The Neural Galerkin Method

CSE Semester Project – Spring 2024

Francesca Bettinelli

Supervisors: Fabio Nobile, Fabio Zoccolan
Chair of Scientific Computing and Uncertainty Quantification (CSQI)

École Polytechnique Fédérale de Lausanne

25 June 2024

Presentation outline

1 Introduction

2 The Neural Galerkin method

3 Adaptive sampling

4 Optimal sampling

5 Conclusion

Introduction

Goal

Approximate the solution of high-dimensional or advection-dominated PDEs via a nonlinear parametrization, e.g., a deep neural network.

- Global residual minimization → Physics-informed neural networks[1]
- Local-in-time residual minimization → Neural Galerkin method[2]

[1] Raissi, Perdikaris, and Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations".

[2] Bruna, Peherstorfer, and Vanden-Eijnden, "Neural Galerkin schemes with active learning for high-dimensional evolution equations"

Introduction

Goal

Approximate the solution of high-dimensional or advection-dominated PDEs via a nonlinear parametrization, e.g., a deep neural network.

- Global residual minimization → Physics-informed neural networks[1]
- Local-in-time residual minimization → Neural Galerkin method[2]

[1] Raissi, Perdikaris, and Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations".

[2] Bruna, Peherstorfer, and Vanden-Eijnden, "Neural Galerkin schemes with active learning for high-dimensional evolution equations"

Introduction

Goal

Approximate the solution of high-dimensional or advection-dominated PDEs via a nonlinear parametrization, e.g., a deep neural network.

- Global residual minimization → Physics-informed neural networks[1]
- Local-in-time residual minimization → Neural Galerkin method[2]

[1] Raissi, Perdikaris, and Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations".

[2] Bruna, Peherstorfer, and Vanden-Eijnden, "Neural Galerkin schemes with active learning for high-dimensional evolution equations"

Problem formulation

- Time-dependent function $u \in \mathcal{U}$, $u : \mathcal{X} \times [0, \infty) \to \mathbb{R}$ characterized by

$$\begin{cases} \partial_t u(\mathbf{x}, t) = f(\mathbf{x}, t, u) & (\mathbf{x}, t) \in \mathcal{X} \times [0, \infty), \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}) & \mathbf{x} \in \mathcal{X}, \end{cases} \tag{PDE}$$

  where $f : \mathcal{X} \times [0, \infty) \times \mathcal{U} \to \mathbb{R}$ is the source term and $u_0 : \mathcal{X} \to \mathbb{R}$ is the initial condition.

- Nonlinear parametrization $\hat{u} : \mathcal{X} \times \Theta \to \mathbb{R}$ with parameters $\theta = \theta(t) \in \Theta$:

$$u(\mathbf{x}, t) = \hat{u}(\mathbf{x}, \theta(t)) \qquad \forall (\mathbf{x}, t) \in \mathcal{X} \times [0, \infty).$$

- Local-in-time residual $r_t : \mathcal{X} \times \Theta \times \dot{\Theta} \to \mathbb{R}$:

$$r_t(\mathbf{x}, \theta, \dot{\theta}) = \nabla_\theta \hat{u}(\mathbf{x}, \theta) \cdot \dot{\theta} - f(\mathbf{x}, t, \hat{u}(\theta)). \tag{RES}$$

Problem formulation

- Time-dependent function $u \in \mathcal{U}$, $u : \mathcal{X} \times [0, \infty) \to \mathbb{R}$ characterized by

$$\begin{cases} \partial_t u(\mathbf{x}, t) = f(\mathbf{x}, t, u) & (\mathbf{x}, t) \in \mathcal{X} \times [0, \infty), \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}) & \mathbf{x} \in \mathcal{X}, \end{cases} \tag{PDE}$$

  where $f : \mathcal{X} \times [0, \infty) \times \mathcal{U} \to \mathbb{R}$ is the source term and $u_0 : \mathcal{X} \to \mathbb{R}$ is the initial condition.

- Nonlinear parametrization $\hat{u} : \mathcal{X} \times \Theta \to \mathbb{R}$ with parameters $\theta = \theta(t) \in \Theta$:

$$u(\mathbf{x}, t) = \hat{u}(\mathbf{x}, \theta(t)) \qquad \forall (\mathbf{x}, t) \in \mathcal{X} \times [0, \infty).$$

- Local-in-time residual $r_t : \mathcal{X} \times \Theta \times \dot{\Theta} \to \mathbb{R}$:

$$r_t(\mathbf{x}, \theta, \dot{\theta}) = \nabla_\theta \hat{u}(\mathbf{x}, \theta) \cdot \dot{\theta} - f(\mathbf{x}, t, \hat{u}(\theta)). \tag{RES}$$

Problem formulation

- Time-dependent function $u \in \mathcal{U}$, $u : \mathcal{X} \times [0, \infty) \to \mathbb{R}$ characterized by

$$\begin{cases} \partial_t u(\mathbf{x}, t) = f(\mathbf{x}, t, u) & (\mathbf{x}, t) \in \mathcal{X} \times [0, \infty), \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}) & \mathbf{x} \in \mathcal{X}, \end{cases} \tag{PDE}$$

  where $f : \mathcal{X} \times [0, \infty) \times \mathcal{U} \to \mathbb{R}$ is the source term and $u_0 : \mathcal{X} \to \mathbb{R}$ is the initial condition.

- Nonlinear parametrization $\hat{u} : \mathcal{X} \times \Theta \to \mathbb{R}$ with parameters $\theta = \theta(t) \in \Theta$:

$$u(\mathbf{x}, t) = \hat{u}(\mathbf{x}, \theta(t)) \qquad \forall (\mathbf{x}, t) \in \mathcal{X} \times [0, \infty).$$

- Local-in-time residual $r_t : \mathcal{X} \times \Theta \times \dot{\Theta} \to \mathbb{R}$:

$$r_t(\mathbf{x}, \theta, \dot{\theta}) = \nabla_\theta \hat{u}(\mathbf{x}, \theta) \cdot \dot{\theta} - f(\mathbf{x}, t, \hat{u}(\theta)). \tag{RES}$$

Optimization problem

We approximate the solution of (PDE) by solving the optimization problem

$$\dot{\theta} \in \arg\min_{\eta \in \dot{\Theta}} J_t(\theta, \eta), \tag{MIN}$$

where the objective function $J_t : \Theta \times \dot{\Theta} \to \mathbb{R}$ is defined as:

$$J_t(\theta, \eta) = \frac{1}{2} \int_{\mathcal{X}} |r_t(\mathbf{x}, \theta, \eta)|^2 \, \mathrm{d}\mu_t(\mathbf{x}). \tag{OBJ}$$

In (OBJ), μ_t is a positive measure with support on $\mathcal{X}$.

- Static measure: $\mu_t = \mu$ (e.g., uniform distribution over $\mathcal{X}$)
- Adaptive measure

Optimization problem

We approximate the solution of (PDE) by solving the optimization problem

$$\dot{\theta} \in \underset{\eta \in \dot{\Theta}}{\arg\min}\, J_t(\theta, \eta), \tag{MIN}$$

where the objective function $J_t : \Theta \times \dot{\Theta} \to \mathbb{R}$ is defined as:

$$J_t(\theta, \eta) = \frac{1}{2} \int_{\mathcal{X}} |r_t(\mathbf{x}, \theta, \eta)|^2 \, \mathrm{d}\mu_t(\mathbf{x}). \tag{OBJ}$$

In (OBJ), μ_t is a positive measure with support on $\mathcal{X}$.

