Team meeting presentation

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

Author : LECOURTIER Frédérique
Supervisors : DUPREZ Michel, FRANCK Emmanuel, LLERAS Vanessa
Decembre 12, 2023

MIMESIS

Introduction

Scientific context

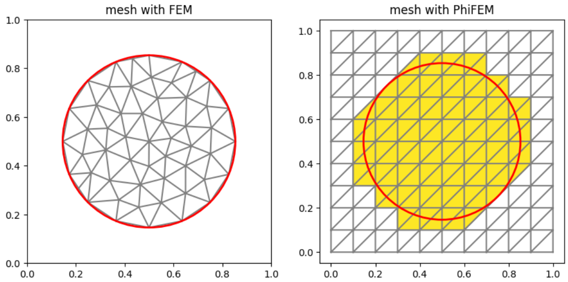
Context : Create real-time digital twins of an organ (such as the liver).

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...

MIMESIS

Objectives

Internship objective : Correct and certify the prediction of a Fourier Neural Operator (FNO), trained with ϕ-FEM solution.

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

OFFLINE

- Learn several geometry representations
- Generate ϕ-FEM solutions as training data on several geometry
- Train a Neural Operator (to map the geometry and the function on the solution)

ONLINE

Data : 1 geometry + 1 function

- Compute representation of 1 geometry and 1 function
- Compute predictions from the Neural Operator
- Use ϕ-FEM to correct the prediction

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

MIMESIS

PDE considered

Poisson problem with Dirichlet conditions :

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

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

with Δ the Laplace operator, Ω a smooth bounded open set and Γ its boundary.

We will define by

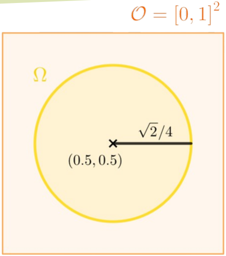
$$||u_{ex} - u_{method}||_{0,\Omega}^{(rel)} = \frac{\int_\Omega (u_{ex} - u_{method})^2}{\int_\Omega u_{ex}^2}$$

the relative error between

- u_{ex} : the exact solution
- u_{method} : the solution obtained by a method (can be : FEM or ϕ-FEM, a correction solver or the prediction of an neural network).

MIMESIS

Problem - Unknown solution on a Circle



→ **Level-set function :**

$$\phi(x,y) = -1/8 + (x - 1/2)^2 + (y - 1/2)^2$$

→ **FNO solution :**

$$f(x,y) = \exp\left(-\frac{(x-\mu_0)^2 + (y-\mu_1)^2}{2\sigma^2}\right) \quad (1)$$

with $\sigma \sim \mathcal{U}([0.1, 0.6])$

$\mu_0, \mu_1 \sim \mathcal{U}(]X_0 - r, X_0 + r[)$

→ **Theoretical experiment solution :**

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

Remark : $\varphi = 0 \Rightarrow u = 0$ on Γ

→ **PINNs solution**

$$u_{ex}(x,y) = \phi(x,y)\sin(x)\exp(y) \quad (3)$$

MIMESIS

Finite Element Methods

Standard FEM method

ϕ-FEM method

Finite Element Methods

Standard FEM method

ϕ-FEM method

Presentation of standard FEM method

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.

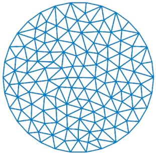
Approach Problem : Find $u_h \in V_h \mid a(u_h, v_h) = l(v_h), \ \forall v_h \in V_h$

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

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



$\mathcal{T}_h = \{K_1, \ldots, K_{N_e}\}$
(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 $(\varphi_1, \ldots, \varphi_{N_h})$ is a basis of V_h.

MIMESIS

Finite Element Methods

Standard FEM method

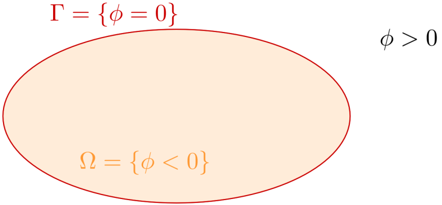
ϕ-FEM method

Problem

Let $u = \phi w + g$ such that

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = g, & \text{on } \Gamma, \end{cases}$$

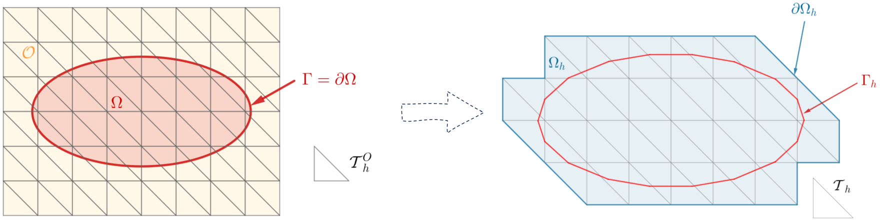
where ϕ is the level-set function and Ω and Γ are given by :



The level-set function ϕ is supposed to be known on $\mathbb{R}^d$ and sufficiently smooth. For instance, the signed distance to Γ is a good candidate.

