MatLab and Julia Toolboxes for Physically-Driven DR-PDEE

Jian-Bing Chen (陈建兵)

Ph.D., University Distinguished Professor

College of Civil Engineering, Tongji University

State Key Laboratory of Disaster Reduction in Civil Engineering, Tongji University

E-mail: chenjb@tongji.edu.cn

Meng-Ze Lyu (律梦泽)

Ph.D., Postdoctoral Fellow

College of Civil Engineering, Tongji University

State Key Laboratory of Disaster Reduction in Civil Engineering, Tongji University

E-mail: lyumz@tongji.edu.cn

Jia-Shu Yang (杨家树)

Ph.D., Postdoctoral Fellow

College of Civil Engineering, Xi'an University of Architecture & Technology

E-mail: jiashu.yang@xauat.edu.cn

Ting-Ting Sun (孙婷婷)

Ph.D. Student

College of Civil Engineering, Tongji University

E-mail: suntingting@tongji.edu.cn

Yi Luo (罗漪)

Ph.D. Student

Institute for Risk & Reliability, Leibniz University Hannover

George R. Brown School of Engineering, Rice University

E-mail: yi.luo@irz.uni-hannover.de, yl205@rice.edu

1. Introduction

The physically-driven DR-PDEE (full spelling: dimension-reduced probability density evolution equation) is a novel method for response determination and reliability analysis of high-dimensional nonlinear stochastic dynamical systems. The advantage of the method is that high-accuracy numerical results of stochastic response and reliability can be obtained, especially for the probability density function (PDF) on the tail and the small probability of failure under rare events, only via the data from a number in the order of magnitude of 10^2 of representative deterministic analyses.

The unified formalism of physically-driven DR-PDEE is developed in Lyu and Chen (2022). The theoretical foundation for response determination of high-dimensional nonlinear stochastic dynamical systems via physically-driven DR-PDEE can be found in Chen and Lyu (2022), Luo et al. (2022b, 2023), and Sun and Chen (2022), and for reliability analysis can be found in Lyu and Chen (2021, 2022, 2023) and Sun et al. (2023), and the application in engineering practices can be found in Luo et al. (2022a) and Lyu et al. (2023).

In this manual, the numerical implementation procedure and corresponding codes for the physically-driven DR-PDEE is given in detail. The main conclusions of the physically-driven DR-PDEE are briefly introduced in Sec. 2, but for short no specific derivation and proof are given. The numerical implementation of the physically-driven DR-PDEE includes two steps, identification of the intrinsic drift functions and solution of the DR-PDEE, which are introduced in detail in Secs. 3 and 4, respectively. Finally, some numerical examples are shown in Sec. 5.

2. Theoretical Foundation for Physically-Driven DR-PDEE

2.1. DR-PDEE for Generic Path-Continuous Processes

Without loss of generality, consider a one-dimensional path-continuous stochastic process $X(t)$[*]. The continuity of $X(t)$ can be described by the *Dynkin-Kinney condition* (Gardiner 2004).

[*] Path-continuous processes are widespread in general physics or engineering problems. Even for a mechanical system under discontinuous excitations (such as Gaussian or Poisson white noise), its responses, such as displacements, velocities or restoring forces, are generally path-continuous. The one-dimensional path-continuous process discussed in this section can be Markovian or non-Markovian, and can be any one-dimensional response of a high-dimensional stochastic dynamical system.

Lemma 2.1. Dynkin-Kinney condition. The transition probability density of $X(t)$ during the interval $[t,t+\Delta t]$ is denoted as $p_X(x,t+\Delta t|x',t)$. If there is

$$\lim_{\Delta t\to 0}\frac{1}{\Delta t}\int_{|x-x'|>\varepsilon} p_X(x,t+\Delta t|x',t)\,\mathrm{d}x=0 \tag{2.1}$$

uniformly in x', t, and Δt for any $\varepsilon>0$, then the sample paths of $X(t)$ are continuous functions of t with probability one.

The following theorem can be given for a path-continuous process $X(t)$.

Theorem 2.2. DR-PDEE. If $X(t)$ is a path-continuous process with known initial condition $X(0)=x_0$, where x_0 is a constant or random variable with known distribution, then its transient PDF, denoted as $p_X(x,t)$, satisfies the following partial differential equation:

$$\frac{\partial p_X(x,t)}{\partial t}=-\frac{\partial[a^{(\mathrm{int})}(x,t)\,p_X(x,t)]}{\partial x}+\frac{1}{2}\frac{\partial^2[b^{(\mathrm{int})}(x,t)\,p_X(x,t)]}{\partial x^2}, \tag{2.2}$$

where $a^{(\mathrm{int})}(x,t)$ and $b^{(\mathrm{int})}(x,t)$ are referred to as *intrinsic drift and diffusion functions*, respectively, and can be expressed as

$$\begin{cases} a^{(\mathrm{int})}(x,t)=\lim\limits_{\Delta t\to 0}\dfrac{1}{\Delta t}\mathrm{E}[\Delta X(t)\mid X(t)=x], \\ b^{(\mathrm{int})}(x,t)=\lim\limits_{\Delta t\to 0}\dfrac{1}{\Delta t}\mathrm{E}\{[\Delta X(t)]^2\mid X(t)=x\}, \end{cases} \tag{2.3}$$

in which $\mathrm{E}(\cdot)$ is the expectation operator; $\Delta X(t)=X(t+\Delta t)-X(t)$ is the increment of $X(t)$ during $[t,t+\Delta t]$.

Eq. (2.2) is called as the *globally-evolving-based generalized density evolution equation (DR-PDEE)*. Likewise, the DR-PDEE for two-dimensional path-continuous (non-Markovian) processes can be given as well.

Corollary 2.3. DR-PDEE for two-dimensions. If $X(t)$ and $V(t)$ are two path-continuous processes with known initial conditions $X(0)=x_0$ and $V(0)=v_0$, where x_0 and v_0 are constants or random variables with known distribution, respectively, then their joint PDF, denoted as $p_{XV}(x,v,t)$, satisfies the following partial differential equation:

$$\frac{\partial p_{XV}(x,v,t)}{\partial t} = -\frac{\partial[a_1^{(\mathrm{int})}(x,v,t)\,p_{XV}(x,v,t)]}{\partial x} - \frac{\partial[a_2^{(\mathrm{int})}(x,v,t)\,p_{XV}(x,v,t)]}{\partial v} + \frac{1}{2}\frac{\partial^2[b_{11}^{(\mathrm{int})}(x,v,t)\,p_{XV}(x,v,t)]}{\partial x^2} + \frac{\partial^2[b_{12}^{(\mathrm{int})}(x,v,t)\,p_{XV}(x,v,t)]}{\partial x\,\partial v} + \frac{1}{2}\frac{\partial^2[b_{22}^{(\mathrm{int})}(x,v,t)\,p_{XV}(x,v,t)]}{\partial v^2}, \tag{2.4}$$

where $a_i^{(\mathrm{int})}(x,v,t)$ and $b_{ij}^{(\mathrm{int})}(x,v,t)$, for $i,j=1,2$, are intrinsic drift and diffusion functions, respectively, and can be expressed as

$$\begin{cases} a_1^{(\mathrm{int})}(x,v,t) = \lim\limits_{\Delta t\to 0}\dfrac{1}{\Delta t}\mathrm{E}[\Delta X(t) \mid X(t)=x;\ V(t)=v], \\ a_2^{(\mathrm{int})}(x,v,t) = \lim\limits_{\Delta t\to 0}\dfrac{1}{\Delta t}\mathrm{E}[\Delta V(t) \mid X(t)=x;\ V(t)=v], \\ b_{11}^{(\mathrm{int})}(x,v,t) = \lim\limits_{\Delta t\to 0}\dfrac{1}{\Delta t}\mathrm{E}\{[\Delta X(t)]^2 \mid X(t)=x;\ V(t)=v\}, \\ b_{12}^{(\mathrm{int})}(x,v,t) = \lim\limits_{\Delta t\to 0}\dfrac{1}{\Delta t}\mathrm{E}[\Delta X(t)\Delta V(t) \mid X(t)=x;\ V(t)=v], \\ b_{22}^{(\mathrm{int})}(x,v,t) = \lim\limits_{\Delta t\to 0}\dfrac{1}{\Delta t}\mathrm{E}\{[\Delta V(t)]^2 \mid X(t)=x;\ V(t)=v\}, \end{cases} \tag{2.5}$$

in which $\Delta V(t)=V(t+\Delta t)-V(t)$ is the increment of $V(t)$ during $[t,t+\Delta t]$ as well.

The proofs of Theorem 2.2 and Corollary 2.3 can be found in Lyu and Chen (2022).

In is worth pointing that Eqs. (2.2) and (2.4) are in the form of Fokker-Planck-Kolmogorov (FPK) equation. However, they are not traditional FPK equations in the sense that: (1) The unknown functions in Eqs. (2.2) and (2.4) are one-time PDFs rather than conditional PDFs; (2) More importantly, they hold for any generic path-continuous stochastic processes, either Markovian or non-Markovian, whereas the traditional FPK equations hold only for Markov processes.

The intrinsic drift and diffusion functions in the DR-PDEE are conditional expectation functions in the mathematic sense. However, they are the physically driving force for evolution of the transient PDF in the physical sense, and can be identified by the mechanism of the original high-dimensional equation of motion. In this sense, the method for the determination of transient PDF of a response quantity of interest via "identifying intrinsic drift/diffusion function - solving DR-PDEE" can be called as "*physically-driven DR-PDEE*" (Chen & Lyu 2022, Luo et al. 2022b, Lyu & Chen 2022, Sun & Chen 2022).

2.2. DR-PDEE for Absorbing Boundary Processes

Definition 2.4. First-passage reliability. The *first-passage reliability* of a stochastic process $X(t)$ within a specific safe domain Ω_s can be defined as

$$p_R(t) = \Pr\{X(\tau) \in \Omega_s,\ 0 \leqslant \tau \leqslant t\}, \tag{2.6}$$

where $X(t)$ can be any response process of a high-dimensional system; $\Pr\{\cdot\}$ denotes the probability of the bracketed random event; Ω_s is defined as an open set on the real number field with the boundary $\partial\Omega_s$. The first-passage reliability, $p_R(t)$, is a deterministic function with respect to t, and its complement is called as the *probability of failure*, i.e.,

$$p_F(t) = 1 - p_R(t) = \Pr\{X(\tilde{\tau}) \notin \Omega_s,\ \exists \tilde{\tau} \in [t_0, t]\}. \tag{2.7}$$

A feasible way for solving $R(t)$ is to construct an *absorbing boundary process (ABP)* corresponding to $X(t)$ in terms of the safe domain Ω_s. If $X(t)$ is a path-continuous process, its ABP can be given by the following definition.

