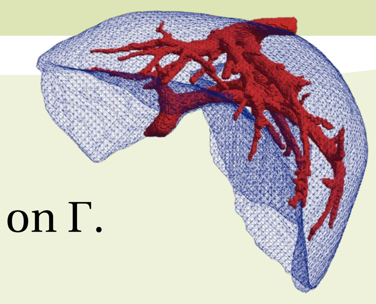
Combining Finite Element methods and Neural Networks to solve elliptic problem on complex 2D geometries

Hélène BARUCQ[3], Michel DUPREZ[1], Florian FAUCHER[3], Emmanuel FRANCK[2], **Frédérique LECOURTIER**[1], Vanessa LLERAS[4], Victor MICHEL-DANSAC[2], and Nicolas VICTORION[3]

[1] Mimesis team, INRIA Nancy grand Est, Icube [2] Macaron team, INRIA Nancy grand Est, IRMA [3] Makutu team, INRIA Bordeaux, TotalEnergies [4] Montpellier University

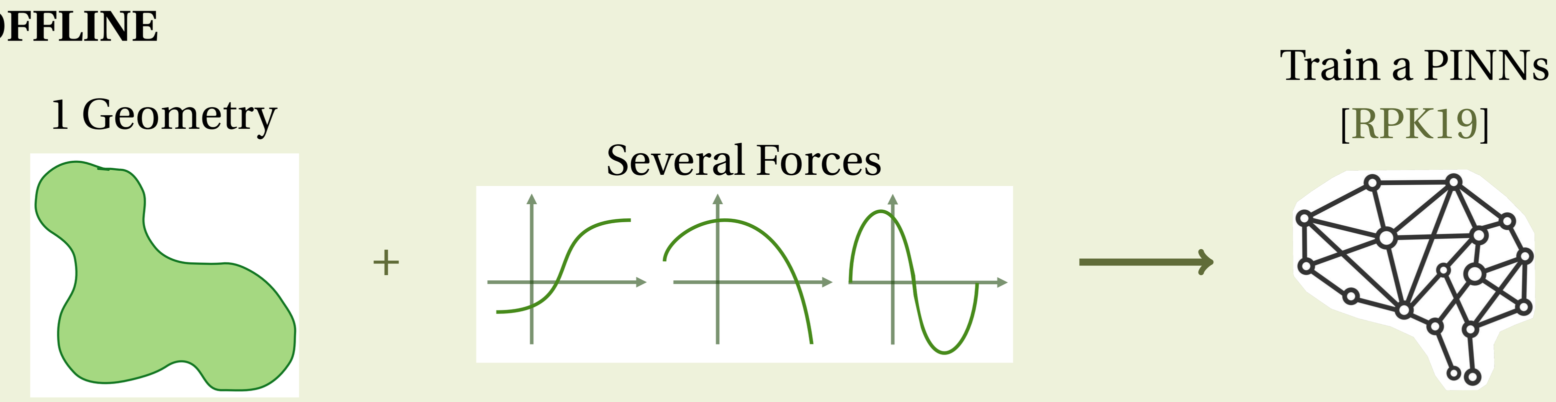


Motivations

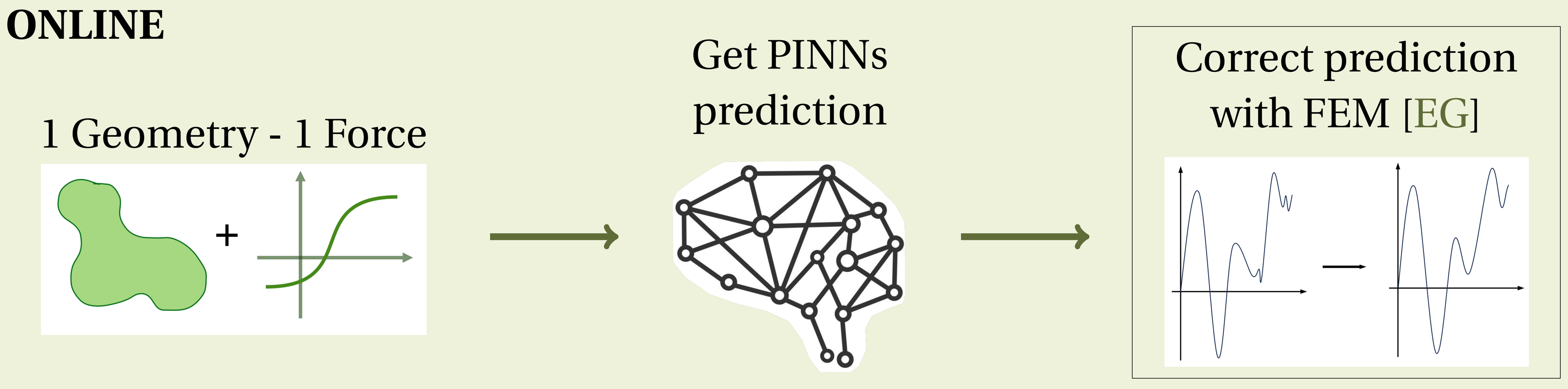
Current Objective : Develop hybrid **finite element** / **neural network** methods.
accurate quick + parameterized

