

A MULTISTAGE WIENER CHAOS EXPANSION METHOD FOR STOCHASTIC ADVECTION-DIFFUSION-REACTION EQUATIONS*

Z. ZHANG[†], B. ROZOVSKII[†], M. V. TRETYAKOV[‡], AND G. E. KARNIADAKIS[†]

Abstract. Using Wiener chaos expansion (WCE), we develop numerical algorithms for solving second-order linear parabolic stochastic partial differential equations (SPDEs). We propose a deterministic WCE-based algorithm for computing moments of the SPDE solutions without any use of the Monte Carlo technique. We also compare the proposed deterministic algorithm with two other numerical methods based on the Monte Carlo technique and demonstrate that the new method is more efficient for highly accurate solutions. Numerical tests verify that the scheme is of mean-square order $O(\frac{\Delta^{N/2}}{\sqrt{(N+1)!}})$ for diffusion and for diffusion-reaction SPDEs with constant or variable coefficients, where Δ is the time step, and N is the Wiener chaos order.

Key words. long-term integration, passive scalar, multiplicative white noise

AMS subject classifications. Primary, 60H15; Secondary, 35R60, 60H40

DOI. 10.1137/110849572

1. Introduction. In this paper we develop a new numerical method, based on nonlinear filtering ideas and spectral expansions, for advection-diffusion-reaction equations perturbed by random fluctuations, which form a broad class of second-order linear parabolic stochastic differential equations (SPDEs). The standard approach to constructing SPDE solvers starts with a space discretization of an SPDE, for which spectral methods (see, e.g., [4, 10, 14]), finite element methods (see, e.g., [1, 8, 32]) or spatial finite differences (see, e.g., [1, 11, 30, 33]) can be used. The result of such a space discretization is a large system of ordinary stochastic differential equations (SDEs) which requires time discretization to complete a numerical algorithm. In [5, 6] an SPDE is first discretized in time and then a finite element or finite difference method can be applied to this semidiscretization. Other numerical approaches include those making use of splitting techniques [2, 17, 12], quantization [9], or an approach based on the averaging-over-characteristic formula [26, 27]. In [22, 19] numerical algorithms based on the Wiener chaos expansion (WCE) were introduced for solving the nonlinear filtering problem for hidden Markov models. Since then the WCE-based numerical methods have been successfully developed in a number of directions (see, e.g., [13, 31]).

In computing moments of SPDE solutions, the existing approaches to solving SPDEs are usually complemented by the Monte Carlo technique. Consequently, in these approaches numerical approximations of SPDE moments have two errors: numerical integration error and Monte Carlo (statistical) error. To reach a high accuracy, we have to run a very large number of independent simulations of the SPDE to reduce the Monte Carlo error. Instead, here we exploit WCE numerical methods to construct a *deterministic* algorithm for computing moments of the SPDE solutions without any

*Submitted to the journal's Methods and Algorithms for Scientific Computing section September 28, 2011; accepted for publication January 30, 2012; published electronically March 27, 2012. This work was supported by the DOE and the OSD/AFOSR MURI.

<http://www.siam.org/journals/sisc/34-2/84957.html>

[†]Division of Applied Mathematics, Brown University, Providence, RI 02912 (zhongqiang-zhang@brown.edu, boris_rozovsky@brown.edu, george_karniadakis@brown.edu).

[‡]Department of Mathematics, University of Leicester, Leicester LE1 7RH, UK (M.Tretyakov@le.ac.uk).

use of the Monte Carlo technique.

The rest of the paper is organized as follows. In section 2 we introduce the linear SPDE considered in the paper and recall the definition of its Wiener chaos solution. In section 3 we revisit the method employed in [19] and apply it to a more general linear SPDE than the one treated in [19]. Based on this, the algorithm for computing moments of the SPDE solutions is introduced in section 4. To demonstrate the effectiveness of the proposed algorithm, we perform a number of numerical tests. In section 5 we test the algorithm on two one-dimensional SPDEs and confirm its theoretical order of convergence. In section 6 we apply this algorithm to the passive scalar equation in the periodic case in two dimensions. In both section 5 and section 6 we also compare the WCE-based algorithm with algorithms exploiting the Monte Carlo technique and demonstrate that while the proposed WCE-based algorithm is slower than Monte Carlo-type methods in getting results of low accuracy, in reaching higher accuracy the WCE-based algorithm can be more efficient. A summary and discussion on possible extensions are given in section 7.

2. WCE of the SPDE solution. Let (Ω, \mathcal{F}, P) be a complete probability space, \mathcal{F}_t , $0 \leq t \leq T$ be a filtration satisfying the usual hypotheses, and $(w(t), \mathcal{F}_t) = (\{w_k(t), k \geq 1\}, \mathcal{F}_t)$ be a system of one-dimensional independent standard Wiener processes. Let \mathcal{D} be an open domain in \mathbb{R}^d . Consider the following SPDE written in the form of Itô:

$$(2.1) \quad \begin{aligned} du(t, x) &= \mathcal{L}u(t, x) + f(x) + \sum_{k \geq 1} [\mathcal{M}_k u(t, x) + g_k(x)] dw_k(t), \quad (t, x) \in (0, T] \times \mathcal{D}, \\ u(0, x) &= u_0(x), \quad x \in \mathcal{D}, \end{aligned}$$

where

$$(2.2) \quad \begin{aligned} \mathcal{L}u(t, x) &= \sum_{i,j=1}^d a_{ij}(x) D_i D_j u(t, x) + \sum_{i=1}^d b_i(x) D_i u(t, x) + c(x) u(t, x), \\ \mathcal{M}_k u(t, x) &= \sum_{i=1}^d b_i^k(x) D_i u(t, x) + h^k(x) u(t, x), \end{aligned}$$

and $D_i := \partial_{x_i}$. We assume that \mathcal{D} is either bounded with a regular boundary or that $\mathcal{D} = \mathbb{R}^d$. In the former case we will consider periodic boundary conditions and in the latter the Cauchy problem. We also assume that the coefficients of operators \mathcal{L} and \mathcal{M} are uniformly bounded and $\mathcal{L} - \frac{1}{2} \sum_{k \geq 1} \mathcal{M}_k \mathcal{M}_k$ is nonnegative definite. When the coefficients of \mathcal{L} and \mathcal{M} are sufficiently smooth, existence and uniqueness results for the solution of (2.1)–(2.2) are available (see, e.g., [28]) and under weaker assumptions (see, e.g., [23, 21]).

Now let us recall (see details in [23, 20, 21]) the definition of a Wiener chaos solution to the linear SPDE (2.1)–(2.2). Denote by \mathcal{J} the set of multi-indices $\alpha = (\alpha_{k,l})_{k,l \geq 1}$ of finite length $|\alpha| = \sum_{i,k=1}^{\infty} \alpha_{k,l}$, i.e.,

$$\mathcal{J} = \{ \alpha = (\alpha_{k,l}, k, l \geq 1), \alpha_{k,l} \in \{0, 1, 2, \dots\}, |\alpha| < \infty \}.$$

Here k denotes the number of Wiener processes and l the number of Gaussian random variables approximating each Wiener process, as will be shown shortly. We represent

the solution of (2.1)–(2.2) as

$$(2.3) \quad u(t, x) = \sum_{\alpha \in \mathcal{J}} \frac{1}{\sqrt{\alpha!}} \varphi_\alpha(t, x) \xi_\alpha,$$

where $\{\xi_\alpha\}$ is a complete orthonormal system (CONS) in $L^2(\Omega, \mathcal{F}_t, P)$, $\alpha! = \prod_{k,l} (\alpha_{k,l}!)$, and φ_α satisfies the following system of equations (the propagator):

$$(2.4) \quad \begin{aligned} \frac{\partial \varphi_\alpha(s, x)}{\partial s} &= \mathcal{L} \varphi_\alpha(s, x) + f(x) \mathbf{1}_{\{|\alpha|=0\}} \\ &+ \sum_{k,l} \alpha_{k,l} m_l(s) [\mathcal{M}_k \varphi_{\alpha^-(l,k)}(s, x) + g_k(x) \mathbf{1}_{\{|\alpha|=1\}}], \quad 0 < s \leq t, \quad x \in \mathcal{D}, \\ \varphi_\alpha(0, x) &= u_0(x) \mathbf{1}_{\{|\alpha|=0\}}, \quad x \in \mathcal{D}, \end{aligned}$$

where $\alpha^-(l, k)$ is the multi-index with components

$$(2.5) \quad (\alpha^-(l, k))_{i,j} = \begin{cases} \max(0, \alpha_{i,j} - 1) & \text{if } i = l \text{ and } j = k, \\ \alpha_{i,j} & \text{otherwise.} \end{cases}$$

The random variables ξ_α in (2.3) are constructed according to the Cameron–Martin theorem [3]:

$$(2.6) \quad \xi_\alpha := \prod_{\alpha} \left(\frac{H_{\alpha_{k,l}}(\xi_{k,l})}{\sqrt{\alpha_{k,l}!}} \right), \quad \alpha \in \mathcal{J},$$

where $\{m_l\} = \{m_l(s)\}_{l \geq 1}$ is a CONS in $L^2([0, t])$, $\xi_{k,l} = \int_0^t m_l(s) dw_k(s)$, and H_n is the n th Hermite polynomial:

$$(2.7) \quad H_n(x) = (-1)^n e^{x^2/2} \frac{d^n}{dx^n} e^{-x^2/2}.$$

The representation (2.3)–(2.7) is called a WCE of the SPDE solution. It is clear that a truncation of the WCE (2.3) presents a possibility for constructing numerical methods for SPDEs. This is considered in the next section.

3. Multistage WCE method. In addition to the multi-index length $|\alpha| = \sum_{i,k=1}^\infty \alpha_{k,i}$, we define the order of multi-index

$$\alpha : d(\alpha) = \max \{l \geq 1 : \alpha_{k,l} > 0 \text{ for some } k \geq 1\}$$

and the truncated set of multi-indices

$$\mathcal{J}_{N,n} = \{\alpha \in \mathcal{J} : |\alpha| \leq N, \quad d(\alpha) \leq n\}.$$

Here N is the highest Hermite polynomial order and n is the maximum number of Gaussian random variables for each Wiener process. Using (2.3), we introduce the truncated Wiener chaos solution

$$(3.1) \quad u_{N,n}(t, x) = \sum_{\alpha \in \mathcal{J}_{N,n}} \frac{1}{\sqrt{\alpha!}} \varphi_\alpha(t, x) \xi_\alpha.$$

We choose the basis $\{m_l(s)\}_{l \geq 1}$ as

$$(3.2) \quad m_1(s) = \frac{1}{\sqrt{t}}, \quad m_l(s) = \sqrt{\frac{2}{t}} \cos\left(\frac{\pi(l-1)s}{t}\right), \quad l \geq 2, \quad 0 \leq s \leq t.$$

See a discussion on selection of basis in [19].