Difinition 2.5. Absorbing boundary process. The ABP corresponding to a path-continuous process $X(t)$ under the safe domain Ω_s can be defined as

$$\breve{X}(t) = \begin{cases} X(t), & \text{if } t \leqslant T, \\ X(T), & \text{if } t > T, \end{cases} \tag{2.8}$$

where T is the *first-passage time* (Redner 2001), i.e.,

$$T = \inf\{t |\ X(t) \notin \Omega_s,\ t \geqslant 0\}, \tag{2.9}$$

in which $\inf\{\cdot\}$ denotes the infimum of the bracketed variable.

A schematic illustration of the sample paths of $X(t)$ and $\breve{X}(t)$ is shown in Fig. 2.1.

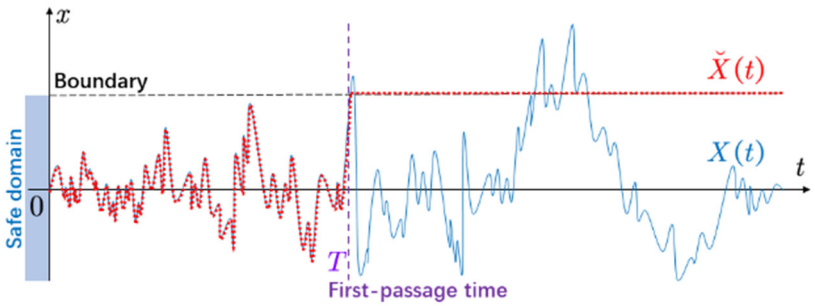


Fig. 2.1. Schematic illustration of the sample path of ABP.

Clearly, $\breve{X}(t)$ is a path-continuous process if $X(t)$ is path-continuous as well. Hence, according to Theorem 2.2, the transient PDF of $\breve{X}(t)$, denoted as $p_{\breve{X}}(x,t)$, satisfies the following DR-PDEE:

$$\frac{\partial p_{\breve{X}}(x,t)}{\partial t} = -\frac{\partial\left[\breve{a}^{(\mathrm{int})}(x,t)\,p_{\breve{X}}(x,t)\right]}{\partial x} + \frac{1}{2}\frac{\partial^2\left[\breve{b}^{(\mathrm{int})}(x,t)\,p_{\breve{X}}(x,t)\right]}{\partial x^2}, \tag{2.10}$$

where the intrinsic drift and diffusion functions can be expressed as

$$\breve{a}^{(\mathrm{int})}(x,t) = \begin{cases} \lim\limits_{\Delta t\to 0}\dfrac{1}{\Delta t}\mathrm{E}\left[\Delta X(t)\,\middle|\,\breve{X}(t)=x\right], & \text{if } x\in\Omega_{\mathrm{s}}, \\ 0, & \text{if } x\in\partial\Omega_{\mathrm{s}}, \end{cases} \tag{2.11}$$

and

$$\breve{b}^{(\mathrm{int})}(x,t) = \begin{cases} \lim\limits_{\Delta t\to 0}\dfrac{1}{\Delta t}\mathrm{E}\left\{\left[\Delta X(t)\right]^2\,\middle|\,\breve{X}(t)=x\right\}, & \text{if } x\in\Omega_{\mathrm{s}}, \\ 0, & \text{if } x\in\partial\Omega_{\mathrm{s}}, \end{cases} \tag{2.12}$$

respectively. The derivation of Eq. (2.10) can be found in Lyu and Chen (2022).

Then, the first-passage reliability $p_{\mathrm{R}}(t)$ can be solved as the integration of $p_{\breve{X}}(x,t)$ in the safe domain Ω_{s}, i.e.,

$$p_{\mathrm{R}}(t) = \Pr\left\{\breve{X}(t)\in\Omega_{\mathrm{s}}\right\} = \int_{\Omega_{\mathrm{s}}} p_{\breve{X}}(x,t)\,\mathrm{d}x. \tag{2.13}$$

The intrinsic drift and diffusion functions can also be called as the *effective drift and diffusion coefficients* in the sense that they are analytically (Sun & Chen 2022, Luo et al. 2023) or numerically (Lyu & Chen 2021, Chen & Lyu 2022) available.

2.3. Establishment of DR-PDEE for Specific Stochastic Systems

The establishment of the DR-PDEE for a quantity of interest in a high-dimensional stochastic system enforced by additive Gaussian white or non-white noise is introduced in this subsection.

Case 2.1. The quantity of interest has non-zero diffusion.

Consider an n-dimensional process vector $\boldsymbol{Y}(t) = (Y_1(t),\cdots,Y_n(t))^{\mathrm{T}}$ governed by the following *stochastic differential equations (SDEs)*:

$$\mathrm{d}\boldsymbol{Y}(t) = \boldsymbol{f}[\boldsymbol{Y}(t),\boldsymbol{\Theta},t]\,\mathrm{d}t + \mathbf{g}\,\mathrm{d}\boldsymbol{W}(t), \tag{2.14}$$

where $\boldsymbol{f}(\cdot)$ is an n-dimensional deterministic function vector; $\mathbf{g}$ is an $n\times r$ constant matrix; $\boldsymbol{\Theta}$ is an s-dimensional random vector; $\boldsymbol{W}(t)$ is an r-

dimensional Wiener process vector with $r\times r$ intensity matrix $\boldsymbol{\sigma}_{\mathrm{w}}$, and the expectation vector and variance matrix of its increment are

$$\mathrm{E}[\mathrm{d}\boldsymbol{W}(t)]=\boldsymbol{0}_r,\ \text{and}\ \mathrm{E}[\mathrm{d}\boldsymbol{W}(t)\mathrm{d}\boldsymbol{W}^{\mathrm{T}}(t)]=\boldsymbol{\sigma}_{\mathrm{w}}\mathrm{d}t, \tag{2.15}$$

respectively. The initial condition of $\boldsymbol{Y}(t)$ is $\boldsymbol{Y}(0)=\boldsymbol{y}_0$, where $\boldsymbol{y}_0$ is a constant vector or random vector with known PDF $p_{\boldsymbol{Y}_0}(\boldsymbol{y})$.

If the l-th component of $\boldsymbol{Y}(t)$, i.e., $Y_l(t)$ for $1\leqslant l\leqslant n$, is of interest, then $Y_l(t)$ is a path-continuous process, and according to Theorem 2.2, its transient PDF, denoted as $p_{Y_l}(y,t)$, satisfies the DR-PDEE. According to the l-th component of Eq. (2.14), namely,

$$\mathrm{d}Y_l(t)=f_l[\boldsymbol{Y}(t),\boldsymbol{\Theta},t]\mathrm{d}t+\boldsymbol{g}_{(l,\cdot)}\mathrm{d}\boldsymbol{W}(t), \tag{2.16}$$

the DR-PDEE in terms of $Y_l(t)$ can be written as

$$\frac{\partial p_{Y_l}(y,t)}{\partial t}=-\frac{\partial[a^{(\mathrm{int})}(y,t)\,p_{Y_l}(y,t)]}{\partial y}+\frac{\sigma_{ll}}{2}\frac{\partial^2 p_{Y_l}(y,t)}{\partial y^2}, \tag{2.17}$$

where σ_{ll} is the (l,l) element of $n\times n$ matrix $\boldsymbol{\sigma}=\mathbf{g}\boldsymbol{\sigma}_{\mathrm{w}}\mathbf{g}^{\mathrm{T}}$; $a^{(\mathrm{int})}(y,t)$ is the intrinsic drift function expressed as

$$a^{(\mathrm{int})}(y,t)=\mathrm{E}\{f_l[\boldsymbol{Y}(t),\boldsymbol{\Theta},t]\mid Y_l(t)=y\} \tag{2.18}$$

according to Eq. (2.3).

In Case 2.1, if $\sigma_{ll}=0$, i.e., $Y_l(t)$ is not directly related to diffusion, then Eq. (2.17) becomes a hyperbolic partial differential equation (PDE) with singular intrinsic drift function at initial time. To establish the DR-PDEE that is easier to solve numerically, Case 2.2 should be considered, and an *adjoint process* with non-zero diffusion should be introduced.

Case 2.2. The quantity of interest is not directly related to diffusion, but its velocity has non-zero diffusion.

Consider an m-degree-of-freedom (-DOF) stochastic system enforced by Gaussian white noise. The equations of motion can be written as

$$\ddot{\boldsymbol{X}}(t)-\boldsymbol{f}\left[\boldsymbol{X}(t),\dot{\boldsymbol{X}}(t),\boldsymbol{\Theta},t\right]=\mathbf{g}\boldsymbol{\xi}(t), \tag{2.19}$$

where $\boldsymbol{X}(t)$, $\dot{\boldsymbol{X}}(t)$, and $\ddot{\boldsymbol{X}}(t)$ are the m-dimensional displacement, velocity, and acceleration process vector, respectively; $\boldsymbol{f}(\cdot)$ is an m-dimensional deterministic function vector; $\mathbf{g}$ is an $m\times r$ constant matrix; $\boldsymbol{\Theta}$ is an s-dimensional random vector; $\boldsymbol{\xi}(t)$ is an r-dimensional Gaussian white noise process vector with $r\times r$ intensity matrix $\boldsymbol{\sigma}_{\mathrm{w}}$, and its expectation vector and autocorrelation matrix are

$$\mathrm{E}[\boldsymbol{\xi}(t)]=\mathbf{0}_r,\ \text{and}\ \mathrm{E}[\boldsymbol{\xi}(t)\boldsymbol{\xi}^{\mathrm{T}}(t+\tau)]=\boldsymbol{\sigma}_{\mathrm{w}}\delta(\tau), \tag{2.20}$$

respectively. The initial conditions are $\boldsymbol{X}(0)=\boldsymbol{x}_0$ and $\dot{\boldsymbol{X}}(0)=\boldsymbol{v}_0$ where $\boldsymbol{x}_0$ and $\boldsymbol{v}_0$ are constant vectors or random vectors with known joint PDF $p_{\boldsymbol{X}_0\boldsymbol{V}_0}(\boldsymbol{x},\boldsymbol{v})$. By denoting $\boldsymbol{V}(t)=\dot{\boldsymbol{X}}(t)$, Eq. (2.19) can be rewritten as $2m$-dimensional SDEs, namely

$$\begin{cases}\mathrm{d}\boldsymbol{X}(t)=\boldsymbol{V}(t)\mathrm{d}t,\\ \mathrm{d}\boldsymbol{V}(t)=\boldsymbol{f}[\boldsymbol{X}(t),\boldsymbol{V}(t),\boldsymbol{\Theta},t]\mathrm{d}t+\mathbf{g}\mathrm{d}\boldsymbol{W}(t),\end{cases} \tag{2.21}$$

where $\boldsymbol{W}(t)$ is an r-dimensional Wiener process vector with $r\times r$ intensity matrix $\boldsymbol{\sigma}_{\mathrm{w}}$. Eq. (2.21) has the same form as Eq. (2.14), but $\boldsymbol{X}(t)$ is not directly related to diffusion.

