

# Kernelization of Natural Gradient Methods for Physics Informed Neural Networks

CRUNCH Seminar

Nilo Schwencke

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

October 10, 2025



# Physics informed neural networks (PINNs)

## Problem statement

We aim to solve:

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

## PINNs key idea

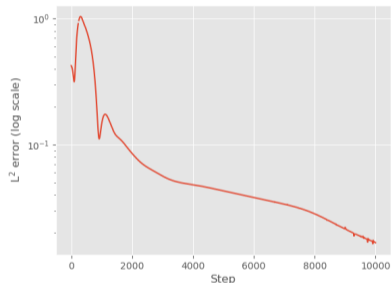
Optimize a neural network  $u|_{\theta}$  on the loss:

$$\begin{aligned} \hat{\ell}_{D,B}(\theta) := & \frac{1}{2S_D} \sum_{i=1}^{S_D} \left( D[u|_{\theta}](x_i^D) - f(x_i^D) \right)^2 \\ & + \frac{1}{2S_B} \sum_{i=1}^{S_B} \left( B[u|_{\theta}](x_i^B) - g(x_i^B) \right)^2, \end{aligned}$$

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

## Problem

**This leads to low accuracy with Adam.**



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

Why does it work so bad ?

Why does it work so bad ?

## Intuition from NTK

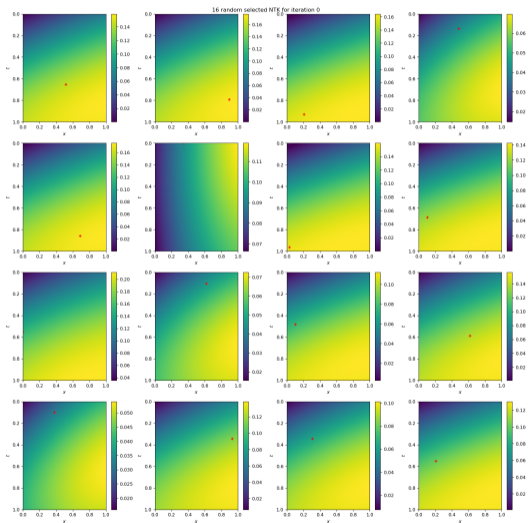


Figure: NTK for Laplace equation.

## Intuition from Fourier

Consider the Fourier partial series:

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

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

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

Conclusion: Differential operators strongly impact the spectral conditioning.

# Natural Gradient

## Reinterpreting quadratic loss

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

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

In the population limit:

$$\hat{\ell}(\boldsymbol{\theta}) \xrightarrow{S \rightarrow \infty} \mathcal{L}(u_{|\boldsymbol{\theta}}); \quad \mathcal{L}(u) := \frac{1}{2} \|u - f\|_{L^2(\Omega)}^2$$

This yields the Fréchet derivative:

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

and thus the gradient flow:

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

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

## A functional geometry perspective

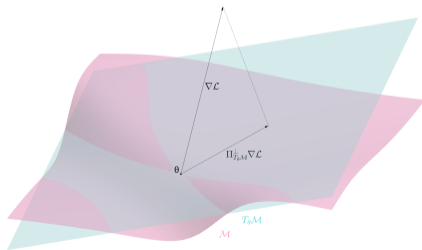
### Natural gradient in functional space

The functional space is constrained to:

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

The Natural Gradient is then defined as:

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



# 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  $x, y \in \Omega$

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

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.

# Computational perspective on Natural Gradient

## Definition-Proposition (Schwencke and Furtlehner (2025))

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

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

## Corollary

The Natural Gradient update rewrites:  $\theta_{t+1} \leftarrow \theta_t - \eta G_{\theta_t}^{\dagger} \nabla \ell(\theta_t)$ ;  $\ell(\theta) := \mathcal{L}(u_{|\theta})$ .

