June 4, 2026

Enriching continuous Lagrange finite element approximation spaces using neural networks

Michel Duprez[1], Emmanuel Franck[2], **Frédérique Lecourtier**[1] and Vanessa Lleras[3]

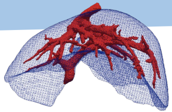
[1]Project-Team MIMESIS, Inria, Strasbourg
[2]Project-Team MACARON, Inria, Strasbourg
[3]IMAG, University of Montpellier, Montpellier

Joint work with:
H. Barucq, F. Faucher, N. Victorion and V. Michel-Dansac.

MS : "Perspectives et avancées récentes en apprentissage pour la résolution numérique d'équations aux dérivées partielles"

CANUM 2026

Scientific context



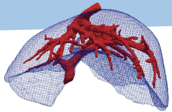
Context: Create real-time digital twins of an organ (e.g. liver).

Objective: Develop an hybrid finite element / neural network method.

finite element: accurate

neural network: quick + parameterized

Scientific context



Context: Create real-time digital twins of an organ (e.g. liver).

Objective: Develop an hybrid finite element / neural network method.

accurate — quick + parameterized

Parametric Poisson problem (with homogeneous Dirichlet BC):

For one or several $\boldsymbol{\mu} \in \mathcal{M}$, find $u : \Omega \to \mathbb{R}$ such that

$$\begin{cases} -\Delta u(\boldsymbol{x};\boldsymbol{\mu}) = f(\boldsymbol{x};\boldsymbol{\mu}), & (\boldsymbol{x},\boldsymbol{\mu}) \in \Omega \times \mathcal{M}, \\ u(\boldsymbol{x};\boldsymbol{\mu}) = 0, & (\boldsymbol{x},\boldsymbol{\mu}) \in \partial\Omega \times \mathcal{M}, \end{cases} \qquad (\mathcal{P})$$

with $\Omega \subset \mathbb{R}^d$ a domain (d the spatial dimension).

Classical approaches

Physics-Informed Neural Networks

Standard PINNs: Find the optimal weights $\theta^\star$, such that

$$\theta^\star = \underset{\theta}{\operatorname{argmin}} \left(\omega_r J_r(\theta) + \omega_b J_b(\theta)\right), \qquad (\mathcal{P}_\theta)$$

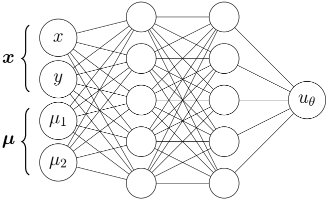
with

residual loss
$$J_r(\theta) = \int_{\mathcal{M}} \int_{\Omega} \left|\Delta u_\theta(\boldsymbol{x}, \boldsymbol{\mu}) + f(\boldsymbol{x}, \boldsymbol{\mu})\right|^2 d\boldsymbol{x} d\boldsymbol{\mu},$$

boundary loss
$$J_b(\theta) = \int_{\mathcal{M}} \int_{\partial\Omega} \left|u_\theta(\boldsymbol{x}, \boldsymbol{\mu})\right|^2 d\boldsymbol{x} d\boldsymbol{\mu},$$

with ω_r and ω_b are some weights.

u_θ is a neural network, e.g. fully-connected NN.
(see example in 2D with 2 parameters)



Monte-Carlo method: Discretize the cost functions by random process.

Finite Element Method[1]

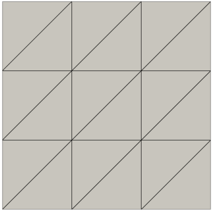
Variational Problem:

$$\text{Find } u_h \in V_h^0 \text{ such that, } \forall v_h \in V_h^0, a(u_h, v_h) = l(v_h), \qquad (\mathcal{P}_h)$$

with h the characteristic mesh size, a and l the bilinear and linear forms given by

$$a(u_h, v_h) = \int_\Omega \nabla u_h \cdot \nabla v_h, \quad l(v_h) = \int_\Omega f v_h,$$

and V_h^0 a continuous, piecewise polynomial space of degree k.



