CS1

Development of hybrid finite element/neural network methods to help create digital surgical twins

Authors:
Frédérique LECOURTIER

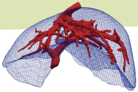
Supervisors:
Emmanuel FRANCK
Michel DUPREZ
Vanessa LLERAS

June 14, 2024

MIMESIS

Introduction

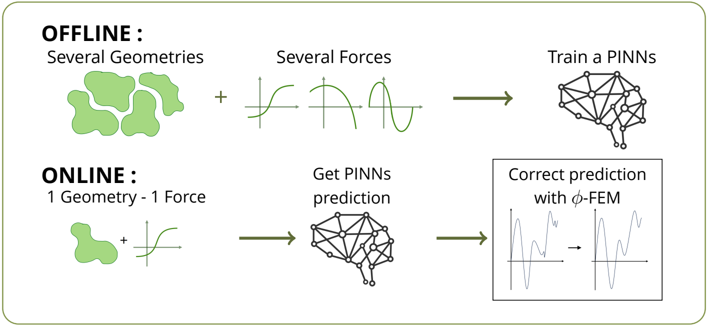
Scientific context



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

Current Objective : Develop hybrid finite element / neural network methods.

accurate — quick + parameterized



ϕ-FEM : New fictitious domain finite element method.

$\Rightarrow$ domain given by a level-set function

Appendix 1

Current work

Elliptic problem with Dirichlet conditions :

Find $u : \Omega \to \mathbb{R}^d (d = 1, 2, 3)$ such that

$$\begin{cases} L(u) = -\nabla \cdot (A(x)\nabla u(x)) + c(x)u(x) = f(x) \quad \text{in } \Omega, \\ u(x) = g(x) \quad \text{on } \partial\Omega \end{cases} \tag{1}$$

with A a definite positive coercivity condition and c a scalar. We consider Δ the Laplace operator, Ω a smooth bounded open set and Γ its boundary.

Two lines of research :

1. How to deal with complex geometry in PINNs ?
2. Once we have the prediction, how can we improve it (using FEM-type methods) ?

MIMESIS

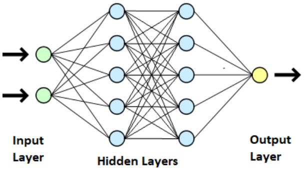
How to deal with complex geometry in PINNs ?

Standard PINNs

Implicit neural representation.

$$u_\theta(x) = u_{NN}(x)$$

with u_{NN} a neural network (e.g. a MLP).



DoFs Minimization Problem :

Considering the least-square form of (1), our discrete problem is

$$\theta_u = \underset{\theta \in \mathbb{R}^N}{\operatorname{argmin}}\, \alpha J_{in}(\theta) + \beta J_{bc}(\theta) \tag{2}$$

with N the number of parameters of the NN and

$$J_{in}(\theta) = \frac{1}{2} \int_\Omega (L(u_\theta) - f)^2 \quad \text{and} \quad J_{bc}(\theta) = \frac{1}{2} \int_{\partial\Omega} (u_\theta - g)^2$$

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

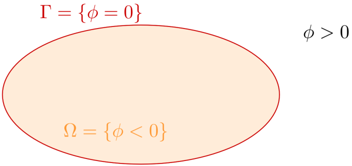
MIMESIS

Limits

Claim on PINNs : No mesh, so easy to go on complex geometry !

⚠ *In practice :* Not so easy ! We need to find how to sample in the geometry.

Solution : Approach by levelset.



Advantages :

→ Sample is easy in this case.

→ Allow to impose in hard the BC :

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

Natural LevelSet :

Signed Distance Function (SDF)

Problem : SDF is a $\mathcal{C}^0$ function

$\Rightarrow$ its derivatives explodes

$\Rightarrow$ we need a regular levelset

Learn a regular levelset

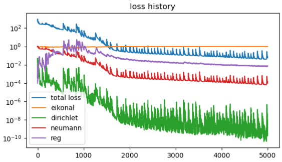
If we have a boundary domain Γ, the SDF is solution to the Eikonal equation:

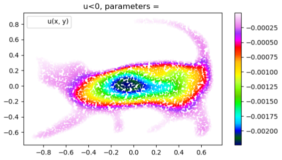
$$\begin{cases} ||\nabla\phi(X)|| = 1, \ X \in \mathcal{O} \\ \phi(X) = 0, \ X \in \Gamma \\ \nabla\phi(X) = n, \ X \in \Gamma \end{cases}$$

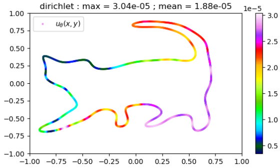
with $\mathcal{O}$ a box which contains Ω completely and n the exterior normal to Γ.

How make that ? with a PINNs [2] by adding a term to regularize.

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





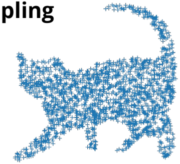


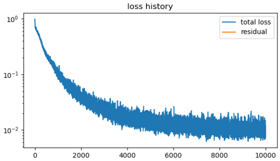
MIMESIS

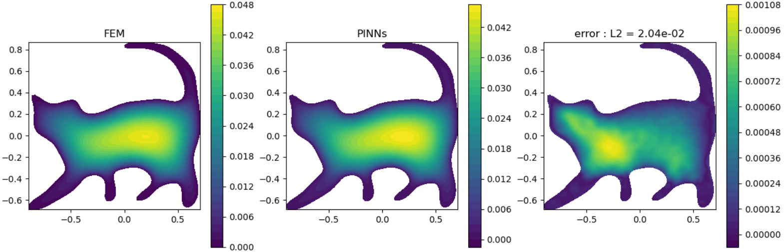
Poisson On Cat

- Solving the Poisson problem with $f = 1$ and homogeneous Dirichlet BC.
- Looking for $u_\theta = \phi w_\theta$ with ϕ the levelset learned.

Sampling







How improve PINNs prediction (on simple geometry) ?

Idea

TODO

Standard FEM

TODO

Theoretical results

TODO

Numerical results I

TODO

MIMESIS

Numerical results II

TODO

Conclusion

Supplementary work

TODO

MIMESIS

Conclusion

TODO

MIMESIS

Thank you !

Bibliography

[1] Alexander Belyaev, Pierre-Alain Fayolle, and Alexander Pasko. Signed Lp-distance fields. *Computer-Aided Design*.

[2] Mattéo Clémot and Julie Digne. Neural skeleton: Implicit neural representation away from the surface. *Computers and Graphics*.

[3] Pierre-Alain Fayolle. Signed Distance Function Computation from an Implicit Surface.

[4] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*.

[5] N. Sukumar and Ankit Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Computer Methods in Applied Mechanics and Engineering*.

[6] Sifan Wang, Shyam Sankaran, Hanwen Wang, and Paris Perdikaris. An Expert's Guide to Training Physics-informed Neural Networks.

Appendix

Appendix 1 : ϕ-FEM

App1

Appendix 2 : Test2

App2

Appendix 3 : Test3

App3