If the l-th component of $\boldsymbol{X}(t)$, i.e., $X_l(t)$ for $1\leqslant l\leqslant m$, is of interest, then $V_l(t)$ can be introduced as an adjoint process, and according to Corollary 2.3, their transient joint PDF, denoted as $p_{X_lV_l}(x,v,t)$, satisfies the DR-PDEE. According to the l-th and $(l+m)$-th components of Eq. (2.21), namely,

$$\begin{cases}\mathrm{d}X_l(t)=V_l(t)\mathrm{d}t,\\ \mathrm{d}V_l(t)=f_l[\boldsymbol{X}(t),\boldsymbol{V}(t),\boldsymbol{\Theta},t]\mathrm{d}t+\boldsymbol{g}_{(l,\cdot)}\mathrm{d}\boldsymbol{W}(t),\end{cases} \tag{2.22}$$

the DR-PDEE in terms of $X_l(t)$ and $V_l(t)$ can be written as

$$\frac{\partial p_{X_lV_l}(x,v,t)}{\partial t}=-v\frac{\partial p_{X_lV_l}(x,v,t)}{\partial x}-\frac{\partial[a^{(\mathrm{int})}(x,v,t)\,p_{X_lV_l}(x,v,t)]}{\partial v}+\frac{\sigma_{ll}}{2}\frac{\partial^2 p_{X_lV_l}(x,v,t)}{\partial v^2}, \tag{2.23}$$

where σ_{ll} is the (l,l) element of the $n\times n$ matrix $\boldsymbol{\sigma}=\mathbf{g}\boldsymbol{\sigma}_{\mathrm{w}}\mathbf{g}^{\mathrm{T}}$ as well; $a^{(\mathrm{int})}(x,v,t)$ is the intrinsic drift function expressed as

$$a^{(\mathrm{int})}(x,v,t)=\mathrm{E}\{f_l[\boldsymbol{X}(t),\boldsymbol{V}(t),\boldsymbol{\Theta},t]\mid X_l(t)=x;\ V_l(t)=v\} \tag{2.24}$$

according to Eq. (2.5).

In Case 2.2, if $\sigma_{ll}=0$, i.e., $V_l(t)$ is not directly related to diffusion either, then some other quantity which has non-zero diffusion should be chosen as the adjoint process instead of $V_l(t)$, as discussed in Case 2.3.

Case 2.3. Neither quantity of interest nor its velocity has diffusion.

Consider an m-DOF stochastic system enforced by Gaussian non-white noise. Its equations of motion can be written as

$$\ddot{\boldsymbol{X}}(t)-\boldsymbol{f}\left[\boldsymbol{X}(t),\dot{\boldsymbol{X}}(t),\boldsymbol{\Theta},t\right]=\mathbf{g}\boldsymbol{F}(t), \tag{2.25}$$

which is the same as Eq. (2.19) except the excitation is changed as $\boldsymbol{F}(t)$, an r-dimensional stationary Gaussian non-white noise process vector. Here $\boldsymbol{F}(t)$ has the known r-dimensional expectation vector $\boldsymbol{\mu}_{\mathrm{F}}(t)$ and $r\times r$ power spectral density (PSD) matrix $\mathbf{s}_{\mathrm{F}}(\omega)$. According to the linear filtering theory (Spanos & Mignolet 1989), $\boldsymbol{F}(t)$ can be regarded as a function of the outputs of some linear filter to Gaussian white noise, i.e., $\boldsymbol{F}(t)=\boldsymbol{h}[\boldsymbol{U}_{\mathrm{f}}(t)]$, where $\boldsymbol{h}(\cdot)$ is an r-dimensional function vector; $\boldsymbol{U}_{\mathrm{f}}(t)$ is a d-dimensional process vector governed by the following Itô SDEs:

$$\mathrm{d}\boldsymbol{U}_{\mathrm{f}}(t)=\mathbf{k}_{\mathrm{f}}\boldsymbol{U}_{\mathrm{f}}(t)\mathrm{d}t+\mathbf{g}_{\mathrm{f}}\mathrm{d}\boldsymbol{W}(t), \tag{2.26}$$

in which $\mathbf{k}_{\mathrm{f}}$ is a $d\times d$ matrix; $\mathbf{g}_{\mathrm{f}}$ is a $d\times r$ matrix; $\boldsymbol{W}(t)$ is an r-dimensional Wiener process vector with the $r\times r$ intensity matrix $\boldsymbol{\sigma}_{\mathrm{w}}$. The constant matrices $\mathbf{k}_{\mathrm{f}}$, $\mathbf{g}_{\mathrm{f}}$, and the function $\boldsymbol{h}(\cdot)$ can be determined if PSD is rational, or approximated otherwise. Details can be found in Luo et al. (2022a). By denoting $\boldsymbol{V}(t)=\dot{\boldsymbol{X}}(t)$, Eqs. (2.25) and (2.26) can be combined and rewritten as $(2m+d)$-dimensional SDEs, namely

$$\begin{cases}\mathrm{d}\boldsymbol{X}(t)=\boldsymbol{V}(t)\mathrm{d}t,\\ \mathrm{d}\boldsymbol{V}(t)=\{\boldsymbol{f}[\boldsymbol{X}(t),\boldsymbol{V}(t),\boldsymbol{\Theta},t]+\mathbf{g}\boldsymbol{h}[\boldsymbol{U}_{\mathrm{f}}(t)]\}\mathrm{d}t,\\ \mathrm{d}\boldsymbol{U}_{\mathrm{f}}(t)=\mathbf{k}_{\mathrm{f}}\boldsymbol{U}_{\mathrm{f}}(t)\mathrm{d}t+\mathbf{g}_{\mathrm{f}}\mathrm{d}\boldsymbol{W}(t).\end{cases} \tag{2.27}$$

Eq. (2.27) has the same form as Eq. (2.14) as well, but neither $\boldsymbol{X}(t)$ nor $\boldsymbol{V}(t)$ has non-zero diffusions.

If the l-th component of $\boldsymbol{X}(t)$, i.e., $X_l(t)$ for $1\leqslant l\leqslant m$, is of interest, then a component of $\boldsymbol{U}_{\mathrm{f}}(t)$ with non-zero diffusion, denoted as $U_{\mathrm{f},k}(t)$ for $1\leqslant k\leqslant d$, can be introduced as an adjoint process. According to Corollary 2.3, their transient joint PDF, denoted as $p_{X_lU_{\mathrm{f},k}}(x,u,t)$, satisfies the DR-PDEE. According to the l-th and $(k+2m)$-th components of Eq. (2.27), namely,

$$\begin{cases}\mathrm{d}X_l(t)=V_l(t)\mathrm{d}t,\\ \mathrm{d}U_{\mathrm{f},k}(t)=\boldsymbol{k}_{\mathrm{f},(k,\cdot)}\boldsymbol{U}_{\mathrm{f}}(t)\mathrm{d}t+\boldsymbol{g}_{\mathrm{f},(k,\cdot)}\mathrm{d}\boldsymbol{W}(t),\end{cases} \tag{2.28}$$

the DR-PDEE in terms of $X_l(t)$ and $V_l(t)$ can be written as

$$\frac{\partial p_{X_lU_{\mathrm{f},k}}(x,u,t)}{\partial t}=-\frac{\partial[a_1^{(\mathrm{int})}(x,u,t)\,p_{X_lU_{\mathrm{f},k}}(x,u,t)]}{\partial x}-\frac{\partial[a_2^{(\mathrm{int})}(x,u,t)\,p_{X_lU_{\mathrm{f},k}}(x,u,t)]}{\partial u}+\frac{\sigma_{\mathrm{f},kk}}{2}\frac{\partial^2 p_{X_lU_{\mathrm{f},k}}(x,u,t)}{\partial u^2}, \tag{2.29}$$

where $\sigma_{\mathrm{f},kk}\neq 0$ is the (k,k) element of $d\times d$ matrix $\boldsymbol{\sigma}_{\mathrm{f}}=\mathbf{g}_{\mathrm{f}}\boldsymbol{\sigma}_{\mathrm{w}}\mathbf{g}_{\mathrm{f}}{}^{\mathrm{T}}$; $a_1^{(\mathrm{int})}(x,u,t)$ and $a_2^{(\mathrm{int})}(x,u,t)$ are the intrinsic drift functions expressed as

$$\begin{cases} a_1^{(\mathrm{int})}(x,u,t)=\mathrm{E}\{V_l(t)\mid X_l(t)=x;\ U_{\mathrm{f},k}(t)=u\}, \\ a_2^{(\mathrm{int})}(x,u,t)=\mathrm{E}\{\boldsymbol{k}_{\mathrm{f},(k,\cdot)}\boldsymbol{U}_{\mathrm{f}}(t)\mid X_l(t)=x;\ U_{\mathrm{f},k}(t)=u\}, \end{cases} \tag{2.30}$$

respectively, according to Eq. (2.5).

3. Identification of Intrinsic Drift Functions

3.1. Locally weighted smoothing scatterplots

The basic idea of locally weighted smoothing scatterplots (LOWESS) is proposed by Cleveland (1979). Here, the LOWESS technique is employed for the identification of intrinsic drift functions (Lyu & Chen 2021). For instance, the intrinsic drift function in Eq. (2.24) can be identified via the following numerical procedure of the LOWESS, while for other cases the procedure can be performed similarly.

