

Addressing Spectral Bias in PINNs

SCALP kick-off

Nilo Schwencke

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

February 10, 2025



1 Physics informed neural networks (PINNs)

- PINNs in a nutshell
- PINNs shortcomings and drawbacks
- Intuition from Fourier
- Intuition from PINNs NTK

2 Natural Gradient

- Natural Gradient in a nutshell
- A functional analysis perspective

3 Neural Tangent Kernel (NTK)

- NTK in a nutshell
- Some consequences

4 empirical Natural Gradient (eNG)

- Reproducing Kernel Hilbert Spaces (RKHS) *détour*
- An empirical Natural Gradient computation : ANaGRAM

5 Application to PINNs

- PINNs are a quadratic regression problem
- Natural Gradient for PINNs
- Natural Gradient and Green's function

6 Experiments

- 2 D Laplace equation
- 1+1 D Heat equation
- 5 D Laplace equation
- 1+1 D Allen-Cahn equation
- Burgers equation

7 Conclusion and Perspectives

Physics informed neural networks (PINNs)

Neural networks

Let \mathcal{H} be a Hilbert space of functions $\Omega \rightarrow \mathbb{R}$.

Definition

A neural network is a differentiable non-linear functional

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

Universal approximation property of neural networks (Leshno et al., 1993): if P large enough, $u|_{\theta}$ can approximate any function in $L^2(\Omega \rightarrow \mathbb{R})$, in particular any solution to a PDE.

PINNs in a nutshell

How to approximate such a solution?

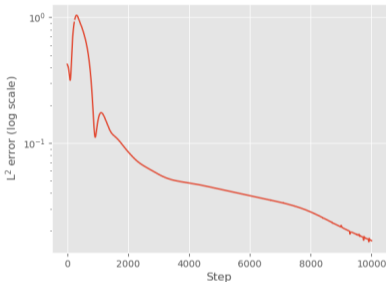
Answer: Just as we would for any neural network, *i.e.* by gradient descent (Lagaris et al., 1998; Raissi et al., 2019). More precisely, given the PDE:

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

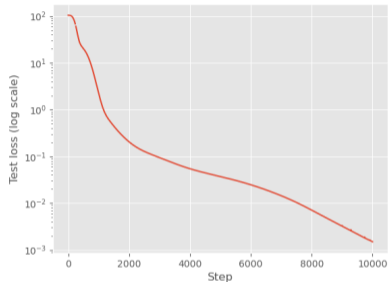
we will optimize the loss:

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

PINNs shortcomings and drawbacks



(a) L² error of PINN solution



(b) Test loss of PINN solution

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

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

Why does it work so bad ?

Intuition from Fourier

Let us consider a Fourier basis: for all $-N \leq k \leq N$, $e_k := x \in [0, 1] \mapsto e^{2ik\pi x}$.
We can then build the Fourier model:

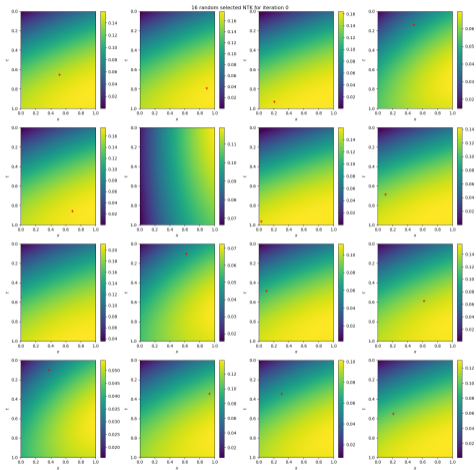
$$S_N : \begin{cases} \mathbb{R}^{\llbracket -N, N \rrbracket} & \rightarrow & L^2([0, 1]) \\ (\alpha_k) & \mapsto & \left(x \in [0, 1] \mapsto \sum_{-N \leq k \leq N} \alpha_k e^{2i\pi kx} \right) \end{cases}$$

