SIAM J. SCI. COMPUT.
Vol. 0, No. 0, pp. 000–000

A RECURSIVE SPARSE GRID COLLOCATION METHOD FOR DIFFERENTIAL EQUATIONS WITH WHITE NOISE*

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

Abstract. We consider a sparse grid collocation method in conjunction with a time discretization of the differential equations for computing expectations of functionals of solutions to differential equations perturbed by time-dependent white noise. We first analyze the error of Smolyak's sparse grid collocation used to evaluate expectations of functionals of solutions to stochastic differential equations discretized by the Euler scheme. We show theoretically and numerically that this algorithm can have satisfactory accuracy for small noise magnitude or small integration time, however it does not converge either with decrease of the Euler scheme's time step size or with increase of Smolyak's sparse grid level. Subsequently, we use this method as a building block for proposing a new algorithm by combining sparse grid collocation with a recursive procedure. This approach allows us to numerically integrate linear stochastic partial differential equations over longer times, which is illustrated in numerical tests on a stochastic advection-diffusion equation.

Key words. Smolyak's sparse grid, stochastic collocation, long time integration, stochastic partial differential equations

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

DOI. 10.1137/130938906

1. Introduction.

In a number of applications from physics, financial engineering, biology, and chemistry it is of interest to compute expectations of some functionals of solutions of ordinary stochastic differential equations (SDE) and stochastic partial differential equations (SPDE) driven by white noise. Usually, evaluation of such expectations requires one to approximate solutions of stochastic equations and then to compute the corresponding averages with respect to the approximate trajectories. We will not consider the former in this paper (see, e.g., [31] and references therein) and will concentrate on the latter. The most commonly used approach for computing the averages is the Monte Carlo technique, which is known for its slow rate of convergence and hence limiting computational efficiency of stochastic simulations. To speed up computation of the averages, variance reduction techniques (see, e.g., [31, 32] and the references therein), quasi-Monte Carlo algorithms [35, 43] and multilevel quasi-Monte Carlo methods [21], and the multilevel Monte Carlo method [14, 15] have been proposed and used.

An alternative approach to computing the averages is (stochastic) collocation methods in random space, which are deterministic methods in comparison with the Monte Carlo-type methods that are based on a statistical estimator of a mean. The expectation can be viewed as an integral with respect to the measure corresponding to

*Submitted to the journal's Methods and Algorithms for Scientific Computing section September 27, 2013; accepted for publication (in revised form) June 3, 2014; published electronically DATE.
http://www.siam.org/journals/sisc/x-x/93890.html

†Division of Applied Mathematics, Brown University, Providence, RI, 02912 (zhongqiang_zhang@brown.edu, boris_rozovsky@brown.edu, george_karniadakis@brown.edu). The work of these authors was partially supported by an OSD/MURI grant FA9550-09-1-0613, by NSF/DMS grant DMS-1216437, and also by the Collaboratory on Mathematics for Mesoscopic Modeling of Materials (CM4) which is sponsored by DOE. The third author was also partially supported by ARO grant W911NF-13-1-0012 and NSF/DMS grant DMS-1148284.

‡School of Mathematical Sciences, University of Nottingham, Nottingham, NG7 2RD, UK (Michael.Tretyakov@nottingham.ac.uk). This author's work was partially supported by the Leverhulme Trust Fellowship SAF-2012-006.

approximate trajectories. In stochastic collocation methods, one uses (deterministic) high-dimensional quadratures to evaluate these integrals. In the context of uncertainty quantification where moments of stochastic solutions are sought, collocation methods and their close counterparts (e.g., Wiener chaos expansion-based methods) have been very effective in reducing the overall computational cost in engineering problems; see, e.g., [13, 45, 48].

Stochastic equations or differential equations with randomness can be split into differential equations perturbed by time-independent noise and by time-dependent noise. It has been demonstrated in a number of works (see, e.g., [2, 3, 1, 47, 33, 36, 49] and references therein) that stochastic collocation methods can be a competitive alternative to the Monte Carlo technique and its variants in the case of differential equations perturbed by time-independent noise. The success of these methods relies on smoothness in the random space and can usually be achieved when it is sufficient to consider only a limited number of random variables (i.e., in the case of a low-dimensional random space). The small number of random variables significantly limits the applicability of stochastic collocation methods to differential equations perturbed by time-dependent noise as, in particular, will be demonstrated in this paper.

The class of stochastic collocation methods for SDE with time-dependent white noise includes cubatures on Wiener space [26], derandomization [34], optimal quantization [39, 40], and sparse grids of Smolyak type [11, 12, 18]. While derandomization and optimal quantization aim at finding quadrature rules which are in some sense optimal for computing a particular expectation under consideration, cubatures on Wiener space and a stochastic collocation method using Smolyak sparse grid quadratures (a sparse grid collocation method, SGC) use predetermined quadrature rules in a universal way without being tailored towards a specific expectation unless some adaptive strategies are applied. Since SGC is endowed with negative weights, it is, in practice, different from cubatures on Wiener space, where only quadrature rules with positive weights are used. Among quadrature rules, SGC is of particular interest due to its computational convenience. It has been considered in computational finance [11, 18], where high accuracy was observed. We note that the use of SGC in [11, 18] relies on exact sampling of geometric Brownian motion and of solutions of other simple SDE models, i.e., SGC in these works was not studied in conjunction with SDE approximations.

In this paper, we consider a SGC method accompanied by time discretization of differential equations perturbed by time-dependent noise. Our objective is twofold. *First*, using both analytical and numerical results, we warn that straightforward carrying over of stochastic collocation methods and, in particular, SGC to the case of differential equations perturbed by time-dependent noise (SDE or SPDE) usually leads to a failure. The main reason for this failure is that when integration time increases and/or time discretization step decreases, the number of random variables in approximation of SDE and SPDE grows quickly. The number of collocation points required for sufficient accuracy of collocation methods grows exponentially with the number of random variables. This results in failure of algorithms based on SGC and SDE time discretizations. Further, due to empirical evidence (see, e.g., [41]), the use of SGC is limited to problems with random space dimensionality of up to 40. Consequently, SGC algorithms for differential equations perturbed by time-dependent noise can be used only over small time intervals unless a cure for its fundamental limitation is found.

In section 2 (after a brief introduction to the sparse grid of Smolyak [44] (see also [46, 12, 47]) and to the weak-sense numerical integration for SDE (see, e.g., [31])), we obtain an error estimate for a SGC method accompanied by the Euler scheme

for evaluating expectations of smooth functionals of solutions of a scalar linear SDE with additive noise. In particular, we conclude that the SGC can successfully work for a small noise magnitude and relatively short integration time while in general it does not converge either with decrease of the time discretization step used for SDE approximation or with increase of the level of Smolyak's sparse grid; see Remark 2.6. Numerical tests in section 4 confirm our theoretical conclusions and we also observe first-order convergence in time step size of the algorithm using the SGC method as long as the SGC error is small relative to the error of time discretization of SDE. We note that our conclusion is, to some extent, similar to that for cubatures on Wiener space [6], for the Wiener chaos method [19, 23, 24, 50], and for some other functional expansion approaches [4, 5].

The *second objective* of the paper is to suggest a possible cure for the aforementioned deficiencies, which prevent SGC from being used over longer time intervals. For longer time simulation, deterministic replacements (such as stochastic collocation methods and functional expansion methods) of the Monte Carlo technique in simulation of differential equations perturbed by time-dependent noise do not work effectively unless some restarting strategies allowing it to "forget" random variables from earlier time steps are employed. Examples of such strategies are the recursive approach for Wiener chaos expansion methods to compute moments of solutions to linear SPDE [23, 50] and an approach for cubatures on Wiener space based on compressing the history data via a regression at each time step [22].

Here we exploit the idea of the recursive approach to achieve accurate longer time integration by numerical algorithms using the SGC. For linear SPDE with *time-independent* coefficients, the recursive approach works as follows. We first find an approximate solution of an SPDE at a relatively small time $t = h$, and subsequently take the approximation at $t = h$ as the initial value in order to compute the approximate solution at $t = 2h$, and so on, until we reach the final integration time $T = Nh$. To find second moments of the SPDE solution, we store a covariance matrix of the approximate solution at each time step kh and recursively compute the first two moments. Such an algorithm is proposed in section 3; in section 4 we demonstrate numerically that this algorithm converges in time step h and that it can work well on longer time intervals. At the same time, a major challenge remains: how to effectively use restarting strategies for SGC in the case of nonlinear SDE and SPDE and further work is needed in this direction.

2. Sparse grid for weak integration of SDE.

2.1. Smolyak's sparse grid. Sparse grid quadrature is a certain reduction of product quadrature rules which decreases the number of quadrature nodes and allows effective integration in moderately high dimensions [44] (see also [46, 38, 12]). Here we introduce it in the form suitable for our purposes.

We will be interested in evaluating d-dimensional integrals of a function $\varphi(y)$, $y \in \mathbb{R}^d$, with respect to a Gaussian measure:

$$I_d\varphi := \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} \varphi(y) \exp\left(-\frac{1}{2}\sum_{i=1}^{d} y_i^2\right) dy_1 \cdots dy_d. \tag{2.1}$$

Consider a sequence of one-dimensional Gauss–Hermite quadrature rules Q_n with number of nodes $n \in \mathbb{N}$ for univariate functions $\psi(\mathsf{y})$, $\mathsf{y} \in \mathbb{R}$:

$$Q_n\psi(\mathsf{y}) = \sum_{\alpha=1}^{n} \psi(\mathsf{y}_{n,\alpha})\mathsf{w}_{n,\alpha}, \tag{2.2}$$

TABLE 1
The number of sparse grid points for the sparse grid quadrature (2.4) *using the one-dimensional Gauss–Hermite quadrature rule* (2.2)*, when the sparse grid level* $L \le d$.

	$L=1$	$L=2$	$L=3$	$L=4$	$L=5$
$\#S$	1	$2d+1$	$2d^2+2d+1$	$\frac{4}{3}d^3+2d^2+\frac{14}{3}d+1$	$\frac{2}{3}d^4+\frac{4}{3}d^3+\frac{22}{3}d^2+\frac{8}{3}d+1$

where $\mathsf{y}_{n,1} < \mathsf{y}_{n,2} < \cdots < \mathsf{y}_{n,n}$ are the roots of the Hermite polynomial $H_n(\mathsf{y}) = (-1)^n e^{\mathsf{y}^2/2}\frac{d^n}{d\mathsf{y}^n}e^{-\mathsf{y}^2/2}$ and $\mathsf{w}_{n,\alpha} = n!/(n^2[H_{n-1}(\mathsf{y}_{n,\alpha})]^2)$ are the associated weights. It is known that $Q_n\psi$ is exactly equal to the integral $I_1\psi$ when ψ is a polynomial of degree less than or equal to $2n-1$, i.e., the polynomial degree of exactness of Gauss–Hermite quadrature rules Q_n is equal to $2n-1$.

We can approximate the multidimensional integral $I_d\varphi$ by a quadrature expressed as the tensor product rule

$$I_d\varphi \approx \bar{I}_d\varphi := Q_n \otimes Q_n \cdots \otimes Q_n\varphi(y_1, y_2, \ldots, y_d) = Q_n^{\otimes d}\varphi(y_1, y_2, \ldots, y_d) \tag{2.3}$$
$$= \sum_{\alpha_1=1}^{n} \cdots \sum_{\alpha_d=1}^{n} \varphi(\mathsf{y}_{n,\alpha_1}, \ldots, \mathsf{y}_{n,\alpha_d})\mathsf{w}_{n,\alpha_1} \cdots \mathsf{w}_{n,\alpha_d},$$

where for simplicity we use the same amount on nodes in all the directions. The quadrature $\bar{I}_d\varphi$ is exact for all polynomials from the space $\mathcal{P}_{k_1} \otimes \cdots \otimes \mathcal{P}_{k_d}$ with $\max k_i = 2n-1$, where $\mathcal{P}_k$ is the space of one-dimensional polynomials of degree less than or equal to k (we note in passing that this fact is easy to prove using probabilistic representations of $I_d\varphi$ and $\bar{I}_d\varphi$). Computational costs of quadrature rules are measured in terms of the number of function evaluations which is equal to n^d in the case of the tensor product (2.3), i.e., the computational cost of (2.3) grows exponentially fast with dimension.