## Shortcomings

- Computation of the Gram matrix  $G_{\theta_t}$  is quadratic in the number of parameters.
- Inversion of  $G_{\theta_t}$  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 functional dynamic of (N)GD on the empirical loss  $\hat{\ell}$  is described by (Jacot et al., 2018):

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

### Key Idea

The empirical dynamics takes place in:

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

We can define the empirical Natural Gradient:

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

### Byproduct

Yields an optimal criterion for  $(x_i)$  choice:

$$(x_i)^{\star} = \underset{(x_i) \in \Omega^S}{\operatorname{argmin}} \left\| \Pi_{\hat{T}_{\theta, K}^{(x_i)} \mathcal{M}}^{\perp} \nabla \mathcal{L}_{|u_{|\theta_t}} - \nabla \mathcal{L}_{|u_{|\theta_t}} \right\|_{L^2(\Omega)}$$

## empirical Natural Gradient

### Theorem (ANaGRAM)

Under mild assumptions:

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

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

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

### Key fact

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

### Corollary

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

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

## Application to PINNs

## Key remark

The only difference between the losses:

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

and  $\hat{\ell}(\theta) := \frac{1}{2S} \sum_{i=1}^S (u|_{\theta}(x_i) - f(x_i))^2$  is the use of the operators  $D$  and  $B$ .

## Proposition

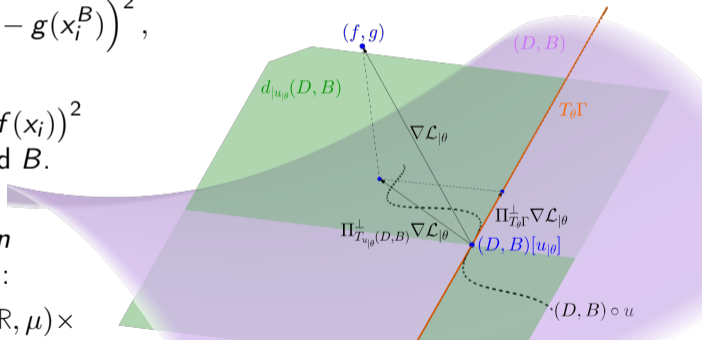
*PINNs are a quadratic regression problem with model:  $(D, B) \circ u$ :*

