velocity $v(x)$:

$$v(x)=-\sin(\pi x), \tag{17}$$

with uniform initial condition $u(x,t)=1,\ \forall x\in[-1,1]$ and periodic boundary conditions.

Advection–diffusion equations are a special case of the linear parabolic Eq. (1). As described above for the general case, $u(x,t)$ can be calculated using the SDEs:

$$\begin{cases} dX_\omega=-v(X_\omega,s)ds+\sigma dW_\omega(s), & X_\omega(t,t,x)=x,\\ dY_\omega=-\frac{\partial}{\partial x}v(X_\omega,s)Y_\omega ds, & Y_\omega(t,t,x,y)=y, \end{cases} \tag{18}$$

where $\sigma=\sqrt{D}$.

Then,

$$u(x,t)=\mathbb{E}_\omega[\varphi(X_\omega(0,t,x))Y_\omega(0,t,x,1)]=\mathbb{E}_\omega\left[\varphi(X_\omega(0,t,x))exp\left(-\int_0^t\frac{\partial}{\partial x}v(X_\omega(0,t,x))dt\right)\right]. \tag{19}$$

If D = 0, the solution is given exactly by integration along the trajectories:

$$u(x,t)=\varphi(X(0,t,x))exp\left(-\int_0^t\frac{\partial}{\partial x}v(X(0,t,x))dt\right). \tag{20}$$

With and without diffusion, Eqs. (19) and (20) tell us that the particle density at a point x depends on where the particles at that point come from and also on their trajectories backward in time. If, on average, they spent most of their time travelling in regions where $\nabla\cdot v>0$, i.e. regions of particle depletion, then $u(x,t)$ decreases with time. If, on average, they spent most of their time travelling in regions where $\nabla\cdot v<0$, i.e. regions of particle focusing, then $u(x,t)$ increases with time.

On $[-1,1]$, the velocity field, $v=-sin(\pi x)$, has three fixed points, one stable, $x=0$, and the two others unstable, $x=\pm1$, see Fig. 2(a). Also shown in Fig. 2 are the regions of particle depletion or focusing for our example. If there is no diffusion, all particles subject to this velocity field will go to $x=0$. When diffusion is present, particles will move away from the stable fixed point through Brownian motion and spend some time in regions of particle depletion which lowers the particle density at the stable fixed point from that in the non diffuse case.

For this test case, we know the analytical solution of the steady state. We have:

$$\lim_{t\to\infty}u(x,t)=u_\infty(x)=\frac{1}{I_0(1/(\pi D))}exp(cos(\pi x)/(\pi D)), \tag{21}$$

where $I_v(x)$ is the modified Bessel function of the first kind. Fig. 2(b) shows the steady state solution for several diffusion coefficients. For small values of D, the solution is well approximated by a zero mean Gaussian on the real line. When $D\to0$, the solution converges to a Dirac delta distribution centred at $x=0$.

We now study how the approximation error behaves when the grid and the timestep of the SDE solver are fixed (Δx and Δt fixed) but varying the filter width ε used for filtering and hence the number of points M used as $\varepsilon=M\Delta x$. To solve the SDE system (18), we use a 4 stage Stochastic Runge Kutta Method with weak order 2 and order 3 for the deterministic part inspired from [27]. Theorem 1 tells us that the error is:

$$|u(x,t)-\tilde{u}_\epsilon^M(x,t)|_2\leqslant\frac{\varepsilon^2}{2}\mu_2(F)||u||_{W^{2,2}}+\frac{K_s}{\sqrt{2M+1}}+K_t\Delta t^2.$$

The filtering error is bounded by $K_f\varepsilon^2\mu_2(F)$ where K_f is independent of the choice of filter F. In order to compare the four spatial filters presented above in a fair manner, we choose a width ε for each filter such that the value of $\varepsilon^2\mu_2(F)$ is the same for all four filters. The value of $\mu_2(F)$ for the four spatial filters is given in Table 1. When the spatial grid is fixed, the sampling error is bounded by a function of the form $1/\sqrt{\varepsilon}$. As $\mu_2(F_\varepsilon)=\varepsilon^2\mu_2(F)$ is set to be the same for all four filters but ε is different

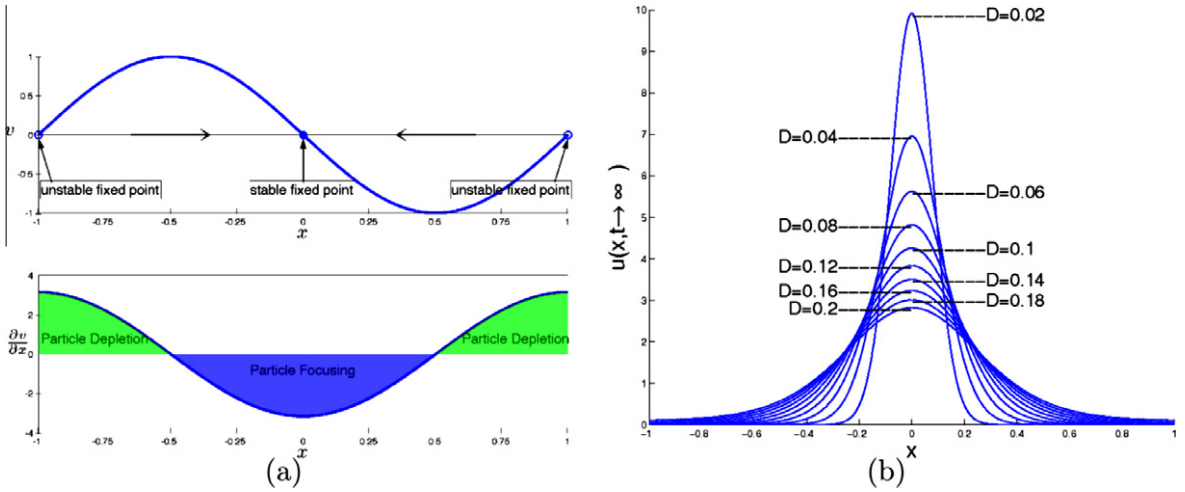


Fig. 2. (a) Study of the advective velocity $v(x) = -\sin(\pi x)$ and its divergence $\frac{\partial v}{\partial x}$. (b) Concentration at $t \to \infty$ for different values of diffusion coefficient D when the advection velocity is $v(x) = -\sin(\pi x)$.

