Neural network representation of the probability density function of diffusion processes

Wayne Isaac Tan Uy[a,b], Mircea Grigoriu[a]
[a]Cornell University Center for Applied Math
[b]Courant Institute of Mathematical Sciences

Diffusion processes: Introduction and Motivation

Stochastic differential equation for diffusion processes:

$$d\boldsymbol{X}(t) = \boldsymbol{a}\big(\boldsymbol{X}(t-)\big)\,dt + \boldsymbol{b}\big(\boldsymbol{X}(t-)\big)\,d\boldsymbol{B}(t) + \boldsymbol{c}\big(\boldsymbol{X}(t-)\big)\,d\boldsymbol{C}(t),\;\; \boldsymbol{X}(t) \in \mathbb{R}^d, t \in [0,T]$$

- $\boldsymbol{B}$ is a vector of m_B independent standard Brownian motions
- $\boldsymbol{C}$ is a vector of m_C independent compound Poison processes $C_r(t)$
  - $C_r(t) = \sum_{\nu=1}^{N_r(t)} Y_{r,\nu}$ where $Y_{r,\nu}$ are iid jump sizes with cdf F_r
  - $\{N_r\}$ are homogeneous Poisson processes with intensities $\{\lambda_r\}$

Objectives:

- Compute the probability density function (pdf) $f(\boldsymbol{x},t)$ of $\boldsymbol{X}(t)$
- Solve the PDE for the pdf (Fokker-Planck) and the characteristic function (chf)
- Numerical methods, e.g. FD/FEM, for Fokker-Planck equation can be unstable for large dimensions

Physics-informed neural networks (PINNs):

- Investigate the use of PINNs to represent the pdf and chf
- Examine theoretically and numerically which PDE to solve
- Utilize prior information on the diffusion process to design the network architecture

PDE for the characteristic function

By stochastic analysis, the chf $\varphi(\boldsymbol{u},t), \boldsymbol{u} \in \mathbb{R}^d$ satisfies

$$\begin{aligned}\frac{\partial \varphi(\boldsymbol{u},t)}{\partial t} &= i\sum_{k=1}^{d} u_k\, E\Big[e^{i\,\boldsymbol{u}'\boldsymbol{X}(t-)}\, a_k\big(\boldsymbol{X}(t-)\big)\Big] \\ &- \frac{1}{2}\sum_{k,l=1}^{d} u_k\, u_l\, E\Big[\exp\big(i\,\boldsymbol{u}'\boldsymbol{X}(t-)\big)\sum_{w=1}^{m_B} b_{kw}\big(\boldsymbol{X}(t-)\big)\, b_{lw}\big(\boldsymbol{X}(t-)\big)\Big] \\ &+ \sum_{r=1}^{m_c}\lambda_r\, E\Big[\int_{\mathbb{R}} e^{i\,\boldsymbol{u}'\left(\boldsymbol{X}(t-)+c^{(r)}(\boldsymbol{X}(t-))\,y\right)}\, dF_r(y) - e^{i\,\boldsymbol{u}'\boldsymbol{X}(t-)}\Big],\\ \varphi(\boldsymbol{0},t) &= 1, \quad \forall t, \boldsymbol{0} \in \mathbb{R}^d.\end{aligned}$$

- $a_k = [\boldsymbol{a}]_k$, $b_{ij} = [\boldsymbol{b}]_{ij}$, $c^{(r)}$ is r-th column of $\boldsymbol{c}$
- Becomes a PDE for $\varphi(\boldsymbol{u},t)$ if $\boldsymbol{a}, \boldsymbol{b}$ are polynomials, $\boldsymbol{c}$ has special structure
- This is generally a complex-valued integro-differential equation

PDE for the probability density function (Fokker-Planck equation)

