Wasserstein gradient flows on the push-forward generative model

Yijie Jin(Georgia Tech), Wuchen Li(University of South Carolina), Shu Liu(UCLA), Hao Wu (Georgia Tech), Xiaojing Ye (Georgia State University), Hongyuan Zha(CUHKSZ), and Haomin Zhou (Georgia Tech)

These are the slides for the following two manuscripts:

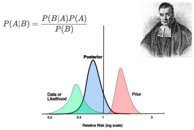
1. Neural parametric Fokker-Planck equations, 2022.
2. Parameterized Wasserstein gradient flow, 2025.

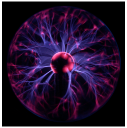
Table of Contents

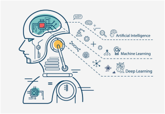
What are Wasserstein Gradient flows (WGFs)?

Wasserstein gradient flows[1] are a series of time-evolution partial differential equations (PDE), describing the gradient descent dynamic of the probability density function ρ with respect to a certain functional $\mathcal{F}(\rho)$ of ρ.

- (Fokker-Planck) $\frac{\partial \rho}{\partial t} = \nabla \cdot (\rho \nabla V) + D\Delta\rho$, $D > 0$;
- (Porous medium) $\frac{\partial \rho}{\partial t} = \Delta(\rho^m)$, $m > 1$;
- (Aggregation) $\frac{\partial \rho}{\partial t} = \nabla \cdot (\rho \nabla (W * \rho))$.







[1] R. Jordan, D. Kinderlehrer, F. Otto. The Variational Formulation of the Fokker–Planck Equation. SIMA, 1998

Literature Review

Applications of deep learning techniques on time-evolution PDEs:

- (General time evolution PDEs) [Du, et al. 2021], [Anderson, et al. 2021], [Bruna, et al. 2022], [Gaby, et al. 2023], etc.

With a special focus on gradient flows.

- (L^2-gradient flow & general diffusions) [Hu et al. 2022], [Lee et al. 2024], etc.
- (Wasserstein gradient flows) [Mokrov, et al. 2021], [Fan, et al. 2022], [Bonet, et al. 2022], etc.

Generative models

What is a (push-forward) generative model[1,2]?

- Consider a class of push-forward maps $\{T_\theta\}_{\theta\in\Theta}$ ($T_\theta : \mathbb{R}^d \to \mathbb{R}^d$) indexed by parameter $\theta \in \Theta \subset \mathbb{R}^m$.
- Obtain a family of parametric distributions (known as (push-forward) generative model)

$$\mathcal{P}_\Theta = \left\{\rho_\theta = T_{\theta\sharp}\lambda \mid \theta \in \Theta\right\}$$

  Here $T_{\theta\sharp}\lambda$ is defined as $\rho_\theta(E) = \lambda(T_\theta^{-1}(E))$ for any measurable set $E \subset \mathbb{R}^d$.
- λ can be chosen as easy-to-sample reference distribution, such as Gaussian distribution or uniform distribution.

[1] Ian J. Goodfellow et al. Generative Adversarial Nets. NeurIPS, 2014

[2] Martin Arjovsky et al. Wasserstein Generative Adversarial Networks. ICML, 2016

Generative models

There are different ways to choose T_θ of the generative model.

- Affine: $T_\theta(x) = Ux + b$, $\theta = (U, b), U \in GL_d(\mathbb{R}),\ b \in \mathbb{R}^d$;
- Linear combination of basis functions: $T_\theta(x) = \sum_{k=1}^m \theta_k \varphi_k(x)$, $\theta = (\theta_1, ..., \theta_m)$, $\varphi_k : \mathbb{R}^d \to \mathbb{R}^d$ basis function;
- Certain kind of neural network with parameter θ; Such as Normalizing Flows[1], Neural ODEs[2]

Easy to sample: Suppose we want to sample from a probability distribution ρ, and we manage to find a $\theta \in \Theta$ s.t. $\rho_\theta \approx \rho$, then samples from ρ_θ can be conveniently and efficiently generated:

- Sample $\boldsymbol{Z}_1, ..., \boldsymbol{Z}_N$ form λ;
- Then $T_\theta(\boldsymbol{Z}_1), ..., T_\theta(\boldsymbol{Z}_N)$ are the desired samples.