Table 1
Second moment of the four spatial filters in $\mathbb{R}^n$.

	Square filter	Triangular filter	Gaussian filter	Cos^2 filter
$\mu_2(F)$	$\frac{n}{3}$	$\frac{n}{6}$	$\approx n$	$\frac{n(\pi^2-6)}{3\pi^2}$

for each filter, we study the error term as a function of $\mu_s = \sqrt{\mu_2(F_\varepsilon)} = \varepsilon\sqrt{\mu_2(F)}$ which behaves like ε since $\mu_2(F)$ only depends on the choice of filter.

We compare solutions given by our new method to the analytical solution of the steady state in (21) and to the solution given by a commercial finite element software at finite time. When the diffusion coefficient in our test case is more than 0.002, (the Péclet number is less than 1000), then this simple one dimensional advection–diffusion equation can be easily and very accurately solved by any commercial finite element software. In Fig. 3(a) we show the state of simulation at time $t = 1$ with $D = 0.2$ for our 1D example. The values $\Psi_{y_k,t}^{\omega(y_k)}$ for $y_k = [-1 : 5e-3 : 1]$ are shown together with the finite element solution at $t = 1$ and the steady state solution. For each point y_k, independent Brownian motion is used to solve the SDE system and hence the value $\Psi_{y_k,t}^{\omega(y_k)}$ show their stochastic nature. Traditional Monte Carlo averaging over multiple stochastic simulations would result in convergence to the finite element solution of the PDE which is seen in black in the figure. But, it is hard to predict from the noisy $\Psi_{y_k,t}^{\omega(y_k)}$ the shape of the solution after only one set of realizations. In Fig. 3(b) and (c), the filtered approximate solution $\tilde{u}(t,x)$ is plotted in green line for a Gaussian filter, in red and line a Triangular filter and in blue line a Square filter, and in black line for Cos^2 filter at $t = 1$ and $t = 0.5$ with $\mu_s = 0.05$ for each filter. The zoom around $x = 0$ in Fig. 3(c), permits to see that after Gaussian, Triangular and Cos^2 filtering, the solution is very smooth whereas the Square filtered solution, although giving the correct shape, is still noisy. This is indicated by the study of the FFT of the four different filters shown in Fig. 1(b). Notice how the Square filter does not uniformly dampen higher frequencies and hence is expected to keep a fair amount of high-frequency content.

To examine how error changes with ε, we plot the L^2 norm of error versus the square root of the second moment of the four filters for $D = 0.05$ on a Log–Log scale in Fig. 4. The curves are very similar for the four filters. For small $\mu_s = \sqrt{\mu_2(F_\varepsilon)}$, which is equivalent to small ε, the L^2 norm of the error decays rapidly and can be shown to scale as $1/\sqrt{\mu_s}$. Theorem 1 tells us it should decay with a $1/\sqrt{2M+1} = 1/\sqrt{2\varepsilon/\Delta x + 1}$ rule. For small ε, we can Taylor expand this expression and show that this error is of order $\sqrt{\Delta x}/\mu_s$ which gives the observed $1/\mu_s$ decay for fixed Δx. For large $\mu_s = \sqrt{\mu_2(F_\varepsilon)}$ i.e. large ε it increases as ε^2 as predicted in Theorem 1.

We find coefficient α, β such that $\frac{\alpha}{\sqrt{\mu_s}} + \beta\mu_s^2$ fits the curve for the Gaussian filter (Fig. 5) for different coefficient of diffusion D. In Fig. 5(a), the coefficient α is plotted. As it corresponds to the stochastic part of the algorithm, its value changes with the random realization and should not depend strongly with the coefficient of diffusion D. Its mean value is 0.012. The values of coefficient β are plotted in Fig. 5(b). According to Lemma 2, the numerical approximation error from filtering is given by $c_f = ||F_\varepsilon * u - u||_2 \leqslant \frac{\varepsilon^2}{p}\mu_2(F)||u||_{W^{2,p}}$. The filtering error depends on the second derivatives of the solution $u(x,t)$. These derivatives are highly dependent on the coefficient of diffusion. Knowing the steady state solution for our test example, we can estimate $||u||_{W^{2,p}}$.

