Toward a Unified Framework for PINNs and FEMs

A Petrov–Galerkin perspective through the lens of kernel theory

47[e] CANUM — Saint-Jacut-de-la-Mer

Nilo Schwencke

joint work with Roland Maier

LIP – ENS Lyon • INRIA Lyon • CNRS

Institute for Applied and Numerical Mathematics – KIT

June 02, 2026

ENS DE LYON

Karlsruher Institut für Technologie

société de mathématiques appliquées et industrielles

Outline

The two realms

Galerkin / FEM

- **Variational**: weak form on a mesh
- Stiffness matrix assembled *once*
- Sparse, **well-conditioned** (elliptic)
- Mature **error theory**

PINNs

- **Collocation** of the *strong* residual
- **Mesh-free**, flexible, high-d
- Few DoFs can match FEM accuracy…
- …but **training** is the bottleneck

The question

Two communities, two languages — yet natural-gradient PINNs now reach FEM-level accuracy. **Can we put both in a single framework, and *couple* them during training?**

Answer: yes — both are **projected (generalized) Green functions**.

Boundary value problem

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

Elliptic model: $-\mathrm{div}(A\nabla u) = f$, $u|_{\partial\Omega} = 0$.

Energy form

$$a(u, w) := \langle A\nabla u \, , \, \nabla w \rangle_{\mathrm{L}^2(\Omega)}$$

Galerkin

Seek $u_p \in V_p \subset \mathrm{H}^1_0(\Omega)$:

$$a(u_p, w_p) = \langle f \, , \, w_p \rangle_{\mathrm{L}^2(\Omega)} \quad \forall w_p \in V_p$$

Petrov–Galerkin

Test space $W_q \neq V_p$:

$$a(u_p, w_q) = \langle f \, , \, w_q \rangle_{\mathrm{L}^2(\Omega)} \quad \forall w_q \in W_q$$

Keep the trial/test distinction in mind — it is the whole story.

PINNs and Natural gradients

PINNs loss

$$\mathcal{L}_{D,B}(u) := \int_{\Omega} \|D[u]-f\|^2 + \int_{\partial\Omega} \|B[u]-g\|^2$$

PINNs key idea (Lagaris et al.1998; Raissi et al.2019)

- Model u with a neural network u_θ
- Use autodiff to compute D and B
- Minimize the sampled loss:

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

Optimization is hard

Usual optimizers (Adam, L-BFGS) often stagnate far from the true solution. Why?

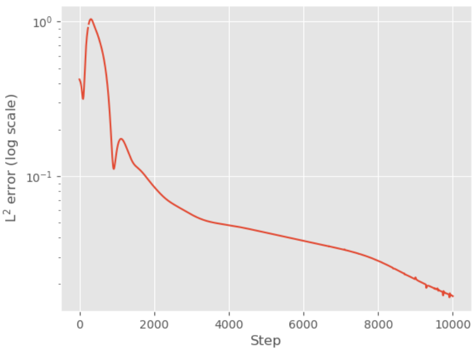


Figure: L^2-error of a PINN optimized with Adam on the 2D Laplace equation.

Natural gradient in functional space

Parametric model

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

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

Quadratic regression

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

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

Induces the gradient flow:

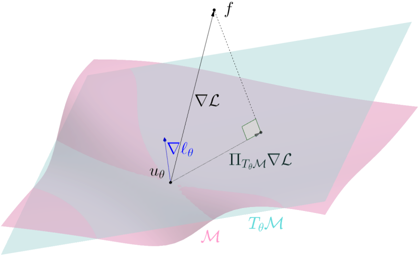
$$\begin{cases} u_0 \in \mathrm{L}^2(\Omega) \\ \dot{u}_t = -\nabla \mathcal{L}_{|u_t} = f - u_t \end{cases}$$

<u>Solution:</u> $u_t = f - e^{-t}(f - u_0)$.

But u_θ lives on the manifold $\mathcal{M}$.
$\Rightarrow$ Project $\nabla \mathcal{L}$ onto $T_\theta \mathcal{M}$.