- Static measure: $\mu_t = \mu$ (e.g., uniform distribution over $\mathcal{X}$)
- Adaptive measure

Optimization problem

We approximate the solution of (PDE) by solving the optimization problem

$$\dot{\theta} \in \operatorname*{arg\,min}_{\eta \in \dot{\Theta}} J_t(\theta, \eta), \tag{MIN}$$

where the objective function $J_t : \Theta \times \dot{\Theta} \to \mathbb{R}$ is defined as:

$$J_t(\theta, \eta) = \frac{1}{2} \int_{\mathcal{X}} |r_t(\mathbf{x}, \theta, \eta)|^2 \, \mathrm{d}\mu_t(\mathbf{x}). \tag{OBJ}$$

In (OBJ), μ_t is a positive measure with support on $\mathcal{X}$.

- Static measure: $\mu_t = \mu$ (e.g., uniform distribution over $\mathcal{X}$)
- Adaptive measure

System of ODEs

From ($\mathtt{MIN}$), we can derive the system of ODEs

$$\begin{cases} M(\theta)\dot{\theta} = F(t,\theta), \\ \theta(0) = \theta_0, \end{cases} \tag{SYS}$$

$$M(\theta) = \int_{\mathcal{X}} \nabla_\theta \hat{u}(\mathbf{x},\theta) \otimes \nabla_\theta \hat{u}(\mathbf{x},\theta)\, \mathrm{d}\mu_t(\mathbf{x}), \quad F(t,\theta) = \int_{\mathcal{X}} \nabla_\theta \hat{u}(\mathbf{x},\theta) f(\mathbf{x}, t, \hat{u}(\mathbf{x},\theta))\, \mathrm{d}\mu_t(\mathbf{x}).$$

In practice, we draw a set of samples $\{\mathbf{x}_i^t\}_{i=1}^n$ from μ_t to assemble the Monte Carlo estimators

$$\mathbf{M} = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \hat{u}(\mathbf{x}_i^t,\theta) \otimes \nabla_\theta \hat{u}(\mathbf{x}_i^t,\theta), \quad \mathbf{F} = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \hat{u}(\mathbf{x}_i^t,\theta) f(\mathbf{x}_i^t, t, \hat{u}(\mathbf{x}_i^t,\theta)),$$

and we solve ($\mathtt{SYS}$) using a numerical integrator (e.g., Runge-Kutta-Fehlberg[3]).

[3] Ernst Hairer, *Solving Ordinary Differential Equations I*

System of ODEs

From (MIN), we can derive the system of ODEs

$$\begin{cases} M(\theta)\dot{\theta} = F(t,\theta), \\ \theta(0) = \theta_0, \end{cases} \tag{SYS}$$

$$M(\theta) = \int_{\mathcal{X}} \nabla_\theta \hat{u}(\mathbf{x},\theta) \otimes \nabla_\theta \hat{u}(\mathbf{x},\theta)\ \mathrm{d}\mu_t(\mathbf{x}), \quad F(t,\theta) = \int_{\mathcal{X}} \nabla_\theta \hat{u}(\mathbf{x},\theta) f(\mathbf{x}, t, \hat{u}(\mathbf{x},\theta))\ \mathrm{d}\mu_t(\mathbf{x}).$$

In practice, we draw a set of samples $\{\mathbf{x}_i^t\}_{i=1}^n$ from μ_t to assemble the Monte Carlo estimators

$$\mathbf{M} = \frac{1}{n}\sum_{i=1}^n \nabla_\theta \hat{u}(\mathbf{x}_i^t,\theta) \otimes \nabla_\theta \hat{u}(\mathbf{x}_i^t,\theta), \quad \mathbf{F} = \frac{1}{n}\sum_{i=1}^n \nabla_\theta \hat{u}(\mathbf{x}_i^t,\theta) f(\mathbf{x}_i^t, t, \hat{u}(\mathbf{x}_i^t,\theta)),$$

and we solve (SYS) using a numerical integrator (e.g., Runge-Kutta-Fehlberg[3]).

[3] Ernst Hairer, *Solving Ordinary Differential Equations I*

Test cases – Korteweg-de Vries (KdV) equation

1D KdV equation

$$\partial_t u + \partial_x^3 u + 6u\partial_x u = 0, \tag{KDV}$$

where $u = u(x,t)$, $x \in \mathcal{X} = [-20, 40]$ with periodic boundary conditions, $t \in [0,4]$.

Nonlinear parametrization:

$$\hat{u}(x,\theta) = \sum_{i=1}^{m} c_i \phi_G^L(x, w_i, b_i),$$

where $\phi_G^L(x,w,b)$ is a periodic unit (with period $L = |\mathcal{X}|$).

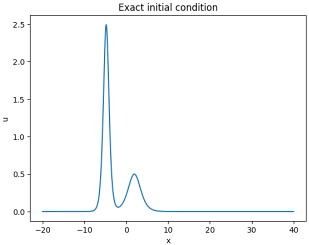


Figure: KdV – $u(x,0)$

Test cases – Allen-Cahn (AC) equation

1D AC equation

$$\partial_t u = \epsilon \partial_x^2 u + a(x,t)(u - u^3), \tag{AC}$$

where $u = u(x,t)$, $x \in \mathcal{X} = [0, 2\pi)$ with periodic boundary conditions, $t \in [0, 12]$, $\epsilon = 5 \cdot 10^{-2}$, $a(x,t) = 1.05 + t\sin(x)$.

Nonlinear parametrization:

$$\hat{u}(x,\theta) = \mathbf{w}_l^T \tanh(\mathbf{W}_{l-1} \tanh(\ldots \mathbf{W}_1(\tanh(\Psi(x)))\ldots) + \mathbf{p}_{l-1}),$$

where $\Psi(x) = (\psi(x, a_k, b_k, c_k))_{k=1}^m$ and $\psi(x,a,b,c)$ is a periodic unit (with period $L = |\mathcal{X}|$).

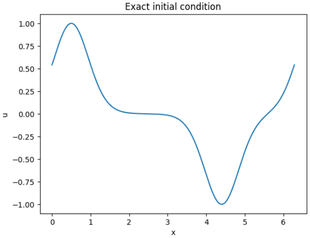
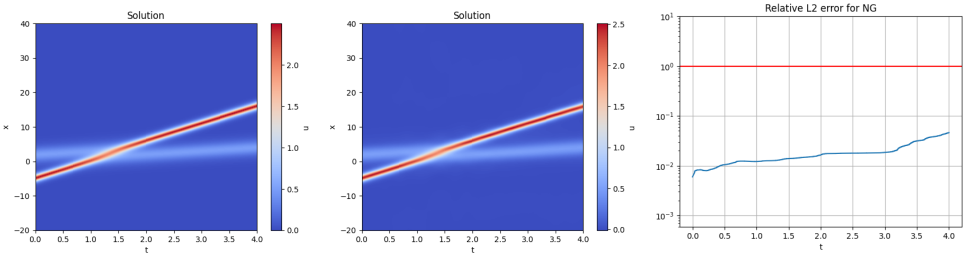


Figure: AC – $u(x, 0)$

Numerical experiments – KdV with static sampling



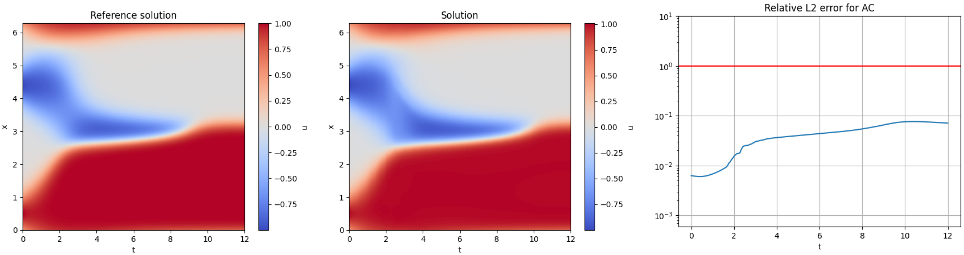
(a) KdV – Reference solution

(b) KdV – Neural Galerkin solution

(c) KdV – Relative ℓ^2 error

Figure: KdV – Results for Neural Galerkin with static sampling from a uniform distribution over $\mathcal{X}$, $n = 1000$ samples.

