SIAM J. Sci. Comput.
Vol. 37, No. 4, pp. A1872–A1890

ADAPTIVE WICK–MALLIAVIN APPROXIMATION TO NONLINEAR SPDEs WITH DISCRETE RANDOM VARIABLES[*]

MENGDI ZHENG[†], BORIS ROZOVSKY[†], AND GEORGE E. KARNIADAKIS[†]

Abstract. We propose an adaptive Wick–Malliavin (WM) expansion in terms of the Malliavin derivative of order Q to simplify the propagator of general polynomial chaos (gPC) of order P (a system of deterministic equations for the coefficients of gPC) and to control the error growth with respect to time. Specifically, we demonstrate the effectiveness of the WM method by solving a stochastic reaction equation and a Burgers equation with several discrete random variables. Exponential convergence is shown numerically with respect to Q when $Q \geq P - 1$. We also analyze the computational complexity of the WM method and identify a significant speedup with respect to gPC, especially in high dimensions.

Key words. discrete stochastic process, general polynomial chaos, high dimensions, Wiener–Askey scheme

AMS subject classifications. 33C45, 60H07, 60H15, 60H35, 68U20

DOI. 10.1137/140975930

1. Introduction.

Nonlinear SPDEs with discrete random variables (RVs) and jump processes are of practical use, since sources of stochastic excitations including uncertain parameters and boundary/initial conditions are typically observed at discrete values. Many complex systems of fundamental and industrial importance are significantly affected by the underlying fluctuations/variations in random excitations, such as the stochastic-volatility jump-diffusion model in mathematical finance [3, 4, 8, 9, 10, 51], stochastic simulation algorithms for modeling diffusion, reaction, and taxis in biology [14], truncated Levy flight model in turbulence [27, 35, 37, 46], quantum-jump models in physics [12], etc. Numerically, nonlinear SPDEs with discrete processes are often solved by general polynomial chaos (gPC) involving a system of coupled deterministic nonlinear equations [50], or the probabilistic collocation method (PCM) [17, 49, 52] involving N nonlinear corresponding PDEs obtained at the collocation points. In this paper, we use the Wick–Malliavin (WM) approximation [47] (defined in section two) of nonlinear SPDEs to improve the efficiency of gPC.

The *Wick product* was first proposed as a renormalization technique in quantum field theory [48], hence a nonlinear equation driven by singular noise can be interpreted in the renormalized Wick form, e.g., the viscous Burgers equation with a white noise source [19, 25]. Subsequently, Wick calculus was introduced in stochastic analysis [20, 21], as it can be a generalization of the Skorokhod–Malliavin integral [33]. *Malliavin derivative* is a notion of derivative in the *Malliavin calculus*, which extends the calculus of variations from functions to stochastic processes. This calculus allows the computation of derivatives and integration by parts of RVs, e.g., in mathematical finance to compute the sensitivities of financial derivatives [5, 28, 29, 30, 34, 38]. A

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section July 7, 2014; accepted for publication (in revised form) April 7, 2015; published electronically August 4, 2015.

http://www.siam.org/journals/sisc/37-4/97593.html

[†] Division of Applied Mathematics, Brown University, Providence, RI 02906 (mengdi_zheng@brown.edu, boris_rozovsky@brown.edu, george_karniadakis@brown.edu). The first and the third authors of this work were supported partially by ARL and University of Utah (CED3M), NSF/DMS-1216437, and by OSD-MURI grant FA9550-09-1-0613. The second author was supported by ARO grant W911Nf-13-1-0012.

Taylor-like series expansion involving both Wick products and Malliavin derivatives was proposed as the *WM expansion* to approximate the polynomial nonlinearity in SPDEs [36]. This has been applied to moment statistics for SPDEs with polynomial nonlinearity and one RV with Gaussian or uniform distribution [47]. If an SPDE is driven by RVs or a stochastic process that can be represented by independent continuous RVs, in other distributions than Gaussian and uniform with finite moments, we can still apply the WM expansion in a similar way as long as we can generate orthogonal polynomials with respect to their measures [52]. For example, if a nonlinear SPDE is driven by a Brownian motion, by the Karhunen–Loève expansion [32], we can represent the Brownian motion by independent Gaussian RVs, and apply the WM approximation [47]. Our objective here is to solve SPDEs with polynomial nonlinearity and multiple discrete RVs by the WM approximation using gPC based on the Wiener–Askey scheme of hypergeometric polynomials [50].

The paper is organized as follows: in section 2, we define the WM expansion and derive the *WM propagators* for a stochastic reaction equation and a stochastic Burgers equation; in section 3, we present several numerical results for SPDEs with one RV and multiple RVs, including an adaptive procedure to control the error in time. We also compare the computational complexity between gPC and WM for the stochastic Burgers equation with the same level of accuracy. We conclude in section 4 with a short summary. In the appendix, we provide an iterative algorithm to generate coefficients in the WM approximation.

2. WM approximation.

The WM propagator simplifies the gPC propagator by considering a fewer number of product terms from the polynomial nonlinearity. In this section, we present this simplification procedure and derive WM propagators for a stochastic reaction equation and a stochastic Burgers equation. The following procedure can be done for any discrete stochastic input with finite moments of all orders. To demonstrate the approximation procedure, we take the Poisson RV as an example.

2.1. WM series expansion.

Given a discrete Poisson RV $\xi \sim Pois(\lambda)$ with measure $\Gamma(x) = \sum_{k\in\mathbf{S}} \frac{e^{-\lambda}\lambda^k}{k!}\delta(x-k)$, on a finite support[1] $\mathbf{S} = \{0, 1, 2, \dots, N\}$, there is an associated unique set of monic orthogonal polynomials [11, 18], called Charlier polynomials, denoted as $\{c_k(x;\lambda), k = 0, 1, 2, \dots\}$, such that

$$\sum_{k\in\mathbf{S}} \frac{e^{-\lambda}\lambda^k}{k!} c_m(k;\lambda)c_n(k;\lambda) = \begin{cases} n!\lambda^n\delta_{mn} & \text{if } m = n, \\ 0 & \text{if } m \neq n. \end{cases} \tag{2.1}$$

The monic Charlier polynomials associated with $Pois(\lambda)$ are defined as

$$c_n(x;\lambda) = \sum_{k=0}^{n} \binom{n}{k} (-\lambda)^{n-k} x(x-1)\dots(x-(k-1)), \quad n = 0, 1, 2, \dots. \tag{2.2}$$

Here $\binom{n}{k}$ is the binomial coefficient. The first few Charlier polynomials are

$$c_0(x;\lambda) = 1, \tag{2.3}$$

$$c_1(x;\lambda) = x - \lambda, \tag{2.4}$$

$$c_2(x;\lambda) = x^2 - 2\lambda x - x + \lambda^2, \tag{2.5}$$

$$c_3(x;\lambda) = x^3 - 3\lambda x^2 - 3x^2 + 3\lambda^2 x + 3\lambda x + 2x - \lambda^3. \tag{2.6}$$

[1] For numerical computation, here we consider the support $\mathbf{S}$ to be from 0 to N instead of 0 to ∞, such that $\mathbf{P}(\xi = N) \leq 1e-32$.

Since $\{c_k(x;\lambda), k = 0, 1, 2, \ldots\}$ belongs to the Askey scheme of hypergeometric polynomials [2], the product of any two polynomials can be expanded as [1]

$$c_m(x)c_n(x) = \sum_{k=0}^{m+n} a(k,m,n)c_k(x), \quad m,n = 0,1,2,\ldots, \tag{2.7}$$

where $a(k,m,n)$ can be evaluated both analytically[2] and numerically [6, 16, 18, 40, 41]. Numerically we may generate $a(k,m,n)$ by

$$a(k,m,n) = \frac{\sum_{j\in\mathbf{S}} \frac{e^{-\lambda}\lambda^j}{j!} c_k(j;\lambda)c_m(j;\lambda)c_n(j;\lambda)}{\sum_{j\in\mathbf{S}} \frac{e^{-\lambda}\lambda^j}{j!} c_k(j;\lambda)c_k(j;\lambda)}, \quad k = 0,1,2,\ldots,m+n. \tag{2.8}$$

Analytically $a(k,m,n)$ is given by [26]

$$a(k,m,n) = \frac{\sum_{l=0}^{\llcorner (m+n-k)/2 \lrcorner} \frac{m!n!k!\lambda^{l+k}}{l!(k-m+l)!(k-n+l)!(m+n-k-2l)!}}{k!\lambda^k}, k = 0,1,\ldots,m+n. \tag{2.9}$$

Here $\llcorner x \lrcorner$ is the floor function.

The alternative analytical method to generate $a(k,m,n)$ in (2.8) is given in the appendix.

For convenience, let us denote $a(m+n-2p,m,n)$ by K_{mnp} as follows (for $\xi \sim Pois(\lambda)$),

$$K_{mnp} = \frac{\sum_{l=0}^{\llcorner p \lrcorner} \frac{m!n!(m+n-2p)!\lambda^{l+m+n-2p}}{l!(n-2p+l)!(m-2p+l)!(2p-2l)!}}{(m+n-2p)!\lambda^{m+n-2p}}, \quad p = 0, 1/2, \ldots, \frac{m+n}{2}. \tag{2.10}$$

Then (2.7) can be rewritten as

$$c_m(x;\lambda)c_n(x;\lambda) = \sum_{p=0}^{\frac{m+n}{2}} K_{mnp}c_{m+n-2p}(x;\lambda), \tag{2.11}$$

where p takes half-integer values as $p = 0, 1/2, 1, \ldots, \frac{m+n}{2}$. Equation (2.11) is completely equivalent to (2.7).

