Macaron/Tonus retreat presentation

Mesh-based methods and physically informed learning

Authors:
LECOURTIER Frédérique

Supervisors:
DUPREZ Michel
FRANCK Emmanuel
LLERAS Vanessa

February 6-7, 2024

MIMESIS

Introduction

Scientific context

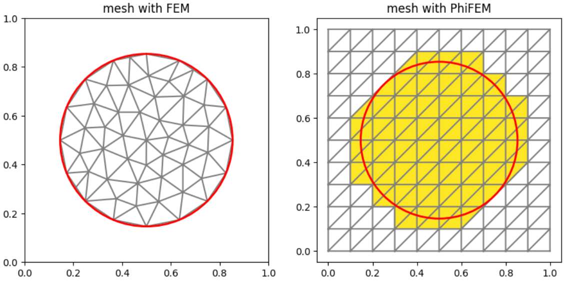
Context : Create real-time digital twins of an organ (such as the liver).

ϕ-FEM Method : New fictitious domain finite element method.

→ domain given by a level-set function $\Rightarrow$ don't require a mesh fitting the boundary

→ allow to work on complex geometries

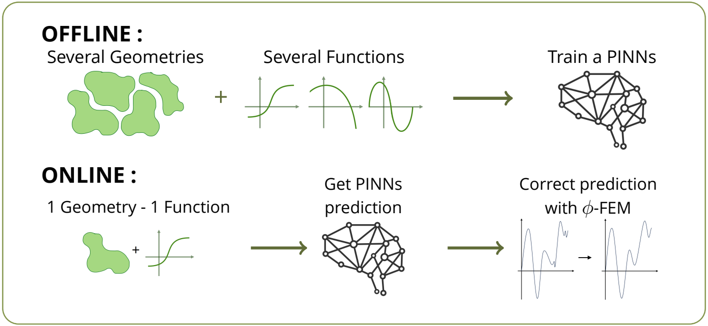
→ ensure geometric quality



Practical case: Real-time simulation, shape optimization...

Objective

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



Evolution :

- Geometry : 2D, simple, fixed (as circle, ellipse..) $\rightarrow$ 3D / complex / variable
- PDE : simple, static (Poisson problem) $\rightarrow$ complex / dynamic (elasticity, hyper-elasticity)
- Neural Network : simple and defined everywhere (PINNs) $\rightarrow$ Neural Operator

Problem considered

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.

Weak formulation :

$$\text{Find } u \in V \text{ such that } a(u, v) = l(v) \forall v \in V$$

with

$$a(u, v) = \int_\Omega (A(x)\nabla u(x)) \cdot \nabla v(x) + c(x)u(x)v(x)\, dx$$

$$l(v) = \int_\Omega f(x)v(x)\, dx$$

Remark : For simplicity, we will not consider 1st order terms.

MIMESIS

Numerical methods

Objective : Show that the philosophy behind most ofd the methods are the same.

Mesh-based methods // Physically informed learning

Numerical methods : Discrete an infinite-dimensional problem (unknown = function) and solve it in a finite-dimensional space (unknown = vector).

- **Encoding :** we encode the problem in a finite-dimensional space
- **Approximation :** solve the problem in finite-dimensional space
- **Decoding :** bring the solution back into infinite dimensional space

Encoding	Approximation	Decoding
$f \rightarrow \theta_f$	$\theta_f \rightarrow \theta_u$	$\theta_u \rightarrow u_\theta$

MIMESIS

Mesh-based methods

Encoding/Decoding

Approximation

Mesh-based methods

Encoding/Decoding

Approximation

Encoding/Decoding - FEMs

- **Decoding :** Linear combination of piecewise polynomial function φ_i.

$$\mathcal{D}_{\theta_u}(x) = \sum_{i=1}^{N} (\theta_u)_i \varphi_i$$

$\Rightarrow$ linear decoding $\Rightarrow$ approximation space V_N = vectorial space

- **Encoding :** Orthogonal projection on vector space V_N. Appendix 1 1

$$\theta_f = E(f) = M^{-1} b(f)$$

with $M_{ij} = \int_\omega \varphi_i(x)\varphi_j(x)$ and $b_i(f) = \int_\omega \varphi_i(x) f(x)$.

Mesh-based methods

Encoding/Decoding

Approximation

Internship results

Correction Methods

Results - with FNO

Other results

Internship results

Correction Methods

We are given u_θ the FNO prediction (for the problem under consideration).

By adding :

We will consider

$$\tilde{u} = u_\theta + \boxed{\tilde{C}} \approx 0$$

We want $\tilde{C} : \Omega \to \mathbb{R}^d$ such that

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

with $\tilde{f} = f + \Delta u_\theta$ and $\tilde{C} = \phi C$ for the ϕ-FEM method.