If Poisson white noise is absent: $f(\boldsymbol{x},t)$ satisfies

$$\frac{\partial f(\boldsymbol{x},t)}{\partial t} = -\sum_{i=1}^{d}\frac{\partial}{\partial x_i}[a_i(\boldsymbol{x})\, f(\boldsymbol{x},t)] + \frac{1}{2}\sum_{i,j=1}^{d}\frac{\partial^2}{\partial x_i \partial x_j}\left[(\boldsymbol{b}(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x})')_{ij}\, f(\boldsymbol{x},t)\right]$$

$$\int_{\mathbb{R}^d} f(\boldsymbol{x},t)\, d\boldsymbol{x} = 1,\;\; t \in [0,T].$$

- This is a parabolic PDE with maximum order of partial derivatives at most 2

If Poisson white noise is present: a generalized Fokker-Planck equation may not exist

- The PDE for the chf may be more preferable to solve. Consider

$$dX(t) = -\rho X(t-)\,dt + dC(t), \;\; t \geq 0.$$

  - $\rho > 0$, $C(t) = \sum_{w=1}^{N(t)} Y_w$, $N(t)$ is a Poisson process with intensity λ, Y_w are iid
  - PDE for $\varphi(u,t)$ is $\frac{\partial \varphi(u,t)}{\partial t} = -\rho u \frac{\partial \varphi(u,t)}{\partial u} + \lambda\left(E[e^{iuY}] - 1\right)\varphi(u,t)$
  - PDE for $f(x,t)$ is $\frac{\partial f(x,t)}{\partial t} = \rho\frac{\partial}{\partial x}(x f(x,t)) + \lambda\sum_{k=1}^{\infty}\frac{(-1)^k E[Y^k]}{k!}\frac{\partial^k f(x,t)}{\partial x^k}$

Physics-informed neural network representation of the pdf and chf

Training neural networks to approximate the chf

- Represent $\varphi(\boldsymbol{u},t)$ by a neural network $\widetilde{\varphi}(\boldsymbol{u},t)$ with 2-dimensional output
- Prior information on the SDE (i.e. symmetry) can be utilized to show that $\varphi(\boldsymbol{u},t)$ is real-valued

Algorithm:

- Truncate the frequency domain to $D \subset \mathbb{R}^d$
- Select N_{Op} collocation points $\{(\boldsymbol{u}_i^{Op}, t_i^{Op})\}_{i=1}^{N_{Op}} \subset D \times [0,T]$ to enforce the governing equations
- Select N_{IC} collocation points $\{(\boldsymbol{u}_i^{IC}, 0)\}_{i=1}^{N_{IC}} \subset D \times 0$ to enforce the initial condition
- Select N_0 collocation points $\{(\boldsymbol{0}, t_i^0)\}_{i=1}^{N_0} \subset \boldsymbol{0} \times [0,T]$ to enforce the condition at the origin
- Solve for the neural network parameters to minimize the loss

$$\mathcal{L} = \frac{1}{N_{Op}}\sum_{i=1}^{N_{Op}}\left|\mathcal{Q}[\widetilde{\varphi}(\boldsymbol{u}_i^{Op}, t_i^{Op})]\right|^2 + \frac{1}{N_{IC}}\sum_{i=1}^{N_{IC}}\left|\widetilde{\varphi}(\boldsymbol{u}_i^{IC},0) - \varphi(\boldsymbol{u}_i^{IC},0)\right|^2 + \frac{1}{N_0}\sum_{i=1}^{N_0}\left|\widetilde{\varphi}(\boldsymbol{0},t_i^0) - 1\right|^2$$

  where $\mathcal{Q}$ is the PDE satisfied by $\varphi(\boldsymbol{u},t)$
- Compute the NN approximation $\widetilde{f}(\boldsymbol{x},t)$ of $f(\boldsymbol{x},t)$ via $\widetilde{f}(\boldsymbol{x},t) = \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} e^{-i\boldsymbol{u}'\boldsymbol{x}}\,\widetilde{\varphi}(\boldsymbol{u},t)\, d\boldsymbol{u}$

Training neural networks to approximate the pdf

- Express $f(\boldsymbol{x},t) = \frac{e^{-v(\boldsymbol{x},t)}}{\int_{\mathbb{R}^d} e^{-v(\boldsymbol{x},t)}\, d\boldsymbol{x}}$ to satisfy the normalization constraint
- Represent $v(\boldsymbol{x},t)$ by a neural network $\widetilde{v}(\boldsymbol{x},t)$
- The PDE for $v(\boldsymbol{x},t)$ is an integro-differential equation with no unique solution