The truncated expansion (3.1) together with (2.4), (2.6), and (3.2) gives us a constructive approximation of the solution to (2.1), where implementation requires that we numerically solve the propagator (2.4).

It is proved in [19, Theorem 2.2] that when $b_i^k(t, x) = 0$, $c = 0$, $g = 0$ (reaction-diffusion equation), and the number of noises is finite there is a constant $C > 0$ such that for any $t \in (0, T]$

$$(3.3) \quad E \|u_{N,n}(t, \cdot) - u(t, \cdot)\|_{L^2}^2 \leq Ce^{Ct} \left(\frac{(Ct)^{N+1}}{(N+1)!} + \frac{t^3}{n} \right).$$

Our preliminary analysis shows that in the *general case* of (2.1), the error estimate (3.3) is expected to be

$$(3.4) \quad E \|u_{N,n}(t, \cdot) - u(t, \cdot)\|_{L^2}^2 \leq Ce^{Ct} \left(\frac{(Ct)^{N+1}}{(N+1)!} + \frac{t^2}{n} \right).$$

A rigorous proof of such a result will appear elsewhere.

It follows from the error estimates (3.3) and (3.4) that the error of the approximation $u_{N,n}(t, \cdot)$ grows exponentially in time t , which severely limits its practical use. To overcome this difficulty, it was proposed in [19] to introduce a time discretization with step $\Delta > 0$ and view (3.1), (2.4), (2.6), (3.2) as the one-step approximation of the SPDE solution based on which an effective numerical method applicable to longer time intervals was constructed.

To this end, let us introduce the multistep basis for the WCE and its corresponding propagator. Let $0 = t_0 < t_1 < \dots < t_K = T$ be a uniform partition of the time interval $[0, T]$ with time step size Δ ; see Figure 3.1. Let $\{m_k^{(i)}\} = \{m_k^{(i)}(s)\}_{k \geq 1}$ be the following CONS in $L^2([t_{i-1}, t_i])$:

$$(3.5) \quad \begin{aligned} m_l^{(i)} &= m_l(s - t_{i-1}), \quad t_{i-1} \leq s \leq t_i, \\ m_l(s) &= \frac{1}{\sqrt{\Delta}}, \quad m_l(s) = \sqrt{\frac{2}{\Delta}} \cos\left(\frac{\pi(l-1)s}{\Delta}\right), \quad l \geq 2, \quad 0 \leq s \leq \Delta, \\ m_l(s) &= 0, \quad l \geq 1, \quad s \notin [0, \Delta]. \end{aligned}$$

Define the random variables $\xi_\alpha^{(i)}$, $i = 1, \dots, K$, as

$$(3.6) \quad \xi_\alpha^{(i)} := \prod_\alpha^{(i)} \left(\frac{H_{\alpha_{k,l}}(\xi_{k,l}^{(i)})}{\sqrt{\alpha_{k,l}!}} \right), \quad \alpha \in \mathcal{J},$$

where $\xi_{k,l}^{(i)} = \int_{t_{i-1}}^{t_i} m_l^{(i)}(s) dw_k(s)$, and H_n are Hermite polynomials (2.7).

Let

$$(3.7) \quad u_{\Delta,N,n}(0, x) = u_0(x),$$

and by induction for $i = 1, \dots, K$ let

$$(3.8) \quad u_{\Delta,N,n}(t_i, x) = \sum_{\alpha \in \mathcal{J}_{N,n}} \frac{1}{\sqrt{\alpha!}} \varphi_\alpha^{(i)}(\Delta, x) \xi_\alpha^{(i)},$$



FIG. 3.1. Illustration of the idea of multistage WCE. The dotted line denotes the “offline” computation, where we solve the propagator up to time Δ . The dashed line implies that one solves only the solution on certain time levels instead of on the entire time interval.

where $\varphi_\alpha^{(i)}(\Delta, x)$ solves the system

$$(3.9) \quad \begin{aligned} \frac{\partial \varphi_\alpha^{(i)}(s, x)}{\partial s} &= \mathcal{L}\varphi_\alpha^{(i)}(s, x) + f(x)\mathbf{1}_{\{|\alpha|=0\}} \\ &+ \sum_{k,l} \alpha_{k,l} m_l^{(i)}(s) \left[\mathcal{M}_k \varphi_{\alpha-(l,k)}^{(i)}(s, x) + g_k(x)\mathbf{1}_{\{|\alpha|=1\}} \right], \quad s \in (0, \Delta], \\ \varphi_\alpha^{(i)}(0, x) &= u_{\Delta, N, n}(t_{i-1}, x)\mathbf{1}_{\{|\alpha|=0\}}. \end{aligned}$$

Thus, (3.7)–(3.9) together with (3.5) and (3.6) gives us a recursive method (called the RWCE method) for solving the SPDE (2.1), where implementation requires us to numerically solve the propagator (3.9) at every time step.

Based on the one-step error (3.3), the following global error estimate for the RWCE method is proved in [19, Theorem 2.4] (the case of $b_i^k(t, x) = 0, c = 0, g = 0$, and a finite number of noises):

$$(3.10) \quad E \|u_{\Delta, N, n}(t_i, \cdot) - u(t_i, \cdot)\|_{L^2}^2 \leq C e^{CT} \left(\frac{(C\Delta)^N}{(N+1)!} + \frac{\Delta^2}{n} \right), \quad i = 1, \dots, K,$$

for some $C > 0$ independent of Δ, N , and n ; i.e., this method is of global mean-square order $O(\frac{\Delta^{N/2}}{\sqrt{(N+1)!}} + \frac{\Delta}{\sqrt{n}})$. Moreover, based on (3.4), one can prove that in the general case of (2.1) (advection-diffusion-reaction equations) the error estimate for the RWCE method will have the form

$$(3.11) \quad E \|u_{\Delta, N, n}(t_i, \cdot) - u(t_i, \cdot)\|_{L^2}^2 \leq C e^{CT} \left(\frac{(C\Delta)^N}{(N+1)!} + \frac{\Delta}{n} \right), \quad i = 1, \dots, K;$$

i.e., this method is of mean-square order $O(\frac{\Delta^{N/2}}{\sqrt{(N+1)!}} + \sqrt{\frac{\Delta}{n}})$.

As we already mentioned, the RWCE method requires us to solve the propagator (3.9) at every time step, which is computationally rather expensive. To reduce the cost, we introduce a modification of this method following [19]. The idea is to expand the initial condition $u_0(x)$ in a basis $\{e_m\}$, present $u_{\Delta, N, n}(t_{i-1}, x)$ as $u_{\Delta, N, n}(t_{i-1}, x) = \sum_m c_m e_m(x)$, and note that $\varphi_\alpha(\Delta, x; u_{\Delta, N, n}(t_{i-1}, \cdot)) = \sum_m c_m \varphi_\alpha(\Delta, x; e_m)$, where $\varphi_\alpha(s, x; \phi)$ is the solution of the propagator (3.9) with the initial condition $\phi(x)$.

The idea is illustrated in Figure 3.1 with the help of a sketch. We can first compute the propagator (3.12) (see below) on $(0, \Delta]$ and obtain a problem-dependent basis $q_{\alpha, l, m}$ (3.13). This step is called “offline” as in [19]. Thus, one recursively computes the solution “online” by (3.14) and (3.15) only at time $i\Delta$ ($i = 2, \dots, K$) using the obtained basis $q_{\alpha, l, m}$. Specifically, we proceed as follows. Let $\{e_m\} = \{e_m(x)\}_{m \geq 1}$ be a CONS in $L^2(\mathcal{D})$ with boundary conditions satisfied and (\cdot, \cdot) be the inner product

in that space. Let $\varphi_\alpha(s, x; \phi)$ solve the following propagator:

$$(3.12) \quad \begin{aligned} \frac{\partial \varphi_\alpha(s, x; \phi)}{\partial s} &= \mathcal{L}\varphi_\alpha(s, x; \phi) + f(x)\mathbf{1}_{\{|\alpha|=0\}} \\ &+ \sum_{k,l} \alpha_{k,l} m_l(s) [\mathcal{M}_k \varphi_{\alpha-(l,k)}(s, x; \phi) + g_k(x)\mathbf{1}_{\{|\alpha|=1\}}], \quad s \in (0, \Delta], \\ \varphi_\alpha^{(i)}(0, x) &= \phi(x)\mathbf{1}_{\{|\alpha|=0\}}, \end{aligned}$$

where $m_l(s)$ are as in (3.2). Define

$$(3.13) \quad q_{\alpha,l,m} = (\varphi_\alpha(\Delta, \cdot; e_l), e_m), \quad l, m \geq 1,$$

and then find by induction the coefficients

$$(3.14) \quad \begin{aligned} \psi_m(0; N, n) &:= (u_0, e_m), \\ \psi_m(i; N, n) &:= \sum_{\alpha \in \mathcal{J}_{N,n}} \sum_l \frac{1}{\sqrt{\alpha!}} \psi_l(i-1; N, n) q_{\alpha,l,m} \xi_\alpha^{(i)}, \quad i = 1, \dots, K. \end{aligned}$$

It is proved in [19, Theorem 2.5] that

$$(3.15) \quad u_{\Delta, N, n}(t_i, x) = \sum_m \psi_m(i; N, n) e_m(x), \quad i = 0, \dots, K, \quad P\text{-a.s.}$$

We refer to the numerical method (3.15), (3.12)–(3.14) together with (3.5)–(3.6) as the *multistage WCE method* for the SPDE (2.1).

In practice, if (2.1) has an infinite number of Wiener processes, we truncate them to a finite number $r \geq 1$ of noises. We introduce the correspondingly truncated set $\mathcal{J}_{N,n,r}$ so that

$$\mathcal{J}_{N,n,r} = \{\alpha \in \mathcal{J} : |\alpha| \leq N, \quad d_r(\alpha) \leq n\},$$

where $d_r(\alpha) = \max\{l \geq 1 : \alpha_{k,l} > 0 \text{ for some } 1 \leq k \leq r\}$.

ALGORITHM 1. Choose a truncation of the number of noises $r \geq 1$ and the algorithm’s parameters: a CONS $\{e_m(x)\}_{m \geq 1}$ and its truncation $\{e_m(x)\}_{m=1}^M$; a time step Δ ; N and n which together with r determine the size of the multi-index set $\mathcal{J}_{N,n,r}$.

