Adaptive deep density approximation for Fokker-Planck equations

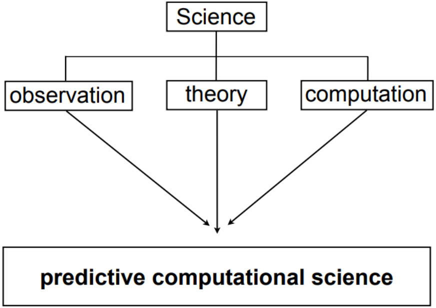
Peng Cheng Laboratory

Kejun Tang
tangkj@pcl.ac.cn
July 15, 2021

Joint work with Xiaoliang Wan (LSU) and Qifeng Liao (ShanghaiTech)

Outline

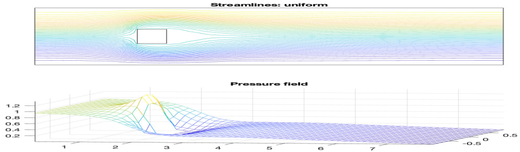
Background



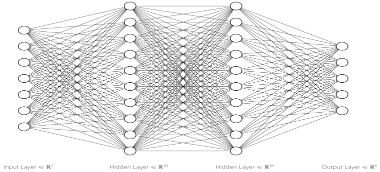
- Micro-electromechanical system (MEMS)
- Aerospace
- Underwater Acoustics
- ...

Background

- Mathematical (physical) model: PDEs or ODEs



- Data-driven model (e.g., deep neural networks): no proper physical model but massive available data



- Numerical methods
  Both of them need numerical methods

Machine learning in scientific computing

- Uncertainty quantification (UQ): (Bayesian) Surrogate model, [Zhu and Zabaras, 2018]; Physical informed neural networks [Raissi, Perdikaris and Karniadakis, 2018]
- Density estimation:
  Domain decomposition for uncertainty quantification, [Liao and Willcox, 2015]; Importance sampling estimator by flow model, [Wan and Wei, 2020]
- Deep neural networks for PDEs:
  Deep Ritz, [E and Yu, 2017] ; Deep Galerkin [Sirignano and Spiliopoulos, 2018]; Physical constraint, [Zhu and Zabaras, 2019]; D3M, [Li, Tang, Wu and Liao, 2019]; PFNN, [Sheng and Yang, 2020]

Goal

Traditional numerical method

- high fidelity
- suffers from the curse of dimensionality

Machine (deep) learning approach

- low fidelity
- weaker dependence on dimensionality

our purpose:

Develop a new deep generative model for density estimation and apply it to solve Fokker-Planck equations

- deep networks to alleviate curse of dimensionality
- develop adaptive scheme using machine learning technique

Differential equations

$$\begin{aligned} \mathcal{L}(\mathrm{x}; u(\mathrm{x})) &= s(\mathrm{x}) \qquad \forall \mathrm{x} \in \Omega, \\ \mathfrak{b}(\mathrm{x}; u(\mathrm{x})) &= g(\mathrm{x}) \qquad \forall \mathrm{x} \in \partial\Omega. \end{aligned}$$

$\mathcal{L}$: partial differential operator, $\mathfrak{b}$: boundary operator.

FEM:

1. mesh
2. basis

→

Deep methods:

1. samples
2. neural networks

Why deep methods

- fast inference
- attack high dimensional problems

Differential equations

$$\mathcal{L}\left(x; u\left(x\right)\right) = s(x) \qquad \forall\left(x\right) \in \Omega,$$
$$\mathfrak{b}\left(x; u\left(x\right)\right) = g(x) \qquad \forall\left(x\right) \in \partial\Omega.$$

$\mathcal{L}$: partial differential operator, $\mathfrak{b}$: boundary operator.