Cartesian mesh

[1] [Ern and Guermond, 2004]

Finite Element Method[1]

Variational Problem:

$$\text{Find } u_h \in V_h^0 \text{ such that, } \forall v_h \in V_h^0, a(u_h, v_h) = l(v_h), \qquad (\mathcal{P}_h)$$

with *h* the characteristic mesh size, *a* and *l* the bilinear and linear forms given by

$$a(u_h, v_h) = \int_\Omega \nabla u_h \cdot \nabla v_h, \quad l(v_h) = \int_\Omega f v_h,$$

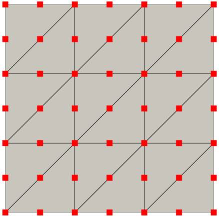
and V_h^0 a continuous, piecewise polynomial space of degree k.

Linear system: Let $(\phi_1, \dots, \phi_{N_{\text{dofs}}})$ a basis of V_h^0.
Find $U \in \mathbb{R}^{N_{\text{dofs}}}$ such that

$$AU = b,$$

$$A = \big(a(\phi_i, \phi_j)\big)_{1 \le i,j \le N_{\text{dofs}}} \in \mathbb{R}^{N_{\text{dofs}} \times N_{\text{dofs}}},$$

$$b = \big(l(\phi_j)\big)_{1 \le j \le N_{\text{dofs}}} \in \mathbb{R}^{N_{\text{dofs}}}.$$



degrees of freedom $(k = 2)$

[1] [Ern and Guermond, 2004]

Finite Element Method[1]

Variational Problem:

$$\text{Find } u_h \in V_h^0 \text{ such that, } \forall v_h \in V_h^0, a(u_h, v_h) = l(v_h), \qquad (\mathcal{P}_h)$$

with h the characteristic mesh size, a and l the bilinear and linear forms given by

$$a(u_h, v_h) = \int_\Omega \nabla u_h \cdot \nabla v_h, \quad l(v_h) = \int_\Omega f v_h,$$

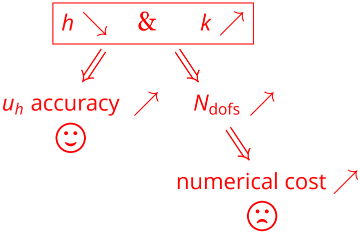
and V_h^0 a continuous, piecewise polynomial space of degree k.

Linear system: Let $(\phi_1, \ldots, \phi_{N_{\text{dofs}}})$ a basis of V_h^0.
Find $U \in \mathbb{R}^{N_{\text{dofs}}}$ such that

$$AU = b,$$

$$A = \left(a(\phi_i, \phi_j)\right)_{1 \le i,j \le N_{\text{dofs}}} \in \mathbb{R}^{N_{\text{dofs}} \times N_{\text{dofs}}},$$

$$b = \left(l(\phi_j)\right)_{1 \le j \le N_{\text{dofs}}} \in \mathbb{R}^{N_{\text{dofs}}}.$$



[1] [Ern and Guermond, 2004]

Enriched finite element method using PINNs

Pipeline of the Enriched FEM

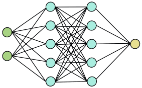
OFFLINE : PINN training

Inputs

$\boldsymbol{x} \in \Omega$: space coordinates

$\boldsymbol{\mu} \in \mathcal{M}$: parameter

$\longrightarrow$



Parametric PINN

$\longrightarrow$

Output

$u_\theta(\boldsymbol{x}, \boldsymbol{\mu})$: prediction of the PDE solution $u(\boldsymbol{x}, \boldsymbol{\mu})$

Pipeline of the Enriched FEM

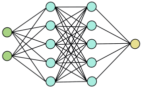
OFFLINE : PINN training

Inputs

$\boldsymbol{x} \in \Omega$: space coordinates

$\boldsymbol{\mu} \in \mathcal{M}$: parameter

→



Parametric PINN

→

Output

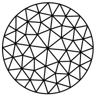
$u_\theta(\boldsymbol{x}, \boldsymbol{\mu})$: prediction of the PDE solution $u(\boldsymbol{x}, \boldsymbol{\mu})$

ONLINE (fixed $\boldsymbol{\mu}$) : PINN evaluation + Enriched FEM resolution

PINN evaluation

$u_\theta(\cdot, \boldsymbol{\mu})$: prediction for the given parameter μ

→



FEM solver enriched by u_θ

→

Output

enriched FEM solution (depending on the mesh size h)