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

## Application to PINNs

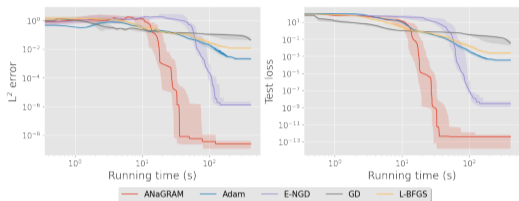
### Natural Gradient of PINNs

Figure: Illustration of PINNs Natural Gradient



# Empirical Evidence for the Natural Gradient Relative to Adapted Model (*ANaGRAM*) Algorithm

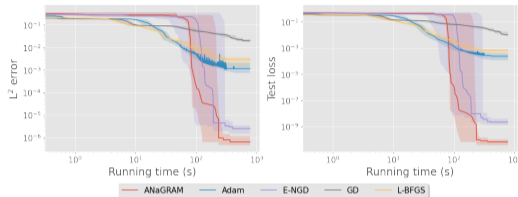
## 2D Laplace equation



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

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

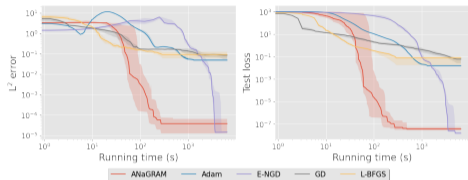
## 1+1 D Heat equation



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

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

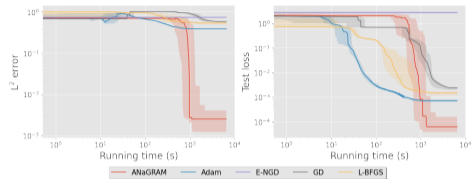
## 5 D Laplace equation



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

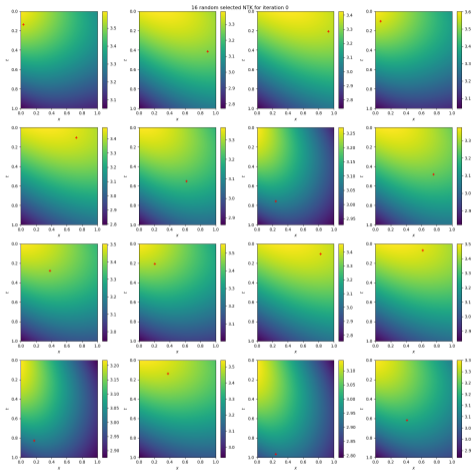
$$\begin{cases} \Delta u = \pi^2 \sum_{k=1}^5 \sin(\pi x_k) & \text{in } \Omega = [0, 1]^5 \\ u = \sum_{k=1}^5 \sin(\pi x_k) & \text{on } \partial\Omega \end{cases} \quad \begin{cases} \partial_t u - 10^{-3} \partial_{xx} u = 5(u - u^3) & \text{in } \Omega = [0, 1] \times [-1, 1] \\ u = -1 & \text{on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

## 1+1 D Allen-Cahn equation

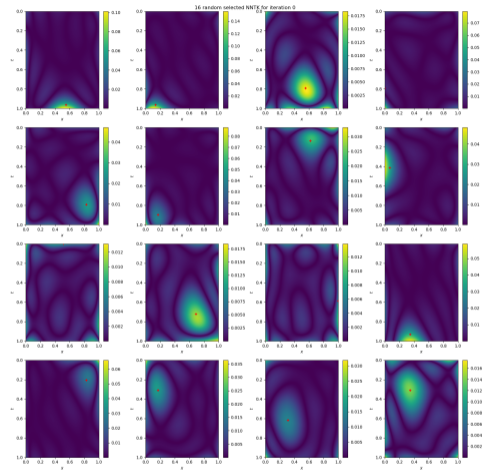


**Figure:** Performance comparison w.r.t running time for Allen-Cahn equation in 1+1 D:

# NTK vs NNTK of PINNs



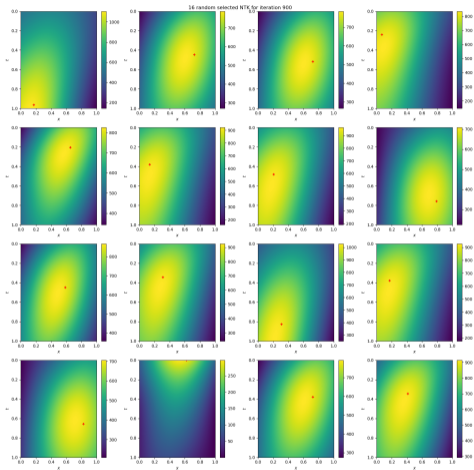
(a) NTK



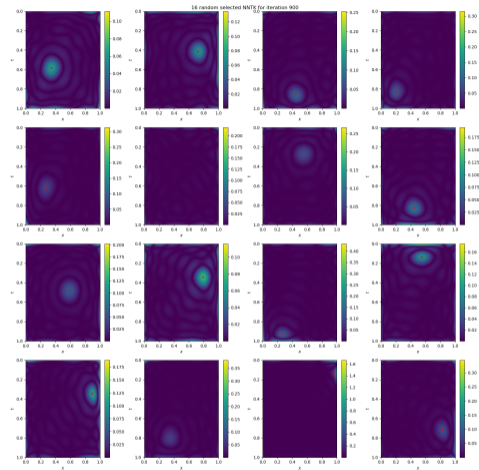
(b) NNTK

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

# NTK vs NNTK of PINNs



(a) NTK



(b) NNTK

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

## In-Depth Empirical Analysis of *Cutoff* Regularization in *ANaGRAM*

# In-Depth Empirical Analysis of *Cutoff* Regularization in ANaGRAM

## SVD pseudoinverse details

$$\hat{\phi}_{\theta} = \hat{V} \hat{\Delta} \hat{U}^{\top}; \quad \hat{\phi}_{\theta_t}^{\dagger} = \hat{U} \hat{\Delta}^{\dagger} \hat{V}^{\top}.$$

