CS1

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

Authors:
Frédérique LECOURTIER

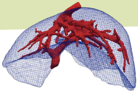
Supervisors:
Emmanuel FRANCK
Michel DUPREZ
Vanessa LLERAS

June 14, 2024

MIMESIS

Introduction

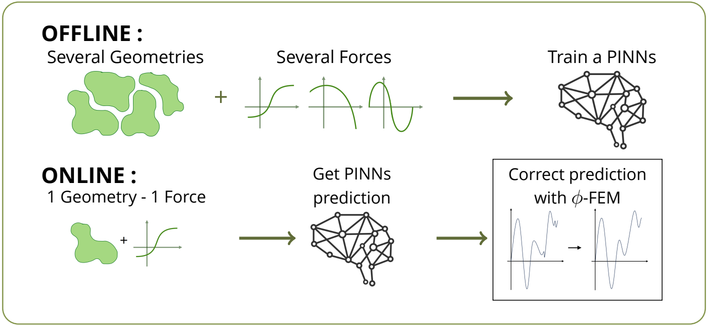
Scientific context



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 2

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) ?

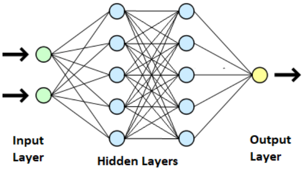
How to deal with complex geometry in PINNs ?

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.

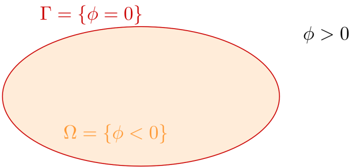
MIMESIS

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

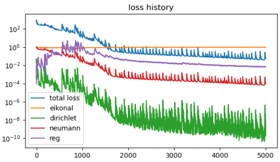
If we have a boundary domain Γ, the SDF is solution to the Eikonal equation:

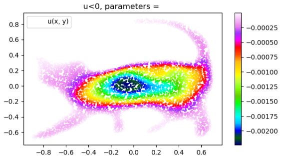
$$\begin{cases} ||\nabla\phi(X)|| = 1, \; X \in \mathcal{O} \\ \phi(X) = 0, \; X \in \Gamma \\ \nabla\phi(X) = n, \; X \in \Gamma \end{cases}$$

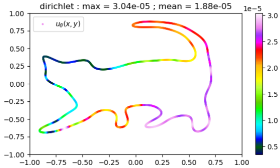
with $\mathcal{O}$ a box which contains Ω completely and n the exterior normal to Γ.

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

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



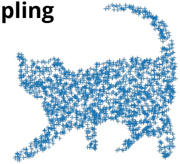


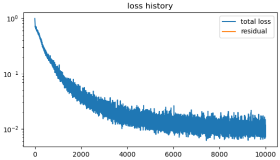


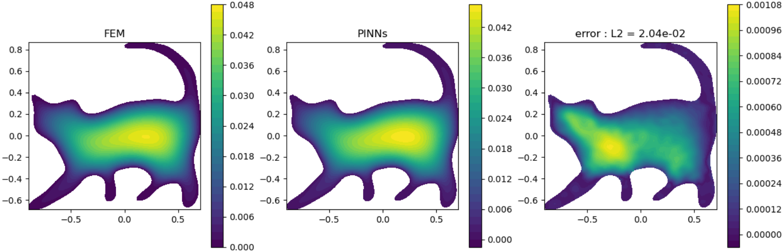
Poisson On Cat

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

Sampling





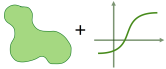


How improve PINNs prediction ?

⚠Considering simple geometry (i.e analytic levelset ϕ).

Idea

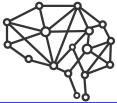
1 Geometry + 1 Force



ϕ and f

(and g)

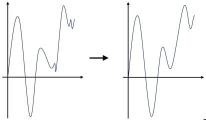
Get PINNs prediction



$$\boxed{u_{NN} = \phi w_{NN} + g}$$

$u_{NN} = g$ on Γ

Correct prediction with FEM



$u_{NN} \rightarrow \tilde{u} = u_{NN} + \tilde{C}$

Correct by adding : Considering u_{NN} as the prediction of our PINNs for (1), the correction problem consists in writing the solution as

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

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

$$\begin{cases} L(\tilde{C}) = \tilde{f}, & \text{in } \Omega, \\ \tilde{C} = 0, & \text{on } \Gamma, \end{cases}$$

with $\tilde{f} = f - L(u_{NN})$. Appendix 1

Poisson on Square

Solving the Poisson problem with homogeneous Dirichlet BC.

