Applied Numerical Mathematics 93 (2015) 123–139

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

Particle methods for PDEs arising in financial modeling

Shumo Cui [a], Alexander Kurganov [a,*], Alexei Medovikov [b]

[a] Mathematics Department, Tulane University, New Orleans, LA 70118, USA
[b] Susquehanna International Group, 401 City Avenue, Bala Cynwyd, PA 19131, USA

ARTICLE INFO

Article history:
Available online 13 April 2014

Keywords:
Convection–diffusion equations
Anisotropic diffusion
PDE-based bond pricing models
Deterministic and stochastic particle methods
Monte-Carlo method

ABSTRACT

We numerically study convection–diffusion equations arising in financial modeling. We focus on the convection-dominated cases, in which the diffusion coefficients are relatively small. Both finite-difference and Monte-Carlo methods which are widely used in the problems of this kind might be inefficient due to severe restrictions on the meshsize and the number of realizations needed to achieve high resolution.

We propose an alternative approach based on particle methods which have extremely low numerical diffusion and thus do not have the aforementioned restrictions. Our approach is based on the operator splitting: The hyperbolic steps are made using the method of characteristics, while the parabolic steps are performed using either a special discretization of the integral representation of the solution (which leads to a deterministic particle method) or a stochastic random walk approach.

We apply the designed particle methods to a variety of test problems and the numerical results indicate high accuracy, efficiency and robustness of both the deterministic and stochastic methods. In addition, our numerical experiments clearly demonstrate that the deterministic particle method outperforms its stochastic counterpart.

1. Introduction

In this paper, we are interested in development of highly accurate and efficient numerical method for multidimensional convection–diffusion equations with strictly anisotropic diffusion acting in just one spatial direction. Such equations arise, for instance, in bond pricing models. We will consider a particular example of the following linear two-dimensional (2-D) convection–diffusion equation:

$$u_t + \kappa\big[(\hat{x} - x)u\big]_x + xu_y = \frac{\sigma^2}{2}u_{xx}, \tag{1.1}$$

subject to the singular initial condition

$$u(x, y, 0) = \delta(x - r_0, y), \tag{1.2}$$

where δ stands for the Dirac *delta*-function and $\kappa, \hat{x}, \sigma$ and r_0 are positive constants. In Appendix A, we show how the initial value problem (IVP) (1.1), (1.2) can be derived from the Vasicek bond pricing model [31].

* Corresponding author.
E-mail addresses: scui2@tulane.edu (S. Cui), kurganov@math.tulane.edu (A. Kurganov), Alexei.Medovikov@sig.com (A. Medovikov).
URL: http://www.math.tulane.edu/~kurganov (A. Kurganov).

http://dx.doi.org/10.1016/j.apnum.2014.04.005
0168-9274/

Designing an efficient and highly accurate numerical method for the IVP (1.1), (1.2) is a challenging task due to the following three reasons:

- The convection–diffusion equation (1.1) is typically convection-dominated since in many practical application σ is very small.
- The initial datum (1.2) is a singular δ-function.
- The diffusion in (1.1) is strictly anisotropic, namely, it only acts in the x-direction.

We note that strictly anisotropic (singular) diffusion also arises in other financial models such as, for example, Asian option in local volatility model [10] and option prices in the SABR stochastic volatility model [14].

The most popular numerical methods for convection–diffusion equations are finite-difference methods, see, e.g., [18]. However, when a finite-difference method is applied to (1.1), one has to deal with a very small diffusion in the x-direction and the lack of diffusion in the y-direction. For example, a symmetric centered difference approximation of the convective part of (1.1) will not be stable even when the mesh size is very small. Therefore one has to develop an upwind scheme for (1.1), which may be stable but might be too diffusive (and thus inefficient) to achieve the desired resolution on a reasonably coarse grid. Therefore, low-diffusion particle methods seem to be a good alternative to finite-difference schemes.

Particle methods were original developed for hyperbolic transport equation [9,16,15,17,22,27] and thus they can be directly applied to the inviscid version of (1.1). There are several ways to extend particle methods to convection–diffusion equations [5,7,8,13,25,26]. However, not all of these methods would apply to the case of a strictly anisotropic diffusion as in (1.1). The random walk approach from [5,13] would apply and as we demonstrate in Appendix B, it reduces the resulting stochastic particle method to the Monte-Carlo method for the corresponding stochastic differential equations (SDEs) discussed in Appendix A.

In this paper, we propose a deterministic alternative to the random walk approach. The proposed method is based on the fast explicit operator splitting method [3], according to which the 2-D convection equation is solved using the method of characteristics, while the one-dimensional (1-D) heat equation is solved using a special discretization of the integral representation of its exact solution.

The paper is organized as follows. First, a brief overview of the particle and splitting methods is given in Section 2. We then introduce the new deterministic particle method in Section 3. In Appendix A, we discuss different bond pricing models and derive the IVP (1.1), (1.2). Finally, the numerical results are presented in Section 4.

2. Background

In this section, we give a very brief overview of the particle method for transport equation and of the splitting methods.

2.1. Particle method for transport equations

Consider the 2-D linear transport equation

$$w_t + \big[a(x,y)w\big]_x + \big[b(x,y)w\big]_y = 0 \tag{2.1}$$

with variable coefficients $a(x,y)$ and $b(x,y)$. The main idea of particle methods is to seek solutions in the form of the linear combination of δ-functions,

$$w^N(x,y,t) = \sum_{i=1}^{N} \alpha_i(t)\delta\big(x - x_i(t), y - y_i(t)\big), \tag{2.2}$$

where $(x_i(t), y_i(t))$ is the position of the ith particle at time t and $\alpha_i(t)$ is its weight. We would like to emphasize that the introduced particles carry a certain amount of w, but they are mathematical objects rather than particles of a certain material.

Plugging (2.2) into the weak formulation of (2.1) results in the following system of ODEs (see, e.g., [9,16,15,17,22]):

$$\begin{cases} \dfrac{dx_i(t)}{dt} = a\big(x_i(t), y_i(t), t\big), \\ \dfrac{dy_i(t)}{dt} = b\big(x_i(t), y_i(t), t\big), \\ \dfrac{d\alpha_i(t)}{dt} = 0, \end{cases} \tag{2.3}$$

which describes the dynamics of the ith particle for $i = 1, \ldots, N$. It follows from the third equation in (2.3) that the weights remain constant in time ($\alpha_i(t) = \alpha_i(0)$). The evolution of the particle positions is typically captured using a numerical ODE solver.

For a general initial datum $w(x, y, 0)$, the implementation of the particle method will require an approximation of the initial datum by a linear combination of δ-functions. This can be done, for instance, as follows. We divide the computational domain Ω into a set of N nonoverlapping subdomains Ω_i: $\bigcup_{i=1}^{N} \Omega_i = \Omega$, $\Omega_i \cap \Omega_l = \emptyset$, $\forall i \neq l$. We then set the location of the ith particle, $(x_i(0), y_i(0))$, to be the center of mass of Ω_i and "place" the entire amount of $w(x, y, 0)$ present in Ω_i into the ith particle so that its initial weight is

$$\alpha_i(0) := \int_{\Omega_i} w(x, y, 0)\, dxdy \approx |\Omega_i|\, w\big(x_i(0), y_i(0), 0\big). \tag{2.4}$$

In (2.4), we have used the second-order midpoint rule to approximate the integral over Ω_i. A more accurate (special) quadrature can be used if the function $w(x, y, 0)$ has singularities or is of a highly oscillatory nature.

2.2. Operator splitting methods

Consider a PDE of the following form:

$$w_t = \mathcal{L}_1 w + \mathcal{L}_2 w, \tag{2.5}$$

where $w = w(\cdot, t)$ and $\mathcal{L}_1$ and $\mathcal{L}_2$ are (linear) differential operators. We then split Eq. (2.5) into the two equations:

$$w_t = \mathcal{L}_1 w$$

and

$$w_t = \mathcal{L}_2 w,$$

and denote their solution operators by $\mathcal{S}_{\mathcal{L}_1}$ and $\mathcal{S}_{\mathcal{L}_2}$, respectively. A first-order splitting approximation [30] of the solution of the original equation (2.5) can be then obtained either by

$$w_{12}^{(1)}(\cdot, t + \Delta t) = \mathcal{S}_{\mathcal{L}_1}(\Delta t)\mathcal{S}_{\mathcal{L}_2}(\Delta t) w(\cdot, t) \tag{2.6}$$

or by

$$w_{21}^{(1)}(\cdot, t + \Delta t) = \mathcal{S}_{\mathcal{L}_2}(\Delta t)\mathcal{S}_{\mathcal{L}_1}(\Delta t) w(\cdot, t). \tag{2.7}$$

A second-order accurate approximations are obtained using the second-order Strang splitting [29], which results in either

$$w_{121}^{(2)}(\cdot, t + \Delta t) = \mathcal{S}_{\mathcal{L}_1}(\Delta t/2)\mathcal{S}_{\mathcal{L}_2}(\Delta t)\mathcal{S}_{\mathcal{L}_1}(\Delta t/2) w(\cdot, t) \tag{2.8}$$

or

$$w_{212}^{(2)}(\cdot, t + \Delta t) = \mathcal{S}_{\mathcal{L}_2}(\Delta t/2)\mathcal{S}_{\mathcal{L}_1}(\Delta t)\mathcal{S}_{\mathcal{L}_2}(\Delta t/2) w(\cdot, t). \tag{2.9}$$

As it has been demonstrated in [20], the order of accuracy can be increased by taking a linear combination of (2.6)–(2.9), and the third-order method developed in [20] reads

$$\begin{aligned} w^{(3)}(\cdot, t + \Delta t) = {} & \frac{2}{3}\big(w_{121}^{(2)}(\cdot, t + \Delta t) + w_{212}^{(2)}(\cdot, t + \Delta t)\big) \\ & - \frac{1}{6}\big(w_{12}^{(1)}(\cdot, t + \Delta t) + w_{21}^{(1)}(\cdot, t + \Delta t)\big). \end{aligned} \tag{2.10}$$

3. Particle methods for (1.1), (1.2)

In this section, we introduce two particle methods for the IVP (1.1), (1.2).

