Geometrical perspectives on Physics-Informed Neural Networks

Méca−AAC day

Nilo Schwencke

TAU Team, INRIA Saclay−A&O, LISN, Paris-Saclay University−CNRS

March 25, 2025

Outline

Physics informed neural networks (PINNs)

Problem statement

We aim to solve:

$$\begin{cases} D(u) = f \in \mathrm{L}^2(\Omega \to \mathbb{R}, \mu) & \text{in } \Omega \\ B(u) = g \in \mathrm{L}^2(\partial\Omega \to \mathbb{R}, \sigma) & \text{on } \partial\Omega \end{cases}.$$

PINNs key idea

Optimize a neural network $u_{|\boldsymbol{\theta}}$ on the loss:

$$\widehat{\ell}_{D,B}(\boldsymbol{\theta}) := \frac{1}{2S_D} \sum_{i=1}^{S_D} \left(D[u_{|\boldsymbol{\theta}}](x_i^D) - f(x_i^D) \right)^2 + \frac{1}{2S_B} \sum_{i=1}^{S_B} \left(B[u_{|\boldsymbol{\theta}}](x_i^B) - g(x_i^B) \right)^2,$$

using autodiff to compute D, B (Raissi et al., 2019).

Problem

This leads to low accuracy with SGD.

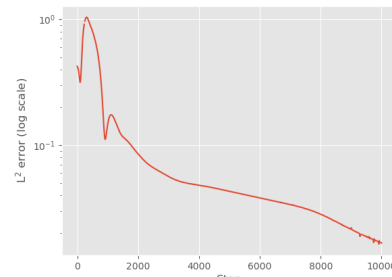


Figure: PINN solution under standard Adam optimization, to Laplace equation in 2 D.

Why does it work so bad ?

Intuition from NTK

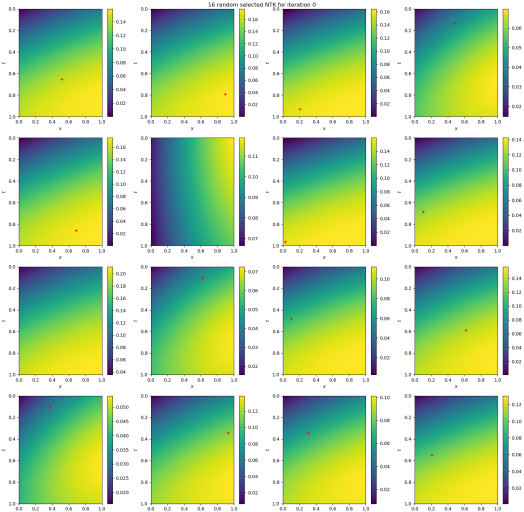


Figure: NTK for Laplace equation.

Intuition from Fourier

Consider the Fourier partial series:

$$S_N : \begin{cases} \mathbb{C}^{[\![-N,N]\!]} & \to & \mathrm{L}^2([0,1]) \\ (\alpha_k) & \mapsto & \sum_{k=-N}^{N} \alpha_k e^{2i\pi kx} \end{cases} .$$

S_N singular values are all 1 (perfect conditionning of the spectrum). **BUT:**

$$\Delta[S_N] \text{ spectrum is } \{4\pi^2 k^2 : 1 \leqslant k \leqslant N\}$$

Conclusion: Differential operators strongly impact the spectral condition.

Natural Gradient

Reinterpreting quadratic loss

Consider the loss of a classical quadratic regression problem, with batch (x_i):

$$\widehat{\ell}(\boldsymbol{\theta}) := \frac{1}{2S}\sum_{i=1}^{S}\left(u_{|\boldsymbol{\theta}}(x_i) - f(x_i)\right)^2.$$

In the population limit:

$$\widehat{\ell}(\theta) \overset{S\to\infty}{\to} \mathcal{L}(u_{|\boldsymbol{\theta}})\,; \quad \mathcal{L}(u) := \frac{1}{2}\|u-f\|^2_{\mathrm{L}^2(\Omega)}$$

This yields the Fréchet derivative:

$$\mathrm{d}\mathcal{L}|_u(h) = \langle \underbrace{u-f}_{\nabla\mathcal{L}_{|u}}, h\rangle_{\mathrm{L}^2(\Omega)},$$

and thus the gradient flow:

$$\begin{cases} u_0 \in \mathrm{L}^2(\Omega) \\ \dot{u}_t = -\nabla\mathcal{L}_{|u_t} = f - u_t \end{cases}.$$

Solution: $u_t = f - e^{-t}(u_0 - f)$.

