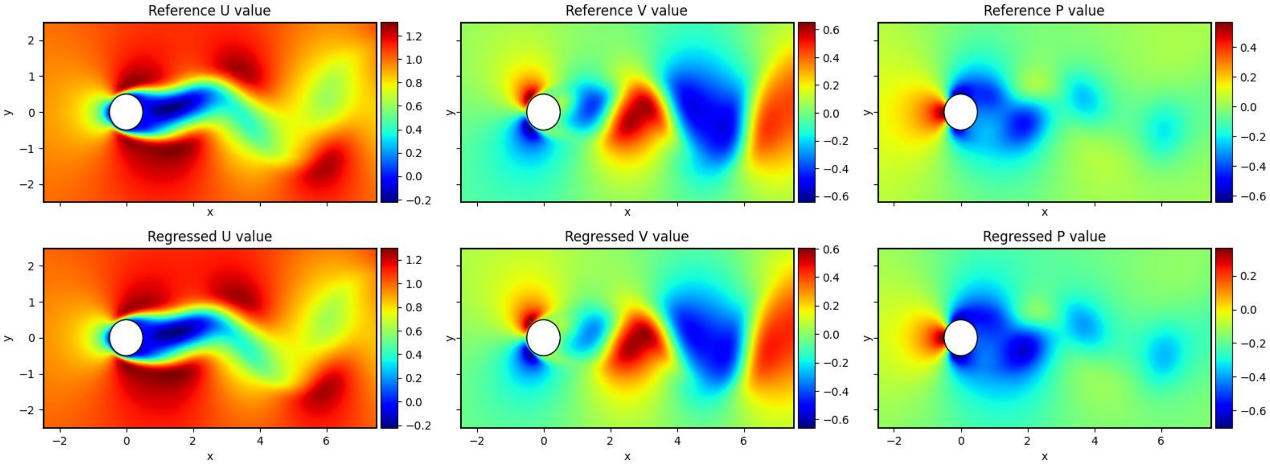
THE CHALLENGES OF THE NONLINEAR REGIME FOR PHYSICS-INFORMED NEURAL NETWORKS

A. Bonfanti, G. Bruno, C. Cipriani

PHYSICS-INFORMED NEURAL NETWORKS

Partial Differential Equation (**PDE**) on a domain Ω:

$$\begin{aligned} \mathcal{R}u(x) &= f(x), \quad x \in \Omega, \\ u(x) &= g(x), \quad x \in \partial\Omega. \end{aligned}$$



Approximate the PDE solution with a neural network (**PINN**) u_θ

The solution minimizes:

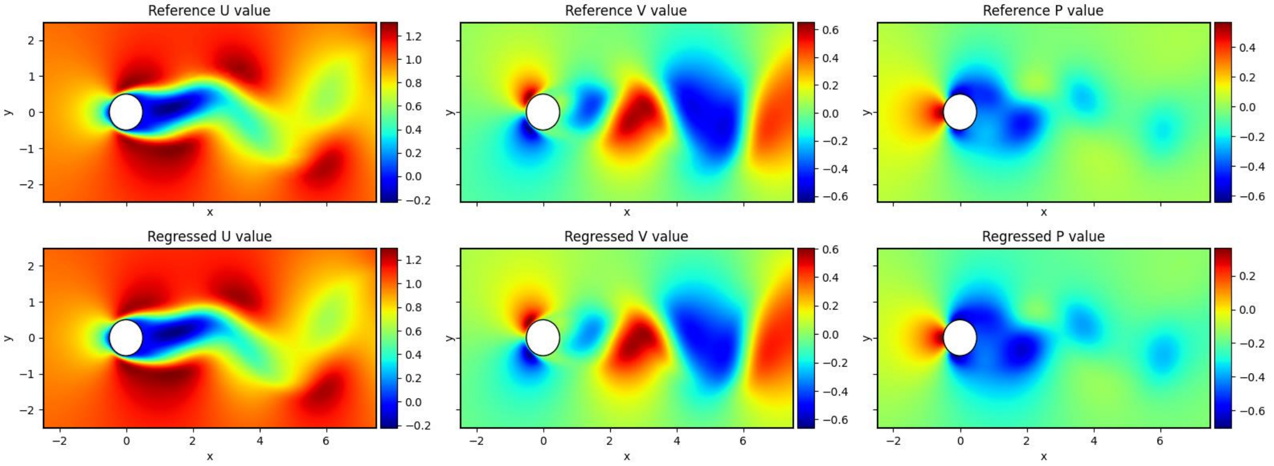
$$\mathcal{L}(\theta) = \frac{1}{2}\int_\Omega |\mathcal{R}u_\theta(x) - f(x)|^2 dx + \frac{1}{2}\int_{\partial\Omega} |u_\theta(x) - g(x)|^2 d\sigma(x)$$