Complete ONLINE process : quick + accurate

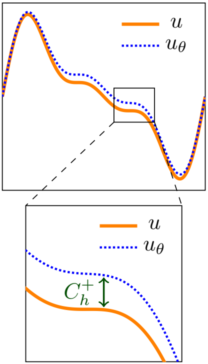
Additive approach

Variational Problem: Let $u_\theta \in H^{k+1}(\Omega) \cap H_0^1(\Omega)$ be a PINN prior.

$$\text{Find } C_h^+ \in V_h^0 \text{ such that, } \forall v_h \in V_h^0, a(C_h^+, v_h) = l(v_h) - a(u_\theta, v_h), \qquad (\mathcal{P}_h^+)$$

with the enriched trial space V_h^+ defined by

$$V_h^+ = \left\{ u_h^+ = u_\theta + C_h^+, \quad C_h^+ \in V_h^0 \right\}.$$



General Dirichlet BC: If $u = g$ on $\partial\Omega$, then

$$C_h^+ = g - u_\theta \quad \text{on } \partial\Omega,$$

with u_θ the PINN prior.

We expect that the modified problem will give the same results as the standard one on coarser meshes.

Convergence analysis

Theorem 1: Convergence analysis of the standard FEM [Ern and Guermond, 2004]

We denote $u_h \in V_h$ the solution of $(\mathcal{P}_h)$ with V_h the standard trial space. Then,

$$\|u - u_h\|_{L^2} \leqslant C_{L^2}\, h^{k+1} |u|_{H^{k+1}}\,.$$

Convergence analysis

Theorem 1: Convergence analysis of the standard FEM [Ern and Guermond, 2004]

We denote $u_h \in V_h$ the solution of $(\mathcal{P}_h)$ with V_h the standard trial space. Then,

$$\|u - u_h\|_{L^2} \leqslant C_{L^2}\, h^{k+1} |u|_{H^{k+1}}.$$

Theorem 2: Convergence analysis of the enriched FEM [Lecourtier et al., 2026]

We denote $u_h^+ \in V_h^+$ the solution of $(\mathcal{P}_h^+)$ with V_h^+ the enriched trial space. Then,

$$\|u - u_h^+\|_{L^2} \leqslant \boxed{\frac{|u - u_\theta|_{H^{k+1}}}{|u|_{H^{k+1}}}} \left(C_{L^2}\, h^{k+1} |u|_{H^{k+1}} \right).$$

$$\boxed{\frac{|u - u_\theta|_{H^{k+1}}}{|u|_{H^{k+1}}}} < 1 \Rightarrow \text{Gains of the additive approach.}$$

Numerical results

- 2D Poisson problem on Square - Dirichlet BC
- 3D Poisson problem on Cube - Dirichlet BC
- 2D Anisotropic Elliptic problem on Square - Dirichlet BC
- 2D Poisson problem on Annulus - Mixed BC

Numerical results

2D Poisson problem on Square - Dirichlet BC

3D Poisson problem on Cube - Dirichlet BC

2D Anisotropic Elliptic problem on Square - Dirichlet BC

2D Poisson problem on Annulus - Mixed BC

Problem considered

Problem statement: Consider the Poisson problem in 2D with Dirichlet BC:

$$\begin{cases} -\Delta u = f, & \text{in } \Omega \times \mathcal{M}, \\ u = 0, & \text{on } \partial\Omega \times \mathcal{M}, \end{cases}$$

with $\Omega = (-0.5\pi, 0.5\pi)^2$ and $\mathcal{M} = [-0.5, 0.5]^2$ ($p = 2$ parameters).

Analytical solution: $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathcal{M}$

$$u(\boldsymbol{x}, \boldsymbol{\mu}) = \exp\left(-\frac{(x-\mu_1)^2 + (y-\mu_2)^2}{2}\right) \sin(2x)\sin(2y).$$

Parametric PINN training:

MLP 5 layers (sine - 40, 60, 60, 60, 40); LBFGs optimizer (5000 epochs - 6000 col. pts).

