Innovative non-conformal finite element methods for augmented surgery

Internship presentation

LECOURTIER Frédérique, DUPREZ Michel, FRANCK Emmanuel, LLERAS Vanessa

Strasbourg University

August 24, 2023

MIMESIS

Introduction

Presentation of the teams

MIMESIS

- project-team as sub-team of MLMS ("Machine Leraning, Modélisation et Simulation") of Inria
- **Aim :** to create real-time digital twins of an organ
- **Scientific challenges :**
  - scientific computing
  - data assimilation
  - machine learning
  - control
- **Main application domains :**
  - surgical training
  - surgical guidance during complex interventions

Scientific context

A COMPLETER !!

Objectives - Deliverables

Objectives :

- Train a Fourier Neural Operator (FNO), with ϕ-FEM solutions, to predict the solutions of a given PDE.
- Apply a correction on the FNO predictions.

Aim : Neural networks are very fast, but not very accurate
=> finite element methods are used to improve prediction accuracy.

Deliverables :

- a weekly tracking report (in French)
- a github repository containing all the code allowing to reproduce the results presented in this report
- a report of the internship
- an online report generated with a tool called antora (made by a github CI)

Problem considered

Poisson problem with Dirichlet conditions :

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

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

with Δ the Laplace operator, Ω a smooth bounded open set and $\partial\Omega$ its boundary.

General methods and tools

Standard FEM method

ϕ-FEM method

Fourier Neural Operator (FNO)

General methods and tools

Standard FEM method

ϕ-FEM method

Fourier Neural Operator (FNO)

Presentation of standard FEM method

Variational Problem :

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

where V is a Hilbert space, a is a bilinear form and l is a linear form.

Approach Problem :

$$\text{Find } u_h \in V_h \text{ such that } a(u_h, v_h) = l(v_h), \ \forall v_h \in V$$

with u_h an approximate solution in V_h, a finite-dimensional space dependent on h such that $V_h \subset V, \ dimV_h = N_h < \infty, \ \forall h > 0$

As $u_h = \sum_{i=1}^{N_h} u_i \varphi_i$ with $(\varphi_1, \ldots, \varphi_{N_h})$ a basis of V_h, finding an approximation of the PDE solution implies solving the following linear system:

$$AU = b$$

with

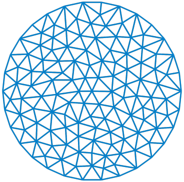
$$A = (a(\varphi_i, \varphi_j))_{1 \leq i,j \leq N_h}, \quad U = (u_i)_{1 \leq i \leq N_h} \quad \text{and} \quad b = (l(\varphi_j))_{1 \leq j \leq N_h}$$

In practice

- Construct a mesh of our Ω geometry with a family of elements (in 2D: triangle, rectangle; in 3D: tetrahedron, parallelepiped, prism) defined by

$$\mathcal{T}_h = \{K_1, \dots, K_{N_e}\}$$

where N_e is the number of elements.



- Construct a space of piece-wise affine continuous functions, defined by

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

where $\mathbb{P}_k$ is a vector space of polynomials of total degree less than or equal to k.

General methods and tools

Standard FEM method

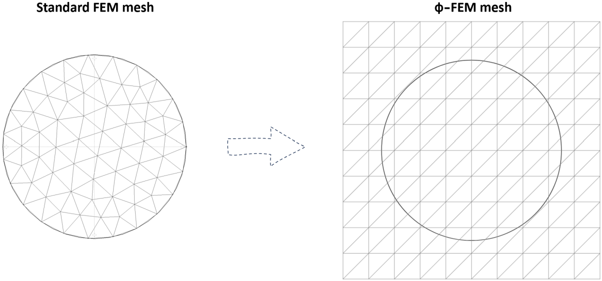
ϕ-FEM method

Fourier Neural Operator (FNO)

MIMESIS

Context

Idea : ϕ-FEM method = new fictitious domain finite element method that does not require a mesh conforming to the real boundary.