Step 1. For each $m = 1, \dots, M$, solve the propagator (3.12) for $\alpha \in \mathcal{J}_{N,n,r}$ on the time interval $[0, \Delta]$ with the initial condition $e_m(x)$ and denote the obtained solution as $\varphi_\alpha(\Delta, x; e_m)$, $\alpha \in \mathcal{J}_{N,n,r}$, $m = 1, \dots, M$. Note in this step that we need to also choose a time step size δt to solve the equations in the propagator numerically.

Step 2. Evaluate $\psi_m(0; N, n, M) = (u_0, e_m)$, $m = 1, \dots, M$, where $u_0(x)$ is the initial condition for (2.1), and $q_{\alpha,l,m} = (\varphi_\alpha(\Delta, \cdot; e_l), e_m(\cdot))$, $l, m = 1, \dots, M$.

Step 3. On the i th time step (at time $t = i\Delta$), generate the Gaussian random variables $\xi_\alpha^{(i)}$, $\alpha \in \mathcal{J}_{N,n,r}$, according to (3.6), compute the coefficients

$$\psi_m(i; N, n, M) = \sum_{\alpha \in \mathcal{J}_{N,n,r}} \sum_{l=1}^M \frac{1}{\sqrt{\alpha!}} \psi_l(i-1; N, n, M) q_{\alpha,l,m} \xi_\alpha^{(i)}, \quad m = 1, \dots, M,$$

and obtain the approximate solution of (2.1)

$$u_{\Delta, N, n}^M(t_i, x) = \sum_{m=1}^M \psi_m(i; N, n, M) e_m(x).$$

Algorithm 1 coincides with the algorithm proposed in [19] for (2.1) in the case of $b_i^k(t, x) = 0$, $c = 0$, $g = 0$, and a finite number of noises but generalizes it to a wider class of linear SPDEs of the form (2.1). In particular, the algorithm from [19] was applied to the nonlinear filtering problem for hidden Markov models in the case of independent noises in signal and observation, while Algorithm 1 is also applicable when noises in signal and observation are dependent.

Algorithm 1 allows us to simulate mean-square approximations of the solution to the SPDE (2.1). It can also be used together with the Monte Carlo technique for computing expectations of functionals of the solution to (2.1). In the next section we propose an algorithm based on Algorithm 1, which allows us to compute moments of the solution to (2.1) without using the Monte Carlo technique.

Remark 3.1. We note that the cost of simulation of the random field $u(t_i, x)$ by Algorithm 1 over K time steps is proportional to $KM^2 \frac{(N+nr)!}{N!(nr)!}$.

4. Algorithm for computing moments. Implementation of Algorithm 1 requires the generation of the random variables $\xi_\alpha^{(i)}$ (see (3.6)). Then, for computing moments of the solution of the SPDE problem (2.1), we also need to make use of the Monte Carlo technique. As is well known, Monte Carlo methods have a low rate of convergence. In this section we present a deterministic algorithm (Algorithm 2) for computing moments, i.e., an algorithm which does not require any random numbers and does not have a statistical error. In sections 5 and 6 we compare Algorithm 2 with some Monte Carlo-type methods and demonstrate that Algorithm 2 can be more computationally efficient when higher accuracy is required.

First, it is not difficult to see that the mean solution $Eu(t, x)$ is equal to the solution $\varphi_{(0)}(t, x)$ of the propagator (3.12) with $\alpha = (0)$:

$$Eu(t, x) = \varphi_{(0)}(t, x).$$

Thus evaluating the mean $Eu(t, x)$ is reduced to numerical solution of the linear deterministic PDE for $\varphi_{(0)}(t, x)$.

We limit ourselves here to presenting an algorithm for computing the second moment of the solution, $Eu^2(t, x)$. Other moments of the solution $u(t, x)$ can be considered analogously.

According to Algorithm 1, we approximate the solution $u(t_i, x)$ of (2.1) by $u_{\Delta, N, n}^M(t_i, x)$ as follows:

$$\begin{aligned} \psi_m(0; N, n, M) &= (u_0, e_m), \quad m = 1, \dots, M, \\ \psi_m(t_i; N, n, M) &= \sum_{\alpha \in \mathcal{J}_{N, n, r}} \sum_{l=1}^M \frac{1}{\sqrt{\alpha!}} \psi_l(t_{i-1}; N, n, M) q_{\alpha, l, m} \xi_\alpha^{(i)}, \quad m = 1, \dots, M, \\ u_{\Delta, N, n}^M(t_i, x) &= \sum_{m=1}^M \psi_m(t_i; N, n, M) e_m(x), \quad i = 1, \dots, K, \end{aligned}$$

where $q_{\alpha, l, m}$ are from (3.13) and $\xi_\alpha^{(i)}$ are from (3.6). Then, we can evaluate the

covariance matrices

$$\begin{aligned}
 (4.1) \quad Q_{lm}(0; N, n, M) &:= \psi_l(0; N, n, M)\psi_m(0; N, n, M), \quad l, m = 1, \dots, M, \\
 Q_{lm}(t_i; N, n, M) &:= E[\psi_l(t_i; N, n, M)\psi_m(t_i; N, n, M)] \\
 &= \sum_{j,k=1}^M Q_{jk}(t_{i-1}; N, n, M) \sum_{\alpha \in \mathcal{J}_{N,n,r}} \frac{1}{\alpha!} q_{\alpha,j,l} q_{\alpha,k,m}, \\
 l, m &= 1, \dots, M, \quad i = 1, \dots, K,
 \end{aligned}$$

and, consequently, the second moment of the approximate solution

$$(4.2) \quad E[u_{\Delta,N,n}^M(t_i, x)]^2 = \sum_{l,m=1}^M Q_{lm}(t_i; N, n, M) e_l(x) e_m(x), \quad i = 1, \dots, K.$$

We note that implementation of (4.1)–(4.2) does not require generation of the random variables $\xi_\alpha^{(i)}$. Hence we have constructed a deterministic algorithm for computing the second moments of the solution to the SPDE (2.1), which we formulate below.

ALGORITHM 2. Choose a truncation of the number of noises $r \geq 1$ in (2.1) and the algorithm’s parameters: a CONS $\{e_m(x)\}_{m \geq 1}$ and its truncation $\{e_m(x)\}_{m=1}^M$; a time step Δ ; N and n which together with r determine the size of the multi-index set $\mathcal{J}_{N,n,r}$.

Step 1. For each $m = 1, \dots, M$, solve the propagator (3.12) for $\alpha \in \mathcal{J}_{N,n,r}$ on the time interval $[0, \Delta]$ with the initial condition $\phi(x) = e_m(x)$ and denote the obtained solution as $\varphi_\alpha(\Delta, x; e_m)$, $\alpha \in \mathcal{J}_{N,n,r}$, $m = 1, \dots, M$. Also, choose a time step size δt to solve the equations in the propagator numerically.

Step 2. Evaluate $\psi_m(0; N, n, M) = (u_0, e_m)$, $m = 1, \dots, M$, where $u_0(x)$ is the initial condition for (2.1), and $q_{\alpha,l,m} = (\varphi_\alpha(\Delta, \cdot; e_l), e_m(\cdot))$, $l, m = 1, \dots, M$.

Step 3. Recursively compute the covariance matrices $Q_{lm}(t_i; N, n, M)$ according to (4.1), and obtain the second moment $E[u_{\Delta,N,n}^M(t_i, x)]^2$ of the approximate solution to (2.1) by (4.2).

We again emphasize that Algorithm 2 for computing moments does not have a statistical error.

Let us discuss the error of Algorithm 2. One can show (see, e.g., [19]) that due to the orthogonality of the random variables $\xi_\alpha^{(i)}$ in the sense that $E\xi_\alpha^{(i)}\xi_\beta^{(j)} = 0$ unless $i = j$ and $\alpha = \beta$, the following equality holds:

$$(4.3) \quad Eu^2(t, x) - Eu_{N,n}^2(t, x) = E[u(t, x) - u_{N,n}(t, x)]^2.$$

Hence the error estimates for approximation of the second moment $E[u(t_i, x)]^2$ by $E[u_{\Delta,N,n}(t_i, x)]^2$ is equal to the errors given in (3.10) and (3.11).

We do not discuss here errors arising from noise truncation and from truncation of the basis $\{e_m(x)\}_{m \geq 1}$.

Remark 4.1. It is not difficult to show that the computational costs of Steps 1 and 2 of Algorithm 2 are proportional to $M^2 \frac{(N+nr)!}{N!(nr)!}$. In general the computational cost of Step 3 over K time steps is proportional to $K M^4 \frac{(N+nr)!}{N!(nr)!}$. Taking this into account together with the error estimates (3.10) and (3.11), it is usually computationally beneficial to choose $n = 1$ and $N = 2$ or 1. The main computational cost of Algorithm 2 is due to the total number of basis functions M (in physical space) required for reaching a satisfactory accuracy. As is well known, for a fixed accuracy the number

M of basis functions $\{e_m\}_{m=1}^M$ is proportional to C^d , where C depends on a choice of the basis and on the problem. If the variance of $u^2(t, x_i)$ is relatively large and the problem considered does not require a very large number of basis functions M , then one expects Algorithm 2 to be computationally more efficient in evaluating second moments than the combination of Algorithm 1 with the Monte Carlo technique.

Algorithm 2's efficiency can often be improved by choosing an appropriate basis $\{e_m\}$ so that the majority of functions $q_{\alpha,l,m}$ are identically zero or negligible and hence can be dropped from computing the covariance matrix $\{Q_{lm}(t_i; N, M)\}_{l,m=1}^M$, significantly decreasing the computational cost of Step 3. For instance, for the periodic passive scalar equation considered in section 6 we choose the Fourier basis $\{e_m\}$. In this case, the number of zero $q_{\alpha,l,m}$ is proportional just to M (the total number of $q_{\alpha,l,m}$ is proportional to M^2) and, consequently, the computational cost of Step 3 (and hence of the whole Algorithm 2) becomes proportional to M^2 instead of the original M^4 . Moreover, computation of the covariance matrix according to (4.1) can be done in parallel. Clearly, the use of reduced-order methods with offline/online strategies [29] can greatly reduce the value of M and hence will make the proposed method very efficient.

Remark 4.2. It is more expensive to compute higher-order moments by a deterministic algorithm analogous to Algorithm 2. Since second moments give us such important, from the physical point of view, characteristics as energy and correlation functions, Algorithm 2 can be a competitive alternative to Monte Carlo-type methods in practical situations.