Singular values

In practice, we apply a *cutoff*:

$$\hat{\Delta}^{\dagger \alpha} := \begin{cases} \hat{\Delta}_i^{-1} & \text{if } \hat{\Delta}_i \geq \alpha \\ 0 & \text{otherwise} \end{cases},$$

with  $\alpha > 0$  the cutoff level. Thus:

$$\hat{\phi}_{\theta}^{\dagger \alpha} \widehat{\nabla \mathcal{L}}_{\theta} = \sum_{i=1}^{r_{\alpha}} \hat{U}_i \hat{\Delta}_i^{-1} \hat{V}_i^{\top} \widehat{\nabla \mathcal{L}}_{\theta},$$

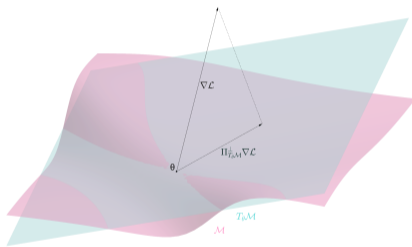
with  $r_{\alpha} := \#\{i : \hat{\Delta}_i \geq \alpha\} \leq \min(P, S)$ .

## Reconstruction Error (RCE)

$$\text{RCE}_n = \frac{1}{\sqrt{S}} \left\| \widehat{\nabla \mathcal{L}}_{\theta} - \sum_{i=1}^n \hat{V}_i \hat{V}_i^{\top} \widehat{\nabla \mathcal{L}}_{\theta} \right\|_{\mathbb{R}^S}$$

## Intuition on RCE

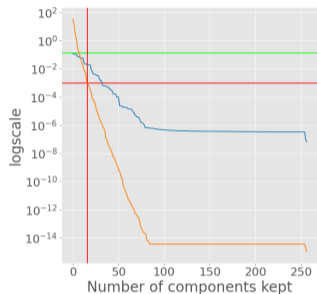
$\text{RCE}_n$  accounts for the part of  $\nabla \mathcal{L}_{\theta}$  orthogonal to the  $n$  most important components of  $\hat{T}_{\theta} \mathcal{M}$ .



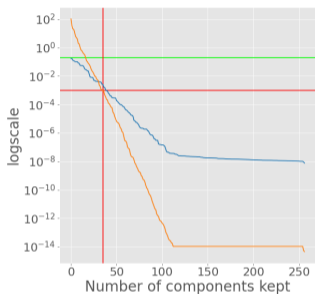
## Remark

- $\lim_{S \rightarrow \infty} \hat{T}_{\theta} \mathcal{M} = T_{\theta} \mathcal{M}$
- $\text{RCE}_0^2 = \frac{1}{S} \left\| \widehat{\nabla \mathcal{L}}_{\theta} \right\|_{\mathbb{R}^S}^2 = \hat{\ell}(\theta)$

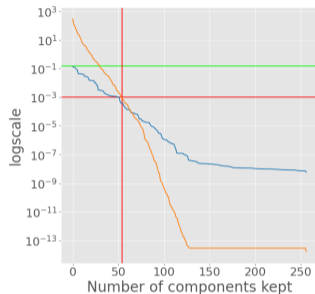
# Empirical insights on the *cutoff* impact in ANaGRAM



(a) Iteration 0: intersection point between singular values and RCE lies before cutoff.