Advantage : boundary represented by a level-set function $\Rightarrow$ only this function will change over time during real-time simulation

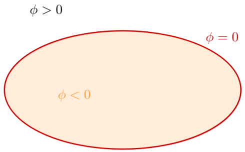
Problem

We pose $u = \phi w$ such that

$$\begin{cases} -\Delta(\phi w) = f, \text{ on } \Omega, \\ u = g, \text{ in } \Gamma, \end{cases}$$

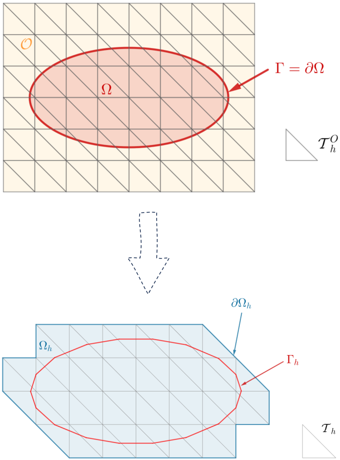
where ϕ is the level-set function and Ω and Γ are given by :

$$\Omega = \{\phi < 0\} \quad \text{and} \quad \Gamma = \{\phi = 0\}.$$



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

Fictitious domain



- $\mathcal{O}$: fictitious domain such that $\Omega \subset \mathcal{O}$
- $\mathcal{T}_h^{\mathcal{O}}$: simple quasi-uniform mesh on $\mathcal{O}$
- $\phi_h = I_{h,\mathcal{O}}^{(l)}(\phi) \in V_{h,\mathcal{O}}^{(l)}$: approximation of ϕ with $I_{h,\mathcal{O}}^{(l)}$ the standard Lagrange interpolation operator on

$$V_{h,\mathcal{O}}^{(l)} = \left\{ v_h \in H^1(\mathcal{O}) : v_{h|_T} \in \mathbb{P}_l(T)\ \forall T \in \mathcal{T}_h^{\mathcal{O}} \right\}$$

- $\Gamma_h = \{\phi_h = 0\}$: approximate boundary of Γ
- $\mathcal{T}_h$: sub-mesh of $\mathcal{T}_h^{\mathcal{O}}$ defined by

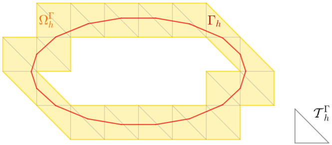
$$\mathcal{T}_h = \left\{ T \in \mathcal{T}_h^{\mathcal{O}} : T \cap \{\phi_h < 0\} \neq \emptyset \right\}$$

- Ω_h : domain covered by the $\mathcal{T}_h$ mesh defined by

$$\Omega_h = \left(\cup_{T \in \mathcal{T}_h} T \right)^O$$

($\partial\Omega_h$ its boundary)

Facets and Cells sets

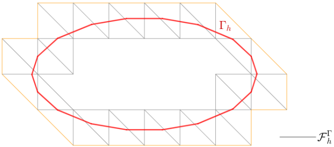


- $\mathcal{T}_h^\Gamma \subset \mathcal{T}_h$: contains the mesh elements cut by Γ_h, i.e.

$$\mathcal{T}_h^\Gamma = \{T \in \mathcal{T}_h : T \cap \Gamma_h \neq \emptyset\},$$

- Ω_h^Γ : domain covered by the $\mathcal{T}_h^\Gamma$ mesh, i.e.

$$\Omega_h^\Gamma = \left(\cup_{T \in \mathcal{T}_h^\Gamma} T\right)^o$$



- $\mathcal{F}_h^\Gamma$: collects the interior facets of $\mathcal{T}_h$ either cut by Γ_h or belonging to a cut mesh element, i.e.

$$\mathcal{F}_h^\Gamma = \{E \text{ (an internal facet of } \mathcal{T}_h\text{) such that } \exists T \in \mathcal{T}_h : T \cap \Gamma_h \neq \emptyset \text{ and } E \in \partial T\}$$

Application to the Poisson problem

We start by consider the **homogeneous case** ($g = 0$ on Γ).

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

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

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

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

1st term : Ghost penality [4], ensure continuity of the solution by penalizing gradient jumps.
2nd term : 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
- permit to restore the coercivity of the bilinear scheme

MIMESIS

Non-homogeneous case

TO DO !