[1] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. ICML, 2015

[2] R. T. Chen, Y. Rubanova, J. Bettencourt, D. K. Duvenaud, Neural ordinary differential equations, NeurIPS, 2018

Our treatment Project the WGF onto the parameter space Θ. $\Rightarrow$ Obtain trajectory $\{\theta_t\}$ on Θ.

Hopefully, $T_{\theta_t \sharp}\lambda$ serve as a valid approximation of the true solution ρ_t.

Motivation

- **Dimension reduction**:

  WGF on probability manifold (infinite-dimensional)

  $\Downarrow$

  ODE on parameter space (finite-dimensional)

- $\rho_\theta = T_{\theta\sharp}\lambda$ automatically preserves **positivity** and **mass conservation**.
- Samples from ρ_θ can be **efficiently** generated.

Goal

Propose a *scalable* and *sampling-friendly* method for simulating WGF.

Sketch of our idea

Question: How can we "project" the WGF onto the parameter space Θ?

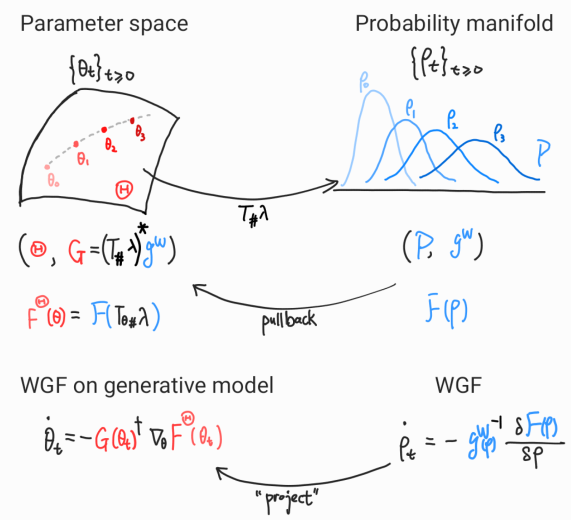


Table of Contents

Wasserstein metric[1]

- In order to introduce the Wasserstein metric on probability manifold, we consider the space of pushforward maps with reference measure λ,

$$\mathcal{O} = \{T : \mathbb{R}^d \to \mathbb{R}^d \mid T \in L^2(\mathbb{R}^d; \mathbb{R}^d, \lambda)\}.$$

Define the $L^2(\lambda)$ metric $g_{L^2(\lambda)}$ on $\mathcal{O}$ as

$$g_{L^2(\lambda)}(\sigma, \sigma) \triangleq \int_{\mathbb{R}^d} \|\sigma(z)\|^2 \, d\lambda(z), \quad \forall\, \sigma \in \mathcal{T}_T\mathcal{O} \cong L^2(\mathbb{R}^d; \mathbb{R}^d, \lambda).$$

$g_{L^2(\lambda)}$ will be consistent with the L^2 distance defined on $\mathcal{O}$.

- We consider the probability manifold $\mathcal{P}$ with finite second-order momentum, i.e. $\int_{\mathbb{R}^d} |x|^2 d\rho < \infty$ for any $\rho \in \mathcal{P}$.

- Then the pushforward operation $T \mapsto T_\sharp \lambda$ yields a *submersion* Π from $\mathcal{O}$ to the probability manifold $\mathcal{P}$.

[1] F. Otto, The geometry of dissipative evolution equations the porous medium equation. Communications in Partial Differential Equations Volume 26, 2001 - Issue 1-2.

Wasserstein metric

- Denote $\rho = \Pi(T) = T_\sharp\lambda \in \mathcal{P}$, the submersion Π induces the pushforward map $\Pi_*|_{T\in\mathcal{O}} : \mathcal{T}_T\mathcal{O} \to \mathcal{T}_\rho\mathcal{P}$ at any $T \in \mathcal{O}$. Π_* is the linear surjective map, but generally not injective.