If the data of responses denoted as

$$\begin{cases} X_l(t_h)=x_q^{(h)}, \\ V_l(t_h)=v_q^{(h)}, \\ A_l(t_h)=f_l[\boldsymbol{X}(t_h),\boldsymbol{V}(t_h),]=a_q^{(h)}, \end{cases} \quad \text{for } h=0,1,\ldots,n_{\mathrm{t}}, \text{ and } q=1,\ldots,n_{\mathrm{sel}}, \tag{3.1}$$

have been obtained by some observed records or representative deterministic analyses of the original system, where $A_l(t)=f_l[\boldsymbol{X}(t),\boldsymbol{V}(t)]$ is a stochastic process related to $X_l(t)$ and $V_l(t)$; the right-hand sides represent the q-th data point of these processes at instant t_h; n_{t} is the total number of time steps; n_{sel} is the total number of data points; then the intrinsic drift function at instant t_h, denoted as

$$a^{(\mathrm{int})}(x,v,t_h)=\mathrm{E}\{A_l(t_h)\mid X_l(t_h)=x;\ V_l(t_h)=v\},\ \text{for } h=0,1,\ldots,n_{\mathrm{t}}, \tag{3.2}$$

can be identified by Algorithm 3.1 under the meshing

$$\begin{cases} x_i=x_{\mathrm{L}}+i\,\Delta\overline{x}, & \text{for } i=0,1,\ldots,\overline{n}_{\mathrm{x}}, \\ v_j=v_{\mathrm{L}}+j\,\Delta\overline{v}, & \text{for } j=0,1,\ldots,\overline{n}_{\mathrm{v}}, \end{cases} \tag{3.3}$$

in the specific state quantities domain $(x,v)\in[x_{\mathrm{L}},x_{\mathrm{U}}]\times[v_{\mathrm{L}},v_{\mathrm{U}}]$, where $\overline{n}_{\mathrm{x}}\Delta\overline{x}=x_{\mathrm{U}}-x_{\mathrm{L}}$, and $\overline{n}_{\mathrm{v}}\Delta\overline{v}=v_{\mathrm{U}}-v_{\mathrm{L}}$.

Algorithm 3.1. LOWESS for identification of intrinsic drift function.

Input: All data points of $x_q^{(h)}$, $v_q^{(h)}$, and $a_q^{(h)}$, for $q=1,\ldots,N_{\mathrm{sel}}$;

(1) For each $(x,v)=(x_i,v_j)$, where $i=0,1,...,\bar{n}_{\mathrm{x}}$, and $j=0,1,...,\bar{n}_{\mathrm{v}}$, perform Steps (2) to (6);

(2) Calculate the distances from (x_i,v_j) to each $(x_q^{(h)},v_q^{(h)})$, i.e.,

$$d_{ij,q}^{(h)}=\sqrt{\left(\frac{x_i-x_q^{(h)}}{\hat{\sigma}_{\mathrm{x}}^{(h)}}\right)^2+\left(\frac{v_j-v_q^{(h)}}{\hat{\sigma}_{\mathrm{v}}^{(h)}}\right)^2},\ \text{for}\ q=1,...,n_{\mathrm{sel}}, \tag{3.4}$$

where $\hat{\sigma}_{\mathrm{x}}^{(h)}$ and $\hat{\sigma}_{\mathrm{v}}^{(h)}$ are the standard deviation estimations of all data points $x_q^{(h)}$ and $v_q^{(h)}$ for $q=1,...,n_{\mathrm{sel}}$, respectively;

(3) Calculate the smooth length, i.e.,

$$\tilde{d}_{ij}^{(h)}=\min_{q=1,...,n_{\mathrm{sel}}}^{(r)}\{d_{ij,q}^{(h)}\}, \tag{3.5}$$

where $\min^{(r)}\{\cdot\}$ represent the r-th value in ascending order of elements in the bracket; $r=\vartheta n_{\mathrm{sel}}$, and ϑ can be taken as 0.2 - 0.5 usually;

(4) Calculate the weight, i.e.,

$$w_{ij,q}^{(h)}=\begin{cases}\left[1-\left(\dfrac{d_{ij,q}^{(h)}}{\tilde{d}_{ij}^{(h)}}\right)^3\right]^3, & \text{if}\ d_{ij,q}^{(h)}<\tilde{d}_{ij}^{(h)},\\ 0, & \text{if}\ d_{ij,q}^{(h)}\geqslant\tilde{d}_{ij}^{(h)},\end{cases}\quad \text{for}\ q=1,...,n_{\mathrm{sel}}; \tag{3.6}$$

(5) Solve the following linear equations

$$\begin{pmatrix}\sum_{q=1}^{n_{\mathrm{sel}}}w_{ij,q}^{(h)} & \sum_{q=1}^{n_{\mathrm{sel}}}w_{ij,q}^{(h)}x_q^{(h)} & \sum_{q=1}^{n_{\mathrm{sel}}}w_{ij,q}^{(h)}v_q^{(h)}\\ \sum_{q=1}^{n_{\mathrm{sel}}}w_{ij,q}^{(h)}x_q^{(h)} & \sum_{q=1}^{n_{\mathrm{sel}}}w_{ij,q}^{(h)}x_q^{(h)2} & \sum_{q=1}^{n_{\mathrm{sel}}}w_{ij,q}^{(h)}x_q^{(h)}v_q^{(h)}\\ \sum_{q=1}^{n_{\mathrm{sel}}}w_{ij,q}^{(h)}v_q^{(h)} & \sum_{q=1}^{n_{\mathrm{sel}}}w_{ij,q}^{(h)}x_q^{(h)}v_q^{(h)} & \sum_{q=1}^{n_{\mathrm{sel}}}w_{ij,q}^{(h)}v_q^{(h)2}\end{pmatrix}\begin{pmatrix}\beta_{0,ij}^{(h)}\\ \beta_{1,ij}^{(h)}\\ \beta_{2,ij}^{(h)}\end{pmatrix}=\begin{pmatrix}\sum_{q=1}^{n_{\mathrm{sel}}}w_{ij,q}^{(h)}a_q^{(h)}\\ \sum_{q=1}^{n_{\mathrm{sel}}}w_{ij,q}^{(h)}x_q^{(h)}a_q^{(h)}\\ \sum_{q=1}^{n_{\mathrm{sel}}}w_{ij,q}^{(h)}v_q^{(h)}a_q^{(h)}\end{pmatrix} \tag{3.7}$$

to obtain the repression coefficients $(\beta_{0,ij}^{(h)},\beta_{1,ij}^{(h)},\beta_{2,ij}^{(h)})^{\mathrm{T}}$;

(6) Calculate the intrinsic drift function, i.e.,

$$a_{ji}^{(\mathrm{int},h)}=a^{(\mathrm{int})}(x_i,v_j,t_h)=\beta_{0,ij}^{(h)}+\beta_{1,ij}^{(h)}x_i+\beta_{2,ij}^{(h)}v_j; \tag{3.8}$$

Output: $\mathbf{a}^{(\mathrm{int},h)}=[a_{ji}^{(\mathrm{int},h)}]_{(\bar{n}_v+1)\times(\bar{n}_x+1)}$ as a numerical matrix of the intrinsic drift function at instant t_h.

The corresponding theoretical elaboration of Algorithm 3.1 can be found in Lyu and Chen (2021).

Algorithm 3.1 can be implemented by the MatLab function `f_EDC_lowess`. The syntax of the function is

```
Aeff = f_EDC_lowess (x, v, X, V, A, theta)
```

that returns the values of the intrinsic drift function evaluated at the values in `x` and `v` via the data `X`, `V`, and `A` with the smooth scale `theta`.

There are six input arguments for the function:

(1) `x`, an $\bar{n}_x+1$-dimensional column vector, the grid sequence of state quantity x given by Eq. (3.3);

(2) `v`, an $\bar{n}_v+1$-dimensional column vector, the grid sequence of state quantity v given by Eq. (3.3);

(3) `X`, an n_{sel}-dimensional column vector, the data points of $X_l(t_h)$ given by Eq. (3.1);

(4) `V`, an n_{sel}-dimensional column vector, the data points of $V_l(t_h)$ given by Eq. (3.1);

(5) `A`, an n_{sel}-dimensional column vector, the data points of $A_l(t_h)$ given by Eq. (3.1);

(6) `theta`, a scalar, the smooth scale taken in (0,1].

There is one output argument of the function:

`Aeff`, an $(\bar{n}_v+1)\times(\bar{n}_x+1)$ matrix, the intrinsic drift function estimation given by Eq. (3.2).

Note that the k-nearest neighbors (k-NN) technique is adopted in this function for better performance.

4. Solution of DR-PDEE

The path integral solution (PIS) is a classical numerical scheme used for solving FPK-like equation (Wehner & Wolfer 1983, Naess & Johnsen 1993). Here this technique is used for solving the DR-PDEE (Chen & Rui 2018, Lyu & Chen 2021).

4.1. PIS for the DR-PDEE in terms of one quantity and its velocity

For Case 2.2, DR-PDEE (2.23) can be solved step by step via the following numerical procedure to obtain the transient PDF of the quantity of interest with the prescribed initial condition.

If the joint PDF of $X_l(t)$ and $V_l(t)$ at instant t_{h-1} is known as $p_{X_lV_l}(x,v,t_{h-1})$, then the joint PDF of $X_l(t)$ and $V_l(t)$ at instant t_h, namely $p_{X_lV_l}(x,v,t_h)$, can be calculated by Algorithm 4.1, for $h=1,\ldots,n_t$. In this algorithm, the PDF is discrete under the meshing

$$\begin{cases} x_i = x_{\mathrm{L}} + i\,\Delta x, & \text{for } i = 0, 1, ..., n_{\mathrm{x}}, \\ v_j = v_{\mathrm{L}} + j\,\Delta v, & \text{for } j = 0, 1, ..., n_{\mathrm{v}}, \end{cases} \tag{4.1}$$

in the specific state quantities domain $(x, v) \in [x_{\mathrm{L}}, x_{\mathrm{U}}] \times [v_{\mathrm{L}}, v_{\mathrm{U}}]$, where $n_{\mathrm{x}} \Delta x = x_{\mathrm{U}} - x_{\mathrm{L}}$, and $n_{\mathrm{v}} \Delta v = v_{\mathrm{U}} - v_{\mathrm{L}}$.