Now let us define the *Wick product* $\diamond$ as [13, 22, 31, 33, 48]

$$c_m(x;\lambda) \diamond c_n(x;\lambda) = c_{m+n}(x;\lambda), \quad m,n = 0,1,2,\ldots \tag{2.12}$$

and define the *Malliavin derivative* $\mathcal{D}^p$ as[3] [33, 39]

$$\mathcal{D}^p c_i(x;\lambda) = \frac{i!}{(i-p)!}c_{i-p}(x;\lambda), \quad i = 0,1,2,\ldots, \quad p = 0, 1/2, 1, \ldots, i. \tag{2.13}$$

[2] For monic polynomials $\{c_i(x), i = 0,1,2,\ldots\}$, we can derive $a(m+n,m,n)$ to $a(0,m,n)$ iteratively by matching the coefficient of x^{m+n} to x^0 for the left- and right-hand sides of (2.7), as an alternative method to deriving $a(k,m,n)$, than in (2.8). We notice that $a(m+n,m,n) = 1$.

[3] In this definition p has to take half-integer values in order to balance (2.17) with (2.11). Although here in the definition of Malliavin derivative c_{i-p} may take integer values, the Malliavin derivative will always appear with the weighted Wick product; therefore, after taking the Malliavin derivative and Wick product the resulting polynomial will always be an integer.

We define $\mathcal{D}^{p_1,\ldots,p_d}$ as the product of operators from $\mathcal{D}^{p_1}$ to $\mathcal{D}^{p_d}$:

$$\mathcal{D}^{p_1,\ldots,p_d} c_{i_1}(x;\lambda)\ldots c_{i_d}(x;\lambda) = \Pi_{j=1}^{d} \frac{i_j!}{(i_j-p_j)!} c_{i_j-p_j}(x;\lambda), \\ i_j = 0,1,2,\ldots, p_j = 0, 1/2, 1, \ldots, i_j. \tag{2.14}$$

We define the *weighted Wick product* $\diamond_p$ in terms of the Wick product as

$$c_m \diamond_p c_n = \frac{p!m!n!}{(m+p)!(n+p)!} K_{m+p,n+p,p} c_m \diamond c_n, \tag{2.15}$$

and define $\diamond_{p_1,\ldots,p_d}$ as

$$(c_{m_1}\ldots c_{m_d}) \diamond_{p_1,\ldots,p_d} (c_{n_1}\ldots c_{n_d}) = \Pi_{j=1}^{d} \frac{p_j!m_j!n_j!}{(m_j+p_j)!(n_j+p_j)!} K_{m_j+p_j,n_j+p_j,p_j} c_{m_j} \diamond c_{n_j}. \tag{2.16}$$

Therefore, (2.11) can be rewritten as

$$c_m(x;\lambda)c_n(x;\lambda) = \sum_{p=0}^{\frac{m+n}{2}} \frac{\mathcal{D}^p c_m \diamond_p \mathcal{D}^p c_n}{p!}. \tag{2.17}$$

We note that the definition of the weighted Wick product (2.15) depends on the measure Γ. Assume that we are given two random fields u and v on the same probability space $(\mathbf{S}, \mathcal{B}(\mathbf{S}), \Gamma)$, with their expansions $u = \sum_{i=0}^{\infty} u_i c_i$ and $v = \sum_{i=0}^{\infty} v_i c_i$. Then, we can expand uv by

$$uv = \sum_{p=0}^{\infty} \frac{\mathcal{D}^p u \diamond_p \mathcal{D}^p v}{p!} \tag{2.18}$$

(index p takes half-integer values), if we define

$$\mathcal{D}^p u = \sum_{i=0}^{\infty} u_i \mathcal{D}^p c_i. \tag{2.19}$$

Now let us introduce a nonnegative half integer $Q \in \{0, 1/2, 1, \ldots\}$ as the *WM order*[4], hence (2.18) can be approximated by the following *WM expansion*

$$uv \approx \sum_{p=0}^{Q} \frac{\mathcal{D}^p u \diamond_p \mathcal{D}^p v}{p!}, Q = 0, 1/2, 1, \ldots, \tag{2.20}$$

and p here also takes half-integer values.

Now let us assume η to be an RV with discrete measure of finite moments of all orders on a complete probability space $(\mathbf{S}, \mathcal{B}(\mathbf{S}), \Gamma)$. There is an associated unique set of monic orthogonal polynomials with respect to this measure [18], denoted as $\{P_i(\eta), i = 0, 1, 2, \ldots\}$ for $\eta \in \mathbf{S}$, such that

$$\int_{\mathbf{S}} P_m(\eta)P_n(\eta)d\Gamma(\eta) \begin{cases} > 0 & \text{if } m = n, \\ = 0 & \text{if } m \neq n. \end{cases} \tag{2.21}$$

Following the same procedure from (2.7) to (2.17), we can expand the product of

[4] As the upper limit of index p in (2.20), Q takes half-integer values, in the same way as p in (2.17) takes half-integer values from 0 to $\frac{m+n}{2}$.

$u' = \sum_{i=0}^{\infty} u'_i P_i$ and $v' = \sum_{i=0}^{\infty} v'_i P_i$ as

$$u'v' \approx \sum_{p=0}^{Q} \frac{\mathcal{D}^p u' \diamond_p \mathcal{D}^p v'}{p!}, Q = 0, 1/2, 1, \dots. \tag{2.22}$$

2.2. WM propagators.

In this section, we will study a stochastic reaction equation and a stochastic Burgers equation, and derive their *WM propagators*.

2.2.1. Reaction equation. Let us consider the following reaction equation with a random coefficient:

$$\frac{dy}{dt} = -\sigma k(t, \xi_1, \xi_2, \dots, \xi_d) y(t;\omega), \quad y(0;\omega) = y_0, \tag{2.23}$$

where $\xi_1, \dots, \xi_d \sim Pois(\lambda)$ are independent identically distributed (i.i.d.), and[5] $k(t, \xi_1, \dots, \xi_d) = \sum_{i_1,\dots,i_d=0}^{\infty} a_{i_1,\dots,i_d}(t) c_{i_1}(\xi_1;\lambda) \dots c_{i_d}(\xi_d;\lambda)$; σ controls the variance of the reaction coefficient. Also the $\{c_k(\xi;\lambda), k = 0, 1, 2, \dots\}$ are monic Charlier polynomials associated with the Poisson distribution and with mean λ [15, 23, 42, 45].

Remark. Here we present the WM approximation method for the Poisson distribution; however, the method is not restricted to Poisson distribution, since we can generate orthogonal polynomials with respect to other discrete measures [6, 16, 18, 40], at least for the Wiener–Askey family of polynomials [1, 2].

By (2.20), the WM approximation to (2.23) is

$$\frac{dy}{dt} \approx -\sigma \sum_{p_1,\dots,p_d=0}^{Q_1,\dots,Q_d} \frac{\mathcal{D}^{p_1,\dots,p_d} k(t, \xi_1, \dots, \xi_d) \diamond_{p_1,v,p_d} \mathcal{D}^{p_1,\dots,p_d} y}{p_1! \dots p_d!}, y(0;\omega) = y_0. \tag{2.24}$$

Here $Q_1, \dots, Q_d$ are WM orders for RVs $\xi_1, \dots, \xi_d$, respectively. We expand the solution to (2.23) in a finite dimensional series as

$$y(t;\omega) = \sum_{j_1,j_2,\dots,j_d=0}^{P_1,\dots,P_d} \hat{y}_{j_1,\dots,j_d}(t) c_{j_1}(\xi_1) \dots c_{j_d}(\xi_d), \tag{2.25}$$

where $P_1, \dots, P_d$ are polynomial chaos expansion order for RVs $\xi_1, \dots, \xi_d$, respectively. By substituting (2.25) into (2.24) and the Galerkin projection onto $c_{i_1}(\xi_1) \dots c_{i_d}(\xi_d)$,

$$\langle f(\xi_1, \dots, \xi_d) c_{i_1}(\xi_1) \dots c_{i_d}(\xi_d) \rangle = \int_{S_1} d\Gamma_1(\xi_1) \dots \int_{S_d} d\Gamma_1(\xi_d) f c_{i_1}(\xi_1) \dots c_{i_d}(\xi_d) \tag{2.26}$$

(S_i and Γ_i are the support and the measure of ξ_i), we obtain the *WM propagator* for problem (2.23) as

$$\begin{aligned} \frac{d\hat{y}_{i_1\dots i_d}(t)}{dt} = -\sigma \sum_{l_1,\dots,l_d=0}^{P_1,\dots,P_d} \sum_{m_1,\dots,m_d=0}^{Q_1,\dots,Q_d} \Big(& K_{l_1,2m_1+i_1-l_1,m_1} \dots K_{l_d,2m_d+i_d-l_d,m_d} \\ & a_{l_1\dots l_d}(t) \hat{y}_{2m_1+i_1-l_1,\dots,2m_d+i_d-l_d} \Big), \\ \hat{y}_{i_1\dots i_d}(0) = y_0 \delta_{i_1,0} \delta_{i_2,0} \dots \delta_{i_d,0}, & \end{aligned} \tag{2.27}$$

for $i_1 = 0, 1, \dots, P_1, \dots, i_d = 0, 1, \dots, P_d$.

[5] Such $k(t, \xi_1, \dots, \xi_d)$ is meaningful to be considered because many stochastic processes have series representations, e.g., Karhunen–Loeve expansion for the Gaussian process [24, 32], and shot noise expansion for Levy pure jump processes [7, 43, 44].