Numerical experiments – AC with static sampling



(a) AC – Reference solution

(b) AC – Neural Galerkin solution

(c) AC – Relative ℓ^2 error

Figure: AC – Results for Neural Galerkin with static sampling from a uniform distribution over $\mathcal{X}$, $n = 1000$ samples.

Adaptive measure

- Time-dependent Gibbs measure[4]:

$$\mu_t^G(d\mathbf{x}) = Z^{-1}_{\theta(t),\dot{\theta}(t)} \exp\left(-V_{\theta(t),\dot{\theta}(t)}(\mathbf{x})\right) d\mathbf{x}, \tag{GIB}$$

where $V_{\theta(t),\dot{\theta}(t)} : \mathcal{X} \to \mathbb{R}$ is a potential and $Z_{\theta(t),\dot{\theta}(t)} \in \mathbb{R}$ is the normalization constant.

- Adaptive measure as a function of the PDE residual (RES)[5]:

$$\mu_t^G(d\mathbf{x}) \propto \left| r_t(\mathbf{x}, \theta(t), \dot{\theta}(t)) \right|^{2\gamma} \nu(d\mathbf{x})^{\gamma}, \tag{ADP}$$

where $\gamma > 0$ is a tempering parameter and ν is a static distribution with support on $\mathcal{X}$.

[4] Pavliotis, *Stochastic processes and applications*

[5] Wen, Vanden-Eijnden, and Peherstorfer, "Coupling parameter and particle dynamics for adaptive sampling in Neural Galerkin schemes"

Adaptive measure

- Time-dependent Gibbs measure[4]:

$$\mu_t^G(d\mathbf{x}) = Z^{-1}_{\theta(t),\dot\theta(t)} \exp\left(-V_{\theta(t),\dot\theta(t)}(\mathbf{x})\right) d\mathbf{x}, \tag{GIB}$$

where $V_{\theta(t),\dot\theta(t)} : \mathcal{X} \to \mathbb{R}$ is a potential and $Z_{\theta(t),\dot\theta(t)} \in \mathbb{R}$ is the normalization constant.

- Adaptive measure as a function of the PDE residual (RES)[5]:

$$\mu_t^G(d\mathbf{x}) \propto \left| r_t(\mathbf{x}, \theta(t), \dot\theta(t)) \right|^{2\gamma} \nu(d\mathbf{x})^\gamma, \tag{ADP}$$

where $\gamma > 0$ is a tempering parameter and ν is a static distribution with support on $\mathcal{X}$.

[4] Pavliotis, *Stochastic processes and applications*

[5] Wen, Vanden-Eijnden, and Peherstorfer, "Coupling parameter and particle dynamics for adaptive sampling in Neural Galerkin schemes"

Particle dynamics for adaptive sampling

The dynamics of a set of particles $\{\mathbf{x}_i(t)\}_{i=1}^n$ can be described by the Langevin SDE:

$$d\mathbf{x}_i(t) = -\alpha \nabla V_{\theta(t),\dot{\theta}(t)} \mathbf{x}_i(t) dt + \sqrt{2\alpha} dW_i(t), \tag{LAN}$$

where $V_{\theta(t),\dot{\theta}(t)}$ is a potential, $\{W_i(t)\}_{i=1}^n$ are i.i.d. Wiener processes in $\mathbb{R}^d$, and $\alpha > 0$.

The Fokker-Planck equation associated with (LAN) is:

$$\partial_t \mu_t = \alpha \nabla \cdot \left(\nabla \mu_t + \mu_t \nabla V_{\theta(t),\dot{\theta}(t)} \right).$$

Under suitable assumptions on $V_{\theta(t),\dot{\theta}(t)}$, μ_t converges to the Gibbs measure (GIB)[6].

[6] Pavliotis, *Stochastic processes and applications*

Particle dynamics for adaptive sampling

The dynamics of a set of particles $\{\mathbf{x}_i(t)\}_{i=1}^n$ can be described by the Langevin SDE:

$$d\mathbf{x}_i(t) = -\alpha \nabla V_{\theta(t),\dot{\theta}(t)} \mathbf{x}_i(t) dt + \sqrt{2\alpha} dW_i(t), \tag{LAN}$$

where $V_{\theta(t),\dot{\theta}(t)}$ is a potential, $\{W_i(t)\}_{i=1}^n$ are i.i.d. Wiener processes in $\mathbb{R}^d$, and $\alpha > 0$.

The Fokker-Planck equation associated with (LAN) is:

$$\partial_t \mu_t = \alpha \nabla \cdot \left(\nabla \mu_t + \mu_t \nabla V_{\theta(t),\dot{\theta}(t)} \right).$$

Under suitable assumptions on $V_{\theta(t),\dot{\theta}(t)}$, μ_t converges to the Gibbs measure (GIB)[6].

[6] Pavliotis, *Stochastic processes and applications*

Stein Variational Gradient Descent[8]

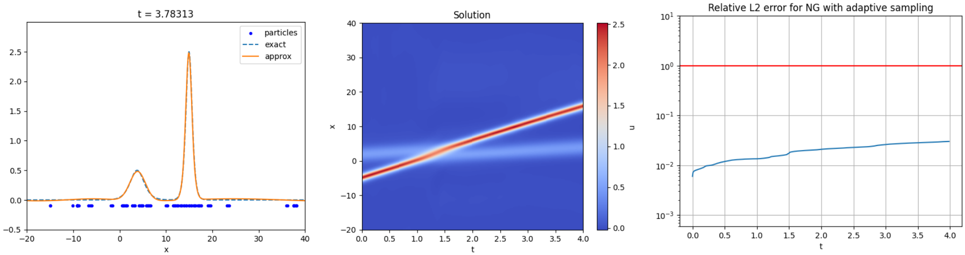
```
Algorithm 1: Stein Variational Gradient Descent (SVGD)

Input: Target distribution $p$, kernel function $k(\cdot,\cdot)$, initial particles $\{\mathbf{x}_i^0\}_{i=1}^n$
Output: Final particles $\{\mathbf{x}_i^L\}_{i=1}^n$ that approximate the target distribution $p$
for $l = 1, \dots, L$ do
    $\mathbf{x}_i^{l+1} \leftarrow \mathbf{x}_i^l + \epsilon_l \left( \frac{1}{n} \sum_{j=1}^{n} \left[ k(\mathbf{x}_j^l, \mathbf{x}_i^l) \nabla_{\mathbf{x}_j^l} \log p(\mathbf{x}_j^l) + \nabla_{\mathbf{x}_j^l} k(\mathbf{x}_j^l, \mathbf{x}_i^l) \right] \right)$
end
```

Alternatively, we can consider an SVGD variant with the addition of a noise term[7] $\eta^l \sim \mathcal{N}(\mathbf{0}, 2\epsilon_l \mathbf{D}/n)$, where $\mathbf{D}_{i,j}(\mathbf{x}^l) = k(\mathbf{x}_i^l, \mathbf{x}_j^l)$.

