Enriching continuous Lagrange finite element approximation spaces using neural networks

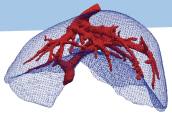
Michel Duprez[1], Emmanuel Franck[2], **Frédérique Lecourtier**[1] and Vanessa Lleras[3]

[1]Project-Team MIMESIS, Inria, Strasbourg, France
[2]Project-Team MACARON, Inria, Strasbourg, France
[3]IMAG, University of Montpellier, Montpellier, France

July 15, 2025

ICOSAHOM
MONTRÉAL 2025

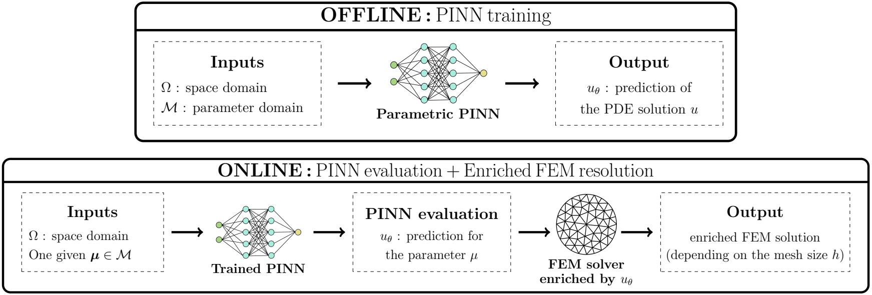
Scientific context



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

Objective : Develop an hybrid $\boxed{\text{finite element}}$ / $\boxed{\text{neural network}}$ method.

accurate — quick + parameterized



Problem considered

Stationary incompressible Navier-Stokes equations (with buoyancy and gravity) :
We consider $\Omega = [-1,1]^2$ a squared domain and $\boldsymbol{e}_y = (0,1)$.
Find the velocity $\boldsymbol{u} = (u,v)$, the pressure p and the temperature T such that

$$\begin{cases} (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p - \mu\Delta\boldsymbol{u} - g(\beta T + 1)\boldsymbol{e}_y = 0 \ \text{ in } \Omega & \text{(momentum)} \\ \nabla\cdot\boldsymbol{u} = 0 \ \text{ in } \Omega & \text{(incompressibility)} \\ \boldsymbol{u}\cdot\nabla T - k_f\Delta T = 0 \ \text{ in } \Omega & \text{(energy)} \\ + \text{ suitable BC} & \end{cases} \qquad (\mathcal{P})$$

with $g = 9.81$ the gravity, $\beta = 0.1$ the expansion coefficient, μ the viscosity and k_f the thermal conductivity. [Coulaud et al., 2024]

Problem considered

Objective: Simulate the flow for a range of $\boldsymbol{\mu} = (\mu, k_f) \in \mathcal{M} = [0.01, 0.1]^2$.

Stationary incompressible Navier-Stokes equations (with buoyancy and gravity) :
We consider $\boldsymbol{x} = (x, y) \in \Omega$ and $\boldsymbol{e}_y = (0, 1)$.
Find $\boldsymbol{U} = (\boldsymbol{u}, p, T) = (u, v, p, T)$ such that

$$\begin{cases} R_{mom}(U; \boldsymbol{x}, \boldsymbol{\mu}) = 0 \ \text{ in } \Omega & \text{(momentum)} \\ R_{inc}(U; \boldsymbol{x}, \boldsymbol{\mu}) = 0 \ \text{ in } \Omega & \text{(incompressibility)} \\ R_{ener}(U; \boldsymbol{x}, \boldsymbol{\mu}) = 0 \ \text{ in } \Omega & \text{(energy)} \\ + \text{ suitable BC} \end{cases} \qquad (\mathcal{P})$$

with $g = 9.81$ the gravity, $\beta = 0.1$ the expansion coefficient, μ the viscosity and k_f the thermal conductivity. [Coulaud et al., 2024]

Problem considered

Objective: Simulate the flow for a range of $\boldsymbol{\mu} = (\mu, k_f) \in \mathcal{M} = [0.01, 0.1]^2$.

Stationary incompressible Navier-Stokes equations (with buoyancy and gravity) :
We consider $\boldsymbol{x} = (x, y) \in \Omega$ and $\boldsymbol{e}_y = (0, 1)$.
Find $\boldsymbol{U} = (\boldsymbol{u}, p, T) = (u, v, p, T)$ such that

$$\begin{cases} R_{mom}(U; \boldsymbol{x}, \boldsymbol{\mu}) = 0 \quad \text{in } \Omega & \text{(momentum)} \\ R_{inc}(U; \boldsymbol{x}, \boldsymbol{\mu}) = 0 \quad \text{in } \Omega & \text{(incompressibility)} \\ R_{ener}(U; \boldsymbol{x}, \boldsymbol{\mu}) = 0 \quad \text{in } \Omega & \text{(energy)} \end{cases} \qquad (\mathcal{P})$$

with $g = 9.81$ the gravity, $\beta = 0.1$ the expansion coefficient, μ the viscosity and k_f the thermal conductivity. [Coulaud et al., 2024]