2.2.2. The Burgers equation. Let us now consider the following Burgers equation with a random forcing term:

$$u_t + uu_x = \nu u_{xx} + \sigma \sum_{j=1}^{d} c_1(\xi_j)\psi_j(x,t), \quad x \in [-\pi, \pi], \tag{2.28}$$

with initial condition

$$u(x,0) = 1 - sin(x) \tag{2.29}$$

and periodic boundary conditions. Here $\xi_{1,\dots,d} \sim Pois(\lambda)$ are i.i.d. RVs, and σ is a constant that controls the magnitude of the force. The WM approximation of (2.28) is

$$u_t + \sum_{p_1,\dots,p_d=0}^{Q_1,\dots,Q_d} \frac{1}{p_1!\dots p_d!} \mathcal{D}^{p_1\dots p_d} u \diamond_p \mathcal{D}^{p_1\dots p_d} u_x \approx \nu u_{xx} + \sigma \sum_{j=1}^{d} c_1(\xi_j)\psi_j(x,t). \tag{2.30}$$

If we expand the solution in a finite dimensional series as

$$u(x,t;\xi_1,\dots,\xi_d) = \sum_{k_1,\dots,k_d=0}^{P_1,\dots,P_d} \tilde{u}_{k_1,\dots,k_d}(x,t) c_{k_1}(\xi_1;\lambda)\dots c_{k_d}(\xi_d;\lambda), \tag{2.31}$$

then by substituting (2.31) into (2.30) and performing a Galerkin projection onto $c_{k_1}(\xi_1)\dots c_{k_d}(\xi_d)$, we derive the WM propagator for problem (2.28) as

$$\begin{aligned} \frac{\partial}{\partial t}\tilde{u}_{k_1\dots k_d}(x,t) + \sum_{p_1,\dots,p_d=0}^{Q_1,\dots,Q_d} \sum_{m_1,\dots,m_d=0}^{P_1,\dots,P_d} \Big(& K_{m_1,k_1+2p_1-m_1,p_1} \dots K_{m_d,k_d+2p_d-m_d,p_d} \\ & \tilde{u}_{m_1\dots m_d} \frac{\partial}{\partial x} \tilde{u}_{k_1+2p_1-m_1,\dots,k_d+2p_d-m_d} \Big) \\ = \nu \frac{\partial^2}{\partial x^2}\tilde{u}_{k_1\dots k_d} + \sigma(\delta_{1,k_1}\delta_{0,k_2}\dots\delta_{0,k_d}\psi_1 + \dots + \delta_{0,k_1}\delta_{0,k_2}\dots\delta_{1,k_d}\psi_d) & \end{aligned} \tag{2.32}$$

for $k_1, \dots, k_d = 0, 1, \dots, P$, with the restriction $0 \le k_i + 2p_i - m_i \le P$ for $i = 1, \dots, d$. The initial conditions (I.C.) and boundary conditions (B.C.) are given by

$$\begin{cases} \tilde{u}_{0,0,\dots,0}(x,0) = u(x,0) = 1 - sin(x) \quad \text{(I.C.)}, \\ \tilde{u}_{k_1,\dots,k_d}(x,0) = 0, \quad \text{if} \quad (k_1,\dots,k_d) \neq (0,\dots,0) \quad \text{(I.C.)}, \\ \tilde{u}_{k_1\dots k_d}(-\pi,t) = \tilde{u}_{k_1\dots k_d}(\pi,t) \quad \text{(periodic B.C. on}[-\pi,\pi]). \end{cases} \tag{2.33}$$

3. Results.

In this section, we will provide numerical results for solving the reaction and the Burgers equations with different discrete random inputs by the WM method. We will compare the computational complexity of WM and gPC for the Burgers equation with multiple RVs.

3.1. Reaction equation with one RV. In Figure 1, we show results from computing the WM propagator given in (2.27) for the reaction equation with one Poisson RV ($d = 1$ in (2.23)). We plot the errors of second moments at final time Ts with respect to different WM expansion order Q.[6] The polynomial expansion order

[6] In Figure 1 we show errors for Q taking integer values because the error line for $Q = k$ is almost the same as $k + \frac{1}{2}$. We observe similar behavior in Figure 4.

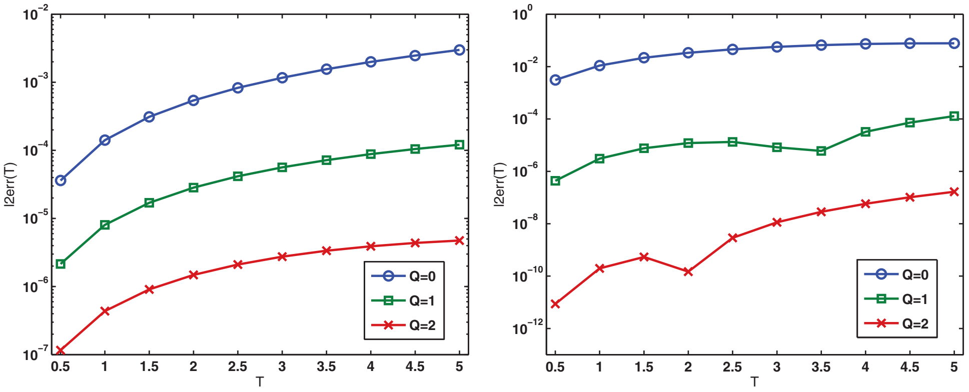

FIG. 1. *Reaction equation with one Poisson RV $\xi \sim Pois(\lambda)$ $(d = 1)$: errors versus final time T defined in* (3.1) *for different WM order Q in equation* (2.27)*, with polynomial order $P = 10$, $y_0 = 1$, $\lambda = 0.5$. We used RK4 scheme with time step $dt = 1e-4$; $k(\xi) = \frac{c_0(\xi;\lambda)}{2!} + \frac{c_1(\xi;\lambda)}{3!} + \frac{c_2(\xi;\lambda)}{4!}$, $\sigma = 0.1$(left); $k(\xi) = \frac{c_0(\xi;\lambda)}{0!} + \frac{c_1(\xi;\lambda)}{3!} + \frac{c_2(\xi;\lambda)}{6!}$, $\sigma = 1$ (right).*

P in (2.25) was chosen sufficiently large in order to mainly examine the convergence with respect to Q. We used the fourth-order Runge–Kutta method (RK4) to solve (2.27) with sufficiently small time steps. The error of the second moment at final time T is defined as:

$$l2err(T) = \left| \frac{E[y_{ex}^2(T;\omega)] - E[y_{num}^2(T;\omega)]}{E[y_{ex}^2(T;\omega)]} \right|. \tag{3.1}$$

From Figure 1 with a fixed polynomial order P, we take $k(\xi) = a_0(t)c_0(\xi;\lambda) + a_1(t)c_1(\xi;\lambda) + a_2(t)c_2(\xi;\lambda)$; therefore, the WM order $Q = 2$ is the highest order that equates (2.23) with (2.24) in (2.20) (when $Q \geq 2$, the WM propagator is exactly the same as the gPC propagator). We observe that in Figure 1, when Q increases by one, the error is improved by at least one order of magnitude when $\sigma = 0.1$, and four orders of magnitude when $\sigma = 1$. Therefore, with less computational cost than gPC, the WM method can achieve the same accuracy as gPC. In gPC, the polynomial order P serves as a resolution parameter for the stochastic system. In the WM method, for each P we may further refine the system by another resolution parameter Q. We observe that the right plot in Figure 1 has a dip for error lines corresponding to $Q = 1$ and 2. When σ is larger, the solution of (2.23) decays faster, and hence this trend in the error; however, with polynomial order P we ignore the terms in the sum (2.25) with polynomial order larger than P, which increases the error with respect to time. Because of this balance of decreasing and increasing errors, we observe that errors go down at first and then up in the right plot in Figure 1. On the left plot of Figure 1 we do not observe that the error goes down and up because σ is small and the solution decays slower so the error mainly increases with time. We can evaluate the coefficients K_{mnp} in (2.10) offline, and we compute only the WM propagator in (2.24) online. We consider the number of terms to evaluate $k(t,\xi)y(t;\omega)$ in (2.23) in the WM propagator (2.24) as the primary contribution to the computational complexity. We consider the online CPU time in Table 1 as the CPU time to evaluate the right-hand side of (2.24) excluding the time to compute coefficients K_{mnp} in (2.10). In Table 1 we compare the complexity and corresponding computational time between gPC of different orders P and WM with a fixed order of $P = 3, Q = 2$ for the reaction equation (2.23) with one

Table 1

For gPC with different orders P and WM with a fixed order of $P = 3, Q = 2$ in reaction equation (2.23) with one Poisson RV ($\lambda = 0.5$, $y_0 = 1$, $k(\xi) = \frac{c_0(\xi;\lambda)}{2!} + \frac{c_1(\xi;\lambda)}{3!} + \frac{c_2(\xi;\lambda)}{4!}$, $\sigma = 0.1$, RK4 scheme with time step $dt = 1e-4$), we compare (1) computational complexity ratio to evaluate $k(t,\xi)y(t;\omega)$ between gPC and WM (upper); (2) CPU time ratio to compute $k(t,\xi)y(t;\omega)$ between gPC and WM (lower). We simulated in MATLAB on Intel (R) Core (TM) i5-3470 CPU @ 3.20 GHz.

gPC order P	$P=4$	$P=6$	$P=8$	$P=10$
Ratio of complexity (gPC/WM)	1.4054	2.2162	3.027	3.8378
Ratio of CPU time (gPC/WM)	1.2679	1.8036	2.3393	2.875

RV (with the same parameters as on the left of Figure 1). Notice that the $l2err$ from WM with $P = 3, Q = 2$ is $1.5e-8$ and the $l2err$ from gPC with $P = 10$ is $1.4e-8$ (almost the same), however the online CPU time for gPC is 2.875 times greater than that of WM.