[7] Gallego and Insua, "Stochastic gradient MCMC with repulsive forces"

[8] Liu and Wang, "Stein variational gradient descent: A general purpose bayesian inference algorithm"

Numerical experiments – KdV with adaptive sampling (SVGD)



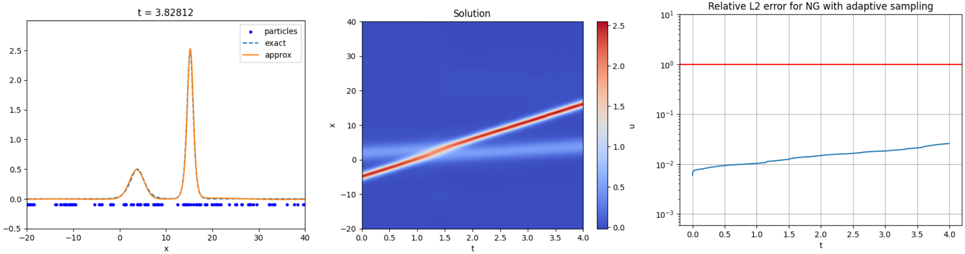
(a) KdV – Solution at final time

(b) KdV – Space-time solution

(c) KdV – Relative ℓ^2 error

Figure: KdV – Results for Neural Galerkin with adaptive sampling (SVGD) from the residual-dependent measure (ADP), $n = 100$ samples.

Numerical experiments – KdV with adaptive sampling (SVGD with noise)



(a) KdV – Solution at final time

(b) KdV – Space-time solution

(c) KdV – Relative ℓ^2 error

Figure: KdV – Results for Neural Galerkin with adaptive sampling (SVGD with noise) from the residual-dependent measure (ADP), $n = 100$ samples.

Conditioning

The system of ODEs (SYS) is typically ill-conditioned. The condition number of the mass matrix $\kappa(\mathbf{M})$ tends to increase with the size of the neural network (i.e., the dimension of θ).

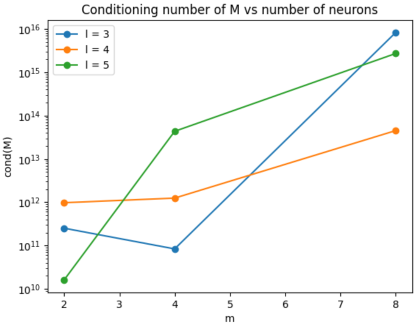


Figure: AC – Trend of the condition number of the mass matrix as a function of the number of layers *l* and number of neurons per layer *m*

Optimal sampling outline

Goal

Define a sampling measure leading to a better-conditioned problem.

Recall the optimization problem

$$\dot{\theta} \in \arg\min_{\eta \in \dot{\Theta}} J_t(\theta, \eta), \tag{MIN}$$

where the objective function J_t is defined as:

$$J_t(\theta, \eta) = \frac{1}{2} \int_{\mathcal{X}} |r_t(\mathbf{x}, \theta, \eta)|^2 \, \mathrm{d}\mu_t(\mathbf{x}).$$

1. Reinterpret (MIN) as a least squares problem in a *tangent space*
2. Generalize (MIN) via weighted least squares
3. Define a weighted sampling measure and exploit a theoretical bound on the conditioning

Optimal sampling outline

Goal

Define a sampling measure leading to a better-conditioned problem.

Recall the optimization problem

$$\dot{\theta} \in \underset{\eta \in \dot{\Theta}}{\arg\min}\, J_t(\theta, \eta), \tag{MIN}$$

where the objective function J_t is defined as:

$$J_t(\theta, \eta) = \frac{1}{2} \int_{\mathcal{X}} |r_t(\mathbf{x}, \theta, \eta)|^2 \, \mathrm{d}\mu_t(\mathbf{x}).$$

1. Reinterpret (MIN) as a least squares problem in a *tangent space*
2. Generalize (MIN) via weighted least squares
3. Define a weighted sampling measure and exploit a theoretical bound on the conditioning

Geometric interpretation of Neural Galerkin

- Manifold induced by the nonlinear parametrization:

$$\mathcal{M}_\Theta = \{\hat{u}(\cdot,\theta) | \theta \in \Theta\}.$$

- Tangent space of $\mathcal{M}_\Theta$ at a point $\hat{u}(\mathbf{x},\theta)$:

$$\mathcal{T}_\theta = \text{span}\{\partial_{\theta_1}\hat{u}(\mathbf{x},\theta), ..., \partial_{\theta_p}\hat{u}(\mathbf{x},\theta)\}.$$

For now, let $\dim(\mathcal{T}_\theta) = p$ (this is not true in general → collapsing tangent space[9]).

The minimization problem (`MIN`) can be rewritten as:

$$\min_{\dot{\theta}\in\mathbb{R}^p} \| \underbrace{\nabla_\theta \hat{u}(\cdot,\theta)\dot{\theta}}_{\in\mathcal{T}_\theta} - f(\cdot,\hat{u}(\cdot,\theta))\|^2_{L^2(\mathcal{X},d\mu_t)}.$$

[9] Zhang et al., "Sequential-in-time training of nonlinear parametrizations for solving time-dependent partial differential equations"

Geometric interpretation of Neural Galerkin

- Manifold induced by the nonlinear parametrization:

$$\mathcal{M}_\Theta = \{\hat{u}(\cdot,\theta) | \theta \in \Theta\}.$$

- Tangent space of $\mathcal{M}_\Theta$ at a point $\hat{u}(\mathbf{x},\theta)$:

$$\mathcal{T}_\theta = \mathrm{span}\{\partial_{\theta_1}\hat{u}(\mathbf{x},\theta), \dots, \partial_{\theta_p}\hat{u}(\mathbf{x},\theta)\}.$$

For now, let $\dim(\mathcal{T}_\theta) = p$ (this is not true in general → collapsing tangent space[9]).

The minimization problem (`MIN`) can be rewritten as:

$$\min_{\dot{\theta}\in\mathbb{R}^p} \| \underbrace{\nabla_\theta \hat{u}(\cdot,\theta)\dot{\theta}}_{\in \mathcal{T}_\theta} - f(\cdot,\hat{u}(\cdot,\theta)) \|^2_{L^2(\mathcal{X},d\mu_t)}.$$

[9] Zhang et al., "Sequential-in-time training of nonlinear parametrizations for solving time-dependent partial differential equations"

Geometric interpretation of Neural Galerkin

- Manifold induced by the nonlinear parametrization:

$$\mathcal{M}_\Theta = \{\hat{u}(\cdot,\theta) | \theta \in \Theta\}.$$

- Tangent space of $\mathcal{M}_\Theta$ at a point $\hat{u}(\mathbf{x},\theta)$:

$$\mathcal{T}_\theta = \text{span}\{\partial_{\theta_1}\hat{u}(\mathbf{x},\theta), ..., \partial_{\theta_p}\hat{u}(\mathbf{x},\theta)\}.$$

For now, let $\dim(\mathcal{T}_\theta) = p$ (this is not true in general → collapsing tangent space[9]).

The minimization problem (MIN) can be rewritten as:

$$\min_{\dot{\theta}\in\mathbb{R}^p} \| \underbrace{\nabla_\theta \hat{u}(\cdot,\theta)\dot{\theta}}_{\in\mathcal{T}_\theta} - f(\cdot, \hat{u}(\cdot,\theta))\|^2_{L^2(\mathcal{X}, d\mu_t)}.$$

[9] Zhang et al., "Sequential-in-time training of nonlinear parametrizations for solving time-dependent partial differential equations"

Weighted least squares

If we draw a set of samples $\{\mathbf{x}_i\}_{i=1}^n$ from μ_t, we can assemble:

$$\min_{\dot{\theta}\in\mathbb{R}^p} \|\mathbf{J}(\theta)\dot{\theta} - \mathbf{f}(\theta)\|_2^2. \tag{LS}$$

The normal equations associated to (LS) are:

$$\mathbf{J}^T\mathbf{J}\dot{\theta} = \mathbf{J}^T\mathbf{f},$$

where $\mathbf{J}^T\mathbf{J}$ and $\mathbf{J}^T\mathbf{f}$ are the Monte Carlo estimators $\mathbf{M}$ and $\mathbf{F}$, respectively.