A functional geometry perspective

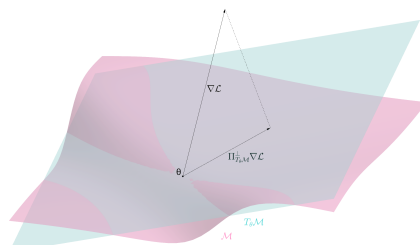
Natural gradient in functional space

The functional space is constrained to:

- $\mathcal{M} := \operatorname{Im} u = \{u_{\boldsymbol{\theta}} \,:\, \boldsymbol{\theta} \in \mathbb{R}^P\}$
- $T_{\boldsymbol{\theta}}\mathcal{M} := \operatorname{Im} \mathrm{d}u_{|\boldsymbol{\theta}} = \operatorname{Span}(\partial_p u_{\boldsymbol{\theta}})$

The Natural Gradient is then defined as:

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta\, \mathrm{d}u^{\dagger}_{|\boldsymbol{\theta}_t}\left(\Pi^{\perp}_{T_{\boldsymbol{\theta}_t}\mathcal{M}}\nabla\mathcal{L}_{|u_{|\boldsymbol{\theta}_t}}\right),$$



Definition-Proposition (Schwencke and Furtlehner (2025))

*The **Natural Neural Tangent Kernel (NNTK)** is the kernel of the projection $\Pi_{T_{\boldsymbol{\theta}}\mathcal{M}} : L^2(\Omega) \to L^2(\Omega)$ onto $T_{\boldsymbol{\theta}}\mathcal{M}$. It is given by the formula:*

$$NNTK_{\boldsymbol{\theta}}(x,y) := \sum_{1 \leqslant p,q \leqslant P} \left(\partial_p u_{|\boldsymbol{\theta}}(x)\right) \quad G^{\dagger}_{\boldsymbol{\theta}\, pq} \left(\partial_q u_{|\boldsymbol{\theta}}(y)\right)^t ; \quad G_{\boldsymbol{\theta}\, p,q} \quad := \left\langle \partial_p u_{|\boldsymbol{\theta}} \, , \, \partial_q u_{|\boldsymbol{\theta}} \right\rangle_{L^2(\Omega)} .$$

Corollary

The Natural Gradient update rewrites: $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta \, G^{\dagger}_{\boldsymbol{\theta}_t} \nabla \ell(\boldsymbol{\theta}_t)$; $\ell(\boldsymbol{\theta}) := \mathcal{L}(u_{|\boldsymbol{\theta}})$.

Shortcomings

- Computation of the Gram matrix $G_{\boldsymbol{\theta}_t}$ is quadratic in the number of parameters.
- Inversion of $G_{\boldsymbol{\theta}_t}$ is cubic

We introduce a the empirical Natural Gradient that scales linearly with the number of parameters.

empirical Natural Gradient

(N)NTK in a nutshell

The functionnal dynamic of (N)GD on the empirical loss $\widehat{\ell}$ is described by (Jacot et al., 2018; Rudner et al., 2019):

$$\frac{\mathrm{d}u_{\boldsymbol{\theta}_t}}{\mathrm{d}t}(x) = -\sum_{i=1}^{S} (N)NTK_{\boldsymbol{\theta}_t}(x, x_i)(u_{|\boldsymbol{\theta}_t}(x_i) - y_i),$$

Key Idea

The empirical dynamics taking place in:

$$\widehat{T}_{\boldsymbol{\theta}}\mathcal{M} := \mathrm{Span}\left(NNTK_{\boldsymbol{\theta}}(x_i, \cdot) \,:\, (x_i)_{1\leqslant i\leqslant N}\right),$$

we can define the empirical Natural Gradient:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\, \mathrm{d}u^{\dagger}_{|\boldsymbol{\theta}_t}\left(\Pi^{\perp}_{\widehat{T}_{\boldsymbol{\theta}_t}\mathcal{M}} \nabla\mathcal{L}_{|u_{|\boldsymbol{\theta}_t}}\right).$$

Byproduct

Yields an optimal criterion for (x_i) choice:

$$(x_i)^{\star} = \underset{(x_i)\in\Omega^S}{\mathrm{argmin}} \left\| \Pi^{\perp}_{\widehat{T}^{(x_i)}_{\theta,K}\mathcal{M}} \nabla\mathcal{L}_{|u_{|\boldsymbol{\theta}_t}} - \nabla\mathcal{L}_{|u_{|\boldsymbol{\theta}_t}} \right\|_{\mathrm{L}^2}$$

empirical Natural Gradient

Theorem (ANaGRAM)

Under mild assumptions:

$$du^{\dagger}_{|\boldsymbol{\theta}_t}\left(\Pi^{\perp}_{\widehat{T}_{\boldsymbol{\theta}_t}\mathcal{M}} \nabla\mathcal{L}_{|u_{|\boldsymbol{\theta}_t}}\right) = \widehat{\phi}^{\dagger}_{\boldsymbol{\theta}_t} \widehat{\nabla\mathcal{L}}_{\boldsymbol{\theta}_t},$$

with: for all $1 \leqslant p \leqslant P, 1 \leqslant i \leqslant S$

- $\widehat{\phi}_{\boldsymbol{\theta}_t\, i,p} := \partial_p u_{|\boldsymbol{\theta}_t}(x_i)$
- $\widehat{\nabla\mathcal{L}}_{\boldsymbol{\theta}_t\, i} := \nabla\mathcal{L}_{|u_{|\boldsymbol{\theta}_t}}(x_i)$

Key fact

$\hat{\phi}^{\dagger}_{\boldsymbol{\theta}_t}$ **can be computed with a SVD, with complexity** $\mathrm{O}\big(\min(PS^2, P^2S)\big)$**.**