How deep methods do

$$J\left(u(\mathrm{x};\Theta)\right) = \|r(\mathrm{x};\Theta)\|_{2,\Omega}^2 + \|b(\mathrm{x};\Theta)\|_{2,\partial\Omega}^2,$$

where $r(\mathrm{x};\Theta) = \mathcal{L}u(\mathrm{x};\Theta) - s(\mathrm{x})$, and $b(\mathrm{x};\Theta) = \mathfrak{b}u(\mathrm{x};\Theta) - g(\mathrm{x})$

Key point: $u(\mathrm{x};\Theta) \rightarrow u(\mathrm{x})$ compute residual loss by uniform sampling

Fokker-Planck equations

$$\mathcal{L}p(\mathrm{x}) = \nabla \cdot [p(\mathrm{x})\nabla V(\mathrm{x})] + \nabla \cdot [\nabla \cdot (p(\mathrm{x})\mathrm{D}(\mathrm{x}))] = 0,$$

with the boundary condition

$$p(\mathrm{x}) \to 0 \quad \text{as} \quad \|\mathrm{x}\|_2 \to \infty, \tag{1}$$

and some extra constraints on $p(\mathrm{x})$

$$\int_{\mathbb{R}^d} p(\mathrm{x})d\mathrm{x} = 1, \quad \text{and} \quad p(\mathrm{x}) \geq 0, \tag{2}$$

where $\|\mathrm{x}\|_2$ indicates the ℓ_2 norm of x.

Several difficulties

- The boundary condition and the constraints of $p(\mathrm{x})$ may not be easily satisfied
- It requires a fine mesh to capture the whole information when the target density is multimodal problems

Fokker-Planck equations

$$\mathcal{L}p(\mathrm{x}) = \nabla \cdot [p(\mathrm{x})\nabla V(\mathrm{x})] + \nabla \cdot [\nabla \cdot (p(\mathrm{x})\mathrm{D}(\mathrm{x}))] = 0,$$

with the boundary condition

$$p(\mathrm{x}) \to 0 \quad \text{as} \quad \|\mathrm{x}\|_2 \to \infty,$$

and some extra constraints on $p(\mathrm{x})$

$$\int_{\mathbb{R}^d} p(\mathrm{x})d\mathrm{x} = 1, \quad \text{and} \quad p(\mathrm{x}) \geq 0,$$

where $\|\mathrm{x}\|_2$ indicates the ℓ_2 norm of x.

Flow-based generative model

- A PDF model: Flow-based generative model provides an explicit density function that satisfies naturally all constraints on $p(\mathrm{x})$
- Adaptive procedure: A simple but effective adaptive strategy for the approximation of Fokker-Planck equations

Adaptive procedure

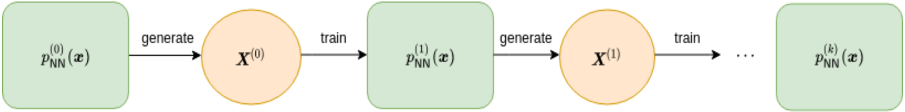
Residual loss functional

$$J\left(p_X(\mathrm{x};\Theta)\right) = \mathbb{E}_{X\sim p(\mathrm{x})} r^2\left(X;\Theta\right) = \mathbb{E}_{X\sim p(\mathrm{x})}\left(\mathcal{L}(p_X(X;\Theta))\right)^2$$

- using current points to minimize residual loss

$$\min_{\Theta} \frac{1}{N}\sum_{i=1}^{N} r^2(X^{(i)};\Theta)$$

- update PDF model and generate new samples
- repeat the above two steps



Deep generative models

Key points

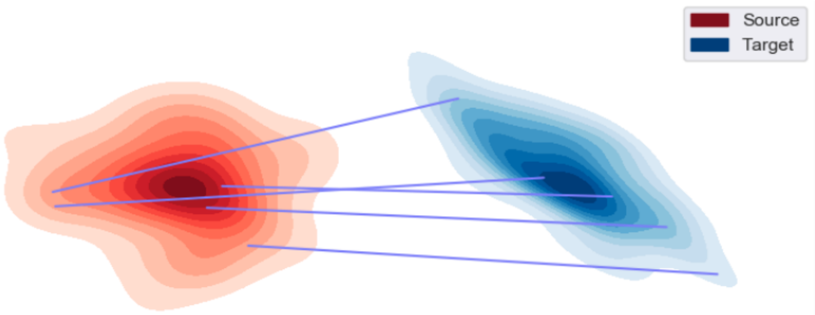
- design a valid PDF model
- efficient sampling

Related works

- GAN [Goodfellow et.al, 2014] [Arjovsky, Chintala and Bottou, 2017]
- VAE [Kingma and Welling, 2014]
- NICE [Dinh, Krueger and Bengio, 2014], Real NVP [Dinh, Dickstein, and Bengio, 2016]