- We can introduce the **Wasserstein metric** g^W defined on $\mathcal{TP}$ as

$$g^W(\rho)(s,s) = \min_{\sigma\in\mathcal{T}_T\mathcal{O},\ \Pi_*\sigma=s} g_{L^2(\lambda)}(\sigma,\sigma), \quad \forall\ s \in \mathcal{T}_\rho\mathcal{P}^1.$$

  One can compute $\Pi_*\sigma = -\nabla\cdot(\rho\sigma)$, thus, the constraint reads $-\nabla\cdot(\rho\sigma) = s$.

- By introducing Lagrange multiplier ψ, it is not hard to verify that

$$g^W(\rho)(s,s) = \int |\nabla\psi|^2\rho dx, \quad -\nabla\cdot(\rho\nabla\psi) = s.$$

[1] $\mathcal{T}_\rho\mathcal{P}$ can be treated as the functional space $\left\{\text{functions } f \text{ on } \mathbb{R}^d \text{ with } \int_{\mathbb{R}^d} f\ dx = 0\right\}$.

Wasserstein metric

- At least formally, one can write $g^W(\rho) = -(\nabla \cdot (\rho \nabla))^\dagger$.
- When $\rho > 0$, $(\mathcal{P}, g^W)$ can be treated as a "Riemannian manifold", the metric g^W is consistent with the Wasserstein-2 distance on $\mathcal{P}$. Thus, we denote $(\mathcal{P}, g^W)$ as the **Wasserstein manifold**.

Recall the Wasserstein-2 distance:
For any $\rho_a, \rho_b \in \mathcal{P}$, the 2-Wasserstein distance between ρ_a, ρ_b is

$$W_2^2(\rho_a, \rho_b) = \min_{\substack{\text{Marginals of } \pi \\ \text{are fixed to be } \rho_a, \rho_b.}} \left\{ \iint_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2 \, d\pi(x, y) \right\}.$$

Wasserstein gradient flow (WGF)

- Given a differentiable functional $\mathcal{F}(\cdot)$ on $(\mathcal{P}, g^W)$.
- We define the **Wasserstein gradient** as the manifold gradient on $(\mathcal{P}, g^W)$. Thus, Wasserstein gradient of $\mathcal{F}(\cdot)$ is

$$\mathrm{grad}_W \mathcal{F}(\rho) = g^W(\rho)^\dagger \ \frac{\delta \mathcal{F}(\rho)}{\delta \rho} = -\nabla \cdot (\rho \nabla \frac{\delta \mathcal{F}(\rho)}{\delta \rho}).$$

And the **Wasserstein gradient flow (WGF)** of $\mathcal{F}(\rho)$ is

$$\frac{\partial \rho}{\partial t} = -\mathrm{grad}_W \mathcal{F}(\rho) = \nabla \cdot (\rho \nabla \frac{\delta \mathcal{F}(\rho)}{\delta \rho}).$$

[Manifold gradient] Consider Riemannian manifold $(\mathcal{M}, g)$ and a differentiable function $f : \mathcal{M} \to \mathbb{R}$, the manifold gradient $\mathrm{grad} f$ is defined as

$$g(\dot{\gamma}_0, \ \mathrm{grad} f(x)) = \frac{d}{d\tau} f(\gamma_\tau)\Big|_{\tau=0},$$

for any smooth curve $\{\gamma_\tau\}_{-\epsilon \le \tau \le \epsilon}$ passes through x, i.e., $\gamma_0 = x$. Informally, we can write $\mathrm{grad} f(x) = g(x)^{-1} \nabla f(x)$.

Examples of Wasserstein gradient flows

WGF	Equation	Associated functional $\mathcal{F}(\rho)$
Fokker-Planck equation	$\partial_t \rho = \nabla \cdot (\rho \nabla V) + D\Delta\rho$	$\int V\rho + D\rho \log \rho \; dx$
Porous Medium equation	$\partial_t \rho = \Delta(\rho^m)$	$\frac{1}{m-1} \int \rho^m \; dx$
Aggregation equation	$\partial_t \rho = \nabla \cdot (\rho \nabla (W * \rho))$	$\iint W(x-y)\rho(x)\rho(y) \; dxdy$
...	...	...

Table: Some examples of Wasserstein gradient flows.