Boundary Conditions:

- $\boldsymbol{u} = 0$ on $\partial\Omega$
- $T = 1$ on the left wall ($x = -1$) and $T = -1$ on the right wall ($x = 1$)
  $\dfrac{\partial T}{\partial n} = 0$ on the top and bottom walls ($y = \pm 1$)

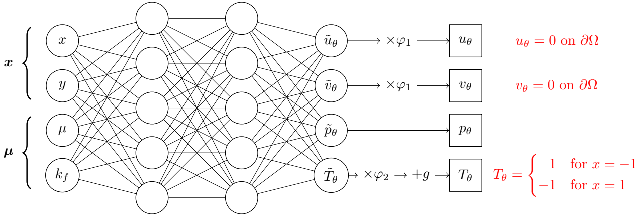
Physics-informed neural network (PINN)

Neural Network considered

We consider a parametric NN with 4 inputs and 4 outputs, defined by

$$U_\theta(\boldsymbol{x}, \boldsymbol{\mu}) = \big(u_\theta, v_\theta, p_\theta, T_\theta\big)(\boldsymbol{x}, \boldsymbol{\mu}).$$

The Dirichlet boundary conditions are imposed on the outputs of the MLP by a **post-processing** step. [Sukumar and Srivastava, 2022]



We consider two levelsets functions φ_1 and φ_2, and the linear function g defined by

$$\varphi_1(x,y) = (x-1)(x+1)(y-1)(y+1),$$

$$\varphi_2(x,y) = (x-1)(x+1) \quad \text{and} \quad g(x,y) = 1-(x+1).$$

PINN training

Approximate the solution of ($\mathcal{P}$) by a PINN : Find the optimal weights $\theta^\star$, such that

$$\theta^\star = \underset{\theta}{\operatorname{argmin}} \left(J_{inc}(\theta) + J_{mom}(\theta) + J_{ener}(\theta) + J_{ad}(\theta) \right), \qquad (\mathcal{P}_\theta)$$

where the different cost functions[1] are defined by

adiabatic condition

$$\boxed{J_{ad}(\theta) = \int_{\mathcal{M}} \int_{\partial\Omega|_{y=\pm 1}} \left| \frac{\partial T_\theta(\boldsymbol{x}, \boldsymbol{\mu})}{\partial n} \right|^2 d\boldsymbol{x} d\boldsymbol{\mu},}$$

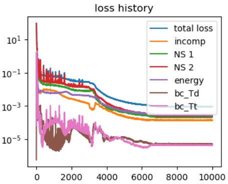
3 residual losses

$$\boxed{J_\bullet(\theta) = \int_{\mathcal{M}} \int_{\Omega} \left| R_\bullet(U_\theta(\boldsymbol{x}, \boldsymbol{\mu}); \boldsymbol{x}, \boldsymbol{\mu}) \right|^2 d\boldsymbol{x} d\boldsymbol{\mu},}$$

with U_θ the parametric NN and $\bullet$ the PDE considered (i.e. *inc*, *mom* or *ener*).

Network - MLP	
layers	$40, 60, 60, 60, 40$
σ	sine

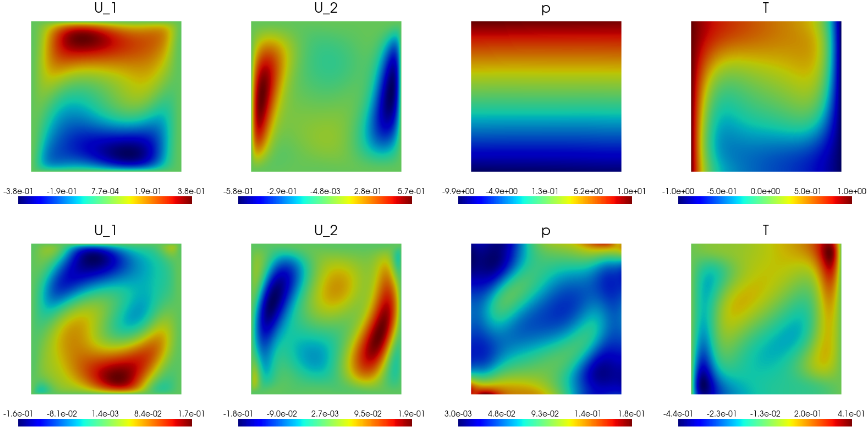
Training (ADAM / LBFGs)			
lr	7e-3	N_{col}	40000
n_{epochs}	10000	N_{bc}	30000



[1] Discretized by a random process using Monte-Carlo method.

