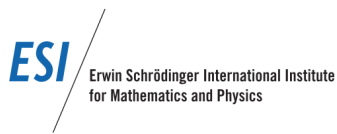
Kernelizing natural gradient, with applications to PINNs

Infinite-dimensional Geometry: Theory and Applications,

Nilo Schwencke

TAU Team, INRIA Saclay−LISN, Paris-Saclay University−CNRS

February 05th, 2025



Outline

Neural networks in a Nutshell

Let $\mathcal{H}$ be a Hilbert space of functions $\Omega \to \mathbb{R}$, Ω compact domain of $\mathbb{R}^n$.

Definition

A neural network is a differentiable non-linear functional

$$u : \begin{cases} \mathbb{R}^P & \to & \mathcal{H} \\ \theta & \mapsto & u_{|\theta} \end{cases}$$

Universal approximation property of neural networks (Leshno et al., 1993):
for all $\varepsilon > 0$ and $f \in \mathrm{L}^2(\Omega \to \mathbb{R})$, there are $P \in \mathbb{N}$ and $\theta^\star \in \mathbb{R}^P$ such that:

$$\left\| u_{|\theta^\star} - f \right\|_{\mathrm{L}^2(\Omega \to \mathbb{R})} < \varepsilon$$

Neural networks in a Nutshell

How to approximate such a solution in practice ?

Answer: Apply Euler's method[a] to:

$$\begin{cases} \theta_0 \sim \mathbb{P} & \text{in } \mathbb{R}^P \\ \dot{\theta}_t = -\nabla \ell_{|\theta_t} & \text{for all } t > 0 \end{cases},$$

where for all $\theta \in \mathbb{R}^P$, ℓ is the **loss**

$$\ell(\theta) = \frac{1}{2} \left\| u_\theta - f \right\|_{\mathcal{H}}^2$$

Problem: Which metric should be taken to compute the gradient $\nabla \ell_{|\theta}$?

Classic method: use Euclidian metric of $\mathbb{R}^P$.

Better founded: use the metric of $\mathcal{H}$.

[a] *aka* gradient descent

Natural Gradient

Reinterpreting the loss ℓ

Introducing the **functional loss** $\mathcal{L}$:

$$\mathcal{L} : \begin{cases} \mathcal{H} & \to & \mathbb{R}^+ \\ v & \mapsto & \frac{1}{2}\|v - f\|_{\mathcal{H}}^2 \end{cases}$$

ℓ can be rewritten:

$$\ell = \mathcal{L} \circ u$$

Taking the Fréchet derivative:

- $\mathrm{d}\mathcal{L}_{|v}(h_v) = \big\langle \underbrace{v - f}_{\nabla \mathcal{L}_{|v}}, h_v \big\rangle_{\mathcal{H}}$
- $\mathrm{d}\ell_{|\theta}(h_\theta) = \Big\langle \nabla \mathcal{L}_{|u_{|\theta}}\,,\, \mathrm{d}u_{|\theta}(h_\theta) \Big\rangle_{\mathcal{H}}$

Pseudo-manifold structure

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

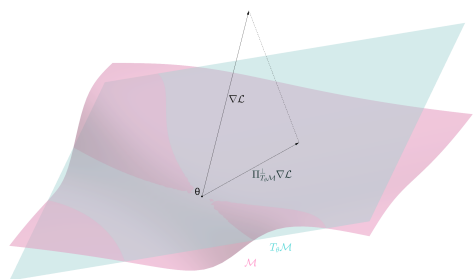
Functionnal perspective

Euclidian gradient [pushforward metric]

$$\nabla^E \ell_{|\theta} = \left(\partial_p \ell_{|\theta}\right)_{1 \leqslant p \leqslant P}$$

Natural gradient [pullback metric]

$$\begin{aligned} \nabla^N \ell_{|\theta} &= \mathrm{d}u^\dagger_{|\theta} \left(\nabla \mathcal{L}_{|u_{|\theta}}\right) \\ &= \mathrm{d}u^\dagger_{|\theta} \left(\Pi^\perp_{T_\theta\mathcal{M}} \nabla \mathcal{L}_{|u_{|\theta}}\right) \end{aligned}$$



Natural gradient rewrites:

$$\nabla^N \ell_{|\theta} = G_\theta^\dagger \nabla^E \ell_{|\theta}$$

with G_θ the Gram matrix:

$$G_{\theta\, p,q} := \left\langle \partial_p u_{|\theta}\, ,\, \partial_q u_{|\theta} \right\rangle_{\mathcal{H}}$$

Shortcomings

- Computation of G_θ is challenging (*e.g.* involves integral estimation in L^2) and quadratic in the number of parameters.
- Inversion of G_θ is cubic.