Remark : In practice, it may be useful to integrate by parts the term containing Δu_θ.

By multiplying :

We will consider

$$\tilde{u} = u_\theta \boxed{C} \approx 1$$

We want $C : \Omega \to \mathbb{R}^d$ such that

$$\begin{cases} -\Delta(u_\theta C) = f, & \text{on } \Omega, \\ C = 1, & \text{on } \Gamma. \end{cases} \qquad (\mathcal{C}_\times)$$

MIMESIS

Correction Methods

We are given u_θ the FNO prediction (for the problem under consideration).

By adding :

We will consider

$$\tilde{u} = u_\theta + \tilde{C}$$

We want $\tilde{C} : \Omega \to \mathbb{R}^d$ such that

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

with $\tilde{f} = f + \Delta u_\theta$ and $\tilde{C} = \phi C$ for the ϕ-FEM method.

Remark : In practice, it may be useful to integrate by parts the term containing Δu_θ.

By multiplying - elevated problem :

Find $\hat{u} : \Omega \to \mathbb{R}^d$ such that

$$\begin{cases} -\Delta \hat{u} = f, & \text{in } \Omega, \\ \hat{u} = g + m, & \text{on } \Gamma, \end{cases} \qquad (\mathcal{P}^{\mathcal{M}})$$

with $\hat{u} = u + m$ (m a constant).

We will consider

$$\tilde{u} = \hat{u_\theta} C$$

with $\hat{u_\theta} = u_\theta + m$.

We want $C : \Omega \to \mathbb{R}^d$ such that

$$\begin{cases} -\Delta(\hat{u_\theta} C) = f, & \text{in } \Omega, \\ C = 1, & \text{on } \Gamma. \end{cases} \qquad (\mathcal{C}_\times^{\mathcal{M}})$$

MIMESIS

Internship results

Explanation

Train a FNO :

Train data : (n_{vert}, n_{vert})

$X_{train} = \{F_i, G_i, \phi_i\}$

$Y_{train} = \{W_i\}$

$i = 1, \ldots, n_{data}$

$\phi - FEM$ solution

→

FNO training :

$$loss_\theta = \frac{1}{n_{data}} \sum_{i=1}^{n_{data}} \|\phi_i W_{\theta,i} - \phi_i W_i\|_2$$

$$Y_{pred} = \{\phi_i W_{\theta,i}\}$$

Correct the predictions of the FNO :

Test data :

$X_{test} = \{F_i, G_i, \phi_i\}$

$Y_{pred} = \{\phi_i W_{\theta,i}\}$

$i = 1, \ldots, n_{test}$

→

Correction :

→ by adding

→ by multiplying

with $FEM / \phi - FEM$

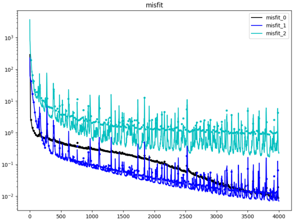
Some important points on the FNO :

- → widely used in PDE solving and constitute an active field of research
- → FNO are Neural Operator networks : Unlike standard neural networks, which learn using inputs and outputs of fixed dimensions, neural operators **learn operators, which are mappings between spaces of functions**.
- → **Mesh resolution independent :** can be evaluated at almost any data resolution without the need for retraining

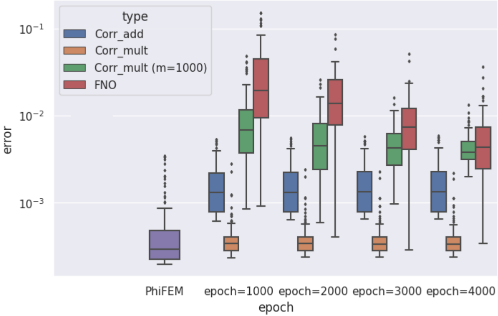
Correction on a FNO prediction - ϕ-FEM

We consider an unknown solution on the circle with f Gaussian (**??**), $n_{vert} = 63$, $n_{data} = 1000$ (including validation sample) and $n_{test} = 100$.

Training on 4000 epochs (bs=64,lr=0.01) :



Correction with the different methods :



Remark : We should try to reduce the resolution for correction, maybe we will gain in the time-to-error ratio.

Internship results

Precision of the prediction - FEM

We consider the trigonometric solution on the circle (??) with

$$u_{ex}(x,y) = S\sin\left(8\pi f\left((x-0.5)^2+(y-0.5)^2\right)+\varphi\right)$$

with $S = 0.5$ and $\varphi = 0$.