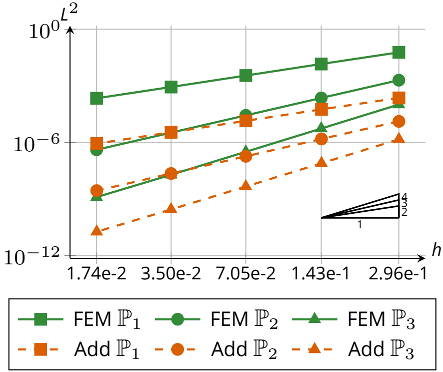
Imposing the Dirichlet BC exactly in the PINN with the levelset φ defined by

$$\varphi(\boldsymbol{x}) = (x + 0.5\pi)(x - 0.5\pi)(y + 0.5\pi)(y - 0.5\pi).$$

Online phase - 1 parameters instance

$$\boldsymbol{\mu}^{(1)} = (0.05, 0.22)$$

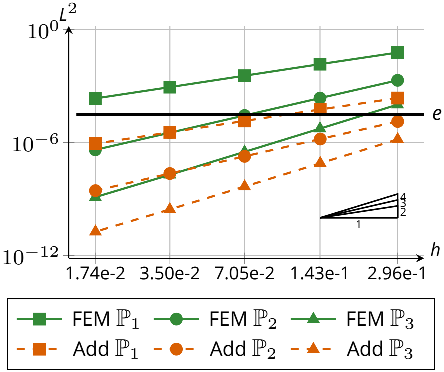
Error estimates in L^2 norm:



Online phase - 1 parameters instance

$$\mu^{(1)} = (0.05, 0.22)$$

Error estimates in L^2 norm:



N_{dofs} required to reach the same error e:

		N_{dofs}	
k	**e**	**FEM**	**Add**
1	$1 \cdot 10^{-3}$	14 161	64
	$1 \cdot 10^{-4}$	143 641	576
2	$1 \cdot 10^{-4}$	6 889	225
	$1 \cdot 10^{-5}$	31 329	1 089
3	$1 \cdot 10^{-5}$	6 724	784
	$1 \cdot 10^{-6}$	20 164	2 704

$$h \approx N_{\text{dofs}}^{-1/2}$$

Numerical results

2D Poisson problem on Square - Dirichlet BC

3D Poisson problem on Cube - Dirichlet BC

2D Anisotropic Elliptic problem on Square - Dirichlet BC

2D Poisson problem on Annulus - Mixed BC

Problem considered

Problem statement: Consider the Poisson problem in 3D with Dirichlet BC:

$$\begin{cases} -\Delta u = f, & \text{in } \Omega \times \mathcal{M}, \\ u = 0, & \text{on } \partial\Omega \times \mathcal{M}, \end{cases}$$

with $\Omega = (-0.5\pi, 0.5\pi)^3$ and $\mathcal{M} = [-0.5, 0.5]^3$ ($p = 3$ parameters).

Analytical solution: $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3) \in \mathcal{M}$

$$u(\boldsymbol{x}, \boldsymbol{\mu}) = \exp\left(-\frac{(x-\mu_1)^2 + (y-\mu_2)^2 + (z-\mu_3)^2}{2}\right) \sin(2x)\sin(2y)\sin(2z).$$

Parametric PINN training:

MLP 5 layers (tanh - 40, 60, 60, 60, 40); LBFGs optimizer (5000 epochs - 40000 col. pts).
Imposing the Dirichlet BC exactly in the PINN with the levelset φ defined by

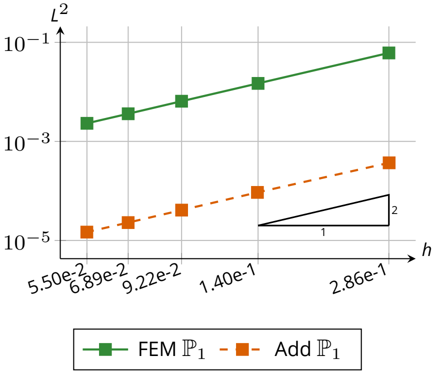
$$\varphi(\boldsymbol{x}) = (x+0.5\pi)(x-0.5\pi)(y+0.5\pi)(y-0.5\pi)(z+0.5\pi)(z-0.5\pi).$$

Training time: 12 minutes on NVIDIA RTX 2000 (8GB VRAM).