Remark : Thanks to ϕ and g, the conditions on the boundary are respected.

Fictitious domain

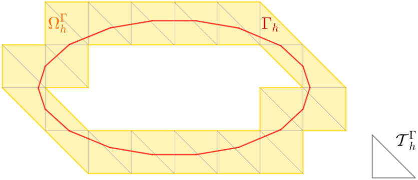


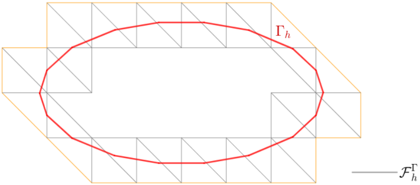
- ϕ_h : approximation of ϕ
- $\Gamma_h = \{\phi_h = 0\}$: approximate boundary of Γ
- Ω_h : computational mesh
- $\partial\Omega_h$: boundary of Ω_h ($\partial\Omega_h \neq \Gamma_h$)

Remark : n_{vert} will denote the number of vertices in each direction for $\mathcal{O}$

MIMESIS

Facets and Cells sets





- $\mathcal{T}_h^\Gamma$: mesh elements cut by Γ_h
- $\mathcal{F}_h^\Gamma$: collects the interior facets of $\mathcal{T}_h^\Gamma$ (either cut by Γ_h or belonging to a cut mesh element)

ϕ-FEM Method - Poisson problem

Approach Problem : Find $w_h \in V_h^{(k)}$ such that

$$a_h(w_h, v_h) = l_h(v_h) \quad \forall v_h \in V_h^{(k)}$$

where

$$a_h(w,v) = \int_{\Omega_h} \nabla(\phi_h w) \cdot \nabla(\phi_h v) - \int_{\partial\Omega_h} \frac{\partial}{\partial n}(\phi_h w)\phi_h v + G_h(w,v),$$

$$l_h(v) = \int_{\Omega_h} f\phi_h v + G_h^{rhs}(v)$$

and

$$V_h^{(k)} = \left\{ v_h \in H^1(\Omega_h) : v_{h|_T} \in \mathbb{P}_k(T),\ \forall T \in \mathcal{T}_h \right\}.$$

For the non homogeneous case, we replace

$$u = \phi w \quad \rightarrow \quad u = \phi w + g$$

by supposing that g is currently given over the entire Ω_h.

MIMESIS

ϕ-FEM Method - Poisson problem

Approach Problem : Find $w_h \in V_h^{(k)}$ such that

$$a_h(w_h, v_h) = l_h(v_h) \quad \forall v_h \in V_h^{(k)}$$

where

$$a_h(w,v) = \int_{\Omega_h} \nabla(\phi_h w) \cdot \nabla(\phi_h v) - \int_{\partial\Omega_h} \frac{\partial}{\partial n}(\phi_h w)\phi_h v + \boxed{G_h(w,v)},$$

$$l_h(v) = \int_{\Omega_h} f\phi_h v + \boxed{G_h^{rhs}(v)}$$

Stabilization terms

and

$$V_h^{(k)} = \left\{ v_h \in H^1(\Omega_h) : v_{h|_T} \in \mathbb{P}_k(T), \ \forall T \in \mathcal{T}_h \right\}.$$

For the non homogeneous case, we replace

$$u = \phi w \quad \rightarrow \quad u = \phi w + g$$

by supposing that g is currently given over the entire Ω_h.

MIMESIS

Stabilization terms

Independent parameter of h

Jump on the interface E

$$G_h(w,v) = \boxed{\sigma h \sum_{E \in \mathcal{F}_h^\Gamma} \int_E \left[\frac{\partial}{\partial n}(\phi_h w)\right]\left[\frac{\partial}{\partial n}(\phi_h v)\right]} + \boxed{\sigma h^2 \sum_{T \in \mathcal{T}_h^\Gamma} \int_T \Delta(\phi_h w)\Delta(\phi_h v)}$$

1st order term

$$G_h^{rhs}(v) = \boxed{-\sigma h^2 \sum_{T \in \mathcal{T}_h^\Gamma} \int_T f \Delta(\phi_h v)}$$

$-$

2nd order term

$$\sigma h^2 \sum_{T \in \mathcal{T}_h^\Gamma} \int_T (\Delta(\phi_h w) + f)\,\Delta(\phi_h v)$$

<u>1st term :</u> ensure continuity of the solution by penalizing gradient jumps.