General methods and tools

Standard FEM method

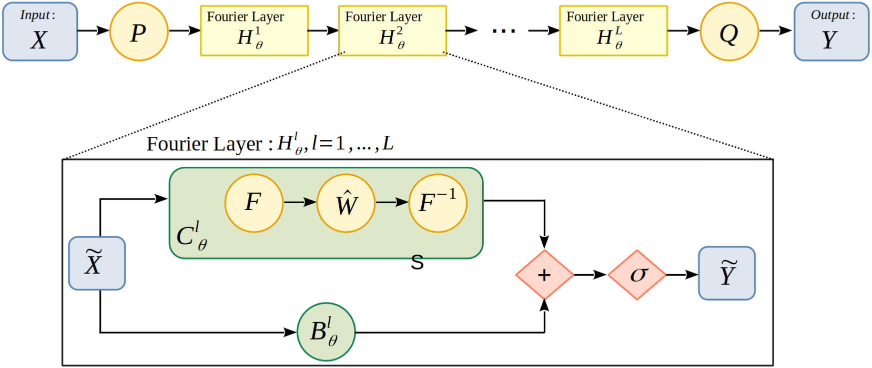
ϕ-FEM method

Fourier Neural Operator (FNO)

Presentation

- ➙ 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**.
- ➙ can be evaluated at any data resolution without the need for retraining

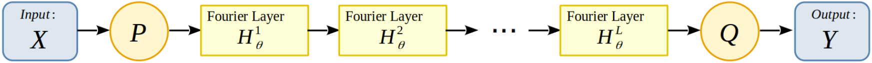
Architecture of the FNO



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

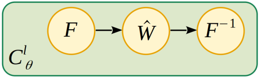


- perform a P transformation, to move to a space with more channels (to build a sufficiently rich representation of the data)
- apply L Fourier layers defined by

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

  with $\tilde{X}$ the input of the current layer and
  - σ an activation function (ReLU or GELU)
  - $\mathcal{C}^l_\theta$: convolution sublayer (convolution performed by Fast Fourier Transform)
  - $\mathcal{B}^l_\theta$: "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 : $\mathcal{C}^l_\theta(X) = \mathcal{F}^{-1}(\mathcal{F}(X) \cdot \hat{W})$

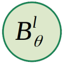
- $\hat{W}$: a trainable kernel
- $\mathcal{F}$: 2D Discrete Fourier Transform (DFT) defined by

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

$\mathcal{F}^{-1}$: its inverse.

- $(Y \cdot \hat{W})_{ijk} = \sum_{k'} Y_{ijk'} \hat{W}_{ijk'}$ $\Rightarrow$ applied channel by channel

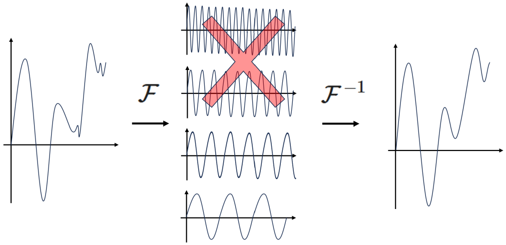
Bias-sublayer : $\mathcal{B}^l_\theta(X)_{ijk} = \sum_{k'} X_{ijk} W_{k'k} + B_k$



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

Some details on the FNO

- **Mesh resolution independent :**
  - P and Q = fully-connected multi-layer perceptron $\Rightarrow$ perform local transformations at each point
  - Fourier layers also independent of mesh resolution : learn in Fourier space so the value of the Fourier modes does not depend on the mesh resolution
- **Low pass filter :** truncate high Fourier modes to ignore high frequencies $\Rightarrow$ enable a kind of regularization that helps the generalization



Application

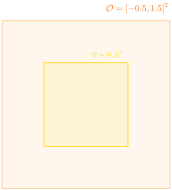
TO DO !

Correction

Methods considered
Theoretical results
Numerical results

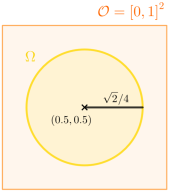
Problems considered

1st problem considered : Trigonometric solution on a Square.