3.2. The Burgers equation with one RV. Now let us compute the WM propagator for a Burgers equation with one Poisson RV in (2.32). We solved the WM propagator by a second-order implicit-explicit (IMEX) time splitting scheme.[7] For spatial discretization we used the Fourier collocation method. The reference solution was established by running the Burgers equation with ξ taking all the possible values.[8] In this problem we define the L_2 norm of error for second moments as follows, for a certain final time T:

$$l2u2(T) = \frac{||E[u^2_{num}(x,T;\xi)] - E[u^2_{ex}(x,T;\xi)]||_{L^2([-\pi,\pi])}}{||E[u^2_{ex}(x,T;\xi)]||_{L^2([-\pi,\pi])}}. \tag{3.2}$$

In Figure 2, we observe monotonic convergence with respect to Q, that is, by increasing the WM order Q by one, the error decreases effectively by five to six orders of magnitude at $T = 1$. If we use gPC in this problem, we will calculate $(P+1)^3$ terms in $\sum_{i,j=0}^{P} u_i(x)\frac{\partial u_j}{\partial x}$ for $(P+1)$ equations in the gPC propagator (343 terms in this problem). However by the WM method, in order to have good accuracy, say $1e-12$, as shown in Figure 2, we consider many fewer terms resulting from the nonlinear term $u\frac{\partial u}{\partial x}$ in the Burgers equation by only taking $Q = 3$ (231 terms in this problem).

In Figure 3, we plot the error defined in (3.2) with respect to polynomial expansion order P, for different WM order Q. We also compare it with the error by the PCM with $(P + 1)$ points.[9] We observe that for a fixed polynomial order P in gPC, the smallest Q to match the error from the WM propagator to the same order with PCM is when $Q = P - 1$. For example, in Figure 3 when $P = 2$, the first error line by WM that touches the black solid line by PCM is the one that corresponds to $Q = 1$. Although this observation is only empirical, it allows us to compare the computational complexity between gPC and WM with the same level of accuracy, i.e., we are going to compare the computational cost later between gPC of polynomial order P and WM of polynomial order P and of WM order $Q = P - 1$. We also observe from Figure 3 the smallest value of Q we need to model the stochastic Burgers equation with one

[7] We used the second-order RK2 scheme for nonlinear terms and the forcing term, and the Crank–Nicolson scheme for the diffusion term.

[8] Although the Poisson RV has an infinite number of points in the support, we only consider the points with probability more than $1e-16$.

[9] gPC with polynomial order P has the same magnitude of error with PCM implemented with $(P+1)$ quadrature points, therefore, by plotting PCM with $(P+1)$ quadrature points against WM with polynomial order P, we are comparing the gPC with WM at the same polynomial order P.

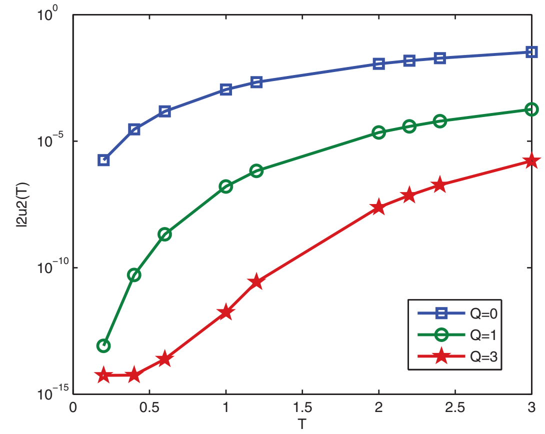

FIG. 2. *The Burgers equation with one Poisson RV $\xi \sim Pois(\lambda)$ $(d = 1, \psi_1(x,t) = 1)$: $l2u2(T)$ error defined in* (3.2) *versus time, with respect to different WM order Q. Here we take in* (2.32) *the polynomial expansion order $P = 6$, $\lambda = 1$, $\nu = 1/2$, $\sigma = 0.1$, IMEX (Crank–Nicolson/RK2) scheme with time step $dt = 2e-4$, and* 100 *Fourier collocation points on $[-\pi, \pi]$.*

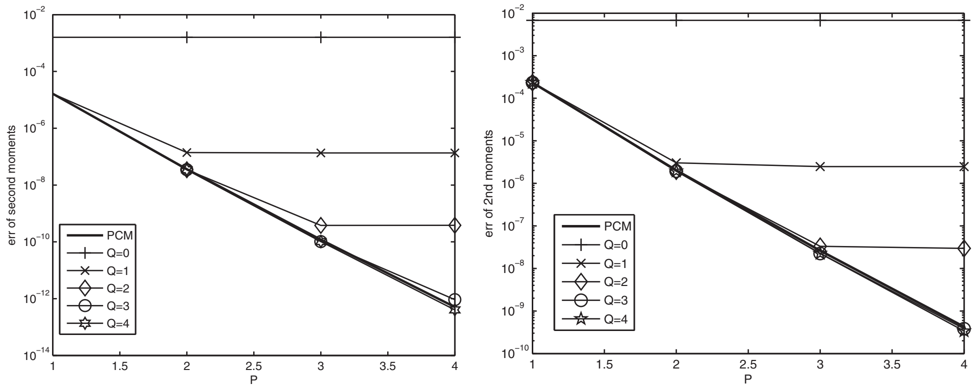

FIG. 3. *P-convergence for the Burgers equation with one Poisson RV $\xi \sim Pois(\lambda)$ $(d = 1, \psi_1(x,t) = 1)$: errors defined in* (3.2) *versus polynomial expansion order P, for different WM order Q, and by PCM with $P + 1$ points with the parameters $\nu = 1$, $\lambda = 1$, final time $T = 0.5$, IMEX (Crank–Nicolson/RK2) scheme with time step $dt = 5e-4$,* 100 *Fourier collocation points on $[-\pi, \pi]$, $\sigma = 0.5$ (left), and $\sigma = 1$ (right).*

discrete RV for a specific polynomial order P, to achieve the same accuracy with gPC of polynomial order P. When $Q \geq P-1$, we see from Figure 3 that even if we increase P the convergence rate versus P will be slower than P-convergence from gPC.

In Figure 4, we investigate the Q-convergence of the WM approximation by plotting the error defined in (3.2) with respect to WM order Q, for different polynomial expansion orders P. The first observation is that when Q increases from integer k to the next larger half-integer $k + \frac{1}{2}$, the error is not prominently improved, but the error is obviously improved when Q increases from integer k to integer $k + 1$. This is very similar to a phenomenon in spectral methods that the magnitude of error oscillates between even orders and odd orders. The second observation is that the choice of $Q = P - 1$ is optimum for the WM approximation, because in Figure 4 the error remains at the same magnitude when Q is taking values larger than $P-1$. For example, we note the error line for the left figure in Figure 4 with respect to $P = 2$: the error

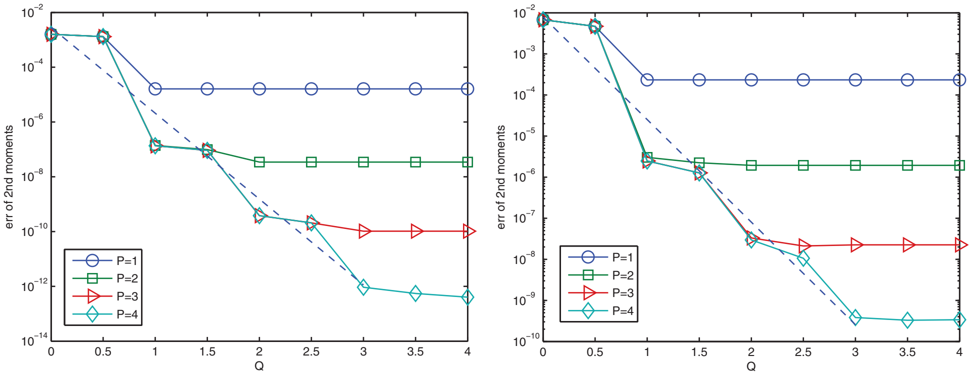

FIG. 4. *Q-convergence for the Burgers equation with one Poisson RV $\xi \sim Pois(\lambda)$ $(d = 1, \psi_1(x,t) = 1)$: errors defined in (3.2) versus WM order Q, for different polynomial order P, with the parameters, $\nu = 1$, $\lambda = 1$, final time $T = 0.5$, IMEX(RK2/Crank–Nicolson) scheme with time step $dt = 5e-4$, 100 Fourier collocation points on $[-\pi, \pi]$, $\sigma = 0.5$ (left), and $\sigma = 1$ (right). The dashed lines serve as a reference for the convergence rate.*

decreases when Q is smaller than $P - 1 = 1$, however, when Q takes values such as 2 or 3, the error remains at the same magnitude. This is an important observation that allows us to save computational time when simulating nonlinear SPDEs i.e., we may use smaller values of P for a certain Q and obtain the maximum possible accuracy.

From Figures 3 and 4, we conclude that in order to model a stochastic Burgers equation with one discrete RV, to achieve the same P-convergence rate with gPC, we may take $Q = P - 1$ in the WM method, with much less computational cost than gPC.

3.3. Reaction equation with multiple RVs. Now let us compute (2.27) with five i.i.d. Poisson RVs with mean λ $(d = 5)$. We solve problem (2.23) assuming a new model, where $k(\xi_1, \xi_2, \dots, \xi_5, t) = \sum_{i=1}^{5} cos(it)c_1(\xi_i)$. The WM propagator in this problem was solved by the RK2 scheme. For a fixed polynomial expansion order P in Figure 5, we plot the error defined in (3.1) with respect to time and for different WM order Q.