Table of Contents

Derivation of parametrized Wasserstein gradient flow

Project WGF of $\mathcal{F}(\rho)$ onto the generative model $\{T_\theta\}_{\theta\in\Theta}$:

- Pull back the Wasserstein metric g^W defined on $\mathcal{P}$ to Θ. We obtain the metric tensor $G(\theta)$;
- Define the corresponding functional $F^\Theta(\theta) = \mathcal{F}(T_{\theta\sharp}\lambda)$;
- Compute the gradient flow of $F^\Theta(\theta)$ on (Θ, G).

$$\boxed{\dot{\theta}_t = -G(\theta_t)^\dagger \nabla_\theta F(\theta_t).} \quad \left(\text{Recall: } \frac{\partial \rho_t}{\partial t} = -g^W(\rho_t)^{-1}\frac{\delta \mathcal{F}(\rho_t)}{\delta \rho}(x)\right)$$

$G(\theta)$ may not be invertible for general choices of T_θ. We use its pseudo-inverse $G(\theta)^\dagger$ instead of $G(\theta)^{-1}$.

- This is an ODE on Θ. We call it **Parametrized Wasserstein gradient flow (PWGF)**.

Formulation of $G(\theta)$ [L. Li Zha Zhou 2022][1]

Denote $\mathscr{G} : \Theta \ni \theta \mapsto \rho_\theta = T_{\theta\sharp}\lambda \in (\mathcal{P}, g^W)$, recall that Θ is an open set of $\mathbb{R}^m$. Then $G(\theta) = \mathscr{G}^* g^W$ is an $m \times m$ positive semi-definite matrix, with

$$G_{ij}(\theta) = \int_{\mathbb{R}^d} \nabla\psi_i(T_\theta(z)) \cdot \nabla\psi_j(T_\theta(z)) \, d\lambda(z), \quad 1 \leq i, j \leq m.$$

For each $k = 1, 2, \ldots, m$, ψ_k solves the equation

$$\nabla \cdot (\rho_\theta \nabla\psi_k(x)) = \nabla \cdot (\rho_\theta \frac{\partial T_\theta}{\partial \theta_k}(T_\theta^{-1}(x))). \qquad (1)$$

Here $\rho_\theta = T_{\theta\sharp}\lambda$. No explicit solution when dimension $d > 1$.

$G(\theta)$ could be hard to evaluate. Any remedies?

[1] S. Liu, W. Li, H. Zha, and H. Zhou. Neural parametric Fokker-Planck equations. SINUM 2022

Another perspective: WGF via Lagrangian scheme

- Recall the $L^2(\lambda)$ space of pushforward maps (diffeomorphisms) with reference measure λ,

$$\mathcal{O} = \{T : \mathbb{R}^d \to \mathbb{R}^d \mid T \in L^2(\mathbb{R}^d; \mathbb{R}^d, \lambda)\}.$$

- Recall the $L^2(\lambda)$ metric $g_{L^2(\lambda)}$ defined on $\mathcal{O}$ as

$$g_{L^2(\lambda)}(\sigma, \sigma) \triangleq \int_{\mathbb{R}^d} \|\sigma(z)\|^2 \, d\lambda(z), \quad \forall \, \sigma \in \mathcal{T}_T\mathcal{O} \cong L^2(\mathbb{R}^d; \mathbb{R}^d, \lambda).$$

- Consider the functional $\mathcal{F}^\sharp : \mathcal{O} \to \mathbb{R}$, $\mathcal{F}^\sharp(T) := \mathcal{F}(T_\sharp \lambda)$.
- WGF under Lagrangian coordinate[2]:

$$\frac{\partial T_t}{\partial t} = -g_{L^2(\lambda)}^{-1} \frac{\delta \mathcal{F}^\sharp(T_t)}{\delta T} = -\frac{1}{\lambda} \frac{\delta \mathcal{F}^\sharp(T_t)}{\delta T} = -\nabla \frac{\delta \mathcal{F}(T_{t\sharp}\lambda)}{\delta \rho}$$

- Consistency with WGF: $\rho_t = T_{t\sharp}\lambda$ solves $\partial_t \rho_t = \nabla \cdot (\rho_t \nabla \frac{\delta \mathcal{F}(\rho_t)}{\delta \rho_t})$.