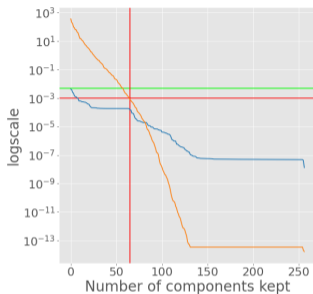
(b) Iteration 40: intersection point shifts rightward toward cutoff.



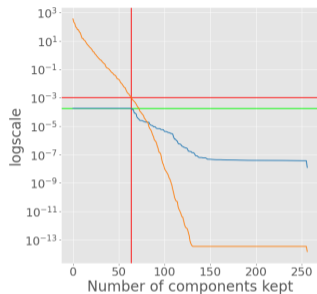
(c) Iteration 90: intersection point passes the cutoff threshold.



# Empirical insights on the *cutoff* impact in ANaGRAM



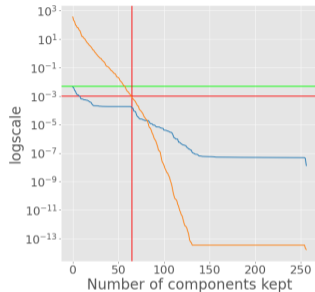
(d) Iteration 120. Beginning of *flattening*: RCE stabilizes at constant level before cutoff.



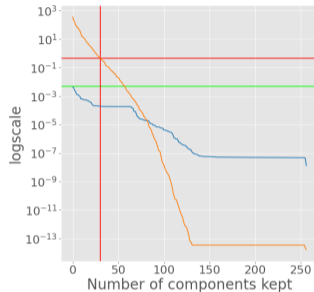
(e) Iteration 150: End of the *flattening phenomenon*. The train loss reaches the flattened part of the RCE.



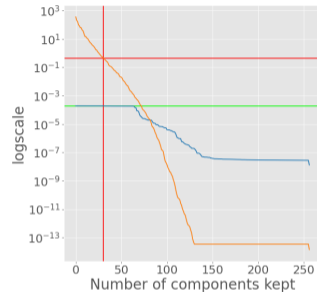
## Incomplete and instant *flattening*



(f) Incomplete flattening of the RCE with a fixed cutoff at  $10^{-3}$ .



(g) New cutoff located roughly at the location of the "elbow" in the RCE curve.



(h) Complete flattening after one natural gradient step with the new cutoff.



# Adaptive Multi-cutoff Strategy Modification for ANaGRAM (*AMStramGRAM*) algorithm

## Outline of the algorithm

---

### Algorithm 1: *AMStramGRAM* (sketch)

---

**Input:**

Initial parameters:  $\theta_0 \in \mathbb{R}^P$

Precision target:  $\epsilon > 0$

1 **repeat**

2      $\hat{U}_t, \hat{\Delta}_t, \hat{V}_t \leftarrow \text{SVD}(\hat{\phi}_t)$

3     Compute  $\text{RCE}_t$

4      $r_\cap \leftarrow \#\{n : \text{RCE}_{t_n} \geq \hat{\Delta}_{t_n}\}$

5      $r_\epsilon \leftarrow \#\{n : \text{RCE}_{t_n} \geq \epsilon\}$

6     Apply ANaGRAM with cutoff rank

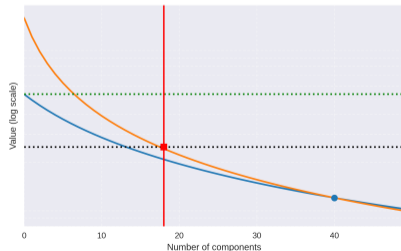
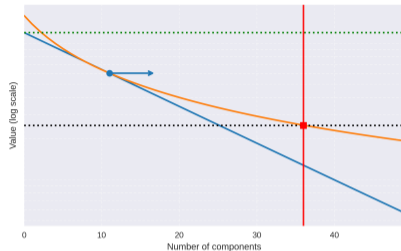
$r_\alpha \leftarrow \min(r_\cap, r_\epsilon)$