The sparse grid of Smolyak [44] reduces computational complexity of the tensor product rule (2.3) via exploiting the difference quadrature formulas:

$$A(L,d)\varphi := \sum_{d \le |\mathbf{i}| \le L+d-1} (Q_{i_1} - Q_{i_1-1}) \otimes \cdots \otimes (Q_{i_d} - Q_{i_d-1})\varphi,$$

where $Q_0 = 0$ and $\mathbf{i} = (i_1, i_2, \ldots, i_d)$ is a multi-index with $i_k \ge 1$ and $|\mathbf{i}| = i_1 + i_2 + \cdots + i_d$. The number L is usually referred to as the level of the sparse grid. The sparse grid rule (2.4) can also be written in the following form [46]:

$$A(L,d)\varphi = \sum_{L \le |\mathbf{i}| \le L+d-1} (-1)^{L+d-1-|\mathbf{i}|} \binom{d-1}{|\mathbf{i}|-L} Q_{i_1} \otimes \cdots \otimes Q_{i_d}\varphi. \tag{2.4}$$

The quadrature $A(L,d)\varphi$ is exact for polynomials from the space $\mathcal{P}_{k_1} \otimes \cdots \otimes \mathcal{P}_{k_d}$ with $|\mathbf{k}| = 2L-1$, i.e., for polynomials of total degree up to $2L-1$ [38, Corollary 1]. Due to (2.4), the total number of nodes used by this sparse grid rule is estimated by

$$\#S \le \sum_{L \le |\mathbf{i}| \le L+d-1} i_1 \times \cdots \times i_d.$$

Table 1 lists the number of sparse grid points, $\#S$, up to level 5 when the level is not greater than d.

The quadrature $\bar{I}_d\varphi$ from (2.3) is exact for polynomials of total degree $2L-1$ when $n = L$. It is not difficult to see that if the required polynomial exactness (in

terms of total degree of polynomials) is relatively small then the sparse grid rule (2.4) substantially reduces the number of function evaluations compared with the tensor-product rule (2.3). For instance, suppose that the dimension $d = 40$ and the required polynomial exactness is equal to 3. Then the cost of the tensor product rule (2.3) is $3^{40} \doteq 1.2158 \times 10^{19}$ while the cost of the sparse grid rule (2.4) based on one-dimensional rule (2.2) is 3281.

Remark 2.1. In this paper we consider the isotropic SGC. More efficient algorithms might be built using anisotropic SGC methods [18, 37], which employ more quadrature points along the "most important" direction. Goal-oriented quadrature rules, e.g., [34, 39, 40], can also be exploited instead of predetermined quadrature rules used here. However, the effectiveness of adaptive sparse grids relies heavily on the order of importance in random dimensions of numerical solutions to stochastic differential equations, which is not always easy to reach. Furthermore, all these sparse grids as integration methods in random space grow quickly with random dimensions and thus cannot be used for longer time integration (usually with large random dimensions). Hence, we consider only isotropic SGC.

2.2. Weak-sense integration of SDE. Let $(\Omega, \mathcal{F}, P)$ be a probability space and $(w(t), \mathcal{F}_t^w) = ((w_1(t), \dots, w_r(t))^\intercal, \mathcal{F}_t^w)$ be an r-dimensional standard Wiener process, where $\mathcal{F}_t^w,\ 0 \le t \le T$, is an increasing family of σ-subalgebras of $\mathcal{F}$ induced by $w(t)$.

Consider the system of Ito SDEs

$$dX = a(t, X)dt + \sum_{l=1}^{r} \sigma_l(t, X)dw_l(t), \quad t \in (t_0, T],\ X(t_0) = x_0, \tag{2.5}$$

where X, a, σ_r are m-dimensional column vectors and x_0 is independent of w. We assume that $a(t, x)$ and $\sigma(t, x)$ are sufficiently smooth and globally Lipschitz. We are interested in computing the expectation

$$u(x_0) = \mathbb{E} f(X_{t_0,x_0}(T)), \tag{2.6}$$

where $f(x)$ is a sufficiently smooth function with growth at infinity not faster than a polynomial:

$$|f(x)| \le K(1 + |x|^\varkappa) \tag{2.7}$$

for some $K > 0$ and $\varkappa \ge 1$.

To find $u(x_0)$, we first discretize the solution of (2.5). Let

$$h = (T - t_0)/N, \quad t_k = t_0 + kh, \quad k = 0, \dots, N.$$

In application to (2.5) the Euler scheme has the form

$$X_{k+1} = X_k + a(t_k, X_k)h + \sum_{l=1}^{r} \sigma_l(t_k, X_k)\Delta_k w_l, \tag{2.8}$$

where $X_0 = x_0$ and $\Delta_k w_l = w_l(t_{k+1}) - w_l(t_k)$. The Euler scheme can be realized in practice by replacing the increments $\Delta_k w_l$ with Gaussian random variables:

$$X_{k+1} = X_k + a(t_k, X_k)h + \sum_{l=1}^{r} \sigma_l(t_k, X_k)\sqrt{h}\xi_{l,k+1}, \tag{2.9}$$

where $\xi_{r,k+1}$ are independently and identically distributed (i.i.d.) $\mathcal{N}(0,1)$ random variables. Due to our assumptions, the following error estimate holds for (2.9) (see, e.g., [31, Chapter 2]):

$$|\mathbb{E}f(X_N) - \mathbb{E}f(X(T))| \le Kh, \tag{2.10}$$

where $K > 0$ is a constant independent of h. This first-order weak convergence can also be achieved by replacing $\xi_{l,k+1}$ with discrete random variables [31], e.g., the weak Euler scheme has the form

$$\tilde{X}_{k+1} = \tilde{X}_k + ha(t_k, \tilde{X}_k) + \sqrt{h}\sum_{l=1}^{r}\sigma_l(t_k, \tilde{X}_k)\zeta_{l,k+1}, \quad k = 0, \dots, N-1, \tag{2.11}$$

where $\tilde{X}_0 = x_0$ and $\zeta_{l,k+1}$ are i.i.d. random variables with the law

$$P(\zeta = \pm 1) = 1/2. \tag{2.12}$$

The following error estimate holds for (2.11)–(2.12) (see, e.g., [31, Chapter 2]):

$$|\mathbb{E}f(\tilde{X}_N) - \mathbb{E}f(X(T))| \le Kh\ , \tag{2.13}$$

where $K > 0$ can be a different constant than in (2.10).

Introducing the function $\varphi(y)$, $y \in \mathbb{R}^{rN}$, so that

$$\varphi(\xi_{1,1}, \dots, \xi_{r,1}, \dots, \xi_{1,N}, \dots, \xi_{r,N}) = f(X_N), \tag{2.14}$$

we have

$$\begin{aligned} u(x_0) &\approx \bar{u}(x_0) \\ &:= \mathbb{E}f(X_N) = \mathbb{E}\varphi(\xi_{1,1}, \dots, \xi_{r,1}, \dots, \xi_{1,N}, \dots, \xi_{r,N}) \\ &= \frac{1}{(2\pi)^{rN/2}}\int_{\mathbb{R}^{rN}} \varphi(y_{1,1}, \dots, y_{r,1}, \dots, y_{1,N}, \dots, y_{r,N}) \exp\left(-\frac{1}{2}\sum_{i=1}^{rN} y_i^2\right) dy. \end{aligned} \tag{2.15}$$

Further, it is not difficult to see from (2.11)–(2.12) and (2.3) that

$$\begin{aligned} u(x_0) \approx \tilde{u}(x_0) &:= \mathbb{E}f(\tilde{X}_N) = \mathbb{E}\varphi(\zeta_{1,1}, \dots, \zeta_{r,1}, \dots, \zeta_{1,N}, \dots, \zeta_{r,N}) \\ &= Q_2^{\otimes rN}\varphi(y_{1,1}, \dots, y_{r,1}, \dots, y_{1,N}, \dots, y_{r,N}), \end{aligned} \tag{2.16}$$

where Q_2 is the Gauss–Hermite quadrature rule defined in (2.2) with $n = 2$, i.e., $\mathsf{y}_1 = 1$, $\mathsf{y}_2 = -1$ with weights $\mathsf{w}_1 = \mathsf{w}_2 = 1/2$). Comparing (2.15) and (2.16), we can say that $\tilde{u}(x_0)$ is a tensor-product quadrature rule for the multidimensional integral $\bar{u}(x_0)$. In other words, the weak Euler scheme (2.11)–(2.12) can be interpreted as the strong Euler scheme with tensor-product integration in random space.We note that the approximation, $\tilde{u}(x_0)$, of $\bar{u}(x_0)$ satisfies (cf. (2.10) and (2.13))

$$\bar{u}(x_0) - \tilde{u}(x_0) = O(h). \tag{2.17}$$

Remark 2.2. Let $\zeta_{l,k+1}$ in (2.11) be i.i.d. random variables with the law

$$P(\zeta = \mathsf{y}_{n,j}) = \mathsf{w}_{n,j}, \quad j = 1, \dots, n, \tag{2.18}$$

where $\mathsf{y}_{n,j}$ are nodes of the Gauss–Hermite quadrature Q_n and $\mathsf{w}_{n,j}$ are the corresponding quadrature weights (see (2.2)). Then

$$\mathbb{E}f(\tilde{X}_N) = \mathbb{E}\varphi(\zeta_{1,N},\dots,\zeta_{r,N}) = Q_n^{\otimes rN}\varphi(y_{1,1},\dots,y_{r,N}),$$

which can be a more accurate approximation of $\bar{u}(x_0)$ than $\tilde{u}(x_0)$ from (2.16) but the weak-sense error for the SDE approximation $\mathbb{E}f(\tilde{X}_N) - \mathbb{E}f(X(T))$ remains of order $O(h)$.

Practical implementation of $\bar{u}(x_0)$ and $\tilde{u}(x_0)$ usually requires the use of the Monte Carlo technique since the computational cost of, e.g., the tensor product rule in (2.16) is prohibitively high (cf. section 2.1). In this paper, we consider application of the sparse grid rule (2.4) to the integral in (2.15) motivated by lower computational cost of (2.4).

In this approach, the total error has two parts:

$$|\mathbb{E}f(X(T)) - A(L,N)\varphi| \le \left|\mathbb{E}f(X(T)) - \mathbb{E}f(\tilde{X}_N)\right| + \left|\mathbb{E}f(\tilde{X}_N) - A(L,N)\varphi\right|,$$

where $A(L,N)$ is defined in (2.4) and φ is from (2.14). The first part is controlled by the time step size h (see (2.10)), and it converges to zero with order one in h. The second part is controlled by the sparse grid level L but it depends on h since decrease of h increases the dimension of the random space. Some illustrative examples will be presented in section 2.3.

2.2.1. Probabilistic interpretation of SGC. It is not difficult to show that SGC admits a probabilistic interpretation, e.g., in the case of level $L = 2$ we have

$$\begin{aligned}(2.19)\quad & A(2,N)\varphi(y_{1,1},\dots,y_{r,1},\dots,y_{1,N},\dots,y_{r,N})\\ &= (Q_2\otimes Q_1\otimes\cdots\otimes Q_1)\,\varphi + (Q_1\otimes Q_2\otimes Q_1\otimes\cdots\otimes Q_1)\,\varphi\\ &\quad + \cdots + (Q_1\otimes Q_1\otimes\cdots\otimes Q_2)\,\varphi - (Nr-1)\,(Q_1\otimes Q_1\otimes\cdots\otimes Q_1)\,\varphi\\ &= \sum_{i=1}^{N}\sum_{j=1}^{r}\mathbb{E}\varphi(0,\dots,0,\zeta_{j,i},0,\dots,0) - (Nr-1)\varphi(0,0,\dots,0),\end{aligned}$$

where $\zeta_{j,i}$ are i.i.d. random variables with the law (2.12). Using (2.16), (2.19), Taylor's expansion, and symmetry of $\zeta_{j,i}$, we obtain the relationship between the weak Euler scheme (2.11) and the SGC (2.4):

$$\begin{aligned}(2.20)\quad & \mathbb{E}f(\tilde{X}_N) - A(2,N)\varphi\\ &= \mathbb{E}\varphi(\zeta_{1,1},\dots,\zeta_{r,1},\dots,\zeta_{1,N},\dots,\zeta_{r,N})\\ &\quad - \sum_{i=1}^{N}\sum_{j=1}^{r}\mathbb{E}\varphi(0,\dots,0,\zeta_{j,i},0,\dots,0) - (Nr-1)\varphi(0,0,\dots,0)\\ &= \sum_{|\alpha|=4}\frac{4}{\alpha!}\mathbb{E}\left[\prod_{i=1}^{N}\prod_{j=1}^{r}(\zeta_{j,i})^{\alpha_{j,i}}\int_0^1(1-z)^3 D^{\alpha}\varphi(z\zeta_{1,1},\dots,z\zeta_{r,N})\,dz\right]\\ &\quad - \frac{1}{3!}\sum_{i=1}^{N}\sum_{j=1}^{r}\mathbb{E}\left[\zeta_{j,i}^4\int_0^1(1-z)^3\frac{\partial^4}{(\partial y_{j,i})^4}\varphi(0,\dots,0,z\zeta_{j,i},0,\dots,0)\,dz\right],\end{aligned}$$