- GAN & VAE generate sample efficiently
- but cannot get PDF

Optimal transport



push the data distribution to a prior distribution

- find a mapping
- prior is simple
- mapping must be highly nonlinear (deep neural networks)

Invertible mapping

Flow: construct a PDF model

$$p_X(\mathrm{x}) = p_Z(f(\mathrm{x})) \left|\det \nabla_{\mathrm{x}} f\right|$$

f is a bijection

$$\mathrm{z} = f(\mathrm{x}) = f_{[L]} \circ \ldots \circ f_{[1]}(\mathrm{x})$$

$$\mathrm{x} = f^{-1}(\mathrm{z}) = f_{[1]}^{-1} \circ \ldots \circ f_{[L]}^{-1}(\mathrm{z})$$

why invertible mapping

- GAN and VAE can not provide an explicit PDF though they can generate samples efficiently
- Invertible mapping provides an explicit PDF
- Flow (invertible mapping) can generate samples efficiently

Invertible mapping

Flow: construct a PDF model

$$p_X(\mathrm{x};\Theta_f) = p_Z(f(\mathrm{x}))\left|\det \nabla_{\mathrm{x}} f\right|$$

f is a bijection

$$\mathrm{z} = f(\mathrm{x}) = f_{[L]} \circ \ldots \circ f_{[1]}(\mathrm{x})$$

$$\mathrm{x} = f^{-1}(\mathrm{z}) = f_{[1]}^{-1} \circ \ldots \circ f_{[L]}^{-1}(\mathrm{z})$$

key points

- f is a bijection
- $\det \nabla_{\mathrm{x}} f$ can be easily computed

A new affine coupling layer

Each $f_{[i]}$

- $f_{[i]}$ is a bijection
- $\det \nabla_{\mathrm{x}} f_{[i]}$ can be easily computed
- $|\det \nabla_{\mathrm{x}} f| = \prod_{i=1}^{L} |\det \nabla_{\mathrm{x}_{[i-1]}} f_{[i]}|$

structure of $f_{[i]}$

$$\mathrm{x}_{[i],1} = \mathrm{x}_{[i-1],1}$$
$$\mathrm{x}_{[i],2} = \mathrm{x}_{[i-1],2} \odot \left(1 + \alpha \ tanh(\mathrm{s}_i(\mathrm{x}_{[i-1],1}))\right) + e^{\beta_i} \odot tanh(\mathrm{t}_i(\mathrm{x}_{[i-1],1})),$$

where $\mathrm{x}_{[i]} = [\mathrm{x}_{[i],1}, \mathrm{x}_{[i],2}]^T \in \mathbb{R}^d$, $\mathrm{s}_i : \mathbb{R}^m \mapsto \mathbb{R}^{d-m}$ and $\mathrm{t}_i : \mathbb{R}^m \mapsto \mathbb{R}^{d-m}$ are the scaling and the translation depending on $\mathrm{x}_{[i-1],1}$

$$(\mathrm{s}_i, \mathrm{t}_i) = NN_{[i]}(\mathrm{x}_{[i-1],1}).$$

A new affine coupling layer

Each $f_{[i]}$

- $f_{[i]}$ is a bijection
- $\det \nabla_{\mathsf{x}} f_{[i]}$ can be easily computed
- $|\det \nabla_{\mathsf{x}} f| = \prod_{i=1}^{L} |\det \nabla_{\mathsf{x}_{[i-1]}} f_{[i]}|$

inverse and determinant of Jacobian for $f_{[i]}$

$$
\begin{aligned}
\mathsf{x}_{[i-1],1} &= \mathsf{x}_{[i],1} \\
\mathsf{x}_{[i-1],2} &= \left(\mathsf{x}_{[i],2} - e^{\beta_i} \odot \tanh(\mathsf{t}_i(\mathsf{x}_{[i-1],1}))\right) \odot \left(1 + \alpha \tanh(\mathsf{s}_i(\mathsf{x}_{[i-1],1}))\right)^{-1}
\end{aligned}
$$

$$
\nabla_{\mathsf{x}_{[i-1]}} f_{[i]} = \begin{bmatrix} \mathsf{I} & 0 \\ \nabla_{\mathsf{x}_{[i-1],1}} \mathsf{x}_{[i],2} & \mathrm{diag}(1 + \alpha \tanh(\mathsf{s}_i(\mathsf{x}_{[i-1],1}))) \end{bmatrix}
$$