$\rightarrow$ Ghost penalty [Burman, 2010]

<u>2nd term :</u> require the solution to verify the strong form on Ω_h^Γ.

Purpose :

- reduce the errors created by the "fictitious" boundary
- ensure the correct condition number of the finite element matrix
- restore the coercivity of the bilinear scheme

Internship results

Correction Methods
Results - with FNO
Other results

Internship results

Correction Methods

Results - with FNO

Other 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

Correction Methods
Results - with FNO
Other results

Explanation

Train a FNO :

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

$$X_{train} = \{F_i, G_i, \phi_i\}$$
$$Y_{train} = \{W_i\}_{i=1,\dots,n_{data}}$$

$\phi - FEM$ 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 :

$$X_{test} = \{F_i, G_i, \phi_i\}$$
$$Y_{pred} = \{\phi_i W_{\theta,i}\}_{i=1,\dots,n_{test}}$$

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**.

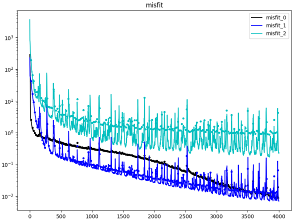
→ **Mesh resolution independent :** can be evaluated at almost any data resolution without the need for retraining

MIMESIS

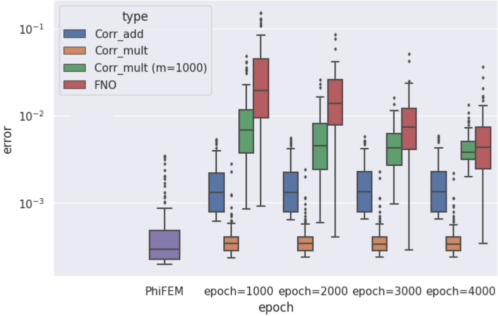
Correction on a FNO prediction - ϕ-FEM

We consider an unknown solution on the circle with f Gaussian (1), $n_{vert} = 63$, $n_{data} = 1000$ (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.

MIMESIS

Internship results

Correction Methods
Results - with FNO
Other results

Precision of the prediction - FEM

We consider the trigonometric solution on the circle (2) 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

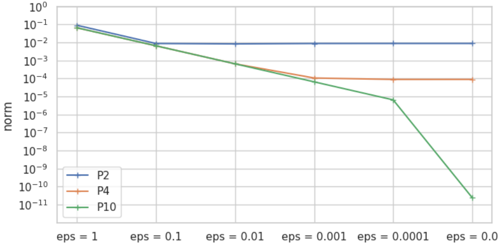
Precision of the prediction - FEM

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

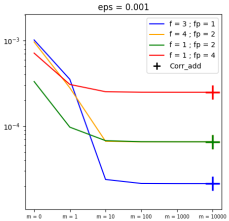
Correction by multiplication on the elevated problem : We consider

- $\hat{u_{ex}} = u_{ex} + m$: 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 $(\mathcal{C}_+)$

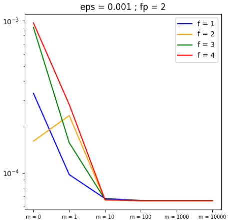
Results : $n_{vert} = 32, \epsilon = 0.001$



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

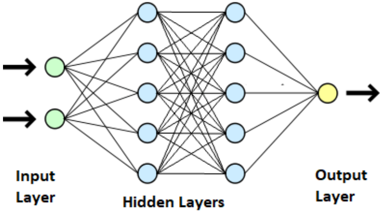
Explanation

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

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

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

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

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

Remark : We impose exact boundary conditions.

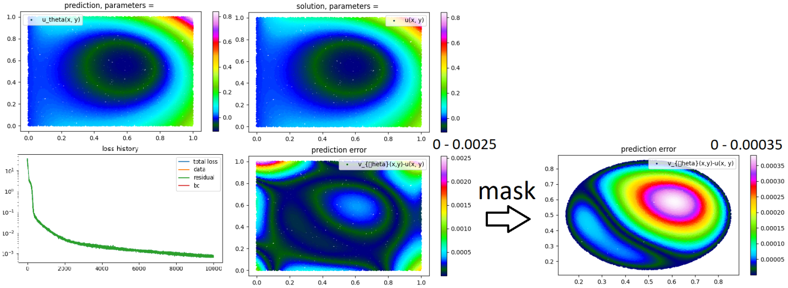
MIMESIS

PINNs Training

We consider the solution on the circle defined in (3) 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 $\mathcal{O}$).