Then (Amari and Douglas1998):

$$\theta_{t+1} \leftarrow \theta_t - \eta \, \mathrm{d}u^\dagger_{\theta_t} \left(\Pi^\perp_{T_{\theta_t}\mathcal{M}} \nabla \mathcal{L}_{|u_{\theta_t}} \right) ,$$



$$\theta_{t+1} \leftarrow \theta_t - \eta \, G^\dagger_\theta \nabla_\theta \ell(\theta); \quad G_{\theta_{pq}} = \langle \partial_p u_\theta \, , \, \partial_q u_\theta \rangle_{\mathcal{H}} \, ;$$

$$\ell(\theta) := \mathcal{L}(u_\theta).$$

NNTK (Rudner et al.2019) in a nutshell

Discretized loss: $\widehat{\ell}(u) := \frac{1}{2S}\sum_{i=1}^{S}(u(x_i) - f(x_i))^2$

The functional dynamics under NGD is then:

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

with: $K(x,y) := \sum_{p,q} \partial_p u_\theta(x) G^\dagger_{\theta_{pq}} \partial_q u_\theta(x)$[a].

Key Observation

The empirical dynamics takes place in:

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

We can define the empirical Natural Gradient:

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

[a] Under GD, $G_\theta = I_p$ and we get the usual NTK (Jacot et al.2018)

Theorem (ANaGRAM)

Under mild assumptions:

$$du^\dagger_{\theta_t}\left(\Pi^\perp_{\hat{T}_{\theta_t}\mathcal{M}} \nabla\mathcal{L}_{|u_{\theta_t}}\right) \simeq \hat{\phi}^\dagger_{\theta_t}\widehat{\nabla\mathcal{L}}_{\theta_t},$$

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

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

Corollary

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

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

Key remark

The only difference between the losses:

$$\mathcal{L}_{D,B}(u) = \int_\Omega \|D[u] - f\|^2 + \int_{\partial\Omega} \|B[u] - g\|^2$$

and $\mathcal{L}(u) = \frac{1}{2}\|u - f\|^2_{\mathrm{L}^2(\Omega)}$ is the use of the operators D and B.

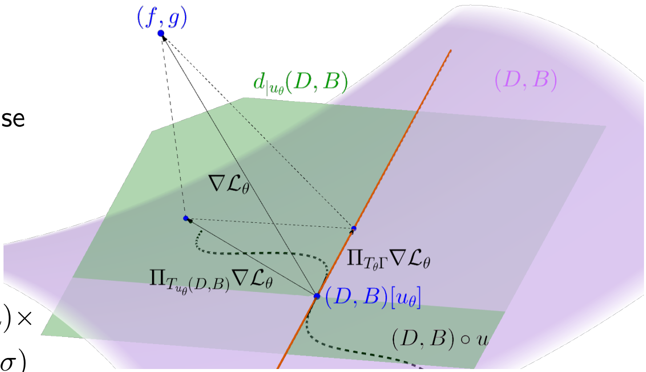
Proposition

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

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

Natural Gradient of PINNs

Figure: Illustration of PINNs Natural Gradient



1+1 D Heat equation

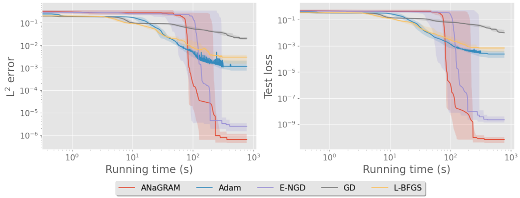


Figure: Performance comparison w.r.t running time for Heat equation in 1+1 D:

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

1+1 D Allen-Cahn equation

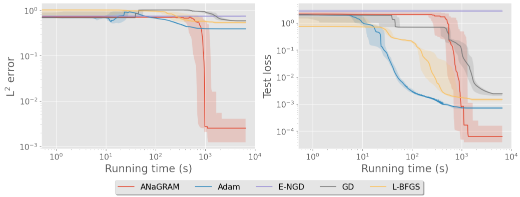


Figure: Performance comparison w.r.t running time for Allen-Cahn equation in 1+1 D:

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

Note: E-NGD refers to Müller and Zeinhofer2023).