Exact solution : Testing different correction methods for different frequencies.

$$u_\theta = u_{ex} \in \mathbb{P}^{10} \; \rightarrow \; \tilde{u} \in \mathbb{P}^1$$

Correction with FEM ($n_{vert} = 100$) :

	fem	Corr_add	Corr_add_IPP	Corr_mult
f = 1	2.10e-03	2.44e-10	1.29e-13	2.97e-13
f = 2	6.62e-03	1.53e-10	1.28e-13	2.80e-13
f = 3	1.41e-02	8.86e-11	1.27e-13	2.68e-13
f = 4	2.42e-02	9.52e-11	1.26e-13	2.61e-13

MIMESIS

Precision of the prediction - FEM

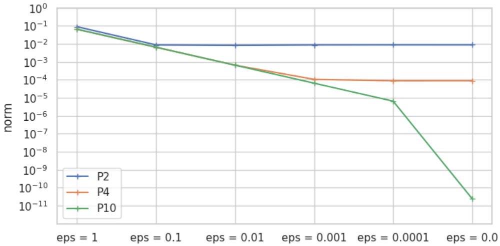
We consider $(S, f, \varphi) = (0.5, 1, 0)$.

Disturbed solution : Testing different ϵ and different degree k.

$$u_\theta = u_{ex} + \epsilon P \in \mathbb{P}^k \rightarrow \tilde{u} \in \mathbb{P}^1$$

with ϵ a real number and P a perturbation.

Correction ($\mathcal{C}_+$) with FEM ($n_{vert} = 32$) :



Results for $k = 10$:

eps	corr_add
1.00e+00	6.57e-02
1.00e-01	6.57e-03
1.00e-02	6.57e-04
1.00e-03	6.57e-05
1.00e-04	6.57e-06
0.00e+00	2.44e-11

Remark : $P(x, y) = S_p \sin\left(8\pi f_p \left((x-0.5)^2 + (y-0.5)^2\right) + \varphi_p\right)$ with $(S_p, f_p, \varphi_p) = (0.5, 2, 0)$

MIMESIS

Theoretical results - FEM

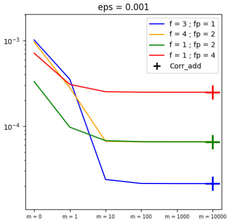
Correction by multiplication on the elevated problem : We consider

- $\hat{u_{ex}} = u_{ex} + m$: the exact solution of $(\mathcal{P}^{\mathcal{M}})$
- $\hat{u_\theta} = u_\theta + m$: a disturbed solution of $(\mathcal{P}^{\mathcal{M}})$.
- $\tilde{u_h} = \hat{u_\theta} C_h$: the approximate solution of $(\mathcal{C}_\times^{\mathcal{M}})$

1. When m tends to infinity :

solution of $(\mathcal{C}_\times^{\mathcal{M}}) \rightarrow$ solution of $(\mathcal{C}_+)$

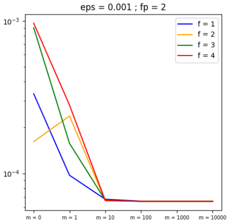
Results : $n_{vert} = 32, \epsilon = 0.001$



2. For m sufficiently large : $C_{ex} = \hat{u_{ex}} / \hat{u_\theta}$

$$||C_{ex} - C_h||_{0,\Omega} \leq ch^{k+1} \epsilon \, ||P''||_{0,\Omega}$$

Results : $n_{vert} = 32, \epsilon = 0.001, f_p = 2$



MIMESIS

PhD results

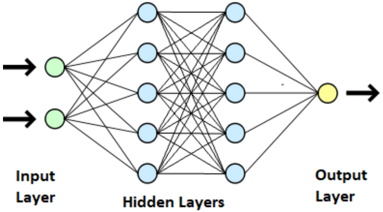
Explanation

Context : Need $u_\theta \in \mathbb{P}^k$ with k of high degree

FNO (on a regular grid) $\rightarrow$ NN which can predict solution at any point

Solutions :

1. MLP - Multi-Layer Perceptron (= Fully connected)



Problem : As the prediction is injected into an FEM solver, the accuracy of the derivatives is very important.

2. PINNs - MLP with a physical loss

$$loss = mse(\Delta(\phi(x_i, y_i) w_{\theta,i}) + f_i)$$

$$\boxed{\begin{aligned} inputs &= \{(x_i, y_i)\} \\ outputs &= \{u_i\}_{i=1,\dots,n_{pts}} \end{aligned}}$$

$u_i = \phi(x_i, y_i) w_\theta(x_i, y_i)$

with $(x_i, y_i) \in \mathcal{O}$.

