1st CSI

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

Authors:
Frédérique LECOURTIER

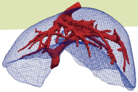
Supervisors:
Emmanuel FRANCK
Michel DUPREZ
Vanessa LLERAS

June 14, 2024

MIMESIS

Introduction

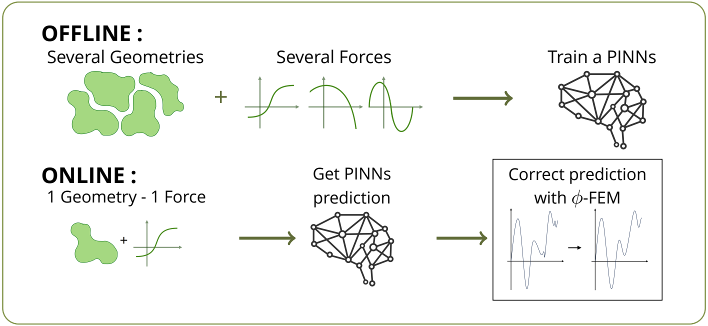
Scientific context



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

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

accurate (finite element) — quick + parameterized (neural network)



ϕ-FEM : New fictitious domain finite element method.

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

Appendix 2

MIMESIS

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

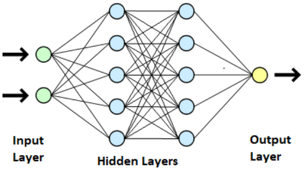
How to deal with complex geometry in PINNs ?

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

$$\theta_u = \operatorname*{argmin}_{\theta \in \mathbb{R}^N} \alpha J_{in}(\theta) + \beta J_{bc}(\theta)$$

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.

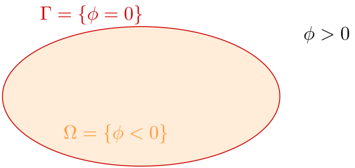
MIMESIS

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 explode

$\Rightarrow$ we need a regular levelset

Learn a regular levelset

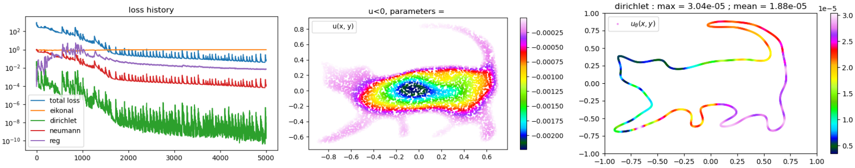
If we have a boundary domain Γ, the SDF is solution to the Eikonal equation:

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

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

How to do that ? with a PINNs [2] by adding a regularization term.

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

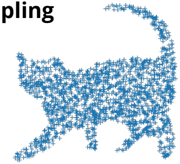


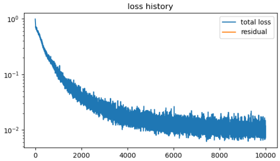
Poisson On Cat

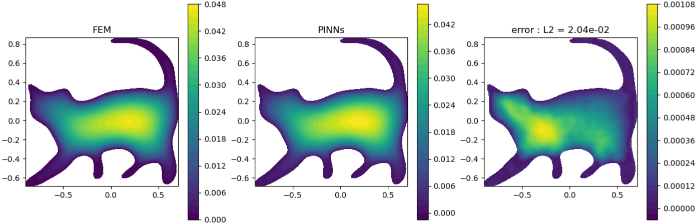
→ Solving the Poisson problem with $f = 1$ and homogeneous Dirichlet BC.

→ Looking for $u_\theta = \phi w_\theta$ with ϕ the levelset learned.

Sampling







Remark : Poisson on Bean Appendix 3

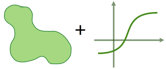
MIMESIS

How improve PINNs prediction ?