Empirical validation

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

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

* refers to the order two method of Urbán et al.2025), with adaptive sampling and hard constraint enforcement on boundary conditions.

Unification through Petrov–Galerkin test-space choices

ANaGRAM and Natural Gradient as a Petrov–Galerkin method

Assume for simplicity that BC are enforced.

Petrov–Galerkin approach

Idea: at first order, choose $h \in \mathbb{R}^P$ s.t:

$$\mathrm{d}\Gamma_\theta(h) \simeq f - \Gamma_\theta; \quad \Gamma_\theta := D[u_\theta].$$

Choosing $\mathcal{T} \subset \mathrm{L}^2(\Omega)$, we could impose

$$\langle \Gamma_\theta + \mathrm{d}\Gamma_\theta(h) - f \, , \, \tau \rangle_{\mathrm{L}^2(\Omega)} = 0, \; \forall \tau \in \mathcal{T}$$

- **Moving solution space:** $T_\theta \Gamma = \operatorname{Im} \mathrm{d}\Gamma_\theta$.
- **Test space:** can be freely chosen.

All previous methods are recovered as specific test-space choices

method	test space
exact NG	$\mathrm{Span}\{\partial_p \Gamma_\theta\}$
empirical NG	$\mathrm{Span}\{K_\theta(\cdot, x_i)\}$
ANaGRAM	$\mathrm{Span}\{\widetilde{K}_\theta(\cdot, x_i)\}$

with $\widetilde{K}_\theta$, the reproducing kernel of $\widetilde{T}\Gamma_\theta := \mathrm{Span}\{\partial_p \Gamma_\theta, f - \Gamma_\theta\}$.

Variational methods are already here

Choosing the test functions $\zeta \in \mathrm{H}_0^1(\Omega)$ recovers the **weak formulation** world:

$$\begin{aligned} &\langle \Gamma_\theta + \mathrm{d}\Gamma_\theta(h) - f \,,\, \zeta_i \rangle_{\mathrm{L}^2(\Omega)} = 0 \\ \iff &a(u_\theta + \mathrm{d}u_\theta(h), \zeta_i) = \langle f \,,\, \zeta_i \rangle_{\mathrm{L}^2(\Omega)} \end{aligned}$$

The whole variational picture is one instance of the same construction.

Green's function tests $\Rightarrow$ interpolation

Assume there is a Green's function $\mathfrak{g}_A$.
Taking $\zeta_i = \mathfrak{g}_A(x_i, \cdot)$ then yields

$$\begin{aligned} &\langle \Gamma_\theta + \mathrm{d}\Gamma_\theta(h) - f \,,\, \mathfrak{g}_A(x_i, \cdot) \rangle_{\mathrm{L}^2(\Omega)} = 0 \\ \iff &u_\theta(x_i) + \mathrm{d}u_\theta\big(h\big)(x_i) = u^\star(x_i), \end{aligned}$$

where $u^\star$ is the PDE solution, *i.e.*

$$u^\star(x) := \int_\Omega \mathfrak{g}_A(x, s) f(s) \mathrm{d}s.$$

Turns into a simple interpolation of the solution.

A familiar special case

In 1D, the Green kernel of the Dirichlet Laplacian is $\mathfrak{g}(x, y) = \min(x, y) - xy$. Its sections at the mesh nodes span exactly the P_1 space, and the usual **hat functions are the cardinal interpolants** of the solution at the nodes (Owhadi and Scovel2019).

A hybrid FEM–neural method

Idea

Split the approximation

$$u_{\theta,h} = \underbrace{v_\theta}_{\text{network}} + \underbrace{u_h(\theta)}_{\text{FEM part}}.$$

where $u_h(\theta) \in V_h$. The FEM is designed to capture the coarse part on V_h; the network is trained **only on the complement** $V_h^{\perp,a}$.

