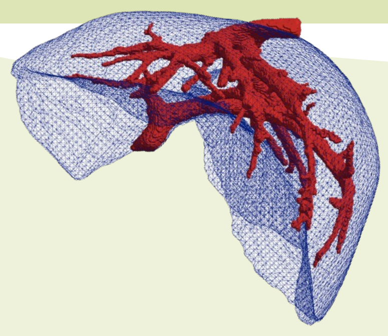
Combining Finite Element methods and Neural Networks to solve elliptic problem on complex 2D geometries

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

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

MIMESIS

Scientific context

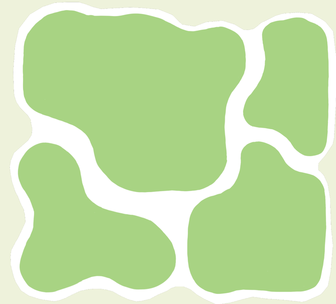


Create real-time digital twins of an organ (e.g. liver)

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

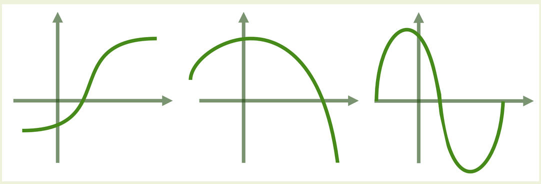
OFFLINE

Several Geometries



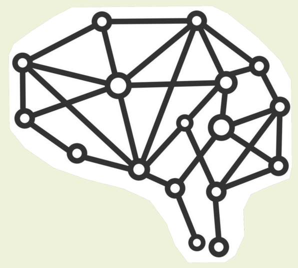
\+

Several Forces



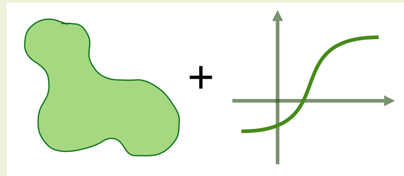
→

Train a PINNs [RPK19]



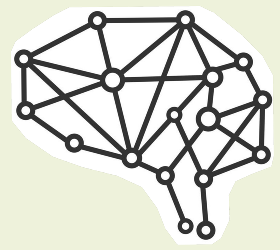
ONLINE

1 Geometry - 1 Force



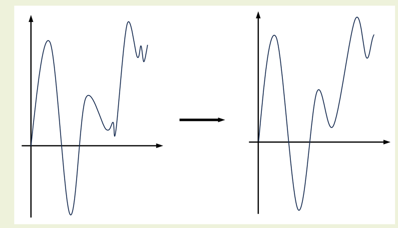
→

Get PINNs prediction



→

Correct prediction with FEM

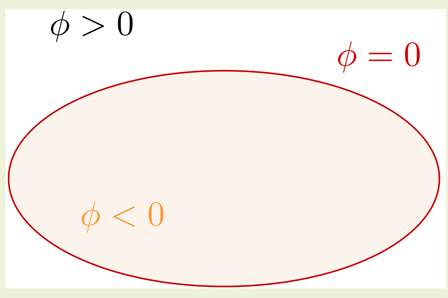


Problem considered : Poisson problem with Dirichlet boundary conditions.

How to deal with complex geometry 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 more J_{bc}) :

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

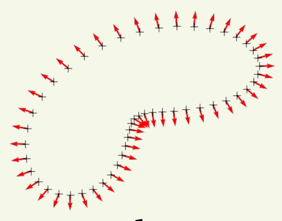
with ϕ a levelset function and w_θ a NN.

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

Theorem 1: Eikonal equation.

If we have a boundary domain Γ, the SDF is solution to:

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



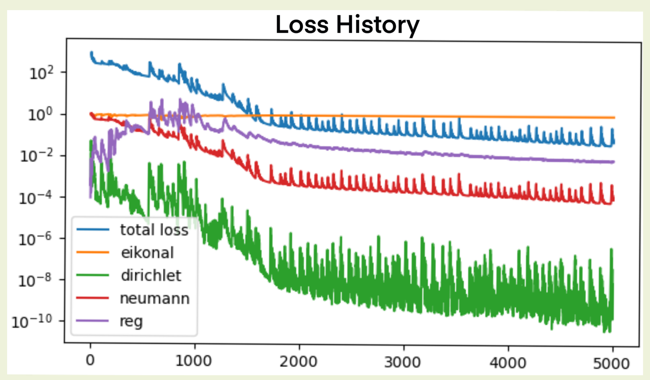
with $\mathcal{O}$ a box which contains Ω completely and n the exterior normal to Γ.

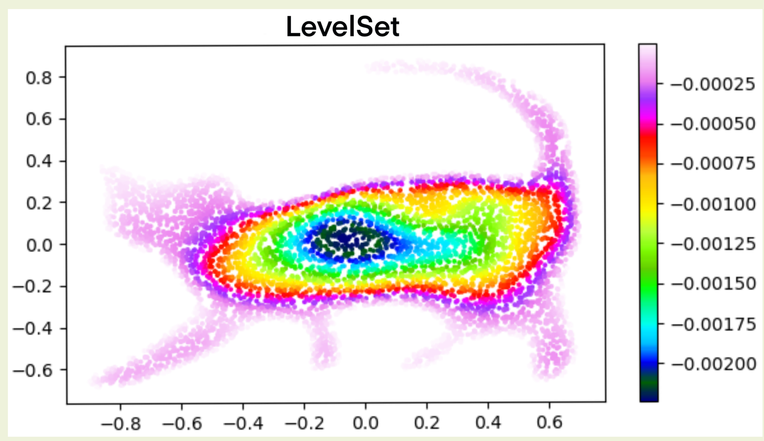
How to do that ? with a PINNs [CD23], by adding the following regularization term

$$J_{\text{reg}} = \int_{\mathcal{O}} |\Delta\phi|^2.$$

Results - Complex geometry

Levelset learning.