The convection–diffusion equation (1.1) is split into the hyperbolic,

$$u_t + \kappa\big[(\hat{x} - x)u\big]_x + [xu]_y = 0, \tag{3.1}$$

and parabolic,

$$u_t = \frac{\sigma^2}{2} u_{xx}, \tag{3.2}$$

parts. Our approach is based on the operator splitting described in Section 2.2: The hyperbolic part (3.1) is solved using the particle method described in Section 2.1, while the numerical method for parabolic part (3.2) is based on either a random walk approach (see Section 3.2.1) or a special discretization of the integral representation of the exact solution of the 1-D heat equation (see Section 3.2.2). Here, we denote by $\mathcal{S}_{\mathcal{H}}$ and $\mathcal{S}_{\mathcal{P}}$ the numerical solution operators of the hyperbolic and parabolic parts, respectively.

For simplicity of presentation, in the rest of this section, we will describe the first-order operator splitting only, where we give the details of the solution operators $\mathcal{S}_{\mathcal{H}}$ and $\mathcal{S}_{\mathcal{P}}$.

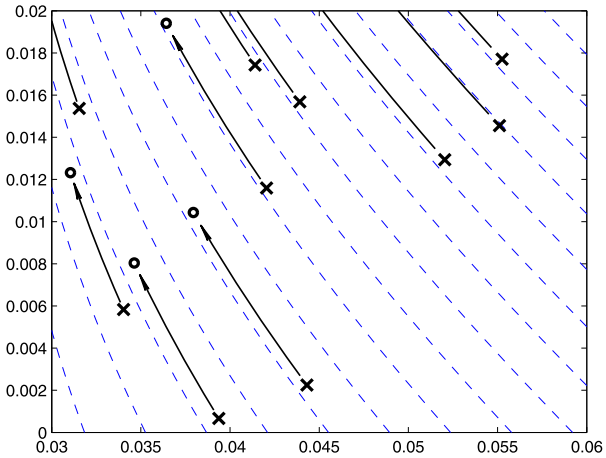

Fig. 3.1. During the hyperbolic step, particles initially positioned at $\{x_i(t), y_i(t)\}$ (marked by $\times$'s) propagate along the characteristic lines to their new positions $\{x_i^*, y_i^*\}$ (marked by $\circ$'s).

3.1. Hyperbolic solution operator $\mathcal{S}_{\mathcal{H}}(\Delta t)$

Eq. (3.1) is the transport equation (2.1) with $a(x, y, t) = \kappa(\hat{x} - x)$ and $b(x, y, t) = x$. Therefore the first two equations of the ODE system (2.3) take the following form:

$$\begin{cases} \dfrac{dx_i(t)}{dt} = \kappa\big(\hat{x} - x_i(t)\big), \\ \dfrac{dy_i(t)}{dt} = x_i(t), \end{cases} \qquad i = 1, \ldots, N. \tag{3.3}$$

Unlike the general system (2.3), the system (3.3) can be easily solved analytically and thus

$$\mathcal{S}_{\mathcal{H}}(\Delta t)\big[\alpha_i \delta\big(x - x_i(t), y - y_i(t)\big)\big] = \alpha_i \delta\big(x - x_i^*, y - y_i^*\big),$$

where

$$\begin{cases} x_i^* = \big(x_i(t) - \hat{x}\big)e^{-\kappa \Delta t} + \hat{x}, \\ y_i^* = y_i(t) + \Delta t \hat{x} - \dfrac{1}{\kappa}\big(x_i(t) - \hat{x}\big)\big(e^{-\kappa \Delta t} - 1\big), \end{cases} \qquad i = 1, \ldots, N. \tag{3.4}$$

The hyperbolic step is schematically shown in Fig. 3.1.

3.2. Parabolic solution operator $\mathcal{S}_{\mathcal{P}}(\Delta t)$

Since the diffusion acts only in the x-direction, none of the methods from [7,8,25,26] applies to Eq. (3.2) studied in the 2-D domain. The random walk approach from [5,13] would apply (see Section 3.2.1), but it reduces to the Monte-Carlo method for (A.1), and thus suffers of all disadvantages of stochastic methods. We propose an alternative deterministic approach described in Section 3.2.2.

3.2.1. Random walk solution operator $\mathcal{S}_{\mathcal{P}}^{\mathrm{R}}(\Delta t)$

According to the random walk approach [5,13], we solve the IVP

$$\begin{cases} u_t = \dfrac{\sigma^2}{2} u_{xx}, \\ u^*(x, t) = \displaystyle\sum_{i=1}^{N} \alpha_i \delta\big(x - x_i^*, y - y_i^*\big), \end{cases} \tag{3.5}$$

by randomly perturbing the locations of each particle. Namely, we obtain

$$\begin{cases} x_i(t + \Delta t) = x_i^* + \xi_i, \\ y_i(t + \Delta t) = y_i^*, \end{cases} \tag{3.6}$$

where $\xi_i \sim N(0, \sigma\sqrt{\Delta t})$ is a normally distributed random number. Thus,

$$\mathcal{S}_{\mathcal{P}}^{\mathrm{R}}(\Delta t)\big[\alpha_i \delta\big(x - x_i^*, y - y_i^*\big)\big] = \alpha_i \delta\big(x - x_i^* - \xi_i, y - y_i^*\big).$$

Since the distribution of ξ_i is symmetric about 0, we also expect $x_i(t+\Delta t)$ to be symmetric about x_i. This can be done by replacing (3.6) with

$$\begin{cases} x_{2k-1}(t+\Delta t) = x^*_{2k-1} + \xi_{2k-1}, \\ y_{2k-1}(t+\Delta t) = y^*_{2k-1}, \end{cases} \qquad \begin{cases} x_{2k}(t+\Delta t) = x^*_{2k} - \xi_{2k-1}, \\ y_{2k}(t+\Delta t) = y^*_{2k}, \end{cases} \tag{3.7}$$

where the x-coordinates of each pair of particles, (x_{2k-1}, y_{2k-1}) and (x_{2k}, y_{2k}), get perturbations of the same magnitude, but with different signs. Such modification results in halving the number of generated random numbers and leads to the exact conservation of the first moment (the definition of the kth moment is given in (3.11) in Section 3.2.2). The latter guarantees that the expectation of $y(T)$, defined in (4.2) in Section 4, is calculated within the machine error as it will be demonstrated in Section 4.

Equipped with $\mathcal{S}_{\mathcal{H}}$ and $\mathcal{S}^{\mathrm{R}}_{\mathcal{P}}$, we can now solve the IVP (1.1), (1.2) using the operator splitting. Recall that the considered initial condition (1.2) consists of just one particle initially located at $(x_1(0), y_1(0)) = (r_0, 0)$. The trajectory of this particle is not deterministic due to the randomness of $\mathcal{S}^{\mathrm{R}}_{\mathcal{P}}$. Therefore, in order to obtain a numerical solution of (1.1), (1.2), one needs to conduct a number of experiments, which would give several particle trajectories. By repeatedly doing such realizations, we will obtain a series of samples whose mean approaches the bond price. As it is typical for stochastic numerical methods, a high accuracy can be achieved only when the number of realization is very large.

As we show in Appendix B, the resulting stochastic particle method is in fact equivalent to the Monte-Carlo method for the corresponding SDEs (A.1) and (A.3).

3.2.2. Deterministic solution operator $\mathcal{S}^{\mathrm{D}}_{\mathcal{P}}(\Delta t)$

Since Eq. (3.2) is linear, its solution satisfies the superposition principle and thus, we only need to show how the parabolic solution operator acts on the ith particle. To this end, we simply solve the IVP

$$\begin{cases} u_t = \dfrac{\sigma^2}{2} u_{xx}, \\ u(x, y, t) = \alpha_i \delta\big(x - x^*_i, y - y^*_i\big), \end{cases} \tag{3.8}$$

exactly with the help of the heat kernel. This results in

$$u(x, y, t+\Delta t) = \alpha_i \delta\big(y - y^*_i\big)\, G\left(x - x^*_i, \frac{\sigma^2}{2}\Delta t\right), \quad G(x,\tau) = \frac{1}{\sqrt{4\pi\tau}} e^{-\frac{x^2}{4\tau}}. \tag{3.9}$$

Now, to complete the parabolic step, we need to obtain a particle approximation of (3.9). Since the parabolic operator $\mathcal{S}_{\mathcal{P}}$ acts in the x-direction only, we approximate (3.9) using $N_{\mathcal{P}}$ new particles, each of which has the same y-coordinate y^*_i. This means that we place the new particles along the line $y = y^*_i$ at $(x^*_i + \zeta_j, y^*_i)$, $j = 1, \ldots, N_{\mathcal{P}}$, so that

$$\mathcal{S}^{\mathrm{D}}_{\mathcal{P}}\big[\alpha_i \delta\big(x - x^*_i, y - y^*_i\big)\big] = \sum_{j=1}^{N_{\mathcal{P}}} \beta_{ij} \delta\big(x - x^*_i - \zeta_j, y - y^*_i\big). \tag{3.10}$$

The weights and locations of the new particles are determined using the conservation of the first K moments. The kth moments of u are defined as

$$M_k(u) = \iint\limits_{\mathbb{R}^2} u(x, y, \cdot) x^k \, dx dy, \quad k = 0, 1, 2, \ldots \tag{3.11}$$

Thus, we need to select β_{ij} and ζ_j such that

$$\begin{aligned} &\alpha_i \iint\limits_{\mathbb{R}^2} \delta\big(y - y^*_i\big)\, G\left(x - x^*_i, \frac{\sigma^2}{2}\Delta t\right) x^k \, dx dy \\ &\quad = \sum_{j=1}^{N_{\mathcal{P}}} \beta_{ij} \iint\limits_{\mathbb{R}^2} \delta\big(x - x^*_i + \zeta_j, y - y^*_i\big) x^k \, dx dy \end{aligned} \tag{3.12}$$

for $k = 0, \ldots, K-1$.