Corollary

There exist P *points* $(\hat{x}_i)$ *such that:*

$$\Pi^{\perp}_{\widehat{T}_{\boldsymbol{\theta}}\mathcal{M}} \nabla\mathcal{L}_{|u_{|\boldsymbol{\theta}}} = \Pi^{\perp}_{T_{\boldsymbol{\theta}}\mathcal{M}} \nabla\mathcal{L}_{|u_{|\boldsymbol{\theta}}}.$$

Remark

$$(x_i)^\star = \underset{(x_i)\in\Omega^S}{\operatorname{argmin}} \left\| \Pi^{\perp}_{\widehat{T}^{(x_i)}_{\theta,K}\mathcal{M}} \nabla\mathcal{L}_{|u_{|\theta_t}} - \nabla\mathcal{L}_{|u_{|\theta_t}} \right\|_{\mathrm{L}^2} = \underset{(x_i)\in\Omega^S}{\operatorname{argmin}} \inf_{\alpha\in\mathbb{R}^S} \left\| \sum_{i=1}^{S} \alpha_i K_\theta(x_i,\cdot) - \nabla\mathcal{L}_{|u_{|\theta_t}} \right\|^2_{\mathrm{L}^2}$$

Consequence

$(x_i)^\star$ can be "learned" by the minimization through natural gradient descent of

$$u : \begin{cases} \Omega^S \times \mathbb{R}^S & \to & \mathrm{L}^2(\Omega \to \mathbb{R}, \mu) \\ \big((x_i),\alpha\big) & \mapsto & \sum_{i=1}^{S} \alpha_i K_\theta(x_i,\cdot) \end{cases}$$

Even better: exact formulas exist !

Proposition

- $\left\langle \partial_{\alpha_i} u_\theta, \partial_{\alpha_j} u_\theta \right\rangle = K_\theta(x_i,x_j)$
- $\left\langle \partial_{x_i} u_\theta, \partial_{\alpha_j} u_\theta \right\rangle = \alpha_i \partial_1 K_\theta(x_i,x_j)$
- $\left\langle \partial_{x_i} u_\theta, \partial_{x_j} u_\theta \right\rangle = \alpha_i \partial_2\partial_1 K_\theta(x_i,x_j)\alpha_j.$
- $\langle \partial_{\alpha_i} u_\theta, \nabla\mathcal{L} \rangle = \Pi_{T_\theta\mathcal{M}} \nabla\mathcal{L}(x_i) \simeq \nabla\mathcal{L}(x_i)$
- $\langle \partial_{x_i} u_\theta, \nabla\mathcal{L} \rangle = \alpha_i \Pi_{T_\theta\mathcal{M}} \nabla\mathcal{L}'(x_i) \simeq \alpha_i \nabla\mathcal{L}'(x_i)$

Application to PINNs

PINNs are a quadradic regression problem

Natural Gradient of PINNs

Key remark

The only difference between the losses:

$$\widehat{\ell}_{D,B}(\boldsymbol{\theta}) := \frac{1}{2S_D}\sum_{i=1}^{S_D}\left(D[u_{|\boldsymbol{\theta}}](x_i^D) - f(x_i^D)\right)^2 + \frac{1}{2S_B}\sum_{i=1}^{S_B}\left(B[u_{|\boldsymbol{\theta}}](x_i^B) - g(x_i^B)\right)^2,$$

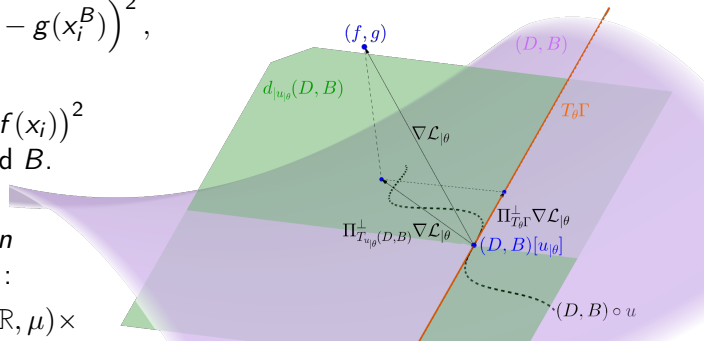
and $\widehat{\ell}(\boldsymbol{\theta}) := \frac{1}{2S}\sum_{i=1}^{S}\left(u_{|\boldsymbol{\theta}}(x_i) - f(x_i)\right)^2$ is the use of the operators D and B.