Common solutions are approximations through Kronecker factorization (Martens and Grosse, 2015).

Algorithmic perspective

We will introduce a new kind of natural gradient that scales linearly with the number of parameters.

Generalization to pseudo-Riemannian manifolds

$$u : \mathbb{R}^P \to (\mathcal{N}, \mathcal{G})$$

Gram matrix becomes:

$$G_{\theta\, p,q} := \mathcal{G}_{|u_{|\theta}}(\partial_p u_{|\theta}, \partial_q u_{|\theta}),$$

Typical application : Fisher-Rao metric (Amari and Douglas, 1998).

Reproducing Kernel Hilbert Spaces (RKHS) *détour*

Reproducing Kernel Hilbert Spaces (RKHS) *détour*

Definition-Proposition

An Hilbert space $\mathcal{H}$ of functions $\Omega \to \mathbb{R}$ is a RKHS if and only if the following equivalent conditions are met:

1. *There exist a function $k : \Omega \times \Omega \to \mathbb{R}$ such that:*
   - $\mathcal{H} = \overline{\text{Span}\,(k(x,\cdot) : x \in \Omega)}$
   - $\langle k(x,\cdot)\,,\, k(y,\cdot)\rangle_{\mathcal{H}} = k(x,y)$
2. *for all $x \in \Omega$, the evaluation form $e_x : f \in \mathcal{H} \mapsto f(x)$ is continuous.*

Proposition

Any finite dimensional space $\mathcal{H}$ is a RKHS

Proposition

If $\mathcal{H} := \overline{\text{Span}(u_i : i \in \mathbb{N})}$, is an RKHS, then its kernel is given by: for all $x, y \in \Omega$

$$k(x,y) = \sum_{i,j \in \mathbb{N}} u_i(x) G^{\dagger}_{i,j} u_j(y)$$

where $G_{ij} := \langle u_i\,,\, u_j\rangle_{\mathcal{H}}$.

Proposition

If $\mathcal{H} \supset \mathcal{H}_0$ is an RKHS with kernel k, then the orthogonal projection $\Pi^{\perp}_{\mathcal{H}_0}$ onto $\mathcal{H}_0$ is given by:

$$\Pi^{\perp}_{\mathcal{H}_0}(f)(x) = \langle k(x,\cdot)\,,\, f\rangle_{\mathcal{H}}$$

Remark

Applying to $\mathcal{H}_0 = T_\theta \mathcal{M}$, we get natural gradient.

empirical Natural Gradient (eNG)

Given a gradient $\nabla \ell : \mathbb{R}^P \to \mathbb{R}^P$, the parameters ODE:

$$\dot{\theta}_t = -\nabla \ell_{|\theta_t}$$

translates into:

$$\frac{\mathrm{d}u_{|\theta_t}}{\mathrm{d}t}(x) = \mathrm{d}u_{|\theta_t}\left(\dot{\theta}_t\right)(x) = -\mathrm{d}u_{|\theta_t}\left(\nabla \ell_{|\theta_t}\right)(x),$$

in the tangent space $T_\theta \mathcal{M} \subset \mathcal{H}$.

Empirical dynamic

In practice, ℓ is estimated, *i.e.* replaced by:

$$\widehat{\ell}(\theta) := \frac{1}{2S} \sum_{i=1}^{S} (u_\theta(x_i) - y_i)^2,$$

This yields a new Euclidian gradient $\nabla \widehat{\ell} : \mathbb{R}^P \to \mathbb{R}^P$.