Remark 3.1. We would like to remind the reader that $u(x, y, t)$ is the probability density function (pdf) of the random vector $(x(t), y(t))^T$ as discussed in Appendix A. In order to preserve some key properties of the random variable $x(t)$ (note that $y(t)$ is not affected by the above particle approximation of (3.9)), we have to ensure that the particle approximation of the pdf u satisfies the moment conservation property. For example, conserving the first three moments guarantees that the mean and variance of the random variable $x(t)$ are preserved.

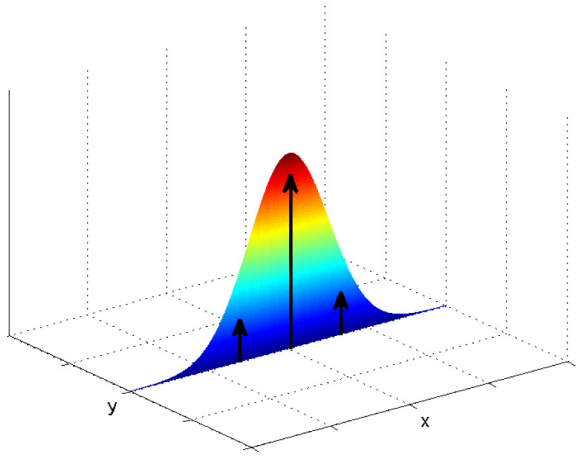

Fig. 3.2. The exact solution (3.9) is replaced by $N_{\mathcal{P}} = 3$ new particles, whose locations are indicated by the three arrows. Notice that the support of the solution (3.9) is the line $y = y_i^*$.

For example, if $K = 4$, β_{ij} and ζ_j have to satisfy the following four conditions:

$$\sum_{j=1}^{N_{\mathcal{P}}} \beta_{ij} = \alpha_i, \quad \sum_{j=1}^{N_{\mathcal{P}}} \beta_{ij}\zeta_j = 0, \quad \sum_{j=1}^{N_{\mathcal{P}}} \beta_{ij}\zeta_j^2 = \alpha_i\sigma^2\Delta t, \quad \sum_{j=1}^{N_{\mathcal{P}}} \beta_{ij}\zeta_j^3 = 0.$$

We thus need at least two new particles to have enough degrees of freedom. To avoid the unnecessary computational burden, we choose the smallest sufficient $N_{\mathcal{P}} = 2$. Therefore, we will have two new particles with $\zeta_1 = -\sigma\sqrt{\Delta t}$, $\zeta_2 = \sigma\sqrt{\Delta t}$ and the corresponding weights $\beta_{i1} = \beta_{i2} = \alpha_i/2$.

Similarly, if first $K = 6$ moments are to be conserved, it will suffice to take $N_{\mathcal{P}} = 3$ new particles with $\zeta_1 = -\sigma\sqrt{3\Delta t}$, $\zeta_2 = 0$, $\zeta_3 = \sigma\sqrt{3\Delta t}$ and the corresponding weights $\beta_{i1} = \alpha_i/6$, $\beta_{i2} = 2\alpha_i/3$ and $\beta_{i3} = \alpha_i/6$. The latter case is illustrated in Fig. 3.2.

After applying the parabolic solution operator $\mathcal{S}_{\mathcal{P}}^{\mathrm{D}}$, each particle is replaced by $N_{\mathcal{P}}$ new particles, so that even though initially we begin with just one particle (1.2), the total number of particles will grow exponentially fast and after n time steps, there will be $N_{\mathcal{P}}^n$ particles, which will make the computational workload intolerably large and the overall method will be impractical. To overcome this difficulty, some particles have to get merged upon completion of each $\mathcal{S}_{\mathcal{P}}^{\mathrm{D}}$ step. The idea of the merging procedure presented in Section 3.2.3 is to replace the particles that coalesce with a smaller number of particles, while preserving (some of) the moments.

3.2.3. Merger procedure $\mathcal{M}$

The main idea of the merger procedure, schematically presented in Fig. 3.3, is to introduce an auxiliary (Cartesian) grid and to replace all of the particles that are located in the same cell with one new particle located at their center of mass. The error introduced by such a replacement can be controlled by varying the size of the auxiliary grid which consists of the cells $\mathcal{C}_{jk} = (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}) \times (y_{k-\frac{1}{2}}, y_{k+\frac{1}{2}})$ of the size $x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}} = \Delta x_j$ by $x_{k+\frac{1}{2}} - x_{k-\frac{1}{2}} = \Delta y_k$.

Let us assume that after the parabolic splitting step, the total number of particle is $N \cdot N_{\mathcal{P}}$ and that N_{jk} of them are located in the auxiliary cell $\mathcal{C}_{jk}$. We then merge these N_{jk} particles into one particle carrying the weight

$$\beta_{jk} = \sum_{i=1}^{N_{jk}} \alpha_i \tag{3.13}$$

and located at the center of mass of the merged particles. Namely, the location of the new particle is (ξ_{jk}, η_{jk}) with

$$\xi_{jk} = \frac{1}{\beta_{jk}} \sum_{i=1}^{N_{jk}} \alpha_i x_i(t+\Delta t), \qquad \eta_{jk} = \frac{1}{\beta_{jk}} \sum_{i=1}^{N_{jk}} \alpha_i y_i(t+\Delta t). \tag{3.14}$$

Notice that the sums in (3.13), (3.14) are taken over all i such that $(x_i(t+\Delta t), y(t+\Delta t)) \in \mathcal{C}_{jk}$.

Finally, the merger procedure can be represented in the following operator form:

$$\mathcal{M}\left[\sum_{i=1}^{N\cdot N_{\mathcal{P}}} \alpha_i \delta\big(x - x_i(t+\Delta t), y - y_i(t+\Delta t)\big)\right] = \sum_{j,k} \beta_{jk}\delta(x - \xi_{jk}, y - \eta_{jk}),$$

where the summation on the right-hand side is taken over all of the auxiliary cells that contain any particles.

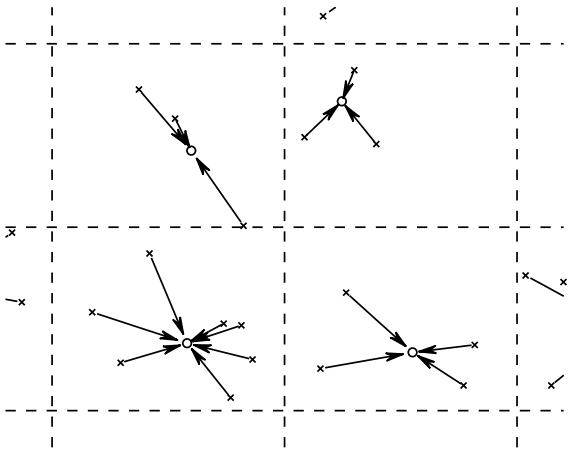

Fig. 3.3. The merger: In each auxiliary cell, the old particles (marked by ×'s) are replaced by the new particle (marked by ∘'s), located at the center of mass of the old particles. Notice the center of the mass does not coincide with the center of the cell.

4. Numerical results

In this section, we apply the designed particle methods to the IVP (1.1), (1.2). For the sake of brevity, we will denote by D1, D2 and D3 the deterministic particle methods obtained using the first-, second- and third-order splitting, respectively. A similar notation (R1, R2 and R3) will be used in the cases when the parabolic step is performed using the random walk. We take the following values of the parameters:

$$\hat{x} = 0.025, \qquad r_0 = 0.09, \qquad \kappa = 2, \qquad \sigma = 0.01 \quad \text{or} \quad 0.3. \tag{4.1}$$

Our main goal is to calculate the following three integral quantities:

$$E(u;T) = \iint_{\mathbb{R}^2} yu(x,y,T)\,dxdy, \tag{4.2}$$

$$V(u;T) = \iint_{\mathbb{R}^2} \big(y - E(u;T)\big)^2 u(x,y,T)\,dxdy, \tag{4.3}$$

$$B(u;T) = \iint_{\mathbb{R}^2} e^{-y} u(x,y,T)\,dxdy. \tag{4.4}$$

The quantities E, V are the expectation and variance of the path integral of the interest rate at time T, and B is the bond price at time T.