→ **Domain :** $\Omega = [-0.5\pi, 0.5\pi]^2$

→ **Analytical levelset function :**

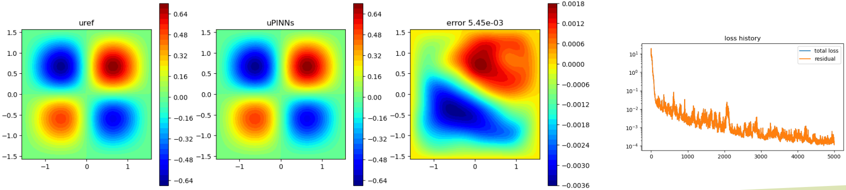
$$\phi(x,y) = (x - 0.5\pi)(x + 0.5\pi)(y - 0.5\pi)(y + 0.5\pi)$$

→ **Analytical solution :**

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

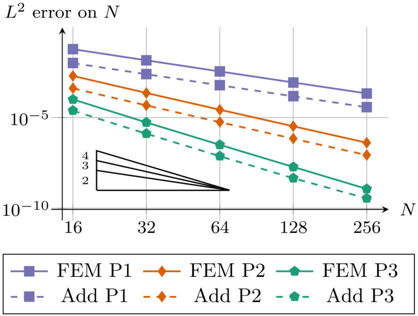
with $\mu_1, \mu_2 \in [-0.5, 0.5]$.

Taking $\mu_1 = 0.05, \mu_2 = 0.22$, the solution is given by



Theoretical results

TODO $\mu_1 = 0.05, \mu_2 = 0.22$



Gains using our approach

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

Solution $\mathbb{P}_2$

	Gains on PINNs				Gains on FEM			
N	**min**	**max**	**mean**	**std**	**min**	**max**	**mean**	**std**
20	244.81	996.23	655.08	153.63	67.12	165.13	135.21	21.37
40	2,056.2	8,345.4	5,504.89	1,287.16	66.52	159.73	132.05	20.38

Solution $\mathbb{P}_3$

	Gains on PINNs				Gains on FEM			
N	**min**	**max**	**mean**	**std**	**min**	**max**	**mean**	**std**
20	2,804.27	11,797.23	7,607.51	1,780.7	39.72	72.99	61.85	7.05
40	50,989.23	212,714.99	137,711.77	32,125.57	40.02	73	61.98	6.92

Time/Precision

TODO

Conclusion

Supplementary work

TODO

Conclusion

TODO

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.

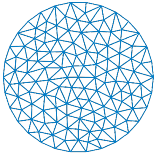
Appendix

Appendix 1 : Standard FEM

Variational Problem : Find $u \in V \mid a(u, v) = l(v), \ \forall v \in V$
with V - Hilbert space, a - bilinear form, l - linear form.

Approach Problem : Find $u_h \in V_h \mid a(u_h, v_h) = l(v_h), \ \forall v_h \in V_h$
with • $u_h \in V_h$ an approximate solution of u,
•$V_h \subset V, \ dimV_h = N_h < \infty, \ (\forall h > 0)$
$\Rightarrow$ Construct a piecewise continuous functions space

$$V_h := P^k_{C,h} = \{v_h \in C^0(\bar{\Omega}), \forall K \in \mathcal{T}_h, v_{h|K} \in \mathbb{P}_k\}$$



$\mathcal{T}_h = \{K_1, \dots, K_{N_e}\}$
(N_e : number of elements)

where $\mathbb{P}_k$ is the vector space of polynomials of total degree $\leq k$.

Finding an approximation of the PDE solution $\Rightarrow$ solving the following linear system:

$$AU = b$$

with

$$A = (a(\varphi_i, \varphi_j))_{1 \leq i,j \leq N_h}, \quad U = (u_i)_{1 \leq i \leq N_h} \quad \text{and} \quad b = (l(\varphi_j))_{1 \leq j \leq N_h}$$

where $(\varphi_1, \dots, \varphi_{N_h})$ is a basis of V_h.

MIMESIS

Appendix 2 : ϕ-FEM

App1

Appendix 3 : Test3

App3