We can generalize (LS) via weighted least squares:

$$\min_{\dot{\theta}\in\mathbb{R}^p} |\mathbf{J}(\theta)\dot{\theta} - \mathbf{f}(\theta)|_w^2,$$

$$|\mathbf{v}|_w^2 = \frac{1}{n}\sum_{i=1}^n w(\mathbf{x}_i)\,|v(\mathbf{x}_i)|^2, \qquad \mathbf{v} := [v(\mathbf{x}_1), ..., v(\mathbf{x}_n)]^T \in \mathbb{R}^n,$$

where $w : \mathcal{X} \to \mathbb{R}$ is a weight function.

Weighted least squares

If we draw a set of samples $\{\mathbf{x}_i\}_{i=1}^n$ from μ_t, we can assemble:

$$\min_{\dot{\theta}\in\mathbb{R}^p} \|\mathbf{J}(\theta)\dot{\theta} - \mathbf{f}(\theta)\|_2^2. \tag{LS}$$

The normal equations associated to (LS) are:

$$\mathbf{J}^T\mathbf{J}\dot{\theta} = \mathbf{J}^T\mathbf{f},$$

where $\mathbf{J}^T\mathbf{J}$ and $\mathbf{J}^T\mathbf{f}$ are the Monte Carlo estimators $\mathbf{M}$ and $\mathbf{F}$, respectively.

We can generalize (LS) via weighted least squares:

$$\min_{\dot{\theta}\in\mathbb{R}^p} |\mathbf{J}(\theta)\dot{\theta} - \mathbf{f}(\theta)|_w^2,$$

$$|\mathbf{v}|_w^2 = \frac{1}{n}\sum_{i=1}^n w(\mathbf{x}_i)\,|v(\mathbf{x}_i)|^2, \qquad \mathbf{v} := [v(\mathbf{x}_1), \ldots, v(\mathbf{x}_n)]^T \in \mathbb{R}^n,$$

where $w : \mathcal{X} \to \mathbb{R}$ is a weight function.

Optimal sampling

$$\text{Arbitrary basis } \{\partial_{\theta_i}\hat{u}\}_{i=1}^{p} \text{ of } \mathcal{T}_\theta \quad \rightarrow \quad \text{Orthonormal basis } \{L_i\}_{i=1}^{p} \text{ of } \mathcal{T}_\theta$$

$$\min_{\dot{\theta}\in\mathbb{R}^p} |\mathbf{J}(\theta)\dot{\theta} - \mathbf{f}(\theta)|_w^2 \quad \rightarrow \quad \min_{\tau\in\mathbb{R}^p} |\Lambda(\theta)\tau - \mathbf{f}(\theta)|_w^2 \tag{WLS}$$

We denote the normal equations associated with (WLS) as:

$$\mathbf{G}\tau = \mathbf{d}.$$

The condition number of $\mathbf{G}$ can be controlled with high probability[10] if the points $\{\mathbf{x}_i\}_{i=1}^{n}$ are sampled from a measure $d\mu_{t,\mathrm{opt}}$ such that $w \; d\mu_{t,\mathrm{opt}} = d\mu_t$, where

$$w(\mathbf{x}) = \frac{p}{\sum_{j=1}^{p} |L_j(\mathbf{x},\theta)|^2}.$$

[10] Cohen and Migliorati, "Optimal weighted least-squares methods"

Optimal sampling

$$\text{Arbitrary basis } \{\partial_{\theta_i}\hat{u}\}_{i=1}^{p} \text{ of } \mathcal{T}_\theta \qquad \rightarrow \qquad \text{Orthonormal basis } \{L_i\}_{i=1}^{p} \text{ of } \mathcal{T}_\theta$$

$$\min_{\dot{\theta}\in\mathbb{R}^p} |\mathbf{J}(\theta)\dot{\theta} - \mathbf{f}(\theta)|_w^2 \qquad \rightarrow \qquad \min_{\tau\in\mathbb{R}^p} |\Lambda(\theta)\tau - \mathbf{f}(\theta)|_w^2 \tag{WLS}$$

We denote the normal equations associated with (WLS) as:

$$\mathbf{G}\tau = \mathbf{d}.$$

The condition number of **G** can be controlled with high probability[10] if the points $\{\mathbf{x}_i\}_{i=1}^{n}$ are sampled from a measure $d\mu_{t,\mathrm{opt}}$ such that $w\ d\mu_{t,\mathrm{opt}} = d\mu_t$, where

$$w(\mathbf{x}) = \frac{p}{\sum_{j=1}^{p} |L_j(\mathbf{x},\theta)|^2}.$$

[10] Cohen and Migliorati, "Optimal weighted least-squares methods"

Two-step sampling strategy

For the tangent space $\mathcal{T}_\theta$ in Neural Galerkin, no exact orthonormal basis is available a priori.

We can follow a two-step sampling strategy[11]:

1. Sample $\{\mathbf{z}_i\}_{i=1}^m$ from $d\mu_t$ to compute an (approximate) orthonormal basis $\{L_i\}_{i=1}^p$, e.g., by orthogonalizing $\{\partial_{\theta_i}\hat{u}\}_{i=1}^p$ with the Gram-Schmidt algorithm;
2. Sample $\{\mathbf{x}_i\}_{i=1}^n$ from $d\mu_{t,\mathrm{opt}}$ to solve the weighted least squares problem (WLS).

In general, $\{\partial_{\theta_i}\hat{u}\}_{i=1}^p$ are *not* linearly independent (i.e., $\dim(\mathcal{T}_\theta) < p$).

Approximating the orthonormal basis with Gram-Schmidt introduces numerical errors and extra computational costs.

[11] Dolbeault and Cohen, "Optimal sampling and Christoffel functions on general domains"

Two-step sampling strategy

For the tangent space $\mathcal{T}_\theta$ in Neural Galerkin, no exact orthonormal basis is available a priori.

We can follow a two-step sampling strategy[11]:

1. Sample $\{\mathbf{z}_i\}_{i=1}^m$ from $d\mu_t$ to compute an (approximate) orthonormal basis $\{L_i\}_{i=1}^p$, e.g., by orthogonalizing $\{\partial_{\theta_i}\hat{u}\}_{i=1}^p$ with the Gram-Schmidt algorithm;
2. Sample $\{\mathbf{x}_i\}_{i=1}^n$ from $d\mu_{t,\mathrm{opt}}$ to solve the weighted least squares problem (`WLS`).

In general, $\{\partial_{\theta_i}\hat{u}\}_{i=1}^p$ are *not* linearly independent (i.e., $\dim(\mathcal{T}_\theta) < p$).

Approximating the orthonormal basis with Gram-Schmidt introduces numerical errors and extra computational costs.

[11] Dolbeault and Cohen, "Optimal sampling and Christoffel functions on general domains"

Two-step sampling strategy

For the tangent space $\mathcal{T}_\theta$ in Neural Galerkin, no exact orthonormal basis is available a priori.

We can follow a two-step sampling strategy[11]:

1. Sample $\{\mathbf{z}_i\}_{i=1}^m$ from $d\mu_t$ to compute an (approximate) orthonormal basis $\{L_i\}_{i=1}^p$, e.g., by orthogonalizing $\{\partial_{\theta_i}\hat{u}\}_{i=1}^p$ with the Gram-Schmidt algorithm;
2. Sample $\{\mathbf{x}_i\}_{i=1}^n$ from $d\mu_{t,\mathrm{opt}}$ to solve the weighted least squares problem (`WLS`).

In general, $\{\partial_{\theta_i}\hat{u}\}_{i=1}^p$ are *not* linearly independent (i.e., $\dim(\mathcal{T}_\theta) < p$).

Approximating the orthonormal basis with Gram-Schmidt introduces numerical errors and extra computational costs.