where the multi-index $\alpha = (\alpha_{1,1},\dots,\alpha_{r,N}) \in \mathbb{N}_0^{rN}$, $|\alpha| = \sum_{i=1}^{N}\sum_{j=1}^{r}\alpha_{j,i}$, $\alpha! = \prod_{i=1}^{N}\prod_{j=1}^{r}\alpha_{j,i}!$, and $D^{\alpha} = \frac{\partial^{|\alpha|}}{(\partial y_{1,1})^{\alpha_{1,1}}\cdots(\partial y_{r,N})^{\alpha_{r,N}}}$. Similarly, we can write down a

probabilistic interpretation for any L and derive a similar error representation. For example, we have for $L = 3$ that

$$\begin{aligned}
&\mathbb{E}[\varphi(\zeta^{(3)}_{1,1},\dots,\zeta^{(3)}_{r,N})] - A(3,N)\varphi \\
&\quad= \sum_{|\alpha|=6} \frac{6}{\alpha!}\mathbb{E}\left[\prod_{i=1}^{N}\prod_{j=1}^{r}(\zeta^{(3)}_{j,i})^{\alpha_{j,i}} \int_0^1 (1-z)^5 D^\alpha \varphi(z\zeta^{(3)}_{1,1},\dots,z\zeta^{(3)}_{r,N})\,dz\right] \\
&\qquad- \sum_{\substack{|\alpha|=\alpha_{j,i}+\alpha_{l,k}=6 \\ (j-l)^2+(i-k)^2\neq 0}} \frac{6}{\alpha_{j,i}!\alpha_{k,l}!} \\
&\qquad\mathbb{E}\left[(\zeta^{(3)}_{j,i})^{\alpha_{j,i}}(\zeta^{(3)}_{l,k})^{\alpha_{l,k}} \int_0^1 (1-z)^5 D^\alpha \varphi(\dots,z\zeta^{(3)}_{j,i},0,\dots,0,z\zeta^{(3)}_{l,k},\dots)\,dz\right] \\
&\qquad-\sum_{i=1}^{N}\sum_{j=1}^{r}\frac{6}{6!}\mathbb{E}\left[(\zeta_{j,i})^6\int_0^1(1-z)^5 D^\alpha\varphi(0,\dots,z\zeta_{j,i},\dots,0)\,dz\right],
\end{aligned}$$

where $\zeta_{j,i}$ are defined in (2.12) and $\zeta^{(3)}_{j,i}$ are i.i.d. random variables with the law $P(\zeta^{(3)}_{j,i} = \pm\sqrt{3}) = 1/6$, $P(\zeta^{(3)}_{j,i} = 0) = 2/3$.

The error of the SGC applied to the weak-sense approximation of SDE is further studied in section 2.3.

2.2.2. Second-order schemes. In the SGC context, it is beneficial to exploit higher-order or higher-accuracy schemes for approximating the SDE (2.5) because they can allow us to reach a desired accuracy using larger time step sizes and therefore less random variables than the first-order Euler scheme (2.9) or (2.11). For instance, we can use the second-order weak scheme for (2.5) (see, e.g., [31, Chapter 2]):

$$\begin{aligned}
X_{k+1} = X_k + ha(t_k,X_k) + \sqrt{h}\sum_{i=1}^{r}\sigma_i(t_k,X_k)\xi_{i,k+1} + \frac{h^2}{2}\mathfrak{L}a(t_k,X_k) \\
+ h\sum_{i=1}^{r}\sum_{j=1}^{r}\Lambda_i\sigma_j(t_k,X_k)\eta_{i,j,k+1} \\
+ \frac{h^{3/2}}{2}\sum_{i=1}^{r}(\Lambda_i a(t_k,X_k) + \mathfrak{L}\sigma_i(t_k,X_k))\xi_{i,k+1}, \\
k = 0,\dots,N-1,
\end{aligned} \tag{2.21}$$

where $X_0 = x_0$; $\eta_{i,j} = \frac{1}{2}\xi_i\xi_j - \gamma_{i,j}\zeta_i\zeta_j/2$ with $\gamma_{i,j} = -1$ if $i < j$ and $\gamma_{i,j} = 1$ otherwise;

$$\Lambda_l = \sum_{i=1}^{m}\sigma^i_l\frac{\partial}{\partial x_i}, \quad \mathfrak{L} = \frac{\partial}{\partial t} + \sum_{i=1}^{m}a^i\frac{\partial}{\partial x_i} + \frac{1}{2}\sum_{l=1}^{r}\sum_{i,j=1}^{m}\sigma^i_l\sigma^i_l\frac{\partial^2}{\partial x_i\partial x_j};$$

and $\xi_{i,k+1}$ and $\zeta_{i,k+1}$ are mutually independent random variables with Gaussian distribution or with the laws $P(\xi = 0) = 2/3$, $P(\xi = \pm\sqrt{3}) = 1/6$, and $P(\zeta = \pm 1) = 1/2$. The following error estimate holds for (2.21) (see, e.g., [31, Chapter 2]):

$$|\mathbb{E}f(X(T)) - Ef(X_N)| \le Kh^2.$$

Roughly speaking, to achieve $O(h)$ accuracy using (2.21), we need only $\sqrt{2rN}$ ($\sqrt{rN}$ in the case of additive noise) random variables, while we need rN random

variables for the Euler scheme (2.9). This reduces the dimension of the random space and hence can increase efficiency and widen applicability of SGC methods (see, in particular Example 4.1 in section 4 for a numerical illustration). We note that when noise intensity is relatively small, we can use high-accuracy low-order schemes designed for SDE with small noise [30] (see also [31, Chapter 3]) in order to achieve a desired accuracy using a fewer number of random variables than the Euler scheme (2.9).

2.3. Illustrative examples. In this section we show limitations of the use of SGC in a weak approximation of SDE. To this end, it is convenient and sufficient to consider the scalar linear SDE

$$dX = \lambda X dt + \varepsilon\, dw(t), \quad X_0 = 1, \tag{2.22}$$

where λ and ε are some constants.

We will compute expectations $\mathbb{E}f(X(T))$ for some $f(x)$ and $X(t)$ from (2.22) by applying the Euler scheme (2.9) and the SGC (2.4). This simple example provides us with a clear insight as to when algorithms of this type are able to produce accurate results and when they are likely to fail. Using direct calculations, we first (see Examples 2.3–2.4 below) derive an estimate for the error $|\mathbb{E}f(X_N) - A(2,N)\varphi|$ with X_N from (2.9) applied to (2.22) and for some particular $f(x)$. This will illustrate how the error of SGC with practical level (no more than six) behaves. Then (Proposition 2.5) we obtain an estimate for the error $|\mathbb{E}f(X_N) - A(L,N)\varphi|$ for a smooth $f(x)$ which grows no faster than a polynomial function at infinity. We will observe that the considered algorithm is not convergent in time step h and the algorithm is not convergent in level L unless when noise intensity and integration time are small.

It follows from (2.10) and (2.13) that

$$\begin{aligned} |\mathbb{E}f(X_N) - A(L,N)\varphi| &\le \left|\mathbb{E}f(\tilde{X}_N) - A(L,N)\varphi\right| + |\mathbb{E}f(X_N) - \mathbb{E}f(\tilde{X}_N)| \\ &\le \left|\mathbb{E}f(\tilde{X}_N) - A(L,N)\varphi\right| + Kh, \end{aligned} \tag{2.23}$$

where $\tilde{X}_N$ is from the weak Euler scheme (2.11) applied to (2.22), which can be written as $\tilde{X}_N = (1+\lambda h)^N + \sum_{j=1}^N (1+\lambda h)^{N-j}\varepsilon\sqrt{h}\zeta_j$. Introducing the function

$$\bar{X}(N;y) = (1+\lambda h)^N + \sum_{j=1}^N (1+\lambda h)^{N-j}\varepsilon\sqrt{h}y_j, \tag{2.24}$$

we see that $\tilde{X}_N = \bar{X}(N;\zeta_1,\dots,\zeta_N)$. We have

$$\frac{\partial}{\partial y_i}\bar{X}(N;y) = (1+\lambda h)^{N-i}\varepsilon\sqrt{h} \ \text{ and } \ \frac{\partial^2}{\partial y_i \partial y_j}\bar{X}(N;y) = 0. \tag{2.25}$$

Then we obtain from (2.20)

(2.26)

$$\begin{aligned} R &:= \mathbb{E}f(\tilde{X}_N) - A(2,N)\varphi \\ &= \varepsilon^4 h^2 \sum_{|\alpha|=4} \frac{4}{\alpha!}\mathbb{E}\left[\prod_{i=1}^N (\zeta_i(1+\lambda h)^{N-i})^{\alpha_i} \int_0^1 (1-z)^3 z^4 \frac{d^4}{dx^4} f(\bar{X}(N, z\zeta_1,\dots,z\zeta_N))\, dz\right] \\ &\quad - \frac{1}{3!}\varepsilon^4 h^2 \sum_{i=1}^N \mathbb{E}\left[\zeta_i^4 \int_0^1 (1-z)^3 z^4 \frac{d^4}{dx^4} f(\bar{X}(0,\dots,0,z\zeta_i,0,\dots,0))\,(1+\lambda h)^{4N-4i} dz\right]. \end{aligned}$$

2.3.1. Nonconvergence in time step h. We will illustrate no convergence in h for SGC for levels two and three through two examples, where sharp error estimates of $|\mathbb{E}f(X_N) - A(2,N)\varphi|$ are derived for SGC. Higher level SGC can also be considered but the conclusion does not change. In contrast, the algorithm of tensor-product integration in random space and the strong Euler scheme in time (i.e., the weak Euler scheme (2.11)–(2.12)) is convergent with order one in h. We also note that in practice, typically SGCs with level no more than six are employed.

Example 2.3. For $f(x) = x^p$ with $p = 1, 2, 3$, it follows from (2.26) that $R = 0$, i.e., SGC does not introduce any additional error, and hence by (2.23)

$$|\mathbb{E}f(X_N) - A(2,N)\varphi| \le Kh, \quad f(x) = x^p, \quad p = 1, 2, 3.$$

For $f(x) = x^4$, we get from (2.26)

$$\begin{aligned} R &= \frac{6}{35}\varepsilon^4 h^2 \sum_{i=1}^{N}\sum_{j=i+1}^{N} (1+\lambda h)^{4N-2i-2j} \\ &= \frac{6}{35}\varepsilon^4 \times \begin{cases} \frac{(1+\lambda h)^{2N}-1}{\lambda^2(2+\lambda h)^2}\left[\frac{(1+\lambda h)^{2N}+1}{1+(1+\lambda h)^2} - 1\right], & \lambda \neq 0, \quad 1+\lambda h \neq 0, \\ \frac{T^2}{2} - \frac{Th}{2}, & \lambda = 0. \end{cases} \end{aligned}$$

We see that R does not go to zero when $h \to 0$ and that for sufficiently small $h > 0$

$$|\mathbb{E}f(X_N) - A(2,N)\varphi| \le Kh + \frac{6}{35}\varepsilon^4 \times \begin{cases} \frac{1}{\lambda^2}(1+e^{4T\lambda}), & \lambda \neq 0, \\ \frac{T^2}{2}, & \lambda = 0. \end{cases}$$

We observe that the SGC algorithm does not converge with $h \to 0$ for higher moments. In the considered case of linear SDE, increasing the level L of SGC leads to the SGC error R being 0 for higher moments, e.g., for $L = 3$ the error $R = 0$ for up to the 5th moment but the algorithm will not converge in h for the 6th moment and so on (see Proposition 2.5 below). Further (see the continuation of the illustration below), in the case of, e.g., $f(x) = \cos x$ for any L this error R does not converge in h, which is also the case for nonlinear SDE. We also note that one can expect that this error R is small when noise intensity is relatively small and either time T is small or SDE has, in some sense, stable behavior (in the linear case it corresponds to $\lambda < 0$).

