ANaGRAM: A Natural Gradient Relative to Adapted Model for efficient PINNs learning

ICLR 2025

Nilo Schwencke, Cyril Furtlehner

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

March 31, 2025

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 $\mathrm{u}_{|\boldsymbol{\theta}}$ on the loss:

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

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

Problem

This leads to low accuracy with SGD.

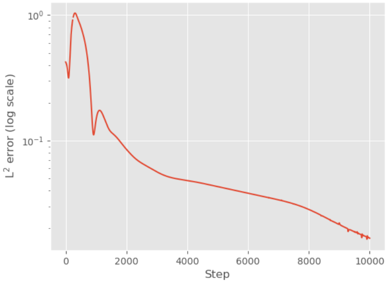


Figure: PINN solution under standard Adam optimization, to Laplace equation in 2 D.

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:

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

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

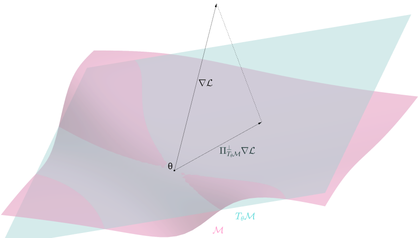
Natural gradient in functional space

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

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):

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

Key Idea

The empirical dynamics takes place in:

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

We can define the empirical Natural Gradient:

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

Theorem (ANaGRAM)

Under mild assumptions:

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

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

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

Key fact

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

Corollary

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

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

Application to 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.

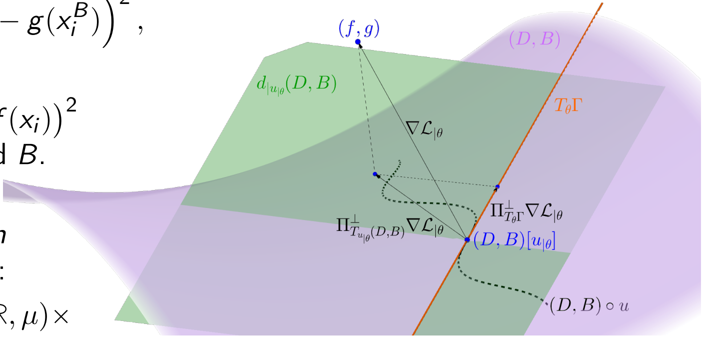
Proposition

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

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

Natural Gradient of PINNs

Figure: Illustration of PINNs Natural Gradient



Experiments

2 D Laplace equation

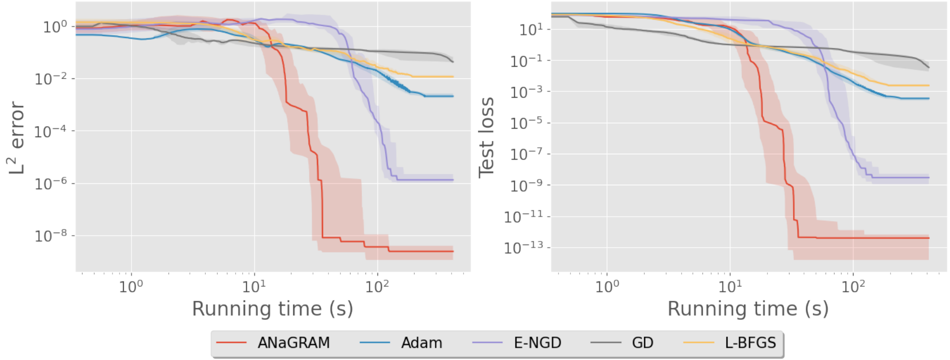


Figure: Performance comparison w.r.t running time for Laplace equation in 2 D:

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

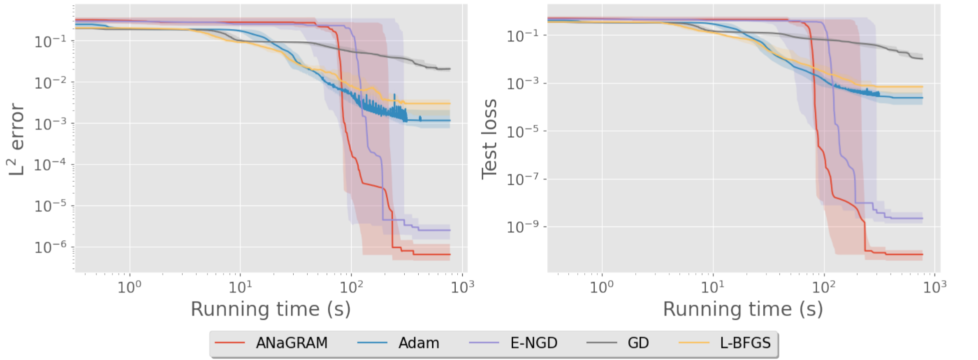


Figure: Performance comparison w.r.t running time for Heat equation in 1+1 D:

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

5 D Laplace equation

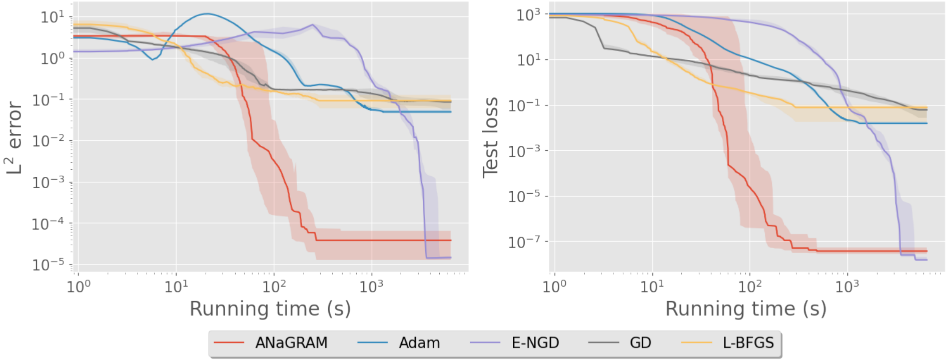


Figure: Performance comparison w.r.t running time for Laplace equation in 5 D:

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

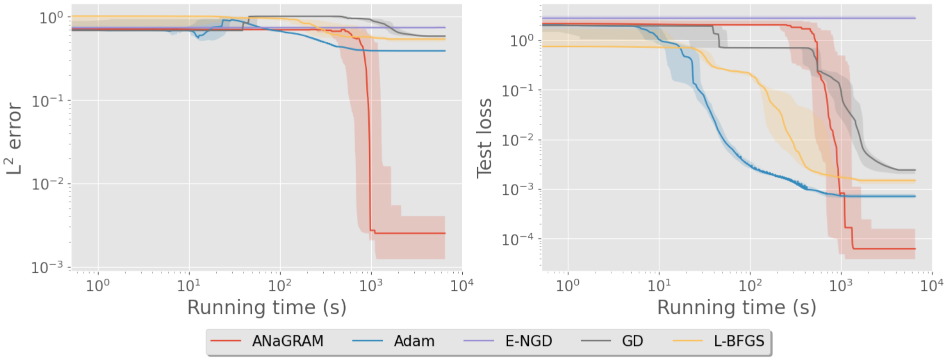


Figure: Performance comparison w.r.t running time for Allen-Cahn equation in 1+1 D:

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

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 !

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*.