5. Numerical tests in one dimension. We start (section 5.1) with a description of two one-dimensional test problems used in the numerical tests. Then, for clarity of exposition, we illustrate application of Algorithm 2 to these problems (section 5.2). We present results of numerical tests of Algorithm 2 in section 5.3 and its comparison with some Monte Carlo-type algorithms in section 5.4. In the next section (section 6) we also perform numerical tests with a two-dimensional passive scalar equation.

5.1. Test problems. We consider the following two model problems. The first one is the *stochastic advection-diffusion equation* with periodic boundary conditions written in the Stratonovich form as

$$(5.1) \quad \begin{aligned} du(t, x) &= \epsilon u_{xx}(t, x) dt + \sigma u_x(t, x) \circ dw(t), \quad t > 0, \quad x \in (0, 2\pi), \\ u(0, x) &= \sin(x) \end{aligned}$$

or in the Itô form as

$$du(t, x) = au_{xx}(t, x) dt + \sigma u_x(t, x) dw(t), \quad u(0, x) = \sin(x).$$

Here $w(t)$ is a standard one-dimensional Wiener process, $\sigma > 0$, $\epsilon \geq 0$ are constants, and $a = \epsilon + \sigma^2/2$. The solution of (5.1) is

$$(5.2) \quad u(t, x) = e^{-\epsilon t} \sin(x + \sigma w(t)),$$

and its first and second moments are

$$Eu(t, x) = e^{-at} \sin(x), \quad Eu^2(t, x) = e^{-2\epsilon t} \left(\frac{1}{2} - \frac{1}{2} e^{-2\sigma^2 t} \cos(2x) \right).$$

We note that for $\epsilon = 0$ (5.1) becomes degenerate.

The second model problem is the following Itô *reaction-diffusion equation* with periodic boundary conditions:

$$(5.3) \quad \begin{aligned} du(t, x) &= au_{xx}(t, x) dt + \sigma u(t, x) dw(t), \quad t > 0, x \in (0, 2\pi), \\ u(0, x) &= \sin(x), \end{aligned}$$

where $\sigma > 0$ and $a \geq 0$ are constants. Its solution is

$$(5.4) \quad u(t, x) = \exp\left(-\left(a + \frac{\sigma^2}{2}\right)t + \sigma w(t)\right) \sin(x),$$

and its first and second moments are

$$Eu(t, x) = e^{-at} \sin(x), \quad Eu^2(t, x) = \exp(-2a - \sigma^2)t \sin^2(x).$$

In sections 5.3 and 5.4 we will test Algorithm 2 by evaluating the second moments $Eu^2(t, x)$ of the solutions to (5.1) and (5.3).

5.2. Application of WCE algorithms to the model problem. The problems (5.1) and (5.3) are simpler than the general linear SPDE (2.1) we have considered in the paper, and, consequently, Algorithm 2 applied to them takes a simpler form (see Algorithm 3 below).

We note that when an SPDE has a single Wiener process only, the multi-index α takes the form $\alpha = (\alpha_1, \alpha_2, \dots)$, where α_i are nonnegative integers. For instance, if $|\alpha| = 0$ (i.e., $\alpha = (0, 0, \dots)$), then the corresponding $\xi_\alpha = 1$ (cf. (2.6)). If $|\alpha| = 1$, then the multi-index $\alpha = (0, \dots, 0, 1, 0, \dots)$ with $\alpha_i = 1$ and the other $\alpha_k = 0$, and the corresponding $\xi_\alpha = H_1(\xi_i) = \xi_i = \int_0^t m_i(s) dw(s)$. If $|\alpha| = 2$, then the multi-index is either of the type $\alpha = (0, \dots, 0, 1, 0, \dots, 0, 1, 0, \dots)$ with $\alpha_i = \alpha_j = 1$ and the other $\alpha_k = 0$, and consequently $\xi_\alpha = H_1(\xi_i)H_1(\xi_j) = \int_0^t m_i(s) dw(s) \int_0^t m_j(s) dw(s)$; or $\alpha = (0, \dots, 0, 2, 0, \dots)$ with $\alpha_i = 2$ and the other $\alpha_k = 0$, and consequently $\xi_\alpha = H_2(\xi_i)/\sqrt{2} = \frac{1}{\sqrt{2}}[(\int_0^t m_i(s) dw(s))^2 - 1]$; and so on.

The model problems (5.1) and (5.3) have a single Wiener process, and they possess the following interesting feature. We observe that their solutions (5.2) and (5.4) have the form $u(t, x) = f(t, x, w(t))$, where $f(t, x, y)$ is a smooth function. Consequently, the solutions are expandable in the basis consisting just of $\xi_\alpha = H_k(w(t)/\sqrt{t})/\sqrt{k!} = H_k(\xi_1)/\sqrt{k!}$, $\alpha = (k, 0, \dots, 0)$, $k = 0, 1, \dots$; i.e., we have

$$(5.5) \quad u(t, x) = \sum_{\alpha \in \mathcal{J}} \frac{\varphi_\alpha(t, x)}{\sqrt{\alpha!}} \xi_\alpha = \sum_{N=0}^\infty \sum_{\alpha \in \mathcal{J}_{N,1}} \frac{\varphi_\alpha(t, x)}{\sqrt{\alpha!}} \xi_\alpha = \sum_{k=0}^\infty \frac{\varphi_k(t, x)}{\sqrt{k!}} \eta_k,$$

where $\eta_k = \xi_\alpha$ with $\alpha = (k, 0, \dots, 0)$, $k = 0, 1, \dots$. Hence

$$(5.6) \quad u_{N,1}(t, x) = :u_N(t, x) = \sum_{k=0}^N \frac{\varphi_k(t, x)}{\sqrt{k!}} \eta_k,$$

which corresponds to setting $n = 1$ in (3.1). It is not difficult to show (see also the discussion on error estimates after Algorithm 2 in section 4) that applying Algorithm 2 to the model problems (5.1) and (5.3) is more accurate than in general cases of (2.1) (cf. (3.10) and (3.11) and also (4.3)):

$$(5.7) \quad \|Eu^2(t, \cdot) - Eu_{\Delta,N}^2(t, \cdot)\|_{L^2} \leq C \frac{(C\Delta)^N}{(N+1)!}$$

for all sufficiently small $\Delta > 0$ and a constant $C > 0$ independent of Δ and N (as before, here we neglected errors arising from truncation of the basis $\{e_m\}$).

For the problems (5.1) and (5.3), the propagator (3.12) takes the form (recall that here the multi-index α degenerates to $\alpha = (k, 0, \dots, 0)$, $k = 0, 1, \dots$)

$$(5.8) \quad \begin{aligned} \partial_t \varphi_0 &= a \partial_{xx}^2 \varphi_0, \quad \varphi_0(0, x; \phi) = \phi(x), \\ \partial_t \varphi_k &= a \partial_{xx}^2 \varphi_k + \frac{1}{\sqrt{\Delta}} \sigma k \partial_x \varphi_{k-1}, \quad \varphi_k(0, x; 0) = 0, \quad k > 0, \end{aligned}$$

and

$$(5.9) \quad \begin{aligned} \partial_t \varphi_0 &= a \partial_{xx}^2 \varphi_0, \quad \varphi_0(0, x; \phi) = \phi(x), \\ \partial_t \varphi_k &= a \partial_{xx}^2 \varphi_k + \frac{1}{\sqrt{\Delta}} \sigma k \varphi_{k-1}, \quad \varphi_k(0, x; 0) = 0, \quad k > 0, \end{aligned}$$

respectively. We solve these propagators numerically using the Fourier collocation method with M nodes in physical space and the Crank–Nicolson time discretization with step δt in time. Denote by $L_m(x)$, $m = 1, \dots, M$, the m th Lagrangian trigonometric polynomials using M Fourier collocation nodes; i.e., $L_m(x)$ are M th-order trigonometric polynomials satisfying $L_m(x_l) = \delta_{m,l}$ and $x_l = \frac{2\pi}{M}(l-1)$, $l = 1, \dots, M$. Now, for completeness, we formulate the realization of Algorithm 2 in the case of the model problems.

ALGORITHM 3. For given values of the model parameters a and σ , choose the algorithm parameters: a number of Fourier collocation nodes M , a time step δt for solving the propagator (5.8) (or (5.9)), a time step Δ , and the number of Hermite polynomials N .

Step 1. Solve the propagator (5.8) (or (5.9)) on the time interval $[0, \Delta]$ with the initial condition $\phi(x) = L_k(x)$ using the Fourier collocation method with M nodes in physical space and the Crank–Nicolson scheme with step δt in time, and denote the obtained numerical approximation of $\varphi_k(\Delta, x_l; L_m)$ as $\varphi_k^{M, \delta t}(\Delta, x_l; L_m)$, $l, m = 1, \dots, M$, $k = 1, \dots, N$.

Step 2. Recursively compute the covariance matrices

$$Q_{lm}(t_i; N, M) := E[u_{\Delta, N}^{M, \delta t}(t_i, x_l) u_{\Delta, N}^{M, \delta t}(t_i, x_m)] \quad \text{at} \quad t_i = i\Delta, \quad i = 0, \dots, K,$$

of the approximate solution to (5.1) (or (5.3)):

$$\begin{aligned} Q_{lm}(0; N, M) &= u_0(x_l) u_0(x_m), \quad l, m = 1, \dots, M, \\ Q_{lm}(t_i; N, M) &= \sum_{k=1}^N \sum_{q=1}^M \sum_{r=1}^M \frac{1}{k!} Q_{qr}(t_{i-1}; N, M) \varphi_k^{M, \delta t} \\ &\quad (\Delta, x_l; L_q) \varphi_k^{M, \delta t}(\Delta, x_m; L_r), \\ &\quad l, m = 1, \dots, M, \quad i = 1, \dots, K, \end{aligned}$$

where $u_0(x)$ is the initial condition of (5.1) (or (5.3)). In particular, we obtain the second moment of the approximate solution to (5.1) (or (5.3)):

$$E[u_{\Delta, N}^{M, \delta t}(t_i, x_j)]^2 = Q_{jj}(t_i; N, M), \quad j = 1, \dots, M, \quad i = 1, \dots, K.$$

We note that Algorithm 3 has four errors: (1) an error due to time discretization of the SPDE, which is controlled by Δ ; (2) the truncation error of the one-step WCE, which is controlled by N ; (3) an error due to the truncation of the spatial basis $\{e_m\}$, which is controlled by M ; and (4) the numerical integration error in solving the propagator. The last one, in its turn, consists of the error due to space discretization, which is controlled by M , and of the error due to time discretization, which is controlled by δt .