Example 2.4. Now consider $f(x) = \cos(x)$. It follows from (2.26) that

$$\begin{aligned} R = \varepsilon^4 h^2 \sum_{|\alpha|=4} \frac{4}{\alpha!}\mathbb{E}\Bigg[\prod_{i=1}^{N}(\zeta_i(1+\lambda h)^{N-i})^{\alpha_i} \int_0^1 (1-z)^3 z^4 \\ \times \cos\left((1+\lambda h)^N + z\sum_{j=1}^{N}(1+\lambda h)^{N-j}\varepsilon\sqrt{h}\zeta_j\right) dz\Bigg] \\ - \frac{1}{3!}\varepsilon^4 h^2 \sum_{i=1}^{N}(1+\lambda h)^{4N-4i}\int_0^1 (1-z)^3 z^4 \mathbb{E}[\zeta_i^4 \cos((1+\lambda h)^N + z(1+\lambda h)^{N-i}\varepsilon\sqrt{h}\zeta_i)]\, dz \end{aligned}$$

and after routine calculations we obtain

$$
\begin{aligned}
R = {} & \varepsilon^4 h^2 \cos((1+\lambda h)^N) \\
& \left[\left(\frac{1}{6} \sum_{i=1}^{N} (1+\lambda h)^{4N-4i} + 2 \sum_{i=1}^{N} \sum_{j=i+1}^{N} (1+\lambda h)^{4N-2i-2j} \right) \right. \\
& \times \int_0^1 (1-z)^3 z^4 \prod_{l=1}^{N} \cos(z(1+\lambda h)^{N-l} \varepsilon \sqrt{h}) dz \\
& + \left(\frac{2}{3} \sum_{i,j=1; i \neq j}^{N} (1+\lambda h)^{4N-3i-j} + 2 \sum_{\substack{k,i,j=1 \\ i \neq j, i \neq k, k \neq j}}^{N} (1+\lambda h)^{4N-2k-i-j} \right) \\
& \times \int_0^1 (1-z)^3 z^4 \prod_{l=i,j} \sin(z(1+\lambda h)^{N-l} \varepsilon \sqrt{h}) \prod_{\substack{l=1 \\ l \neq i, l \neq j}}^{N} \cos(z(1+\lambda h)^{N-l} \varepsilon \sqrt{h}) dz \\
& + 4 \sum_{\substack{i,j,k,m=1 \\ i \neq j, i \neq k, i \neq m, j \neq k, j \neq m, k \neq m}}^{N} (1+\lambda h)^{4N-i-j-k-m} \\
& \times \int_0^1 (1-z)^3 z^4 \prod_{l=i,j,k,m} \sin(z(1+\lambda h)^{N-l} \varepsilon \sqrt{h}) \prod_{\substack{l=1 \\ l \neq i, l \neq j, l \neq k, l \neq m}}^{N} \cos(z(1+\lambda h)^{N-l} \varepsilon \sqrt{h}) dz \\
& \left. - \frac{1}{6} \sum_{i=1}^{N} (1+\lambda h)^{4N-4i} \int_0^1 (1-z)^3 z^4 \cos(z(1+\lambda h)^{N-i} \varepsilon \sqrt{h})] \, dz \right].
\end{aligned}
$$

It is not difficult to see that R does not go to zero when $h \to 0$. In fact, taking into account that $|\sin(z(1+\lambda h)^{N-j} \varepsilon \sqrt{h})| \leq z(1+\lambda h)^{N-j} \varepsilon \sqrt{h}$, and that there are N^4 terms of order h^4 and N^3 terms of order h^3, we get for sufficiently small $h > 0$

$$|R| \leq C\varepsilon^4 (1 + e^{4T\lambda}),$$

where $C > 0$ is independent of ε and h. Hence

$$|\mathbb{E} f(X_N) - A(2,N)\varphi| \leq C\varepsilon^4 (1 + e^{4T\lambda}) + Kh, \tag{2.27}$$

and we have arrived at a similar conclusion for $f(x) = \cos x$ as for $f(x) = x^4$. Similarly, we can also have for $L = 3$ that

$$|\mathbb{E} f(X_N) - A(3,N)\varphi| \leq C\varepsilon^6 (1 + e^{6T\lambda}) + Kh.$$

This example shows, for $L = 3$, the error of SGC with the Euler scheme in time does not converge in h.

2.3.2. Error estimate for SGC with fixed level. Now we will address the effect of the SGC level L. To this end, we will need the following error estimate of a Gauss–Hermite quadrature. Let $\psi(y)$, $y \in \mathbb{R}$, be a sufficiently smooth function whose derivatives and itself are growing not faster than a polynomial at infinity. Using the Peano kernel theorem (see, e.g., [9]) and that a Gauss–Hermite quadrature with n-nodes has the order of polynomial exactness $2n-1$, we obtain, for the approximation

error $R_{n,\gamma}\psi$ of the Gauss–Hermite quadrature $Q_n\psi$,

$$R_{n,\gamma}(\psi) := Q_n\psi - I_1\psi = \int_{\mathbb{R}} \frac{d^\gamma}{dy^\gamma}\varphi(y) R_{n,\gamma}(\Gamma_{y,\gamma})\, dy, \quad 1 \le \gamma \le 2n, \tag{2.28}$$

where $\Gamma_{y,\gamma}(z) = (z-y)^{\gamma-1}/(\gamma-1)!$ if $z \ge y$ and 0 otherwise. One can show (see, e.g., [27, Theorem 2]) that there is a constant $c > 0$ independent of n and y such that for any $0 < \beta < 1$

$$|R_{n,\gamma}(\Gamma_{y,\gamma})| \le \frac{c}{\sqrt{2\pi}} n^{-\gamma/2} \exp\left(-\frac{\beta y^2}{2}\right), \quad 1 \le \gamma \le 2n. \tag{2.29}$$

We also note that (2.29) and the triangle inequality imply, for $1 \le \gamma \le 2(n-1)$,

$$|R_{n,\gamma}(\Gamma_{y,\gamma}) - R_{n-1,\gamma}(\Gamma_{y,\gamma})| \le \frac{c}{\sqrt{2\pi}} [n^{-\gamma/2} + (n-1)^{-\gamma/2}] \exp\left(-\frac{\beta y^2}{2}\right). \tag{2.30}$$

Now we consider an error of the sparse grid rule (2.4) accompanied by the Euler scheme (2.9) for computing expectations of solutions to (2.22).

PROPOSITION 2.5. *Assume that a function $f(x)$ and its derivatives up to $2L$th order satisfy the polynomial growth condition* (2.7). *Let X_N be obtained by the Euler scheme* (2.9) *applied to the linear SDE* (2.22) *and $A(L,N)\varphi$ be the sparse grid rule* (2.4) *with level L applied to the integral corresponding to $\mathbb{E}f(X_N)$ as in* (2.15). *Then for any L and sufficiently small $h > 0$*

$$|\mathbb{E}f(X_N) - A(L,N)\varphi| \le K\varepsilon^{2L}(1 + e^{\lambda(2L+\varkappa)T})\Big(1 + (3c/2)^{L\wedge N}\Big)\beta^{-(L\wedge N)/2}T^L, \tag{2.31}$$

where $K > 0$ is independent of h, L, and N; c and β are from (2.29); *$\varkappa$ is from* (2.7).

Proof. We recall (see (2.15)) that

$$\mathbb{E}f(X_N) = I_N\varphi = \frac{1}{(2\pi)^{N/2}} \int_{\mathbb{R}^{rN}} \varphi(y_1,\dots,y_N) \exp\left(-\frac{1}{2}\sum_{i=1}^N y_i^2\right) dy.$$

Introduce the integrals

$$I_1^{(k)}\varphi = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \varphi(y_1,\dots,y_k,\dots,y_N) \exp\left(-\frac{y_k^2}{2}\right) dy_k, \quad k = 1,\dots,N, \tag{2.32}$$

and their approximations $Q_n^{(k)}$ by the corresponding one-dimensional Gauss–Hermite quadratures with n nodes. Also, let $\mathcal{U}_{i_k}^{(k)} = Q_{i_k}^{(k)} - Q_{i_k-1}^{(k)}$.

Using (2.4) and the recipe from the proof of Lemma 3.4 in [37], we obtain

$$I_N\varphi - A(L,N)\varphi = \sum_{l=2}^N S(L,l) \otimes_{k=l+1}^N I_1^{(k)}\varphi + (I_1^{(1)} - Q_L^{(1)}) \otimes_{k=2}^N I_1^{(k)}\varphi, \tag{2.33}$$

where

$$S(L,l) = \sum_{i_1+\cdots+i_{l-1}+i_l = L+l-1} \otimes_{k=1}^{l-1} \mathcal{U}_{i_k}^{(k)} \otimes (I_1^{(l)} - Q_{i_l}^{(l)}). \tag{2.34}$$

Due to (2.28), we have for $n > 1$ and $1 \le \gamma \le 2(n-1)$

$$\mathcal{U}_n\psi = Q_n\psi - Q_{n-1}\psi = [Q_n\psi - I_1(\psi)] - [Q_{n-1}\psi - I_1(\psi)] = \int_{\mathbb{R}} \frac{d^\gamma}{dy^\gamma}\psi(y)[R_{n,\gamma}(\Gamma_{y,\gamma}) - R_{n-1,\gamma}(\Gamma_{y,\gamma})]\,dy, \tag{2.35}$$

and for $n = 1$

$$\mathcal{U}_n\psi = Q_1\psi - Q_0\psi = Q_1\psi = \psi(0). \tag{2.36}$$

By (2.34), (2.32), and (2.28), we obtain for the first term in the right-hand side of (2.33)

$$\begin{aligned} S(L,l)\otimes_{n=l+1}^N I_1^{(n)}\varphi &= \sum_{i_1+\cdots+i_l=L+l-1} \otimes_{n=1}^{l-1}\mathcal{U}_{i_k}^{(k)} \otimes (I_1^{(l)} - Q_{i_l}^{(l)}) \otimes_{n=l+1}^N I_1^{(n)}\varphi \\ &= \sum_{i_1+\cdots+i_l=L+l-1} \otimes_{n=1}^{l-1}\mathcal{U}_{i_k}^{(k)} \otimes (I_1^{(l)} - Q_{i_l}^{(l)}) \\ &\quad \otimes \int_{\mathbb{R}^{N-l}} \varphi(y)\frac{1}{(2\pi)^{(N-l)/2}}\exp\left(-\sum_{k=l+1}^N \frac{y_k^2}{2}\right) dy_{l+1}\dots dy_N \\ &= -\sum_{i_1+\cdots+i_l=L+l-1} \otimes_{n=1}^{l-1}\mathcal{U}_{i_k}^{(k)} \otimes \int_{\mathbb{R}^{N-l+1}} \frac{d^{2i_l}}{dy_l^{2i_l}}\varphi(y)R_{i_l,2i_l}(\Gamma_{y_l,2i_l}) \\ &\quad \times\frac{1}{(2\pi)^{(N-l)/2}}\exp\left(-\sum_{k=l+1}^N \frac{y_k^2}{2}\right) dy_l\dots dy_N. \end{aligned}$$

Now consider two cases: if $i_{l-1} > 1$ then by (2.35)

$$\begin{aligned} &S(L,l)\otimes_{n=l+1}^N I_1^{(n)}\varphi \\ &= -\sum_{i_1+\cdots+i_l=L+l-1} \otimes_{n=1}^{l-2}\mathcal{U}_{i_k}^{(k)} \otimes \int_{\mathbb{R}^{N-l+2}} \frac{d^{2i_{l-1}-2}}{dy_{l-1}^{2i_{l-1}-2}}\frac{d^{2i_l}}{dy_l^{2i_l}}\varphi(y)R_{i_l,2i_l}(\Gamma_{y_l,2i_l}) \\ &\quad \times[R_{i_{l-1},2i_{l-1}-2}(\Gamma_{y_{l-1},2i_{l-1}-2}) - R_{i_{l-1}-1,2i_{l-1}-2}(\Gamma_{y_{i_{l-1}},2i_{l-1}-2})] \\ &\quad \times\frac{1}{(2\pi)^{(N-l)/2}}\exp\left(-\sum_{k=l+1}^N \frac{y_k^2}{2}\right) dy_{l-1}\dots dy_N; \end{aligned}$$

otherwise (i.e., if $i_{l-1} = 1$) by (2.36)

$$\begin{aligned} &S(L,l)\otimes_{n=l+1}^N I_1^{(n)}\varphi \\ &= -\sum_{i_1+\cdots+i_l=L+l-1} \otimes_{n=1}^{l-2}\mathcal{U}_{i_k}^{(k)} \otimes \int_{\mathbb{R}^{N-l+1}} Q_1^{(l-1)}\frac{d^{2i_l}}{dy_l^{2i_l}}\varphi(y)R_{i_l,2i_l}(\Gamma_{y_l,2i_l}) \\ &\quad \times\frac{1}{(2\pi)^{(N-l)/2}}\exp\left(-\sum_{k=l+1}^N \frac{y_k^2}{2}\right) dy_l\dots dy_N. \end{aligned}$$

Repeating the above process for $i_{l-2},\dots,i_1$, we obtain

(2.37)
$$\begin{aligned}
&S(L,l)\otimes_{n=l+1}^{N} I_1^{(n)}\varphi \\
&\quad=\sum_{i_1+\cdots+i_l=L+l-1}\int_{\mathbb{R}^{N-\#F_{l-1}}}[\otimes_{m\in F_{l-1}}Q_1^{(m)}D^{2\alpha_l}\varphi(y)] \\
&\qquad\times\mathcal{R}_{l,\alpha_l}(y_1,\dots,y_l)\frac{1}{(2\pi)^{(N-l)/2}}\exp\left(-\sum_{k=l+1}^{N}\frac{y_k^2}{2}\right)\prod_{n\in G_{l-1}}dy_n\times dy_l\dots dy_N,
\end{aligned}$$

where the multi-index $\alpha_l=(i_1-1,\dots,i_{l-1}-1,i_l,0,\dots,0)$ with the mth element α_l^m, the sets $F_{l-1}=F_{l-1}(\alpha_l)=\{m:\ \alpha_l^m=0,\ m=1,\dots,l-1\}$, and $G_{l-1}=G_{l-1}(\alpha_l)=\{m:\ \alpha_l^m>0,\ m=1,\dots,l-1\}$, the symbols $\#F_{l-1}$ and $\#G_{l-1}$ stand for the number of elements in the corresponding sets, and

$$\begin{aligned}
\mathcal{R}_{l,\alpha_l}(y_1,\dots,y_l)&=-R_{i_l,2i_l}(\Gamma_{y_l,2i_l})\\
&\quad\otimes_{n\in G_{l-1}}[R_{i_n,2i_n-2}(\Gamma_{y_n,2i_n-2})-R_{i_n-1,2i_n-2}(\Gamma_{y_n,2i_n-2})].
\end{aligned}$$