7      $t \leftarrow t + 1$

8 **until**  $r_\epsilon = 0$  or  $t \geq T_{\max}$

**Output:**  $\theta_t$

---



---

## Algorithm 1: *AMStramGRAM* (sketch)

---

**Input:**

Initial parameters:  $\theta_0 \in \mathbb{R}^P$

Precision target:  $\epsilon > 0$

1 **repeat**

2      $\hat{U}_t, \hat{\Delta}_t, \hat{V}_t \leftarrow \text{SVD}(\hat{\phi}_t)$

3     Compute  $\text{RCE}_t$

4      $r_\cap \leftarrow \#\{n : \text{RCE}_{t_n} \geq \hat{\Delta}_{t_n}\}$

5      $r_\epsilon \leftarrow \#\{n : \text{RCE}_{t_n} \geq \epsilon\}$

6     Apply ANaGRAM with cutoff rank

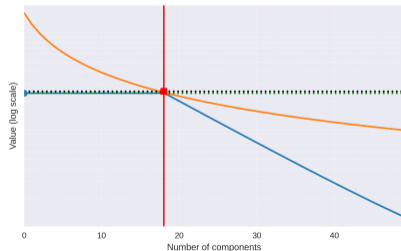
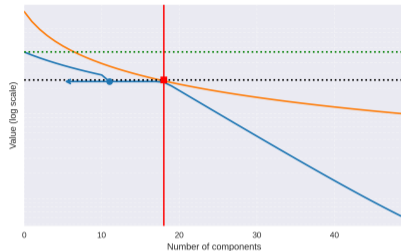
7      $r_\alpha \leftarrow \min(r_\cap, r_\epsilon)$

7      $t \leftarrow t + 1$

8 **until**  $r_\epsilon = 0$  or  $t \geq T_{\max}$

**Output:**  $\theta_t$

---



Experiment	Train Loss		$L_2$ Error	
	AMStramGRAM	ANaGRAM	AMStramGRAM	ANaGRAM
Heat Equation	<b>6.29e-29</b> $\pm$ <b>6.78e-30</b>	8.56e-11 $\pm$ 7.05e-11	<b>2.32e-14</b> $\pm$ <b>1.14e-14</b>	1.28e-06 $\pm$ 1.75e-06
Laplace 2D	<b>1.46e-28</b> $\pm$ <b>1.87e-29</b>	4.27e-13 $\pm$ 4.66e-13	<b>2.24e-15</b> $\pm$ <b>2.52e-16</b>	3.49e-09 $\pm$ 3.58e-09
Laplace 5D	<b>2.04e-08</b> $\pm$ <b>1.16e-08</b>	6.37e-08 $\pm$ 7.01e-08	<b>2.12e-05</b> $\pm$ <b>8.15e-06</b>	4.00e-05 $\pm$ 2.93e-05
Allen–Cahn	<b>3.19e-11</b> $\pm$ <b>2.37e-11</b>	2.19e-04 $\pm$ 4.16e-04	<b>5.87e-05</b> $\pm$ <b>6.25e-06</b>	4.32e-03 $\pm$ 5.93e-03

Experiment	Train Loss		$L_2$ Error	
	AMStramGRAM	SSBroyden	AMStramGRAM	SSBroyden
Burgers (1+1 D)	<b>2.99e-12</b> $\pm$ <b>9.26e-13</b>	2.92e-10 $\pm$ 1.45e-10	<b>1.5e-06</b> $\pm$ <b>9.43e-7</b>	1.59e-06 $\pm$ 1.02e-6
Non-Linear Poisson	<b>8.51e-24</b> $\pm$ <b>2.24e-24</b>	3.03e-16 $\pm$ 3.82e-16	6.81e-10 $\pm$ 1.41e-09	<b>9.29e-12</b> $\pm$ <b>5.85e-12</b>
Allen–Cahn	3.19e-11 $\pm$ 2.37e-11	<b>6.42e-12</b> $\pm$ <b>5.52e-12</b>	5.87e-05 $\pm$ 6.25e-06	<b>3.94e-06</b> $\pm$ <b>1.72e-06</b>