PHYSICS-INFORMED NEURAL NETWORKS

Partial Differential Equation (**PDE**) on a domain Ω:

$$\mathcal{R}u(x) = f(x), \quad x \in \Omega,$$
$$u(x) = g(x), \quad x \in \partial\Omega.$$

Approximate the PDE solution with a neural network (**PINN**) u_θ



The solution minimizes:

$$L(\theta) = \frac{1}{2N_r}\sum_{i=1}^{N_r} |r_\theta(x_i^r)|^2 + \frac{1}{2N_b}\sum_{i=1}^{N_b} |u_\theta(x_i^b) - g(x_i^b)|^2$$

THE NEURAL TANGENT KERNEL OF PINNS

PINN with *m* parameters and NTK rescaling:

$$u_\theta(x) := \frac{1}{\sqrt{m}} W^1 \cdot \sigma(W^0 x + b^0) + b^1$$

Training the parameters of PINNs can be interpreted as a gradient flow:

$$\partial_t \theta(t) = -\nabla L(\theta(t))$$

Infinite-width limit

Consider $J(t) = \begin{bmatrix} \partial_\theta u_{\theta(t)}(\mathbf{x}^b) \\ \partial_\theta r_{\theta(t)}(\mathbf{x}^r) \end{bmatrix}$

$$K(t) = J(t) J(t)^T$$

Is the Neural Tangent Kernel (**NTK**)

The following equation holds:

$$\begin{bmatrix} \partial_t u_{\theta(t)}(\mathbf{x}^b) \\ \partial_t r_{\theta(t)}(\mathbf{x}^r) \end{bmatrix} = -K(t) \begin{bmatrix} u_{\theta(t)}(\mathbf{x}^b) - g(\mathbf{x}^b) \\ r_{\theta(t)}(\mathbf{x}^r) \end{bmatrix}$$

The loss decays as:

$$L(\theta(t)) \leq (1 - \eta\mu)^t L(\theta(0))$$

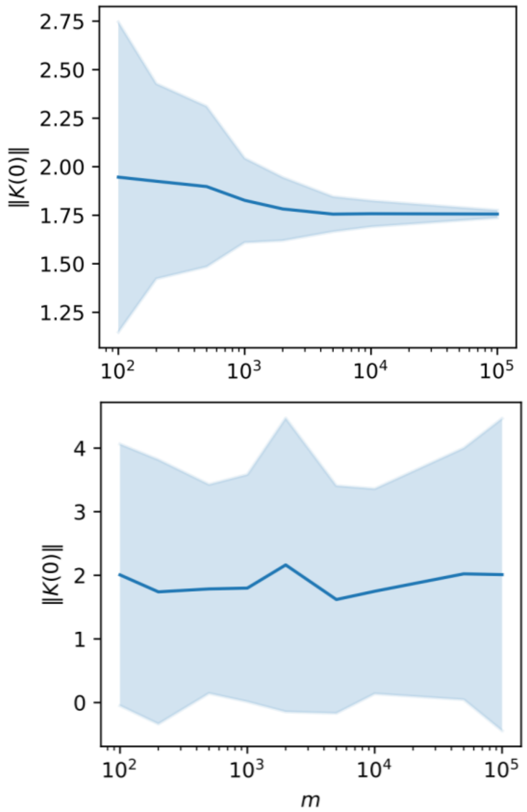
WHAT ABOUT THE NONLINEAR REGIME?