Note that $\#G_{l-1}\le(L-1)\wedge(l-1)$ and also recall that $i_j\ge1$, $j=1,\dots,l$. Using (2.29), (2.30), and the inequality

$$\prod_{n\in G_{l-1}}[i_n^{-(i_n-1)}+(i_n-1)^{-(i_n-1)}]i_l^{-i_l}\le(3/2)^{\#G_{l-1}},$$

we get

$$\begin{aligned}
|\mathcal{R}_{l,\alpha}(y_1,\dots,y_l)|&\le\prod_{n\in G_{l-1}}[i_n^{-(i_n-1)}+(i_n-1)^{-(i_n-1)}]i_l^{-i_l}\frac{c^{\#G_{l-1}+1}}{(2\pi)^{(\#G_{l-1}+1)/2}}\\
&\quad\times\exp\left(-\sum_{n\in G_{l-1}}\frac{\beta y_n^2}{2}-\frac{\beta y_l^2}{2}\right)\\
&\le\frac{(3c/2)^{\#G_{l-1}+1}}{(2\pi)^{(\#G_{l-1}+1)/2}}\exp\left(-\sum_{n\in G_{l-1}}\frac{\beta y_n^2}{2}-\frac{\beta y_l^2}{2}\right).
\end{aligned}\tag{2.38}$$

Substituting (2.38) in (2.37), we arrive at

$$\begin{aligned}
&\left|S(L,l)\otimes_{n=l+1}^{N}I_1^{(n)}\varphi\right|\\
&\quad\le\sum_{i_1+\cdots+i_l=L+l-1}\frac{(3c/2)^{\#G_{l-1}+1}}{(2\pi)^{(N-\#F_{l-1})/2}}\int_{\mathbb{R}^{N-\#F_{l-1}}}\left|\otimes_{m\in F_{l-1}}Q_1^{(m)}D^{2\alpha_l}\varphi(y)\right|\\
&\qquad\times\exp\left(-\sum_{n\in G_{l-1}}\frac{\beta y_n^2}{2}-\frac{\beta y_l^2}{2}-\sum_{k=l+1}^{N}\frac{y_k^2}{2}\right)\prod_{n\in G_{l-1}}dy_n\times dy_l\dots dy_N.
\end{aligned}\tag{2.39}$$

Using (2.25) and the assumption that $|\frac{d^{2L}}{dx^{2L}}f(x)|\le K(1+|x|^{\varkappa})$ for some $K>0$ and $\varkappa\ge1$, we get

$$\begin{aligned}
\left|D^{2\alpha_l}\varphi(y)\right|&=\varepsilon^{2L}h^L\left|\frac{d^{2L}}{dx^{2L}}f(\bar X(N,y))\right|(1+\lambda h)^{2LN-2\sum_{i=1}^{l}i\alpha_l^i}\\
&\le K\varepsilon^{2L}h^L(1+\lambda h)^{2LN-2\sum_{i=1}^{l}i\alpha_l^i}(1+|\bar X(N,y)|^{\varkappa}).
\end{aligned}\tag{2.40}$$

Substituting (2.40) and (2.24) in (2.39) and doing further calculations, we obtain

$$
\begin{aligned}
(2.41)\qquad &\left|S(L,l)\otimes_{n=l+1}^{N} I_1^{(n)}\varphi\right| \\
&\le K\varepsilon^{2L}h^L(1+e^{\lambda\varkappa T})(1+(3c/2)^{L\wedge l})\beta^{-(L\wedge l)/2} \\
&\quad\times \sum_{i_1+\cdots+i_l=L+l-1} (1+\lambda h)^{2LN-2\sum_{i=1}^{l} i\alpha_l^i} \\
&\le K\varepsilon^{2L}h^L(1+e^{\lambda(2L+\varkappa)T})(1+(3c/2)^{L\wedge l})\beta^{-(L\wedge l)/2}\binom{L+l-2}{L-1} \\
&\le K\varepsilon^{2L}h^L(1+e^{\lambda(2L+\varkappa)T})(1+(3c/2)^{L\wedge l})\beta^{-(L\wedge l)/2} l^{L-1}
\end{aligned}
$$

with a new $K>0$ which does not depend on h, ε, L, c, β, and l. In the last line of (2.41) we used

$$
\binom{L+l-2}{L-1}=\prod_{i=1}^{L-1}\left(1+\frac{l-1}{i}\right)\le\left[\frac{1}{L-1}\sum_{i=1}^{L-1}\left(1+\frac{l-1}{i}\right)\right]^{L-1}\le l^{L-1}.
$$

Substituting (2.41) in (2.33) and observing that $|(I_1^{(1)}-Q_L^{(1)})\otimes_{k=2}^{N} I_1^{(k)}\varphi|$ is of order $O(h^L)$, we arrive at (2.31). ∎

Remark 2.6. Due to Examples 2.3 and 2.4, the error estimate (2.31) proved in Proposition 2.5 is quite sharp and we conclude that in general the SGC algorithm for weak approximation of SDE does not converge with either decrease of time step h or with increase of the level L. At the same time, the algorithm is convergent in L (when $L\le N$) if ε^2T is sufficiently small and SDE has some stable behavior (e.g., $\lambda\le 0$). Furthermore, the algorithm is sufficiently accurate when noise intensity ε and integration time T are relatively small.

Remark 2.7. It follows from the proof (see (2.40)) that if $\frac{d^{2L}}{dx^{2L}}f(x)=0$ then the error $I_N(\varphi)-A(L,N)\varphi=0$. We emphasize that this is a feature of the linear SDE (2.22) thanks to (2.25), while in the case of nonlinear SDE this error remains of the form (2.31) even if the $2L$th derivative of f is zero. See also the discussion at the end of Example 2.3 and numerical tests in Example 4.1.

Remark 2.8. We note that it is possible to prove a proposition analogous to Proposition 2.5 for a more general SDE, e.g., for SDE with additive noise. Since such a proposition does not add further information to our discussion of the use of SGC and its proof is more complex than in the case of (2.22), we do not consider such a proposition here.

3. Recursive collocation algorithm for linear SPDE. In the previous section we have demonstrated the limitations of SGC algorithms in application to SDE that, in general, such an algorithm will not work unless integration time T and noise magnitude are small. It is not difficult to understand that SGC algorithms have the same limitations in the case of SPDE as well, which, in particular, is demonstrated in Example 4.2, where a stochastic Burgers equation is considered. To cure this deficiency and achieve longer time integration in the case of linear SPDE, we will exploit the idea of the recursive approach proposed in [23, 50] in the case of a Wiener chaos expansion method. To this end, we apply the algorithm of SGC accompanied by a time discretization of SPDE over a small interval $[(k-1)h,kh]$ instead of the whole interval $[0,T]$ as we did in the previous section and build a recursive scheme to compute the second-order moments of the solutions to linear SPDE.

Consider the following linear SPDE in Ito's form:

(3.1)

$$du(t,x) = [\mathcal{L}u(t,x) + f(x)]\,dt + \sum_{l=1}^{r} [\mathcal{M}_l u(t,x) + g_l(x)]\,dw_l(t),\ (t,x) \in (0,T] \times \mathcal{D},$$
$$u(0,x) = u_0(x), \quad x \in \mathcal{D},$$

where $\mathcal{D}$ is an open domain in $\mathbb{R}^m$ and $(w(t), \mathcal{F}_t)$ is a Wiener process as in (2.5), and

$$\mathcal{L}u(t,x) = \sum_{i,j=1}^{m} a_{ij}(x) \frac{\partial^2}{\partial x_i \partial x_j} u(t,x) + \sum_{i=1}^{m} b_i(x) \frac{\partial}{\partial x_i} u(t,x) + c(x)\, u(t,x), \tag{3.2}$$
$$\mathcal{M}_l u(t,x) = \sum_{i=1}^{m} \alpha_i^l(x) \frac{\partial}{\partial x_i} u(t,x) + \beta^l(x)\, u(t,x).$$

We assume that $\mathcal{D}$ is either bounded with regular boundary or that $\mathcal{D} = \mathbb{R}^m$. In the former case we consider periodic boundary conditions and in the latter the Cauchy problem. We also assume that the coefficients of the operators $\mathcal{L}$ and $\mathcal{M}$ are uniformly bounded and

$$\tilde{\mathcal{L}} := \mathcal{L} - \frac{1}{2} \sum_{1 \le l \le r} \mathcal{M}_l \mathcal{M}_l$$

is nonnegative definite. When the coefficients of $\mathcal{L}$ and $\mathcal{M}$ are sufficiently smooth, existence and uniqueness results for the solution of (3.1)–(3.2) are available, e.g., in [42] and under weaker assumptions; see, e.g., [28, 25].

We will continue to use the notation from the previous section: h is a step of uniform discretization of the interval $[0,T]$, $N = T/h$, and $t_k = kh$, $k = 0, \ldots, N$. We apply the trapezoidal rule in time to the SPDE (3.1):

$$u^{k+1}(x) = u^k(x) + h \left[\tilde{\mathcal{L}} u^{k+1/2}(x) - \frac{1}{2} \sum_{l=1}^{r} \mathcal{M}_l g_l(x) + f(x) \right] \tag{3.3}$$
$$+ \sum_{l=1}^{r} \left[\mathcal{M}_l u^{k+1/2}(x) + g_l(x) \right] \sqrt{h}\, (\xi_{lh})_{k+1},\ x \in \mathcal{D},$$
$$u^0(x) = u_0(x),$$

where $u^k(x)$ approximates $u(t_k, x)$, $u^{k+1/2} = (u^{k+1} + u^k)/2$, and $(\xi_{lh})_k$ are i.i.d. random variables so that

$$\xi_h = \begin{cases} \xi,\ |\xi| \le A_h, \\ A_h,\ \xi > A_h, \\ -A_h,\ \xi < -A_h, \end{cases} \tag{3.4}$$

with $\xi \sim \mathcal{N}(0,1)$ and $A_h = \sqrt{2p|\ln h|}$ with $p \ge 1$. We note that the cut-off of the Gaussian random variables is needed in order to ensure that the implicitness of (3.3) does not lead to nonexistence of the second moment of $u^k(x)$ [29, 31]. Based on the standard results of numerics for SDE [31], it is natural to expect that under some regularity assumptions on the coefficients and the initial condition of (3.1), the approximation $u^k(x)$ from (3.3) converges with order 1/2 in the mean square sense

and with order 1 in the weak sense and in the latter case one can use discrete random variables $\zeta_{l,k+1}$ from (2.12) instead of $(\xi_{lh})_{k+1}$ (see also, e.g., [10, 17, 20] but we are not proving such a result here).

In what follows it will be convenient to also use the notation: $u_H^k(x;\phi(\cdot)) = u_H^k(x;\phi(\cdot);(\xi_{lh})_k, l=1,\ldots,r)$ for the approximation (3.3) of the solution $u(t_k,x)$ to the SPDE (3.1) with $f(x)=0$ and $g_l(x)=0$ for all l (homogeneous SPDE) and with the initial condition $\phi(\cdot)$ prescribed at time

$$t = t_{k-1}; u_O^k(x) = u_O^k(x;(\xi_{lh})_k, l=1,\ldots,r)$$

for the approximation (3.3) of the solution $u(t_k,x)$ to the SPDE (3.1) with the initial condition $\phi(x)=0$ prescribed at time $t=t_{k-1}$. Note that $u_O^k(x)=0$ if $f(x)=0$ and $g_l(x)=0$ for all l.

Let $\{e_i\}=\{e_i(x)\}_{i\geq 1}$ be a complete orthonormal system (CONS) in $L^2(\mathcal{D})$ with boundary conditions satisfied and $(\cdot,\cdot)$ be the inner product in that space. Then we can write

$$u^{k-1}(x) = \sum_{i=1}^{\infty} c_i^{k-1} e_i(x) \tag{3.5}$$

with $c_i^{k-1} = (u^{k-1}, e_i)$ and, due to the SPDE's linearity,

$$u^k(x) = u_O^k(x) + \sum_{i=1}^{\infty} c_i^{k-1} u_H^k(x; e_i(\cdot)).$$

We have

$$c_l^0 = (u_0, e_l), \quad c_l^k = q_{Ol}^k + \sum_{i=1}^{\infty} c_i^{k-1} q_{Hli}^k, \quad l=1,2,\ldots, \quad k=1,\ldots,N,$$

where $q_{Ol}^k = (u_O^k, e_l)$ and $q_{Hli}^k = (u_H^k(\cdot;e_i), e_l(\cdot))$.