# Overfitting

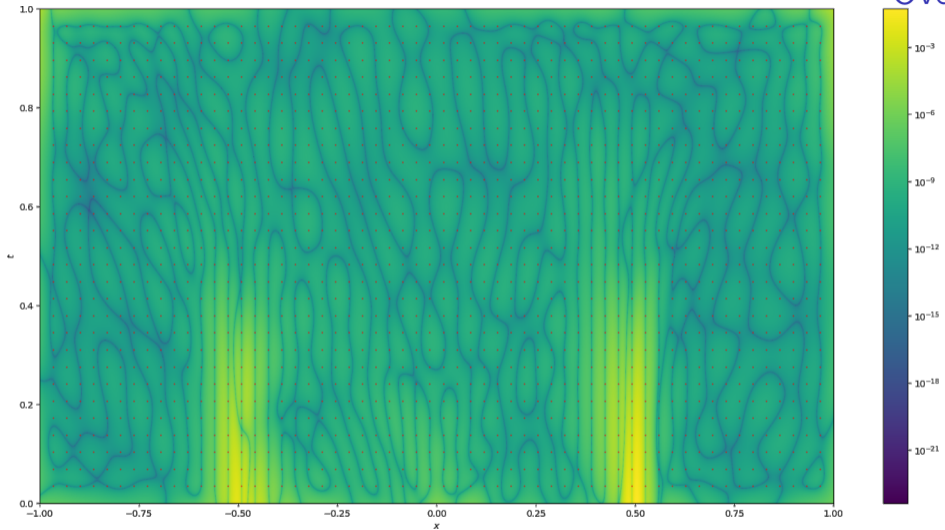


Figure: Overfitting on Allen–Cahn: residual lines align with sampling lines.

## Overfitting

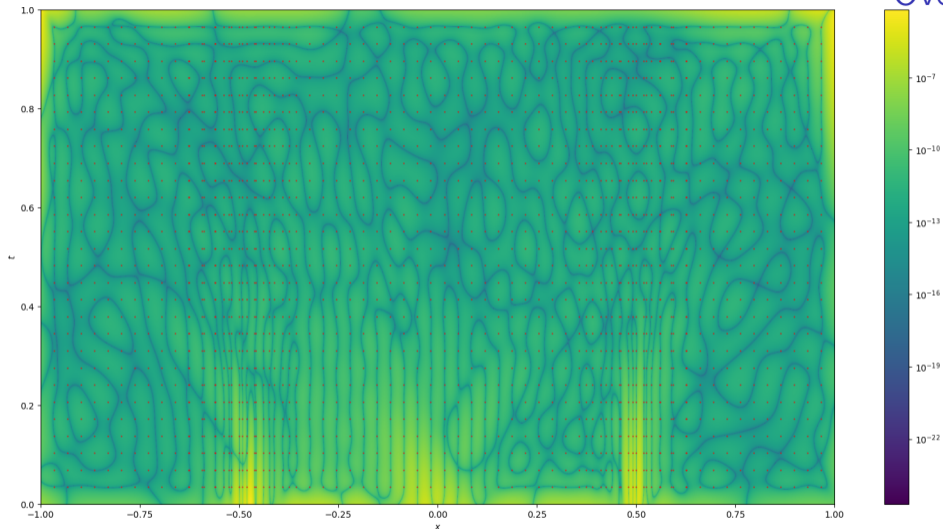


Figure: Overfitting on Allen–Cahn: densifying the sampling in overfitted regions mitigates overfitting.