Remark 5.1. To approximate the solution of (5.1) (or (5.3)), one can use the truncated WCE $u_N(t, x)$ from (5.6) and, in particular, evaluate the second moment $Eu^2(t, x)$ as

$$(5.10) \quad Eu^2(t, x) \approx Eu_N^2(t, x) = \sum_{k=0}^N \frac{\varphi_k^2(t, x)}{k!} \approx \sum_{k=0}^N \frac{[\varphi_k^{M, \delta t}(t, x)]^2}{k!},$$

where $\varphi_0(t, x) = \varphi_0(t, x; u_0(x))$ and $\varphi_k(t, x) = \varphi_k(t, x; 0)$, $k > 0$, are solutions of the propagator (5.8) (or (5.9)) and $\varphi_k^{M, \delta t}(t, x)$ are their numerical approximations obtained, e.g., using the Fourier collocation method with M nodes in physical space and the Crank–Nicolson scheme with step δt in time. The approximation (5.10) can be viewed as a one-step approximation corresponding to Algorithm 3, i.e., the first step of Algorithm 3 with $\Delta = t$, and its error is estimated as

$$\|Eu^2(t, \cdot) - Eu_N^2(t, \cdot)\|_{L^2} \leq Ce^{Ct} \frac{(Ct)^{N+1}}{(N+1)!}.$$

We see that this error grows exponentially with t , which was confirmed by our numerical tests with (5.1) (not presented here). To reach a satisfactory accuracy of the approximation (5.10) for a fixed t , one has to take a sufficiently large N , which is computationally expensive (see also Remark 4.1) even in the case of moderate values of t . In contrast, we demonstrate (see next section) that the error of Algorithm 3 grows linearly with time and it is relatively small even for $N = 1$.

5.3. Numerical results. In this section we present some results of our numerical tests of Algorithm 3 on the two model problems (5.1) and (5.3).

In approximating the propagators (5.8) and (5.9) we choose a sufficiently large number of Fourier collocation nodes M and a sufficiently small time step δt so that errors of numerical solutions to the propagators have a negligible influence on the overall accuracy of Algorithm 3 in our simulations. In all the numerical tests it was sufficient to take $M = 20$; this choice of M was tested by running control tests with $M = 80$.

We measure numerical errors using the norms

$$\rho_2(t) = \left(\frac{2\pi}{M} \sum_{m=1}^M \left(E \left[u_{\Delta, N}^{M, \delta t}(t, x_m) \right]^2 - Eu^2(t, x_m) \right)^2 \right)^{1/2}$$

and

$$\rho_\infty(t) = \max_{1 \leq m \leq M} \left| E \left[u_{\Delta, N}^{M, \delta t}(t, x_m) \right]^2 - Eu^2(t, x_m) \right|.$$

The results of our tests on the model problem (5.1) in the degenerate case (i.e., $\epsilon = 0$) and in the nondegenerate case (i.e., $\epsilon > 0$) are presented in Tables 5.1 and 5.2,

respectively. Table 5.3 corresponds to the tests with the second model problem (5.3). Numerical tests with values of the parameters other than those used for Tables 5.1–5.3 were also performed, and they gave similar results.

Analyzing the results in Tables 5.1, 5.2, and 5.3, we observe the convergence order of Δ^N for a fixed N in all the tests, which confirms our theoretical prediction (5.7). We also run other cases (not presented here) to confirm the conclusion from section 5.2 that the number n of random variables ξ_k used per step does not influence the accuracy of Algorithm 2 in the case of the model problems (5.1) and (5.3).

In Figure 5.1 we demonstrate dependence of the relative numerical error

$$\rho_2^r(t) = \frac{\rho_2(t)}{\|Eu^2(t, \cdot)\|_{L^2}}$$

on integration time. These results were obtained in the degenerate case of the problem (5.1), but similar behavior of errors was observed in our tests with other parameters as well. One can conclude from Figure 5.1 that (after an initial fast growth) the error

TABLE 5.1

Model (5.1)—performance of Algorithm 3. The parameters of the model (5.1) are $\sigma = 1$, $\epsilon = 0$, and the time is $t = 10$. In Algorithm 3 we take $M = 20$.

N	Δ	δt	$\rho_2(10)$	$\rho_\infty(10)$
1	0.1	1×10^{-3}	4.69×10^{-1}	1.87×10^{-1}
	0.01	1×10^{-4}	6.07×10^{-2}	2.42×10^{-2}
	0.001	1×10^{-5}	6.25×10^{-3}	2.49×10^{-3}
2	0.1	1×10^{-3}	1.92×10^{-2}	7.67×10^{-3}
	0.01	1×10^{-4}	2.07×10^{-4}	8.27×10^{-5}
	0.001	1×10^{-5}	2.09×10^{-6}	8.33×10^{-7}
3	0.1	1×10^{-3}	4.82×10^{-4}	1.99×10^{-4}
	0.01	1×10^{-4}	5.16×10^{-7}	2.06×10^{-7}
	0.001	1×10^{-5}	3.37×10^{-10}	1.81×10^{-10}
4	0.1	1×10^{-3}	9.36×10^{-6}	3.73×10^{-6}
	0.01	1×10^{-5}	9.35×10^{-10}	4.17×10^{-10}

TABLE 5.2

Model (5.1)—performance of Algorithm 3. The parameters of the model (5.1) are $\sigma = 1$, $\epsilon = 0.01$, and the time is $t = 10$. In Algorithm 3 we take $M = 20$.

N	Δ	δt	$\rho_2(10)$	$\rho_\infty(10)$
1	0.1	1×10^{-3}	3.84×10^{-1}	1.53×10^{-1}
	0.01	1×10^{-4}	4.97×10^{-2}	1.98×10^{-2}
	0.001	1×10^{-4}	5.11×10^{-3}	2.04×10^{-3}
2	0.1	1×10^{-3}	1.58×10^{-2}	6.28×10^{-3}
	0.01	1×10^{-4}	1.70×10^{-4}	6.77×10^{-5}
	0.001	1×10^{-4}	1.72×10^{-6}	6.88×10^{-7}
3	0.1	1×10^{-3}	3.95×10^{-4}	1.57×10^{-4}
	0.01	1×10^{-4}	4.22×10^{-7}	1.68×10^{-7}
	0.001	1×10^{-5}	3.65×10^{-10}	2.01×10^{-10}
4	0.1	1×10^{-3}	7.67×10^{-6}	3.06×10^{-6}
	0.01	1×10^{-5}	8.39×10^{-10}	3.90×10^{-10}

TABLE 5.3

Model (5.3)—performance of Algorithm 3. The parameters of the model (5.3) are $\sigma = 1$, $a = 0.5$, and the time is $t = 10$. In Algorithm 3 we take $M = 20$.

N	Δ	δt	$\rho_2(10)$	$\rho_\infty(10)$
1	0.1	1×10^{-3}	5.75×10^{-1}	3.74×10^{-1}
	0.01	1×10^{-4}	7.44×10^{-2}	4.85×10^{-2}
	0.001	1×10^{-4}	7.65×10^{-3}	4.98×10^{-3}
2	0.1	1×10^{-3}	2.36×10^{-2}	1.53×10^{-2}
	0.01	1×10^{-4}	2.54×10^{-4}	1.65×10^{-4}
	0.001	1×10^{-4}	2.58×10^{-6}	1.68×10^{-6}
3	0.1	1×10^{-3}	5.90×10^{-4}	3.85×10^{-4}
	0.01	1×10^{-4}	6.32×10^{-7}	4.12×10^{-7}

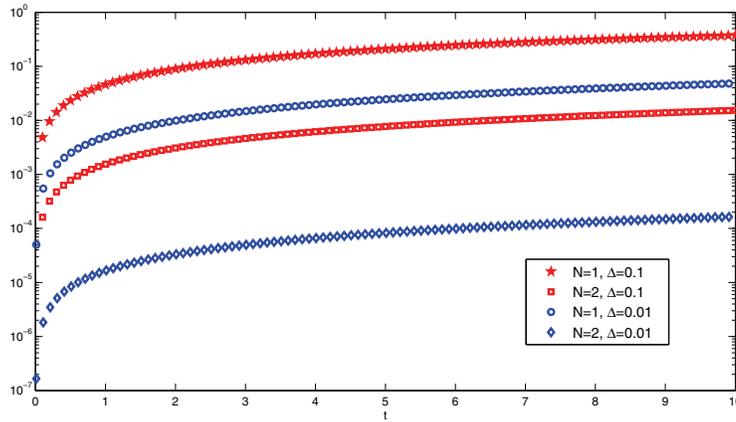


FIG. 5.1. Dependence of the relative numerical error $\rho_2^r(t)$ on integration time. Model (5.1) is simulated by Algorithm 3 with $M = 20$ and $\delta t = \Delta/100$ and various Δ and N . The parameters of (5.1) are $\sigma = 1$ and $\epsilon = 0$.

grows linearly with integration time. This is a remarkable feature of the proposed WCE-based algorithm since it implies that the algorithm can be used for long time integration of SPDEs.

5.4. Comparison of the WCE algorithm and Monte Carlo-type algorithms. As discussed in the introduction, there are other approaches to solving SPDEs, which are usually complemented by the Monte Carlo technique when one is interested in computing moments of SPDE solutions. In this section, using the problem (5.1), we compare the performances of Algorithm 3 and two Monte Carlo-type algorithms, one of which is based on the method of characteristics [26] and another on the Fourier transform of the linear SPDE with subsequent simulation of SDEs and application of the Monte Carlo technique.