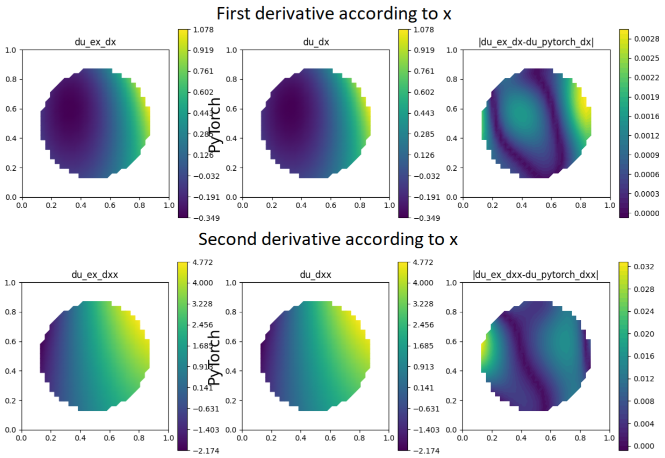


⚠ We consider a single problem (f fixed) on a single geometry (ϕ fixed).

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

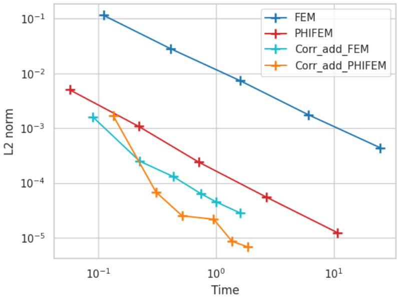
MIMESIS

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.

MIMESIS

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

Bibliography

[1] Erik Burman. “Ghost penalty”. *Comptes Rendus. Mathématique* 348.21 (2010), pp. 1217–1220. ISSN: 1778-3569.

[2] Erik Burman et al. “CutFEM: Discretizing geometry and partial differential equations”. *International Journal for Numerical Methods in Engineering* 104.7 (2015), pp. 472–501. ISSN: 1097-0207.

[3] Stéphane Cotin et al. *ϕ-FEM: an efficient simulation tool using simple meshes for problems in structure mechanics and heat transfer*.

[4] Michel Duprez, Vanessa Lleras, and Alexei Lozinski. “ϕ-FEM: an optimally convergent and easily implementable immersed boundary method for particulate flows and Stokes equations”. *ESAIM: Mathematical Modelling and Numerical Analysis* 57.3 (May 2023), pp. 1111–1142. ISSN: 2822-7840, 2804-7214.

[5] Michel Duprez, Vanessa Lleras, and Alexei Lozinski. “A new ϕ-FEM approach for problems with natural boundary conditions”. *Numerical Methods for Partial Differential Equations* 39.1 (2023), pp. 281–303. ISSN: 1098-2426.

[6] Michel Duprez and Alexei Lozinski. “ϕ-FEM: A Finite Element Method on Domains Defined by Level-Sets”. *SIAM Journal on Numerical Analysis* 58.2 (Jan. 2020), pp. 1008–1028. ISSN: 0036-1429.

[7] Zongyi Li et al. *Neural Operator: Graph Kernel Network for Partial Differential Equations*. Mar. 6, 2020.

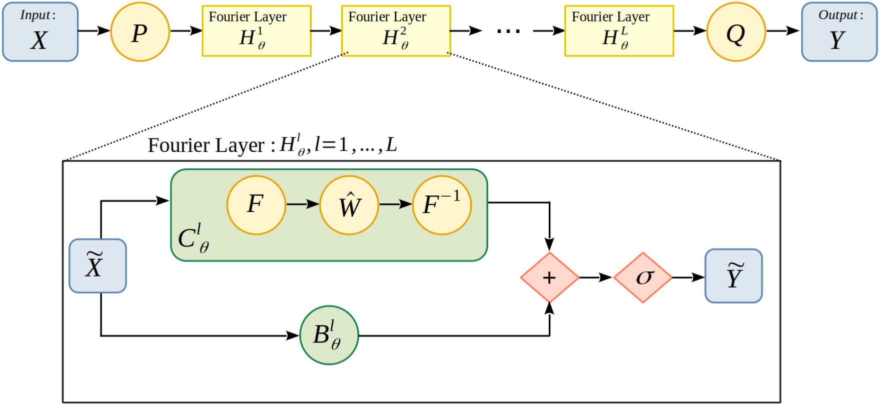
[8] Zongyi Li et al. *Fourier Neural Operator for Parametric Partial Differential Equations*. May 16, 2021.