Proposition

PINNs are a quadradic regression problem with model: $(D, B) \circ u$:

$$\begin{cases} \mathbb{R}^P & \to & \mathcal{H} & \to & L^2(\Omega \to \mathbb{R}, \mu) \times L^2(\partial\Omega \to \mathbb{R}, \sigma) \\ \boldsymbol{\theta} & \mapsto & u_{|\boldsymbol{\theta}} & \mapsto & \left(D\left[u_{|\boldsymbol{\theta}}\right], B\left[u_{|\boldsymbol{\theta}}\right]\right) \end{cases}$$

Figure: Illustration of PINNs Natural Gradient



Experiments

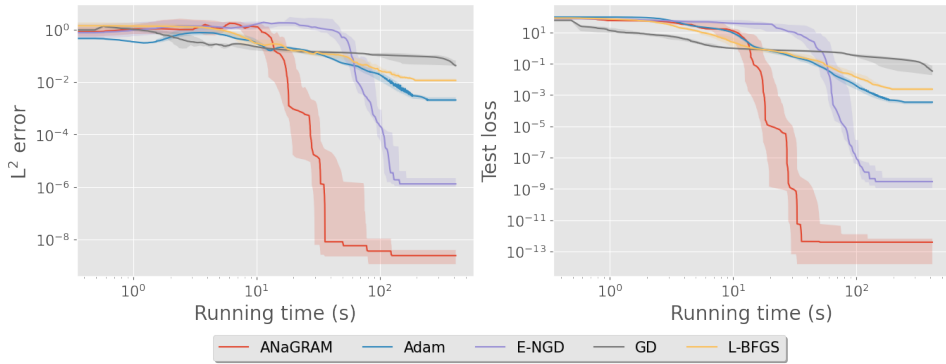


Figure: Performance comparison w.r.t running time for Laplace equation in 2 D:

$$\begin{cases} \Delta u = -2\pi^2 \sin(\pi x_1) \sin(\pi x_2) & \text{in } [0,1]^2 \\ u = 0 & \text{on } \partial[0,1]^2 \end{cases}$$

1+1 D Heat equation

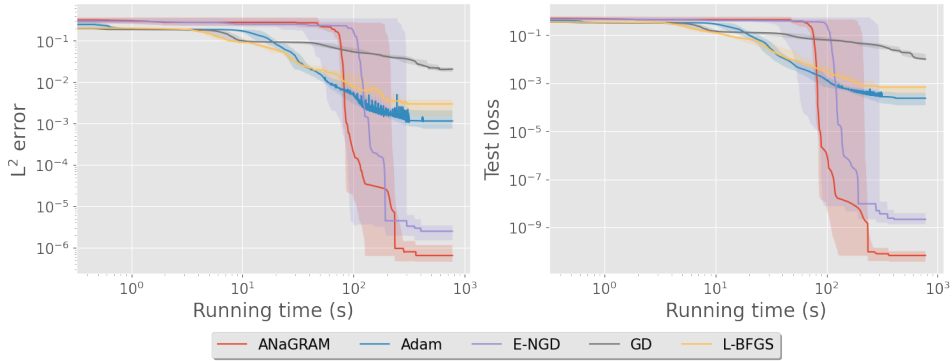


Figure: Performance comparison w.r.t running time for Heat equation in 1+1 D:

$$\begin{cases} \partial_t u - \frac{1}{4}\partial_{xx} u = 0 & \text{in } [0,1]^2 \\ u = 0 & \text{on } [0,1] \times \{0,1\} \\ u = sin(\pi x) & \text{on } \{0\} \times [0,1] \end{cases}$$

5 D Laplace equation

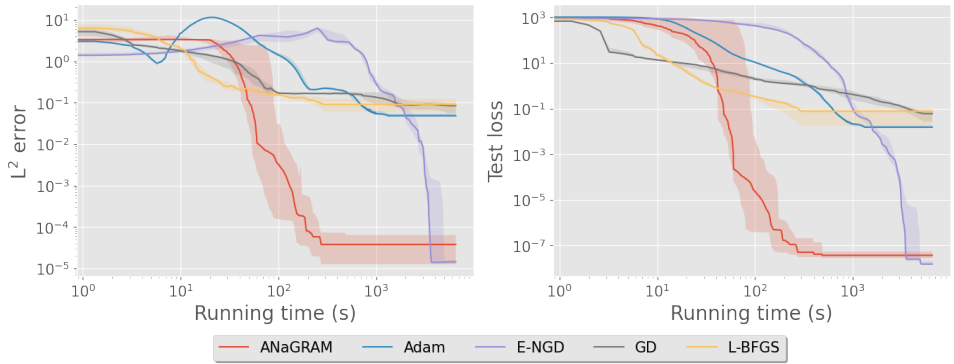


Figure: Performance comparison w.r.t running time for Laplace equation in 5 D:

$$\begin{cases} \Delta u = \pi^2 \sum_{k=1}^{5} \sin(\pi x_k) & \text{in } \Omega = [0,1]^5 \\ u = \sum_{k=1}^{5} \sin(\pi x_k) & \text{on } \partial\Omega \end{cases}$$