Online phase - 1 parameters instance I

$$\mu^{(1)} = (0.05, 0.22, 0.1)$$

Error estimates in L^2 norm:

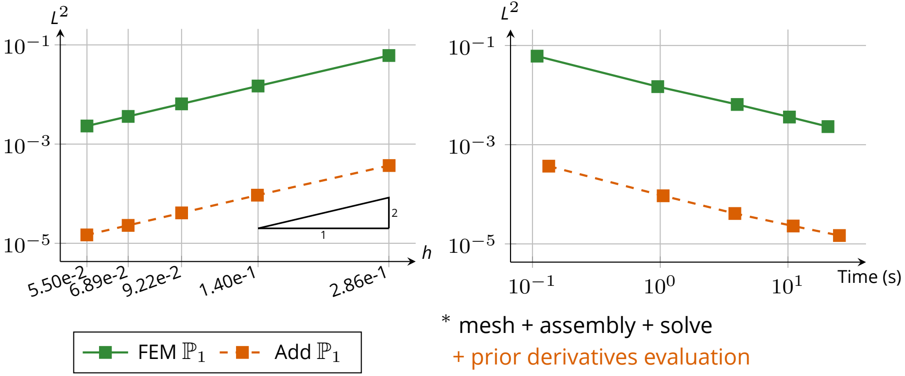


Online phase - 1 parameters instance I

$$\mu^{(1)} = (0.05, 0.22, 0.1)$$

Error estimates in L^2 norm:

L^2 error w.r.t. online time*:



* mesh + assembly + solve

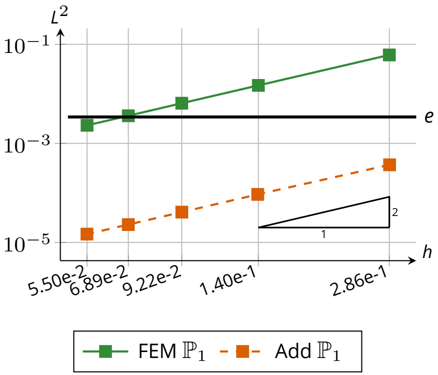
+ prior derivatives evaluation

~~+ training time~~

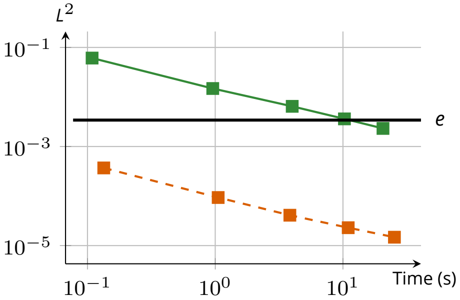
Online phase - 1 parameters instance I

$$\boldsymbol{\mu}^{(1)} = (0.05, 0.22, 0.1)$$

Error estimates in L^2 norm:



L^2 error w.r.t. online time*:



* mesh + assembly + solve
+ prior derivatives evaluation
~~+ training time~~

Online phase - 1 parameters instance II

$$\mu^{(1)} = (0.05, 0.22, 0.1)$$

N_{dofs} and online time required to reach the same error e:

	N		N_{dofs}		Online time*	
e	**FEM**	**Add**	**FEM**	**Add**	**FEM**	**Add**
$1 \cdot 10^{-3}$	152	12	$3.51 \cdot 10^{6}$	$1.73 \cdot 10^{3}$	$(7.65 \cdot 10^{1})$	$5.2 \cdot 10^{-2}$
$1 \cdot 10^{-4}$	484	39	$1.13 \cdot 10^{8}$	$5.93 \cdot 10^{4}$	$(2.8 \cdot 10^{3})$	$9.26 \cdot 10^{-1}$
$1 \cdot 10^{-5}$	1 539	122	$3.65 \cdot 10^{9}$	$1.82 \cdot 10^{6}$	$(1.02 \cdot 10^{5})$	$(5.26 \cdot 10^{1})$

$\div 1471$ →

Cartesian mesh: $N_{dofs} = N^3$ nodes ($\mathbb{P}_1$ elements).

* The results in brackets are extrapolations.