	Linear PDEs	Nonlinear PDEs
NTK at initialization	*Deterministic*	*Random*

Almost sure convergence of the NTK at initialization fails with nonlinear PDEs.

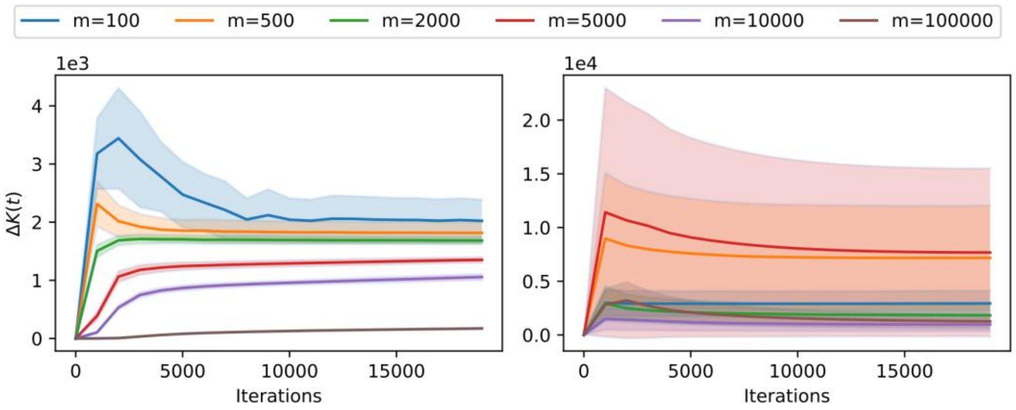
We prove **convergence in law** to a stochastic variable, and its law can be explicitly determined.

$$K(0) \xrightarrow{\mathcal{D}} \bar{K} \quad as\ m \to \infty$$



WHAT ABOUT THE NONLINEAR REGIME?

	Linear PDEs	Nonlinear PDEs
NTK at initialization	*Deterministic*	*Random*
NTK during training	*Constant*	*Dynamic*



Even under generous assumptions, we show that the **constancy of the NTK during training does not hold** for general nonlinear PDEs.

$$\lim_{m\to\infty} \sup_{t\in[0,T]} \|K(t) - K(0)\| > 0 \quad a.s.$$

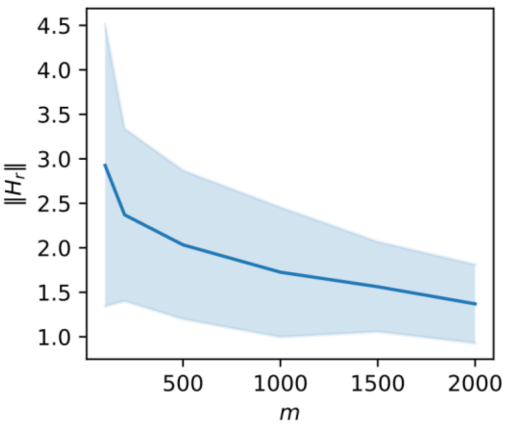
WHAT ABOUT THE NONLINEAR REGIME?

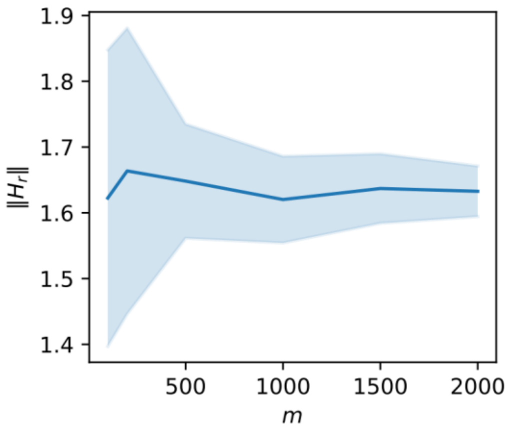
	Linear PDEs	Nonlinear PDEs
NTK at initialization	*Deterministic*	*Random*
NTK during training	*Constant*	*Dynamic*
Hessian H_r	*Sparse*	*Not sparse*

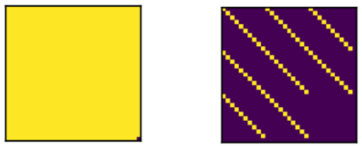
Traditional proofs of the constancy of the NTK fail.