- Level-set function (for formulation) : $\phi(x,y) = x(1-x)y(1-y)$
- Level-set function (for construction) : $\phi_c(X) = ||X - 0.5||_\infty - 0.5$
- Analytical solution : $u_{ex}(x,y) = S \times sin\,(2\pi fx + \varphi) \times sin\,(2\pi fy + \varphi)$
  - $S \in [0,1]$: amplitude of the signal
  - $f \in \mathbb{N}$: "frequency" of the signal
  - $\varphi \in [0,1]$: phase at the origin
- Source term : $f(x,y) = 8\pi^2 Sf^2 \sin\,(2\pi fx + \varphi) \sin\,(2\pi fy + \varphi)$

2nd problem considered : Unknown solution on a Circle.



- Level-set function : $\phi(x,y) = -1/8 + (x-1/2)^2 + (y-1/2)^2$
- Source term : $f(x,y) = \exp\left(-\frac{(x-\mu_0)^2+(y-\mu_1)^2}{2\sigma^2}\right)$
  - $\sigma \sim \mathcal{U}([0.1, 0.6])$
  - $\mu_0, \mu_1 \sim \mathcal{U}([0.5 - \sqrt{2}/4, 0.5 + \sqrt{2}/4])$
- a reference solution u_{ref} : over-refined $\mathbb{P}^1$ solution (with FEM)

Correction

Methods considered

Theoretical results

Numerical results

Correction by adding

We are given $\tilde{\phi}$ an "initial" solution to the problem under consideration.
We will consider

$$\tilde{u} = \tilde{\phi} + \tilde{C}$$

We want to find $\tilde{C} : \Omega \to \mathbb{R}^d$ solution to the problem

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

Rewriting the problem, we seek to find $\tilde{C} : \Omega \to \mathbb{R}^d$ solution to the problem

$$\begin{cases} -\Delta \tilde{C} = \tilde{f}, & \text{on } \Omega, \\ \tilde{C} = 0, & \text{in } \Gamma. \end{cases}$$

with $\tilde{f} = f + \Delta \tilde{\phi}$.
In practice, it may be useful to integrate by parts the term containing $\Delta \tilde{\phi}$.

Correction by multiplying

We will considering

$$\tilde{u} = \tilde{\phi} C$$

We want to find $C : \Omega \to \mathbb{R}^d$ solution to the problem

$$\begin{cases} -\Delta(\tilde{\phi} C) = f, & \text{on } \Omega, \\ C = 1, & \text{on } \Gamma. \end{cases}$$

In the non-homogeneous case, it is important to impose the boundary conditions either by the direct method or by the dual method.

Correction by multiplying (elevated problem)

We introduced an initial modified problem : Find $\hat{u} : \Omega \to \mathbb{R}^d$ such that

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

with $\hat{u} = u + m$ and m a constant.
We then apply the multiplication correction on the elevated problem by considering

$$\hat{\phi} = \tilde{\phi} + m$$

and so we look for $C : \Omega \to \mathbb{R}^d$ solution to the problem

$$\begin{cases} -\Delta(\hat{\phi} C) = f, & \text{in } \Omega, \\ C = 1, & \text{on } \Gamma. \end{cases}$$

In the case of this correction, it is important to impose the boundary conditions either by the direct method, or by the dual method.

Correction

Methods considered

Theoretical results

Numerical results

Correction

Methods considered

Theoretical results

Numerical results

Conclusion

Bibliography

[1] Allal Bedlaoui. *Les éléments finis : de la théorie à la pratique*. URL: https://www.academia.edu/36497995/Les_%C3%A91%C3%A9ments_finis_de_la_th%C3%A9orie_%C3%A0_la_pratique (visited on 08/16/2023).

[2] Allal Bedlaoui. *Les éléments finis : de la théorie à la pratique*. URL: https://www.academia.edu/36497995/Les_%C3%A91%C3%A9ments_finis_de_la_th%C3%A9orie_%C3%A0_la_pratique (visited on 08/16/2023).

[3] Haim Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. New York, NY: Springer New York, 2011. ISBN: 978-0-387-70913-0 978-0-387-70914-7. DOI: 10.1007/978-0-387-70914-7. URL: https://link.springer.com/10.1007/978-0-387-70914-7 (visited on 08/17/2023).