We observe in Figure 5 that by adding only one more Malliavin derivative order Q, the error is improved by two orders of magnitude at $T = 0.5$. When $Q = 1$, the WM propagator has a much simpler form than the gPC propagator. Figure 5 also demonstrates the ability of computing SPDEs with multiple RVs by the WM method. Notice that Levy processes have different types of series expansions by independent RVs, therefore, Figure 5 represents the first step towards dealing with nonlinear SPDEs with Levy processes (including Gaussian processes and pure jump processes that admit series representations).

Next let us compute (2.27) with one Poisson RV (ξ_1) with mean λ and one binomial RV (ξ_2) with the number of trials N and success probability p. We solve problem (2.23) assuming $k(\xi_1, \xi_2) = c_1(\xi_1)k_1(\xi_2)$, where $k_1(\xi_2)$ is the orthogonal polynomial to the binomial distribution for ξ_2. We derive the coefficients K_{mnp} in (2.11) both for the Poisson distribution and the binomial distribution. The WM propagator in this case is still given by (2.27) with $d = 2$, except replacing the corresponding K_{mnp} for ξ_2 by those generated from the binomial distribution. For a fixed polynomial order P in Figure 6, we plot the error defined in (3.1) with respect to time and for different WM order Q.

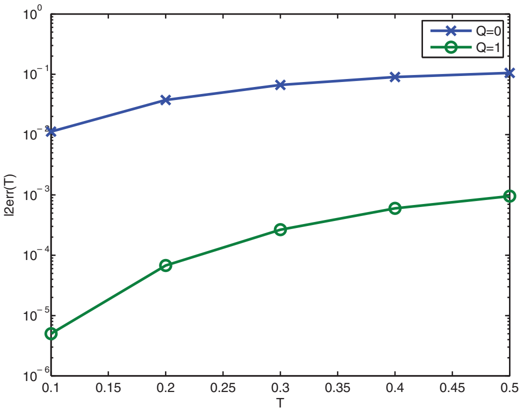

FIG. 5. *Reaction equation with five Poisson RVs* $\xi_{1,\dots,5}$ $\sim$*Pois*(λ) $(d = 5)$*: error defined in* (3.1) *with respect to time, for different WM order* Q*, with parameters* $\lambda = 1$*,* $\sigma = 0.5$*,* $y_0 = 1$*, polynomial order* $P = 4$*, RK*2 *scheme with time step* $dt = 1e - 3$*, and* $k(\xi_1, \xi_2, \dots, \xi_5, t) = \sum_{i=1}^{5} cos(it)c_1(\xi_i)$ *in* (2.23).

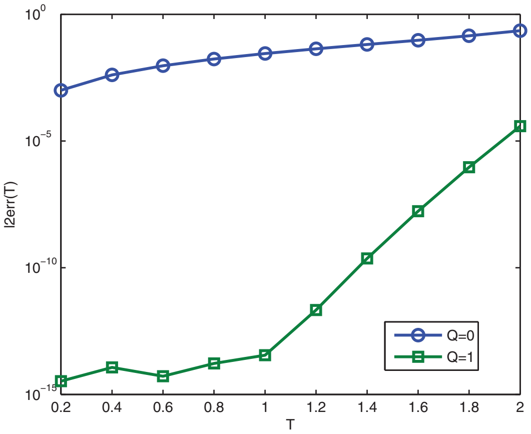

FIG. 6. *Reaction equation with one Poisson RV* ξ_1 $\sim$*Pois*(λ) *and one Binomial RV* $\xi_2 \sim$ *Bino*(N, p) $(d = 2)$*: error defined in* (3.1) *with respect to time, for different WM order* Q*, with parameters* $\lambda = 1$*,* $\sigma = 0.1$*,* $N = 10$*,* $p = 1/2$*,* $y_0 = 1$*, polynomial order* $P = 10$*, RK*4 *scheme with time step* $dt = 1e - 4$*, and* $k(\xi_1, \xi_2, t) = c_1(\xi_1)k_1(\xi_2)$ *in* (2.23).

We observe in Figure 6 that by adding one more Malliavin derivative order Q, the error is improved by ten orders of magnitude at $T = 1$. Figure 6 also demonstrates the ability to compute SPDEs with multiple RVs with different distributions (hybrid-type).

3.4. The Burgers equation with multiple RVs. Now let us compute (2.28) with three Poisson RVs with mean λ $(d = 3)$. We solve problem (2.28) with the random forcing term $\sigma \sum_{j=1}^{d} c_1(\xi_j)\psi_j(x,t) = \sigma \sum_{j=1}^{3} c_1(\xi_j)cos(0.1jt)$. We solved the WM propagator (2.32) by the second-order IMEX time splitting scheme (RK2/Crank–Nicolson). For a fixed polynomial expansion order P in Figure 5, we plotted the error defined in (3.2) with respect to time, for different WM order Q. Here we take $P_1 = P_2 = P_3 = P = 2$.

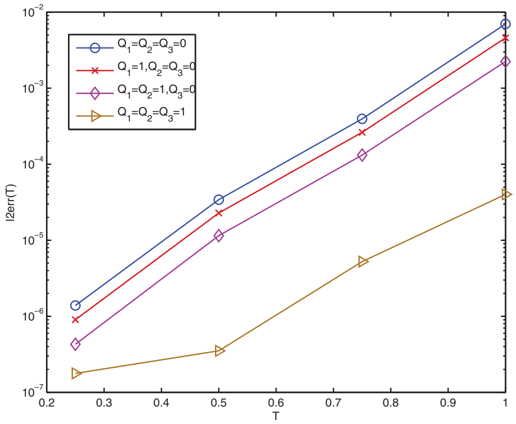

Fig. 7. *The Burgers equation with three Poisson RVs* $\xi_{1,2,3} \sim Pois(\lambda)$ $(d = 3)$*: error defined in* (3.2) *with respect to time, for different WM order* Q*, with parameters* $\lambda = 0.1$*,* $\sigma = 0.1$*,* $y_0 = 1$*,* $\nu = 1/100$*, polynomial order* $P = 2$*, IMEX (RK2/Crank–Nicolson) scheme with time step* $dt = 2.5e-4$*.*

We observe in Figure 7 that the error is not prominently decreased when we increase WM order Q for one or two RVs, but the error is greatly decreased when we increase Q for all three RVs. In this numerical experiment we have also computed the case that $Q_1 = Q_2 = Q_3 = \frac{1}{2}$, and similarly to Figure 4, the error line corresponding to that almost overlapped with the error line for $Q_1 = Q_2 = Q_3 = 0$ in Figure 7. This suggests that when we model the stochastic Burgers equations with multiple discrete RVs, the accuracy in some cases will not be greatly improved by increasing the WM order Q by $\frac{1}{2}$. Therefore, in oder to save computational cost in the WM method for the Burgers equations with multiple discrete RVs, we may use integer values for Q for each RV instead of half-integer values.

3.5. Adaptive WM method. Now let us control the error growth with respect to time under a certain prespecified accuracy. We will show that it is possible to control the error below a certain threshold by increasing the gPC order P and the WM order Q (P–Q refinement). Under a prespecified *adaptive criterion value*, we increase the polynomial order P or the WM order Q, when the absolute value of error is greater than the adaptive criterion value (P-adaptivity and Q-adaptivity).

In Figure 8, we address the long term integration issue of gPC by computing the WM propagator in reaction equation (2.27) with two Poisson RVs with mean λ, for a fixed Q. We plot the error defined in (3.1) with respect to time and we adaptively increase P to keep the error under the indicated adaptive criterion. We observe that increasing the polynomial order P is an effective way to control the error as time progresses for SPDEs with multiple RVs. Besides dealing with the long term integration problem, varying P also allows us to use a smaller polynomial order P at early times, hence expending less computational cost. In gPC, we may also keep the error lower than a value by increasing P; however, increasing P in the gPC propagator costs much more than increasing P in the WM propagator with a small Q.

In Figure 9 we compute the WM propagator in the Burgers equation (2.32) with one Poisson RV, with mean $\lambda = 1$. We plot the error defined in (3.2) with respect to time both in the case that we fix Q or P to control the error to be under the indicated adaptive criterion by increasing P or Q. We observe that increasing the WM expansion order Q is also an effective way to control the error as time progresses.

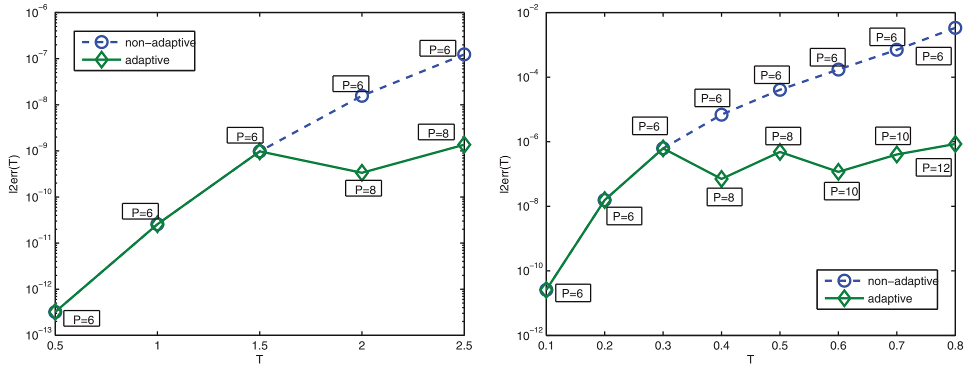

FIG. 8. *Reaction equation with P-adaptivity and two Poisson RVs $\xi_{1,2}$ $\sim$$Pois(\lambda)$ $(d = 2)$: error defined in* (3.1) *with two Poisson RVs by computing the WM propagator in* (2.27) *with respect to time by the RK2 method with fixed WM order $Q = 1$, $y_0 = 1$, $\xi_{1,2} \sim Pois(1)$, $a(\xi_1, \xi_2, t) = c_1(\xi_1; \lambda)c_1(\xi_2; \lambda)$, for fixed polynomial order P (dashed lines), for varied polynomial order P (solid lines), for $\sigma = 0.1$ (left), and $\sigma = 1$ (right). Adaptive criterion values are $l2err(t) \leq 1e-8$(left) and $l2err(t) \leq 1e-6$(right).*