Online phase - 1 parameters instance II

$$\boldsymbol{\mu}^{(1)} = (0.05, 0.22, 0.1)$$

N_{dofs} and online time required to reach the same error e:

	N		N_{dofs}		Online time*	
e	**FEM**	**Add**	**FEM**	**Add**	**FEM**	**Add**
$1 \cdot 10^{-3}$	152	12	$3.51 \cdot 10^6$	$1.73 \cdot 10^3$	$(7.65 \cdot 10^1)$	$5.2 \cdot 10^{-2}$
$1 \cdot 10^{-4}$	484	39	$1.13 \cdot 10^8$	$5.93 \cdot 10^4$	$(2.8 \cdot 10^3)$	$9.26 \cdot 10^{-1}$
$1 \cdot 10^{-5}$	1 539	122	$3.65 \cdot 10^9$	$1.82 \cdot 10^6$	$(1.02 \cdot 10^5)$	$(5.26 \cdot 10^1)$

$\div 1471$

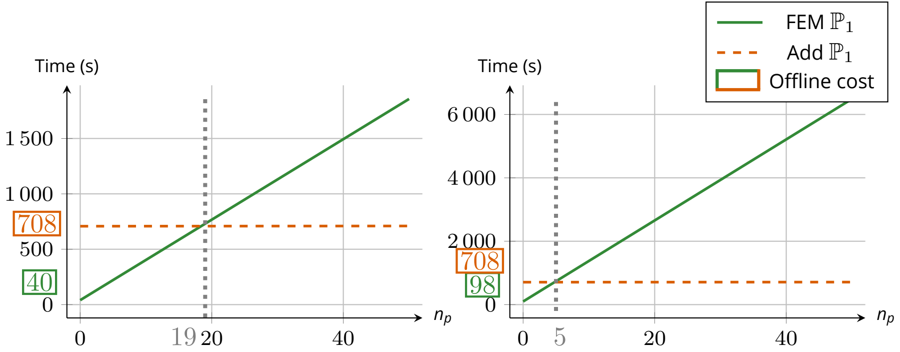
Cartesian mesh: $N_{\text{dofs}} = N^3$ nodes ($\mathbb{P}_1$ elements).

* The results in brackets are extrapolations.

⚠ Training time !

Parametric framework

How many parameters instance n_p are needed to balance training?



To achieve $e = 10^{-3}$ | To achieve $e = 5 \cdot 10^{-4}$

Online: assembly + solve + prior derivatives evaluation

Offline: mesh + training time

Numerical results

Problem considered

Problem statement: Considering an Anisotropic Elliptic problem with Dirichlet BC:

$$\begin{cases} -\mathrm{div}(D\nabla u) = f, & \text{in } \Omega \times \mathcal{M}, \\ u = 0, & \text{on } \partial\Omega \times \mathcal{M}, \end{cases}$$

with $\Omega = (0,1)^2$ and $\mathcal{M} = [0.4, 0.6]^2 \times [0.01, 1] \times [0.1, 0.8]$ ($p = 4$ parameters).

Right-hand side and diffusion matrix: $\boldsymbol{\mu} = (\mu_1, \mu_2, \sigma, \epsilon) \in \mathcal{M}$

$$f(\boldsymbol{x}, \boldsymbol{\mu}) = \exp\left(-\frac{(x-\mu_1)^2 + (y-\mu_2)^2}{0.025\sigma^2}\right).$$

$$D(\boldsymbol{x}, \boldsymbol{\mu}) = \begin{pmatrix} \epsilon x^2 + y^2 & (\epsilon - 1)xy \\ (\epsilon - 1)xy & x^2 + \epsilon y^2 \end{pmatrix}.$$

Parametric PINN training:

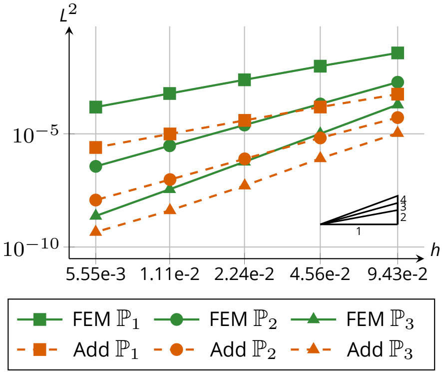
MLP[1] 5 layers (tanh - 40, 60, 60, 60, 40). Adam optimizer (15000 epochs - 8000 col. pts).
Imposing the Dirichlet BC exactly in the PINN with the a level-set function.