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

## Theorem

Let  $D : \mathcal{H} \rightarrow L^2(\Omega \rightarrow \mathbb{R}, \mu)$  be a linear differential operator and  $\mathcal{H}_0 \subset \mathcal{H}$  an RKHS with kernel  $k_0$ . Given the spectral decomposition:

$$\Pi_{\mathcal{H}_0} D^* D \Pi_{\mathcal{H}_0} = \int_0^{+\infty} \lambda \pi_{D, \mathcal{H}_0}(d\lambda) \quad I_{\mathcal{H}_0} = \int_0^{+\infty} \pi_{D, \mathcal{H}_0}(d\lambda)$$

Then the generalized Green's function at regularization level  $\alpha > 0$  is given by:  
for all  $x, y \in \Omega$

$$g_{\mathcal{H}_0, \alpha}(x, y) := D \left[ \int_{\alpha^2}^{+\infty} \lambda^{-1} \pi_{D, \mathcal{H}_0}(d\lambda) [k(x, \cdot)] \right] (y)$$

# Natural gradient of PINNs is a Greens's function

## Theorem

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

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

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

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

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

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

## 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  $\min(P, N)$ .
- AMStramGRAM gives a principled way to adapt cutoff reaching machine-level error.
- We prove that PINNs natural gradient corresponds to an optimal linear update following the Green's function.
- Empirical results are competitive with state-of-the-art PINNs optimizers.

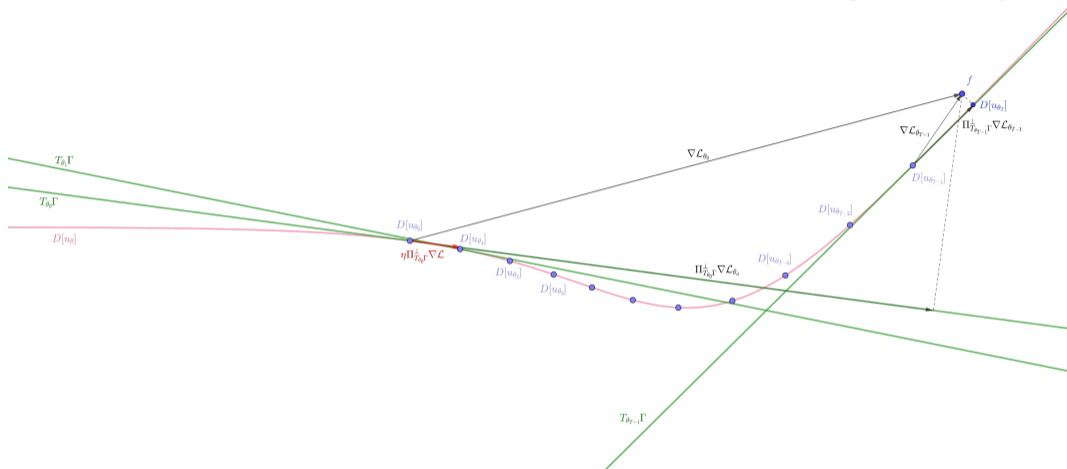
## Perspectives

- Design of an optimal collocation points procedure, coupled with AMStramGRAM's cutoff 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 it to Operator learning

## Thank you for your attention !

- Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.
- Raissi, M., Perdikaris, P., and Karniadakis, G. (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.
- Schwencke, N. and Furtlehner, C. (2025). ANaGRAM: A natural gradient relative to adapted model for efficient PINNs learning.
- Schwencke, N., Rousselot, C., Shilova, A., and Furtlehner, C. (2025). AMStramGRAM: Adaptive multi-cutoff strategy modification for anagram.
- Urbán, J. F., Stefanou, P., and Pons, J. A. (2025). Unveiling the optimization process of physics informed neural networks: How accurate and competitive can PINNs be? *Journal of Computational Physics*, 523:113656.

# Illustration of natural gradient dynamics



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

## Corollary

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

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

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

## Theorem (ANaGRAM for PINNs)

Under mild assumptions, the empirical natural gradient update:

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

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  $1 \leq p \leq P, 1 \leq i \leq S$

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