Algorithm:

- Truncate the spatial domain to $D \subset \mathbb{R}^d$
- Select N_{Op} collocation points $\{(\boldsymbol{x}_i^{Op}, t_i^{Op})\}_{i=1}^{N_{Op}} \subset D \times [0,T]$ to enforce the governing equations
- Select N_{IC} collocation points $\{(\boldsymbol{x}_i^{IC}, 0)\}_{i=1}^{N_{IC}} \subset D \times 0$ to enforce the initial condition
- Solve for the neural network parameters to minimize the loss

$$\mathcal{L} = \frac{1}{N_{Op}}\sum_{i=1}^{N_{Op}}(\mathcal{M}[\widetilde{v}(\boldsymbol{x}_i^{Op}, t_i^{Op})])^2 + \frac{1}{N_{IC}}\sum_{i=1}^{N_{IC}}(\widetilde{v}(\boldsymbol{x}_i^{IC},0) - v(\boldsymbol{x}_i^{IC},0))^2$$

  where $\mathcal{M}$ is the PDE satisfied by $v(\boldsymbol{x},t)$
- Compute the NN approximation $\widetilde{f}(\boldsymbol{x},t) = \frac{e^{-\widetilde{v}(\boldsymbol{x},t)}}{\int_{\mathbb{R}^d} e^{-\widetilde{v}(\boldsymbol{x},t)}\, d\boldsymbol{x}}$ of $f(\boldsymbol{x},t)$

Applications

Duffing oscillator with Gaussian white noise

$$d\begin{bmatrix}X_1(t)\\X_2(t)\end{bmatrix} = \begin{bmatrix}X_2(t)\\ -\nu^2(X_1(t) + \alpha X_1(t)^3) - 2\zeta\nu X_2(t)\end{bmatrix} dt + \begin{bmatrix}0\\ \sqrt{\pi g_0}\end{bmatrix} dB(t)$$

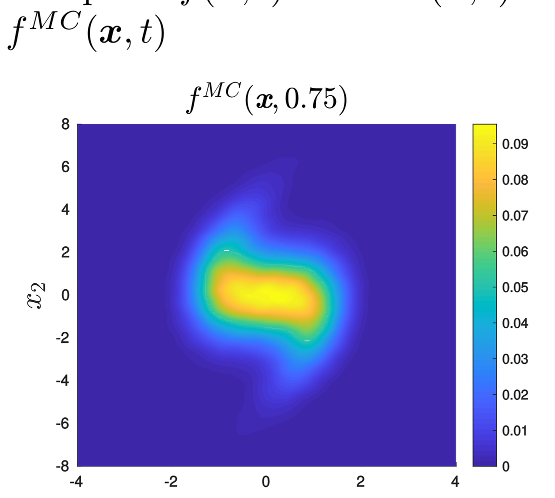
- $\zeta = 0.25, \nu = 1, \alpha = 1, g_0 = 1$,
  $X_1(0) \sim N(0,1), X_2(0) \sim N(0,1), \mathrm{Corr}(X_1(0), X_2(0)) = 0.8$

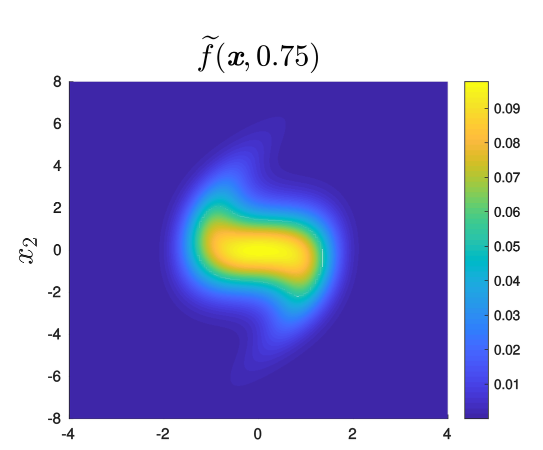
Fokker-Planck equation (PDE for $v(\boldsymbol{x},t)$)