[2] J. Carrillo, D. Matthes, M. Wolfram. Lagrangian schemes for Wasserstein gradient flows. 2020

Derivation of parametrized Wasserstein gradient flow

Project WGF of $\mathcal{F}^\sharp(T)$ onto the generative model $\{T_\theta\}_{\theta\in\Theta}$:

- Pull back $g_{L^2(\lambda)}$ defined on $\mathcal{O}$ to Θ. We obtain $\widehat{G}(\theta)$;
- Corresponding functional $F^\Theta(\theta) = \mathcal{F}(T_{\theta\sharp}\lambda)$;
- Compute the gradient flow of $F^\Theta(\theta)$ on $(\Theta, \widehat{G})$.

$$\boxed{\dot{\theta}_t = -\widehat{G}(\theta_t)^\dagger \nabla_\theta F(\theta_t).} \quad \left(\text{Recall: } \frac{\partial T_t}{\partial t} = -g_{L^2(\lambda)}^{-1} \frac{\delta \mathcal{F}^\sharp(T_t)}{\delta T}\right),$$

- This is an ODE on Θ. We call it **relaxed Parametrized Wasserstein gradient flow (RPWGF)**.

Formulation of $\widehat{G}(\theta)$ [Jin L. Wu Ye Zhou 2024][3]

Denote $\mathscr{T} : \Theta \ni \theta \mapsto T_\theta \in \mathcal{O}$., Then $\widehat{G}(\theta) = \mathscr{T}^* g_{L^2}$ is an $m \times m$ positive semi-definite matrix, with

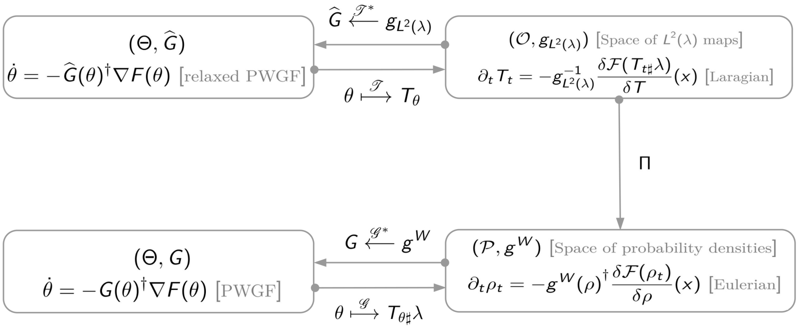
$$\widehat{G}_{ij}(\theta) = \int_{\mathbb{R}^d} \partial_{\theta_i} T_\theta(z) \cdot \partial_{\theta_j} T_\theta(z) \, d\lambda(z), \quad 1 \leq i, j \leq m.$$

$$\widehat{G}(\theta) = \int_{\mathbb{R}^d} \frac{\partial T_\theta(z)}{\partial \theta}^\top \frac{\partial T_\theta(z)}{\partial \theta} \, d\lambda(z).$$

- $\widehat{G}(\theta)$ takes explicit form.
- One can compute the matrix-vector multiplication $\widehat{G}(\theta)\mathbf{v}$ efficienty for arbitrary $\mathbf{v} \in \mathbb{R}^m$. – Important for Krylov subspace iteration!

[1] Y. Jin, S. Liu, H. Wu, X. Ye, H. Zhou. Parametrized Wasserstein gradient flow. JCP, 2024.

Relation diagram



Relation among gradient flows on different spaces

Table of Contents

Error analysis

- $\{\rho_t\}$ is the solution to WGF.
- $\{\theta_t\}$ is the solution to PWGF, denote $\rho_{\theta_t} = T_{\theta_t \sharp}\lambda$.
- How to quantify the discrepancy between ρ_t and ρ_{θ_t}?
- Bound the Wasserstein-2 error $W_2(\rho_{\theta_t}, \rho_t)$.

An important quantity:

$$\delta_0 = \sup_{\theta\in\Theta} \min_{\xi_1,\dots,\xi_m\in\mathbb{R}} \left\{ \int_{\mathbb{R}^d} \left| \sum_{i=1}^m \xi_i \frac{\partial T_\theta}{\partial \theta_i}(z) - \nabla \frac{\delta \mathcal{F}(\rho_\theta)}{\delta \rho}(T_\theta(z)) \right|^2 d\lambda(z) \right\}.$$

- δ_0 quantifies the approximation power of tangent space $\mathrm{span}\{\partial_{\theta_1} T_\theta, \dots, \partial_{\theta_m} T_\theta\}$.
- Explicit bound on δ_0 for 2-layer ReLU Network is available in [1].