1+1 D Allen-Cahn equation

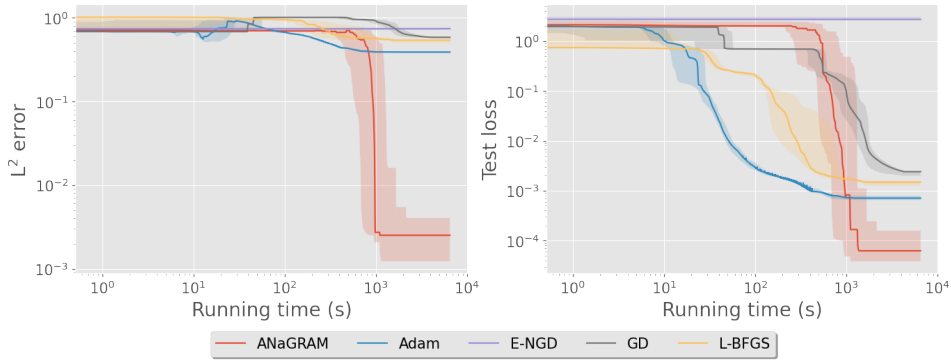
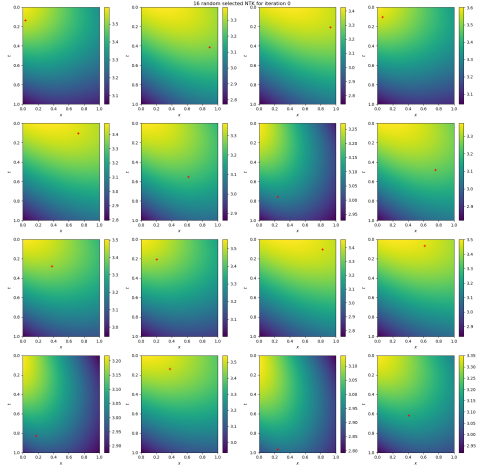


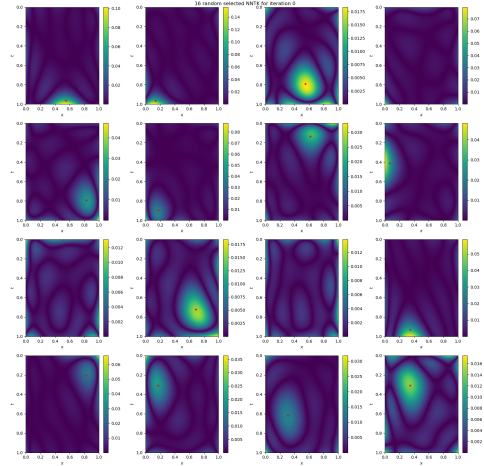
Figure: Performance comparison w.r.t running time for Allen-Cahn equation in 1+1 D:

$$\begin{cases} \partial_t u - 10^{-3}\,\partial_{xx} u - 5(u - u^3) = 0 & \text{in } \Omega = [0,1] \times [-1,1] \\ u = -1 & \text{on } \partial\Omega_{\text{border}} = [0,1] \times \{-1,1\} \\ u(0,x) = x^2 \cos(\pi x) & \text{on } \partial\Omega_0 = \{0\} \times [-1,1] \end{cases}$$

NTK vs NNTK of PINNs

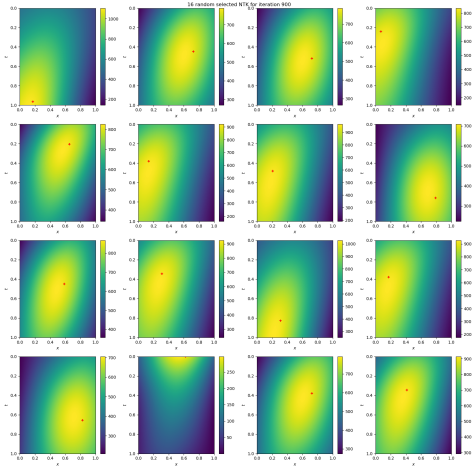


(a) NTK

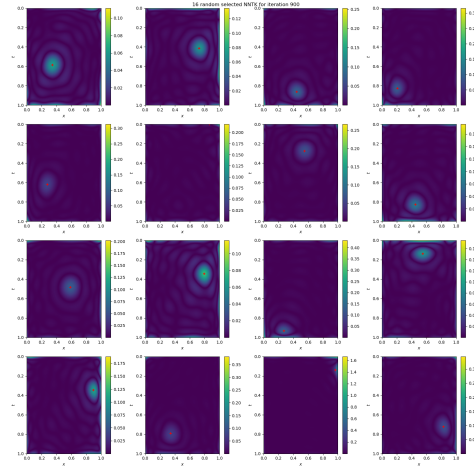


(b) NNTK

Figure: Comparison of NTK and NNTK at initialization for Heat equation in 1+1 D



(a) NTK



(b) NNTK

Figure: Comparison of NTK and NNTK at the end of optimization for Heat equation in 1+1 D