The solution of (5.1) with $\epsilon = 0$ (the degenerate case) can be represented using the method of characteristics [28]:

$$(5.11) \quad u(t, x) = \sin(X_{t,x}(0)),$$

where $X_{t,x}(s)$, $0 \leq s \leq t$, is the solution of the system of backward characteristics

$$(5.12) \quad dX_{t,x}(s) = \sigma \overleftarrow{dw}(s), \quad X_{t,x}(t) = x.$$

The notation $\overleftarrow{dw}(s)$ means backward Itô integral (see, e.g., [28]). It follows from (5.12) that $X_{t,x}(0)$ has the same probability distribution as $x + \sigma\sqrt{t}\zeta$, where ζ is a standard Gaussian random variable (i.e., $\zeta \sim \mathcal{N}(0, 1)$). Since we are interested only in computing statistical moments, it is assumed, without loss of generality, that

$$(5.13) \quad X_{t,x}(0) = x + \sigma\sqrt{t}\zeta.$$

Then we can estimate the second moment $m_2(t, x) := Eu^2(t, x)$ as

$$(5.14) \quad m_2(t, x) \doteq \hat{m}_2(t, x) = \frac{1}{L} \sum_{l=1}^L \sin^2(x + \sigma\sqrt{t}\zeta^{(l)}),$$

where $\zeta^{(l)}$, $l = 1, \dots, L$, are independent and identically distributed (i.i.d.) standard Gaussian random variables. The estimate \hat{m}_2 for m_2 is unbiased, and, hence, the numerical procedure for finding m_2 based on (5.14) has only the Monte Carlo (i.e., statistical) error which, as usual, can be quantified via half of the length of the 95% confidence interval:

$$\rho_{MC}(t, x) = 2 \frac{\sqrt{\text{Var}(\sin^2(x + \sigma\sqrt{t}\zeta))}}{\sqrt{L}}.$$

Table 5.4 gives the statistical error for $\hat{m}_2(t, x)$ from (5.14) (recall that there is no space or time discretization error in this algorithm), which is computed as

$$(5.15) \quad 2 \cdot \max_j \frac{\sqrt{\frac{1}{L} \sum_{l=1}^L \sin^4(x_j + \sigma\sqrt{t}\zeta^{(l)}) - [\hat{m}_2(t, x_j)]^2}}{\sqrt{L}},$$

where the set of x_j is the same as the one used for producing the results of Table 5.5 by Algorithm 3 and $\zeta^{(l)}$ are as in (5.14). All the tests were run using MATLAB R2007b on a Macintosh desktop computer with Intel Xeon CPU E5462 (quad-core, 2.80 GHz). Every effort was made to program and execute the different algorithms as much as possible in an identical way. The cost of simulation due to (5.14) is directly proportional to L . The slower time increase for smaller L in Table 5.4 is due to inclusion of the initialization time of the computer program in the time measurement.

TABLE 5.4

Model (5.1)—performance of the method (5.14). The parameters of the model (5.1) are $\sigma = 1$, $\epsilon = 0$, and the time is $t = 10$. The statistical error is computed according to (5.15).

L	Statistical error	CPU time (sec.)
10^2	8.87×10^{-2}	6×10^{-3}
10^4	7.40×10^{-3}	6.7×10^{-2}
10^6	7.09×10^{-4}	7.4×10^0
10^8	7.07×10^{-5}	7.4×10^2
10^{10}	7.07×10^{-6}	7.3×10^4

TABLE 5.5

Model (5.1)—performance of Algorithm 3. The parameters of the model (5.1) are $\sigma = 1$, $\epsilon = 0$, and the time is $t = 10$. The parameters of Algorithm 3 are $\Delta = 0.1$, $M = 20$, $\delta t = 0.001$.

N	$\rho_\infty(10)$	CPU time (sec.)
1	1.87×10^{-1}	5.7×10^0
2	7.67×10^{-3}	8.1×10^0
3	1.99×10^{-4}	1.1×10^1
4	3.73×10^{-6}	1.3×10^1

In Table 5.5 we repeat some of the results already presented in Table 5.1, which are now also accompanied by CPU time for comparison.

Comparing the results in Tables 5.4 and 5.5, we conclude that when one sets a relatively large error tolerance level the estimate $\hat{m}_2(t, x)$ from (5.14) is computationally more efficient than Algorithm 3 but that Algorithm 3 has lower costs in reaching a higher accuracy (errors of order equal to or smaller than 10^{-6}). We note that variance reduction techniques (see, e.g., [25, 26] and the references therein) can be used in order to reduce the Monte Carlo error. But the aim here is to give a comparison of computational costs for the WCE-based algorithm and direct Monte Carlo methods having in mind that for complex stochastic problems it is usually rather difficult to reduce variance efficiently.

Let us now use the problem (5.1) with $\epsilon = 0$ for comparison of Algorithm 3 with another approach exploiting the Monte Carlo technique. One can represent the solution of this periodic problem via the Fourier transform

$$(5.16) \quad u(t, x) = \sum_{k \in \mathbb{Z}} e^{ikx} u_k(t)$$

with $u_k(t)$, $t \geq 0$, $k \in \mathbb{Z}$, satisfying the system of SDEs

$$(5.17) \quad du_k(t) = -k^2 \frac{1}{2} \sigma^2 u_k(t) dt + ik \sigma u_k(t) dw(t), \quad \operatorname{Re} u_k(0) = 0, \quad \operatorname{Im} u_k(0) = \frac{1}{2} (\delta_{1k} - \delta_{-1k}).$$

Noting that here $u_k(t) \equiv 0$ for all $|k| \neq 1$ and rewriting (5.16)–(5.17) in the trigonometric form, we get

$$(5.18) \quad u(t, x) = u^c(t) \cos x + u^s(t) \sin x,$$

where

$$(5.19) \quad \begin{aligned} du^c(t) &= -\frac{1}{2} \sigma^2 u^c(t) dt + \sigma u^s(t) dw(t), & u^c(0) &= 0, \\ du^s(t) &= -\frac{1}{2} \sigma^2 u^s(t) dt - \sigma u^c(t) dw(t), & u^s(0) &= 1. \end{aligned}$$

The system (5.19) is a Hamiltonian system with multiplicative noise (see, e.g., [24, 25]). It is known [24, 25] that symplectic integrators have advantages in comparison with usual numerical methods in long time simulations of stochastic Hamiltonian systems. An example of a symplectic method is the midpoint scheme, which in application to (5.19) takes the form

$$(5.20) \quad \begin{aligned} \bar{u}^c(t_{k+1}) &= \bar{u}^c(t_k) + \frac{\sigma}{2} (\bar{u}^s(t_k) + \bar{u}^s(t_{k+1})) \sqrt{\Delta t} \zeta_{k+1}, & u^c(0) &= 0, \\ \bar{u}^s(t_{k+1}) &= \bar{u}^s(t_k) - \frac{\sigma}{2} (\bar{u}^c(t_k) + \bar{u}^c(t_{k+1})) \sqrt{\Delta t} \zeta_{k+1}, & u^s(0) &= 1, \end{aligned}$$

where ζ_k are i.i.d. standard Gaussian random variables and $\Delta t > 0$ is a time step. The scheme (5.20) converges with the mean-square order 1/2 and weak order 1 [25]. It is implicit, but (5.20) can be resolved analytically since we are dealing with the linear system here. One can recognize that (5.19) is a Kubo oscillator. A number of numerical tests with symplectic and nonsymplectic integrators are done on a Kubo oscillator in [24, 25].

Using (5.18) and (5.20), we evaluate the second moment of the solution to (5.1) with $\epsilon = 0$ as

$$(5.21) \quad m_2(t_k, x) := Eu^2(t_k, x) \doteq E[\bar{u}^c(t_k) \cos x + \bar{u}^s(t) \sin x]^2 \\ \doteq \hat{m}_2(t_k, x) = \frac{1}{L} \sum_{l=1}^L \left[\bar{u}^{c,(l)}(t_k) \cos x + \bar{u}^{s,(l)}(t_k) \sin x \right]^2,$$

where $\bar{u}^{c,(l)}(t_k)$, $\bar{u}^{s,(l)}(t_k)$ are independent realizations of the random variables $\bar{u}^c(t_k)$, $\bar{u}^s(t_k)$.

The estimate $\hat{m}_2(t_k, x)$ from (5.21) has two errors: the time discretization error due to the approximation of (5.19) by (5.20) and the Monte Carlo error. The errors presented in Table 5.6 are computed as $\max_j [\hat{m}_2(t_k, x_j) - Eu^2(t_k, x_j)]$ and are given together with the 95% confidence interval.

TABLE 5.6

Model (5.1)—performance of the method (5.21). The parameters of the model (5.1) are $\sigma = 1$, $\epsilon = 0$, and the time is $t = 10$.

Δt	L	Error	CPU time (sec.)
0.1	10^4	$8.06 \times 10^{-3} \pm 7.09 \times 10^{-3}$	4.72×10^{-1}
0.01	10^4	$6.55 \times 10^{-4} \pm 7.08 \times 10^{-4}$	3.90×10^2
0.001	10^6	$8.81 \times 10^{-5} \pm 7.07 \times 10^{-5}$	3.81×10^5

Comparing the results in Tables 5.6 and 5.5, we come to the same conclusion as in our first comparison test that Algorithm 3 is computationally more efficient than the Monte Carlo-based algorithms in reaching a higher accuracy.

6. Numerical tests with passive scalar equation. A prominent example of the stochastic advection-diffusion equation (2.1)–(2.2) is a passive scalar equation, which is motivated by the study of the turbulent transport problem (see [7, 16, 18] and the references therein). Here we perform numerical tests on the *two-dimensional* ($d = 2$) passive scalar equation with periodic boundary conditions:

$$(6.1) \quad du(t) + \sum_{k=1}^{\infty} \sum_{i=1}^d \sigma_k^i(x) D_i u \circ dw_k(t) = 0, \\ u(t, x^1 + \ell, x^2) = u(t, x^1, x^2 + \ell) = u(t, x), \quad t > 0, \quad x \in (0, \ell)^2, \\ u(0, x) = u_0(x), \quad x \in (0, \ell)^2,$$

where \circ indicates the Stratonovich version of stochastic integration, $\ell > 0$, the initial condition $u_0(x)$ is a periodic function with the period $(0, \ell)^2$, and $\sigma_k^i(x)$ are divergence-free periodic functions with the period $(0, \ell)^2$:

$$(6.2) \quad \operatorname{div} \sigma_k = 0.$$

In (6.1) we take a combination of such $\sigma_k(x)$ so that the corresponding spatial covariance C is symmetric and stationary: $C(x - y) = \sum_{k=1}^{\infty} \lambda_k \sigma_k(x) \sigma_k^T(y)$, where λ_k are some nonnegative numbers. Namely, we consider

$$(6.3) \quad C(x - y) = \sum_{k=1}^{\infty} \lambda_k C(x - y; n_k, m_k),$$

where n_k, m_k is a sequence of positive integers, and

$$C(x - y; n, m) = \cos(2\pi (n[x^1 - y^1] + m[x^2 - y^2]) / \ell) \begin{bmatrix} m^2 & -nm \\ -nm & n^2 \end{bmatrix},$$

which can be decomposed as

$$\begin{aligned} C(x - y; n, m) &= \cos(2\pi [nx^1 + mx^2] / \ell) \begin{bmatrix} -m \\ n \end{bmatrix} \cos(2\pi [ny^1 + my^2] / \ell) \begin{bmatrix} -m & n \end{bmatrix} \\ &+ \sin(2\pi [nx^1 + mx^2] / \ell) \begin{bmatrix} -m \\ n \end{bmatrix} \sin(2\pi [ny^1 + my^2] / \ell) \begin{bmatrix} -m & n \end{bmatrix}. \end{aligned}$$