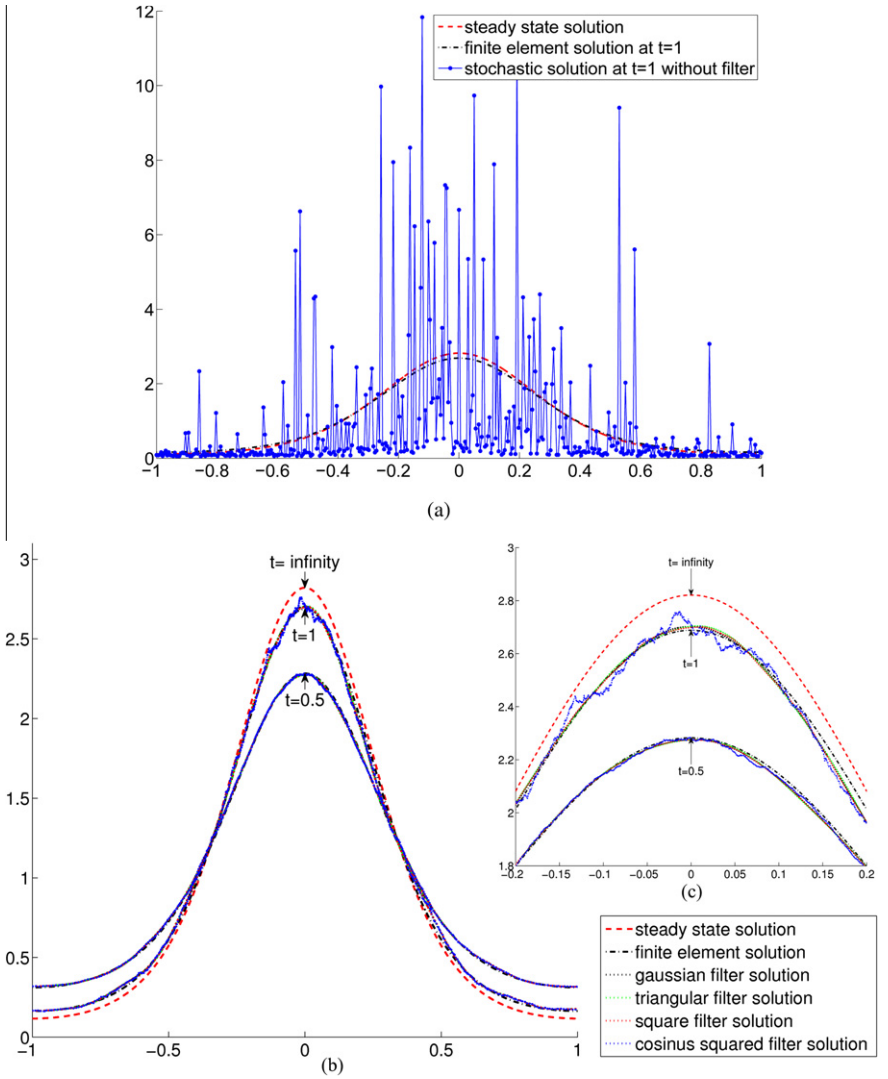


Fig. 3. Comparison of the approximate solution of (16,17) with $D = 0.2$ before filtering and for 4 different spatial filters using $dx = 5e-5$ and $\mu_2(F_\varepsilon) = 0.0025$ to the solution given by a commercial finite element software (black -.- line) and the exact steady state solution, (21), (red - - - line). (a) Solution before filtering. The values of $\Psi_{y_k,t}^{\omega(y_k)}$ at $t = 1$ for $y_k = [-1 : 5e-3 : 1]$ in blue. (b) Filtered approximate solution $\bar{u}_\varepsilon^M(x,t,\Delta t)$ is plotted in green... line for a Gaussian filter, in red... line for a Triangular filter and in blue... line for a Square filter, and in black... line for Cos^2 filter at $t = 1$ and $t = 0.5$. (c) Zoom around $x = 0$ of figure (b). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\frac{\partial^2 u_\infty}{\partial x^2} = \frac{1}{I_0(1/(\pi D))}[-D\pi cos(\pi x) + sin(\pi x)^2]exp(cos(\pi x)/(\pi D)),$$

$$\|u\|_{W^{2,p}} \leqslant max_{x\in[-1,1]}\left|\frac{\partial^2 u_\infty}{\partial x^2}\right| = \frac{\pi exp(1/(\pi D))}{D I_0(1/(\pi D))} \approx \pi/D + O(1/D^2)$$

and thus the filtering error behaves like $1/D$ for large D. We observe this relationship for coefficient β as can be seen in Fig. 5(b). We can then estimate the value of β for other coefficient of diffusion D using a $1/D$ approximation and the value of $\beta_0 = 27.7240$ at $D_0 = 0.05$ given by the simulations.

$$\beta = \frac{\beta_0 D_0}{D}.$$

The difference of rates of convergence of the error for the sampling term and the filtering terms results in the existence of an optimal value of μ_s and hence of filter with ε for which the L^2 norm of the error is minimal (Fig. 6). For $D = 0.05$, the best $\mu_s = \sqrt{\mu_2(F_\varepsilon)} \approx 0.0281$ and at this value of μ_s the L^2 norm of the error for the four filters are about the same but higher for the Square filter than the Triangular filter, Gaussian filter, and Cos^2 filter.

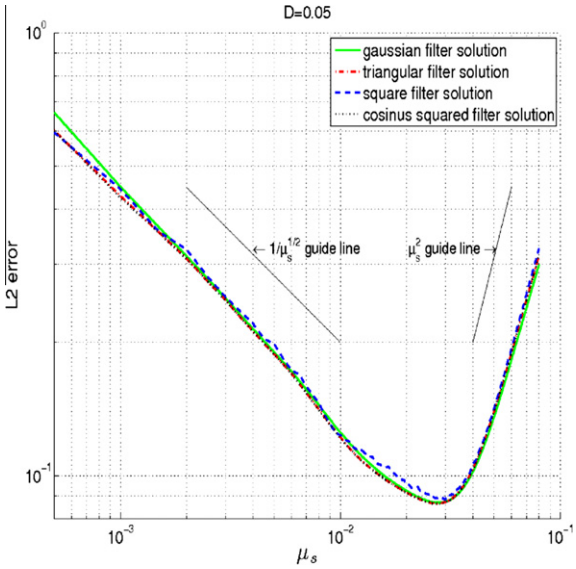


Fig. 4. Plot in Log–Log scale of the L^2 error for different values of $\mu_s = \sqrt{\mu_2(F_\varepsilon)}$ of the filters, in green line for a Gaussian filter, in red dash–dot line for a Triangular filter, in blue dashed line for a Square filter, and in black dotted line for Cos^2 filter at $t = 1$ in the case when $D = 0.05$. Guide lines for $1/\sqrt{\mu_s}$ and μ_s^2 slopes are shown as Theorem 1 shows that the error is bounded by a function of the form $\alpha/\sqrt{\mu_s} + \beta\mu_s^2$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