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

Idea

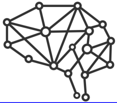
1 Geometry + 1 Force



ϕ and f

(and g)

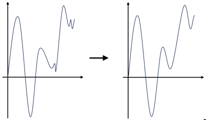
Get PINNs prediction



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

$u_{NN} = g$ on Γ

Correct prediction with FEM



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

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

$$\begin{cases} L(\tilde{C}) = \tilde{f}, & \text{in } \Omega, \\ \tilde{C} = 0, & \text{on } \Gamma, \end{cases} \tag{2}$$

with $\tilde{f} = f - L(u_{NN})$. Appendix 1 Appendix 5

Poisson on Square

Solving the Poisson problem with homogeneous Dirichlet BC.

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

→ **Analytical levelset function :**

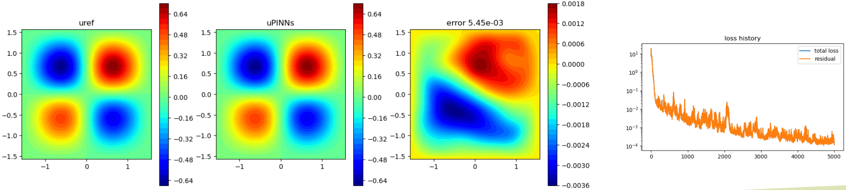
$$\phi(x,y) = (x - 0.5\pi)(x + 0.5\pi)(y - 0.5\pi)(y + 0.5\pi)$$

→ **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



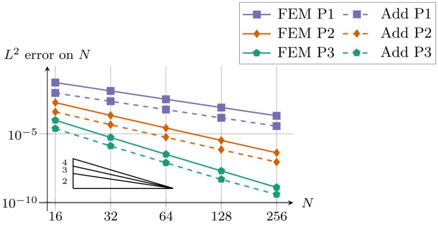
Theoretical results

We denote u the solution of (1) and u_h the discrete solution of the correction problem (10) with V_h a $\mathbb{P}_k$ Lagrange space. Thus

$$||u - u_h||_0 \leq \boxed{\frac{|u - u_\theta|_{H^{k+1}}}{|u|_{H^{k+1}}}} \left(\frac{\gamma}{\alpha} C h^{k+1} |u|_{H^{k+1}}\right)$$

with α and γ respectively the coercivity and continuity constant.

Taking $\mu_1 = 0.05, \mu_2 = 0.22.$



MIMESIS

Gains using our approach

Considering a set of 50 parameters.

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 $\mathbb{P}_2$

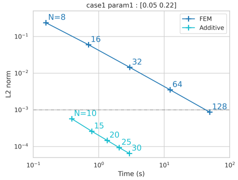
	Gains on PINNs				Gains on FEM			
N	**min**	**max**	**mean**	**std**	**min**	**max**	**mean**	**std**
20	244.81	996.23	655.08	153.63	67.12	165.13	135.21	21.37
40	2,056.2	8,345.4	5,504.89	1,287.16	66.52	159.73	132.05	20.38

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

MIMESIS

Time/Precision



Taking $\mu_1 = 0.05, \mu_2 = 0.22.$

	N		time (s)	
Precision	**FEM**	**Add**	**FEM**	**Add**
$1e-3$	120	8	$\boxed{43}$	$\boxed{0.24}$
$1e-4$	373	25	423.89	1.93
			t_{FEM}	t_{Add}

The training time of the PINNs (parametric) is defined by $t_{PINNs} \approx 240s$.
So if we solve the problem (1) for a set of n_p parameters, the times of our approach and FEM are respectively

$$Tot_{Add} = t_{PINNs} + n_p t_{Add} \quad \text{and} \quad Tot_{FEM} = n_p t_{FEM}.$$

So if we consider a set of at least $n_p = 6$ parameters, our method is faster than FEM when considering network training time.

$$n_p > \frac{t_{PINNs}}{t_{FEM} - t_{Add}} \approx 5.61$$

Remark: Considering that the times are of the same order for each parameter considered.