We prove that the **Hessian of the residuals does not vanish**.

$$\lim_{m\to\infty} ||H_r|| \geq \tilde{c}$$







$H_r(0)$ nonlinear $H_r(0)$ linear

TRAINING WITH SECOND-ORDER METHODS

	Linear PDEs	Nonlinear PDEs
NTK at initialization	*Deterministic*	*Random*
NTK during training	*Constant*	*Dynamic*
Hessian H_r	*Sparse*	*Not sparse*

First order:

Gradient flow: $$\partial_t \theta(t) = -\nabla L(\theta(t))$$

Training dynamics: $$\begin{bmatrix} \partial_t u_{\theta(t)}(\mathbf{x}^b) \\ \partial_t r_{\theta(t)}(\mathbf{x}^r) \end{bmatrix} = -K(t) \begin{bmatrix} u_{\theta(t)}(\mathbf{x}^b) \\ r_{\theta(t)}(\mathbf{x}^r) \end{bmatrix}$$

With K being the NTK

Second order:

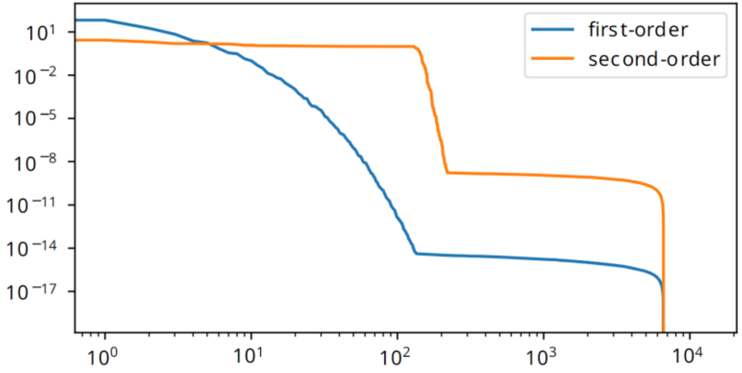
"Gauss-Newton" flow: $$\partial_t \theta(t) = -(J^T(t)J(t))^{\dagger} \nabla L(\theta(t))$$

Training dynamics: $$\begin{bmatrix} \partial_t u_{\theta(t)}(\mathbf{x}^b) \\ \partial_t r_{\theta(t)}(\mathbf{x}^r) \end{bmatrix} = -U(t)D(t)U(t)^T \begin{bmatrix} u_{\theta(t)}(\mathbf{x}^b) \\ r_{\theta(t)}(\mathbf{x}^r) \end{bmatrix}$$

With U unitary, and D diagonal with entries 0 or 1.