Algorithm 4.1. PIS for the DR-PDEE in terms of one quantity and its velocity.

Input: $\mathbf{p}^{(h-1)} = [p_{ji}^{(h-1)}]_{(n_{\mathrm{v}}+1)\times(n_{\mathrm{x}}+1)}$ as a numerical discrete matrix of $p_{X_t V_t}(x, v, t_{h-1})$ under the meshing (4.1); $\mathbf{a}^{(\mathrm{int}, h-1)} = [a_{ji}^{(\mathrm{int}, h-1)}]_{(\bar{n}_{\mathrm{v}}+1)\times(\bar{n}_{\mathrm{x}}+1)}$ as a numerical matrix of the intrinsic drift function at the instant t_{h-1} under the meshing (3.3);

(1) Calculate the intrinsic drift function matrix $\tilde{\mathbf{a}}^{(\mathrm{int}, h-1)}$ by interpolation of $\mathbf{a}^{(\mathrm{int}, h-1)}$, where each element of $\tilde{\mathbf{a}}^{(\mathrm{int}, h-1)}$ is

$$\tilde{a}_{ji}^{(\mathrm{int}, h-1)} = a^{(\mathrm{int})}(x_i - v_j \Delta t, v_j, t_{h-1}), \text{ for } i = 0, 1, ..., n_{\mathrm{x}} \text{ and } j = 0, 1, ..., n_{\mathrm{v}}; \tag{4.2}$$

(2) Calculate the PDF matrix $\tilde{\mathbf{p}}^{(h-1)}$ by interpolation of $\mathbf{p}^{(h-1)}$, where each element of $\tilde{\mathbf{p}}^{(h-1)}$ is

$$\tilde{p}_{ji}^{(h)} = p_{X_t V_t}(x_i - v_j \Delta t, v_j, t_h), \text{ for } i = 0, 1, ..., n_{\mathrm{x}} \text{ and } j = 0, 1, ..., n_{\mathrm{v}}; \tag{4.3}$$

(3) Normalization: the PDF matrix $\tilde{\mathbf{p}}^{(h-1)}$ should satisfy

$$\left| \sum_{i=0}^{n_{\mathrm{x}}} p_{ji}^{(h-1)} - \sum_{i=0}^{n_{\mathrm{x}}} \tilde{p}_{ji}^{(h-1)} \right| \Delta x \leqslant \epsilon_1, \quad \text{for } j = 0, 1, ..., n_{\mathrm{v}}, \tag{4.4}$$

where ϵ_1 is the allowable error;

(4) For each $x = x_i$, where $i = 0, 1, ..., n_{\mathrm{x}}$, perform Steps (5) to (7);

(5) Calculate the transition probability density (TPD) matrix $\mathbf{q}^{(h,i)}$, where each element is

$$q_{jk}^{(h,i)} = \frac{1}{\sqrt{2\pi\sigma_{ll}\Delta t}} e^{-\frac{[v_j - v_k - a^{(\mathrm{int})}(x_i - v_k \Delta t, v_k, t_{h-1})\Delta t]^2}{2\sigma_{ll}\Delta t}}, \text{ for } j, k = 0, 1, ..., n_{\mathrm{v}}; \tag{4.5}$$

(6) Normalization: the TPD matrix $\mathbf{q}^{(h,i)}$ should satisfy

$$\left| \sum_{j=0}^{n_{\mathrm{v}}} q_{jk}^{(h,i)} \Delta v - 1 \right| \leqslant \epsilon_2, \text{ for } k = 0, 1, ..., n_{\mathrm{v}}, \tag{4.6}$$

where ϵ_2 is the allowable error;

(7) Calculate the i-th column of PDF matrix $\mathbf{p}^{(h)}$ according to $\mathbf{q}^{(h,i)}$ and the i-th column of $\tilde{\mathbf{p}}^{(h-1)}$, i.e.,

$$\boldsymbol{p}_{(\cdot,i)}^{(h)} = \mathbf{q}^{(h,i)} \tilde{\boldsymbol{p}}_{(\cdot,i)}^{(h-1)} \Delta v; \tag{4.7}$$

(8) Assemble into the PDF matrix $\mathbf{p}^{(h)}$;

(9) Normalization: the PDF matrix $\mathbf{p}^{(h)}$ should satisfy

$$\left|\sum_{i=0}^{n_x}\sum_{j=0}^{n_v} p_{ji}^{(h)}\Delta x\,\Delta v - 1\right| \leqslant \epsilon_3, \tag{4.8}$$

where ϵ_3 is the allowable error;

Output: $\mathbf{p}^{(h)} = [p_{ji}^{(h)}]_{(n_v+1)\times(n_x+1)}$ as a numerical discrete matrix of $p_{X_t V_t}(x, v, t_h)$ under the meshing (4.1).

The corresponding theoretical elaboration of Algorithm 4.1 can be found in Chen and Lyu (2022). The numerical procedure sketch of Algorithm 4.1 from instant t_{h-1} to t_h is shown as Fig. 4.1.

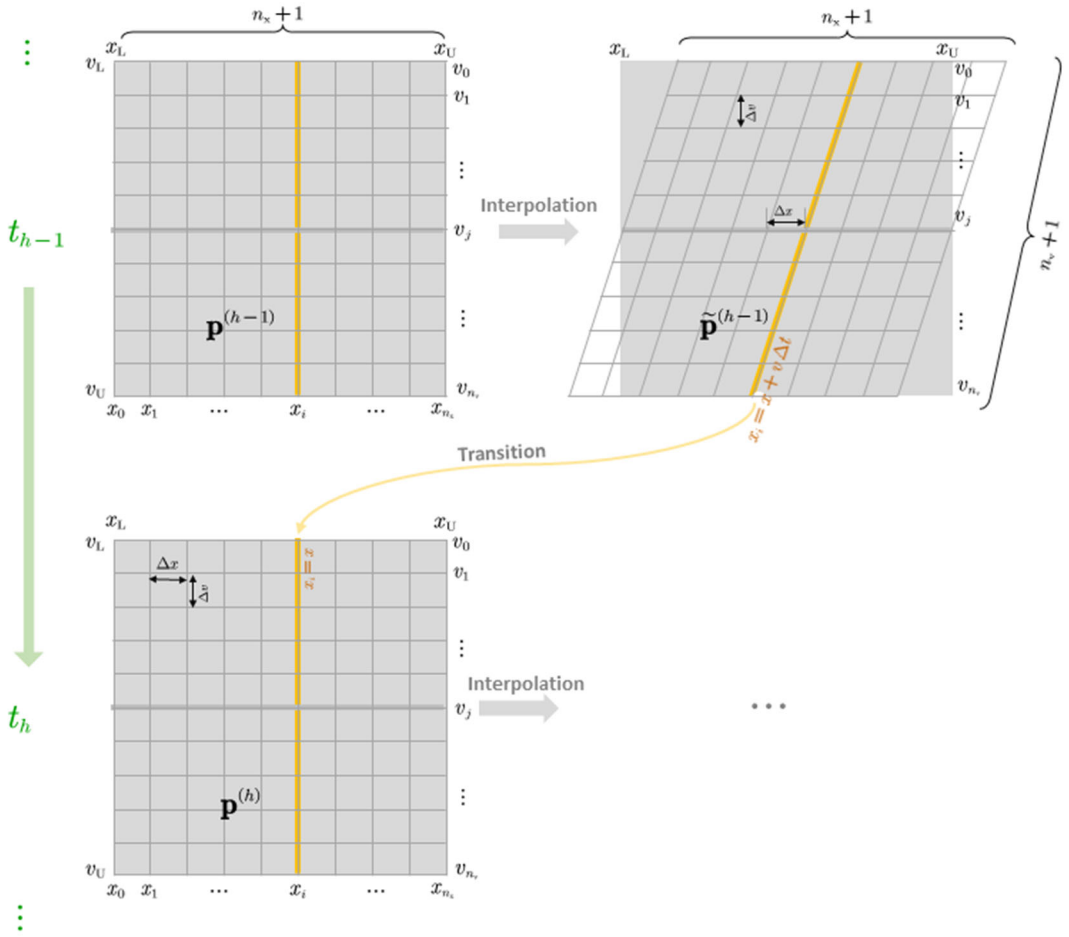


Fig. 4.1. Numerical procedure sketch of Algorithm 4.1.

Algorithm 4.1 can be implemented by the MatLab function `f_GEGDEE_Gwn`. The syntax of the function is

```
Ptemp = f_GEGDEE_Gwn (P, x, v, Aeff, sigma, dt)
```

that returns the values of the PDF evaluated at the values in `x` and `v` at the next step `dt` according to the current PDF `P` with the identified drift `Aeff` and deterministic diffusion `sigma`.

There are six input arguments for the function:

(1) `P`, an $(n_{\mathrm{v}}+1)\times(n_{\mathrm{x}}+1)$ matrix, the joint PDF of $X_l(t_{h-1})$ and $V_l(t_{h-1})$;

(2) `x`, an $n_{\mathrm{x}}+1$-dimensional column vector, the grid sequence of state quantity x given by Eq. (4.1);

(3) `v`, an $n_{\mathrm{v}}+1$-dimensional column vector, the grid sequence of state quantity v given by Eq. (4.1);

(4) `Aeff`, an $(n_{\mathrm{v}}+1)\times(n_{\mathrm{x}}+1)$ matrix, the intrinsic drift function at instant t_{h-1} evaluated via Algorithm 3.1;

(5) `sigma`, a scalar, the diffusion function σ_{ll} in Eq. (2.23);

(6) `dt`, a scalar, the time step.

There is one output argument of the function:

`Ptemp`, an $(n_{\mathrm{v}}+1)\times(n_{\mathrm{x}}+1)$ matrix, the joint PDF of $X_l(t_h)$ and $V_l(t_h)$.