For the particle approximation

$$u^N(x,y,t) = \sum_{i=1}^{N} \alpha_i(t)\delta\big(x - x_i(t), y - y_i(t)\big) \tag{4.5}$$

taken at the final time T, Eqs. (4.2)–(4.4) reduce to

$$E\big(u^N;T\big) = \iint_{\mathbb{R}^2} y \sum_{i=1}^{N} \alpha_i(T)\delta\big(x - x_i(T), y - y_i(T)\big)\,dxdy = \sum_{i=1}^{N} \alpha_i(T) y_i(T) w_i(T),$$

$$\begin{aligned} V\big(u^N;T\big) &= \iint_{\mathbb{R}^2} \big(y - E\big(u^N;T\big)\big)^2 \sum_{i=1}^{N} \alpha_i(T)\delta\big(x - x_i(T), y - y_i(T)\big)\,dxdy \\ &= \sum_{i=1}^{N} \alpha_i(T)\big(y_i(T) - E\big(u^N;T\big)\big)^2, \end{aligned}$$

$$B\big(u^N;T\big) = \iint_{\mathbb{R}^2} e^{-y} \sum_{i=1}^{N} \alpha_i(T)\delta\big(x - x_i(T), y - y_i(T)\big)\,dxdy = \sum_{i=1}^{N} \alpha_i(T) e^{-y_i(T)}.$$

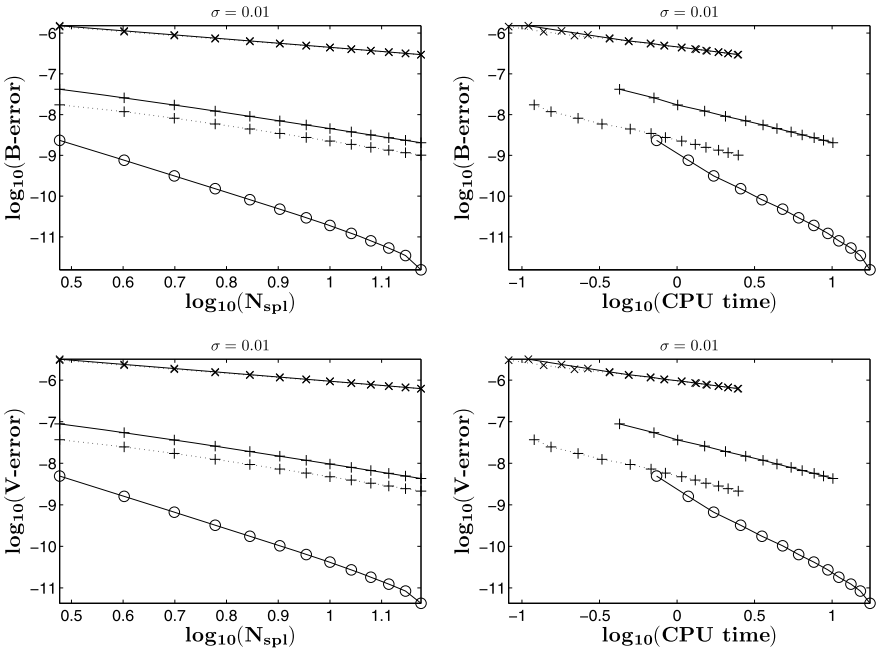

Fig. 4.1. Dependence of the errors $V(u;1) - V(u^N;1)$ and $B(u;1) - B(u^N;1)$ on the number of splitting steps (N_{spl} ranges from 3 to 15) and the CPU time for different splitting algorithms: ∘ represents the third-order algorithm (2.10); + represents the second-order algorithms (2.8) (dot line) and (2.9) (solid line), respectively; × represents the first-order algorithms (2.6) (dot line) and (2.7) (solid line), respectively. In those algorithms, $\mathcal{L}_1$ is the hyperbolic solution operator, while $\mathcal{L}_2$ is the parabolic one. Here, $\sigma = 0.01$.

Since the exact values of E, V and B are available [24], we can calculate the errors and thus to compare performances of different particle methods. The errors in Section 4 are computed using the final time $T = 1$.

Remark 4.1. Note that when a deterministic particle method is used, the number of particles N changes at every time step.

4.1. Comparison of different splitting algorithms

In this section, we denote by N_{spl} the total number of splitting steps. We use a uniform auxiliary grid with $\Delta x_j \equiv \Delta y_k \equiv h$. The number of new particles generated at each deterministic parabolic step (described in Section 3.2.2) is taken to be $N_{\mathcal{P}} = 2$ for $\sigma = 0.01$ and $N_{\mathcal{P}} = 3$ for $\sigma = 0.3$.

It is obvious that the merger procedure introduces errors. To prevent these errors from dominating, we take very small $h = 10^{-5}$ for $\sigma = 0.01$ and $h = 3 \times 10^{-4}$ for $\sigma = 0.3$, which ensure that the merger error is substantially smaller than the splitting error (in Section 4.3 we will show that the merger error is proportional to h^2).

In Figs. 4.1 and 4.2, we plot the errors (in the logarithmic scale) for the deterministic particle methods implemented using five different splitting algorithms. As one can clearly see from the graphs on the left, higher-order splitting outperforms the lower-order ones and the desired order of accuracy is achieved for $\sigma = 0.01$ (Fig. 4.1) and also in the computation of V with $\sigma = 0.3$ (Fig. 4.2).

Notice, however, that the B-error for the D3 method saturates at about $N_{\text{spl}} = 9$ (see the graphs on the upper row in Fig. 4.2). We conjecture that the reason for such an error behavior is related to a small values of $N_{\mathcal{P}} = 3$ and relatively large value of $h = 3 \times 10^{-4}$. If $N_{\mathcal{P}}$ is taken to be substantially larger and h is made much smaller, then the number of particles would increase so dramatically that the method would become inefficient. This sets certain limitations on the applicability of the proposed deterministic particle method. However, we would like to point out that even for the larger value of $\sigma = 0.3$ the deterministic methods clearly outperforms its stochastic counterpart.

The graphs in the right columns in Figs. 4.1 and 4.2 clearly demonstrate that even though the third-order splitting requires computing two first-order and two second-order splitting solutions, it is still much more efficient than the lower-order splitting methods since it achieves the same size of the error using smaller amount of the CPU time. The experimental order of convergence for the D3 method is shown in Table 4.1 for $\sigma = 0.01$. For $\sigma = 0.3$ the V-errors still decrease very fast while the B-errors saturate, as one can see in Table 4.2. Before reaching the saturation, the superconvergence of B-errors can be observed.

Notice that the $D3$ method captures the expectation E within the machine accuracy. This occurs because E is captured by the first moment of particles, which is not affected by the parabolic step as soon as the first moment is conserved. Also,

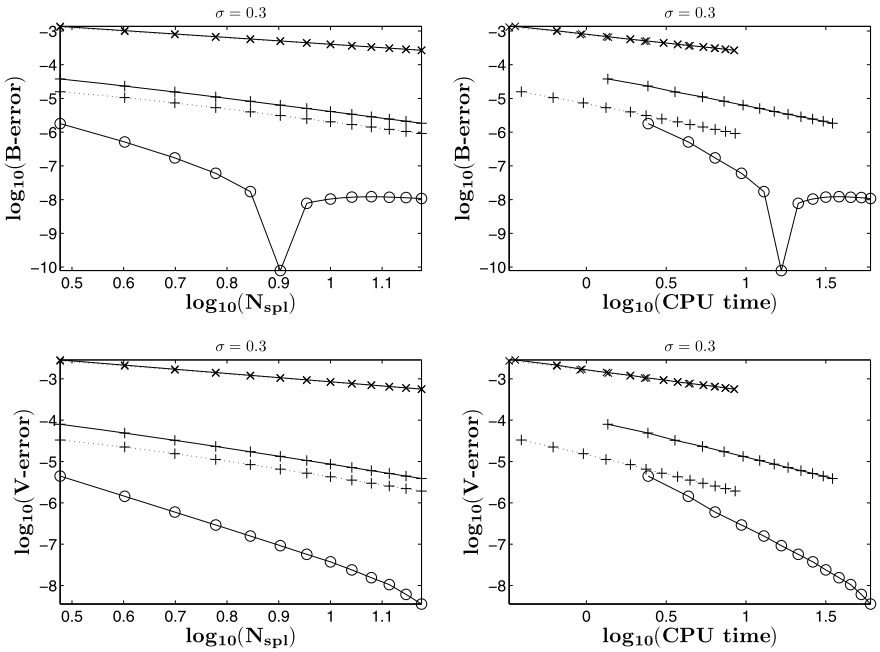

Fig. 4.2. The same as in Fig. 4.1, but for $\sigma = 0.3$.

Table 4.1
Dependence of the errors $E(u;1) - E(u^N;1)$, $V(u;1) - V(u^N;1)$ and $B(u;1) - B(u^N;1)$ on the number of splitting time steps for the D3 method for $\sigma = 0.01$, $N_{\mathcal{P}} = 2$.

$\sigma = 0.01$, $N_{\mathcal{P}} = 2$						
N_{spl}	h	E-error	V-error	rate	B-error	rate
1	1.00E−05	1.39E−17	2.56E−07	–	1.21E−07	–
2	1.00E−05	1.39E−17	2.31E−08	3.47	1.10E−08	3.46
4	1.00E−05	1.39E−17	1.60E−09	3.85	7.58E−10	3.86
8	1.00E−05	6.94E−17	1.03E−10	3.96	4.78E−11	3.99
16	1.00E−05	1.25E−16	1.83E−12	5.81	4.33E−13	6.79

Table 4.2
Dependence of the errors $E(u;1) - E(u^N;1)$, $V(u;1) - V(u^N;1)$ and $B(u;1) - B(u^N;1)$ on the number of splitting time steps for the D3 method for $\sigma = 0.3$, $N_{\mathcal{P}} = 3$.

$\sigma = 0.3$, $N_{\mathcal{P}} = 3$						
N_{spl}	h	E-error	V-error	rate	B-error	rate
1	3.00E−04	6.94E−18	2.30E−04	–	1.07E−04	–
2	3.00E−04	2.08E−17	2.08E−05	3.47	9.21E−06	3.54
4	3.00E−04	1.39E−17	1.44E−06	3.85	5.11E−07	4.17
8	3.00E−04	2.50E−16	9.21E−08	3.97	7.91E−11	12.66
16	3.00E−04	2.03E−15	1.57E−09	5.87	1.02E−08	−7.01

with the first moment conserved in the merger step and exact hyperbolic solution operator, E is computed exactly (up to the machine error) throughout the entire algorithm.