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

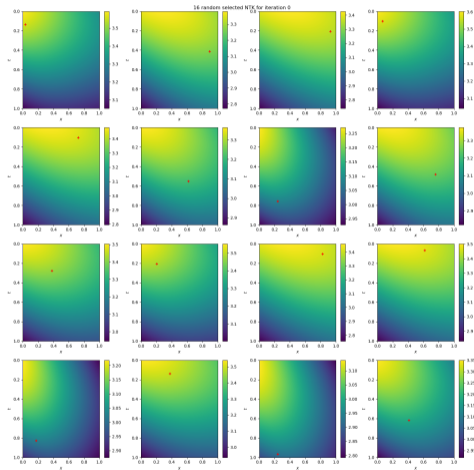
$\Delta[S_N]$ spectrum is $\{k^2 : 1 \leq k \leq N\}$ (Δ : Laplace operator)

Conclusion: Spectral conditioning is strongly impacted by differential operators

Intuition from PINNs NTK



(a) NTK for Laplace equation



(b) NTK for Heat equation

Natural Gradient

Introduced by Amari and Douglas (1998) in the context of Information Geometry. Given a loss: $\ell : \theta \rightarrow \mathbb{R}^+$, the gradient descent:

$$\theta_{t+1} \leftarrow \theta_t - \eta \nabla \ell,$$

is replaced by the update:

$$\theta_{t+1} \leftarrow \theta_t - \eta G_{\theta_t}^\dagger \nabla \ell,$$

with G_{θ_t} the Hessian of the Kullback-Leibler divergence. More generally in the context of Riemannian manifolds:

$$\theta_{t+1} \leftarrow \theta_t - \eta G_{\theta_t}^\dagger \nabla \ell,$$

with $G_{\theta_t p, q} := \mathcal{G}_{\theta_t}(\partial_p u|_{\theta_t}, \partial_q u|_{\theta_t})$, the Gram matrix of partial derivatives w.r.t a Riemannian-(pseudo) metric \mathcal{G}_{θ_t} .

Natural Gradient in a nutshell

Shortcomings

- Computation of the Gram matrix G_{θ_t} is quadratic in the number of parameters.
- Inversion of G_{θ_t} is cubic

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

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

Reinterpreting quadratic loss

Consider the loss of a classical quadratic regression problem, with (x_i) sampled from μ on Ω :

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

In the limit $S \rightarrow \infty$ (population limit), this loss can be reinterpreted as the evaluation on u_θ of the functional loss:

$$\mathcal{L}(u) := \frac{1}{2} \|u - f\|_{L^2(\Omega, \mu)}^2$$

Taking the Fréchet derivative :

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

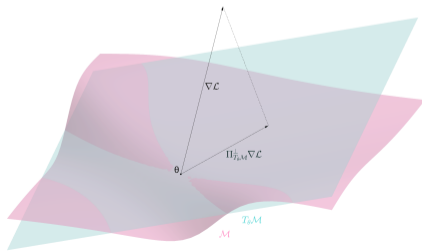
A functional analysis perspective

Reinterpreting natural gradient

- $\mathcal{M} := \text{Im } u = \{u_\theta : \theta \in \mathbb{R}^P\}$
- $T_\theta \mathcal{M} := \text{Im } du|_\theta = \text{Span}(\partial_p u_\theta)$

In the population limit, natural gradient can be reinterpreted as the update:

$$\theta_{t+1} \leftarrow \theta_t - \eta du|_{\theta_t}^\dagger \left(\Pi_{T_{\theta_t} \mathcal{M}}^\perp \nabla \mathcal{L}|_{u_{\theta_t}} \right),$$



Neural Tangent Kernel (NTK)

NTK in a nutshell

Jacot et al. (2018) show that for an empirical quadratic loss:

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

the functional dynamic of the gradient descent on ℓ can be described by:

$$\frac{du_{\theta_t}}{dt}(x) = - \sum_{i=1}^S \text{NTK}_{\theta_t}(x, x_i) (u_{|\theta_t}(x_i) - y_i),$$

with:

$$\text{NTK}_{\theta_t}(x, y) := \sum_{p=1}^P (\partial_p u_{|\theta_t}(x)) (\partial_p u_{|\theta_t}(y))^T$$