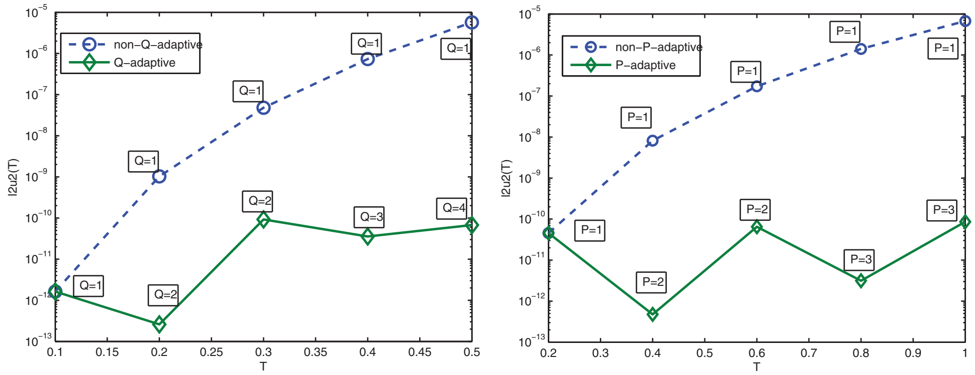

FIG. 9. *The Burgers equation with P-Q adaptivity and one Poisson RV ξ $\sim$$Pois(\lambda)$ $(d = 1, \psi_1(x, t) = 1)$: error defined in* (3.2) *by computing the WM propagator in* (2.32) *with IMEX (RK2/Crank–Nicolson) method ($\lambda = 1$, $\nu = 1/2$, time step $dt = 2e-4$). Fixed polynomial order $P = 6$, $\sigma = 1$, and Q is varied (left); fixed WM order $Q = 3$, $\sigma = 0.1$, and P is varied (right). Adaptive criterion value is $l2u2(T) \leq 1e-10$ (left and right).*

3.6. Computational complexity.

We demonstrate next that the WM propagator is more cost effective in evaluating the statistics of solution than the gPC propagator. Because the computational complexity depends on the form of the equation itself, we analyze this case by case. First let us consider the Burgers equation as an example to compare the complexity of WM and gPC.

3.6.1. The Burgers equation with one RV.

To compare WM and gPC methods for the Burgers equation with one RV ($\xi \sim Pois(\lambda)$), we simply write the gPC and WM propagators separately and compare how they differ from each other. We consider this equation

$$u_t + uu_x = \nu u_{xx} + \sigma c_1(\xi; \lambda), \quad x \in [-\pi, \pi]. \tag{3.3}$$

The gPC propagator for this problem is

$$\frac{\partial \hat{u}_k}{\partial t} + \sum_{m,n=0}^{P} \hat{u}_m \frac{\partial \hat{u}_n}{\partial x} \langle c_m c_n c_k \rangle = \nu \frac{\partial^2 \hat{u}_k}{\partial x^2} + \sigma \delta_{1k}, k = 0, 1, \ldots, P, \tag{3.4}$$

where $\langle c_m c_n c_k \rangle = \int_{\mathbf{S}} d\Gamma(\xi) c_k(\xi) c_m(\xi) c_n(\xi)$.

The WM propagator for this problem is

$$\frac{\partial \tilde{u}_k}{\partial t} + \sum_{p=0}^{Q} \sum_{i=0}^{P} \tilde{u}_i \frac{\partial \tilde{u}_{k+2p-i}}{\partial x} K_{i,k+2p-i,Q} = \nu \frac{\partial^2 \tilde{u}_k}{\partial x^2} + \sigma \delta_{1k}, k = 0, 1, \ldots, P. \tag{3.5}$$

The only difference between gPC and WM propagators is between the term $\sum_{m,n=0}^{P} \hat{u}_m \frac{\partial \hat{u}_n}{\partial x} \langle c_m c_n c_k \rangle$ in gPC and the term $\sum_{p=0}^{Q} \sum_{i=0}^{P} \tilde{u}_i \frac{\partial \tilde{u}_{k+2p-i}}{\partial x} K_{i,k+2p-i,p}$ in WM. Assuming that we are going to solve (3.4) and (3.5) with the same time stepping scheme and the same spatial discretization, for each time step, let us also assume that the computational complexity of computing one term like $\hat{u}_i \frac{\partial \hat{u}_j}{\partial x}$ is α, while the complexity for the rest of the linear terms is 1.

Under this assumption, in (3.4) for gPC, we have $(P+1)$ equations in the system, each one with complexity $1 + (P+1)^2 \alpha$, and therefore the total complexity is $(P+1)[1 + (P+1)^2 \alpha]$.

In (3.5) for WM, we still have $(P+1)$ equations in the system. By denoting the number of terms like $\tilde{u}_i \frac{\partial \tilde{u}_j}{\partial x}$ in the whole WM propagator as $C(P,Q)$, the total complexity will be $(P+1) + C(P,Q)\alpha$, and we compute $C(P,Q)$ numerically. We demonstrate how to count $C(P,Q)$ when $P = 4, Q = 1/2$ in Figure 10: there are five 4×4 grids, for $k = 0, 1, 2, 3, 4$, respectively, for $\tilde{u}_i \frac{\partial \tilde{u}_j}{\partial x}$ in all the five equations in the WM propagator. The horizontal axis represents the index i for $\tilde{u}_i$ and the vertical axis represents the index j for $\frac{\partial \tilde{u}_j}{\partial x}$. We marked the terms like $\tilde{u}_i \frac{\partial \tilde{u}_j}{\partial x}$ in the kth equation in the WM propagator by drawing a circle at the (i,j) dot on the kth grid. In this way we may visualize the nonlinear terms in the propagator and hence visualize the main computational complexity. In the WM method for $P = 4, Q = 1/2$, only the circled dots are considered in the propagator, however, in the gPC method for $P = 4$, all the dots on the five grids are considered in the propagator. Hence, we can see how many fewer terms like $\hat{u}_i \frac{\partial \hat{u}_j}{\partial x}$ we need to consider in WM comparing to gPC.

When P is sufficiently large, the ratio for complexity of WM to gPC is approximately $\frac{C(P,Q)}{(P+1)^3}$, ignoring lower order terms on P. Since we observed in Figures 3 and 4 that when $Q = P - 1$, the errors computed from WM propagators are at the same accuracy level as from gPC propagators, we calculate the ratio of complexity between WM and gPC for the Burgers equation with one RV $\frac{C(P,Q=P-1)}{(P+1)^3}$ when $Q = P - 1$ (so that WM and gPC have the same level of accuracy) and $P \geq 2$ as

$$\frac{C(P, Q = P-1)}{(P+1)^3} = 1 - \frac{10 + \frac{1}{6}P(P+1)(P+2)}{(P+1)^3}. \tag{3.6}$$

3.6.2. The Burgers equation with *d* RVs. Now suppose we are going to solve the Burgers equation with d RVs (each RV ξ_i has polynomial expansion order P_i and WM order Q_i):

$$u_t + uu_x = \nu u_{xx} + \sigma c_{m_1}(\xi_1) \ldots c_{m_d}(\xi_d), \quad x \in [-\pi, \pi]. \tag{3.7}$$

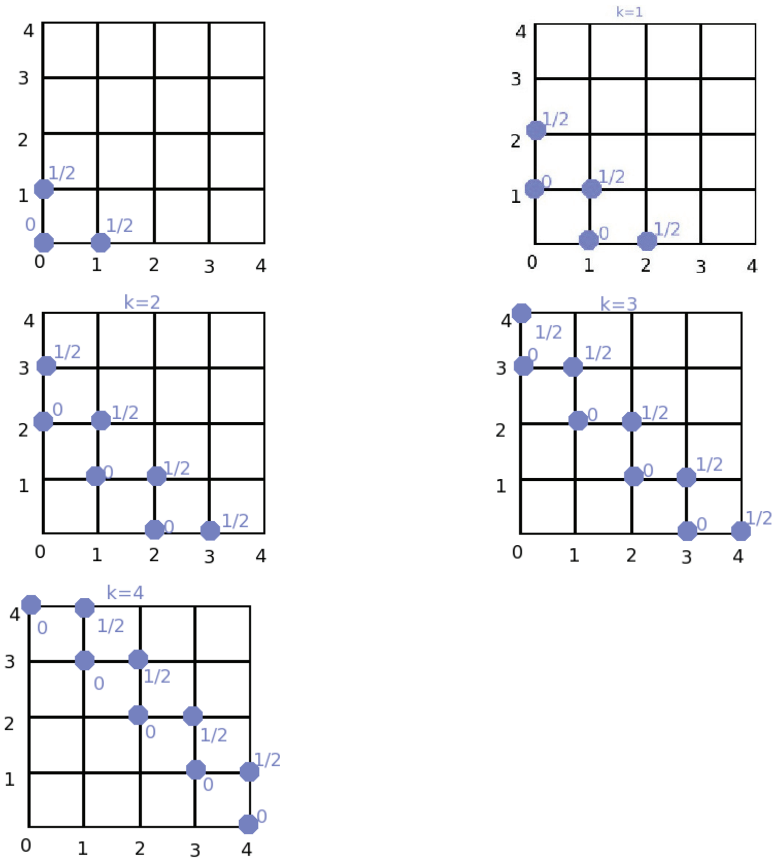

FIG. 10. *Terms in* $\sum_{p=0}^{Q}\sum_{i=0}^{P}\hat{u}_i\frac{\partial\hat{u}_{k+2p-i}}{\partial x}K_{i,k+2p-i,p}$ *for each PDE in the WM propagator for the Burgers equation with one RV in* (3.5) *are denoted by dots on the grids: here* $P=4, Q=\frac{1}{2}, k=0,1,2,3,4$. *Each grid represents a PDE in the WM propagator, labeled by* k. *Each dot represents a term in the sum* $\sum_{p=0}^{Q}\sum_{i=0}^{P}\hat{u}_i\frac{\partial\hat{u}_{k+2p-i}}{\partial x}K_{i,k+2p-i,p}$. *The small index next to the dot is for* p, x *direction is the index* i *for* $\hat{u}_i$, *and* y *direction is the index* $k+2p-i$ *in* $\frac{\partial\hat{u}_{k+2p-i}}{\partial x}$. *The dots on the same diagonal line have the same index* p.