ODEs in the functional space

Associated functional dynamic (Jacot et al., 2018):

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

with:

$$NTK_{\theta_t}(x, y) := \sum_{p=1}^{P} \left(\partial_p u_{|\theta_t}(x)\right) \left(\partial_p u_{|\theta_t}(y)\right)^T$$

For Natural gradient: NTK is replaced with: $NNTK_{\theta_t}(x, y) :=$

$$\sum_{1 \leqslant p,q \leqslant P} \left(\partial_p u_{|\theta_t}(x)\right) {G_{\theta_t}}^{\dagger}_{pq} \left(\partial_q u_{|\theta_t}(y)\right)^T$$

$NNTK_\theta$ is precisely the reproducing kernel of $T_\theta \mathcal{M}$

Corollary

- *The empirical **Euclidian** functional dynamics takes place in:*
$$\text{Span}\left(NTK_\theta(x_i,\cdot) \,:\, (x_i)_{1\leqslant i\leqslant N}\right) \subset T_\theta\mathcal{M}.$$
- *The empirical **Natural** functional dynamics takes place in the **empirical tangent space**:* $\widehat{T}_\theta\mathcal{M} := \text{Span}\left(NNTK_\theta(x_i,\cdot) \,:\, (x_i)_{1\leqslant i\leqslant N}\right) \subset T_\theta\mathcal{M}.$

Corollary

We can define an empirical natural gradient update by:

$$\theta_{t+1} = \theta_t - \eta\, du^\dagger_{|\theta_t}\left(\Pi^\perp_{\widehat{T}_{\theta_t}\mathcal{M}}\nabla\mathcal{L}_{|u_{|\theta_t}}\right).$$

An empirical Natural Gradient computation : ANaGRAM

Theorem (ANaGRAM)

Under mild assumptions, the empirical natural gradient update:

$$\theta_{t+1} = \theta_t - \eta \, du^{\dagger}_{|\theta_t} \left(\Pi^{\perp}_{\widehat{\mathcal{T}}_{\theta_t}\mathcal{M}} \nabla \mathcal{L}_{|u_{|\theta_t}} \right),$$

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

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

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

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

Remark

The pseudoinverse $\widehat{\phi}^{\dagger}_{\theta_t}$ can be computed with a SVD. In particular the complexity of the empirical natural gradient update is $O\big(\min(PS^2, P^2S)\big)$, which has to be compared with:

- $O(PS)$ for classical gradient update.
- $O(P^3)$ for classical natural gradient update.

Corollary

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

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

Application to PINNs

Given a PDE:

$$\begin{cases} D[u] = f & \text{in } \Omega \\ B[u] = g & \text{on } \partial\Omega \end{cases},$$

we want to solve it using a neural network Ansatz.

How to approximate such a solution?

Answer: Just as we would for any neural network, *i.e.* by gradient descent (Lagaris et al., 1998; Raissi et al., 2019). More precisely, we will optimize the loss:

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

PINNs are a quadradic regression problem

The only difference between the losses:

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

and:

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

is the use of the differential operator D and the boundary operator B.

Proposition

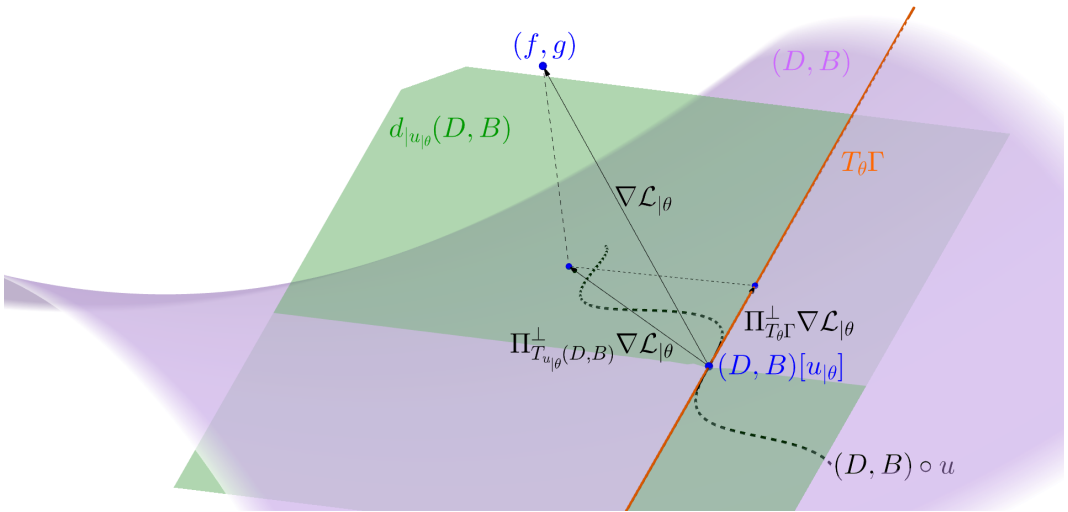
PINNs are a classical quadradic regression problem with model:

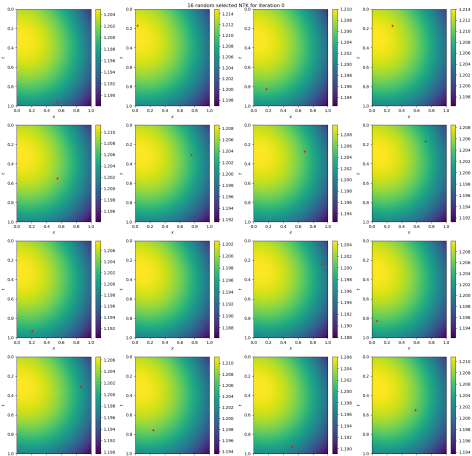
$$(D,B)\circ u : \left\{\begin{array}{ccccc} \mathbb{R}^P & \to & \mathcal{H} & \to & L^2(\Omega\to\mathbb{R},\mu)\times \\ & & & & L^2(\partial\Omega\to\mathbb{R},\sigma) \\ \theta & \mapsto & u_{|\theta} & \mapsto & \left(D\left[u_{|\theta}\right], B\left[u_{|\theta}\right]\right) \end{array}\right.$$

In the same way:

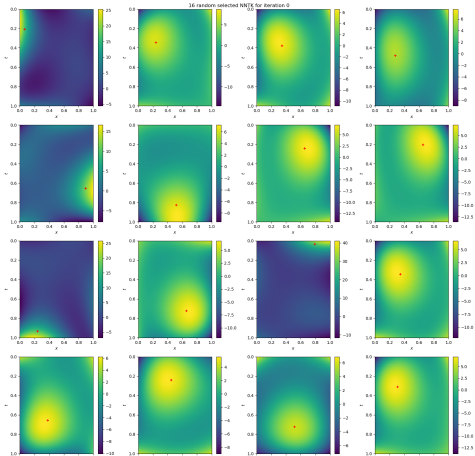
- $\Gamma := \mathrm{Im}\left((D,B)\circ u\right) = \left\{\left(D\left[u_{|\theta}\right], B\left[u_{|\theta}\right]\right) : \theta\in\mathbb{R}^P\right\}$
- $T_\theta\Gamma := \mathrm{Im}\,\mathrm{d}\left((D,B)\circ u\right)_{|\theta} = \mathrm{Span}\left(\left(\mathrm{d}_{|u_{|\theta}} D\left[\partial_p u_{|\theta}\right], \mathrm{d}_{|u_{|\theta}} B\left[\partial_p u_{|\theta}\right]\right)_{p=1}^{P}\right)$
- $\nabla\mathcal{L}_{|u_{|\theta}} = \left(D\left[u_{|\theta}\right] - f, B\left[u_{|\theta}\right] - g\right)$