Proposition

The functional dynamic of the natural gradient descent on:

$$\ell(\theta) := \sum_{i=1}^S \tau(u_{\theta}(x_i))$$

is described by (Rudner et al., 2019):

$$\frac{du_{\theta_t}}{dt}(x) = - \sum_{i=1}^S \text{NNTK}_{\theta_t}(x, x_i) \tau'(u_{\theta_t}(x_i)),$$

with: $\text{NNTK}_{\theta_t}(x, y) :=$

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

Some consequences

Corollary

The empirical functional dynamics takes place in the empirical tangent space:

$$\hat{T}_\theta \mathcal{M} := \text{Span} (NNTK_\theta(x_i, \cdot) : (x_i)_{1 \leq i \leq N}) \subset T_\theta \mathcal{M}.$$

Corollary

We can define an empirical natural gradient update by:

$$\theta_{t+1} = \theta_t - \eta du_{|\theta_t}^\dagger \left(\Pi_{\hat{T}_{\theta_t} \mathcal{M}}^\perp \nabla \mathcal{L}|_{u|\theta_t} \right).$$

empirical Natural Gradient (eNG)

Reproducing Kernel Hilbert Spaces (RKHS) *détour*

Definition-Proposition

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

- 1 There exist a function $k : \Omega \times \Omega \rightarrow \mathbb{R}$ 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 $x, y \in \Omega$

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

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 $\Pi_{\mathcal{H}_0}^{\perp}$ onto \mathcal{H}_0 is given by:

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

Remark

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

An empirical Natural Gradient computation : ANaGRAM

Theorem (ANaGRAM)

Under mild assumptions, the empirical natural gradient update:

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

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

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

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

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

Remark

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

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

Application to PINNs

PINNs are a quadratic regression problem

Proposition

The only difference between the losses:

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

and:

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

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

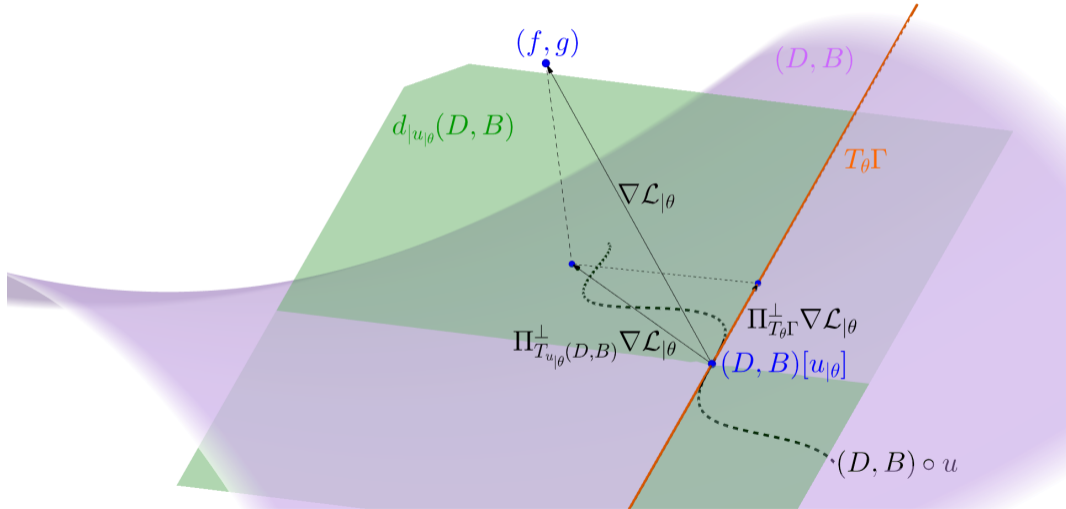
PINNs are a classical quadratic regression problem with model:

$$(D, B) \circ u : \begin{cases} \mathbb{R}^P & \rightarrow \mathcal{H} & \rightarrow L^2(\Omega \rightarrow \mathbb{R}, \mu) \times \\ & & L^2(\partial\Omega \rightarrow \mathbb{R}, \sigma) \\ \theta & \mapsto u_\theta & \mapsto (D[u_\theta], B[u_\theta]) \end{cases}$$