Using (3.5), we represent the second moment of the approximation $u^k(x)$ from (3.3) of the solution $u(t_k,x)$ to the SPDE (3.1) as follows:

$$\mathbb{E}[u^k(x)]^2 = \sum_{i,j=1}^{\infty} C_{ij}^k e_i(x) e_j(x), \tag{3.6}$$

where the covariance matrix $C_{ij}^k = \mathbb{E}[c_i^k c_j^k]$. Introducing also the means M_i^k, one can obtain the recurrent relations in k :

(3.7)

$$\begin{aligned}
&M_i^0 = c_i^0 = (u_0, e_i), \quad C_{ij}^0 = c_i^0 c_j^0,\\
&M_i^k = \mathbb{E}[q_{Oi}^k] + \sum_{l=1}^{\infty} M_l^{k-1} \mathbb{E}[q_{Hil}^k],\\
&C_{ij}^k = \mathbb{E}[q_{Oi}^k q_{Oj}^k] + \sum_{l=1}^{\infty} M_l^{k-1} \left(\mathbb{E}[q_{Oi}^k q_{Hjl}^k] + \mathbb{E}[q_{Oj}^k q_{Hil}^k]\right) + \sum_{l,p=1}^{\infty} C_{lp}^{k-1} \mathbb{E}[q_{Hil}^k q_{Hjp}^k],\\
&i,j = 1,2,\ldots, \quad k=1,\ldots,N.
\end{aligned}$$

Since the coefficients of the SPDE (3.1) are time independent, all the expectations involving the quantities q_{Oi}^k and q_{Hil}^k in (3.7) do not depend on k and hence it is

sufficient to compute them just once, on a single step $k = 1$, and we get

$$
\begin{aligned}
(3.8)\ & M_i^0 = c_i^0 = (u_0, e_i), \quad C_{ij}^0 = c_i^0 c_j^0, \\
& M_i^k = \mathbb{E}[q_{Oi}^1] + \sum_{l=1}^{\infty} M_l^{k-1} \mathbb{E}[q_{Hil}^1], \\
& C_{ij}^k = \mathbb{E}[q_{Oi}^1 q_{Oj}^1] + \sum_{l=1}^{\infty} M_l^{k-1} \left(\mathbb{E}[q_{Oi}^1 q_{Hjl}^1] + \mathbb{E}[q_{Oj}^1 q_{Hil}^1] \right) + \sum_{l,p=1}^{\infty} C_{lp}^{k-1} \mathbb{E}[q_{Hil}^1 q_{Hjp}^1], \\
& i, j = 1, 2, \ldots, \quad k = 1, \ldots, N.
\end{aligned}
$$

These expectations can be approximated by quadrature rules from section 2.1. If the number of noises r is small, then it is natural to use the tensor product rule (2.3) with one-dimensional Gauss–Hermite quadratures of order $n = 2$ or 3 (note that when $r = 1$, we can use just a one-dimensional Gauss–Hermite quadrature of order $n = 2$ or 3). If the number of noises r is large then it might be beneficial to use the sparse grid quadrature (2.4) of level $L = 2$ or 3. More specifically,

$$
\begin{aligned}
(3.9)\qquad & \mathbb{E}[q_{Oi}^1] \doteq \sum_{p=1}^{\eta} (u_O^1(\cdot; \mathrm{y}_p), e_i(\cdot)) \mathsf{W}_p, \quad \mathbb{E}[q_{Hil}^1] \doteq \sum_{p=1}^{\eta} (u_H^1(\cdot; e_l; \mathrm{y}_p), e_i(\cdot)) \mathsf{W}_p, \\
& \mathbb{E}[q_{Oi}^1 q_{Oj}^1] \doteq \sum_{p=1}^{\eta} (u_O^1(\cdot; \mathrm{y}_p), e_i(\cdot)) (u_O^1(\cdot; \mathrm{y}_p), e_j(\cdot)) \mathsf{W}_p, \\
& \mathbb{E}[q_{Oi}^1 q_{Hjl}^1] \doteq \sum_{p=1}^{\eta} (u_O^1(\cdot; \mathrm{y}_p), e_i(\cdot)) (u_H^1(\cdot; e_l; \mathrm{y}_p), e_j(\cdot)) \mathsf{W}_p, \\
& \mathbb{E}[q_{Hil}^1 q_{Hjk}^1] \doteq \sum_{p=1}^{\eta} (u_H^1(\cdot; e_l; \mathrm{y}_p), e_i(\cdot)) (u_H^1(\cdot; e_k; \mathrm{y}_p), e_j(\cdot)) \mathsf{W}_p,
\end{aligned}
$$

where $\mathrm{y}_p \in \mathbb{R}^r$ are nodes of the quadrature, W_p are the corresponding quadrature weights, and $\eta = n^r$ in the case of the tensor-product rule (2.3) with one-dimensional Gauss–Hermite quadratures of order n or η is the total number of nodes $\#S$ used by the sparse grid quadrature (2.4) of level L. To find $u_O^1(x; \mathrm{y}_p)$ and $u_H^1(x; e_l; \mathrm{y}_p)$, we need to solve the corresponding elliptic PDE problems, which we do using the spectral method in physical space, i.e., using a truncation of the CONS $\{e_l\}_{l=1}^{l_*}$ to represent the numerical solution.

To summarize, we formulate the following deterministic recursive algorithm for the second-order moments of the solution to the SPDE problem (3.1).

ALGORITHM 3.1. *Choose the algorithm's parameters: a complete orthonormal basis $\{e_l(x)\}_{l \geq 1}$ in $L^2(\mathcal{D})$ and its truncation $\{e_l(x)\}_{l=1}^{l_*}$; a time step size h; and a quadrature rule (i.e., nodes y_p and the quadrature weights W_p, $p = 1, \ldots, \eta$).*

Step 1. *For each $p = 1, \ldots, \eta$ and $l = 1, \ldots, l_*$, find approximations $\bar{u}_O^1(x; \mathrm{y}_p) \approx u_O^1(x; \mathrm{y}_p)$ and $\bar{u}_H^1(x; e_l; \mathrm{y}_p) \approx u_H^1(x; e_l; \mathrm{y}_p)$ using the spectral method in physical space.*

Step 2. *Using the quadrature rule, approximately find the expectations as in* (3.9) *but with the approximate $\bar{u}_O^1(x; \mathrm{y}_p)$ and $\bar{u}_H^1(x; e_l; \mathrm{y}_p)$ instead of $u_O^1(x; \mathrm{y}_p)$ and $u_H^1(x; e_l; \mathrm{y}_p)$, respectively.*

Step 3. *Recursively compute the approximations of the means M_i^k, $i = 1, \ldots, l_*$, and covariance matrices $\{C_{ij}^k,\ i, j = 1, \ldots, l_*\}$ for $k = 1, \ldots, N$ according to* (3.8) *with the approximate expectations found in Step* 2 *instead of the exact ones.*

Step 4. *Compute the approximation of the second-order moment* $\mathbb{E}[u^k(x)]^2$ *using* (3.6) *with the approximate covariance matrix found in Step* 3 *instead of the exact one* $\{C_{ij}^k\}$.

We emphasize that Algorithm 3.1 for computing moments does not have a statistical error. Based on the error estimate in Proposition 2.5, we expect the one-step error of SGC for our recursive algorithm is of order h^L. Hence, we expect the total global error from trapezoidal rule in time and SGC to be $O(h) + O(h^{L-1})$. Error analysis of this algorithm will be considered elsewhere.

Remark 3.2. Algorithms analogous to Algorithm 3.1 can also be constructed based on other time-discretization methods than the trapezoidal rule used here or based on other types of SPDE approximations, e.g., one can exploit the Wong–Zakai approximation.

Remark 3.3. The cost of this algorithm is, similar to the algorithm in [50], $\frac{T}{\Delta}\eta l_*^4$ and the storage is ηl_*^2. The total cost can be reduced by employing some reduced-order methods in physical space and be proportional to l_*^2 instead of l_*^4. The discussion on computational efficiency of the recursive Wiener chaos method is also valid here; see [50, Remark 4.1].

Remark 3.4. Choosing an orthonormal basis is an important topic in the research of spectral methods, which can be found in [16] and many subsequent works. Here we choose a Fourier basis for (3.1) because of periodic boundary conditions.

4. Numerical experiments. In this section we illustrate via three examples how the SGC algorithms can be used for the weak-sense approximation of SDE and SPDE. The first example is a scalar SDE with multiplicative noise, where we show that the SGC algorithm's error is small when the noise magnitude is small. We also observe that when the noise magnitude is large, the SGC algorithm does not work well. In the second example we demonstrate that the SGC can be successfully used for simulating the Burgers equation with additive noise when the integration time is relatively small. In the last example we show that the recursive algorithm from section 3 works effectively for computing moments of the solution to an advection-diffusion equation with multiplicative noise over a longer integration time.

In all the tests we limit the dimension of random spaces to 40, which is an empirical limitation of the SGC of Smolyak on the dimensionality [41]. Also, we take the sparse grid level as less than or equal to five in order to avoid an excessive number of sparse grid points. All the tests were run using MATLAB R2012b on a Macintosh desktop computer with Intel Xeon CPU E5462 (quad-core, 2.80 GHz).

Example 4.1 (modified Cox–Ingersoll–Ross; see, e.g., [7]). Consider the Ito SDE

$$dX = -\theta_1 X\,dt + \theta_2\sqrt{1+X^2}\,dw(t), \quad X(0) = x_0. \tag{4.1}$$

For $\theta_2^2 - 2\theta_1 \neq 0$, the first two moments of $X(t)$ are equal to

$$\mathbb{E}X(t) = x_0\exp(-\theta_1 t), \quad \mathbb{E}X^2(t) = -\frac{\theta_2^2}{\theta_2^2 - 2\theta_1} + \left(x_0^2 + \frac{\theta_2^2}{\theta_2^2 - 2\theta_1}\right)\exp((\theta_2^2 - 2\theta_1)t).$$

In this example we test the SGC algorithms based on the Euler scheme (2.8) and on the second-order weak scheme (2.21). We compute the first two moments of the SDE's solution and use the relative errors to measure the accuracy of the algorithms as

$$\rho_1^r(T) = \frac{|\mathbb{E}X(T) - \mathbb{E}X_N|}{|\mathbb{E}X(T)|}, \quad \rho_2^r(T) = \frac{\left|\mathbb{E}X^2(T) - \mathbb{E}X_N^2\right|}{\mathbb{E}X^2(T)}. \tag{4.2}$$

TABLE 2

Comparison of the SGC algorithms based on the Euler scheme (left) and on the second-order scheme (2.21) *(right). The parameters of the model* (4.1) *are* $x_0 = 0.1$, $\theta_1 = 1$, $\theta_2 = 0.3$, *and* $T = 1$.

h	L	$\rho_1^r(1)$	order	$\rho_2^r(1)$	order	L	$\rho_1^r(1)$	order	$\rho_2^r(1)$	order
5×10^{-1}	2	3.20×10^{-1}	–	3.72×10^{-1}	–	3	$\mathbf{6.05\times10^{-2}}$	–	8.52×10^{-2}	–
2.5×10^{-1}	2	1.40×10^{-1}	1.2	1.40×10^{-1}	1.4	3	1.14×10^{-2}	2.4	2.10×10^{-2}	2.0
1.25×10^{-1}	2	$\mathbf{6.60\times10^{-2}}$	1.1	4.87×10^{-2}	1.5	3	1.75×10^{-3}	2.7	6.73×10^{-3}	1.6
6.25×10^{-2}	2	3.21×10^{-2}	1.0	8.08×10^{-3}	2.6	4	3.64×10^{-4}	2.3	1.21×10^{-3}	2.5
3.125×10^{-2}	2	1.58×10^{-2}	1.0	1.12×10^{-2}	−0.5	4	8.48×10^{-4}	−1.2	3.75×10^{-4}	1.7
2.5×10^{-2}	2	1.26×10^{-2}		1.49×10^{-2}		2	9.02×10^{-4}		5.72×10^{-2}	
2.5×10^{-2}	3	1.26×10^{-2}		1.48×10^{-2}		3	9.15×10^{-5}		2.84×10^{-3}	
2.5×10^{-2}	4	1.26×10^{-2}		1.55×10^{-2}		4	1.06×10^{-4}		2.77×10^{-4}	
2.5×10^{-2}	5	1.26×10^{-2}		1.56×10^{-2}		5	1.06×10^{-4}		1.81×10^{-4}	

TABLE 3

Comparison of the SGC algorithms based on the Euler scheme (left) and on the second-order scheme (2.21) *(right). The parameters of the model* (4.1) *are* $x_0 = 0.08$, $\theta_1 = -1$, $\theta_2 = 2$, *and* $T = 1$. *The sparse grid level* $L = 4$.

h	$\rho_1^r(1)$	order	$\rho_2^r(1)$	$\rho_1^r(1)$	$\rho_2^r(1)$
5×10^{-1}	1.72×10^{-1}	–	9.61×10^{-1}	2.86×10^{-2}	7.69×10^{-1}
2.5×10^{-1}	1.02×10^{-1}	0.8	8.99×10^{-1}	8.62×10^{-3}	6.04×10^{-1}
1.25×10^{-1}	5.61×10^{-2}	0.9	7.87×10^{-1}	1.83×10^{-2}	7.30×10^{-1}
6.25×10^{-2}	2.96×10^{-2}	0.9	6.62×10^{-1}	3.26×10^{-2}	8.06×10^{-1}
3.125×10^{-2}	1.52×10^{-2}	1.0	5.64×10^{-1}	4.20×10^{-2}	8.40×10^{-1}

Table 2 presents the errors for the SGC algorithms based on the Euler scheme (left) and on the second-order scheme (2.21) (right), when the noise magnitude is small. For the parameters given in the table's description, the exact values (up to 4 decimal places) of the first and second moments are 3.679×10^{-2} and 4.162×10^{-2}, respectively. We see that increase of the SGC level L above 2 in the Euler scheme case and above 3 in the case of the second-order scheme does not improve accuracy. When the SGC error is relatively small in comparison with the error due to time discretization, we observe decrease of the overall error of the algorithms in h: proportional to h for the Euler scheme and to h^2 for the second-order scheme. We underline that in this experiment the noise magnitude is small.