A new affine coupling layer

structure of $f_{[i]}$

$$x_{[i],1} = x_{[i-1],1}$$
$$x_{[i],2} = x_{[i-1],2} \odot \left(1 + \alpha \tanh(s_i(x_{[i-1],1}))\right) + e^{\beta_i} \odot \tanh(t_i(x_{[i-1],1})),$$

where $x_{[i]} = [x_{[i],1}, x_{[i],2}]^T \in \mathbb{R}^d$, $s_i : \mathbb{R}^m \mapsto \mathbb{R}^{d-m}$ and $t_i : \mathbb{R}^m \mapsto \mathbb{R}^{d-m}$ are the scaling and the translation depending on $x_{[i-1],1}$

advantages

- adapts the trick of ResNet [He et. al, 2015]
- e^{β_i} depends on the data points directly instead of the value of $x_{[i-1]}$
- $(1-\alpha)^{d-m} \leq \det\left(\nabla_{x_{[i-1]}} f_{[i]}\right) \leq (1+\alpha)^{d-m}, \alpha \in (0,1)$

Scale and bias layer

normalization

$$\hat{x}_{[i]} = a_i \odot x_{[i]} + b_i, \;\; i = 1, \ldots, L$$

$$x_{[i]} = (\hat{x}_{[i]} - b_{[i]}) \odot a_i^{-1}$$

$$|\det \nabla_{x_{[i]}} \hat{x}_{[i]}| = \prod_{j=1}^{N} a_i(j)$$

advantages

- normalization can improve training performance and stability of deep neural networks [Ioffe and Szegedy, 2015]

KRnet

Knothe-Rosenblatt rearrangement

$$z = \mathcal{T}^{-1}(x) = \begin{bmatrix} \mathcal{T}_1(x_1) \\ \mathcal{T}_2(x_1, x_2) \\ \vdots \\ \mathcal{T}_N(x_1, \dots, x_d) \end{bmatrix}$$

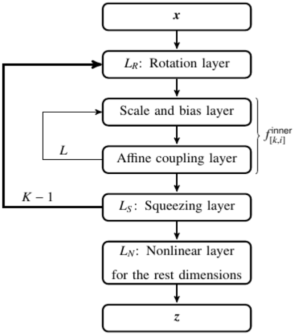
$$z = f_{\mathrm{KR}} = L_N \circ f_{[K-1]}^{\text{outer}} \circ \cdots \circ f_{[1]}^{\text{outer}}(x)$$

$$f_{[k]}^{\text{outer}} = L_S \circ f_{[k,L]}^{\text{inner}} \circ \cdots \circ f_{[k,1]}^{\text{inner}} \circ L_R$$

advantages

- optimal transport [Carlier, Galichon and Santambrogio, 2010]

KRnet



structure of KRnet

- squeezing layer
- rotation layer
- affine coupling layer
- nonlinear layer

KRnet

Squeezing layer

deactivate some dimensions using a mask

$$\mathsf{q} = [\underbrace{1,\ldots,1}_{n},\underbrace{0,\ldots,0}_{d-n}]^T$$

question

- how do we choose which dimensions to be deactivate?

KRnet

rotation layer

learn a rotation matrix

$$\hat{W} = \begin{bmatrix} W & 0 \\ 0 & I \end{bmatrix} \in \mathbb{R}^{d \times d}$$

$$\hat{x} = \hat{W}x$$

$$\hat{W} = \begin{bmatrix} L & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} U & 0 \\ 0 & I \end{bmatrix}$$

- rotation layer tells us about which dimensions deactivate
- rotation layer improves robustness

KRnet

nonlinear layer

Affine coupling layer is an affine transformation (linear with respect to x), and we here define

$$F(x) = \int_0^x p(t)dt,$$

Let $0 = x_0 < x_1 < \ldots < x_{m+1}$ be a partition of $[0, 1]$, and $p(x)$ is a piece-wise linear polynomial (to be learned) defined on these intervals

$$\hat{x} = \begin{cases} F((x+a)/(2a)) & \text{when } x \in [-a, a], \\ \hat{x} \leftarrow x & \text{when } x \in (-\infty, a) \cup (a, \infty). \end{cases}$$