First results for collocation learning in Fourier space

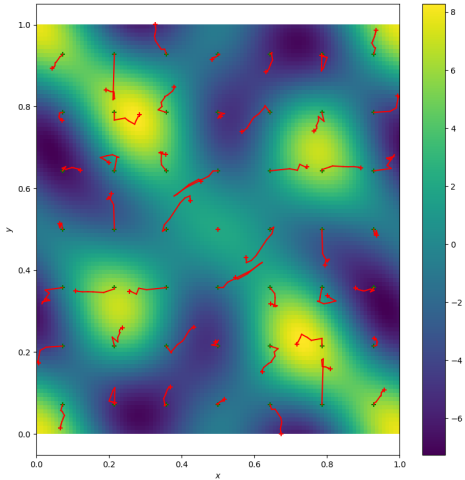


Figure: Points learning dynamic

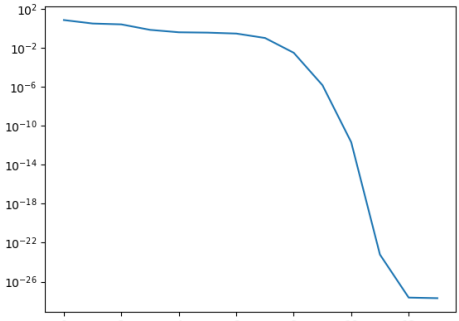


Figure: $\left\| \Pi^{\perp}_{\widehat{T}^{(x_j)}_{\theta,K}\mathcal{M}} \nabla \mathcal{L}_{|u_{|\theta_t}} - \nabla \mathcal{L}_{|u_{|\theta_t}} \right\|_{\mathbf{L^2}}$ wrt (x_i) learning steps

Conclusion and Perspectives

Conclusions

- Anagram gives a theoretically founded simplication to any natural-gradient algorithm lowering the complexity from $O(P^3)$ to $O(\min(PN^2, P^2N))$, which is above stochastic gradient descent only by a factor $\min(P, N)$.
- In the case of PINNs, we prove that natural gradient correspond to an optimal linear update following the Green's function.
- Empirical results are improved by several orders of magnitude.
- The SVD cut-off factor appears to be a pivotal hyper-parameter of the algorithm.

Perspectives

- Design of an optimal collocation points procedure, coupled with SVD cut-off factor adaptation strategy.
- Establish theoretical connections with classical algorithms, such as FEMs, FDMs, *etc.*
- Include data assmilation in this theoretical setting, and understand its regularizing effect.
- Include common optimization techniques (*e.g.* Momentum)
- Extend to order 2 methods
- Extend it to Operator learning
- Application to HJB equation

Thank you for your attention !

References

Jacot, A., F. Gabriel, and C. Hongler (2018): "Neural Tangent Kernel: Convergence and Generalization in Neural Networks," *Advances in neural information processing systems*, 31.

Raissi, M., P. Perdikaris, and G. Karniadakis (2019): "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations," *Journal of Computational Physics*, 378, 686–707.

Rudner, T. G., F. Wenzel, Y. W. Teh, and Y. Gal (2019): "The Natural Neural Tangent Kernel: Neural Network Training Dynamics under Natural Gradient Descent," in *4th Workshop on Bayesian Deep Learning (NeurIPS 2019)*.

Schwencke, N. and C. Furtlehner (2025): "ANaGRAM: A Natural Gradient Relative to Adapted Model for Efficient PINNs Learning," in *The Thirteenth International Conference on Learning Representations*.

Definition (Green's function of D)

A Green's function is any kernel function $g : \Omega \times \Omega \to \mathbb{R}$ such that the operator:

$$R : f \in D\left[\mathcal{H}\right] \mapsto \left(x \in \Omega \mapsto \int_{\Omega} g(x,s) f(s) \mu(\mathrm{d}s) \right) \in \mathcal{H}$$

verifies the equation: $D \circ R = I_{D[\mathcal{H}]}$

Definition (generalized Green's function of D on $\mathcal{H}_0 \subset \mathcal{H}$)

A generalized Green's function is any kernel function $g : \Omega \times \Omega \to \mathbb{R}$ such that the operator:

$$R : f \in \mathrm{L}^2(\Omega \to \mathbb{R}, \mu) \mapsto \left(x \in \Omega \mapsto \int_{\Omega} g(x,s) f(s) \mu(\mathrm{d}s) \right) \in \mathcal{H}$$

verifies the equation: $D \circ R = \Pi^{\perp}_{D[\mathcal{H}_0]}$

Theorem

Let $D : \mathcal{H} \to L^2(\Omega \to \mathbb{R}, \mu)$ be a linear differential operator and $u : \mathbb{R}^P \to \mathcal{H}$ a parametric model. Then for all $\boldsymbol{\theta} \in \mathbb{R}^P$, the generalized Green's function of D on $T_{\boldsymbol{\theta}}\mathcal{M} = \operatorname{Im} du_{|\boldsymbol{\theta}}$ is given by: for all $x, y \in \Omega$

$$g_{T_{\boldsymbol{\theta}}\mathcal{M}}(x, y) := \sum_{1 \leqslant p,q \leqslant P} \partial_p u_{|\boldsymbol{\theta}}(x)\, G^{\dagger}_{p,q} \partial_q D[u_{|\boldsymbol{\theta}}](y),$$

with: for all $1 \leqslant p, q \leqslant P$

$$G_{pq} := \left\langle \partial_p D[u_{|\boldsymbol{\theta}}]\,,\, \partial_q D[u_{|\boldsymbol{\theta}}] \right\rangle_{L^2(\Omega \to \mathbb{R}, \mu)}.$$

In particular, the natural gradient of PINNs can be rewritten:

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta\, du^{\dagger}_{|\boldsymbol{\theta}_t} \left(x \in \Omega \mapsto \int_{\Omega} g_{T_{\boldsymbol{\theta}_t}\mathcal{M}}(x, y) \nabla \mathcal{L}_{|\boldsymbol{\theta}_t}(y) \mu(dy) \right),$$

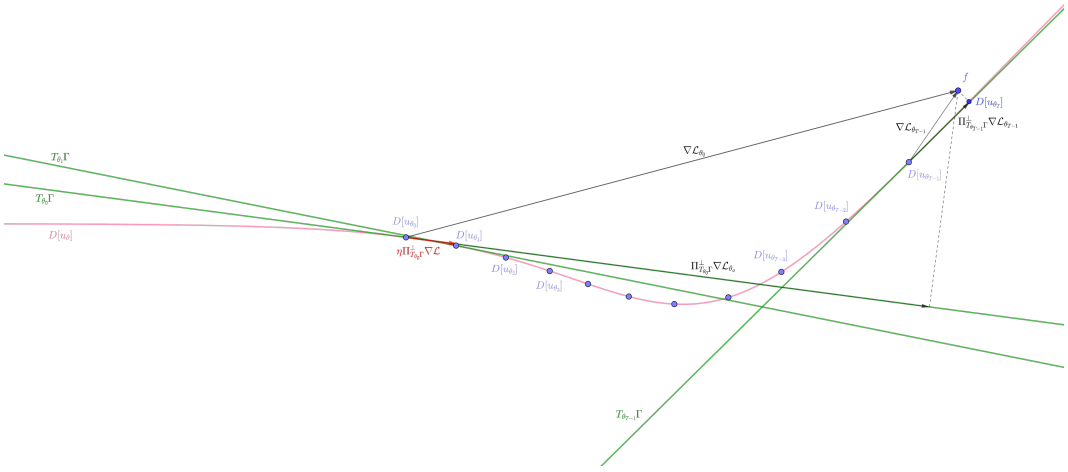


Figure: Illustration of PINNs learning process under natural gradient, as successive applications of Green's function

In the population limit, the natural gradient of PINNs is the update:

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta \, \mathrm{d}\big((D, B) \circ u\big)^{\dagger}_{|\boldsymbol{\theta}_t} \left(\Pi^{\perp}_{T_{\boldsymbol{\theta}_t}\Gamma} \nabla \mathcal{L}_{|u_{|\boldsymbol{\theta}_t}} \right)$$

Corollary

The kernel of $\Pi_{T_{\boldsymbol{\theta}}\Gamma}$ *is: for all* $x, y \in (\Omega \times \partial\Omega)^2$

$$\begin{aligned} NNTK_{\boldsymbol{\theta}}(x, y) &= \sum_{1 \leqslant p,q \leqslant P} \partial_p (D, B) \left[u_{|\boldsymbol{\theta}}\right](x) G^{\dagger}_{\boldsymbol{\theta}\, p,q} \partial_q (D, B) \left[u_{|\boldsymbol{\theta}}\right](y) \\ &= \sum_{1 \leqslant p,q \leqslant P} \left(\partial_p D \left[u_{|\boldsymbol{\theta}}\right](x_1), \partial_p B \left[u_{|\boldsymbol{\theta}}\right](x_2)\right) G^{\dagger}_{\boldsymbol{\theta}\, p,q} \left(\partial_q D \left[u_{|\boldsymbol{\theta}}\right](y_1), \partial_q B \left[u_{|\boldsymbol{\theta}}\right](y_2)\right), \end{aligned}$$

where for all $1 \leqslant p, q \leqslant P$

$$\begin{aligned} G_{\boldsymbol{\theta}\, p,q} &:= \left\langle \partial_p (D, B) \left[u_{|\boldsymbol{\theta}}\right] , \partial_q (D, B) \left[u_{|\boldsymbol{\theta}}\right] \right\rangle_{L^2(\Omega \to \mathbb{R}, \mu) \times L^2(\partial\Omega \to \mathbb{R}, \sigma)} \\ &= \left\langle \partial_p D \left[u_{|\boldsymbol{\theta}}\right] , \partial_q D \left[u_{|\boldsymbol{\theta}}\right] \right\rangle_{L^2(\Omega \to \mathbb{R}, \mu)} + \left\langle \partial_p B \left[u_{|\boldsymbol{\theta}}\right] , \partial_q B \left[u_{|\boldsymbol{\theta}}\right] \right\rangle_{L^2(\partial\Omega \to \mathbb{R}, \sigma)}. \end{aligned}$$

Corollary

The kernel of $\Pi_{\widehat{T}_{\boldsymbol{\theta}}\Gamma}$ *is: for all* $x, y \in (\Omega \times \partial\Omega)^2$

$$\widehat{k}(x,y) = \sum_{1 \leqslant i,j \leqslant S} NNTK_{\boldsymbol{\theta}}(x, x_i)\, \widehat{G}^{\dagger}_{\boldsymbol{\theta}\, i,j}\, NNTK_{\boldsymbol{\theta}}(x_j, y), \text{ where}$$

$$G_{\boldsymbol{\theta}\, i,j} := \left\langle NNTK_{\boldsymbol{\theta}}(\cdot, x_i)\,,\, NNTK_{\boldsymbol{\theta}}(x_j, \cdot) \right\rangle_{L^2(\Omega \to \mathbb{R}, \mu) \times L^2(\partial\Omega \to \mathbb{R}, \sigma)} = NNTK_{\boldsymbol{\theta}}(x_i, x_j)$$