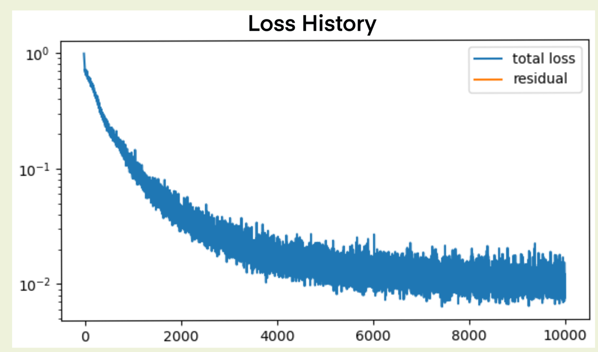


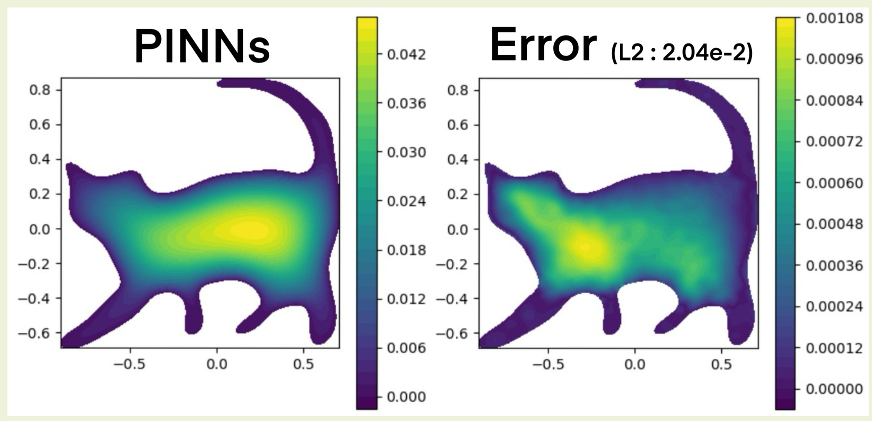


Poisson on Cat.

→ Solving *Poisson* with $f = 1$ (non parametric) and homogeneous Dirichlet BC ($g = 0$).

→ Looking for $u_\theta = \phi w_\theta$ with ϕ the levelset learned.





How can we improve PINNs prediction ?

Using FEM-type methods

Additive approach. Considering u_{NN} as the prediction of our PINNs for *Poisson*, the correction problem consists in writing the solution as

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

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

Problem considered. Poisson on Square

→ Solving *Poisson* with 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**).

Theoretical results.

Theorem 2: [Lec+ss]

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

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

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

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

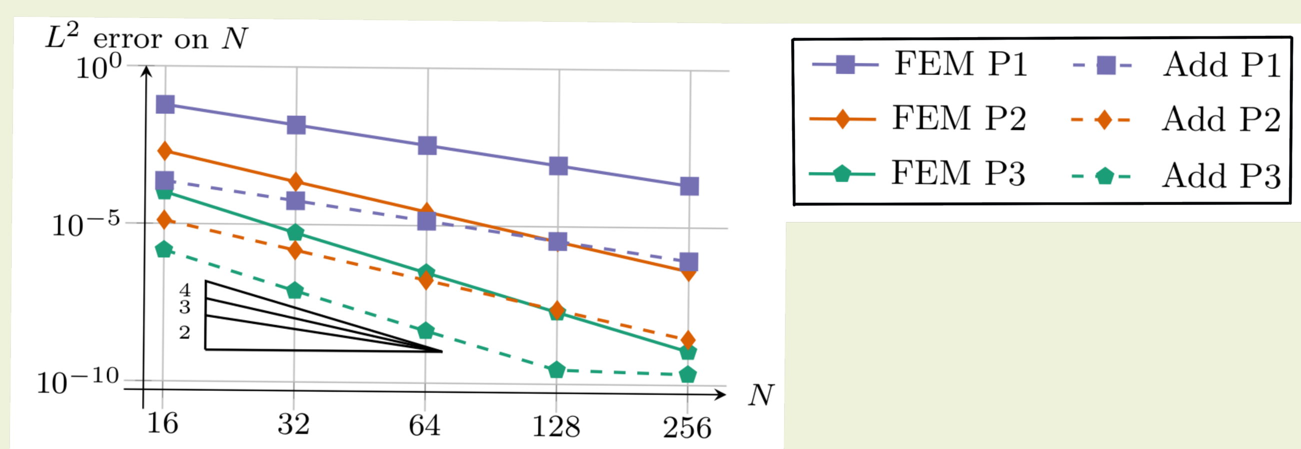
[Lec+ss] **Lecourtier** et al. "Enhanced finite element methods using neural networks". In: (in progress).

[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).

Results - Improve prediction

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



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

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

→ **At a given precision, how long does each method take to solve 1 problem ?**

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

→ **How many parameters n_p to make our method faster than FEM ?**

Total time of Additive approach :

$$Tot_{Add} = t_{PINNs} + n_p t_{Add}$$

Total time of FEM :

$$Tot_{FEM} = n_p t_{FEM}$$

Let's suppose we want to achieve an **error of** $1e-3$.

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