5. Numerical Examples

5.1. Multi-story nonlinear hysteretic frame structure subjected to Gaussian white noise

Consider an m-story two-span hysteretic frame structure subjected to Gaussian white noise. Its equation of motion can be written as

$$\mathbf{m}\ddot{\boldsymbol{X}}(t)+\mathbf{c}\dot{\boldsymbol{X}}(t)+\boldsymbol{f}[\boldsymbol{X}(t),\boldsymbol{Z}(t)]=-\mathbf{m}\mathbf{1}_m\xi(t), \tag{5.1}$$

where $\boldsymbol{X}(t)$, $\dot{\boldsymbol{X}}(t)$, and $\ddot{\boldsymbol{X}}(t)$ are the m-dimensional displacement, velocity, and acceleration process vector; $\mathbf{m}=\mathrm{diag}\{m_1,\ldots,m_m\}$ is the $m\times m$ lumped mass matrix; $\mathbf{c}=\alpha_{\mathrm{m}}\mathbf{m}+\alpha_{\mathrm{k}}\mathbf{k}$ is the $m\times m$ Rayleigh damping matrix;

$$\mathbf{k}=\begin{pmatrix} k_1^*+k_2^* & -k_2^* & & \\ -k_2^* & \ddots & \ddots & \\ & \ddots & k_{m-1}^*+k_m^* & -k_m^* \\ & & -k_m^* & k_m^* \end{pmatrix}$$

is the $m\times m$ initial stiffness matrix; α_{m} and α_{k} are the Rayleigh damping coefficients (Clough & Penzien 2003); k_j^* is the j-th initial inter-story stiffness, for $j=1,\ldots,m$; $\xi(t)$ is a Gaussian white noise process with the intensity σ_{w}; $\boldsymbol{f}(\cdot)$ is

the m-dimensional restoring force vector characterized by the Bouc-Wen model (Bouc 1971, Wen 1976, Baber & Wen 1981, Baber & Noori 1986), i.e.,

$$f_j^*[\boldsymbol{X}(t),\boldsymbol{Z}(t)]=\alpha k_j^* X_j^*(t)+(1-\alpha)k_j^* Z_j(t),\ \text{for } j=1,...,m, \tag{5.2}$$

and $f_j^*(\cdot)$ is the j-th inter-story restoring force; $X_j^*(t)$ is the j-th inter-story drift; $Z_j(t)$ is the j-th imaginary hysteretic displacement, and described by

$$\begin{aligned}\dot{Z}_j(t)=\frac{1}{1+d_\eta E_j(t)}&\left[1-\zeta_s[1-e^{-p\varepsilon_j(t)}]e^{-\left(\frac{Z_j(t)\operatorname{sgn}[\dot{X}_j^*(t)]-\frac{q}{(\beta+\gamma)[1+d_\nu E_j(t)]}}{[\psi+d_\psi\varepsilon_j(t)]\{\lambda+\zeta_s[1-e^{-pE_j(t)}]\}}\right)^2}\right]\\&\cdot\left\{\dot{X}_j^*(t)-\left[\beta\left|\dot{X}_j^*(t)\right|Z_j(t)+\gamma\dot{X}_j^*(t)\left|Z_j(t)\right|\right][1+d_\nu E_j(t)]\right\},\\&\text{for } j=1,...,m;\end{aligned} \tag{5.3}$$

in which $E_j(t)$ is the j-th hysteretic-dissipated energy, and described by

$$\dot{E}_j(t)=\dot{X}_j^*(t)Z_j(t),\ \text{for } j=1,...,m; \tag{5.4}$$

α is the ratio of linear to nonlinear response; β and γ are the basic hysteresis shape controls; d_ν and d_η are the strength and stiffness degradation, respectively; ζ_s is the measure of total slip; q, p, ψ, d_ψ, and λ are the initiation, slope, magnitude, rate, and severity/rate interaction of pinching, respectively (Ma et al. 2004).

In this example, if the l-th displacement, i.e., $X_l(t)$ for $1\leqslant l\leqslant m$, is the quantity of interest, then the transient PDF of $X_l(t)$ and its velocity $V_l(t)=\dot{X}_l(t)$ can be determined via physically-driven DR-PDEE. The corresponding numerical procedure is coded in MatLab script file `Resp_BW_Gwn.m`. There are nine parts in this script separated by `%%`.

The first two parts are input modules, including input of structural parameters and numerical parameters, respectively. The third and fourth parts are the pretreatment for input parameters, i.e., calculation of structural property and treatment of numerical parameters, respectively. The fifth to seventh parts are the main implement procedures of physically-driven DR-PDEE, including representative deterministic analyses, identification of the intrinsic drift functions, and solution of the DR-PDEE, respectively. The last two parts are post-treatment of the numerical results, including comparison with Monte Carlo simulation (MCS) and generating figures. The numerical procedure of the corresponding MCS is elaborated in Appendix A.1; the function `f_BoucWen` called in the script is elaborated in Appendix B.1.

The first part is the input of structural parameters including the followings:

`dof`, the DOFs of the structure;

`l_itrst`, the DOF number of the quantity of interest;

`str_column`, the number of columns of each story;

`str_m` with the dimension “kg”, a `dof`-dimensional column vector, the lumped mass of each story;

`str_h` with the dimension “m”, a `dof`-dimensional column vector, the height of each story;

`str_a` with the dimension “m”, a `dof`-dimensional column vector, the length of column section of each story;

`str_b` with the dimension “m”, a `dof`-dimensional column vector, the width of column section of each story;

`str_E` with the dimension “Pa”, a `dof`-dimensional column vector, the initial Young's modulus of each story;

`zeta`, a two-dimensional column vector, the damping ratios of the first two modes;

`bwm_alpha`, the ratio of linear to nonlinear response;

`bwm_A`, one of the basic hysteresis shape controls of the Bouc-Wen model;

`bwm_n`, the sharpness of yield of the Bouc-Wen model;

`bwm_beta` with the dimension “m^{-1}”, one of the basic hysteresis shape controls of the Bouc-Wen model;

`bwm_gamma` with the dimension “m^{-1}”, one of the basic hysteresis shape controls of the Bouc-Wen model;

`bwm_d_nu` with the dimension “m^{-2}”, the strength degradation of the Bouc-Wen model;

`bwm_d_eta` with the dimension “m^{-2}”, the stiffness degradation of the Bouc-Wen model;

`bwm_p` with the dimension “m^{-2}”, the pinching slope of the Bouc-Wen model;

`bwm_q`, the pinching initiation of the Bouc-Wen model;

`bwm_d_psi` with the dimension “m^{-2}”, the pinching rate of the Bouc-Wen model;

`bwm_lambda`, the pinching severity/rate interaction of the Bouc-Wen model;

`bwm_zeta_s`, the measure of total slip of the Bouc-Wen model;

`bwm_psi` with the dimension “m”, the pinching magnitude of the Bouc-Wen model;

`D` with the dimension “m^2/s^3”, the intensity of Gaussian white noise;

`x0`, a 4`dof`-dimensional column vector, the initial value of displacement, velocity, hysteretic displacement, and hysteretic-dissipated energy of each story.

The second part is the input of numerical parameters including the followings:

`T` with the dimension "s", the termination time;

`dt` with the dimension "s", the time step;

`Neq`, the number of representative deterministic analyses;

`a_x` with the dimension "m", the boundary of the state quantity x;

`a_v` with the dimension "m/s", the boundary of the state quantity v;

`Nxfit`, the number of grids of state quantity x for identifying the intrinsic drift function;

`Nvfit`, the number of grids of state quantity v for identifying the intrinsic drift function;

`Nxl`, the number of grids of state quantity x for solving the DR-PDEE;

`Nvl`, the number of grids of state quantity v for solving the DR-PDEE;

`Alpha`, the smooth scale for LOWESS taken in (0,1].

The results of running the script file are saved into the matrix file `Resp_BW_Gwn.mat`, in which the output quantities contain the following quantities:

`PDF_xl` with the dimension "m^{-1}", an (`Nxl` + 1) × (`Nt` + 1) matrix, the time-variant PDF of $X_l(t)$;

`PDF_vl` with the dimension "s/m", an (`Nvl` + 1) × (`Nt` + 1) matrix, the time-variant PDF of $V_l(t)$;

`xl` with the dimension "m", an (`Nxl` + 1) -dimensional row vector, the corresponding grid sequence of state quantity x to `PDF_xl`;

`vl` with the dimension "m/s", an (`Nvl` + 1) -dimensional row vector, the corresponding grid sequence of state quantity v to `PDF_vl`;

`varX` with the dimension "m^2", an (`Nt` + 1) -dimensional row vector, the variance history of $X_l(t)$;

`varV` with the dimension "m^2/s^2", an (`Nt` + 1) -dimensional row vector, the variance history of $V_l(t)$;

`kurtX`, an (`Nt` + 1) -dimensional row vector, the kurtosis history of $X_l(t)$;

`kurtV`, an (`Nt` + 1) -dimensional row vector, the kurtosis history of $V_l(t)$;

`Aeff` with the dimension "m/s^2", an (`Nvfit` + 1) × (`Nxfit` + 1) × `Nt` array, the intrinsic drift function estimation;

`XXfit` with the dimension "m", an (`Nvfit` + 1) × (`Nxfit` + 1) matrix, the corresponding grid points of state quantity x to `Aeff`;

`VVfit` with the dimension "m/s", an (`Nvfit` + 1) × (`Nxfit` + 1) matrix, the corresponding grid points of state quantity v to `Aeff`;

`Xl` with the dimension "m", an `Neq` × `Nt` matrix, the data history of $X_l(t)$;

`Vl` with the dimension "m/s", an `Neq` × `Nt` matrix, the data history of $V_l(t)$;

`fl` with the dimension "m/s^2", an `Neq` × `Nt` matrix, the data history of $A_l(t)$;

`t` with the dimension "s", an (`Nt` + 1) -dimensional row vector, the time series;

`dt` with the dimension "s", the time step;

`dxl` with the dimension "m", the corresponding grid size of state quantity x to `PDF_xl`;

`dvl` with the dimension "m/s", the corresponding grid size of state quantity v to `PDF_vl`.

The figures generated by running the script file are shown in Figs. 5.1 to Fig. 5.3. In this analysis, the DOFs of the structure is taken as $m=10$; the DOF number of the quantity of interest $l=10$; other parameter values follow the initial settings of the script file as well. Namely, the probabilistic response determination of the top displacement and velocity for a ten-story nonlinear hysteretic frame structure subjected to Gaussian white noise is shown as follows.