Hence, $\{\sigma_k(x)\}_{k \geq 1}$ in (6.1) is an appropriate combination of vector functions of the form

$$\cos(2\pi [nx^1 + mx^2] / \ell) \begin{bmatrix} -m \\ n \end{bmatrix} \quad \text{and} \quad \sin(2\pi [nx^1 + mx^2] / \ell) \begin{bmatrix} -m \\ n \end{bmatrix}.$$

We rewrite (6.1) in the Itô form

$$(6.4) \quad \begin{aligned} du(t) + \frac{1}{2} \sum_{i,j=1}^d C_{ij}(0) D_i D_j u dt + \sum_{k=1}^{\infty} \sum_{i=1}^d \sigma_k^i(x) D_i u dw_k(t) &= 0, \\ u(t, x^1 + \ell, x^2) = u(t, x^1, x^2 + \ell) = u(t, x), \quad t > 0, x \in (0, \ell)^2, \\ u(0, x) = u_0(x), \quad x \in (0, \ell)^2. \end{aligned}$$

Below we present results of numerical tests of Algorithm 2 applied to (6.4) and its comparison with the Monte Carlo-type algorithm based on the method of characteristics from [26]. In the tests we simulated the L_2 -norm of the second moment of the SPDE solution

$$(6.5) \quad \|Eu^2(T, \cdot)\|_{L^2} = \left[\int_{[0, \ell]^2} [Eu^2(T, x)]^2 dx \right]^{1/2}.$$

We considered the particular case of (6.1), (6.3) with $\ell = 2\pi$, the initial condition

$$(6.6) \quad u_0(x) = \sin(2x^1) \sin(x^2),$$

and two noise terms

$$(6.7) \quad \begin{aligned} \sigma_1(x) = \cos(x_1 + x_2) \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad \sigma_2(x) = \sin(x_1 + x_2) \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \\ \sigma_k(x) = 0 \quad \text{for } k > 2. \end{aligned}$$

This example satisfies the so-called commutativity condition (see [25, 15]), and we expect that the error estimate (3.10) will hold in this case, which is confirmed in the tests (a rigorous numerical analysis will be considered elsewhere).

In Algorithm 2 we solve the propagator (3.12) corresponding to the SPDE (6.4) using fourth-order explicit Runge–Kutta with step δt in time and the Fourier spectral method with M modes in physical space. We noted in Remark 4.1 that in general the computational cost of Algorithm 2 is proportional to M^4 but with an appropriate choice of basis functions; this cost can be considerably reduced. Indeed, the Fourier basis is natural for the problem (6.4) and the use of this basis reduces the computational cost to being proportional to M^2 . This significant reduction is based on the following observation. Since we consider a finite number of noises with periodic $\sigma_k(x)$ and $\varphi_\alpha(\Delta, x; e_l)$ to be the solution of the propagator (3.12) with the initial condition equal to a single basis function $e_l(x)$, $\varphi_\alpha(\Delta, x; e_l)$ is expandable in a finite number of periodic functions $e_k(x)$ and this number does not depend on M . Hence for fixed α and l the number of nonzero $q_{\alpha,l,m} = (\varphi_\alpha(\Delta, \cdot; e_l), e_m(\cdot))$ is finite. Therefore, the overall number of nonzero $q_{\alpha,l,m}$ is proportional to M instead of M^2 . This was tested and confirmed in our tests. We use the above fact in our computer realization of Algorithm 2 and reduce the computational cost of obtaining a single entry of the matrix $Q_{l,m}$ from the order of $O(M^2)$ to order $O(1)$. Hence, computational costs of Step 3 (and hence all of Algorithm 2) become proportional to M^2 instead of the original M^4 .

We do not have an exact solution of the problem (6.1), and hence we need a reference solution. To this end, the L_2 -norm of the second moment of the SPDE solution at $T = 1$ was computed by Algorithm 2 with parameters $N = 2$, $n = 1$, $M = 900$ (i.e., 30 basis functions in each space direction), $\delta t = 1 \times 10^{-5}$, and $\Delta = 1 \times 10^{-4}$, which is equal to 1.57976 (5 d.p.). This result was also verified by the Monte Carlo-type method described below with $\Delta t = 1 \times 10^{-3}$, $M_s = 10$, and $L = 8 \times 10^7$, which gave $1.579777 \pm 7.6 \times 10^{-5}$, where \pm reflects the Monte Carlo error only.

For Algorithm 2, we measure the error of computing the L_2 -norm of the second moment of the SPDE solution as follows:

$$\rho(T) = \|Eu_{ref}^2(T, \cdot)\|_{l^2} - \left\| E \left[u_{\Delta, N}^{M, \delta t}(T, \cdot) \right]^2 \right\|_{l^2},$$

where $\|v(\cdot)\|_{l^2} = \frac{\ell}{M_s} (\sum_{i,j=1}^{M_s} v^2(x_i^1, x_j^2))^{1/2}$, $x_i^1 = x_i^2 = (i-1)\ell/M_s$, $i = 1, \dots, M_s$, and $E u_{ref}^2(T, \cdot)$ is the reference solution computed as explained above. The results demonstrating second-order convergence (see (3.10) and the discussion after Algorithm 2) are given in Table 6.1. We note that we also did some control tests with $\delta t = 1 \times 10^{-5}$ and $M = 1600$ which showed that the errors presented in this table are not essentially influenced by the errors caused by the choice of $\delta t = 1 \times 10^{-4}$ and cut-off of the basis at $M = 900$.

TABLE 6.1

Passive scalar equation (6.4)—performance of Algorithm 2. The parameters of Algorithm 2 are $N = 2$, $n = 1$, $M = 900$, $\delta t = 1 \times 10^{-4}$.

Δ	$\rho(1)$
0.05	0.1539
0.02	0.0326
0.01	0.0089
0.005	0.0023
0.0025	0.0006

Let us now describe the Monte Carlo-type algorithm based on the method of characteristics (see further details in [26]) with which here we compare the performance of Algorithm 2. The solution $u(t, x)$ of (6.1) has the following (conditional) probabilistic representation (see [28, 18]):

$$(6.8) \quad u(t, x) = u_0(X_{t,x}(0)) \quad \text{a.s.},$$

where $X_{t,x}(s)$, $0 \leq s \leq t$, is the solution of the system of (backward) characteristics

$$(6.9) \quad -dX = \sum_k \sigma_k(X) \overleftarrow{dw}_k(s), \quad X(t) = x.$$

Due to (6.2) and (see [18])

$$(6.10) \quad \sum_k \frac{\partial \sigma_k}{\partial x} \sigma_k = 0,$$

the phase flow of (6.9) preserves phase volume (see, e.g., [25, eq. (5.5), p. 247]). We also recall that the Itô and Stratonovich forms of (6.9) coincide. As is known [25], it is beneficial to approximate (6.9) using phase volume preserving schemes, e.g., by the midpoint method [25, Chap. 4], which for (6.9) takes the following form (here we exploited that the Itô and Stratonovich forms of (6.9) coincide): for an integer $m \geq 1$,

$$(6.11) \quad \begin{aligned} X_m &= x, \\ X_l &= X_{l+1} + \sum_k \sigma_k \left(\frac{X_l + X_{l+1}}{2} \right) (\zeta_k^{\Delta t})_l \sqrt{\Delta t}, \quad l = n-1, \dots, 0, \end{aligned}$$

where $(\zeta_k^{\Delta t})_l$ are, e.g., i.i.d. random variables with the law

$$(6.12) \quad \zeta_k^{\Delta t} = \begin{cases} \xi_k, & |\xi_k| \leq A_{\Delta t}, \\ A_{\Delta t}, & \xi_k > A_{\Delta t}, \\ -A_{\Delta t}, & \xi_k < -A_{\Delta t}, \end{cases}$$

ξ_k are independent $\mathcal{N}(0, 1)$ -distributed random variables, and $A_{\Delta t} = \sqrt{2c|\ln \Delta t|}$, $c \geq 1$. Its weak order is equal to one. This scheme requires solving the two-dimensional nonlinear equation at each step. To solve it, we used the fixed-point method with the level of tolerance 10^{-13} , and in our example two fixed-point iterations were sufficient to reach this accuracy. Using $\bar{X}_{t,x}(0) = X_0$ obtained by (6.11) with $\Delta t = T/m$, we simulate the L_2 -norm of the second moment of the SPDE solution as follows:

$$(6.13) \quad \begin{aligned} \|Eu^2(T, \cdot)\|_{L^2} &= \left[\int_{[0, \ell]^2} [Eu^2(T, x)]^2 dx \right]^{1/2} \approx \|Eu^2(T, \cdot)\|_{l^2} \\ &= \frac{\ell}{M_s} \left[\sum_{i,j=1}^{M_s} [Eu_0^2(X_{T,x_i^1, x_j^2}(0))]^2 \right]^{1/2} \\ &\approx \frac{\ell}{M_s} \left[\sum_{i,j=1}^{M_s} [Eu_0^2(\bar{X}_{T,x_i^1, x_j^2}(0))]^2 \right]^{1/2} \\ &\approx \frac{\ell}{M_s} \left[\sum_{i,j=1}^{M_s} \left[\frac{1}{L} \sum_{l=1}^L u_0^2(\bar{X}_{T,x_i^1, x_j^2}^{(l)}(0)) \right]^2 \right]^{1/2}, \end{aligned}$$

where $x_i^1 = x_i^2 = (i-1)\ell/M_s$, $i = 1, \dots, M_s$; $\bar{X}_{t,x_i^1,x_j^2}^{(l)}(0)$ are independent realizations of the random variables $\bar{X}_{t,x_i^1,x_j^2}(0)$. The approximation in (6.13) has three errors: (i) the error of discretization of the integral of the space domain $[0, \ell]^2$, which is negligible in our example even for $M_s = 10$; (ii) the error of numerical integration due to replacement of $X_{t,x_i^1,x_j^2}(0)$ by $\bar{X}_{t,x_i^1,x_j^2}(0)$; (iii) the Monte Carlo error, which is measured analogously to how it was done in section 5.4. We note that it is possible to reduce the variance of the estimator on the right-hand side of (6.13), but we do not consider it here. It is interesting that the midpoint scheme used to simulate $\bar{X}_{t,x_i^1,x_j^2}(0)$ gave very accurate results even with relatively large time steps.