[1] X. Zuo, J. Zhao, S. Liu, S. Osher, W. Li Numerical analysis on Neural network projected schemes for approximating one dimensional Wasserstein Gradient flows. arXiv 2402.16821, 2024.

Wasserstein Error Bound (Fokker-Planck case)

We can establish *uniform* (time-independent) error bound[1] for $W_2(\rho_{\theta_t}, \rho_t)$ when $\mathcal{F}(\rho) = D_{\mathrm{KL}}(\rho \| \rho_*) = \int V\rho + D\rho \log \rho \, dx$, with $\rho_*(x) = \frac{1}{Z} e^{-\frac{V(x)}{D}}$.

Uniform error bound [L. Li Zha Zhou 2022][4]

Suppose the Hessian of the potential function V is bounded below by a constant λ, i.e. $\nabla^2 V \succeq \lambda\, I$. The W_2 error $W_2(\rho_{\theta_t}, \rho_t)$ at any time $t > 0$ can be uniformly bounded by:

1. $O(\sqrt{\delta_0} + E_0)$ when $\lambda > 0$;
2. $O\left(\sqrt{\delta_0} \log \left(\frac{1}{\sqrt{\delta_0 + E_0}} \right) + E_0\right)$ when $\lambda = 0$;
3. $O((E_0 + \sqrt{\delta_0})^{\alpha})$ when $\lambda < 0$. Here $\alpha \in (0,1)$ depends on V and D.

Here $E_0 = W_2(\rho_0, \rho_{\theta_0})$ is the initial error.

- Similar bounds can be established for RPWGF.

[4] S. Liu, W. Li, H. Zha, and H. Zhou. Neural parametric Fokker-Planck equations. SIAM Journal of Numerical Analysis, 2022

Table of Contents

Numerical Algorithm (Choice of T_θ)

When dimension d gets larger, we choose T_θ as a special neural network map known as **Normalizing Flow**[1] (NF).

$$T_\theta = f_K \circ f_{K-1} \circ \ldots \circ f_2 \circ f_1,$$

where each f_k takes the form $f_k(x) = x + \tanh(w_k^T x + b_k)u_k$ where $w_k, u_k \in \mathbb{R}^d$, $b_k \in \mathbb{R}$. Then the parameter $\theta = (\ldots, w_k, u_k, b_k, \ldots) \in \mathbb{R}^{(2d+1)K}$.

Two main reasons of choosing NF:

- Strong approximation power;
- Explicit and concise form of $\rho_\theta(T_\theta(x))$, thus $F(\theta) = \mathcal{F}(T_{\theta\sharp}\lambda)$ can be explicitly formed;

[1] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. ICML, 2015

Numerical Algorithm (Brief Sketch)

Solve $\dot{\theta}_t = -G(\theta_t)^{\dagger}\nabla_\theta F(\theta_t)$ with initial condition ρ_0.

- Choose $p = \rho_0$, choose particular parameter θ_0 s.t. $T_{\theta_0} \approx \mathrm{Id}$;
- At each time step $t_k = k \cdot h$, update θ_k by solving the following bi-level optimization problem.

$$\min_\theta \left\{ \int (2\ \nabla\psi(x)\cdot((T_\theta - T_{\theta_k})\circ T_{\theta_k}^{-1}(x)) - |\nabla\psi(x)|^2)\rho_{\theta_k}(x)\ dx + 2hF(\theta) \right\}.$$

$$\text{with } \psi \text{ solves: } \min_\psi \left\{ \int |\nabla\psi(x) - ((T_\theta - T_{\theta_k})\circ T_{\theta_k}^{-1}(x))|^2 \rho_{\theta_k}(x)\ dx \right\}.$$

This bi-level scheme is motivated by the following scheme $\theta_{k+1} = \underset{\theta}{\operatorname{argmin}}\{\langle\theta-\theta_k, G(\theta_k)(\theta-\theta_k)\rangle + 2hF(\theta)\}$. This is the proximal scheme for solving the semi-implicit scheme $\frac{\theta_{k+1}-\theta_k}{h} = -G^{-1}(\theta_k)\nabla F(\theta_{k+1})$. It is also a variant of the well-known JKO scheme for Wasserstein gradient flows[1]

[1] R. Jordan, D. Kinderlehrer, F. Otto. The variational formulation of the Fokker–Planck equation. SIMA, 1998

Numerical Algorithm (Brief Sketch)

Solve $\dot{\theta}_t = -\widehat{G}(\theta_t)^\dagger \nabla_\theta F(\theta_t)$ with initial condition ρ_0.