The results for similar experiments with the stochastic particle method confirm that the third-order splitting is advantageous compared to the low-order ones. We therefore will only use the third-order splitting method in the rest of the paper.

4.2. Dependence of the errors on $N_{\mathcal{P}}$

In this section, we compare the numerical results obtained using $N_{\mathcal{P}} = 2$ and $N_{\mathcal{P}} = 3$ (recall that $N_{\mathcal{P}}$ is the number of new particles appearing at the end of the parabolic step). As we have pointed out in Section 3.2.2, bigger $N_{\mathcal{P}}$ allows one to conserve more moments. However, it is not clear whether taking $N_{\mathcal{P}} > 2$ will improve the quality of numerical solution. To verify this, we apply the D3 method with $N_{\mathcal{P}} = 2$ and $N_{\mathcal{P}} = 3$ to the same IVP (1.1), (1.2), (4.1). The obtained results are plotted in Figs. 4.3 and 4.4. As one can see, when $\sigma = 0.01$ the error with $N_{\mathcal{P}} = 3$ remains the same as with $N_{\mathcal{P}} = 2$ (see

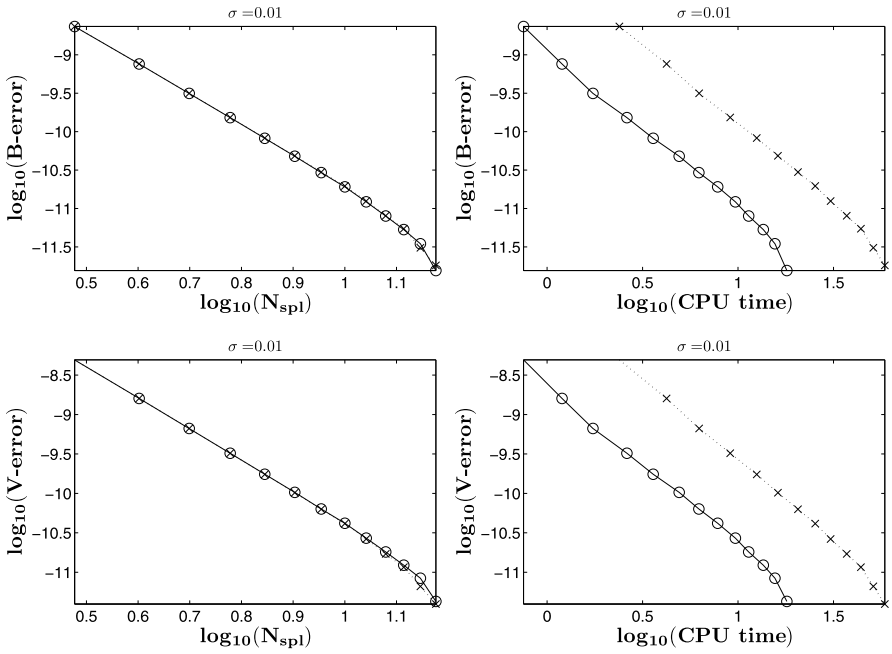

Fig. 4.3. Dependence of the errors $V(u;1)-V(u^N;1)$ and $B(u;1)-B(u^N;1)$ on the number of splitting steps (N_{spl} ranges from 3 to 15) and the CPU time for $N_{\mathcal{P}}=3$ (marked by $\circ$) and $N_{\mathcal{P}}=2$ (marked by $\times$). Here, $\sigma=0.01$.

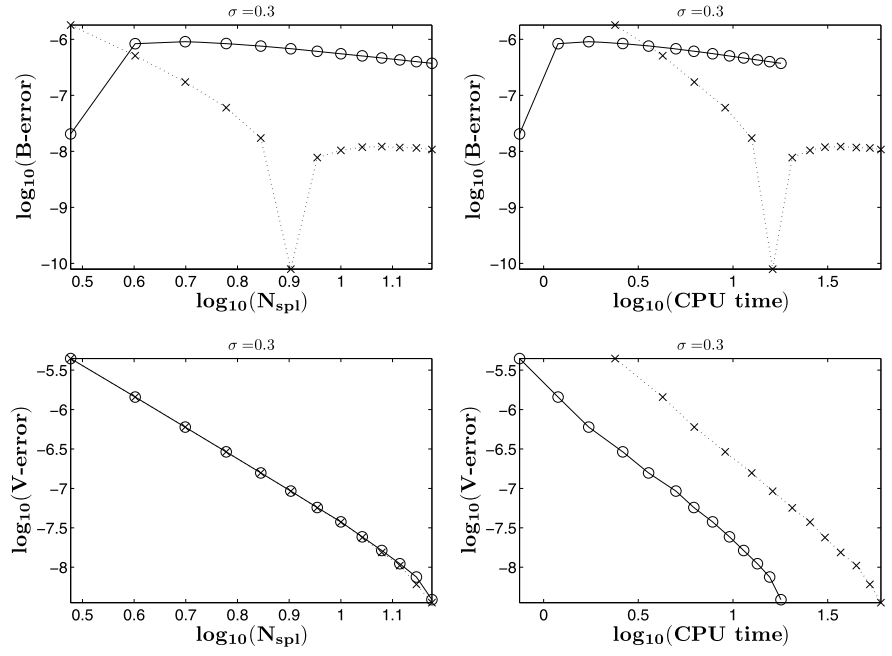

Fig. 4.4. The same as in Fig. 4.3, but for $\sigma=0.3$.

Fig. 4.3), while the overall method becomes less efficient (see the graphs on the right in Fig. 4.3). We will therefore take $N_{\mathcal{P}}=2$ in the rest of the paper when $\sigma=0.01$.

However, when $\sigma=0.3$ the B-errors for $N_{\mathcal{P}}=2$ and $N_{\mathcal{P}}=3$ are different and it is clearly better to take $N_{\mathcal{P}}=3$ (see Fig. 4.4). Our additional numerical experiments (not reported in this paper) clearly indicate that taking $N_{\mathcal{P}}=4$ or larger does not necessarily lead to substantial further improvement in accuracy while significantly affecting the efficiency of the D3 method.

Table 4.3
Dependence of the errors $E(u;1)-E(u^N;1)$, $V(u;1)-V(u^N;1)$ and $B(u;1)-B(u^N;1)$ on the size of auxiliary cells for the D3 method for $\sigma = 0.01$.

$\sigma = 0.01$, $N_{\mathcal{P}} = 2$						
N_{spl}	h	E-error	V-error	rate	B-error	rate
20	2.56E−03	1.39E−17	4.54E−06	–	2.15E−06	–
20	1.28E−03	1.39E−17	4.84E−07	3.23	2.29E−07	3.23
20	6.40E−04	4.16E−17	3.70E−08	3.71	1.75E−08	3.71
20	3.20E−04	2.78E−17	5.49E−09	2.75	2.61E−09	2.75
20	1.60E−04	1.39E−17	1.56E−09	1.82	7.39E−10	1.82
20	8.00E−05	2.78E−17	4.61E−10	1.76	2.19E−10	1.75
20	4.00E−05	2.78E−17	1.10E−10	2.07	5.24E−11	2.06
20	2.00E−05	1.39E−17	1.85E−11	2.57	9.10E−12	2.53
20	1.00E−05	9.71E−17	2.81E−12	2.72	1.68E−12	2.44

Table 4.4
Dependence of the errors $E(u;1)-E(u^N;1)$, $V(u;1)-V(u^N;1)$ and $B(u;1)-B(u^N;1)$ on the size of auxiliary cells for the D3 method for $\sigma = 0.3$.

$\sigma = 0.3$, $N_{\mathcal{P}} = 3$						
N_{spl}	h	E-error	V-error	rate	B-error	rate
20	7.68E−02	6.25E−17	2.43E−03	–	1.14E−03	–
20	3.84E−02	1.46E−16	5.31E−04	2.19	2.49E−04	2.19
20	1.92E−02	3.47E−17	2.83E−05	4.23	1.35E−05	4.21
20	9.60E−03	2.15E−16	7.04E−06	2.01	3.35E−06	2.01
20	4.80E−03	4.02E−16	1.33E−06	2.40	6.39E−07	2.39
20	2.40E−03	6.94E−17	3.80E−07	1.81	1.88E−07	1.77
20	1.20E−03	5.55E−17	1.03E−07	1.88	5.62E−08	1.74
20	6.00E−04	8.60E−16	1.91E−08	2.43	1.61E−08	1.80
20	3.00E−04	1.78E−15	2.84E−09	2.75	8.38E−09	0.94

Remark 4.2. We would like to emphasize that choosing bigger $N_{\mathcal{P}}$ and conserving more moments only improves the accuracy of the numerical solution *locally at the parabolic step*. When $N_{\mathcal{P}} = 3$ the two extra moments, which are conserved in the parabolic step ((3.12) with $k = 5$ and 6), are not conserved in the merger procedure. This explains why the effort of conserving more moments does not necessarily result in a better numerical solution.

4.3. Experimental study of merger errors

In this section, we demonstrate that the merger errors are proportional to the size of auxiliary cells (h^2). To this end, we keep the number of splitting step to be a large constant ($N_{\text{spl}} = 20$) so that the merger error is expected to dominate the splitting one. The obtained results are shown in Tables 4.3 and 4.4, where the expected rates of convergence can be observed for $\sigma = 0.01$ and for computing V with $\sigma = 0.3$. When B is computed with $\sigma = 0.3$, the expected convergence rate is achieved before B-error saturates.