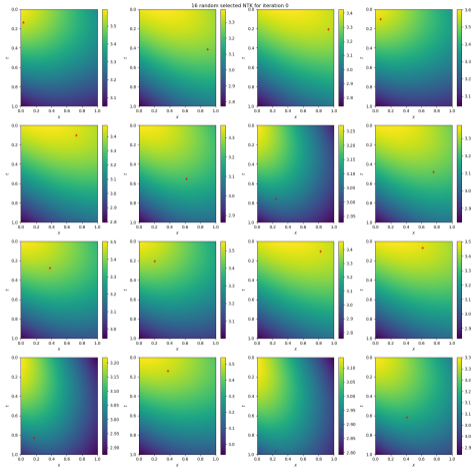
In the same way:

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

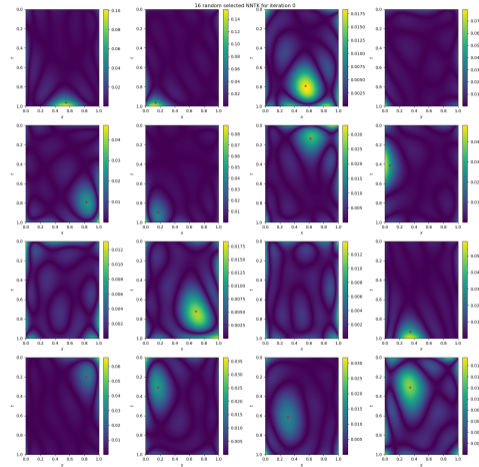
Natural Gradient for PINNs



Natural Gradient for PINNs



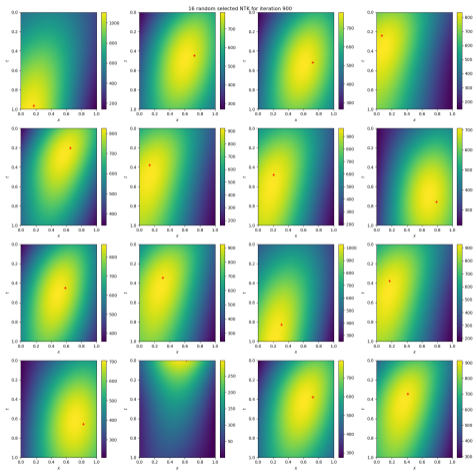
(c) NTK



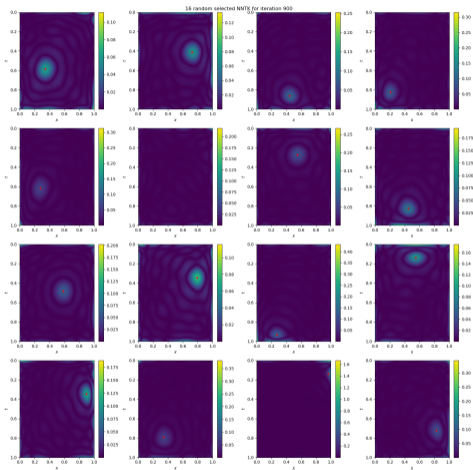
(d) NNTK

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

Natural Gradient for PINNs



(a) NTK



(b) NNTK

Natural Gradient and Green's function

Definition (Green's function of D)

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

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

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 $g : \Omega \times \Omega \rightarrow \mathbb{R}$ such that the operator:

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

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

Natural Gradient and Green's function

Proposition

The Right-inverse operator of the operator D on the space $u_\theta + T_\theta\Gamma$ [resp. $u_\theta + \hat{T}_\theta\Gamma$] is given by:

$$R : f \in L^2(\Omega \rightarrow \mathbb{R}, \mu) \mapsto u_{|\theta} + \int_{\Omega} g(\cdot, s) \left(f(s) - D[u_{|\theta}](s) \right) \mu(ds)$$

where g is the Green's function:

- On $u_\theta + T_\theta\Gamma$: $g(x, y) := \sum_{1 \leq p, q \leq P} \partial_p u_{|\theta}(x) G_{\theta, p, q}^\dagger \partial_q D[u_{|\theta}](y)$
- On $u_\theta + \hat{T}_\theta\Gamma$: $g(x, y) := \sum_{p=1}^P \sum_{i=1}^S \partial_p u_{|\theta}(x) \hat{\phi}_{\theta, p, i}^\dagger \text{NNTK}_\theta(x_i, y)$