- Choose $p = \rho_0$, choose particular parameter θ_0 s.t. $T_{\theta_0} \approx \mathrm{Id}$;
- At each time step $t_k = k \cdot h$,
  - Directly evaluate $\widehat{G}(\theta_k)$ and $\nabla_\theta F(\theta_k)$;
  - Apply MINRES algorithm to solve $\widehat{G}(\theta_k)\zeta = \nabla_\theta F(\theta_k)$;
  - Update θ_k by the forward Euler scheme

$$\theta_{k+1} = \theta_k - h \cdot \zeta.$$

Numerical Algorithm (comparison between G and $\widehat{G}$)

$$\dot{\theta}_t = -G(\theta_t)^\dagger \nabla_\theta F(\theta_t)$$

- Lower error bound (recall that $\delta_0 \leq \widehat{\delta}_0$); Use semi-implicit scheme, which is more stable than explicit schemes.
- Need to deal with a bi-level optimization problem at each time step, which is expensive.

$$\dot{\theta}_t = -\widehat{G}(\theta_t)^\dagger \nabla_\theta F(\theta_t)$$

- Faster and more computationally efficient compared to the gradient flow w.r.t. G.
- **Training free**. No training procedure in our algorithm, so one can avoid the optimization error.
- Use forward Euler scheme, which may require small time stepsize h when the system is stiff.

Table of Contents

Numerical example 1 (2D Aggregation equation)

$$\partial_t \rho_t = \nabla \cdot (\rho \nabla (W * \rho)) = \nabla \cdot (\rho(x) \nabla (\int_\Omega W(x-y)\rho(y)dy)),$$

$$\rho(0,x) = \frac{1}{\sqrt{2\pi} b} e^{-\frac{(x-x^0)^2}{b^2}}.$$

- $W(x) = \frac{|x|^4}{4} - \frac{|x|^2}{2}$, $x^0 = (\frac{5}{4}, \frac{5}{4}), b = \frac{3}{5}$.
- The interaction potential W consists of attractive and repulsive parts, and the steady state concentrates on a ring with radius 0.5 centered at x^0.

Numerical examples (2D Aggregation equation)

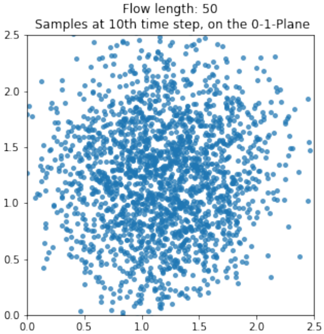


Figure: Samples generated from $T_{\theta_k \sharp}\lambda$ at different time steps.

Numerical example 2 (10D FPE with quadratic potential)

- We consider Fokker-Planck equation (FPE) on $\mathbb{R}^{10}$ with initial distribution $\rho_0 = \mathcal{N}(0, I)$ and quadratic potential

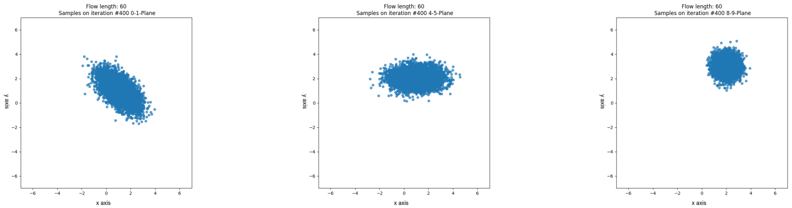
$$V(x) = \frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu) \quad \Sigma = \mathrm{diag}(\Sigma_A, I_2, \Sigma_B, I_2, \Sigma_C)$$
$$\mu = (1,1,0,0,1,2,0,0,2,3)^{\mathrm{T}}.$$