4.4. Comparison of deterministic and stochastic approaches

We now turn to the comparison between the stochastic and deterministic ways of approximating the parabolic solution operator (see Section 3.2.2 and Section 3.2.1). To prevent either merger or splitting error from dominating in the numerical experiments, we keep the balance between h and N_{spl}. Since the splitting error is expected to be of the third-order while the merger one is only second-order, we introduce the parameter

$$C := h^2 (N_{\text{spl}})^3$$

which is being fixed. The errors in the results obtained by the D3 method are shown in Tables 4.5 and 4.6, where one can clearly observe a high experimental order of convergence for $\sigma = 0.01$ and for computing V with $\sigma = 0.3$, while once again when B is computed for $\sigma = 0.3$, the high experimental order is achieved before the B-error saturates when we use $N_{\text{spl}} = 16$ and $h = 3 \times 10^{-4}$.

The D3 and R3 methods are compared in Fig. 4.5, where we plot their errors as functions of the CPU time (in the logarithm scale). As one can see there, the deterministic approach is more efficient. Also, we would like to point out that even though some of the results for the D3 method are affected by the error saturation (see the lower left graph in Fig. 4.5), the D3 method clearly outperforms the R3 one.

Table 4.5
Dependence of the errors $E(u;1)-E(u^N;1)$, $V(u;1)-V(u^N;1)$ and $B(u;1)-B(u^N;1)$ on both N_{spl} and h for the D3 method for $\sigma=0.01$, $N_{\mathcal{P}}=2$ and the fixed parameter C: As N_{spl} is doubled, h is multiplied by factor of $(2\sqrt{2})^{-1}$.

$\sigma=0.01$, $N_{\mathcal{P}}=2$								
N_{spl}	h	E-error	V-error	rate w.r.t. N_{spl}	rate w.r.t. h	B-error	rate w.r.t. N_{spl}	rate w.r.t. h
1	6.40E−04	1.39E−17	2.56E−07	–	–	1.21E−07	–	–
2	2.26E−04	1.39E−17	2.31E−08	3.47	2.31	1.10E−08	3.46	2.30
4	8.00E−05	1.39E−17	1.60E−09	3.85	2.57	7.55E−10	3.86	2.58
8	2.83E−05	1.39E−17	9.58E−11	4.06	2.71	4.45E−11	4.08	2.72
16	1.00E−05	1.25E−16	1.83E−12	5.71	3.80	4.33E−13	6.68	4.45

Table 4.6
Dependence of the errors $E(u;1)-E(u^N;1)$, $V(u;1)-V(u^N;1)$ and $B(u;1)-B(u^N;1)$ on both N_{spl} and h for the D3 method for $\sigma=0.3$, $N_{\mathcal{P}}=3$ and the fixed parameter C: As N_{spl} is doubled, h is multiplied by factor of $(2\sqrt{2})^{-1}$.

$\sigma=0.3$, $N_{\mathcal{P}}=3$								
N_{spl}	h	E-error	V-error	rate w.r.t. N_{spl}	rate w.r.t. h	B-error	rate w.r.t. N_{spl}	rate w.r.t. h
1	1.92E−02	6.94E−18	2.30E−04	–	–	1.07E−04	–	–
2	6.79E−03	2.08E−17	2.08E−05	3.47	2.31	9.21E−06	3.54	2.36
4	2.40E−03	2.78E−17	1.43E−06	3.86	2.57	5.04E−07	4.19	2.79
8	8.49E−04	3.12E−16	8.31E−08	4.11	2.74	4.38E−09	6.85	4.57
16	3.00E−04	2.03E−15	1.57E−09	5.73	3.82	1.02E−08	−1.22	−0.81

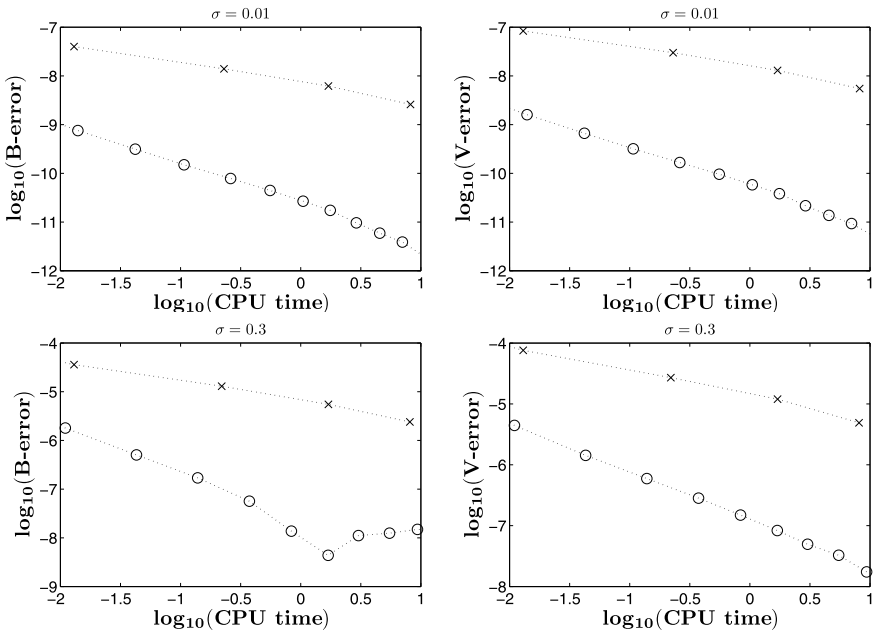

Fig. 4.5. Dependence of the errors $V(u;1)-V(u^N;1)$ and $B(u;1)-B(u^N;1)$ on the CPU time for the D3 (marked by ∘'s) and R3 (marked by ×'s) methods. Here, $\sigma=0.01$ for upper graphs and $\sigma=0.3$ for the lower ones.

4.5. Recovery of the point values out of the particle solutions

If one is interested in obtaining point values of the computed solutions, they must be recovered from the particle solution (4.5). A canonical approach is to use a smooth approximation of δ-functions which gives

$$u^N_\varepsilon(x,y,t)=u^N(x,y,t)*\zeta_\varepsilon=\sum_{i=1}^{N}\alpha_i(t)\zeta_\varepsilon\big(x-x_i(t),y-y_i(t)\big),$$

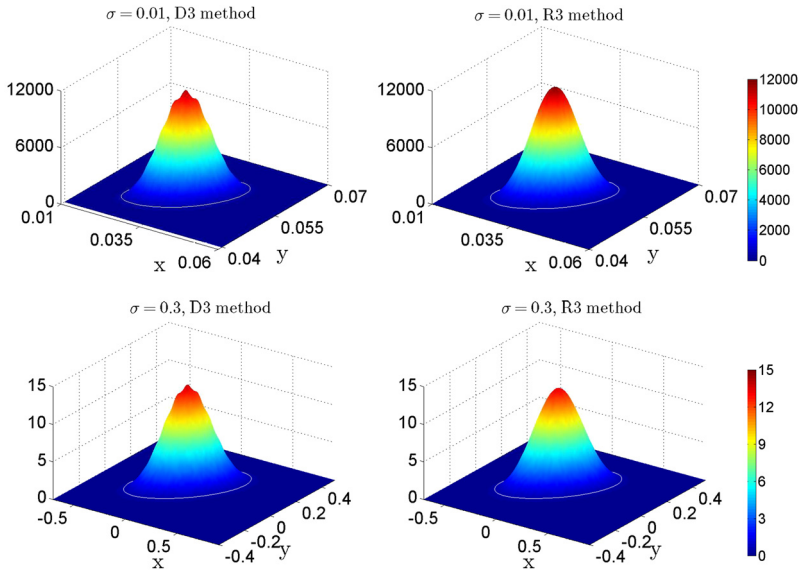

Fig. 4.6. Recovered point-values out of the particle solutions, computed by the D3 (left) and R3 (right) methods. We have used (4.6) with $\varepsilon = 0.0015$ for $\sigma = 0.01$ (upper two graphs) and $\varepsilon = 0.04$ for $\sigma = 0.3$ (lower two graphs).

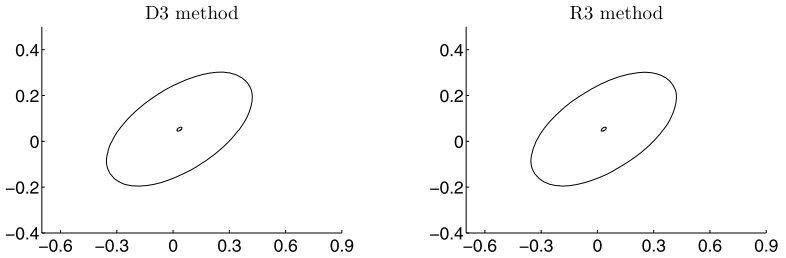

Fig. 4.7. The contour lines of $u = 0.03\,\|u\|_\infty$ for the recovered solutions. The contour lines for the D3 method is shown on the left and the ones for the R3 method are on the right. The small loops are the contour lines for solution with $\sigma = 0.01$, while the bigger loops are for $\sigma = 0.3$. The areas enclosed by the bigger loops are approximately 900 times larger than the area enclosed by the smaller ones.

where ζ_ε is a mollifier, see, e.g., [5,6,27]. In our numerical experiments, we take

$$\zeta_\varepsilon(x, y) = \frac{1}{\pi\varepsilon^2} e^{-\frac{x^2+y^2}{\varepsilon^2}}. \tag{4.6}$$

In Fig. 4.6, we show the solutions recovered from the D3 and R3 particle approximations at time $t = 1$ for both $\sigma = 0.01$ and $\sigma = 0.3$. The presented results are recovered from the same number of particles for both methods, which are 84 898 for $\sigma = 0.01$ and 396 444 for $\sigma = 0.3$. Those are the numbers of particles left at the final time when the D3 method is used: For $\sigma = 0.01$ we pick $h = 10^{-5}$, $N_{\text{spl}} = 16$, $N_{\mathcal{P}} = 2$ and for $\sigma = 0.3$, we take $h = 3 \times 10^{-4}$, $N_{\text{spl}} = 16$, $N_{\mathcal{P}} = 3$. We would like to emphasize that the computational domain has to be substantially enlarged for the larger value of σ, which makes it much more challenging to achieve high resolution at a reasonable computational cost. In Fig. 4.7, we compare the contour lines of $u = 0.03\,\|u\|_\infty$ for $\sigma = 0.01$ and $\sigma = 0.3$ (the same contour lines are also shown in Fig. 4.6). It shows that the area of the computational domain is approximately enlarged by a factor of 900 when σ is increased from $\sigma = 0.01$ to $\sigma = 0.3$.