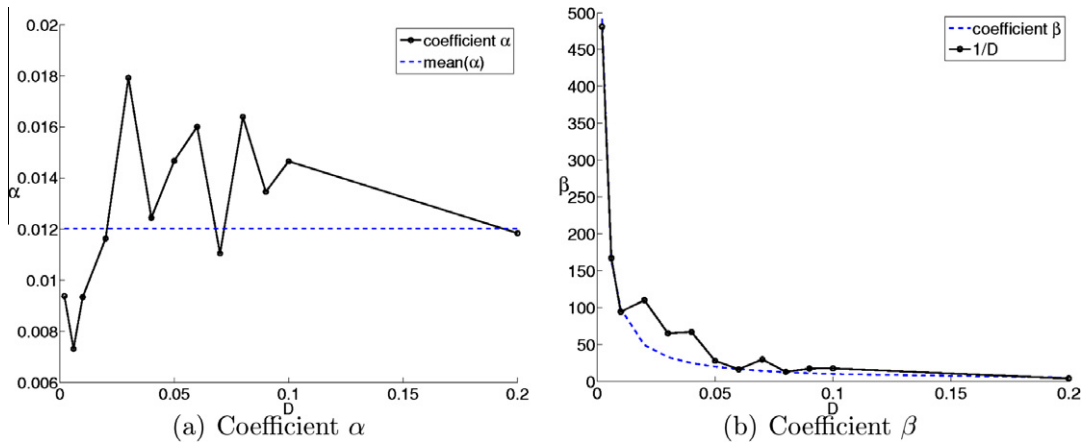


Fig. 5. Plot of the α and β coefficient founds by matching the curves for different coefficient of diffusion D in black line. In figure (a), the mean value of α is given by the dashed blue line. In figure (b), $1/D$ is also plotted in blue dashed line. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

We can estimate the value of the best μ_s at any D using the mean value of $\alpha = 0.012$ and assuming the L^2 error behaves as $\frac{\alpha}{\sqrt{\mu_s}} + \beta\mu_s^2$.

$$\mu_s = \left(\frac{\alpha D}{4\beta_0 D_0}\right)^{2/5}.$$

This formula shows that $\mu_s = 0$ when $D = 0$, which corresponds to saying that no averaging is necessary. When $D = 0$, the solution of the PDE is given by deterministic trajectories. The result for our test example is given in Fig. 6 where the blue dashed line is the estimated μ_s and the black full line show the μ_s obtained from the simulations.

4.2. 2D example

We apply our method to the study of the transport of density by a perturbed cellular, divergence-free velocity field and small diffusion. The example presented here is motivated by the study of optimal mixing with an array of vortices with alternating rotations, by means of Lorentz forces in a two-dimensional fluid layer that carries an electric current [20].

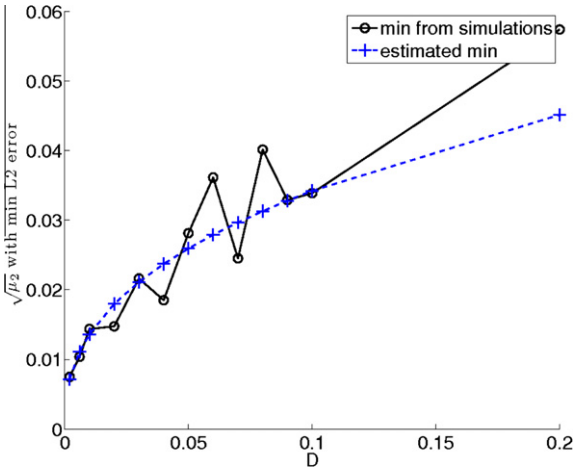


Fig. 6. Plot of the $\mu_s = \sqrt{\mu_2}$ that gives the lowest L^2 error for different values of D where the blue dashed line is the estimated μ_s and the black full line show the μ_s obtained from the simulations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

We consider two separate arrays with different alignments. The corresponding advection velocity field is the sum of two rotating fields:

$$v(x_1, x_2, t) = v_c(x_1, x_2) + a_p cos(2\pi t)\, v_c(x_1 - 0.25, x_2 - 0.25), \tag{22}$$

with $v_c(x, y)$ being the well-understood cellular velocity field shown in Fig. 7(a) and described by:

$$v_c(x_1, x_2) = \begin{cases} -sin(2\pi x_1)cos(2\pi x_2), \\ cos(2\pi x_1)sin(2\pi x_2). \end{cases} \tag{23}$$

For this velocity field, we impose periodic conditions on the boundary of our 2D domain $[0, 1] \times [0, 1]$.

The standard way to visualize such systems in the non-diffusive case is the Poincaré map method. The Poincaré map for the time-periodically perturbed cellular velocity field is shown in Fig. 7(b). This Poincaré map shows the well-understood mixture of chaotic behaviour, as indicated by orbits filling an area of the phase space, and Kolmogorov–Arnold–Moser (KAM) orbits, surrounding elliptic fixed points and elliptic periodic orbits.

To study the effect of small diffusion, we solve, using our spatial filter and backward time approach, the following 2D Fokker–Planck equation:

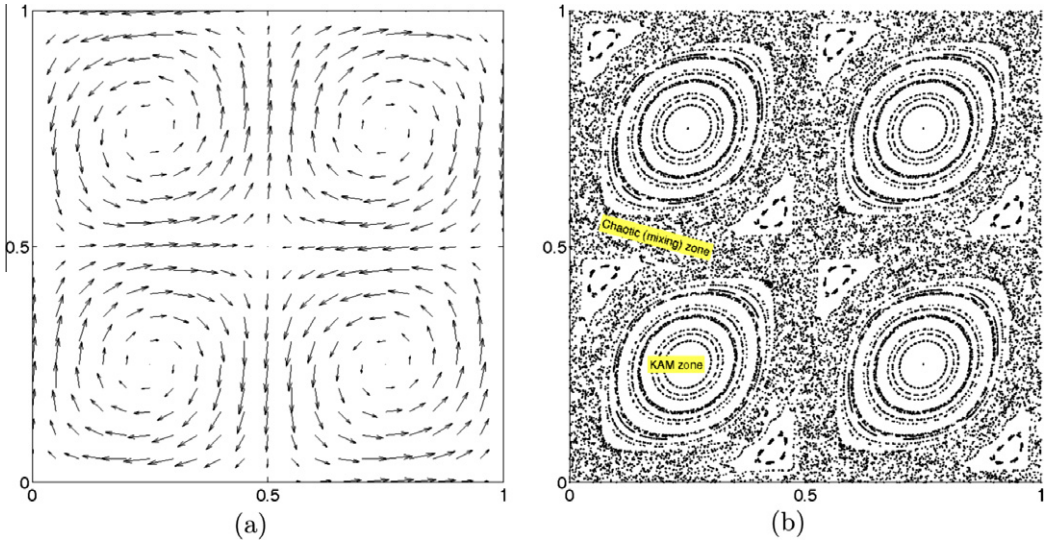


Fig. 7. (a) A cellular, divergence-free velocity field described in Eq. (23), (b) Poincaré map for the time-periodic, divergence free perturbation of the velocity field shown in (a) by a vector field described in Eq. (22) with a perturbation $a_p = 0.1$.

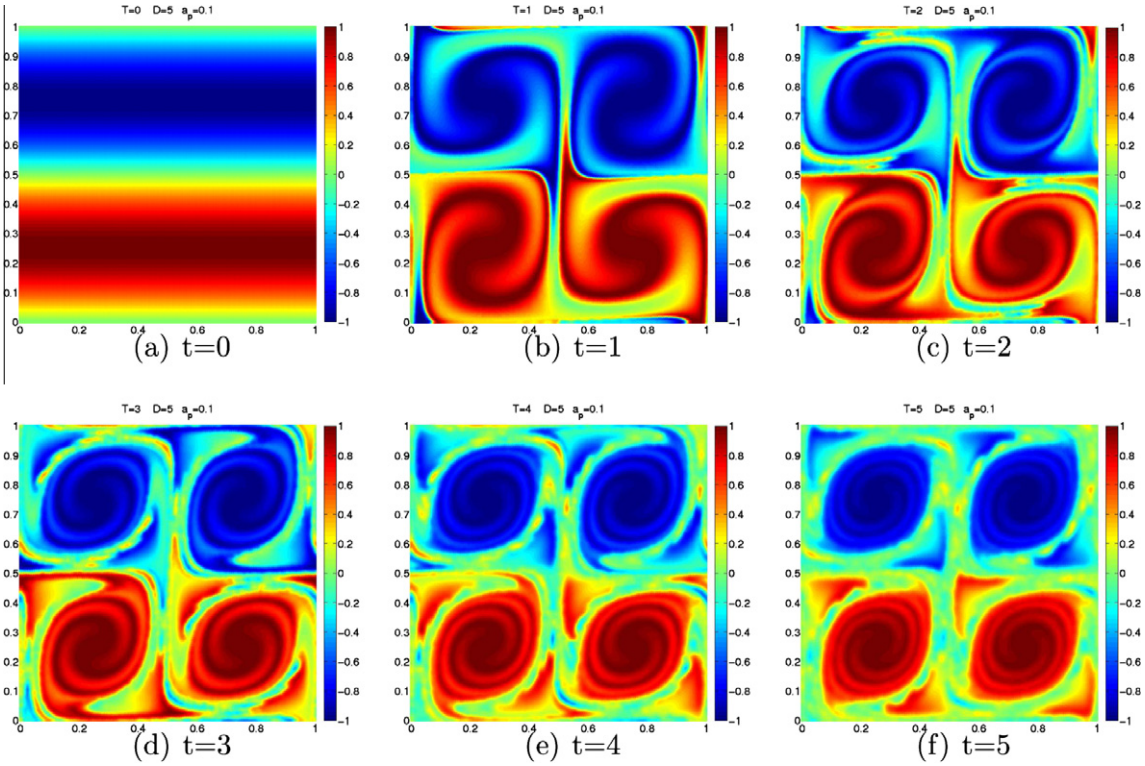


Fig. 8. Snapshot at various times of the density field advected by the velocity in (22) with a perturbation $a_p = 0.1$, solution of (24) with a coefficient of diffusion $D = 10^{-5}$ with a smooth initial condition $u(x_1, x_2, t = 0) = sin(2\pi x_2)$.