Here we set the diagonal blocks as:

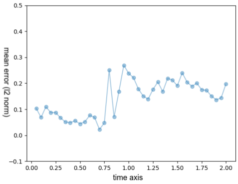
$$\Sigma_A = \begin{bmatrix} \frac{5}{8} & -\frac{3}{8} \\ -\frac{3}{8} & \frac{5}{8} \end{bmatrix} \quad \Sigma_B = \begin{bmatrix} 1 & \\ & \frac{1}{4} \end{bmatrix} \quad \Sigma_C = \begin{bmatrix} \frac{1}{4} & \\ & \frac{1}{4} \end{bmatrix}.$$

- The solution to this Fokker-Planck equation is always Gaussian $\mathcal{N}(\mu(t), \Sigma(t))$. We know the explicit formulation of the mean $\mu(t)$ and the covariance $\Sigma(t)$.

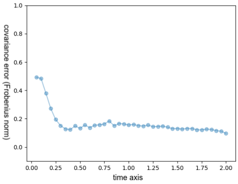
Numerical example 2 (10D FPE with quadratic potential)



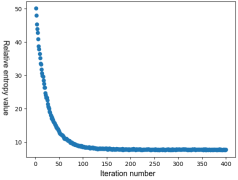
Points sampled from ρ_{θ_t} at $t = 2.0$ (projection on $0-1$, $4-5$, $8-9$ planes).



Mean error (l_2)



Covariance error ($\|\cdot\|_F$)



Plot of $\{H(\theta)\}$

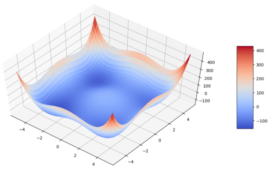
Numerical example 2 (30D Fokker-Planck equation)

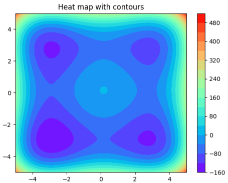
$$\frac{\partial \rho}{\partial t} = \nabla \cdot (\rho \nabla V) + \Delta \rho, \quad \rho_0 = \mathcal{N}(0, I).$$

- The potential function is

$$V(x) = \frac{1}{50}\left(\sum_{i=1}^{30} x_i^4 - 16x_i^2 + 5x_i\right).$$

- Graph of V in 2D:





Numerical example 2 (30D Fokker-Planck equation)

- Time step size $h = 0.005$, solve the equation on $[0.0, 1.8]$.
- We visualize the results on a 2D plane, say the plane spanned by the 5th and 15th components;
- Snapshots at $t = 0.3, 0.6, 0.9, 1.2, 1.5, 1.8$:

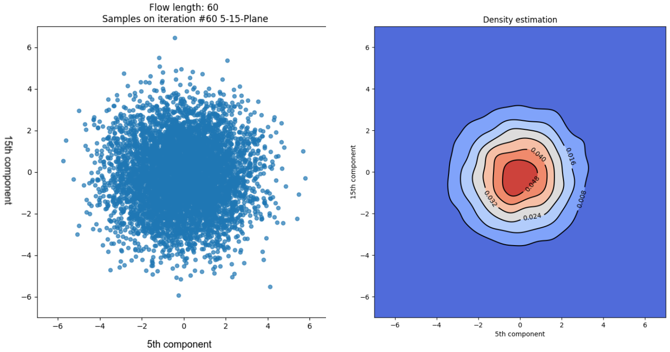


Table of Contents

Summary

The advantages of our computational scheme:

- Sampling-friendly scheme;
- Scalable to higher dimensional space;
- Computation effectiveness;
- Training-free when adopting the metric tensor $\widehat{G}$;
- Theoretical guarantees: energy dissipation, accuracy.

Related References

- Shu Liu, Wuchen Li, Hongyuan Zha, and Haomin Zhou. Neural parametric Fokker-Planck equations. SIAM Journal on Numerical Analysis 2022
- Yijie Jin, Shu Liu, Hao Wu, Xiaojing Ye, Haomin Zhou. Parametrized Wasserstein gradient flow. Journal of Computational Physics, 2025.

Thank you!