As one can see, the D3 solution is not as smoothly recovered as the R3 one. The main reason is the difference in the structure of particle locations, which is plotted in Fig. 4.8. While the R3 particles are distributed rather uniformly (since in the deterministic particle method, each particle trajectory is completely independent of the trajectories of other particles, see Section 3.2.1), the distribution of the D3 particles is affected by the anisotropy in the diffusion operator: All of the deterministic particles are emerged out of one initial particle and thus aligned in the direction determined by the convective term in (1.1). While the values of E, V and B computed by the D3 method are clearly more accurate than the corresponding values obtained by the R3 method (see Section 4.4), recovering point values of the solution is a more challenging task as it is often the case when the solution is to be recovered from its particle distribution, see, e.g., [1,2,4].

Acknowledgement

The work of S. Cui and A. Kurganov was supported in part by the NSF Grant DMS-1115718.

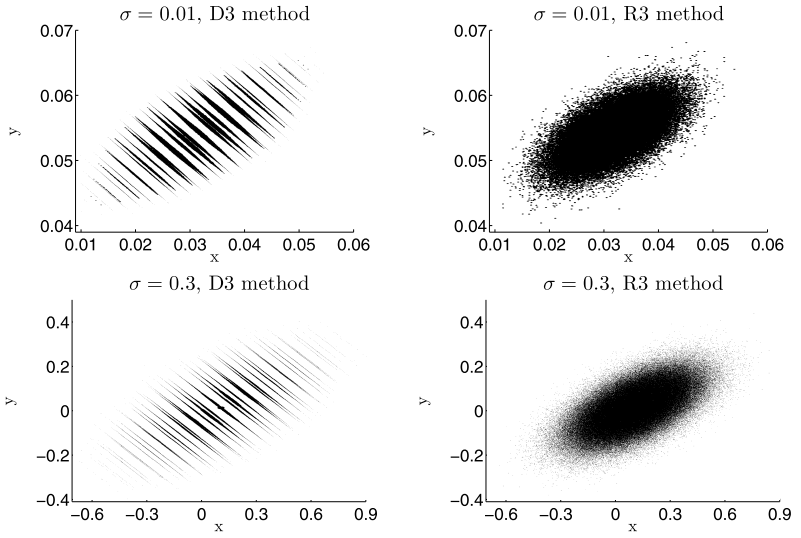

Fig. 4.8. The positions of particles obtained by the D3 (left) and R3 (right) methods. The size of each particle is proportional to its weight.

Appendix A. Derivation of (1.1), (1.2)

In this section, we briefly review several bond pricing models. Such mathematical models are based on either SDEs or partial differential equations (PDEs). For an extensive discussion on continuous time stochastic processes and the relationship between SDEs and PDEs we refer the reader to [23].

After purchasing a non-callable zero-coupon bond, the buyer will receive a fixed final amount of money at future time T, which is called the maturity. Final amount or face value can be very different, but we assume that it is equal to 1 unit of money. The price of bond is of great importance and interest. The tools determining the fair price of bond are called bond pricing models.

There are many factors involved in the bond pricing valuation, like dominated interest rate, probability of default of the issuer, supply and demand, human psychology, etc. For development of short rate models to defaultable bonds we refer the reader to [28], which discusses credit derivative pricing models in the case of a non-zero default probability with expected default payoff.

For the sake of simplicity, we only consider the main factor of the short rate (interest rate), that is, we assume that the bond price at time t_0 with maturity T only depends on the (future) short rates in the time interval $[t_0, T]$. The Vasicek model [31] is one of the SDEs which describes the evolution of the stochastic interest rate. The model gives the short term rate $x(t)$ as a solution of the following SDE:

$$dx(t) = \kappa\big(\hat{x} - x(t)\big)dt + \sigma dW_t, \tag{A.1}$$

where W_t is a Wiener process which models the randomness of market. The standard deviation parameter σ gives the volatility of the interest rate. The long term mean level is $\hat{x}$: We assume that in the long run the trajectories of the interest rate will stay around $\hat{x}$. When the interest rate runs far from the mean level, κ characterizes the speed at which the trajectories tend to return to the mean level. Even though a particular trajectory is random, the stochastic properties of the process are completely characterized by the parameters κ, $\hat{x}$, σ and initial condition $r_0 := x(t_0)$.

Given the interest rate $x(t)$, the bond price at time t_0 with maturity T is given by

$$B(t_0, T) = E\left[\exp\left\{-\int_{t_0}^{T} x(\tau)\,d\tau\right\}\right], \tag{A.2}$$

where E is an expectation. We define the accumulative integral of $x(t)$ by

$$y(t) := \int_{t_0}^{t} x(t)\,dt, \quad t \geq t_0,$$

which can obviously be rewritten in the form of the SDE

$$dy(t) = x(t)dt, \quad t \geq t_0, \tag{A.3}$$

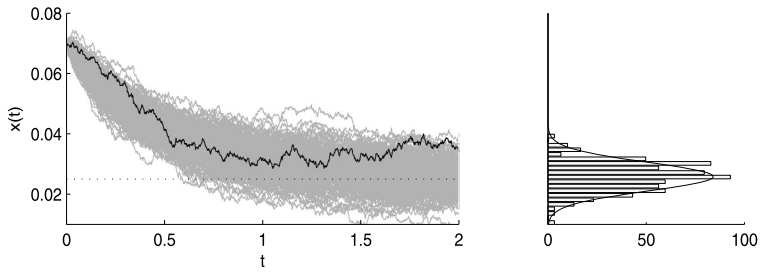

Fig. A.1. The left figure depicts 200 trajectories generated by the Vasicek model (A.1). One of the trajectories is plotted in black and the long term mean level $\hat{x}$ is plotted using a dotted line. On the right, the final values of the trajectories (at time $t = 2$) are represented in a histogram, which approximates the transitional pdf of $x(2)$ (the smooth line).

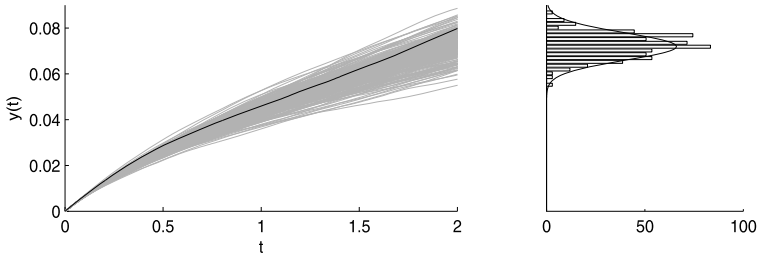

Fig. A.2. Typical trajectories of $y(t)$. The left figure depicts 200 trajectories of $y(t)$. On the right, the final values of the trajectories (at time $t = 2$) are represented in a histogram, which approximates the transitional pdf of $y(2)$ (the smooth line).

supplemented with the initial condition $y(t_0) = 0$. The bond price formula (A.2) can then be also written as

$$B(t_0, T) = E\big[\exp\{-y(T)\}\big], \tag{A.4}$$

which indicates that $B(t_0, T)$ can be computed directly from the pdf of $y(T)$.

For numerical methods for SDE models, we refer the reader, e.g., to [21]. The most popular numerical methods for SDEs are Monte-Carlo methods, which generate trajectories of $x(t)$ and $y(t)$ in order to obtain an approximation of their pdf's. A large number of realization is typically needed to obtain a good approximation, see, e.g., [12]. To illustrate this, we solve the SDEs (A.1) and (A.3) with the parameters $\sigma = 0.01$, $\kappa = 2$, $\hat{x} = 0.025$ and initial condition $x(0) = 0.07$ using a Monte-Carlo method and show the obtained solutions in Figs. A.1 and A.2.

Unlike the SDE models, the PDE ones are based on the evolution of the transitional pdf's. Consider a more general system of SDEs,

$$d\boldsymbol{x}(t) = \boldsymbol{a}(\boldsymbol{x}, t)dt + \boldsymbol{b}(\boldsymbol{x}, \boldsymbol{t})d\boldsymbol{W}_t,$$

for which one can write the Fokker–Planck equation to describe the evolution of the corresponding transitional pdf of $\boldsymbol{x}(t)$:

$$\partial_t u = -\sum_i \frac{\partial}{\partial x_i}\big(\boldsymbol{a}_i(\boldsymbol{x}, t)u\big) + \frac{1}{2}\sum_{i,j}\frac{\partial^2}{\partial x_i \partial x_j}\big[\big[\boldsymbol{b}(\boldsymbol{x}, t)\boldsymbol{b}^T(\boldsymbol{x}, t)\big]_{ij}u\big],$$

where $(\cdot)_i$ denotes the ith component of the vector and $[\cdot]_{ij}$ denotes the (i, j) entry of the matrix. The corresponding initial condition is

$$u(\boldsymbol{x}, 0) = \delta\big(\boldsymbol{x} - \boldsymbol{x}(0)\big),$$

where δ is the Dirac δ-function.