Problem considered : $-\Delta u(x) = f(x)$ in Ω, $u(x) = g(x)$ on Γ.
Poisson problem with Dirichlet boundary conditions (BC).

OFFLINE



ONLINE

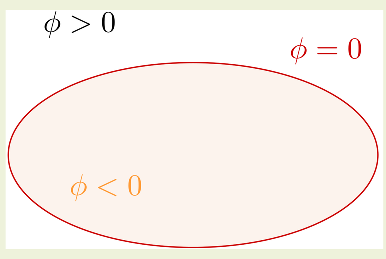


Perspective : Create real-time digital twins of an organ (e.g. liver).

1. How to deal with complex geometries in PINNs ?

No mesh, so easy to go on complex geometry !

Approach by levelset. [SS22]



Advantages :
- ➙ Sample is easy in this case.
- ➙ Allow to impose in hard the BC (no BC loss) :

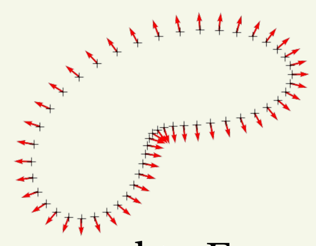
$$u_\theta(X) = \phi(X) w_\theta(X) + g(X)$$

with ϕ a levelset function and w_θ a NN.

Levelset considered. A regularized Signed Distance Function (SDF).

Theorem 1: Eikonal equation. [CD23]
If we have a boundary domain Γ, the SDF is solution to:

$$\begin{cases} \|\nabla\phi(X)\| = 1,\ X \in \mathcal{O} & (1) \\ \phi(X) = 0,\ X \in \Gamma & (2) \\ \nabla\phi(X) = n,\ X \in \Gamma & (3) \end{cases}$$



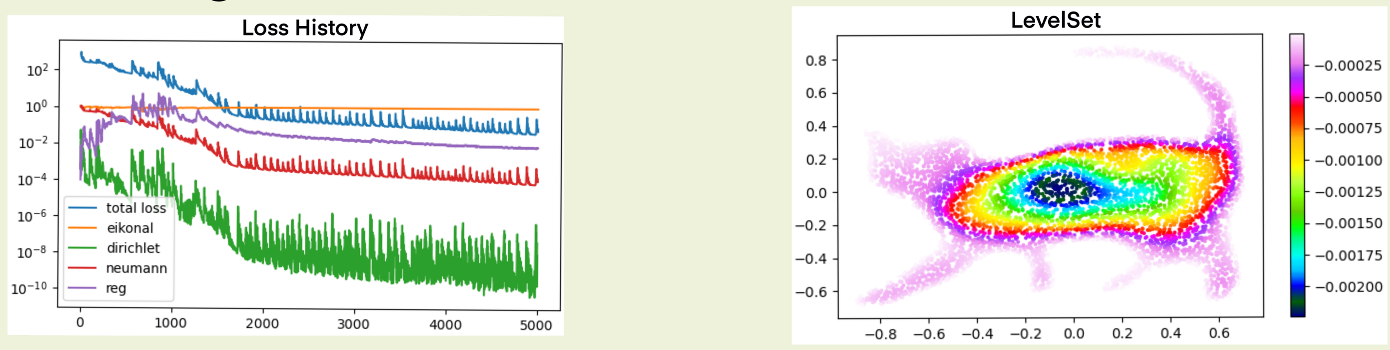
with $\mathcal{O}$ a box which contains Ω completely and n the exterior normal to Γ.