Natural Gradient for PINNs



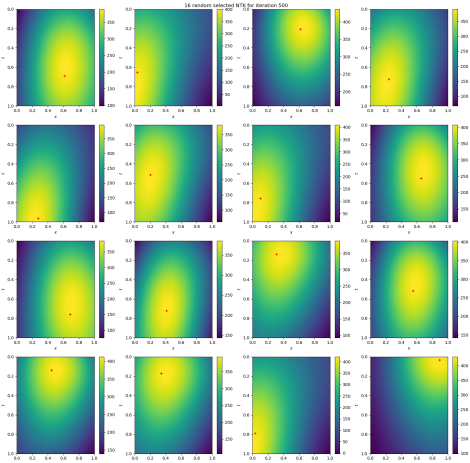


(a) NTK

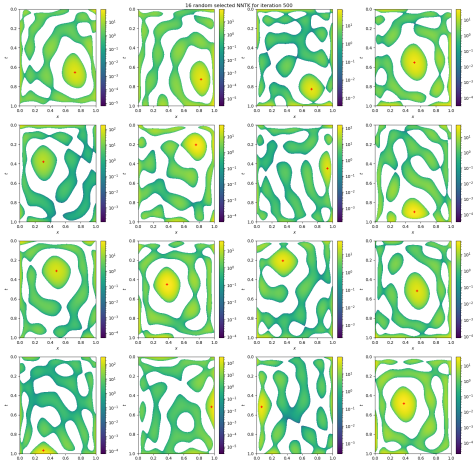


(b) NNTK

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

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 $\theta \in \mathbb{R}^P$, the generalized Green's function of D on $T_\theta \mathcal{M} = \operatorname{Im} du_{|\theta}$ is given by: for all $x, y \in \Omega$

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

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

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

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

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

Natural Gradient and Green's function

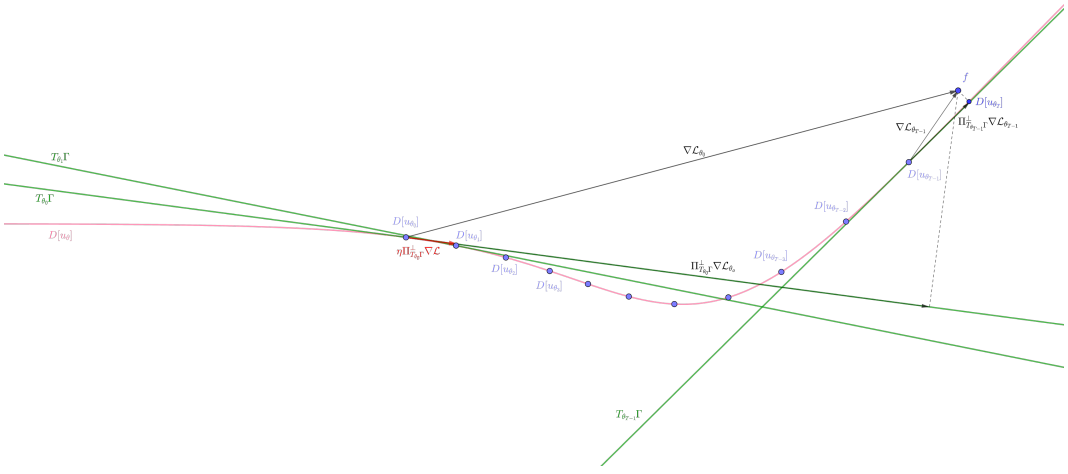


Figure: Illustration of PINNs learning process under natural gradient, as successive applications of Green's function

Experiments

2 D Laplace equation

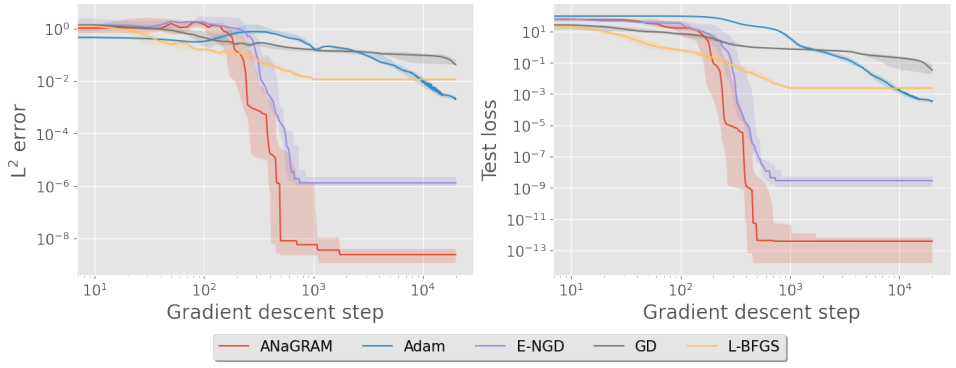


Figure: Performance comparison w.r.t running step for Laplace equation in 2 D:

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

2 D Laplace equation

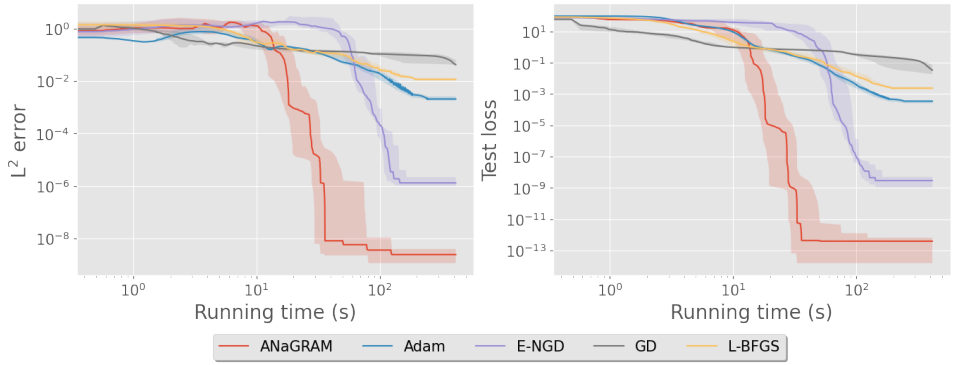


Figure: Performance comparison w.r.t running time for Laplace equation in 2 D:

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

1+1 D Heat equation

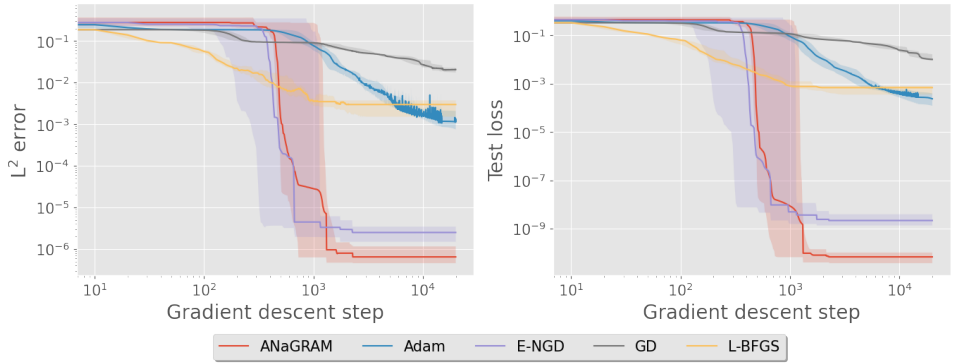


Figure: Performance comparison w.r.t running step for Heat equation in 1+1 D:

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

1+1 D Heat equation

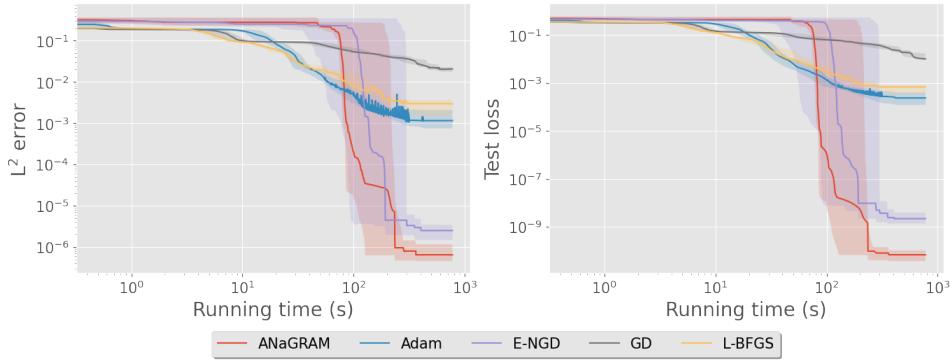


Figure: Performance comparison w.r.t running time for Heat equation in 1+1 D:

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

5 D Laplace equation

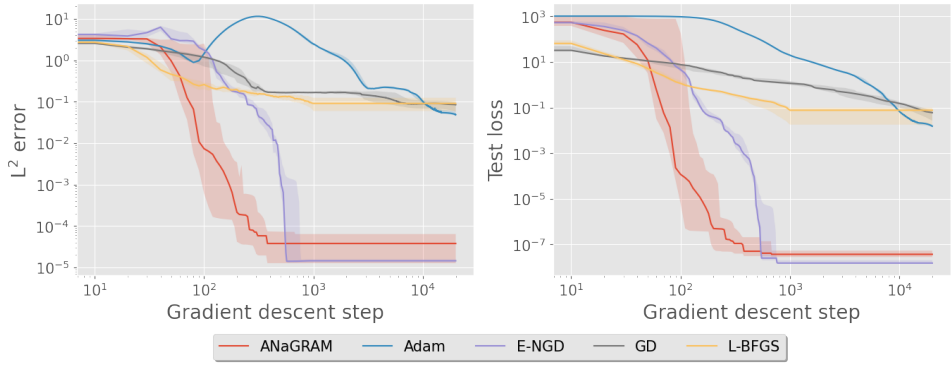


Figure: Performance comparison w.r.t running step for Laplace equation in 5 D:

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

5 D Laplace equation

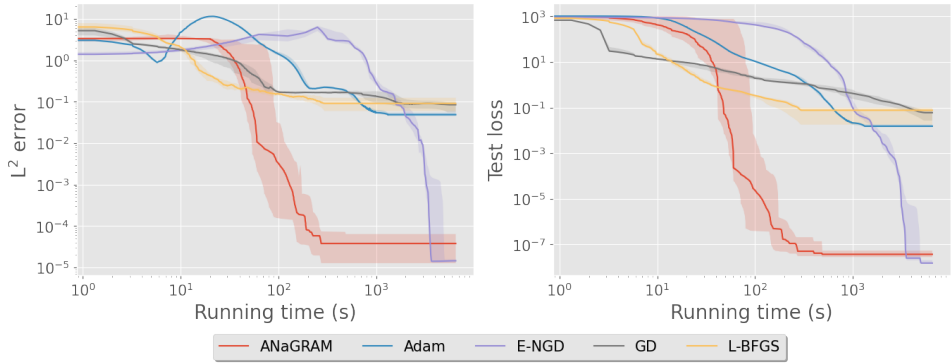


Figure: Performance comparison w.r.t running time for Laplace equation in 5 D:

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

1+1 D Allen-Cahn equation

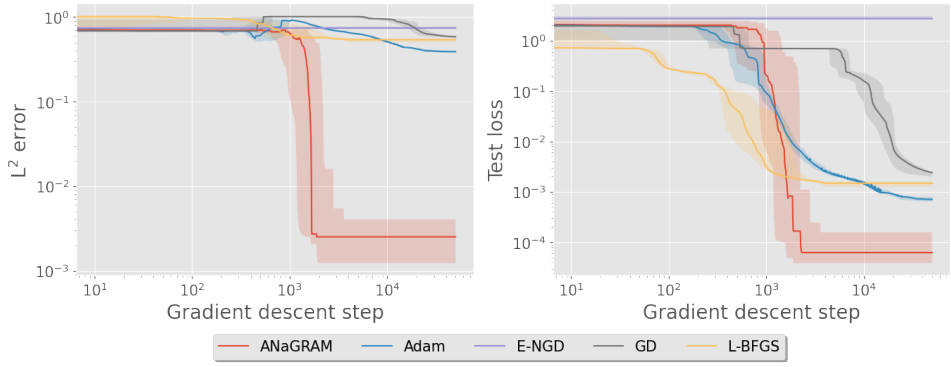


Figure: Performance comparison w.r.t running step for Allen-Cahn equation in 1+1 D:

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

1+1 D Allen-Cahn equation

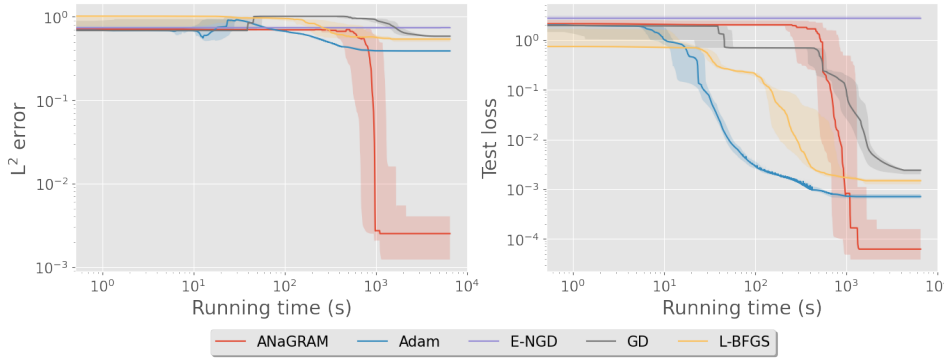
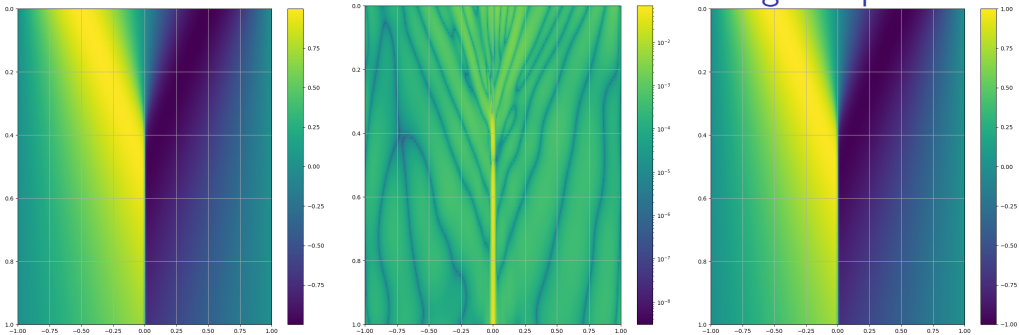


Figure: Performance comparison w.r.t running time for Allen-Cahn equation in 1+1 D:

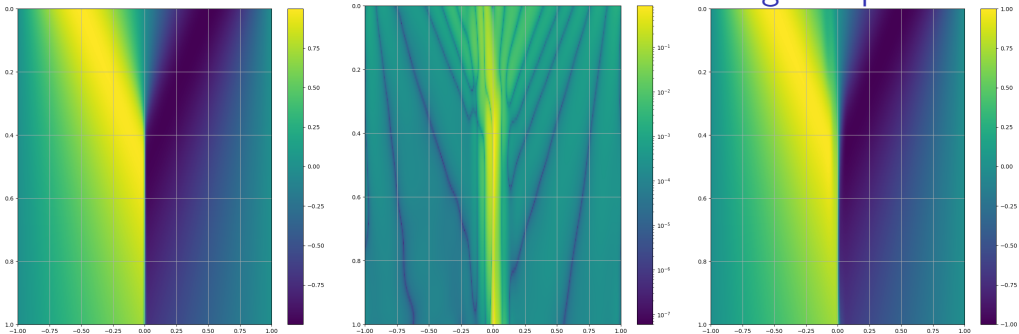
$$\begin{cases} \partial_t u - 10^{-3}\,\partial_{xx} u - 5(u - u^3) = 0 & \text{in } \Omega = [0,1] \times [-1,1] \\ u = -1 & \text{on } \partial\Omega_{\text{border}} = [0,1] \times \{-1,1\} \\ u(0,x) = x^2 \cos(\pi x) & \text{on } \partial\Omega_0 = \{0\} \times [-1,1] \end{cases}$$



(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:

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

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 common optimization techniques (*e.g.* Momentum)
- Extend to order 2 methods
- Extend it to Operator learning

Thank you for your attention !

References I

Amari, S.-I. and S. C. Douglas (1998): "Why Natural Gradient?" in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, IEEE, vol. 2, 1213–1216.

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

Lagaris, I. E., A. Likas, and D. I. Fotiadis (1998): "Artificial Neural Networks for Solving Ordinary and Partial Differential Equations," *IEEE transactions on neural networks*, 9, 987–1000.

Leshno, M., V. Y. Lin, A. Pinkus, and S. Schocken (1993): "Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function," *Neural networks*, 6, 861–867.

Martens, J. and R. Grosse (2015): "Optimizing Neural Networks with Kronecker-factored Approximate Curvature," in *Proceedings of the 32nd International Conference on Machine Learning*, PMLR, 2408–2417.

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)*.

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

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

Corollary

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

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

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

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

Corollary — empirical Natural Gradient and ANaGRAM for PINNs

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

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

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

Theorem (ANaGRAM for PINNs)

Under mild assumptions, the empirical natural gradient update:

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

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

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

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

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