If we take

$$\boldsymbol{x}(t) = \big(x(t), y(t)\big)^T, \quad \boldsymbol{a}(\boldsymbol{x}, t) = \big(\kappa(\hat{x} - x), x\big)^T, \quad \boldsymbol{b}(\boldsymbol{x}, t) = (\sigma, 0)^T,$$

we obtain the SDE system, which in turn leads to the corresponding Fokker–Planck equation and initial condition (see [23]):

$$u_t + \kappa\big[(\hat{x} - x)u\big]_x + xu_y = \frac{\sigma^2}{2}u_{xx}, \tag{A.5}$$

$$u(x, y, 0) = \delta(x - r_0, y), \tag{A.6}$$

where the parameters κ, $\hat{x}$, σ and r_0 are the same as in Eq. (A.1). The IVP (A.5), (A.6) is exactly the IVP (1.1), (1.2), which is studied in this paper. Its solution, $u(x, y, t)$, gives us the transitional pdf of $(x(t), y(t))$. Then, one can compute the bond price as a weighted integral of $u(\cdot, \cdot, T)$:

$$B(t_0, T) = E\left[e^{-y(T)}\right] = \iint_{\mathbb{R}^2} e^{-y} u(x, y, T)\, dxdy. \tag{A.7}$$

Remark A.1. Another way to derive a PDE model associated with the SDE (A.1) is to use the Kolmogorov backward equation and Feynman–Kac formula (similarly to the approach in [11,19]), which result in the following PDE for the bond price $B(x, t)$:

$$B_t + \kappa(\hat{x} - x)B_x + xB + \frac{\sigma^2}{2} B_{xx} = 0, \tag{A.8}$$

which has to be solved backward in time, that is, a final time condition, the so-called payoff function, has to be prescribed. One can show that if the payoff function is

$$B(x, T) \equiv 1, \tag{A.9}$$

then the bond prices computed using (1.1), (1.2) and (A.8), (A.9) are the same.

We note that Eq. (A.8) is different from Eq. (1.1) studied in this paper though they both are derived from the same SDE (A.1).

Appendix B. Random walk as a Monte-Carlo method

In this section, we demonstrate that the stochastic particle method applied to the IVP (1.1), (1.2) is equivalent to the Monte-Carlo method applied to the corresponding SDEs (A.1) and (A.3).

Recall that the considered initial condition (1.2) consists of just one particle initially located at $(x_1(0), y_1(0)) = (r_0, 0)$. The trajectory of this particle is not deterministic due to the randomness of $\mathcal{S}^{\mathrm{R}}_{\mathcal{P}}$. Therefore, in order to obtain a numerical solution of (1.1), (1.2), one needs to conduct a number of experiments, which would give several particle trajectories. Each particle trajectory by itself can give an approximation of the random variable $e^{-y(T)}$. By repeatedly doing such realizations, we will obtain a series of samples whose mean approaches the bond price.

For computational simplicity, we initially set N particles at $(r_0, 0)$, each with the weight $1/N$. These particles spread at latter time due to the random perturbation in $S^R_{\mathcal{P}}$. The resulting bond price computed by (A.7) can be proved equivalent to averaging the bond prices from each trajectory. Furthermore, the obtained N particles form particle solution of the pdf of $(x(t), y(t))$.

According to the derivation of (1.1) given in Appendix A, the x-coordinate of the particle corresponds to the interest rate. We will now show that solving (1.1) using $\mathcal{S}^{\mathrm{R}}_{\mathcal{P}}$ is equivalent to solving the SDEs (A.1) and (A.3) by a Monte-Carlo method. Thus, the designed random walk based particle method is in fact a Monte-Carlo method for (A.1) and (A.3).

Indeed, the evolution of $x_i(t)$ is determined by the initial condition $x_i(0) = r_0$ and the following update formula:

$$x_i(t + \Delta t) = \left(x_i(t) - \hat{x}\right)e^{-\kappa \Delta t} + \hat{x} + \xi_j,$$

which is the combination of Eqs. (3.4) and (3.6) for $x_i(t)$. Adding the random numbers ξ_j's at each time step can be interpreted as a numerical treatment of the Wiener term σdW_t in (A.1). Thus, $x_i(t)$ actually gives a numerical solution of the SDE (A.1).

The y-coordinate also has its connection to the SDE (A.3). Since the operator $\mathcal{S}_{\mathcal{P}}$ does not effect the y-coordinate, the evolution of $y_i(t)$ is only determined by $\mathcal{S}_{\mathcal{H}}$, which is given by (2.3). This means that $y_i(t)$ is an anti-derivative of $x_i(t)$. Given its initial condition $y_i(0) = 0$, the y-coordinate of the particles is in fact a numerical approximation of $y(t)$ in (A.3).

References

[1] A. Chertock, A. Kurganov, On a practical implementation of particle methods, Appl. Numer. Math. 56 (2006) 1418–1431.
[2] A. Chertock, A. Kurganov, Computing multivalued solutions of pressureless gas dynamics by deterministic particle methods, Commun. Comput. Phys. 5 (2009) 565–581.
[3] A. Chertock, A. Kurganov, On splitting-based numerical methods for convection-diffusion equations, in: G. Puppo, G. Russo (Eds.), Numerical Methods for Balance Laws, vol. 24, Aracne editrice S.r.l, Roma, 2009, pp. 303–343.
[4] A. Chertock, A. Kurganov, Y. Rykov, A new sticky particle method for pressureless gas dynamics, SIAM J. Numer. Anal. 45 (2007) 2408–2441.
[5] A. Chorin, Numerical study of slightly viscous flow, J. Fluid Mech. 57 (4) (1973) 785–796.
[6] G.-H. Cottet, P. Koumoutsakos, Vortex Methods, Cambridge University Press, Cambridge, 2000.
[7] G.-H. Cottet, S. Mas-Gallic, A particle method to solve transport-diffusion equations, part 1: the linear case, Tech. Rep. 115, Ecole Polytechnique, Palaiseau, France, 1983.
[8] P. Degond, S. Mas-Gallic, The weighted particle method for convection-diffusion equations. part 1 and part 2, Math. Comput. 53 (1989) 485–525.
[9] M.W. Evans, F.H. Harlow, The particle-in-cell method for hydrodynamic calculations, Tech. Rep. LA-2139, Los Alamos Scientific Laboratory, 1957.

[10] P. Foschi, S. Pagliarani, A. Pascucci, Approximations for Asian options in local volatility models, J. Comput. Appl. Math. 237 (2013) 442–459.
[11] C.W. Gardiner, Handbook of Stochastic Methods: For Physics, Chemistry and the Natural Sciences, Springer, 2002.
[12] P. Glasserman, Monte Carlo Methods in Financial Engineering (Stochastic Modelling and Applied Probability), vol. 53, Springer, 2003.
[13] M. Griebel, M. Schweitzer, A particle-partition of unity method for the solution of elliptic, parabolic, and hyperbolic PDEs, SIAM J. Sci. Comput. 22 (2000) 853–890 (electronic).
[14] O. Grishchenko, X. Han, V. Nistor, A volatility-of-volatility expansion of the option prices in the SABR stochastic volatility model. Preprint available at SSRN: http://dx.doi.org/10.2139/ssrn.2374004,.
[15] F. Harlow, The particle-in-cell computing method for fluid dynamics, in: B. Alder, S. Fernbach, M. Rotenberg (Eds.), Methods in Computational Physics, vol. 3, Academic Press, New York, 1964, pp. 319–343.
[16] F.H. Harlow, A machine calculation method for hydrodynamic problems, Tech. Rep. LAMS-1956, Los Alamos Scientific Laboratory, 1956.
[17] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, Taylor & Francis, 1989.
[18] W. Hundsdorfer, J. Verwer, Numerical Solution of Time-Dependent Advection–Diffusion–Reaction Equations, Springer Ser. Comput. Math., vol. 33, Springer-Verlag, Berlin, 2003.
[19] S. Janson, J. Tysk, Feynman–Kac formulas for Black-Scholes-type operators, Bull. Lond. Math. Soc. 38 (2) (2006) 269–282.
[20] H. Jia, K. Li, A third accurate operator splitting method, Math. Comput. Model. 53 (1–2) (2011) 387–396.
[21] P.E. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, Springer, 2011.
[22] A. Leonard, Vortex methods for flow simulation, J. Comput. Phys. 37 (3) (1980) 289–335.
[23] A. Lipton, Mathematical Methods for Foreign Exchange: A Financial Engineer's Approach, World Scientific Publishing Company, Incorporated, 2001.
[24] R. Mamon, Three ways to solve for bond prices in the Vasicek model, J. Appl. Math. Decis. Sci. 8 (1) (2004) 1–14.
[25] S. Mas-Gallic, P. Raviart, Particle approximation of convection-diffusion equations, International Report 86013, Université Pierre et Marie Curie, 1986.
[26] S. Mas-Gallic, P. Raviart, Particle approximation of convection-diffusion equations, J. Comput. Phys. 97 (1991) 366–397.
[27] P.-A. Raviart, An analysis of particle methods, in: Numerical Methods in Fluid Dynamics, Como, 1983, in: Lect. Notes Math., vol. 1127, Springer, Berlin, 1985, pp. 243–324.
[28] P.J. Schönbucher, Credit Derivatives Pricing Models: Model, Pricing and Implementation, Wiley, 2003.
[29] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506–517.
[30] H. Trotter, On the product of semi-groups of operators, Proc. Am. Math. Soc. 10 (1959) 545–551.
[31] O. Vasicek, An equilibrium characterization of the term structure, J. Financ. Econ. 5 (2) (1977) 177–188.