[1] with Fourier Features, Tancik et al. [2020]

Online phase

Error estimates: 1 parameter instance

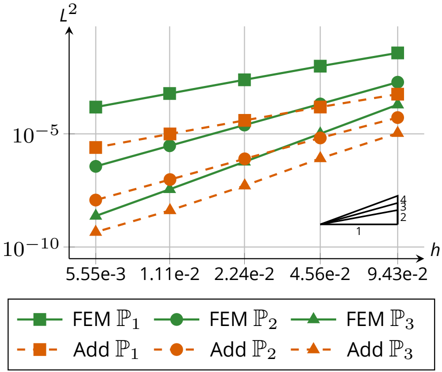
$$\boldsymbol{\mu}^{(1)} = (0.51, 0.54, 0.52, 0.55)$$



Online phase

Error estimates: 1 parameter instance

$$\boldsymbol{\mu}^{(1)} = (0.51, 0.54, 0.52, 0.55)$$



Gains achieved: $\|u - u_h\|_{L^2} / \|u - u_h^+\|_{L^2}$

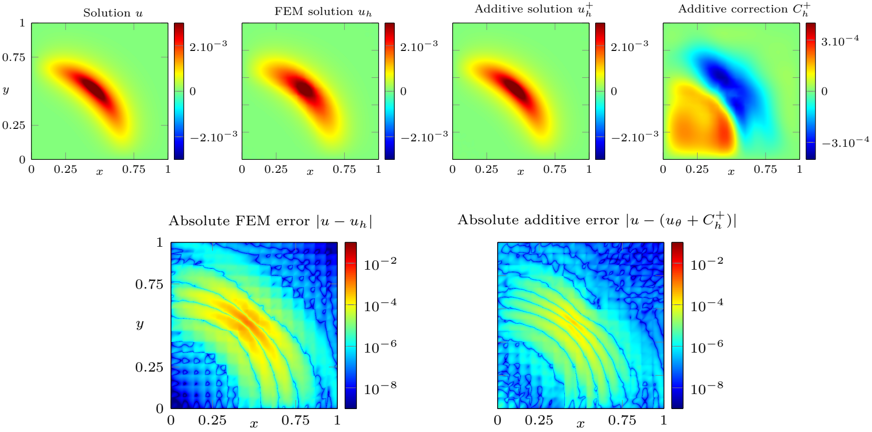
Considering $n_p = 50$ parameters instances

$$\mathcal{S} = \left\{ \boldsymbol{\mu}^{(1)}, \dots, \boldsymbol{\mu}^{(n_p)} \right\}.$$

	Gains in L^2 rel error of our method w.r.t. FEM		
k	**min**	**max**	**mean**
1	7.12	82.57	35.67
2	3.54	35.88	18.32
3	1.33	26.51	8.32

Cartesian mesh: N^2 nodes with $N = 20$.

Numerical solutions and errors



$$\boldsymbol{\mu}^{(2)} = (0.46, 0.52, 0.05, 0.12) \quad (k = 2, N = 16)$$

Numerical results

2D Poisson problem on Square - Dirichlet BC

3D Poisson problem on Cube - Dirichlet BC

2D Anisotropic Elliptic problem on Square - Dirichlet BC

2D Poisson problem on Annulus - Mixed BC

Problem considered

Problem statement: Considering the Poisson problem with mixed BC:

$$\begin{cases} -\Delta u = f, & \text{in } \Omega \times \mathcal{M}, \\ u = g, & \text{on } \Gamma_E \times \mathcal{M}, \\ \dfrac{\partial u}{\partial n} + u = g_R, & \text{on } \Gamma_I \times \mathcal{M}, \end{cases}$$

with $\Omega = \{(x,y) \in \mathbb{R}^2, 0.25 < x^2 + y^2 < 1\}$ and $\mathcal{M} = [2.4, 2.6]$ ($p = 1$ parameter).

Analytical solution and boundary conditions: $\boldsymbol{\mu} = \mu_1 \in \mathcal{M}$

$$u(\boldsymbol{x};\boldsymbol{\mu}) = 1 - \frac{\ln\left(\mu_1 \sqrt{x^2+y^2}\right)}{\ln(4)},$$

$$g(\boldsymbol{x};\boldsymbol{\mu}) = 1 - \frac{\ln(\mu_1)}{\ln(4)} \quad \text{and} \quad g_R(\boldsymbol{x};\boldsymbol{\mu}) = 2 + \frac{4 - \ln(\mu_1)}{\ln(4)}.$$

Parametric PINN training:

MLP 5 layers (tanh - 5×40); LBFGs optimizer (4000 epochs - 6000 col. pts).