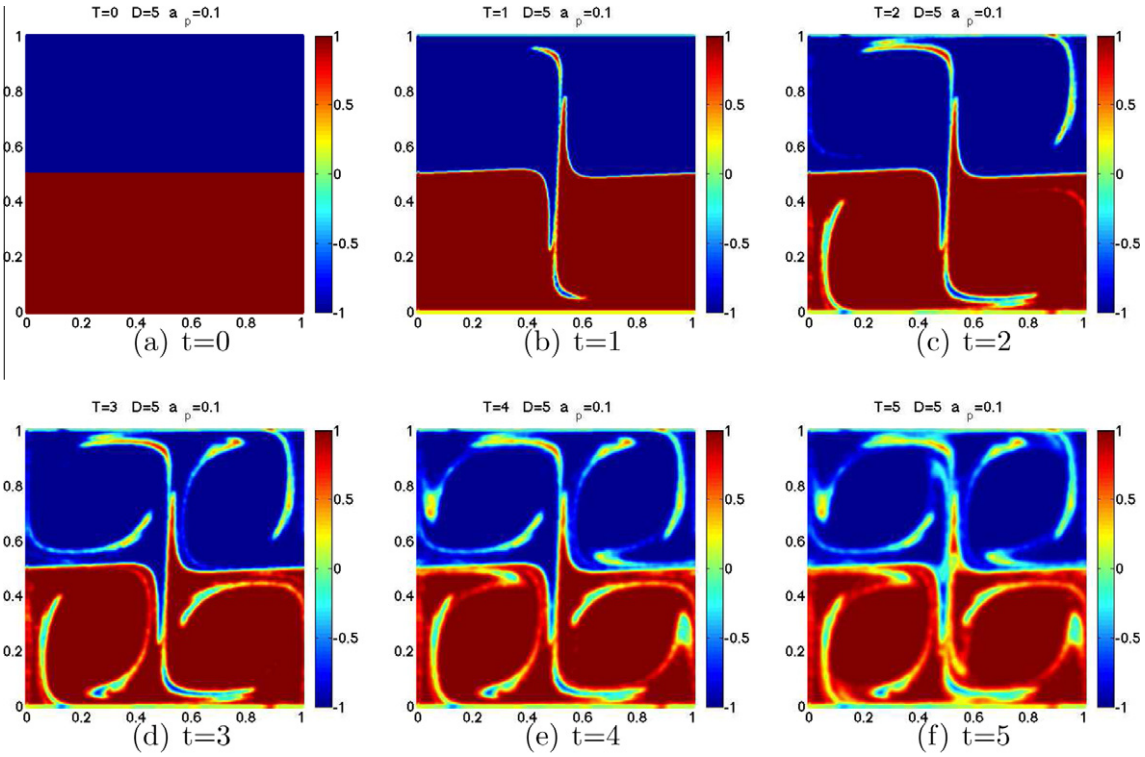


Fig. 9. Snapshot at various times of the density field advected by the velocity in (22) with a perturbation $a_p = 0.1$, solution of (24) with a coefficient of diffusion $D = 10^{-5}$ with a step initial condition $u(x_1, x_2, t = 0) = -1$, if $y > 0.5$ and $u(x_1, x_2, t = 0) = 1$, if $y \leqslant 0.5$.

$$\frac{\partial u}{\partial t}+\sum_{i=1}^{2} v_i(x_1,x_2,t)\frac{\partial u}{\partial x_i}=D\sum_{i,j=1}^{2}\frac{\partial^2 u}{\partial x_i \partial x_j}. \tag{24}$$

For non smooth initial conditions, the method presented here works as long as the solution for $t>0$ belongs to $W^{2,p}\ \forall t>0$. The initial density field is, in our first 2D example, chosen to be $u(x_1,x_2,t=0)=sin(2\pi x_2)$, a smooth initial condition. Then we use the same results of the SDE system to get the solution $u(x,t)$ for $t>0$ in the case of an initial step function with $u(x_1,x_2,t=0)=-1$, if $y>0.5$ and $u(x_1,x_2,t=0)=1$, if $y\leqslant 0.5$.

The results of the numerical simulations are shown in Figs. 8 and 9 for a perturbation $a_p=0.1$ and a coefficient of diffusion $D=10^{-5}$. The simulations are done with a uniform 501×501 grid in the spatial domain, a time step $dt=10^{-4}$ and using a Gaussian spatial filter. Note that $\nabla.v_c=0$, i.e. the velocity field is conservative. As t increases, the density gets slowly more and more uniform. The simulations also show that for slow diffusion, the mixing mechanism is similar to the one without diffusion. Indeed, the zones with best mixing, zones where $c\approx 0$ (see in Fig. 9(f) and 8(f)), are similar to the chaotic mixing zones of the Poincaré map while the zones with low mixing correspond the the KAM zone shown by the Poincaré map.

5. Computational efficiency

In this section, we study the computational complexity of the presented Backward Spatial Filtering method compared to the traditional Monte Carlo method. This calculation does not take into account the computational cost related to solving the SDE system or the influence of the choice of the method used to solve the SDE sytem (multilevel, stratified sampling, etc.) as it would affect the traditional Monte Carlo method and the Spatial Filtering method similarly. In Table 2, we present results from our simulations for the computational effort needed to reach a L^2 accuracy $a=0.05$ for our test 1D example with $v=-sin(\pi x)$ and $D=0.2$. The results are given for the traditional Monte Carlo method, for the Spatial Filtering method and the combination of the two, using different sizes of grid.

Suppose we have a grid with L points with a fixed grid width Δx. Let N be the number of independent simulation launched from each point on the grid such that the Monte Carlo averaging method results in accuracy a for the values of $u(x,t)$ on the grid. The total number of time the SDE system needs to be solved to reach an accuracy a is $L\times N$. As the Monte Carlo averaging error is given by $a=\mathcal{O}\left(1/\sqrt{N}\right)$, then N is of the order of a^{-2}. Hence the computational effort for the traditional Monte Carlo method is of order La^{-2}. For our test example, $N=2000$ independent simulations are needed at each point of the grid to reach a L^2 accuracy $a=0.05$. This leads to a computational effort on a grid of $L=10$ points of $LN=20,000$ and on a grid of $L=26$ points of $LN=52000$ and on a grid of $L=400$ points of $LN=800,000$ (see Table 2).

Let us now determine the computational effort to reach accuracy a for the Backward Spatial Filtering method presented in this paper. Let the filter width be ε which corresponds to $(2M+1)^n$ grid points used for the spatial filtering. Then the accuracy a needs to be equal to $\varepsilon^2+\frac{1}{\sqrt{(2M+1)^n}}$. First, we choose ε that leads to the desired accuracy a for the spatial filtering. Then let's find the spatial grid width Δx and hence L, the number of grid points, needed to reach the accuracy a. We need

$$a \text{ of the order of } \varepsilon^2=\Delta x^2M^2 \text{ and } M^{-n/2},$$

$$\begin{cases}\Delta x^2M^2=M^{-n/2},\\ M=\Delta x^{-\frac{4}{n+4}},\\ a=\varepsilon^2+M^{-n/2}=\Delta x^{\left(2-\frac{8}{n+4}\right)}+\Delta x^{\left(-\frac{n-4}{2n+4}\right)},\\ a=2\Delta x^{\frac{2n}{n+4}},\\ \Delta x=\left(\frac{a}{2}\right)^{\frac{n+4}{2n}}.\end{cases}$$

Table 2
Computational effort to reach a L^2 accuracy of $a=0.05$, for different sizes of grid for our 1D test example (16)–(17) with $D=0.2$ on a grid of L points. At each point of the grid, N simulations are launched. A Gaussian filter with $\mu_s=0.03$ is used and $2M+1$ is the number of neighbouring points used in the filter. The total computational effort is given by the value of LN.