[4] Erik Burman. "Ghost penalty". In: *Comptes Rendus. Mathématique* 348.21 (2010). Number: 21-22, pp. 1217–1220. ISSN: 1778-3569. DOI: 10.1016/j.crma.2010.10.006. URL: http://www.numdam.org/articles/10.1016/j.crma.2010.10.006/ (visited on 07/26/2023).

[5] Erik Burman et al. "CutFEM: Discretizing geometry and partial differential equations". In: *International Journal for Numerical Methods in Engineering* 104.7 (2015). Number: 7 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.4823, pp. 472–501. ISSN: 1097-0207. DOI: 10.1002/nme.4823. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4823 (visited on 07/26/2023).

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

[7] Michel Duprez, Vanessa Lleras, and Alexei Lozinski. “ϕ-FEM: an optimally convergent and easily implementable immersed boundary method for particulate flows and Stokes equations”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 57.3 (May 2023). Number: 3, pp. 1111–1142. ISSN: 2822-7840, 2804-7214. DOI: 10.1051/m2an/2023010. URL: https://www.esaim-m2an.org/10.1051/m2an/2023010 (visited on 07/26/2023).

[8] Michel Duprez, Vanessa Lleras, and Alexei Lozinski. “A new ϕ-FEM approach for problems with natural boundary conditions”. In: *Numerical Methods for Partial Differential Equations* 39.1 (2023). Number: 1 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/num.22878, pp. 281–303. ISSN: 1098-2426. DOI: 10.1002/num.22878. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/num.22878 (visited on 07/26/2023).

[9] Michel Duprez and Alexei Lozinski. “ϕ-FEM: A Finite Element Method on Domains Defined by Level-Sets”. In: *SIAM Journal on Numerical Analysis* 58.2 (Jan. 2020). Number: 2 Publisher: Society for Industrial and Applied Mathematics, pp. 1008–1028. ISSN: 0036-1429. DOI: 10.1137/19M1248947. URL: https://epubs.siam.org/doi/10.1137/19M1248947 (visited on 07/26/2023).

[10] Zongyi Li et al. *Neural Operator: Graph Kernel Network for Partial Differential Equations*. Mar. 6, 2020. DOI: 10.48550/arXiv.2003.03485. arXiv: 2003.03485[cs,math,stat]. URL: http://arxiv.org/abs/2003.03485 (visited on 08/17/2023).

[11] Zongyi Li et al. *Fourier Neural Operator for Parametric Partial Differential Equations*. Issue: arXiv:2010.08895. May 16, 2021. DOI: 10.48550/arXiv.2010.08895. arXiv: 2010.08895[cs,math]. URL: http://arxiv.org/abs/2010.08895 (visited on 07/26/2023).

[12] Zongyi Li et al. *Fourier Neural Operator with Learned Deformations for PDEs on General Geometries*. July 11, 2022. DOI: 10.48550/arXiv.2207.05209. arXiv: 2207.05209[cs,math]. URL: http://arxiv.org/abs/2207.05209 (visited on 08/17/2023).

[13] Zongyi Li et al. *Physics-Informed Neural Operator for Learning Partial Differential Equations*. July 29, 2023. DOI: 10.48550/arXiv.2111.03794. arXiv: 2111.03794[cs,math]. URL: http://arxiv.org/abs/2111.03794 (visited on 08/17/2023).

[14] *Méthodes numériques pour les EDP*. URL: https://feelpp.github.io/csmi-edp/#/ (visited on 08/16/2023).

[15] Nicolas Moës and Ted Belytschko. "X-FEM, de nouvelles frontières pour les éléments finis". In: *Revue Européenne des Éléments Finis* 11.2 (Jan. 2002). Number: 2-4, pp. 305–318. ISSN: 1250-6559. DOI: 10.3166/reef.11.305-318. URL: https://www.tandfonline.com/doi/full/10.3166/reef.11.305-318 (visited on 07/26/2023).

[16] Alfio Quarteroni et al. *Méthodes numériques: algorithmes, analyse et applications*. Ed. revue et augmentée. Milan: Springer, 2007. ISBN: 978-88-470-0495-5.