Conclusion

Conclusion

Current progress :

→ Levelset learning works on complex geometries
Advantage : enables “exact” imposition of BC in PINNs

→ Additive approach works on simple geometries
Advantage (compared with standard FEM) :

- More accurate solution (smaller error)
- Better execution time

Perspectives :

→ combine the 2 axis to improve NN predictions on complex geometries Appendix 4

→ use ϕ-FEM (fictitious domain method) to improve NN predictions
Advantage : The levelset learned by PINNs can be used in ϕ-FEM

→ Start considering 3D cases

Supplementary work I

Temporary employment at the university

- 16h of Computer Science Practical Work (Python) - L2S3
- 34h of Computer Science Practical Work (C++) - L3S6

Formations

- ”Charte de déontologie des métiers de la Recherche” (OBLIGATORY)
- MOOC Bordeaux - ”Intégrité scientifique dans les métiers de la recherche” (OBLIGATORY)
- ”Enseigner et apprendre (public : mission enseignement)”
- ”Gérer ses ressources bibliographiques avec Zotero”
- 3 Workshops on EDP at IRMA
- 19 Remote Sessions ($\approx$ 40h) - ”Formation Introduction au Deep Leraning” (FIDLE)

Supplementary work II

Talks

- Team meeting (Mimesis) - December 12, 2023 - ”Development of hybrid finite element/neural network methods to help create digital surgical twins”
- Retreat (Macaron/Tonus) - February 6, 2024
  ”Mesh-based methods and physically informed learning”
- Exama project, WP2 reunion - March 26, 2024
  ”How to work with complex geometries in PINNs ?”

Coming soon...

- Paper in progress - ”Enhanced finite element methods using neural networks”
  Contribution: numerical results
- Poster for a Workshop on Scientific Machine Learning (SciML 2024)

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 : Standard FEM

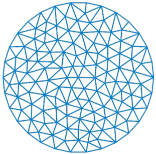
Appendix 1 : General Idea

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

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

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



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

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

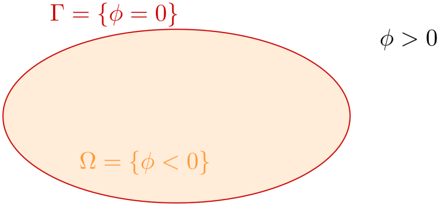
Appendix 2 :ϕ-FEM

Appendix 2 : Problem

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

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

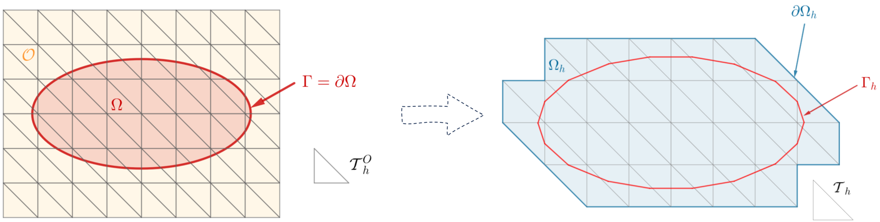
where ϕ is the level-set function and Ω and Γ are given by :



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 boundary conditions are respected.

Appendix 2 : Fictitious domain

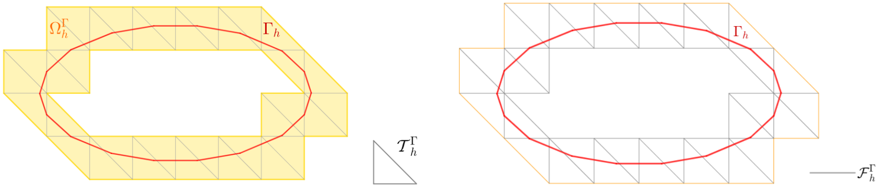


- ϕ_h : approximation of ϕ
- $\Gamma_h = \{\phi_h = 0\}$: approximate boundary of Γ
- Ω_h : computational mesh
- $\partial\Omega_h$: boundary of Ω_h ($\partial\Omega_h \neq \Gamma_h$)

Remark : n_{vert} will denote the number of vertices in each direction

MIMESIS

Appendix 2 : 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)

Appendix 2 : 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.

Appendix 2 : 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

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

1st term : ensure continuity of the solution by penalizing gradient jumps.