L	$2M+1$	N	$(2M+1)N$	LN
10	1	2000	2000	20,000
26	1	2000	2000	52,000
400	1	2000	2000	800,000
400	37	110	4070	44,000
4000	361	15	5415	60,000
8000	721	6	4326	48,000
10,000	901	4	3604	40,000
16,000	1441	3	4323	48,000
20,000	1801	2	3601	40,000
40,000	3601	1	3601	40,000

We have Δx of order L^{-n}. Thus

L scales as $a^{-\frac{n+4}{2n^2}}$.

The total computational effort of the Backward Spatial Filtering method is equal to the number of points on the grid L and hence scales as $a^{-\frac{n+4}{2n^2}}$. For $n = 1$, we need $L = K_a a^{-2.5}$ with $K_a > 0$. For our test 1D example, we need a grid of $L = 40000$ points to reach a L^2 accuracy of $a = 0.05$ and hence the computational effort using the spatial filter method is $LN = 40{,}000$ (see Table 2). For this example, the constant can be estimated to $K_a \approx 22$. As discussed above, the computational effort for the standard Monte Carlo method scales as a^{-2}. Indeed the number of launches at each grid point is estimated by $K_m a^{-2}$ where $K_m \approx 5$ for our test example. If we are interested in a solution on a sparse grid then the Monte Carlo is more efficient. If we are interested in a fine grid, the spatial filter method performs better. More precisely, for our test example, the computational effort for the filtering method is less than the one for the Monte Carlo method when $K_a a^{-2.5} \leqslant L K_m a^{-2}$. Hence for an accuracy $a = 0.05$, this means the spatial averaging is more computationally efficient if the number of grid points $L \geqslant \frac{K_a}{K_m} a^{-0.5} = 20$. The Monte Carlo method on a grid of 400 points has a computational effort about 20 times higher than the Spatial Filtering method for a grid of 40000 points (see Table 2). For any $n > 1$, the computational effort increases much slower for the Spatial Filtering method, $L \sim a^{-p}$ with $p = -\frac{n+4}{2n^2} < 1$, than for the Monte Carlo method. The minimum number of grid points that leads to a smaller computational effort for the spatial filter method scales as $a^{\frac{4n^2-n-4}{2n^2}} \underset{n\to+\infty}{\to} a^2$. Thus, the Spatial Filtering method is significantly more computationally efficient than the Monte Carlo method to give the solution on a large grid of points and allows better accuracy with very little additional computational effort.

Now let us estimate the number of independent simulations, N, launched from each point on the grid to reach accuracy a when the grid size Δx is fixed using the combined Monte Carlo and Spatial Filtering method.

$$\begin{cases} a = \varepsilon^2 + N^{-1/2} M^{-n/2}, \\ a/2 = \Delta x^2 M^2 \text{ hence } M = a^{1/2} \Delta x^{-1} / \sqrt{2}, \\ a/2 = N^{-1/2} M^{-n/2} \text{ hence } N = 2^{\frac{n-4}{2}} a^{-\frac{n+4}{2}} \Delta x^n. \end{cases}$$

If using Monte Carlo averaging alone, we need $N \sim a^{-2}$ runs to reach a certain accuracy. By combining spatial filtering on M points and Monte Carlo averaging using $N \sim 2^{\frac{n-4}{2}} a^{-\frac{n+4}{2}} \Delta x^n$ runs, we can reach the same accuracy by using $N(2M+1)^n$ estimates of the value at each points on the grid. In 1D, the number of runs N scales as Δx which can be seen in Table 2.

The study of the computational effort presented here shows that in many cases it is advantageous to use spatial averaging to calculate solutions to reduce computational time.

6. Conclusion

This Backward Time Probabilistic method with spatial averaging permits to simulate parabolic equation with high Péclet number on a large grid using less computational time than the usual Monte Carlo methods. In fact when $D = 0$, then $M = 1$, this method reduces to backward tracking and no filtering is needed. Thus for very small diffusivity, the method is expected to perform extremely well against classical Monte Carlo. Moreover, backward tracking does not suffer from lack of coverage issues appearing in forward tracking methods. The method provides a computational advantage when solutions of the same parabolic equation is needed for various initial conditions since the SDE systems only need to be solved once, or when an accurate solution is needed on a multidimensional large regular grid. This type of method is also very effective for solving a large number of coupled linear parabolic PDEs since the number of paths used in the solution does not change with the number of PDEs [40].

Acknowledgements

This work has been supported in part by Office of Naval Research through grants N00014 - 07 - 1 - 0587 and N000141010611.

References

[1] T. Arbogast, C.S. Huang, A fully mass and volume conserving implementation of a characteristic method for transport problems, SIAM J. Sci. Comput. 28 (6) (2006) 2001–2022.
[2] J.W. Barrett, K.W. Morton, Approximate symmetrization and Petrov–Galerkin methods for diffusion–convection problems, Comput. Methods Appl. Mech. Eng. 45 (1–3) (1984) 97–122.
[3] W.M. Briggs, V.E. Henson, S.F. McCormick, A Multigrid Tutorial, second ed., SIAM, 2000.
[4] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 32 (1–3) (1982) 199–259.
[5] K. Burrage, P.M. Burrage, Numerical methods for strong solutions of stochastic differential equations: an overview, Proc. R. Soc. A 460 (2004) 373–402.
[6] M.A. Celia, T.F. Russell, I. Herrera, R.E. Ewing, A Eulerian–Lagrangian localized adjoint method for the advection–diffusion equation, Adv. Water Resour. 13 (4) (1990) 187–206.
[7] S. Chandrasekhar, Stochastic problems in physics and astronomy, Rev. Mod. Phys. 15 (1) (1943) 1–89.