How it works

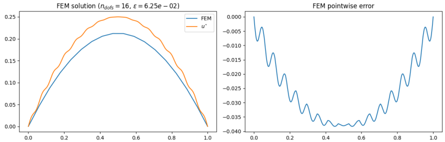
The FEM part removes the V_h component of the weak residual, so after eliminating it the neural update acts through the **projected tangent directions** $\Pi^a_{V_h^\perp} \partial_p v_\theta$.

One iteration (high level)

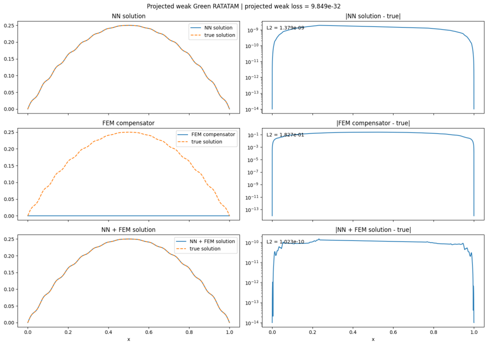
1. **FEM solve**: compute $u_h(\theta)$ with $a(v_\theta + u_h, w_h) = \langle f\,,\, w_h\rangle$, $w_h \in V_h$.
2. Sample weak tests, project them onto $V_h^{\perp,a}$: $\zeta_i^\perp = \Pi^a_{V_h^\perp}\zeta_i$.
3. Assemble test features
$$\widehat{\phi}_{pi} := a(\partial_p v_\theta, \zeta_i^\perp)$$
and test residuals
$$\widehat{\nabla\mathcal{L}}_i := a(u_\theta, \zeta_i^\perp) - \langle f\,,\, \zeta_i^\perp\rangle_{\mathrm{L}^2(\Omega)}$$
4. **Update** θ through ANaGRAM, AMStraMGRAM, RATATAM, etc.
5. **Recompute** u_h for the new θ.

Preliminary numerical experiments

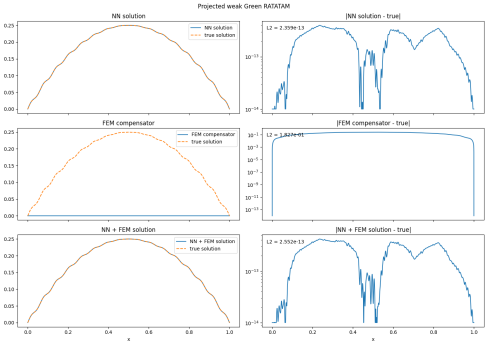
$$A(\varepsilon)(x) = \frac{1}{2+\cos\left(\frac{2\pi x}{\varepsilon}\right)}$$



(a) FEM reference solution (16 dofs)



(b) Test functions : Green's function of $-\mathrm{div}\big(A(\varepsilon)\nabla(u)\big)$



(c) Test functions : Green's function of the Laplacian

1. FEM and natural-gradient PINNs fits into the same framework. They differ only on the solution space choice.
2. Differences between PINN's natural gradient methods correspond to different test space choices in the same Petrov–Galerkin method.
3. There are optimal choices (Green's functions) : **test space is a design lever**.
4. The **hybrid PINN-FEM method** uses that lever: **FEM for the coarse operator-adapted structure, the network for the complement**.

Thank you for your attention ! Questions welcome.

Schwencke & Maier, *A kernelized Petrov–Galerkin framework for hybrid FEM–neural PDE solvers* **(soon available)**.

ScimBa

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.

Müller, J. and M. Zeinhofer (2023): "Achieving High Accuracy with PINNs via Energy Natural Gradient Descent," in *International Conference on Machine Learning*, PMLR, 25471–25485.

Owhadi, H. and C. Scovel (2019): *Operator-Adapted Wavelets, Fast Solvers, and Numerical Homogenization: From a Game Theoretic Approach to Numerical Approximation and Algorithm Design*, vol. 35, Cambridge University Press.

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)*.

Urbán, J. F., P. Stefanou, and J. A. Pons (2025): "Unveiling the Optimization Process of Physics Informed Neural Networks: How Accurate and Competitive Can PINNs Be?" *Journal of Computational Physics*, 523, 113656.