We compare Algorithm 2 and the Monte Carlo algorithm (6.13) by simulating the example (6.1), (6.6), (6.7) at $T = 1$. In these comparison tests, MATLAB R2010b was used for each test on a single core of two Intel Xeon 5540 (2.53 GHz) quad-core Nehalem processors. From Tables 6.2 and 6.3, we can draw the same conclusion as in one dimension that for lower accuracy the Monte Carlo algorithm (6.13) outperforms Algorithm 2 but that Algorithm 2 is more efficient for obtaining higher accuracy.

TABLE 6.2

Passive scalar equation (6.4)—performance of Algorithm 2. The parameters of Algorithm 2 are $N = 2$, $n = 1$, $M = 900$, $\delta t = 1 \times 10^{-4}$.

Δ	$\rho(1)$	CPU time
1×10^{-2}	8.89×10^{-3}	3.7×10^4 (sec.)
1×10^{-3}	1.20×10^{-4}	3.2×10^5 (sec.)
5×10^{-4}	3.73×10^{-5}	1.8×10^2 (hours)

TABLE 6.3

Passive scalar equation (6.4)—performance of the algorithm (6.11). The parameter is $M = 100$.

Δt	L	Error	CPU time
2×10^{-1}	2.5×10^4	$4.68 \times 10^{-3} \pm 4.38 \times 10^{-3}$	1.2×10^1 (sec.)
1×10^{-2}	4×10^7	$1.46 \times 10^{-4} \pm 1.08 \times 10^{-4}$	3.5×10^5 (sec.)
1×10^{-3}	4×10^8	$\sim \times 10^{-5} \pm 3.03 \times 10^{-5}$	9.7×10^3 (hours) ¹

7. Summary. We have developed a multistage WCE method for advection-diffusion-reaction equations with multiplicative noise, which form a wide class of linear parabolic SPDEs. We complemented this method by a deterministic algorithm for computing second moments of the SPDE solutions without any use of the Monte Carlo technique. Our numerical tests demonstrated that the proposed WCE-based deterministic algorithm can be more efficient than Monte Carlo-type methods in obtaining results of higher accuracy, scaling as Δ^N , where Δ is the time step of the “online” integration and N is the order of Wiener chaos. We have also found that for obtaining results of lower accuracy, Monte Carlo-type methods outperform the deterministic algorithm for computing moments even in the one-dimensional case. The proposed WCE-based algorithm is conceptually different from Monte Carlo-type methods, and thus it can be used for independent verification of results obtained by Monte Carlo solvers. The efficiency of the algorithm can be greatly improved if it is combined with

¹This is an estimated time according to the tests with smaller Δt , L and with $M = 100$.

reduced-order methods so that only a handful of modes will be required to represent the solution accurately in physical space, i.e., a case with small M .

Further work is required to extend the theoretical analysis of [19] to the stochastic advection-diffusion-reaction equations we have considered here as well as to weak convergence for WCE-based algorithms. The numerical experiments in section 6 with the periodic passive scalar equation were motivated by a nonviscous transport equation with Kraichnan's velocity, which corresponds to an SPDE with a less regular solution than the one we simulated in section 6. Though we obtained promising results in our numerical tests, simulation of the passive scalar equation with Kraichnan's velocity requires special consideration. These aspects will be addressed in a future publication.

REFERENCES

- [1] E. J. ALLEN, S. J. NOVOSEL, AND Z. ZHANG, *Finite element and difference approximation of some linear stochastic partial differential equations*, Stochastics Stochastics Rep., 64 (1998), pp. 117–142.
- [2] A. BENSOUSSAN, R. GLOWINSKI, AND A. RĂȘCANU, *Approximation of some stochastic differential equations by the splitting up method*, Appl. Math. Optim., 25 (1992), pp. 81–106.
- [3] R. H. CAMERON AND W. T. MARTIN, *The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals*, Ann. of Math. (2), 48 (1947), pp. 385–392.
- [4] P. L. CHOW, J.-L. JIANG, AND J.-L. MENALDI, *Pathwise convergence of approximate solutions to Zakai's equation in a bounded domain*, in Stochastic Partial Differential Equations and Applications (Trento, 1990), Pitman Res. Notes Math. Ser. 268, Longman Scientific and Technical, Harlow, 1992, pp. 111–123.
- [5] A. DE BOUARD AND A. DEBUSSCHE, *Weak and strong order of convergence of a semidiscrete scheme for the stochastic nonlinear Schrödinger equation*, Appl. Math. Optim., 54 (2006), pp. 369–399.
- [6] A. DEBUSSCHE, *Weak approximation of stochastic partial differential equations: The nonlinear case*, Math. Comp., 80 (2011), pp. 89–117.
- [7] K. GAWĘDZKI AND M. VERGASSOLA, *Phase transition in the passive scalar advection*, Phys. D, 138 (2000), pp. 63–90.
- [8] A. GERMANI AND M. PICCIONI, *Semidiscretization of stochastic partial differential equations on \mathbf{R}^d by a finite-element technique*, Stochastics, 23 (1988), pp. 131–148.
- [9] E. GOBET, G. PAGÈS, H. PHAM, AND J. PRINTEMS, *Discretization and simulation of the Zakai equation*, SIAM J. Numer. Anal., 44 (2006), pp. 2505–2538.
- [10] W. GRECKSCH AND P. E. KLOEDEN, *Time-discretised Galerkin approximations of parabolic stochastic PDEs*, Bull. Austral. Math. Soc., 54 (1996), pp. 79–85.
- [11] I. GYÖNGY, *Approximations of stochastic partial differential equations*, in Stochastic Partial Differential Equations and Applications (Trento, 2002), Lecture Notes in Pure and Appl. Math. 227, Dekker, New York, 2002, pp. 287–307.
- [12] I. GYÖNGY AND N. KRYLOV, *On the splitting-up method and stochastic partial differential equations*, Ann. Probab., 31 (2003), pp. 564–591.
- [13] T. Y. HOU, W. LUO, B. ROZOVSKII, AND H.-M. ZHOU, *Wiener chaos expansions and numerical solutions of randomly forced equations of fluid mechanics*, J. Comput. Phys., 216 (2006), pp. 687–706.
- [14] A. JENTZEN AND P. E. KLOEDEN, *The numerical approximation of stochastic partial differential equations*, Milan J. Math., 77 (2009), pp. 205–244.
- [15] A. JENTZEN AND M. RÖCKNER, *A Break of the Complexity of the Numerical Approximation of Nonlinear SPDEs with Multiplicative Noise*, BiBoS Preprint 11–01–372, University of Bielefeld, Bielefeld, Germany, 2011.
- [16] R. H. KRAICHNAN, *Small-scale structure of a scalar field convected by turbulence*, Phys. Fluids, 11 (1968), pp. 945–953.
- [17] F. LE GLAND, *Splitting-up approximation for SPDEs and SDEs with application to nonlinear filtering*, in Stochastic Partial Differential Equations and Their Applications (Charlotte, NC, 1991), Lecture Notes in Control and Inform. Sci. 176, Springer, Berlin, 1992, pp. 177–187.
- [18] S. V. LOTOTSKIĬ AND B. L. ROZOVSKIĬ, *The passive scalar equation in a turbulent incompressible Gaussian velocity field*, Uspekhi Mat. Nauk, 59 (2004), pp. 105–120.

- [19] S. LOTOTSKY, R. MIKULEVICIUS, AND B. L. ROZOVSKII, *Nonlinear filtering revisited: A spectral approach*, SIAM J. Control Optim., 35 (1997), pp. 435–461.
- [20] S. LOTOTSKY AND B. ROZOVSKII, *Stochastic differential equations: A Wiener chaos approach*, in From Stochastic Calculus to Mathematical Finance, Springer, Berlin, 2006, pp. 433–506.
- [21] S. V. LOTOTSKY AND B. L. ROZOVSKII, *Wiener chaos solutions of linear stochastic evolution equations*, Ann. Probab., 34 (2006), pp. 638–662.
- [22] R. MIKULEVICIUS AND B. ROZOVSKII, *Separation of observations and parameters in nonlinear filtering*, in Proceedings of the 32nd IEEE Conference on in Decision and Control, Vol. 2, 1993, pp. 1564–1569.
- [23] R. MIKULEVICIUS AND B. ROZOVSKII, *Linear parabolic stochastic PDEs and Wiener chaos*, SIAM J. Math. Anal., 29 (1998), pp. 452–480.
- [24] G. N. MILSTEIN, YU. M. REPIN, AND M. V. TRETYAKOV, *Numerical methods for stochastic systems preserving symplectic structure*, SIAM J. Numer. Anal., 40 (2002), pp. 1583–1604.
- [25] G. N. MILSTEIN AND M. V. TRETYAKOV, *Stochastic Numerics for Mathematical Physics*, Sci. Comput., Springer-Verlag, Berlin, 2004.
- [26] G. N. MILSTEIN AND M. V. TRETYAKOV, *Solving parabolic stochastic partial differential equations via averaging over characteristics*, Math. Comp., 78 (2009), pp. 2075–2106.
- [27] J. PICARD, *Approximation of nonlinear filtering problems and order of convergence*, in Filtering and Control of Random Processes (Paris, 1983), Lecture Notes in Control and Inform. Sci. 61, Springer, Berlin, 1984, pp. 219–236.
- [28] B. L. ROZOVSKII, *Stochastic Evolution Systems: Linear Theory and Applications to Nonlinear Filtering*, Math. Appl. (Soviet Ser.) 35, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990.
- [29] G. ROZZA, D. B. P. HUYNH, AND A. T. PATERA, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations*, Arch. Comput. Methods Engrg., 15 (2007), pp. 1–47.
- [30] T. SHARDLOW, *Numerical methods for stochastic parabolic PDEs*, Numer. Funct. Anal. Optim., 20 (1999), pp. 121–145.
- [31] J. XU AND J. LI, *Sparse Wiener chaos approximations of Zakai equation for nonlinear filtering*, in Proceedings of the 21st Annual IEEE International Conference on Chinese Control and Decision Conference (CCDC'09), Piscataway, NJ, 2009, pp. 910–913.
- [32] Y. YAN, *Galerkin finite element methods for stochastic parabolic partial differential equations*, SIAM J. Numer. Anal., 43 (2005), pp. 1363–1384.
- [33] H. YOO, *Semi-discretization of stochastic partial differential equations on \mathbf{R}^1 by a finite-difference method*, Math. Comp., 69 (2000), pp. 653–666.