$$\mathcal{M}[v(\boldsymbol{x},t)] = v_t(\boldsymbol{x},t) + x_2 v_{x_1}(\boldsymbol{x},t) + 2\zeta\nu - (\nu^2(x_1 + \alpha x_1^3) + 2\zeta\nu x_2)v_{x_2}(\boldsymbol{x},t) + \frac{\pi g_0}{2}((v_{x_2}(\boldsymbol{x},t))^2 - v_{x_2x_2}(\boldsymbol{x},t)) + \frac{c'(t)}{c(t)} = 0$$

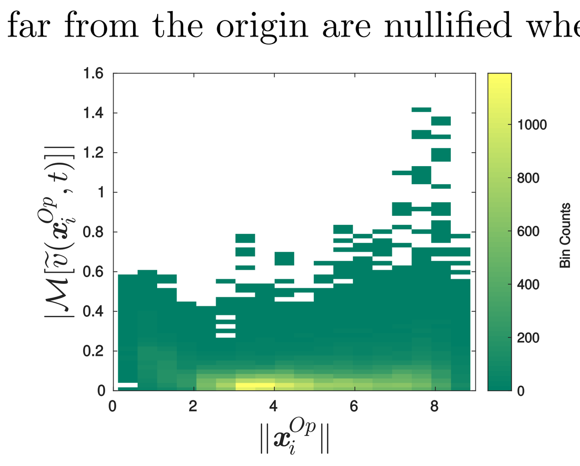
where $c(t) = \int_{\mathbb{R}}\int_{\mathbb{R}} e^{-v(x_1,x_2,t)}\, dx_1\, dx_2$

- Truncated domain is $(x_1,x_2) \in [-4,4] \times [-8,8]$, $t \in [0,1]$
- Represent $\widetilde{v}(\boldsymbol{x},t)$ as a feedforward network with 3 inputs, 1 output, 6 hidden layers with 50 neurons each
- Value of loss function at training collocation points is 0.013158
- Compute $\widetilde{f}(\boldsymbol{x},t)$ from $\widetilde{v}(\boldsymbol{x},t)$ and compare with Monte Carlo estimate $f^{MC}(\boldsymbol{x},t)$





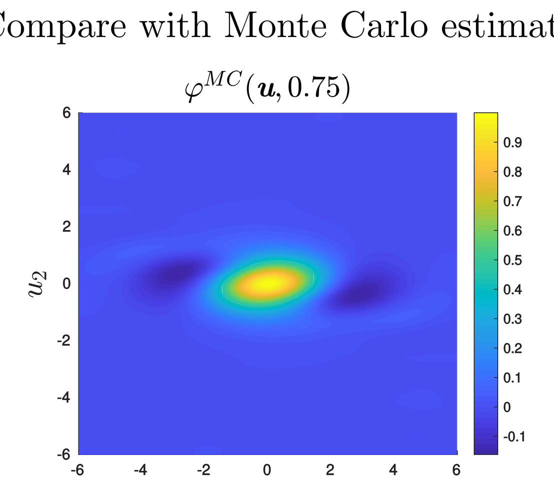
- The neural network approximates the pdf well despite the large loss
- The loss $|\mathcal{M}[\widetilde{v}(\boldsymbol{x}_i^{Op}, t)]|$ is large for large $\|\boldsymbol{x}_i^{Op}\|$ where probability mass is small
- The large errors far from the origin are nullified when $\widetilde{v}(\boldsymbol{x},t)$ is normalized to obtain $\widetilde{f}(\boldsymbol{x},t)$