[8] P.J. Davis, P. Rabinowitz, Methods of Numerical Integration, Academic Press, Orlando, FL, 1984.
[9] E.B. Dynkin, Markov Processes, Springer, Berlin, 1965 (Engl. transl. from Russian 1963).
[10] T.K. Flesch, J.D. Wilson, E. Yee, Backward-time lagrangian stochastic dispersion models and their application to estimate gaseous emissions, J. Appl. Meteorol. 34 (6) (1995) 1320–1332.
[11] A.C. Galeao, E.G. Dutra Do Carmo, A consistent approximate upwind Petrov–Galerkin method for convection-dominated problems, Comput. Methods Appl. Mech. Eng. 68 (1) (1988) 83–95.
[12] G.B. Giles, Multilevel Monte Carlo path simulation, Oper. Res. 256 (2008) 981–986.
[13] P. Glasserman, Monte Carlo Methods in Financial Engineering, Springer-Verlag, New York, 2004.
[14] C. Graham, T. Kurtz, S. Meleard, P. Protter, M. Pulvirenti, D. Talay, Probabilistic numerical methods for partial differential equations: elements of analysis, Probabilistic Models for Nonlinear Partial Differential Equations, Lecture Notes in Mathematics, vol. 1627, Berlin/Heidelberg, 1996.
[15] R.W. Healy, T.F. Russell, A finite-volume Eulerian–Lagrangian localized adjoint method for solution of the advection–dispersion equation, Water Resour. Res. 29 (7) (1993) 2399–2413.
[16] J.R. Howell, The Monte Carlo method in radiative heat transfer, J. Heat Transfer 120 (3) (1998) 547–560.
[17] A.M. Ilyin, A.S. Kalashnikov, O.A. Oleynik, Linear second-order partial differential equations of the parabolic type, J. Math. Sci. 108 (108) (2002) 435–542.
[18] P.E. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, Springer-Verlag, 1992.
[19] V.P. Maslov, V.G. Danilov, K.A. Volosov, Mathematical Modelling of Heat and Mass Transfer Processes, Nauka, 1987.
[20] G. Mathew, I. Mezić, S. Grivopoulos, U. Vaidya, L. Petzold, Optimal control of mixing in stokes fluid flows, J. Fluid Mech. 508 (2007) 261–281.
[21] G.N. Milstein, On the probability-theoretic solution of linear systems of elliptic and parabolic equations, Theory Probab. Appl. 23 (4) (1979) 820–824.
[22] M.F. Modest, Backward Monte Carlo simulations in radiative heat transfer, J. Heat Transfer 125 (1) (2003) 57–62.
[23] R.C. Owen, R.E. Honrath, Technical note: a new method for the Lagrangian tracking of pollution plumes from source to receptor using gridded model output, Atm. Chem. Phys. 9 (7) (2009) 2577–2595.
[24] E. Pardoux, D. Talay, Discretization and simulation of stochastic differential equations, Acta Appl. Math. 3 (1985) 23–47.
[25] D. Pollard, Convergence of Stochastic Processes, Springer-Verlag, New York, 1984.
[26] S. Roberts, A particle method for a scalar advection diffusion equation, Math. Comput. Simul. 32 (1–2) (1990) 155–160.
[27] A. Rossler, Runge–Kutta methods for Stratonovich stochastic differential equation systems with commutative noise, J. Comput. Appl. Math. 164–165 (2004) 613–627.
[28] J. Salles, J.F. Thovert, R. Delannay, L. Prevors, J.L. Auriault, P.M. Adler, Taylor dispersion in porous media determination of the dispersion tensor, Phys. Fluids A 5 (10) (1993) 2348–2376.
[29] A. Savitzky, M.J.E. Golay, Smoothing and differentiation of data by simplified least squares procedures, Anal. Chem. 36 (1964) 1627–1639.
[30] A.P.S. Selvadurai, Partial Differential Equations in Mechanics, vols. 1, 2, Springer, Berlin, 2000.
[31] S. Singh, S. Raha, Five-stage Milstein methods for SDEs, Int. J. Comput. Math. 89 (6) (2012) 760–779.
[32] L.J. Song, Y.J. Wu, A modified Crank–Nicolson scheme with incremental unknowns for convection dominated diffusion equations, Appl. Math. Comput. 215 (9) (2010) 3293–3301.
[33] A. Speight, Multigrid Techniques in Economics, Operat. Res. 58 (2) (2010) 236.
[34] D. Spivakovskaya, A.W. Heemink, E. Deleersnijder, The backward ito method for the lagrangian simulation of transport processes with large space variations of the diffusivity, Ocean Science Disc. 4 (4) (2007) 623–652.
[35] D. Stroock, Partial Differential Equations for Probabilists, Cambridge Stud. Adv. Math., Cambridge University Press, Cambridge, UK, 2008.
[36] P. Szymczak, A.J.C. Ladd, Boundary conditions for stochastic solutions of the convection–diffusion equation, Phys. Rev. E 68 (3) (2003) 036704.
[37] D. Talay, Probabilistic Methods in Applied Physics, Simulation of Stochastic Differential Systems, Springer, Berlin, Heidelberg, 1995. pp. 54–96.
[38] P. Wapperom, R. Keunings, V. Legat, The backward-tracking Lagrangian particle method for transient viscoelastic flows, J. Non-Newtonian Fluid Mech. 91 (2–3) (2000) 273–295.
[39] D. Xiu, Fast numerical methods for stochastic computations: a review, Commun. in Comput. Phys. (2009).
[40] D.L. Valentine, I. Mezić, S. Maćesić, N. Crnjarić-Zic, S. Ivić, P.J. Hogan, Patrick, V.A. Fonoberov, S. Loire, Dynamic autoinoculation and the microbial ecology of a deep water hydrocarbon irruption, PNAS, 2012.
[41] C. Zheng, G.D. Bennett, Applied Contaminant Transport Modeling, Wiley, 2002.