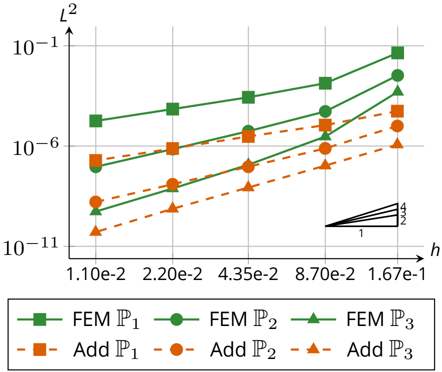
Imposing the mixed BC exactly in the PINN[1] (+ Sobolev training).

[1] [Sukumar and Srivastava, 2022]

Online phase

Error estimates: 1 parameters instance

$$\boldsymbol{\mu}^{(1)} = 2.51$$



Gains achieved: $\|u - u_h\|_{L^2} / \|u - u_h^+\|_{L^2}$

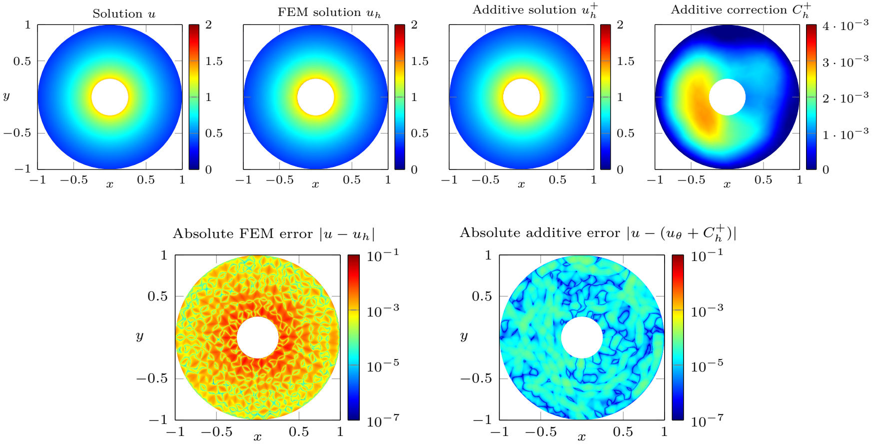
Considering $n_p = 50$ parameters instances

$$\mathcal{S} = \left\{ \boldsymbol{\mu}^{(1)}, \ldots, \boldsymbol{\mu}^{(n_p)} \right\}.$$

	Gains in L^2 rel error of our method w.r.t. FEM		
k	**min**	**max**	**mean**
1	15.12	137.72	55.5
2	31	77.46	58.41
3	18.72	21.49	20.6

Mesh size: $h = 1.33 \cdot 10^{-1}$

Numerical solutions and errors



$$\mu^{(1)} = 2.51 \quad (k = 1, h = 1.67 \cdot 10^{-1})$$

Conclusion

- **Offline/online framework:**
  - **Offline phase:** Training a PINN over a parametric space.
  - **Online phase:** (fixed parameter instance)
    Quick and accurate resolution of the FEM correction problem using the PINN prior.
- **Numerical results:** Several parametrics PDEs.

Some extensions:

- Extension to non-linear problems : First promising results for the incompressible Navier-Stokes equations.
- Extensions to DG methods and non-conforming FEM, currently under development in ScimBa (SciML's Python library) using auto-differentiation

SCAN ME

References

A. Ern and J.-L. Guermond. Theory and Practice of Finite Elements. Springer New York, 2004.

F. Lecourtier, H. Barucq, M. Duprez, F. Faucher, E. Franck, V. Lleras, V. Michel-Dansac, and N. Victorion. Enriching continuous lagrange finite element approximation spaces using neural networks. ESAIM: M2AN, 60(2):541–603, 2026.

N. Sukumar and A. Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. Comput. Method. Appl. M., 389:114333, 2022. ISSN 0045-7825.

M. Tancik, P. Srinivasan, and al. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In Advances in Neural Information Processing Systems, volume 33, pages 7537–7547. Curran Associates, Inc., 2020.

Thank you for your attention!

SCAN ME