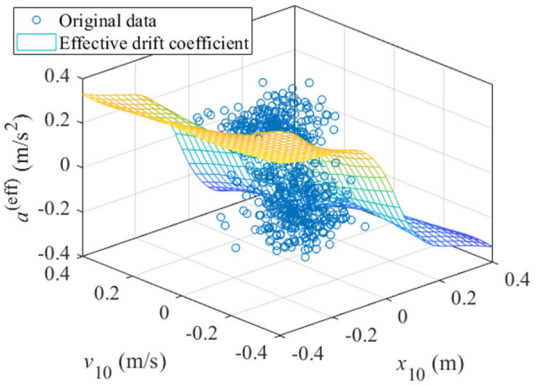
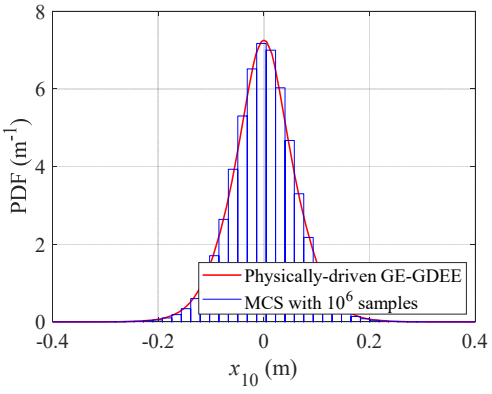
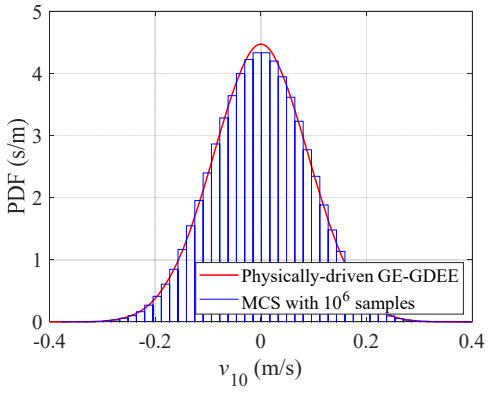


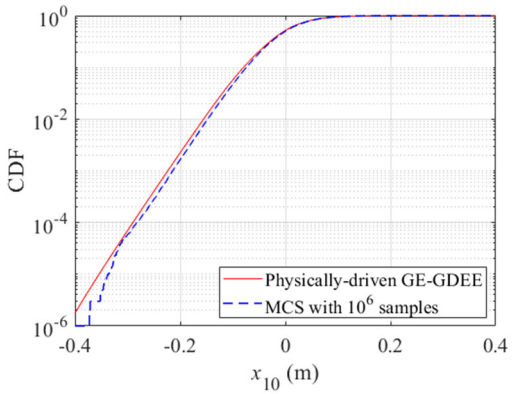
Fig. 5.1. Intrinsic drift function estimation at $t=10$ s.



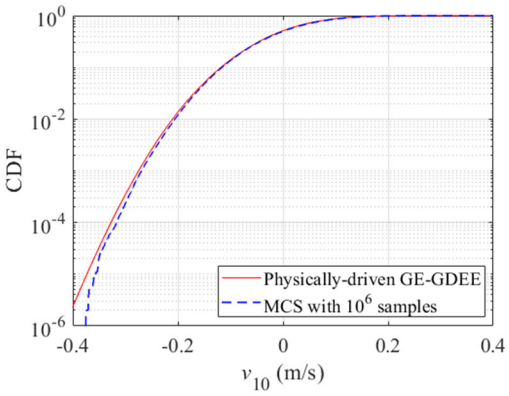
(a) PDF of $X_{10}(t)$;



(b) PDF of $V_{10}(t)$;

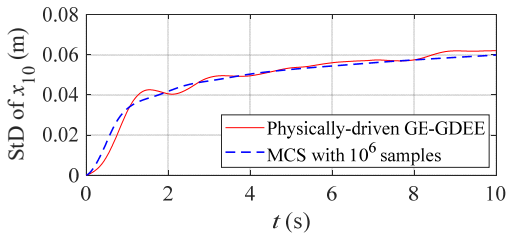


(c) CDF of $X_{10}(t)$;

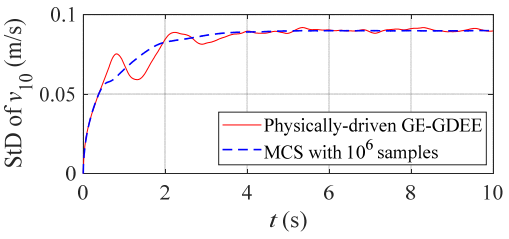


(d) CDF of $V_{10}(t)$.

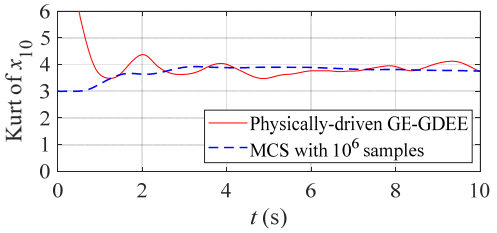
Fig. 5.2. PDFs and CDFs of $X_{10}(t)$ and $V_{10}(t)$ at $t=10$ s.



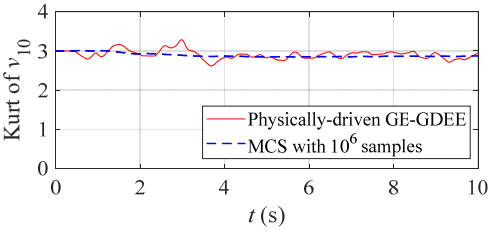
(a) Standard deviation of $X_{10}(t)$;



(b) Standard deviation of $V_{10}(t)$;



(c) Kurtosis of $X_{10}(t)$;



(d) Kurtosis of $V_{10}(t)$.

Fig. 5.3. Standard deviations and kurtoses of $X_{10}(t)$ and $V_{10}(t)$.

Appendix A. Description of other MatLab script files

A.1. MCS verification for Example 5.1

The numerical procedure of MCS verification for Example 5.1 is coded in MatLab script file `MCS_BW_Gwn.m`. There are seven parts in this script separated by `%%`.

The first two parts are the input modules, including the input of structural parameters and numerical parameters, respectively. The third and fourth parts are the pretreatment for input parameters, i.e., calculation of structural property and treatment of numerical parameters, respectively. The fifth part is the main implement procedures of MCS. The last two parts are post-treatment of the numerical results and generating figures. The function `f_BoucWen` called in the script is elaborated in Appendix B.1.

The first part is the input of structural parameters including the followings:

`dof`, the DOFs of the structure;

`l_itrst`, the DOF number of the quantity of interest;

`str_column`, the number of column of each story;

`str_m` with the dimension "kg", a `dof`-dimensional column vector, the lumped mass of each story;

`str_h` with the dimension "m", a `dof`-dimensional column vector, the height of each story;

`str_a` with the dimension "m", a `dof`-dimensional column vector, the length of column section of each story;

`str_b` with the dimension "m", a `dof`-dimensional column vector, the width of column section of each story;

`str_E` with the dimension "Pa", a `dof`-dimensional column vector, the initial Young's modulus of each story;

`zeta`, a two-dimensional column vector, the damping ratios of the first two modes;

`bwm_alpha`, the ratio of linear to nonlinear response;

`bwm_A`, one of the basic hysteresis shape controls of the Bouc-Wen model;

`bwm_n`, the sharpness of yield of the Bouc-Wen model;

`bwm_beta` with the dimension "m^{-1}", one of the basic hysteresis shape controls of the Bouc-Wen model;

`bwm_gamma` with the dimension "m^{-1}", one of the basic hysteresis shape controls of the Bouc-Wen model;

`bwm_d_nu` with the dimension "m^{-2}", the strength degradation of the Bouc-Wen model;

`bwm_d_eta` with the dimension "m^{-2}", the stiffness degradation of the Bouc-Wen model;

`bwm_p` with the dimension "m^{-2}", the pinching slope of the Bouc-Wen model;

`bwm_q`, the pinching initiation of the Bouc-Wen model;

`bwm_d_psi` with the dimension "m^{-2}", the pinching rate of the Bouc-Wen model;

`bwm_lambda`, the pinching severity/rate interaction of the Bouc-Wen model;

`bwm_zeta_s`, the measure of total slip of the Bouc-Wen model;

`bwm_psi` with the dimension "m", the pinching magnitude of the Bouc-Wen model;

`D` with the dimension "m^2/s^3", the intensity of Gaussian white noise;

`x0`, a $4 \times$ `dof`-dimensional column vector, the initial value of displacement, velocity, hysteretic displacement, and hysteretic-dissipated energy of each story.

The second part is the input of numerical parameters including the followings:

`T` with the dimension "s", the termination time;

`dt` with the dimension "s", the time step;

`N`, the number of MCS;

`Nrec`, the number of sample histories recorded;

`Ttyp` with the dimension "s", an arbitrary dimensional row vector, the typical instant for sample points recorded;

`threshold` with the dimension "m", an arbitrary dimensional row vector, the threshold of the quantity of interest;

`N_bar`, the number of bars for histogram.

The results of running the script file are saved into the matrix file `MCS_BW_Gwn.mat`, in which the output quantities contain followings:

`Xsample` with the dimension "m", an `N` × `length (Ttyp)` matrix, the sample data of $X_l(t)$ at some typical instants;

`Vsample` with the dimension "m/s", an `N` × `length (Ttyp)` matrix, the sample data of $V_l(t)$ at some typical instants;

`varX_mcs` with the dimension "m^2", an (`Nt` + 1) -dimensional row vector, the variance history of $X_l(t)$ evaluated via MCS;

`varV_mcs` with the dimension "m^2/s^2", an (`Nt` + 1) -dimensional row vector, the variance history of $V_l(t)$ evaluated via MCS;

`kurtX_mcs`, an (`Nt` + 1) -dimensional row vector, is the kurtosis history of $X_l(t)$ evaluated via MCS;

`kurtV_mcs`, an (`Nt` + 1) -dimensional row vector, the kurtosis history of $V_l(t)$ evaluated via MCS;

`PDF_xl_mcs` with the dimension "m^{-1}", an `N_bar` × `length (Ttyp)` matrix, the transient PDF of $X_l(t)$ at some typical instants evaluated via MCS;

`PDF_vl_mcs` with the dimension "s/m", an `N_bar` × `length (Ttyp)` matrix, the transient PDF of $V_l(t)$ at some typical instants evaluated via MCS;

`Pf_mcs`, an `length (threshold)` × (`Nt` + 1) matrix, the time-variant probability of failure under some specific thresholds evaluated via MCS;

`x_bar` with the dimension "m", an (`N_bar` + 1) × `length (Ttyp)` matrix, the corresponding grid sequence of state quantity x to `PDF_xl_mcs` at some typical instants;

`v_bar` with the dimension "m/s", an (`N_bar` + 1) × `length (Ttyp)` matrix, the corresponding grid sequence of state quantity v to `PDF_vl_mcs` at some typical instants;

`Ttyp` with the dimension "s", an arbitrary dimensional row vector, the typical instant for sample points recorded;

`threshold` with the dimension "m", an arbitrary dimensional row vector, the threshold of the quantity of interest;

`N`, the number of MCS;

`N_bar`, the number of bars for histogram.