[11] Dolbeault and Cohen, “Optimal sampling and Christoffel functions on general domains”

Regularized least squares

Alternatively, we can reduce the conditioning of $(\mathtt{LS})$ by solving the ridge regression problem:

$$\min_{\dot{\theta}\in\mathbb{R}^p} \|\mathbf{J}(\theta)\dot{\theta} - \mathbf{f}(\theta)\|_2^2 + \lambda\|\dot{\theta}\|_2^2,$$

where $\lambda > 0$ is the regularization parameter. The associated normal equations are:

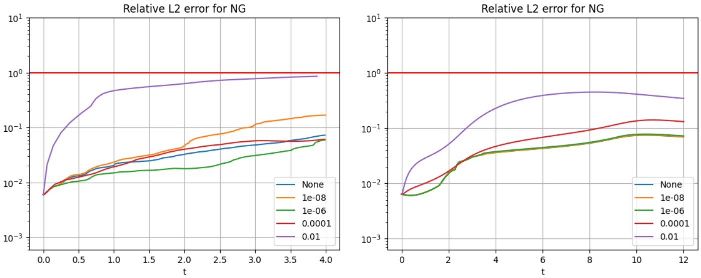
$$(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I}_{p\times p})\dot{\theta} = \mathbf{J}^T\mathbf{f}.$$

Since $\mathbf{M}$ is symmetric, $\kappa(\mathbf{M}) = |\lambda_{\max}/\lambda_{\min}|$, hence adding the regularization term leads to

$$\kappa(\mathbf{M} + \lambda\mathbf{I}_{p\times p}) = \left|\frac{\lambda_{\max} + \lambda}{\lambda_{\min} + \lambda}\right|.$$

Numerical experiments – Regularized least squares

- KdV: $\lambda = 0$, $\kappa(\mathbf{M}) \approx 10^{14} \rightarrow \lambda = 10^{-6}$, $\kappa(\mathbf{M}) \approx 10^{6}$
- AC: $\lambda = 0$, $\kappa(\mathbf{M}) \approx 10^{10} \rightarrow \lambda = 10^{-6}$, $\kappa(\mathbf{M}) \approx 10^{6}$



(a) KdV

(b) AC

Figure: Relative ℓ^2 error with regularization

Conclusion

1. The Neural Galerkin method coupled with adaptive sampling can well represent the dynamics of advection-dominated problems
   → Scalable to higher dimensions

2. The implementation of the optimal sampling strategy can become too expensive and affected by numerical errors if an orthonormal basis of the tangent space $\mathcal{T}_\theta$ is not known
   → Need theoretical results on the conditioning of the least squares problem in the case of an arbitrary (non-orthonormal) basis

3. The collapsing tangent space phenomenon may lead to a loss of representation power
   → Experiment with other classes of architectures

Conclusion

1. The Neural Galerkin method coupled with adaptive sampling can well represent the dynamics of advection-dominated problems
   → Scalable to higher dimensions

2. The implementation of the optimal sampling strategy can become too expensive and affected by numerical errors if an orthonormal basis of the tangent space $\mathcal{T}_\theta$ is not known
   → Need theoretical results on the conditioning of the least squares problem in the case of an arbitrary (non-orthonormal) basis

3. The collapsing tangent space phenomenon may lead to a loss of representation power
   → Experiment with other classes of architectures

Conclusion

1. The Neural Galerkin method coupled with adaptive sampling can well represent the dynamics of advection-dominated problems
   $\rightarrow$ Scalable to higher dimensions

2. The implementation of the optimal sampling strategy can become too expensive and affected by numerical errors if an orthonormal basis of the tangent space $\mathcal{T}_\theta$ is not known
   $\rightarrow$ Need theoretical results on the conditioning of the least squares problem in the case of an arbitrary (non-orthonormal) basis

3. The collapsing tangent space phenomenon may lead to a loss of representation power
   $\rightarrow$ Experiment with other classes of architectures

References

Bruna, Joan, Benjamin Peherstorfer, and Eric Vanden-Eijnden. "Neural Galerkin schemes with active learning for high-dimensional evolution equations". In: *Journal of Computational Physics* 496 (2024), p. 112588. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2023.112588.

Cohen, Albert and Giovanni Migliorati. "Optimal weighted least-squares methods". en. In: *The SMAI Journal of computational mathematics* 3 (2017), pp. 181–203. DOI: 10.5802/smai-jcm.24.

Dolbeault, Matthieu and Albert Cohen. "Optimal sampling and Christoffel functions on general domains". In: *Constructive Approximation* 56.1 (2022), pp. 121–163.

Duncan, Andrew, Nikolas Nüsken, and Lukasz Szpruch. "On the geometry of Stein variational gradient descent". In: *arXiv preprint arXiv:1912.00894* (2019).

Ernst Hairer Gerhard Wanner, Syvert P. Nørsett. *Solving Ordinary Differential Equations I*. Springer Berlin, Heidelberg, 1993.

Gallego, Victor and David Rios Insua. "Stochastic gradient MCMC with repulsive forces". In: *stat* 1050 (2018), p. 30.

Liu, Qiang and Dilin Wang. "Stein variational gradient descent: A general purpose bayesian inference algorithm". In: *Advances in neural information processing systems* 29 (2016).

Pavliotis, Grigorios A. *Stochastic processes and applications*. Vol. 60. Springer, 2014.

Peyré, Gabriel, Marco Cuturi, et al. "Computational optimal transport: With applications to data science". In: *Foundations and Trends in Machine Learning* 11.5-6 (2019), pp. 355–607.

Raissi, M., P. Perdikaris, and G.E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378 (2019), pp. 686–707. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.10.045.

Wen, Yuxiao, Eric Vanden-Eijnden, and Benjamin Peherstorfer. "Coupling parameter and particle dynamics for adaptive sampling in Neural Galerkin schemes". In: *Physica D: Nonlinear Phenomena* 462 (2024), p. 134129.

Zhang, Huan et al. "Sequential-in-time training of nonlinear parametrizations for solving time-dependent partial differential equations". In: *arXiv preprint arXiv:2404.01145* (2024).

Extra slides

Test cases – Training details

1. KdV
   - Gaussian periodic unit:

$$\phi_G^L(x, w, b) = \exp\left(-w^2 \left|\sin\left(\frac{\pi(x-b)}{L}\right)\right|^2\right),$$

   where $w, b \in \mathbb{R}$.
   - Network parameters: $m = 10$.
   - Initial fit: batch size $n_0 = 5000$, 10^4 epochs, Adam optimizer with initial learning rate $\gamma = 0.1$ combined with an exponential scheduler (decay rate 0.9) for the first 10^3 epochs.
2. AC
   - Periodic unit:

$$\psi(x, a, b, c) = a\cos\left(\frac{2\pi}{L}x + b\right) + c,$$

   where $a, b, c \in \mathbb{R}$.
   - Network parameters: $l = 3$, $m = 2$
   - Initial fit: batch size $n_0 = 1000$, 10^4 epochs, Adam optimized with initial learning rate $\gamma = 0.1$ combined with an exponential scheduler (decay rate 0.75) for the first 10^3 epochs.

Test cases – Relative ℓ^2 error

The relative ℓ^2 error is computed over $N = 2048$ equidistant grid points $x_1, ..., x_N$ in $\mathcal{X}$.

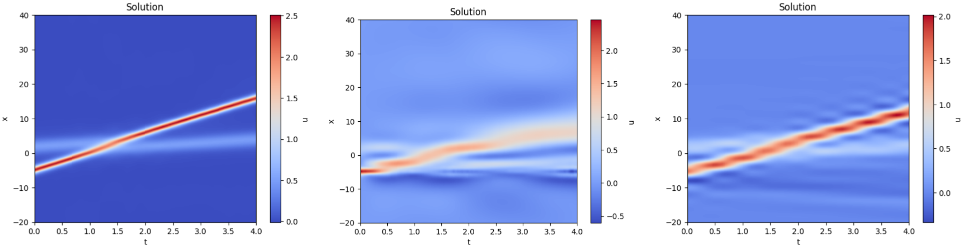
We define $\mathbf{u}(t) = [u(x_1, t), ..., u(x_N, t)]^T \in \mathbb{R}^N$ and $\hat{\mathbf{u}}(t) = [\hat{u}(x_1, t), ..., \hat{u}(x_N, t)]^T \in \mathbb{R}^N$ as the vectors of the exact solution and approximate solution at time t, respectively.