In Table 3 we give results of the numerical experiment when the noise magnitude is not small. For the parameters given in the table's description, the exact values (up to 4 decimal places) of the first and second moments are 0.2718 and 272.3202, respectively. Though for the Euler scheme there is a proportional to h decrease of the error in computing the mean, there is almost no decrease of the error in the rest of this experiment. The large value of the second moment apparently affects efficiency of the SGC here. For the Euler scheme, increasing L and decreasing h can slightly improve accuracy in computing the second moment, e.g., the smallest relative error for the second moment is 56.88% when $h = 0.03125$ and $L = 5$ (this level requires 750337 sparse grid points) out of the considered cases of $h = 0.5$, 0.25, 0.125, 0.0625, and 0.03125 and $L \leq 5$. For the mean, increase of the level L from 2 to 3, 4, or 5 does not improve accuracy. For the second-order scheme (2.21), relative errors for the mean can be decreased by increasing L for a fixed h: e.g., for $h = 0.25$, the relative errors are 0.5121 0.1753, 0.0316, and 0.0086 when $L = 2$, 3, 4, and 5, respectively.

We also see in Table 3 that the SGC algorithm based on the second-order scheme may not admit higher accuracy than the one based on the Euler scheme, e.g., for $h = 0.5,\ 0,25,\ 0.125$ the second-order scheme yields higher accuracy while the Euler scheme demonstrates higher accuracy for smaller $h = 0.0625$ and 0.03125. Further decrease in h was not considered because this would lead to increase of the dimension of the random space beyond 40 where the sparse grid of Smolyak (2.4) may fail and the SGC algorithm may also lose its competitive edge with Monte Carlo-type techniques.

Via this example we have shown that the SGC algorithms based on first- and second-order schemes can produce sufficiently accurate results when noise magnitude is small and that the second-order scheme is preferable since for the same accuracy it uses random spaces of lower dimension than the first-order Euler scheme; compare, e.g., the error values highlighted by bold font in Table 2 and see also the discussion at the end of section 2.2. When the noise magnitude is large (see Table 3), the SGC algorithms do not work well as was predicted in section 2.3.

Example 4.2 (Burgers equation with additive noise). Consider the stochastic Burgers equation [8, 19]:

$$du + u\frac{\partial u}{\partial x}dt = \nu\frac{\partial^2 u}{\partial x^2}dt + \sigma\cos(x)dw, \quad 0 \le x \le \ell, \quad \nu > 0 \tag{4.3}$$

with the initial condition $u_0(x) = 2\nu\frac{2\pi}{\ell}\frac{\sin(\frac{2\pi}{\ell}x)}{a+\cos(\frac{2\pi}{\ell}x)}$, $a > 1$, and periodic boundary conditions. In the numerical tests the used values of the parameters are $\ell = 2\pi$ and $a = 2$.

Apply the Fourier collocation method in physical space and the trapezoidal rule in time to (4.3):

$$\frac{\vec{u}_{j+1} - \vec{u}_j}{h} - \nu D^2\frac{\vec{u}_{j+1} + \vec{u}_j}{2} = -\frac{1}{2}D\left(\frac{\vec{u}_{j+1} + \vec{u}_j}{2}\right)^2 + \sigma\Gamma\sqrt{h}\xi_j, \tag{4.4}$$

where $\vec{u}_j = (u(t_j, x_1), \dots, u(t_j, x_M))^\intercal$, $\quad t_j = jh$, D is the Fourier spectral differential matrix, ξ_j are i.i.d. $\mathcal{N}(0,1)$ random variables, and $\Gamma = (\cos(x_1), \dots, \cos(x_M))^\intercal$. The Fourier collocation points are $x_m = m\frac{\ell}{M}$ $(m = 1, \dots, M)$ in physical space and in the experiment we used $M = 100$. We aim at computing moments of $\vec{u}_j$, which are integrals with respect to the Gaussian measure corresponding to the collection of ξ_j, and we approximate these integrals using the SGC from section 2. The use of the SGC amounts to substituting ξ_j in (4.4) by sparse grid nodes, which results in a system of (deterministic) nonlinear equations of the form (4.4). To solve the nonlinear equations, we used the fixed-point iteration method with tolerance $h^2/100$.

The errors in computing the first and second moments are measured as follows:

$$\begin{aligned}
\rho_1^{r,2}(T) &= \frac{\|\mathbb{E}u_{\rm ref}(T,\cdot) - \mathbb{E}u_{\rm num}(T,\cdot)\|}{\|\mathbb{E}u_{\rm ref}(T,\cdot)\|}, \\
\rho_2^{r,2}(T) &= \frac{\left\|\mathbb{E}u^2_{\rm ref}(T,\cdot) - \mathbb{E}u^2_{\rm num}(T,\cdot)\right\|}{\|\mathbb{E}u^2_{\rm ref}(T,\cdot)\|}, \\
\rho_1^{r,\infty}(T) &= \frac{\|\mathbb{E}u_{\rm ref}(T,\cdot) - \mathbb{E}u_{\rm num}(T,\cdot)\|_\infty}{\|\mathbb{E}u_{\rm ref}(T,\cdot)\|_\infty}, \\
\rho_2^{r,\infty}(T) &= \frac{\left\|\mathbb{E}u^2_{\rm ref}(T,\cdot) - \mathbb{E}u^2_{\rm num}(T,\cdot)\right\|_\infty}{\|\mathbb{E}u^2_{\rm ref}(T,\cdot)\|_\infty},
\end{aligned} \tag{4.5}$$

where $\|v(\cdot)\| = (\frac{2\pi}{M}\sum_{m=1}^M v^2(x_m))^{1/2}$, $\|v(\cdot)\|_\infty = \max_{1\le m\le M}|v(x_m)|$, x_m are the Fourier collocation points, and $u_{\rm num}$ and $u_{\rm ref}$ are the numerical solutions obtained

TABLE 4
Errors of the SGC algorithm applied to the stochastic Burgers equation (4.3) *with parameters* $T = 0.5$, $\nu = 0.1$, *and* $\sigma = 1$.

h	$\rho_1^{r,2}(0.5)$, $L=2$	$\rho_1^{r,2}(0.5)$, $L=3$	$\rho_2^{r,2}(0.5)$, $L=2$	$\rho_2^{r,2}(0.5)$, $L=3$	$\rho_2^{r,2}(0.5)$, $L=4$
2.5×10^{-1}	1.28×10^{-1}	1.3661×10^{-1}	4.01×10^{-2}	1.05×10^{-2}	1.25×10^{-2}
1.00×10^{-1}	4.70×10^{-2}	5.3874×10^{-2}	4.48×10^{-2}	4.82×10^{-3}	4.69×10^{-3}
5.00×10^{-2}	2.75×10^{-2}	2.7273×10^{-2}	4.73×10^{-2}	5.89×10^{-3}	2.82×10^{-3}
2.50×10^{-2}	2.51×10^{-2}	1.4751×10^{-2}	4.87×10^{-2}	6.92×10^{-3}	2.34×10^{-3}
1.25×10^{-2}	2.67×10^{-2}	9.4528×10^{-3}	4.95×10^{-2}	7.51×10^{-3}	2.29×10^{-3}

TABLE 5
Errors of the SGC algorithm applied to the stochastic Burgers equation (4.3) *with parameters* $\nu = 1$, $\sigma = 0.5$, *and* $T = 0.5$.

h	$\rho_1^{r,2}(0.5)$, $L=2$	$\rho_2^{r,2}(0.5)$, $L=2$	$\rho_2^{r,2}(0.5)$, $L=3$
2.5×10^{-1}	4.94×10^{-3}	8.75×10^{-3}	8.48×10^{-3}
1×10^{-1}	8.20×10^{-4}	1.65×10^{-3}	1.13×10^{-3}
5×10^{-2}	4.88×10^{-4}	1.18×10^{-3}	6.47×10^{-4}
2.5×10^{-2}	3.83×10^{-4}	1.08×10^{-3}	5.01×10^{-4}
1.25×10^{-2}	3.45×10^{-4}	1.07×10^{-3}	4.26×10^{-4}

by the SGC algorithm and the reference solution, respectively. The first and second moments of the reference solution u_{ref} were computed by the same solver in space and time (4.4) but accompanied by the Monte Carlo method with a large number of realizations ensuring that the statistical errors were negligible.

First, we choose $\nu = 0.1$ and $\sigma = 1$. We obtain the reference solution with $h = 10^{-4}$ and 1.92×10^6 Monte Carlo realizations. The corresponding statistical error is 1.004×10^{-3} for the mean (maximum of the statistical error for $\mathbb{E}u_{\text{ref}}(0.5, x_j)$) and 9.49×10^{-4} for the second moment (maximum of the statistical error for $\mathbb{E}u^2_{\text{ref}}(0.5, x_j)$) with 95% confidence interval, and the corresponding estimates of L^2-norm of the moments are $\|\mathbb{E}u_{\text{ref}}(0.5, \cdot)\| \doteq 0.18653$ and $\left\|\mathbb{E}u^2_{\text{ref}}(0.5, \cdot)\right\| \doteq 0.72817$. We see from the results of the experiment presented in Table 4 that for $L = 2$ the error in computing the mean decreases when h decreases up to $h = 0.05$ but the accuracy does not improve with further decrease of h. For the second moment, we observe no improvement in accuracy with decrease of h. For $L = 4$, the error in computing the second moment decreases with h. When $h = 0.0125$, increasing the sparse grid level improves the accuracy for the mean: $L = 3$ yields $\rho_1^{r,2}(0.5) \doteq 9.45 \times 10^{-3}$ and $L = 4$ yields $\rho_1^{r,2}(0.5) \doteq 8.34 \times 10^{-3}$. As seen in Table 4, increase of the level L also improves accuracy for the second moment when $h = 0.05$, 0.25, or 0.125.

Second, we choose $\nu = 1$ and $\sigma = 0.5$. We obtain the first two moments of the reference u_{ref} using $h = 10^{-4}$ and the Monte Carlo method with 3.84×10^6 realizations. The corresponding statistical error is 3.2578×10^{-4} for the mean and 2.2871×10^{-4} for the second moment with 95% confidence interval, and the corresponding estimates of L^2-norm of the moments are $\|\mathbb{E}u_{\text{ref}}(0.5, \cdot)\| \doteq 1.11198$ and $\|\mathbb{E}u^2_{\text{ref}}(0.5, \cdot)\| \doteq 0.66199$.

The results of the experiment are presented in Table 5. We see that accuracy is sufficiently high and there is some decrease of errors with decrease of time step h. However, as expected, no convergence in h is observed and further numerical tests (not presented here) showed that taking h smaller than 1.25×10^{-2} and level $L = 2$ or 3 does not improve accuracy. In additional experiments we also noticed that there

was no improvement of accuracy for the mean when we increased the level L up to 5. For the second moment, we observe some improvement in accuracy when L increases from 2 to 3 (see Table 5) but additional experiments (not presented here) showed that further increase of L (up to 5) does not reduce the errors.