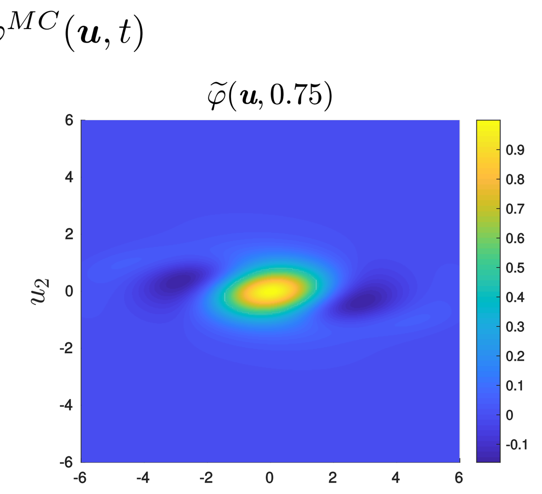


PDE for the characteristic function

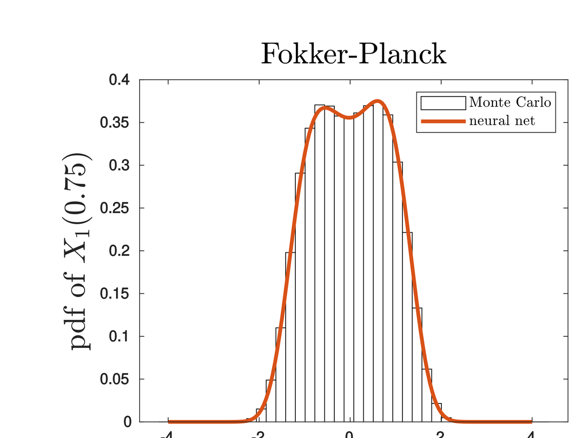
$$\mathcal{Q}[\varphi(\boldsymbol{u},t)] = \frac{\partial \varphi(\boldsymbol{u},t)}{\partial t} - (u_1 + 2\zeta\nu u_2)\frac{\partial \varphi(\boldsymbol{u},t)}{\partial u_2} - \nu^2 u_2\frac{\partial \varphi(\boldsymbol{u},t)}{\partial u_1} + \nu^2\alpha u_2\frac{\partial^3 \varphi(\boldsymbol{u},t)}{\partial u_1^3} - \frac{\pi g_0}{2}u_2^2\varphi(\boldsymbol{u},t) = 0$$

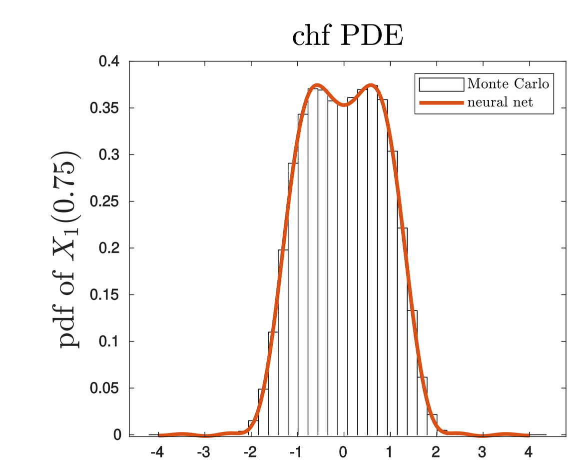
- Domain is $(u_1,u_2) \in [-6,6]^2$, $t \in [0,1]$
- Represent $\widetilde{\varphi}(\boldsymbol{x},t)$ as a feedforward network with 3 inputs, 1 output, 5 hidden layers with 50 neurons each
- Value of loss function at training collocation points is 5.3324×10^{-5}
- Symmetry of Brownian motion and drift terms imply chf is real-valued
- Compare with Monte Carlo estimate $\varphi^{MC}(\boldsymbol{u},t)$





- Marginal pdf of X_1 at $t = 0.75$ from the NN approximation to the Fokker-Planck and chf PDE





- More examples, i.e. Poisson white noise and 4-D example, in cited reference

Reference and Acknowledgements

- W.I.T. Uy, M.D. Grigoriu, *Neural network representation of the probability density function of diffusion cesses.* Chaos: An Interdisciplinary Journal of Nonlinear Science, 30(9):093118, 2020.

- This work is partially supported by NSF grant CMMI-1639669.

Wayne Isaac Tan Uy (wayne.uy@cims.nyu.edu), https://cims.nyu.edu/~wtu1/