[9] Zongyi Li et al. *Fourier Neural Operator with Learned Deformations for PDEs on General Geometries*. July 11, 2022.

[10] Zongyi Li et al. *Physics-Informed Neural Operator for Learning Partial Differential Equations*. July 29, 2023.

[11] 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* 389 (Feb. 2022), p. 114333. ISSN: 00457825.

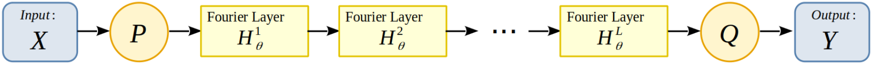
Architecture of the FNO



Input *X* of shape (bs,ni,nj,nk) **Output** *Y* of shape (bs,ni,nj,1)
with bs the batch size, ni and nj the grid resolution and nk the number of channels.

MIMESIS

Description of the FNO architecture



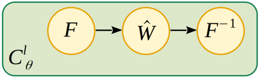
- perform a P transformation, to move to a space with more channels (to build a sufficiently rich representation of the data)
- apply L Fourier layers defined by

$$\mathcal{H}_\theta^l(\tilde{X}) = \sigma\left(\mathcal{C}_\theta^l(\tilde{X}) + \mathcal{B}_\theta^l(\tilde{X})\right), \quad l = 1, \ldots, L$$

  with $\tilde{X}$ the input of the current layer and
  - σ an activation function (ReLU or GELU)
  - $\mathcal{C}_\theta^l$: convolution sublayer (convolution performed by Fast Fourier Transform)
  - $\mathcal{B}_\theta^l$: "bias-sublayer"
- return to the target dimension by performing a Q transformation (in our case, the number of output channels is 1)

Fourier Layer Structure

Convolution sublayer : $\mathcal{C}_\theta^l(X) = \mathcal{F}^{-1}(\mathcal{F}(X) \cdot \hat{W})$



- $\hat{W}$: a trainable kernel
- $\mathcal{F}$: 2D Discrete Fourier Transform (DFT) defined by

$$\mathcal{F}(X)_{ijk} = \frac{1}{ni}\frac{1}{nj}\sum_{i'=0}^{ni-1}\sum_{j'=0}^{nj-1} X_{i'j'k} e^{-2\sqrt{-1}\pi\left(\frac{ii'}{ni}+\frac{jj'}{nj}\right)}$$

  $\mathcal{F}^{-1}$: its inverse.
- $(Y \cdot \hat{W})_{ijk} = \sum_{k'} Y_{ijk'}\hat{W}_{ijk'}$ $\Rightarrow$ applied channel by channel

Bias-sublayer : $\mathcal{B}_\theta^l(X)_{ijk} = \sum_{k'} X_{ijk} W_{k'k} + B_k$ B_θ^l

- 2D convolution with a kernel of size 1
- allowing channels to be mixed via a kernel without allowing interaction between pixels.

MIMESIS

Dual method - Poisson Problem

Problem : Find u on Ω_h and p on Ω_h^Γ such that

$$\int_{\Omega_h} \nabla u \nabla v - \int_{\partial\Omega_h} \frac{\partial u}{\partial n} v + \frac{\gamma}{h^2} \sum_{T \in \mathcal{T}_h^\Gamma} \int_T \left(u - \frac{1}{h}\phi p\right)\left(v - \frac{1}{h}\phi q\right)$$
$$+ G_h(u, v) = \int_{\Omega_h} fv + G_h^{rhs}(v), \ \forall v \text{ on } \Omega_h, \ q \text{ on } \Omega_h^\Gamma$$

with γ an other positive stabilization parameter and G_h and G_h^{rhs} the stabilization terms defined previously.
For the non homogeneous case, we replace

$$\int_T \left(u - \frac{1}{h}\phi p\right)\left(v - \frac{1}{h}\phi q\right) \quad \rightarrow \quad \int_T \left(u - \frac{1}{h}\phi p - g\right)\left(v - \frac{1}{h}\phi q\right)$$

by assuming g is defined on Ω_h^Γ

MIMESIS