Approximate ϕ ? with a PINNs [CD23], by adding the following regularization term

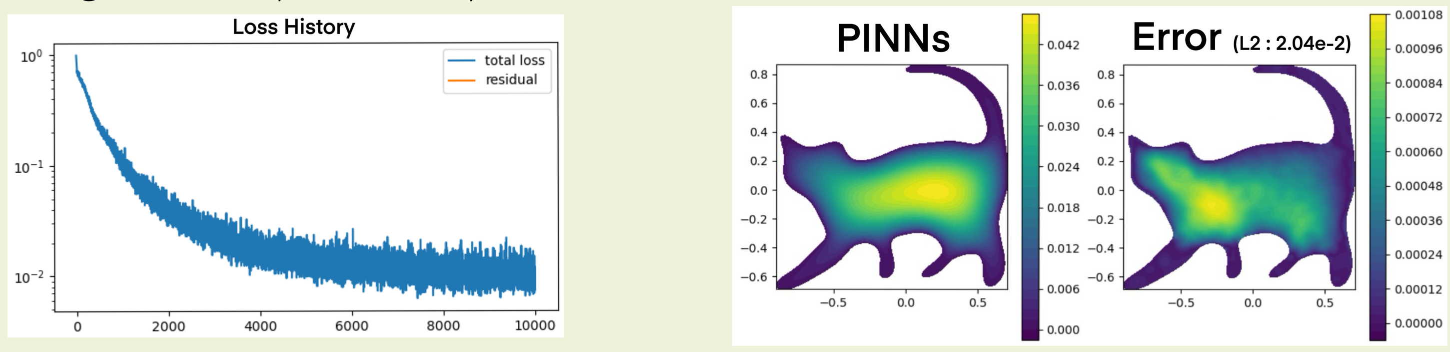
$$\mathcal{L} = \underbrace{\int_{\mathcal{O}} \left(1 - \|\nabla\phi(x)\|\right)^2 dx}_{(1)} + \underbrace{\int_\Gamma |\phi(x)|^2 dx}_{(2)} + \underbrace{\int_\Gamma 1 - \frac{n(x)\cdot\nabla\phi(x)}{\|n(x)\|\,\|\nabla\phi(x)\|} dx}_{(3)} + \underbrace{\int_{\mathcal{O}} |\Delta\phi(x)|^2 dx}_{\text{reg}}.$$

Numerical Results - Learn a complex levelset

Levelset learning.



Poisson problem on Cat.
- ➙ Taking $f = 1$ (**non parametric**) and homogeneous Dirichlet BC ($g = 0$).
- ➙ Looking for $u_\theta = \phi w_\theta$ with ϕ the levelset learned.



2. How can we improve PINNs prediction ? - Using FEM

Additive approach. Considering u_θ as the prediction of our PINNs for the Poisson problem, the correction problem consists in writing the solution as

$$\tilde{u} = u_\theta + \underbrace{\tilde{C}}_{\ll 1}$$

and searching $\tilde{C} : \Omega \to \mathbb{R}^d$ such that

$$\begin{cases} -\Delta\tilde{C} = \tilde{f}, & \text{in } \Omega, \\ \tilde{C} = 0, & \text{on } \Gamma, \end{cases} \qquad (\mathcal{P}^+)$$

with $\tilde{f} = f + \Delta u_\theta$.

Error estimation. Considering u_θ as the prediction of our PINNs.

Theorem 2: [Lec+]
We denote u the solution of the Poisson problem and u_h the discrete solution of the correction problem $(\mathcal{P}^+)$ with V_h a $\mathbb{P}_k$ Lagrange space. Thus

$$\|u - u_h\|_0 \lesssim \underbrace{\frac{|u - u_\theta|_{H^{k+1}}}{|u|_{H^{k+1}}}}_{C_{\text{gain}}} h^{k+1} |u|_{H^{k+1}}$$

Remark : The constant C_{gain} shows that the closer the prior is to the solution, the lower the error constant associated with the method.