Appendix B. Description of other MatLab functions

B.1. Function f_BoucWen

The MatLab function `f_BoucWen` is to calculate the averaging derivatives of responses vector, i.e., $\left(\dot{\boldsymbol{X}}^{\mathrm{T}}(t),\dot{\boldsymbol{V}}^{\mathrm{T}}(t),\dot{\boldsymbol{Z}}^{\mathrm{T}}(t),\dot{\boldsymbol{E}}^{\mathrm{T}}(t)\right)^{\mathrm{T}}$, according to the values of the responses $\left(\boldsymbol{X}^{\mathrm{T}}(t),\boldsymbol{V}^{\mathrm{T}}(t),\boldsymbol{Z}^{\mathrm{T}}(t),\boldsymbol{E}^{\mathrm{T}}(t)\right)^{\mathrm{T}}$ at current instant for the m-DOF Bouc-Wen system governed by Eq. (5.1). The n samples can be calculated simultaneously by calling the function. The syntax of the function is

```
Xdot = f_BoucWen (X, M, C, k, bwm)
```

that returns the values of the averaging derivatives of `X` evaluated with the mass `M`, damping `C`, stiffness `k`, and Bouc-Wen parameters `bwm`.

There are five input arguments for the function:

(1) `X`, an $4m\times n$ matrix, the value of the response samples at current instant, including displacement, velocity, hysteretic displacement, and hysteretic-dissipated energy;

(2) `M`, an $m\times m$ matrix, the lumped mass;

(3) `C`, an $m \times m$ matrix, the damping;

(4) `k`, an m-dimensional column vector, the initial lateral inter-story stiffness;

(5) `bwm`, a thirteen-dimensional column vector, the Bouc-Wen parameters.

There is one output argument of the function:

`Xdot`, an $4m \times n$ matrix, the value of averaging derivatives of the response samples at current instant, including displacement, velocity, hysteretic displacement, and hysteretic-dissipated energy.

Appendix C. Julia implementation of the algorithms

In addition to the MatLab Toolbox described above, a Julia implementation of the algorithms is also provided. The Julia implementation follows the same framework as the MatLab Toolbox, with only a few differences in detail. Therefore, we give only a basic guideline for the Julia implementation here. Detailed explanations can be found in the comments in the source code.

The Julia implementation includes the following files: `DRPDEE.jl`, `typesdef.jl`, `gridknots.jl`, `srk2.jl`, `boucwen.jl`, `idclowess.jl`, `idcfit.jl`, `pathintegral.jl`, `gegdeesolver.jl`, `gegdeepostproc.jl`, and `gegdee_BWresp_Gwn.jl`.

In the file `DRPDEE.jl`, a module for solving the DR-PDEE is defined. By using command `using .DRPDEE` or `import .DRPDEE`, all necessary functions will be loaded. In the file `typesdef.jl`, some composite types are defined for other functions. The function `gridknots` provided in the file `gridknots.jl` can be employed to generate mesh grids inside a hyper-rectangular domain. The file `srk2.jl` defines a function `skr2!` for solving the equation of motion under stochastic excitations using a 2nd-order stochastic Runge-Kutta scheme. The symbol "!" in a function name indicates that the function modifies the values of its inputs, which is a convention in Julia programming. The function `boucwen,` defined in the file `boucwen.jl` calculates the right-hand term of the differential equation that governs the Bouc-Wen model. In the file `idclowess.jl`, the users can find a function named `idclowess` that implements the LOWESS algorithm. The file `idcfit.jl` defines a function, `idcfit!`, which identifies the intrinsic drift functions by calling the `idclowess` function at each time instant. The function `pathintegral!`, provided in the file `pathintegral.jl`, first interpolates the values of the intrinsic drift functions from a coarse mesh into a fine mesh, and then updates the PDF in the next time step using path

integral approach. The function `gegdeesolver!`, defined in the file `gegdeesolver.jl`, solves the DR-PDEE by calling the `pathintegral!` function at each time instant. Finally, the file `gegdeepostproc.jl` presents several functions for computing the cumulative distribution function (CDF), the mean value and the variance. These functions are named `cdfgegdee`, `meangegdee` and `vargegdee`, respectively. The manuscript `gegdee_BWresp_Gwn.jl` provides an example of solving the DR-PDEE using the module, considering the same case as in Section 5.

Since the MatLab Toolbox has been introduced in details, we omit the tedious explanations of the inputs and outputs for all the Julia functions here. Users can find the necessary information by reading the source code of the functions and the manuscript.

Acknowledgements

Financial supports from the National Natural Science Foundation of China (the National Distinguished Youth Fund of NSFC with Grant No. 51725804 and the NSFC-Guangdong Province Joint Project Grant No. U1711264), the Fund for State Key Laboratories from Ministry of Science and Technology of China (SLDRCE19-B-23), China Postdoctoral Science Foundation (2023M732669), and Shanghai Post-doctoral Excellence Program (2022558) are highly appreciated.

Declaration

References

[1] Baber TT, Wen YK, 1981. Random vibrations of hysteretic degrading systems [J]. ASCE-Journal of Engineering Mechanics, 107 (EM6): 1069-1089.

[2] Baber TT, Noori MN, 1986. Modeling general hysteresis behaviour and random vibration applications [J]. Journal of Vibration, Acoustics, Stress & Reliability in Design, 108 (4): 411-420.

[3] Bouc R, 1971. Modèle mathématique d'hystérésis: Application aux systèmesàun degréde liberté [J]. Acustica, 24 (1): 16-25 (in French).

[4] Chen JB, Rui ZM, 2018. Dimension-reduced FPK equation for additive white-noise excited nonlinear structures [J]. Probabilistic Engineering Mechanics, 53: 1-13.

[5] Chen JB, Lyu MZ, 2022. Globally-evolving-based generalized density evolution equation for nonlinear systems involving randomness from both system parameters and excitations [J]. Proceedings of the Royal Society A - Mathematical Physical & Engineering Sciences, 478 (2264): 20220356.

[6] Cleveland WS, 1979. Robust locally weighted regression and smoothing scatterplots [J]. Journal of the American Statistical Association, 74 (368): 829-836.

[7] Clough R, Penzien J, 2003. Dynamics of Structures [M]. 3rd Edn. Computers & Structures, Inc., Berkeley, USA.

[8] Gardiner CW, 2004. Handbook of Stochastic Methods for Physics, Chemistry, and the Natural Sciences [M]. 3rd Edn. Springer-Verlag, Berlin, Germany.

[9] Luo Y, Chen JB, Spanos PD, 2022a. Determination of monopile offshore structure response to stochastic wave loads via analog filter approximation and GV-GDEE procedure [J]. Probabilistic Engineering Mechanics, 67: 103197.

[10] Luo Y, Spanos PD, Chen JB, 2022b. Stochastic response determination of multi-dimensional nonlinear systems endowed with fractional derivative elements by the DR-PDEE [J]. International Journal of Non-Linear Mechanics, 147: 104247.

[11] Luo Y, Lyu MZ, Chen JB, Spanos PD, 2023. Exact low-dimensional partial differential equation governing the probability density evolution of multi-dimensional linear fractional differential systems enforced by Gaussian white noise [J]. Theoretical & Applied Mechanics Letters, 13: 100436.

[12] Lyu MZ, Chen JB, 2021. First-passage reliability of high-dimensional nonlinear systems under additive excitation by the ensemble-evolving-based generalized density evolution equation [J]. Probabilistic Engineering Mechanics, 63: 103119.

[13] Lyu MZ, Chen JB, 2022. A unified formalism of the DR-PDEE for generic continuous responses and first-passage reliability analysis of multi-dimensional nonlinear systems subjected to non-white-noise excitations [J]. Structural Safety, 98: 102233.

[14] Lyu MZ, Chen JB, 2023. DR-PDEE for reliability analysis of high-dimensional nonlinear systems enforced by non-stationary stochastic excitations [J]. Journal of Vibration Engineering (in Chinese) (online).

[15] Lyu MZ, Chen JB, Shen JX, 2023. Refined probabilistic response and seismic reliability evaluation of high-rise reinforced concrete structures via physically-driven DR-PDEE [J]. Acta Mechanica (online).

[16] Ma F, Zhang H, Bockstedte A, Foliente GC, Paevere P, 2004. Parameter analysis of the differential model of hysteresis [J]. Journal of Engineering Mechanics, 71: 342-349.

[17] Naess A, Johnsen JM, 1993. Response statistics of nonlinear, compliant offshore structures by the path integral solution method [J]. Probabilistic Engineering Mechanics, 8 (2): 91-106.

[18] Redner S, 2001. A Guide to First-Passage Processes [M]. Cambridge University Press, Cambridge, UK.

[19] Spanos PD, Mignolet MP, 1989. ARMA Monte Carlo simulation in probabilistic structural analysis [J]. The Shock & Vibration Digest, 21 (11): 3-14.

[20] Sun TT, Chen JB, 2022. Physically driven exact dimension-reduction of a class of nonlinear multi-dimensional systems subjected to additive white noise [J]. ASCE-ASME Journal of Risk & Uncertainty in Engineering Systems, Part A - Civil Engineering, 8 (2): 04022012.

[21] Sun TT, Lyu MZ, Chen JB, 2023. Property of intrinsic drift coefficients in globally-evolving-based generalized density evolution equation for the first-passage reliability assessment [J]. Acta Mechanica Sinica, 39: 722471.

[22] Wehner MF, Wolfer WG, 1983. Numerical evaluation of path-integral solutions to Fokker-Planck equations [J]. Physical Review A, 27 (5): 2663-2670.

[23] Wen YK, 1976. Method for random vibration of hysteretic systems [J]. Journal of Engineering Mechanics, 102: 249-263.