Then, given K points in time $t^1, ..., t^K$ determined adaptively by the Runge-Kutta-Fehlberg (RK45) method, we define the relative ℓ^2 error as:

$$e_{\ell^2} = \frac{\sum_{k=0}^{K} \|\hat{\mathbf{u}}(t^k) - \mathbf{u}(t^k)\|_2^2}{\sum_{k=0}^{K} \|\mathbf{u}(t^k)\|_2^2}.$$

Numerical experiments – KdV linear

1. Linear fitted, $m = 30$: after training the network on the initial condition, the parameters w_i and b_i in the Gaussian periodic unit are frozen before evolving the system in time.
2. Linear equidistant, $m = 30$: the parameters w_i and b_i are frozen *before* training the neural network on the initial condition; they are initialized so that the corresponding Gaussian units have the same shape and are equispaced in the spatial domain.



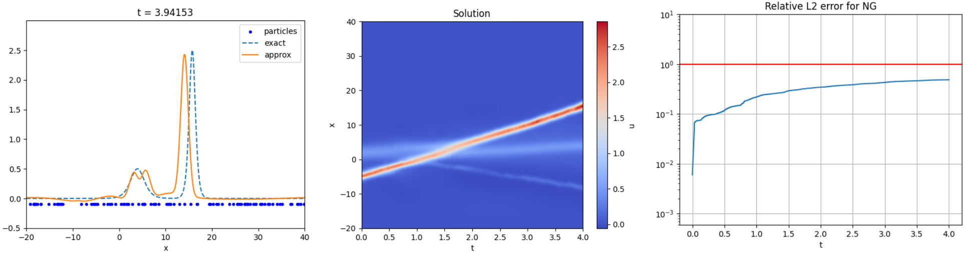
(a) KdV – Nonlinear Neural Galerkin

(b) KdV – Linear fitted Neural Galerkin

(c) KdV – Linear equidistant Neural Galerkin

Figure: KdV – Comparison between the linear and nonlinear settings

Numerical experiments – KdV with static sampling



(a) KdV – Solution at final time

(b) KdV – Space-time solution

(c) KdV – Relative ℓ^2 error

Figure: KdV – Results for Neural Galerkin with static sampling from a uniform distribution over $\mathcal{X}$, $n = 100$ samples. The number of samples is insufficient to accurately approximate the solution.

SVGD algorithm (1)

Given an intractable distribution p, the SVGD algorithm searches for $q^\star$ satisfying:

$$q^\star = \arg\min_{q \in Q} \mathrm{KL}(q||p),$$

where $\mathrm{KL}(q||p) = \mathbb{E}_{\mathbf{x}\sim q}[\log q(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim q}[\log p(\mathbf{x})]$.

Q is the set of distributions of random variables $\mathbf{z}$ that can be written as $\mathbf{z} = T(\mathbf{x})$, where $T(\mathbf{x}) = \mathbf{x} + \epsilon\phi(\mathbf{x})$ is a small perturbation of the identity map, and $\mathbf{x}$ is drawn from some tractable distribution q_0.

An explicit expression for the derivative of the KL divergence can be provided by exploiting a connection with the so-called Stein operator.

We also need to recall the definition of a reproducing kernel Hilbert space (RKHS):

$$\mathcal{H} = \left\{ f : f(\mathbf{x}) = \sum_{i=1}^{n} a_i k(\mathbf{x}, \mathbf{x}_i),\ a_i \in \mathbb{R},\ \mathbf{x}_i \in \mathcal{X} \right\}.$$

SVGD algorithm (2)

Theorem (Gradient of the KL divergence)

Let $T(\mathbf{x}) = \mathbf{x} + \epsilon\phi(\mathbf{x})$ and q_T the density of $\mathbf{z} = T(\mathbf{x})$ when $\mathbf{x} \sim q$. Then:

$$\nabla_\epsilon KL(q_T||p)|_{\epsilon=0} = -\mathbb{E}_{\mathbf{x}\sim q}[trace(\mathcal{A}_p\phi(\mathbf{x}))],$$

where $\mathcal{A}_p\phi(\mathbf{x}) := \nabla \log p(\mathbf{x})\phi(\mathbf{x})^T + \nabla\phi(\mathbf{x})$ is the Stein operator.

Theorem (Steepest descent direction)

We consider all the perturbation directions $\phi(\cdot)$ in the ball $\mathcal{B} = \{\phi \in \mathcal{H}^d : \|\phi\|_{\mathcal{H}^d} \leq \mathbb{D}(q,p)\}$, where $\mathcal{H}^d$ is the RKHS associated to the kernel $k(\cdot,\cdot)$, and $\mathbb{D}(q,p)$ is the kernelized Stein discrepancy

$$\mathbb{D}(q,p) = \max_{\phi\in\mathcal{H}^d} \{\mathbb{E}_{\mathbf{x}\sim q}[trace(\mathcal{A}_p\phi(\mathbf{x})] \; s.t. \; \|\phi\|_{\mathcal{H}_d} \leq 1\}.$$

Then, the steepest descent direction that minimizes the gradient of the KL divergence is

$$\phi^\star(\cdot) = \mathbb{E}_{\mathbf{x}\sim q}[k(\mathbf{x},\cdot)\nabla \log p(\mathbf{x}) + \nabla k(\mathbf{x},\cdot)].$$

SVGD and gradient flows

The Fokker-Planck equation associated with the Langevin SDE is related to the concept of *gradient flow*[12]. In particular, we can rewrite the Fokker-Planck as a continuity equation:

$$\partial_t \mu_t = \alpha \nabla \cdot \left[\left(\nabla \log \mu_t + \nabla V_{\theta(t),\dot{\theta}(t)} \right) \mu_t \right],$$

where $v_t := \left(\nabla \log \mu_t + \nabla V_{\theta(t),\dot{\theta}(t)} \right) \in \mathbb{R}^p$.

The SVGD algorithm approximates the gradient flow in a reproducing kernel Hilbert space (RKHS). In particular, if we set our target distribution as μ_t^G, we get $\nabla \log \mu_t^G = -\nabla V_{\theta(t),\dot{\theta}(t)}$, so we can write:

$$\partial_t \mu_t \approx \alpha \nabla \cdot \left(\mathbb{E}_{\mathbf{x}' \sim \mu_t} \left[k(\mathbf{x}', \mathbf{x}) \nabla V_{\theta(t),\dot{\theta}(t)}(\mathbf{x}') - \nabla_{\mathbf{x}'} k(\mathbf{x}', \mathbf{x}) \right] \mu_t \right).$$

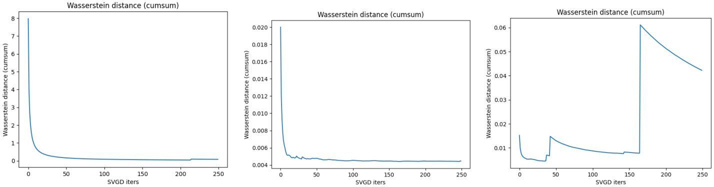
[12] Duncan, Nüsken, and Szpruch, "On the geometry of Stein variational gradient descent".