Natural Gradient and Green's function

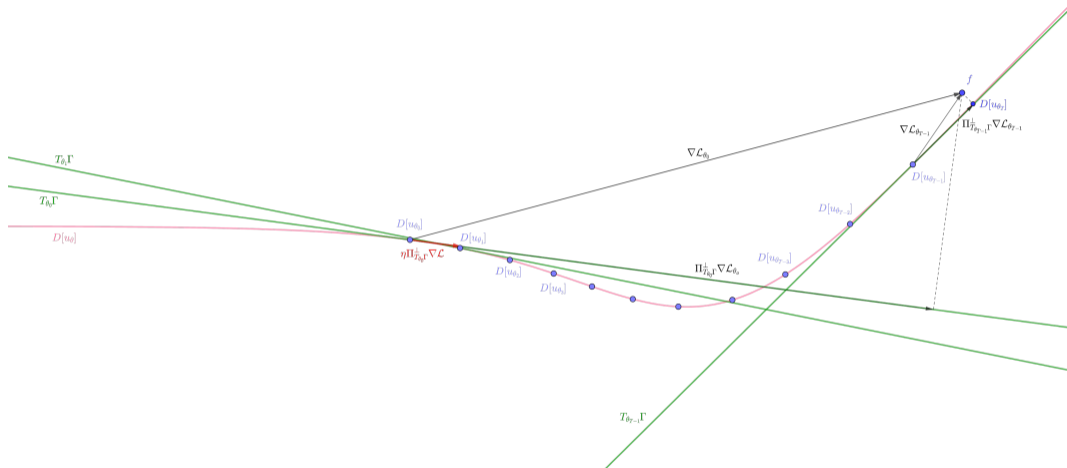


Figure: Illustration of PINNs learning process under natural gradient, as successive applications of Green's function

Experiments

2 D Laplace equation

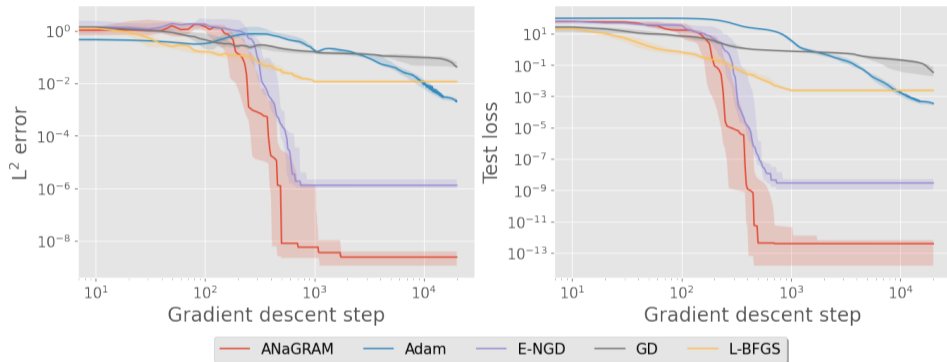


Figure: Performance comparison w.r.t running step for Laplace equation in 2 D:

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

2 D Laplace equation

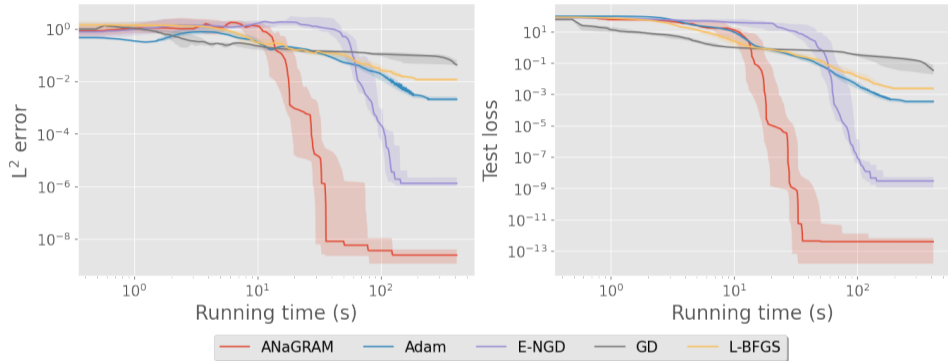


Figure: Performance comparison w.r.t running time for Laplace equation in 2D:

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

1+1 D Heat equation

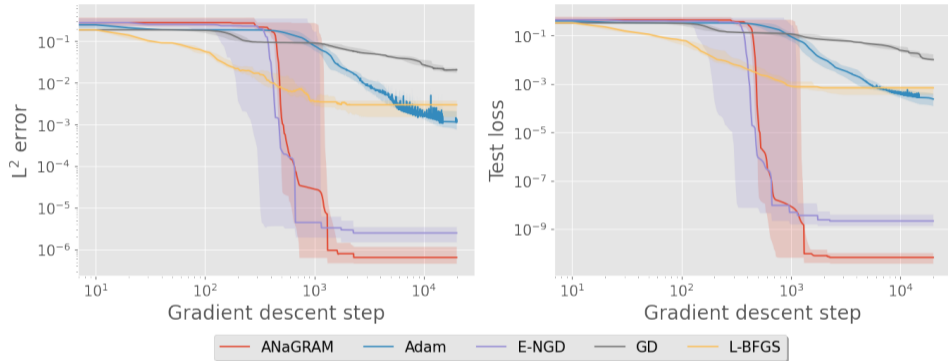


Figure: Performance comparison w.r.t running step for Heat equation in 1+1 D:

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

1+1 D Heat equation

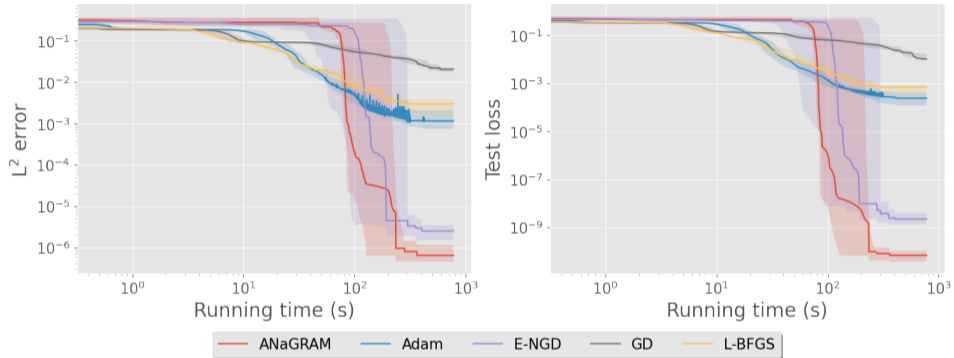


Figure: Performance comparison w.r.t running time for Heat equation in 1+1 D:

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

5 D Laplace equation

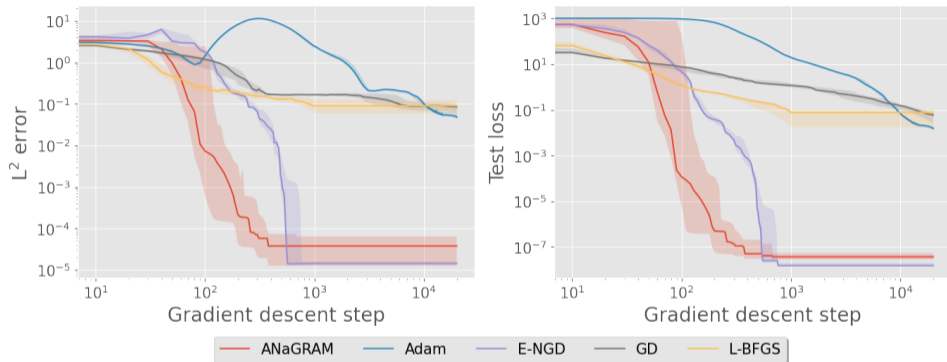


Figure: Performance comparison w.r.t running step for Laplace equation in 5 D:

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

5 D Laplace equation

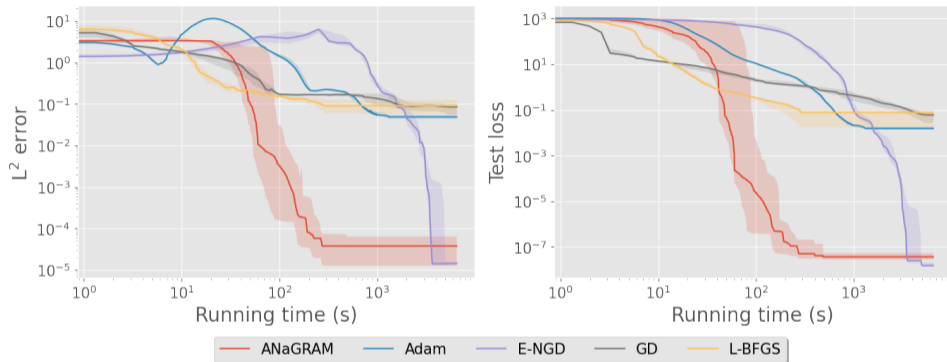


Figure: Performance comparison w.r.t running time for Laplace equation in 5 D:

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

1+1 D Allen-Cahn equation

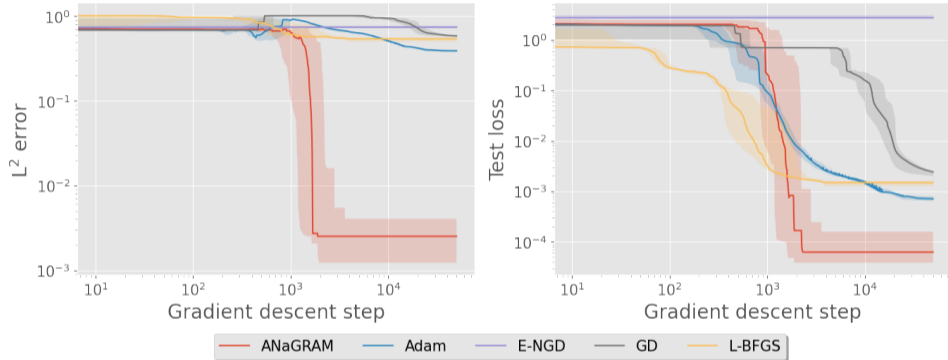


Figure: Performance comparison w.r.t running step for Allen-Cahn equation in 1+1 D:

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

1+1 D Allen-Cahn equation

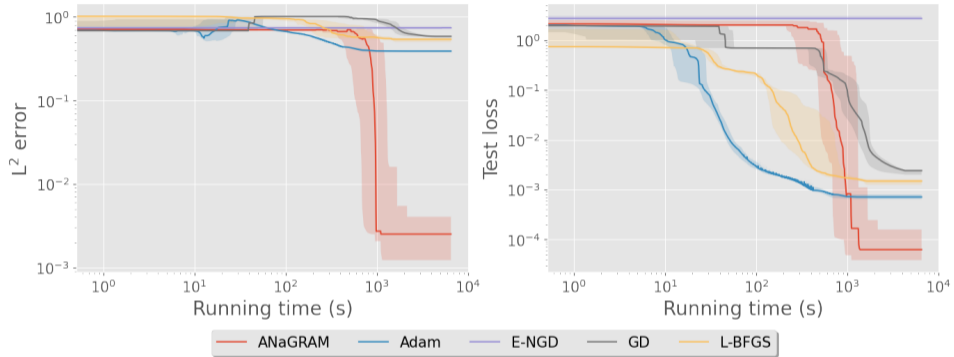


Figure: Performance comparison w.r.t running time for Allen-Cahn equation in 1+1 D:

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

Burgers equation

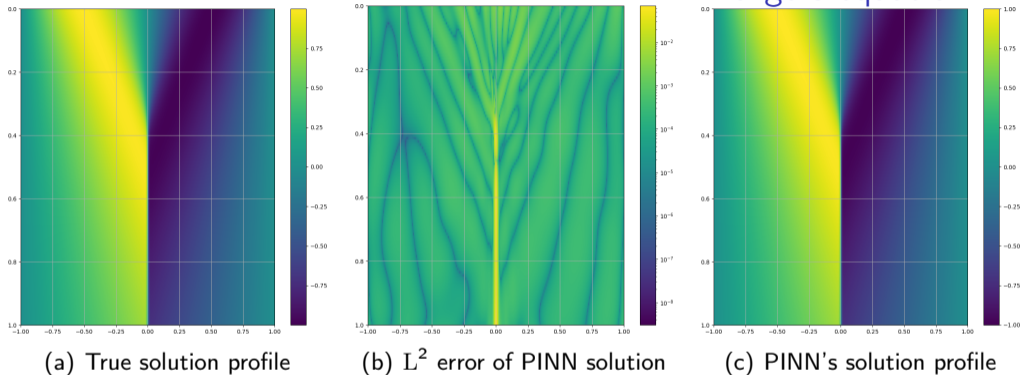


Figure: PINN's solution under Adam (15k steps) + L-BFGS (15k) steps for Burgers equation in 1+1 D:

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

Burgers equation

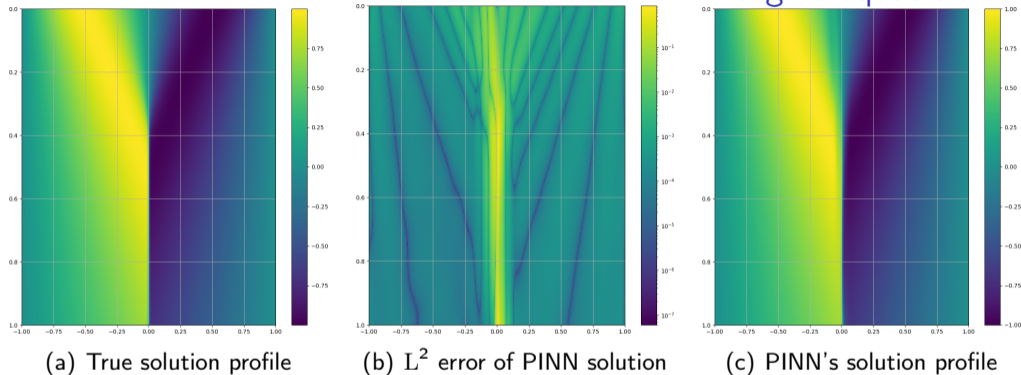


Figure: PINN's solution under Anagram (500 steps) for Burgers equation in 1+1 D:

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

Conclusion and Perspectives

Conclusions

- Anagram gives a theoretically founded simplification 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 assimilation 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

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.

References II

- 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)*.

Natural Gradient for PINNs

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

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

Corollary

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

$$\begin{aligned} NNTK_{\theta}(x, y) &= \sum_{1 \leq p, q \leq P} \partial_p(D, B)[u|_{\theta}](x) G_{\theta_{p,q}}^\dagger \partial_q(D, B)[u|_{\theta}](y) \\ &= \sum_{1 \leq p, q \leq P} (\partial_p D[u|_{\theta}](x_1), \partial_p B[u|_{\theta}](x_2)) G_{\theta_{p,q}}^\dagger (\partial_q D[u|_{\theta}](y_1), \partial_q B[u|_{\theta}](y_2)), \end{aligned}$$

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

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

Corollary empirical Natural Gradient and ANaGRAM for PINNs

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

$$\hat{k}(x, y) = \sum_{1 \leq i, j \leq S} \text{NNTK}_{\theta}(x, x_i) \hat{G}_{\theta i, j}^{\dagger} \text{NNTK}_{\theta}(x_j, y), \text{ where}$$

$$G_{\theta i, j} := \langle \text{NNTK}_{\theta}(\cdot, x_i), \text{NNTK}_{\theta}(x_j, \cdot) \rangle_{L^2(\Omega \rightarrow \mathbb{R}, \mu) \times L^2(\partial\Omega \rightarrow \mathbb{R}, \sigma)} = \text{NNTK}_{\theta}(x_i, x_j)$$

Theorem (ANaGRAM for PINNs)

Under mild assumptions, the empirical natural gradient update:

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

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

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

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

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