$\rightarrow$ Ghost penalty [Burman, 2010]

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
- restore the coercivity of the bilinear scheme

Other results

- Poisson on Bean
- Additive approach on Cat
- Multiplicative approach
- Degree of PINNs evaluation

Other results

- Poisson on Bean
- Additive approach on Cat
- Multiplicative approach
- Degree of PINNs evaluation

Appendix 3 : Learn a levelset

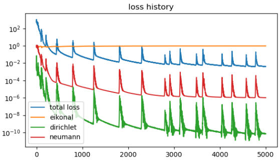
If we have a boundary domain Γ, the SDF is solution to the Eikonal equation:

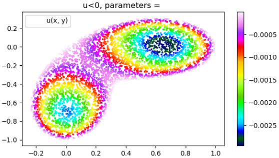
$$\begin{cases} ||\nabla\phi(X)|| = 1, \; X \in \mathcal{O} \\ \phi(X) = 0, \; X \in \Gamma \\ \nabla\phi(X) = n, \; X \in \Gamma \end{cases}$$

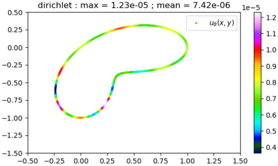
with $\mathcal{O}$ a box which contains Ω completely and n the exterior normal to Γ.

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

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



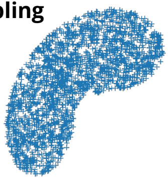


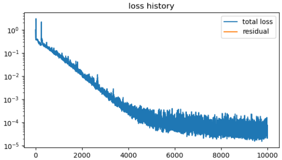


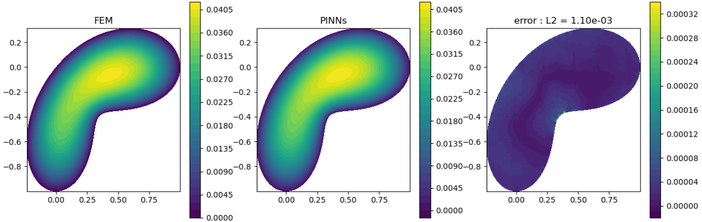
Appendix 3 : Poisson 2D

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

Sampling



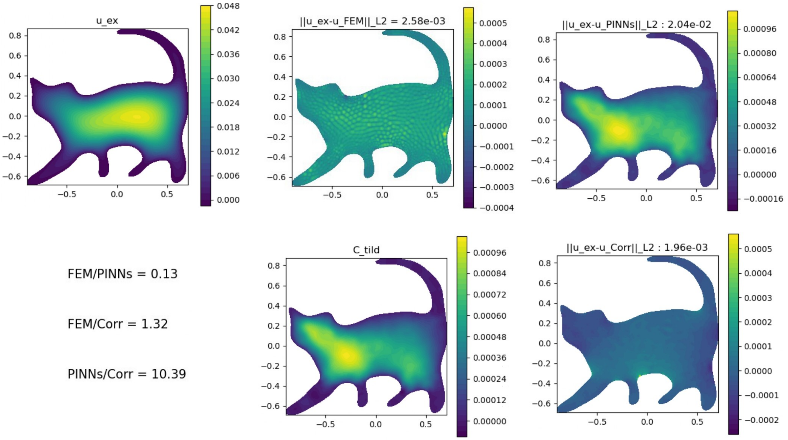




Other results

Poisson on Bean
Additive approach on Cat
Multiplicative approach
Degree of PINNs evaluation

Appendix 4 : Add on Cat



FEM/PINNs = 0.13

FEM/Corr = 1.32

PINNs/Corr = 10.39

MIMESIS

Other results

Poisson on Bean
Additive approach on Cat
Multiplicative approach
Degree of PINNs evaluation

Appendix 5 : Multiplicative approach

Correct by multiplying : Considering u_{NN} as the prediction of our PINNs for (1), we define

$$u_M = u_{NN} + M$$

with M a constant chosen so that $u_M > 0$, called the enhancement constant.
Thus, the correction problem consists in writing the solution as

$$\tilde{u} = u_M \times \underbrace{\boxed{\tilde{C}}}_{\approx 1}$$

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

$$\begin{cases} L(u_M\tilde{C}) = f, & \text{in } \Omega, \\ \tilde{C} = 1, & \text{on } \Gamma. \end{cases}$$