SVGD convergence monitoring

Wasserstein distance estimate based on the empirical measures $\hat{p}$, $\hat{q}$ (1D case)[13]:

$$W_p(\hat{p}, \hat{q}) = \left(\frac{1}{n} \sum_{i=1}^{n} |x^{(i)} - y^{(i)}|^p \right)^{1/p},$$

where $\{x_i\}_{i=1}^n \sim \hat{p}$ and $\{y_i\}_{i=1}^n \sim \hat{q}$, while $\{x^{(i)}\}_{i=1}^n$ and $\{y^{(i)}\}_{i=1}^n$ are the order statistics.



(a) Example 1

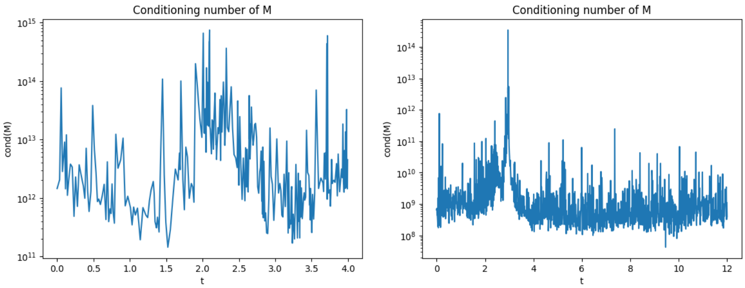
(b) Example 2

(c) Example 3

Figure: KdV – SVGD monitoring with the Wasserstein distance, $\epsilon = 0.05$, $L = 250$

[13] Peyré, Cuturi, et al., "Computational optimal transport: With applications to data science".

Conditioning (1)

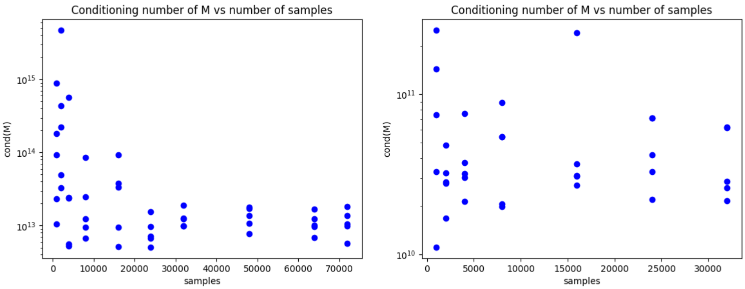


(a) KdV

(b) AC

Figure: Condition number $\kappa(\mathbf{M})$ as a function of time

Conditioning (2)



(a) KdV

(b) AC

Figure: Condition number $\kappa(\mathbf{M})$ as a function of the number of samples n

Optimal sampling – Bound on $\kappa(\mathbf{G})$[14]

Theorem (Condition number with optimal sampling)

For any $r > 0$, if p and n are such that the condition

$$p \leq \kappa \frac{n}{\ln n}, \quad \textit{with } \kappa := \frac{1 - \ln 2}{2 + 2r}$$

is satisfied, and the weight function w is defined as

$$w(\mathbf{x}) = \frac{p}{\sum_{j=1}^{p} |L_j(\mathbf{x}, \theta)|^2},$$

then

$$\mathbb{P}\left(\|\mathbf{G} - \mathbf{I}\|_2 \geq \frac{1}{2} \right) \leq 2n^{-r},$$

which implies $\kappa(\mathbf{G}) \leq 3$ with high probability.

[14] Cohen and Migliorati, "Optimal weighted least-squares methods".

Gram-Schmidt algorithm

Algorithm 2: Gram-Schmidt orthogonalization

Input: Set of linearly independent vectors $\{\partial_{\theta_i}\hat{u}\}_{i=1}^{p}$
Output: Set of orthonormal vectors $\{L_i\}_{i=1}^{p}$

$$\begin{aligned}
\tilde{L}_1 &= \partial_{\theta_1}\hat{u}, && L_1 = \tilde{L}_1/\|\tilde{L}_1\| \\
\tilde{L}_2 &= \partial_{\theta_2}\hat{u} - \langle \partial_{\theta_2}\hat{u}, L_1\rangle L_1, && L_2 = \tilde{L}_2/\|\tilde{L}_2\| \\
\tilde{L}_3 &= \partial_{\theta_3}\hat{u} - \langle \partial_{\theta_3}\hat{u}, L_1\rangle L_1 - \langle \partial_{\theta_3}\hat{u}, L_2\rangle L_2, && L_3 = \tilde{L}_3/\|\tilde{L}_3\| \\
&\vdots \\
\tilde{L}_p &= \partial_{\theta_p}\hat{u} - \langle \partial_{\theta_p}\hat{u}, L_1\rangle L_1 - \ldots - \langle \partial_{\theta_p}\hat{u}, L_{p-1}\rangle L_{p-1}, && L_p = \tilde{L}_p/\|\tilde{L}_p\|
\end{aligned}$$

Optimal sampling – Recover $\dot{\theta}$ from τ

We define an orthonormal basis $\{L_i\}_{i=1}^{p}$ of $\mathcal{T}_\theta$ such that $\sum_{i=1}^{p} \partial_{\theta_i} \hat{u}(\mathbf{x}, \theta)\dot{\theta}_i = \sum_{i=1}^{p} L_i(\mathbf{x}, \theta)\tau_i$ for some $\tau = (\tau_1, \ldots, \tau_p)^T \in \mathbb{R}^p$.

The Gram-Schmidt orthogonalization procedure induces the change of basis $L_i(\mathbf{x}, \theta) = \sum_{j=1}^{p} c_{i,j} \partial_{\theta_j} \hat{u}(\mathbf{x}, \theta)$, where $c_{i,j} = (\mathbf{C}^{-T})_{i,j}$ and

$$\mathbf{C}^T = \begin{bmatrix} \|\tilde{L}_1\| & 0 & \dots & \dots & \dots & 0 \\ \langle \partial_{\theta_2} \hat{u}, L_1 \rangle & \|\tilde{L}_2\| & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & & 0 \\ \vdots & \vdots & & \ddots & \ddots & 0 \\ \vdots & \vdots & & & \ddots & 0 \\ \langle \partial_{\theta_p} \hat{u}, L_1 \rangle & \dots & \dots & \dots & \dots & \|\tilde{L}_p\| \end{bmatrix}$$

The relation between the original parameters $\dot{\theta}$ and the new parameters τ is given by:

$$\mathbf{C}\dot{\theta} = \tau. \tag{1}$$

Optimal sampling for polynomials

We define V_p as the space spanned by the first p monomials on a given domain $\mathcal{X}$,

$$V_p = \mathrm{span}\{x^k : \; k = 0, ..., p-1\}.$$

From $\{x^k\}_{k=0}^{p-1}$, one can compute an orthonormal basis with respect to $L^2(\mathcal{X}, d\mu)$ via Gram-Schmidt (obtaining the first p Legendre polynomials on $\mathcal{X}$).

The least squares problem of interest is the following:

$$\min_{\mathbf{v}\in\mathbb{R}^p} \|A(\cdot)\mathbf{v} - b(\cdot)\|^2_{L^2(\mathcal{X}, d\mu)},$$

where $A = [A_1, ..., A_p] : \mathcal{X} \to \mathbb{R}^p$ is defined s.t. $A_k(x) = x^k$, and $b : \mathcal{X} \to \mathbb{R}$, $b(x) = \sin(x)$.

	$p = 5, \; n = 100$	$p = 10, \; n = 100$	$p = 10, \; n = 1000$
$\kappa(\mathbf{M})$	$7.11 \cdot 10^7$	$1.15 \cdot 10^{15}$	$2.36 \cdot 10^{15}$
$\kappa(\mathbf{G})$	1.52	6.03	2.95
$\kappa(\mathbf{C})$	$7.28 \cdot 10^3$	$4.56 \cdot 10^{10}$	$4.56 \cdot 10^{10}$

Table: Results of optimal sampling for polynomials