Theorem (ANaGRAM for PINNs)

Under mild assumptions, the empirical natural gradient update:

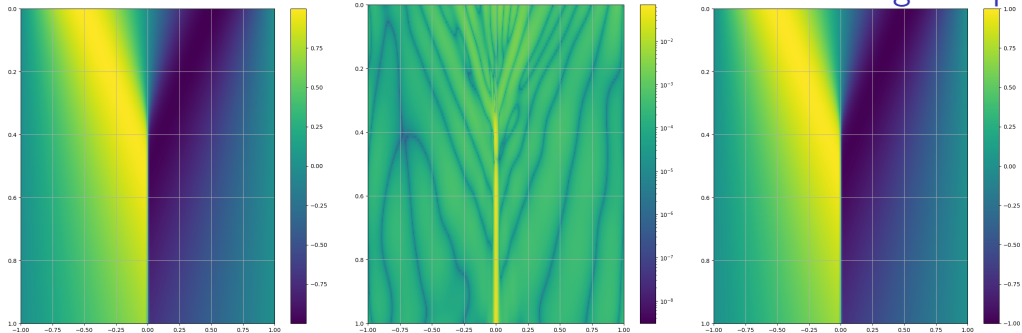
$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta\, d\big((D,B) \circ u\big)^{\dagger}_{|\boldsymbol{\theta}_t} \left(\Pi^{\perp}_{\widehat{T}_{\boldsymbol{\theta}_t}\Gamma} \nabla \mathcal{L}_{|u_{|\boldsymbol{\theta}_t}} \right),$$

does not require to estimate a Gram matrix. More precisely, we have:

$$d\big((D,B) \circ u\big)^{\dagger}_{|\boldsymbol{\theta}_t} \left(\Pi^{\perp}_{\widehat{T}_{\boldsymbol{\theta}_t}\Gamma} \nabla \mathcal{L}_{|u_{|\boldsymbol{\theta}_t}} \right) = \widehat{\phi}^{\dagger}_{\boldsymbol{\theta}_t} \widehat{\nabla \mathcal{L}}_{\boldsymbol{\theta}_t},$$

where: for all $1 \leqslant p \leqslant P, 1 \leqslant i \leqslant S$

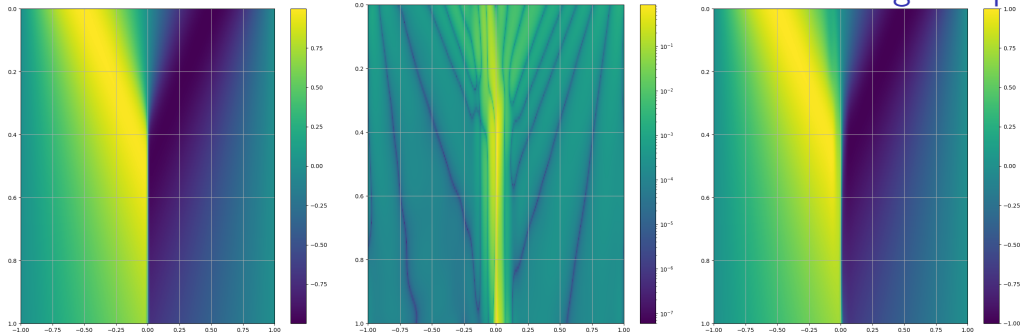
- $\widehat{\phi}_{\boldsymbol{\theta}_t\, i,p} := \left(\partial_p D\left[u_{|\boldsymbol{\theta}_t}\right](x_{i1}), \partial_p B\left[u_{|\boldsymbol{\theta}_t}\right](x_{i2})\right)$
- $\widehat{\nabla \mathcal{L}}_{\boldsymbol{\theta}_t\, i} := \nabla \mathcal{L}_{|u_{|\boldsymbol{\theta}_t}}(x_i)$



(a) True solution profile (b) L^2 error of PINN solution (c) PINN's solution profile

Figure: PINN's solution under Adam (15k steps) + L-BFGS (15k) steps for Burgers equation in 1+1 D:

$$\begin{cases} \partial_t u + u\partial_x u = \nu \partial_{xx} u & \text{in } [0,1] \times [-1,1] \\ u = 0 & \text{on } [0,1] \times \{-1,1\} \\ u = -sin(\pi x) & \text{on } \{0\} \times [-1,1] \end{cases}$$



(a) True solution profile (b) L^2 error of PINN solution (c) PINN's solution profile

Figure: PINN's solution under Anagram (500 steps) for Burgers equation in 1+1 D:

$$\begin{cases} \partial_t u + u\partial_x u = \nu \partial_{xx} u & \text{in } [0,1] \times [-1,1] \\ u = 0 & \text{on } [0,1] \times \{-1,1\} \\ u = -sin(\pi x) & \text{on } \{0\} \times [-1,1] \end{cases}$$