By gPC, we will have $\Pi_{i=1}^{d}(P_i+1)$ equations in the propagator, and if P_i are all equal to P, there will be $(P+1)^d$ equations in the gPC propagator. We will have $(\Pi_{i=1}^{d}(P_i+1))(\Pi_{i=1}^{d}(P_i+1)^2)$ terms like $\hat{u}_k\frac{\partial\hat{u}_j}{\partial x}$. When all the RVs are having the same P, this number is $(P+1)^{3d}$.

By WM, we will still have the same number of equations in the propagator system, but the number of terms like $\hat{u}_k\frac{\partial\hat{u}_j}{\partial x}$ is $\Pi_{i=1}^{d}C(P_i,Q_i)$. Let us assume all the RVs having the same P and Q. This formula can be written as $(C(P,Q))^d$.

When P is sufficiently large (for simplicity we assume $P_i=P, Q_i=Q$ for all $i=1,2,\ldots,d$), the ratio for complexity of WM to gPC is approximately $\frac{C(P,Q)^d}{(P+1)^{3d}}$, ignoring lower order terms on P. We computed the ratio of complexity in Figure 11 for $d=2,3$.

Besides Figure 11, we also want to point out the following observation. From Figure 3 we observed numerically that when $Q\geq P-1$, the error from the WM method with polynomial order P is of the same order as the error from gPC with

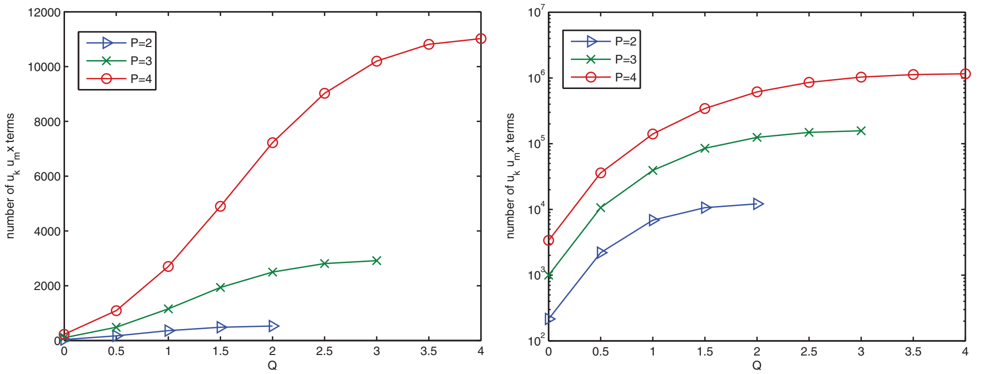

FIG. 11. *The total number of terms as* $\hat{u}_{m_1 \dots m_d} \frac{\partial}{\partial x} \hat{u}_{k_1+2p_1-m_1,\dots,k_d+2p_d-m_d} K_{m_1,k_1+2p_1-m_1,p_1} \dots K_{m_d,k_d+2p_d-m_d,p_d}$ *in the WM propagator for the Burgers equation with d RVs, as* $C(P,Q)^d$*: for dimensions* $d = 2$ *(left) and* $d = 3$ *(right). Here we assume* $P_1 = \cdots = P_d = P$ *and* $Q_1 = \cdots = Q_d = Q$.

TABLE 2

Computational complexity ratio to evaluate the $u\frac{\partial u}{\partial x}$ *term in the Burgers equation with d RVs between WM and gPC, as* $\frac{C(P,Q)^d}{(P+1)^{3d}}$*: here we take the WM order as* $Q = P - 1$*, and gPC with order* P*, in different dimensions* $d = 2, 3,$ *and* 50.

$\frac{C(P,Q)^d}{(P+1)^{3d}}$	$P = 3, Q = 2$	$P = 4, Q = 3$	$P = 5, Q = 4$
$d = 2$	$\frac{2500}{4^6} \approx 61.0\%$	$\frac{10201}{5^6} \approx 65.3\%$	$\frac{31329}{6^6} \approx 67.2\%$
$d = 3$	$\frac{12500}{4^9} \approx 47.7\%$	$\frac{1030301}{5^9} \approx 52.8\%$	$\frac{5545233}{6^9} \approx 55.0\%$
$d = 50$	$\frac{8.89e{+}84}{4^1 50} \approx 0.000436\%$	$\frac{1.64e{+}100}{5^1 50} \approx 0.0023\%$	$\frac{2.5042e{+}112}{6^1 50} \approx 0.0047\%$

polynomial order P. So let us consider the computational cost ratio $\frac{C(P,Q)^d}{(P+1)^{3d}}$ between the two methods for WM with order $Q = P - 1$, and gPC with order P, in Table 2.

We conclude from Figure 11 and Table 2 that (1) the larger the P, the bigger the cost ratio between WM to gPC ($\frac{C(P,Q)^d}{(P+1)^{3d}}$); (2) the higher the dimensions, for the same order P and Q, the lower the ratio $\frac{C(P,Q)^d}{(P+1)^{3d}}$. In other words, the higher the dimensions, the less WM is going to cost than gPC for the same accuracy.

4. Conclusions.

We presented a new WM expansion to approximate polynomial nonlinear terms in SPDEs with random inputs of arbitrary discrete measure with finite moments, on which orthogonal polynomials can be constructed numerically [41, 16, 18, 6, 40]. Specifically, we derived WM propagators for a stochastic reaction and a Burgers equation in (2.27) and (2.32) with multiple discrete RVs. The error was effectively improved by at least two to eight orders of magnitude when the WM order Q was increased into a larger integer in Figures 1 and 2. Linear and nonlinear SPDEs with multiple RVs were considered in Figures 5 and 7 as the first step towards application of the WM method to nonlinear SPDEs with stochastic processes, such as Levy processes with jumps. We found the smallest WM order Q for gPC polynomial order P in the WM method to be $Q = P - 1$ in order to achieve the same order of magnitude of error in gPC with polynomial order P or PCM with $(P+1)$ collocation points, by computing the Burgers equation with one Poisson RV in Figure 3. When Q was larger than $(P - 1)$, the error remained almost constant as in Figure 4. We proposed an adaptive WM method in section 3.5, by increasing the gPC order P

and the WM order Q as a possible solution to control the error growth in long term integration in gPC, shown in Figures 8 and 9. With $Q = P - 1$ we estimated and compared the computational complexity between the WM method and gPC for a stochastic Burgers equation with d RVs in section 3.5. The WM method required much less computational complexity than gPC, especially in higher dimensions, as in Table 2. However WM is still more expensive than PCM or sparse PCM.

Appendix. Algorithm to generate coefficients $\boldsymbol{a(k, m, n)}$ numerically. We now present an algorithm that we used to compute coefficients $a(k, m, n)$ numerically in (2.7). Let us take the monic Charlier polynomials $\{c_i(\xi; \lambda), i = 0, 1, 2, \ldots\}$ for a $\xi \sim Pois(\lambda)$ as an example. Charlier polynomials are known as

$$c_n(\xi; \lambda) = \sum_{j=0}^{n} \xi^j \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) (-1)^{n-k} \lambda^{-k} s(k, j) = \sum_{j=0}^{n} C_{nj} \xi^j, \tag{A.1}$$

where $s(k, j)$ is a Stirling number of the first kind, and C_{nj} is the coefficient for ξ^j in $c_n(\xi; \lambda)$. Let us determine the value of $a(m + n, m, n)$ to $a(0, m, n)$ by matching coefficients for both sides of (2.7) from x^{m+n} to x^0 one by one. Equation (2.7) in terms of Charlier polynomials is written as

$$c_m(\xi; \lambda) c_n(\xi; \lambda) = \sum_{k=0}^{m+n} a(k, m, n) c_k(\xi; \lambda), \quad m, n = 0, 1, 2, \ldots. \tag{A.2}$$

Since the $\{c_i(\xi; \lambda), i = 0, 1, 2, \ldots\}$ are monic polynomials, the coefficient of x^{m+n} on the left-hand side is 1, and on the right-hand side it is $a(m + n, m, n)$. Therefore $a(m + n, m, n) = 1$.

Next by matching the coefficient of x^{m+n-1} on both sides, we have $C_{m,m} C_{n,n-1} + C_{m,m-1} C_{nn} = a(m+n, m, n) C_{m+n,m+n-1} + a(m+n-1, m, n) C_{m+n-1,m+n-1}$, therefore $a(m + n - 1, m, n) = C_{n,n-1} + C_{m,m-1} - C_{m+n,m+n-1}$.