- enhance the representation capability of flow model
- improve nonlinearity

Density estimation and solve the Fokker-Planck equation

KR-net

$$p_X(x) = p_Z(f(x)) \left|\det \nabla_x f_{KR}\right|$$

density estimation and solving the Fokker-Planck equation

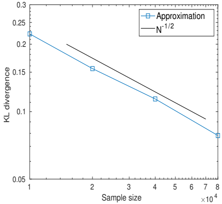
similarities

- construct a KR-net
- minimize a loss function

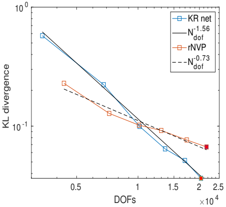
difference

- density estimation: data
  solve the Fokker-Planck equation: no data
- loss function is different
  density estimation: negative log likelihood
  solve the Fokker-Planck equation: residual loss

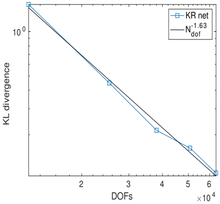
Density estimation



(a) $KL(p_X(x)||p_X(x;\theta))$ w.r.t. N, $d = 4$



(b) $KL(p_X(x)||p_X(x;\theta))$ w.r.t. model complexity, $d = 4$.



(c) $KL(p_X(x)||p_X(x;\theta))$ w.r.t. N, $d = 8$

setting

- $X_i \sim \mathrm{Logistic}(0, s)$,
  $\alpha_s = 3, s = 2, C = 7.6, \theta_{r,i} = \pi/4$ (i is even) *or* $3\pi/4$ (i is odd)
- $|\mathrm{R}_{\alpha_s,\theta_{r,i}} X^{(j)}(i : i+1)| \geq C, \; i = 1, \ldots, d-1$,

$$\mathrm{R}_{\alpha_s,\theta_{r,i}} = \begin{bmatrix} \alpha_s & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_{r,i} & -\sin\theta_{r,i} \\ \sin\theta_{r,i} & \cos\theta_{r,i} \end{bmatrix}$$

Density estimation

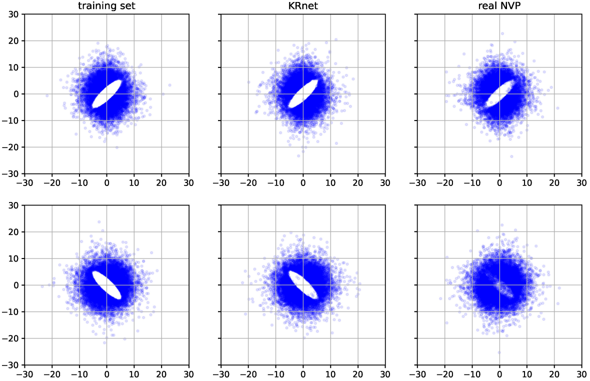
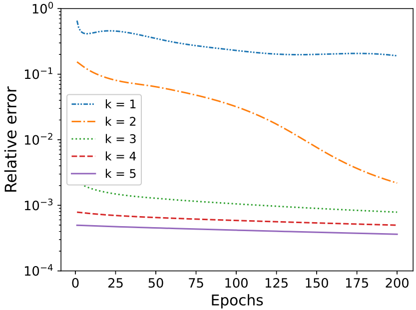


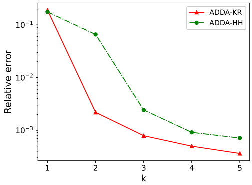
Figure: Training data, and data sampled from KRnet and real NVP. The first row: x_1 and x_2. The second row: x_4 and x_5. .