Problem considered - Numerical results

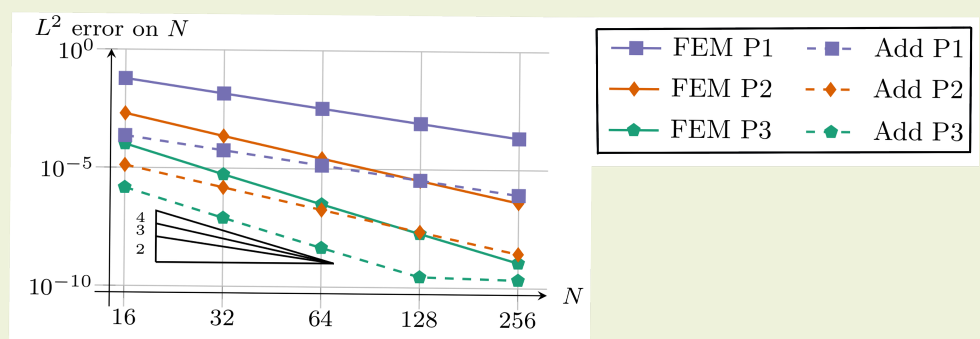
Poisson problem on Square.
- ➙ Considering homogeneous Dirichlet BC ($g = 0$) and $\Omega = [-0.5\pi, 0.5\pi]^2$.
- ➙ Analytical levelset function : $\phi(x, y) = (x - 0.5\pi)(x + 0.5\pi)(y - 0.5\pi)(y + 0.5\pi)$
- ➙ Analytical solution :

$$u_{ex}(x, y) = \exp\left(-\frac{(x - \mu_1)^2 + (y - \mu_2)^2}{2}\right) \sin(2x)\sin(2y)$$

with $\mu_1, \mu_2 \in [-0.5, 0.5]$ (**parametric**) and the associated source term f.

Numerical results - Improve errors

Theoretical results. Taking $\mu_1 = 0.05$, $\mu_2 = 0.22$.



Remark : We note N the number of nodes in each direction of the square (Total : N^2).

Gains on error using additive approach.
Considering a set of $n_p = 50$ parameters : $\left\{(\mu_1^{(1)}, \mu_2^{(1)}), \ldots, (\mu_1^{(n_p)}, \mu_2^{(n_p)})\right\}$.

Solution $\mathbb{P}_1$

	Gains on PINNs				Gains on FEM			
N	min	max	mean	std	min	max	mean	std
20	15.7	48.35	33.64	5.57	134.31	377.36	269.4	43.67
40	61.47	195.75	135.41	23.21	131.18	362.09	262.12	41.67

Numerical results - Improve times

Time/error ratio. Training time for PINNs : $t_{PINNs} \approx 240s$.

➙ After training, how long does each method take to solve 1 problem ?

	N		time (s)	
Precision	FEM	Add	FEM	Add
$1e-3$	120	8	$\boxed{43}$	$\boxed{0.24}$
$1e-4$	373	25	423.89	1.93
			t_{FEM}	t_{Add}

➙ Including training, how long does each method take to solve n_p problems ?

Total time of Additive approach : $Tot_{Add} = t_{PINNs} + n_p t_{Add}$

Total time of FEM : $Tot_{FEM} = n_p t_{FEM}$

How many parameters n_p to make our method faster than FEM ?
Let's suppose we want to achieve an **error of** $1e-3$.

$$Tot_{Add} < Tot_{FEM} \Rightarrow n_p > \frac{t_{PINNs}}{t_{FEM} - t_{Add}} \approx 5.61 \Rightarrow \boxed{n_p = 6}$$

[CD23] M. Clémot and J. Digne. "Neural skeleton: Implicit neural representation away from the surface". In: *Computers and Graphics* (2023).

[EG] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements.* Springer New York (2004).

[Lec+] **Lecourtier** et al. *Enhanced finite element methods using neural networks.* (in progress).

[RPK19] Raissi, Perdikaris, and Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* (2019).

[SS22] Sukumar and Srivastava. "Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks". In: *Computer Methods in Applied Mechanics and Engineering* (2022).