In the same way by matching the coefficients of x^{m+n-2} on both sides, and by knowing the value of $a(m + n, m, n)$ and $a(m + n - 1, m, n)$, we have $a(m + n - 2, m, n) = (C_{m,m-2} + C_{m,m-1} C_{n,n-1} + C_{n,n-2}) - (C_{m+n,m+n-2} + a(m + n - 1, m, n) C_{m+n-1,m+n-2})$.

If we continue this procedure, all the coefficients $a(k, m, n), k = 0, 1, 2, \ldots, m + n$, can be generated. Although we take the Charlier polynomials as an example here, this procedure obviously is applicable to any other orthogonal polynomials.

REFERENCES

[1] P.L. Artes, J.S. Dehesa, A. Martinez-Finkelshtein, and J. Sanchez-Ruiz, *Linearization and connection coefficients for hypergeometric-type polynomials*, J. Comput. Appl. Math., 99 (1998), pp. 5–26.

[2] R. Askey and J. Wilson, *Some basic hypergeometric polynomials that generalize Jacobi polynomials*, Mem. Amer. Math. Soc., 54 (1985).

[3] O.E. Barndorff-Nielsen and N. Shephard, *Non-Gaussian Ornstein-Uhlenbeck-based models and some of their uses in financial economics*, J. R. Stat. Soc. B Stat. Methodol., 63 (2001), pp. 1–42.

[4] O.E. Barndorff-Nielsen and N. Shephard, *Normal modified stable processes*, Theory Probab. Math. Statist., 65 (2002), pp. 1–20.

[5] D. Bell, *The Malliavin Calculus*, Dover, Mineola, NY, 2006.

[6] D. Boley and G. H. Golub, *A survey of matrix inverse eigenvalue problems*, Inverse Problems, 3 (1987), pp. 595–622.

[7] L. Bondesson, *On simulation from infinitely divisible distributions*, Adv. Appl. Probab., 14 (1982), pp. 855–869.

[8] S. Boyarchenko and S. Levendorskii, *Option pricing for truncated Levy processes*, Int. J. Theor. Appl. Finance, 3 (2000), pp. 549–552.

[9] P. Carr, H. Geman, D.B. Madan, and M. Yor, *The fine structure of asset returns: An empirical investigation*, J. Business, 75 (2002), pp. 303–325.

[10] P. Carr, H. Geman, D.B. Madan, and M. Yor, *Stochastic volatility for Levy processes*, Math. Finance, 13 (2003), pp. 345–382.

[11] T.S. Chihara, *An introduction to orthogonal polynomials*, Math. Appl. 13, Gordon and Breach, New York, 1978.

[12] J. Dalibard and Y. Castin, *Wave-function approach to dissipative processes in quantum optics*, Phys. Rev. Lett., 68 (1992), pp. 580–583.

[13] R.L. Dobrushin and R.A. Minlos, *Polynomials in linear random functions*, Russian Math. Surveys, 32 (1977) pp. 71–127.

[14] R. Erban and S.J. Chapman, *Stochastic modelling of reaction diffusion processes: Algorithms for bimolecular reactions*, Phys. Biol., 6 (2009), 046001.

[15] A. Erdelyi, W. Magnus, F. Oberhettinger, and F.G. Tricomi, *Higher Transcendental Functions*, Vol. 2, Krieger, New York, 1981, p. 226.

[16] H.J. Fischer, *On the condition of orthogonal polynomials via modified moments*, Z. Anal. Anwendungen, 15 (1996), pp. 1–18.

[17] J. Foo and G.E. Karniadakis, *The multi-element probabilistic collocation method (ME-PCM): Error analysis and applications*, J. Comput. Phys., 227 (2008), pp. 9572–9595.

[18] W. Gautschi, *On generating orthogonal polynomials*, SIAM J. Sci. Stat. Comput., 3 (1982), pp. 289–317.

[19] M. Grothaus, Y.G. Kondratiev, and G.F. Us, *Wick calculus for regular generalized functions*, Random Oper. Stochastic Equations, 7 (1999), pp. 263–290.

[20] T. Hida and N. Ikeda, *Analysis on Hilbert space with reproducing kernel arising from multiple Wiener integral*, in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 2, University of California Press, Los Angeles, 1967, pp. 117–143.

[21] H. Holden, B. Oksendal, J. Uboe, and T. Zhang, *Stochastic Partial Differential Equations*, 2nd ed., Springer, London, 2010.

[22] S. Janson, *Gaussian Hilbert Spaces*, Cambridge University Press, Cambridge, 1997.

[23] C. Jordan, *Calculus of Finite Differences*, 3rd ed., Chelsea, New York, 1965, p. 473.

[24] K. Karhunen, *Uber lineare methoden in der wahrscheinlichkeitsrechnung*, Ann. Acad. Sci. Fenn. Math., 37 (1947), pp. 1–79.

[25] S. Kaligotla and S.V. Lototsky, *Wick product in the stochastic Burgers equation: A curse or a cure?*, Asymptot. Anal., 75 (2011), pp. 145–168.

[26] D. Kim, D. Stanton, and J. Zeng, *The combinatorics of the Al-Salam-Chihara q-Charlier polynomials*, Sém. Lothar. Combin., 54 (2006), B54i.

[27] I. Koponen, *Analytic approach to the problem of convergence of truncated Levy flights towards the Gaussian stochastic process*, Phys. Rev. E (3), 52 (1995), pp. 1197–1199.

[28] S. Kusuoka and D. Stroock, *Applications of Malliavin calculus* I, *stochastic analysis*, in Anderson Localization: in Proceedings of the Fourth Taniguchi International Symposium Sanda-Shi, Japan, 1981, Springer-Verlag, Heidelberg, 1982, pp. 271–306.

[29] S. Kusuoka and D. Stroock, *Applications of Malliavin Calculus* II, J. Fac. Sci. Univ. Tokyo Sect. 1A Math., 32 (1985), pp. 1–76.

[30] S. Kusuoka and D. Stroock, *Applications of Malliavin Calculus* III, J. Fac. Sci. Univ. Tokyo Sect. 1A Math., 34 (1987), pp. 391–442.

[31] A. Lanconelli and L. Sportelli, *A connection between the Poissonian Wick product and the discrete convolution*, Commun. Stochastic Anal., 5 (1998), pp. 689–699.

[32] M. Loeve, *Probability Theory*, Vol. II, 4th ed., Grad. Texts in Math. 46, Springer-Verlag, New York, 1978.

[33] S.V. Lototsky, B.L. Rozovskii, and D. Selesi, *On generalized Malliavin calculus*, Stochastic Process. Appl., 122 (2012), pp. 808–843.

[34] P. Malliavin and A. Thalmaier, *Stochastic Calculus of Variations in Mathematical Finance*, Springer, Berlin, 2004.

[35] R.N. Mantegna and H.E. Stanley, *Stochastic process with ultraslow convergence to a Gaussian: The truncated Levy flight*, Phys. Rev. Lett., 73 (1994), pp. 2946–2949.

[36] R. Mikulevicius and B.L. Rozovskii, *On unbiased stochastic Navier-Stokes equations*, Probab. Theory Related Fields, (2011), pp. 1–48.

[37] E.A. Novikov, *Infinitely divisible distributions in turbulence*, Phys. Rev. E (3), 50 (1994), pp. 3303–3305.

[38] D. Nualart, *The Malliavin Calculus and Related Topics*, 2nd ed., Springer-Verlag, Berlin, 2006.

[39] D. Nualart, *Malliavin Calculus and its Applications*, CBMS Reg. Conf. Ser. Math. 110, AMS, Providence, RI, 2009.

[40] S. Oladyshkin and W. Nowak, *Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion*, Reliability Eng. System Safety, 106 (2012), pp. 179–190.

[41] S.B. Park and J.-H. Kim, *Integral evaluation of the linearization coefficients of the product of two Legendre polynomials*, J. Appl. Math. Comput., 20 (2006), pp. 623–635.

[42] S. Roman, *The Poisson-Charlier polynomials*, in The Umbral Calculus, Academic, New York, 1984, pp. 119–122.

[43] J. Rosinski, *On series representations of infinitely divisible random vectors*, Ann. Probab., 18 (1990), pp. 405–430.

[44] J. Rosinski, *Series representations of Levy processes from the perspectie of point processes*, in O.E. Barndorff-Nielsen, T. Mikosch, and S. Resnick, eds, Levy Processes: Theory and Applications, Birkhäuser, Boston, 2001, pp. 401–415.

[45] G. Szego, *Orthogonal Polynomials*, 4th ed., AMS, Providence, RI, 1975, pp. 34–35.

[46] E.I. Tzenova, I.J.B.F. Adan, and V.G. Kulkarni, *Fluid models with jumps*, Stochastic Models, 21 (2005), pp. 37–55.

[47] D. Venturi, X. Wan, R. Mikulevicius, B.L. Rozovskii, AND G.E. Karniadakis, *Wick-Malliavin approximation to nonlinear stochastic PDEs: Analysis and simulations*, R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci., 469 (2013), 0001.

[48] G.C. Wick, *The evaluation of the collision matrix*, Phys. Rev. (2), 80 (1950), pp. 268–272.

[49] 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.

[50] D. Xiu and G.E. Karniadakis, *The Wiener–Askey polynomial chaos for stochastic differential equations*, SIAM J. Sci. Comput., 24 (2002), pp. 619–644.

[51] G. Yan and F.B. Hanson, *Option pricing for a stochastic-volatility jump-diffusion model with log uniform jump-amplitudes*, in Proceedings of the 2006 American Control Conference, IEEE, Piscataway, NJ, 2006, pp. 2989–2994.

[52] M. Zheng, X. Wan, and G.E. Karniadakis, *Adaptive multi-element polynomial chaos with discrete measure: Algorithms and application to SPDEs*, Appl. Numer. Math., 90 (2015), pp. 91–110.