For the errors measured in L^∞-norm (4.5) we had similar observations (not presented here) as in the case of L^2-norm.

In summary, this example has illustrated that SGC algorithms can produce accurate results in finding moments of solutions of nonlinear SPDE when the integration time is relatively small. Comparing Tables 4 and 5, we observe better accuracy for the first two moments when the magnitude of the noise is smaller. In some situations higher sparse grid levels L improve accuracy but dependence of errors on L is not monotone. No convergence in time step h and in level L was observed which is consistent with our theoretical prediction in section 2.

Example 4.3 (stochastic advection-diffusion equation). Consider the stochastic advection-diffusion equation in the Ito sense:

$$du = \left(\frac{\epsilon^2+\sigma^2}{2}\frac{\partial^2 u}{\partial x^2} + \beta\sin(x)\frac{\partial u}{\partial x}\right)dt + \sigma\frac{\partial u}{\partial x}\,dw(s), \quad (t,x)\in(0,T]\times(0,2\pi), \tag{4.6}$$
$$u(0,x) = \phi(x), \quad x\in(0,2\pi),$$

where $w(s)$ is a standard scalar Wiener process and $\epsilon \geq 0$, β, and σ are constants. In the tests we took $\phi(x) = \cos(x)$, $\beta = 0.1$, $\sigma = 0.5$, and $\epsilon = 0.2$.

We apply Algorithm 3.1 to (4.6) to compute the first two moments at a relatively large time $T = 5$. The Fourier basis was taken as CONS. Since (4.6) has a single noise only, we used one-dimensional Gauss–Hermite quadratures of order n. The implicitness due to the use of the trapezoidal rule was resolved by the fixed-point iteration with stopping criterion $h^2/100$.

As we have no exact solution of (4.6), we chose to find the reference solution by Algorithm 4.2 from [50] (a recursive Wiener chaos method accompanied by the trapezoidal rule in time and Fourier collocation method in physical space) with the parameters: the number of Fourier collocation points $M = 30$, the length of time subintervals for the recursion procedure $h = 10^{-4}$, the highest order of Hermite polynomials $P = 4$, the number of modes approximating the Wiener process $n = 4$, and the time step in the trapezoidal rule $h = 10^{-5}$. It gives the second moment in the L^2-norm $\|\mathbb{E}u^2_{\mathrm{ref}}(1,\cdot)\| \doteq 1.065195$. The errors are computed as follows:

$$\varrho_2^2(T) = \left|\left\|\mathbb{E}u^2_{\mathrm{ref}}(T,\cdot)\right\| - \left\|\mathbb{E}u^2_{\mathrm{numer}}(T,\cdot)\right\|\right|, \quad \varrho_2^{r,2}(T) = \frac{\varrho_2^2(T)}{\|\mathbb{E}u^2_{\mathrm{ref}}(T,\cdot)\|}, \tag{4.7}$$

where the norm is defined as in (4.5).

The results of the numerical experiment are given in Table 6. We observe first-order convergence in h for the second moments. We notice that increasing the quadrature order n from 2 to 3 does not improve accuracy which is expected. Indeed, the used trapezoidal rule is of weak order one in h in the case of multiplicative noise and a more accurate quadrature rule cannot improve the order of convergence. This observation confirms in some sense that the total error should be expected to be $O(h)+O(h^{L-1})$, as discussed in section 3. We note in passing that in the additive noise case we expect to see the second order convergence in h when $n = 3$ due to the properties of the trapezoidal rule.

In conclusion, we showed that recursive Algorithm 3.1 can work effectively for accurate computing of second moments of solutions to linear stochastic advection-

TABLE 6

Errors in computing the second moment of the solution to the stochastic advection-diffusion equation (4.6) *with* $\sigma = 0.5$, $\beta = 0.1$, $\epsilon = 0.2$ *at* $T = 5$ *by Algorithm* 3.1 *with* $l_* = 20$ *and the one-dimensional Gauss–Hermite quadrature of order* $n = 2$ *(left) and* $n = 3$ *(right).*

h	$\varrho_2^{r,2}(5)$	order	CPU time (sec.)	$\varrho_2^{r,2}(5)$	order	CPU time (sec.)
5×10^{-2}	1.01×10^{-3}	–	7.41	1.06×10^{-3}	–	1.10×10
2×10^{-2}	4.07×10^{-4}	1.0	1.65×10	4.25×10^{-4}	1.0	2.43×10
1×10^{-2}	2.04×10^{-4}	1.0	3.43×10	2.12×10^{-4}	1.0	5.10×10
5×10^{-3}	1.02×10^{-4}	1.0	6.81×10	1.06×10^{-4}	1.0	1.00×10^{2}
2×10^{-3}	4.08×10^{-5}	1.0	1.70×10^{2}	4.25×10^{-5}	1.0	2.56×10^{2}
1×10^{-3}	2.04×10^{-5}	1.0	3.37×10^{2}	2.12×10^{-5}	1.0	5.12×10^{2}

diffusion equations at relatively large time. We observed convergence of order one in h.

Acknowledgment. MVT is grateful to ICERM (Brown University, Providence, RI) for its hospitality.

REFERENCES

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

[2] I. Babuska, R. Tempone, and G. E. Zouraris, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, SIAM J. Numer. Anal., 42 (2004), pp. 800–825.

[3] M. Bieri and C. Schwab, *Sparse high order FEM for elliptic sPDEs*, Comput. Methods Appl. Mech. Engrg., 198 (2009), pp. 1149–1170.

[4] A. Budhiraja and G. Kallianpur, *Approximations to the solution of the Zakai equation using multiple Wiener and Stratonovich integral expansions*, Stochastics Stochastics Rep., 56 (1996), pp. 271–315.

[5] A. Budhiraja and G. Kallianpur, *The Feynman-Stratonovich semigroup and Stratonovich integral expansions in nonlinear filtering*, Appl. Math. Optim., 35 (1997), pp. 91–116.

[6] T. Cass and C. Litterer, *On the Error Estimate for Cubature on Wiener Space*, preprint, arXiv:1105.3996, 2011.

[7] F. Comte, V. Genon-Catalot, and Y. Rozenholc, *Penalized nonparametric mean square estimation of the coefficients of diffusion processes*, Bernoulli, 13 (2007), pp. 514–543.

[8] G. Da Prato, A. Debussche, and R. Temam, *Stochastic Burgers' equation*, NoDEA Nonlinear Differential Equations Appl., 1 (1994), pp. 389–402.

[9] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, 2nd ed., Comput. Sci. Appl. Math., Academic Press, Orlando, FL, 1984.

[10] A. Debussche, *Weak approximation of stochastic partial differential equations: The nonlinear case*, Math. Comp., 80 (2011), pp. 89–117.

[11] T. Gerstner, *Sparse Grid Quadrature Methods for Computational Finance*, Ph.D. thesis, Habilitation, University of Bonn, Bonn, Germany, 2007.

[12] T. Gerstner and M. Griebel, *Numerical integration using sparse grids*, Numer. Algorithms, 18 (1998), pp. 209–232.

[13] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York, 1991.

[14] M. B. Giles, *Multilevel Monte Carlo path simulation*, Oper. Res., 56 (2008), pp. 607–617.

[15] M. B. Giles, *Multilevel Monte Carlo methods*, in Monte Carlo and Quasi-Monte Carlo Methods 2012, Springer, Heidelberg, 2013, pp. 83–103.

[16] D. Gottlieb and S. A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, Reg. Conf. Ser. Appl. Math., SIAM, Philadelphia, 1977.

[17] W. Grecksch and P. E. Kloeden, *Time-discretised Galerkin approximations of parabolic stochastic PDEs*, Bull. Austral. Math. Soc., 54 (1996), pp. 79–85.

[18] M. Griebel and M. Holtz, *Dimension-wise integration of high-dimensional functions with applications to finance*, J. Complexity, 26 (2010), pp. 455–489.
[19] 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.
[20] P. E. Kloeden and S. Shott, *Linear-implicit strong schemes for Itô-Galerkin approximations of stochastic PDEs*, J. Appl. Math. Stochastic Anal., 14 (2001), pp. 47–53.
[21] F. Y. Kuo, C. Schwab, and I. H. Sloan, *Multi-level quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients*, SIAM J. Numer. Anal., 50 (2012), pp. 3351–3374.
[22] C. Litterer and T. Lyons, *High order recombination and an application to cubature on Wiener space*, Ann. Appl. Probab., 22 (2012), pp. 1301–1327.
[23] S. Lototsky, R. Mikulevicius, and B. L. Rozovskii, *Nonlinear filtering revisited: a spectral approach*, SIAM J. Control Optim., 35 (1997), pp. 435–461.
[24] 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.
[25] S. V. Lototsky and B. L. Rozovskii, *Wiener chaos solutions of linear stochastic evolution equations*, Ann. Probab., 34 (2006), pp. 638–662.
[26] T. Lyons and N. Victoir, *Cubature on Wiener space*, R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci., 460 (2004), pp. 169–198.
[27] G. Mastroianni and G. Monegato, *Error estimates for Gauss-Laguerre and Gauss-Hermite quadrature formulas*, in Approximation and Computation, Birkhäuser Verlag, Boston, 1994, pp. 421–434.
[28] R. Mikulevicius and B. Rozovskii, *Linear parabolic stochastic PDEs and Wiener chaos*, SIAM J. Math. Anal., 29 (1998), pp. 452–480.
[29] 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.
[30] G. N. Milstein and M. V. Tret'yakov, *Numerical methods in the weak sense for stochastic differential equations with small noise*, SIAM J. Numer. Anal., 34 (1997), pp. 2142–2167.
[31] G. N. Milstein and M. V. Tretyakov, *Stochastic Numerics for Mathematical Physics*, Springer-Verlag, Berlin, 2004.
[32] G. N. Milstein and M. V. Tretyakov, *Solving parabolic stochastic partial differential equations via averaging over characteristics*, Math. Comp., 78 (2009), pp. 2075–2106.
[33] M. Motamed, F. Nobile, and R. Tempone, *A stochastic collocation method for the second order wave equation with a discontinuous random speed*, Numer. Math., 123 (2012), pp. 493–536.
[34] T. Muller-Gronbach, K. Ritter, and L. Yaroslavtseva, *Derandomization of the Euler scheme for scalar stochastic differential equations*, J. Complexity, 28 (2012), pp. 139–153.
[35] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, CBMS-NSF Regional Conf. Ser. Appl. Math., SIAM, Philadelphia, 1992.
[36] F. Nobile and R. Tempone, *Analysis and implementation issues for the numerical approximation of parabolic equations with random coefficients*, Internat. J. Numer. Methods Engrg., 80 (2009), pp. 979–1006.
[37] F. Nobile, R. Tempone, and C. G. Webster, *An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal., 46 (2008), pp. 2411–2442.
[38] E. Novak and K. Ritter, *Simple cubature formulas with high polynomial exactness*, Constr. Approx., 15 (1999), pp. 499–522.
[39] G. Pagès and H. Pham, *Optimal quantization methods for nonlinear filtering with discrete-time observations*, Bernoulli, 11 (2005), pp. 893–932.
[40] G. Pagès and J. Printems, *Optimal quadratic quantization for numerics: The Gaussian case*, Monte Carlo Methods Appl., 9 (2003), pp. 135–165.
[41] K. Petras, *SMOLPACK: Multidimensional Quadrature Using Sparse Grids*, http://people.sc.fsu.edu/∼jburkardt/c_src/smolpack/smolpack.html (2003).
[42] B. L. Rozovskiĭ, *Stochastic Evolution Systems*, Kluwer, Dordrecht, 1990.
[43] I. H. Sloan and S. Joe, *Lattice Methods for Multiple Integration*, Oxford University Press, New York, 1994.
[44] S. A. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Soviet Math. Dokl., 4 (1963), pp. 240–243.
[45] M. Tatang and G. McRae, *Direct Treatment of Uncertainty in Models of Reaction and Transport*, Technical report, Department of Chemical Engineering, MIT, Cambridge, MA, 1994.

[46] G. W. Wasilkowski and H. Woźniakowski, *Explicit cost bounds of algorithms for multivariate tensor product problems*, J. Complexity, 11 (1995), pp. 1–56.
[47] D. Xiu and J. S. Hesthaven, *High-order collocation methods for differential equations with random inputs*, SIAM J. Sci. Comput., 27 (2005), pp. 1118–1139.
[48] D. Xiu and G. Karniadakis, *Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos*, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 4927–4948.
[49] G. Zhang and M. Gunzburger, *Error analysis of a stochastic collocation method for parabolic partial differential equations with random input data*, SIAM J. Numer. Anal., 50 (2012), pp. 1922–1940.
[50] Z. Zhang, B. Rozovskii, M. V. Tretyakov, and G. E. Karniadakis, *A multistage Wiener chaos expansion method for stochastic advection-diffusion-reaction equations*, SIAM J. Sci. Comput., 34 (2012), pp. A914–A936.