Solve Fokker-Planck equations

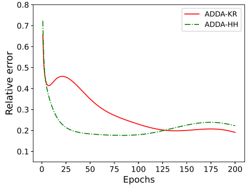


setting

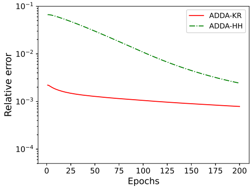
- $\frac{\partial p(x,t)}{\partial t} = \nabla \cdot [p(x,t)\nabla \log(\beta_1 p_1(x) + \beta_2 p_2(x))] + \nabla^2 p(x,t)$
- stationary solution
  $p_{st}(x) = \beta_1 p_1(x) + \beta_2 p_2(x), x \in \mathbb{R}^2, p_i(x)$: Gaussian distribution



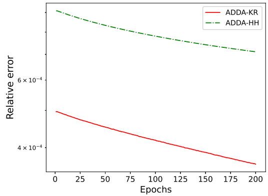
(a) KL divergece w.r.t. k-th model.



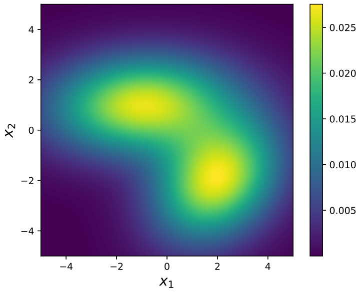
(b) The convergence behavior for $k = 1$.



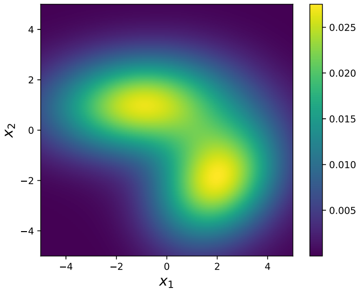
(c) The convergence behavior for $k = 3$.



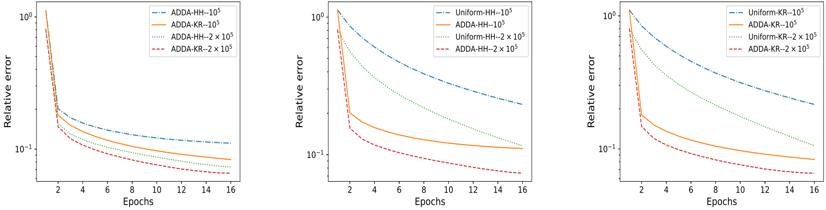
(d) The convergence behavior for $k = 5$.



(e) Exact solution $p(x)$



(f) ADDA approximation



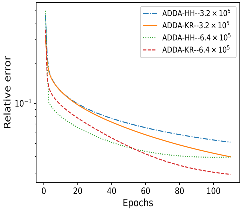
(g) Comparison of KR and HH: KL-divergence w.r.t epochs

(h) KL-divergence w.r.t epochs for HH

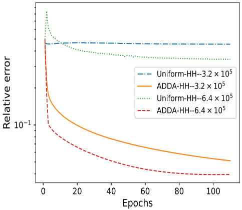
(i) KL-divergence w.r.t epochs for KR

setting (HH refers to Real NVP)

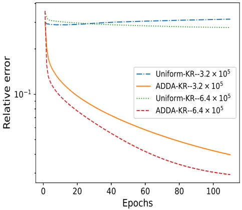
- $\frac{\partial p(x,t)}{\partial t} = \nabla \cdot [p(x,t)\nabla \log(\beta_1 p_1(x) + \beta_2 p_2(x))] + \nabla^2 p(x,t)$
- stationary solution $p_{st}(x) = \beta_1 p_1(x) + \beta_2 p_2(x), x \in \mathbb{R}^4, p_i(x)$: Gaussian distribution



(j) Comparison of KR and HH: KL-divergence w.r.t epochs



(k) KL-divergence w.r.t epochs for HH



(l) KL-divergence w.r.t epochs for KR

setting

- $\frac{\partial p(x,t)}{\partial t} = \nabla \cdot [p(x,t)\nabla \log(\beta_1 p_1(x) + \beta_2 p_2(x))] + \nabla^2 p(x,t)$
- stationary solution
  $p_{st}(x) = \beta_1 p_1(x) + \beta_2 p_2(x), x \in \mathbb{R}^8, p_i(x)$: Gaussian distribution

Conclusion

- Propose a novel flow-based generative model (KRnet)
- Develop a novel adaptive deep density approximation strategy based on KRnet
- Adaptive sampling procedure is efficient for Fokekr-Planck equations

Q & A

Thank you for your attention