TRAINING WITH SECOND-ORDER METHODS

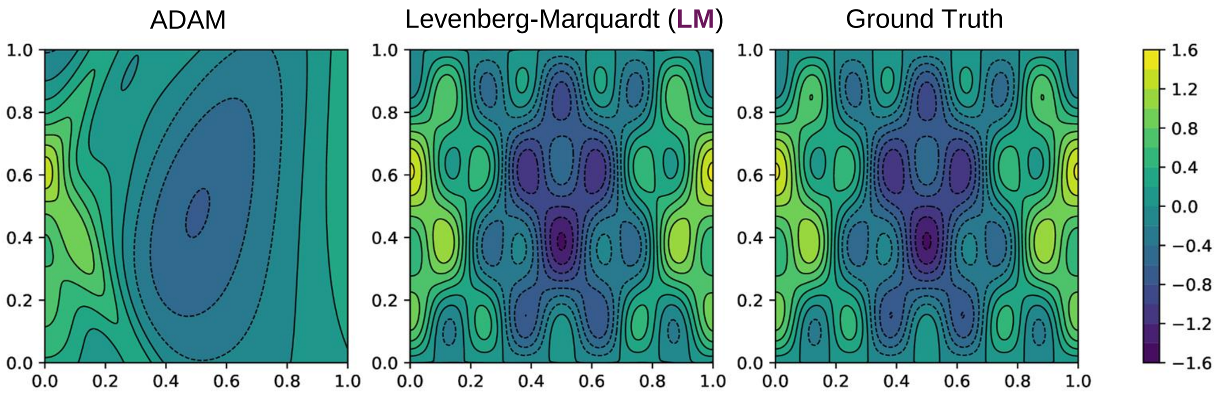
	Linear PDEs	Nonlinear PDEs
NTK at initialization	*Deterministic*	*Random*
NTK during training	*Constant*	*Dynamic*
Hessian H_r	*Sparse*	*Not sparse*
I-order convergence bound	$\sim \lambda_{min}(K)$	~ 0 or $\sim \lambda_{min}(K(t))$
II-order convergence bound	~ 1	~ 0 or ~ 1



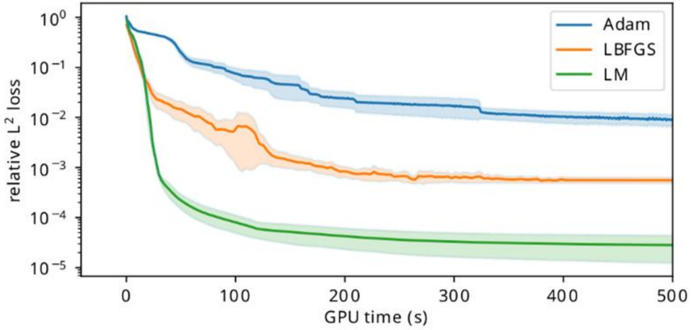
While ensuring **fast convergence**, second-order methods mitigate the issue of **spectral bias** when training PINNs on PDEs containing high-frequency components.

TRAINING WITH SECOND-ORDER METHODS

WAVE EQUATION
(linear, spectrally biased)

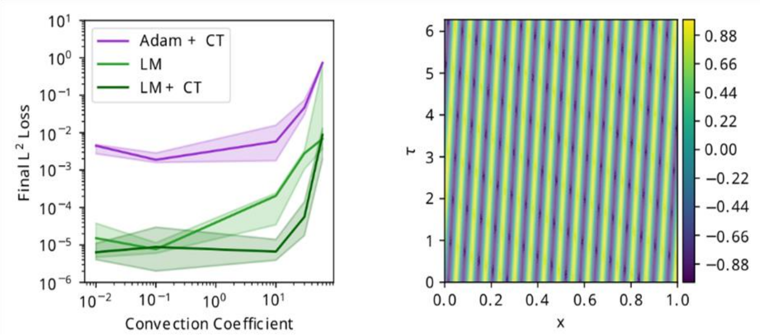


BURGER EQUATION
(nonlinear)

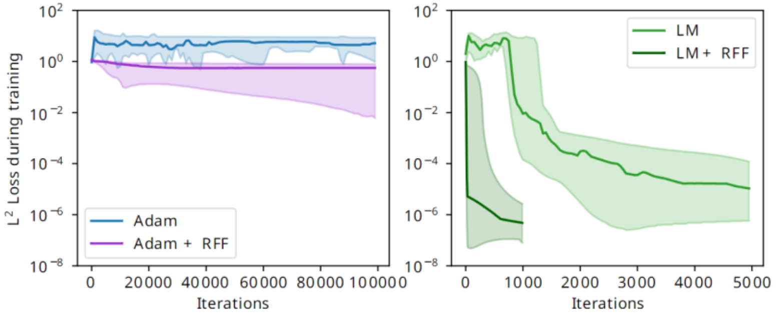


TRAINING WITH SECOND-ORDER METHODS

ADVECTION EQUATION
(linear, Curriculum Training)



POISSON EQUATION
(linear, Random Fourier Features)



NAVIER STOKES' EQUATIONS
(nonlinear, Causality-based training)