Remark : We impose exact boundary conditions.

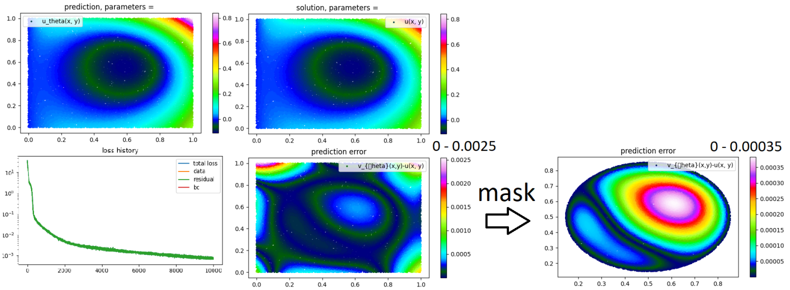
MIMESIS

PINNs Training

We consider the solution on the circle defined in (??) and defined by

$$u_{ex}(x,y) = \phi(x,y)\sin(x)\exp(y)$$

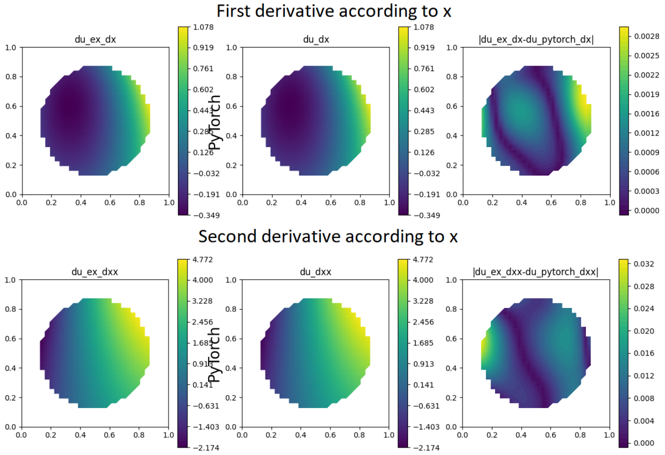
We train a PINNs with 4 layers of 20 neurons over 10000 epochs (with $n_{pts} = 2000$ points selected uniformly over $\mathcal{O}$).



⚠ We consider a single problem (f fixed) on a single geometry (ϕ fixed).

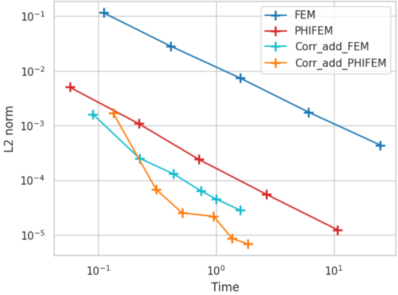
$$||u_{ex} - u_\theta||_{0,\Omega}^{(rel)} \approx 2.81e-3$$

Derivatives



Correction by addition

$u_\theta \in \mathbb{P}^{10} \rightarrow \tilde{u} \in \mathbb{P}^1$



Calculation time (to reach an error of 1e-4)

	mesh	u_PINNs	assemble	solve	TOTAL
FEM	0,08832		29,55516	0,07272	29,71621
PhiFEM	0,33222		1,86924	0,00391	2,20537
Corr_add_FEM	0,00183	0,11187	0,46195	0,00061	0,57626
Corr_add_PhiFEM	0,03213	0,05351	0,22006	0,00040	0,30609

- **mesh** - FEM : construct the mesh (ϕ-FEM : construct cell/facet sets)
- **u_PINNs** - get u_θ in $\mathbb{P}^{10}$ freedom degrees
- **assemble** - assemble the FE matrix
- **solve** - resolve the linear system

FEM / ϕ-FEM : $n_{vert} \in \{8, 16, 32, 64, 128\}$

Corr : $n_{vert} \in \{5, 10, 15, 20, 25, 30\}$

Remark : The stabilisation parameter σ of the ϕ-FEM method has a major impact on the error obtained.

Conclusion

Conclusion

Observations :

1. Correction by addition seems to be the best choice
(based on theoretical results obtained with FEM)

2. We need a high degree prediction ($u_\theta \in \mathbb{P}^{10}$)

$\Rightarrow$ no longer use FNO (needs NN defined at any point)

3. We need to approximate the derivatives of the solution precisely

$\Rightarrow$ no longer use simple MLP, replaced by a PINNs

What's next ?

1. Consider multiple problems (varying f)

2. Consider multiple and more complex geometry (varying ϕ)

3. Replace PINNs with a Neural Operator

Bibliography

MIMESIS

Bibliography

Appendix 1 : Encoding - FEMs

Pourquoi ?