PINN solution

$$\boldsymbol{\mu}^{(1)} = (0.1, 0.1),\ \boldsymbol{\mu}^{(2)} = (0.05, 0.05) \text{ and } \boxed{\boldsymbol{\mu}^{(3)} = (0.01, 0.01)}$$



TODO : renommer figure $u_\theta \ldots$ (solutions et erreurs)

Finite element method (FEM)

Discrete weak formulation I

Find $U_h = (\boldsymbol{u}_h, p_h, T_h) \in [V_h^0]^2 \times Q_h \times W_h$ s.t., $\forall (\boldsymbol{v}_h, q_h, w_h) \in [V_h^0]^2 \times Q_h \times W_h^0$,

$$(\mathcal{P}_h)\quad \begin{cases} \displaystyle\int_\Omega (\boldsymbol{u}_h \cdot \nabla)\boldsymbol{u}_h \cdot \boldsymbol{v}_h \, dx + \mu \int_\Omega \nabla \boldsymbol{u}_h : \nabla \boldsymbol{v}_h \, dx \\ \displaystyle\qquad - \int_\Omega p_h \, \nabla \cdot \boldsymbol{v}_h \, dx - g \int_\Omega (1 + \beta T_h) \boldsymbol{e}_y \cdot \boldsymbol{v}_h \, dx = 0 \qquad \text{(momentum)} \\ \displaystyle\int_\Omega q_h \, \nabla \cdot \boldsymbol{u}_h \, dx + 10^{-4} \int_\Omega q_h \, p_h \, dx = 0 \qquad \text{(incompressibility + pressure penalization)} \\ \displaystyle\int_\Omega (\boldsymbol{u}_h \cdot \nabla T_h) \, w_h \, dx + \int_\Omega k_f \nabla T_h \cdot \nabla w_h \, dx = 0 \qquad \text{(energy)} \end{cases}$$

with

$$\left.\begin{array}{ccccccc} \boldsymbol{u}_h & \in & [V_h^0]^2 & \subset & [H_0^1(\Omega)]^2 & : & \mathbb{P}_2 \\ p_h & \in & Q_h & \subset & L_0^2(\Omega) & : & \mathbb{P}_1 \end{array}\right\} \text{(Taylor–Hood spaces)}$$
$$\begin{array}{ccccccc} T_h & \in & W_h & \subset & W & : & \mathbb{P}_1 \end{array}$$

and

$$W = \{ w \in H^1(\Omega), \; w|_{x=-1} = 1, \; w|_{x=1} = -1 \}.$$

Discrete weak formulation II

Considering $(\phi_i)_{i=1}^{N_u}$, $(\psi_j)_{j=1}^{N_p}$ and $(\eta_k)_{k=1}^{N_T}$ the basis functions of the finite element spaces V_h^0, Q_h and W_h respectively, we can write the discrete solutions as:

$$\boldsymbol{u}_h(\boldsymbol{x}) = \sum_{i=1}^{N_u} \begin{pmatrix} u_i \\ v_i \end{pmatrix} \phi_i(\boldsymbol{x}), \quad p_h(\boldsymbol{x}) = \sum_{j=1}^{N_p} p_j \psi_j(\boldsymbol{x}) \quad \text{and} \quad T_h(\boldsymbol{x}) = \sum_{k=1}^{N_T} T_k \eta_k(\boldsymbol{x}),$$

with the unknown vectors for velocity, pressure and temperature defined by

$$\vec{u} = \left(u_i\right)_{i=1}^{N_u} \in \mathbb{R}^{N_u}, \quad \vec{v} = \left(v_i\right)_{i=1}^{N_u} \in \mathbb{R}^{N_u},$$

$$\vec{p} = \left(p_j\right)_{j=1}^{N_p} \in \mathbb{R}^{N_p} \quad \text{and} \quad \vec{T} = \left(T_k\right)_{k=1}^{N_T} \in \mathbb{R}^{N_T}.$$

Considering $N_h = 2N_u + N_p + N_T$, we can define the global vector of unknowns as:

$$\vec{U} = \left(\vec{u}, \vec{v}, \vec{p}, \vec{T}\right) \in \mathbb{R}^{N_h}.$$

and $F : \mathbb{R}^{N_h} \to \mathbb{R}^{N_h}$ the nonlinear operator associated to the weak formulation $(\mathcal{P}_h)$.

Newton method

We want to solve the non linear system:

$$F(\vec{U}) = 0$$

with $F : \mathbb{R}^{N_h} \to \mathbb{R}^{N_h}$ a non linear operator and $\vec{U} \in \mathbb{R}^{N_h}$ the unknown vector.

Algorithm 1: Newton's algorithm [Aghili et al., 2025]

Initialization step: set $\vec{U}^{(0)} = \vec{U}_0$;

for $k \geq 0$ **do**

  Solve the linear system $F(\vec{U}^{(k)}) + F'(\vec{U}^{(k)})\delta^{(k+1)} = 0$ for $\delta^{(k+1)}$;

  Update $\vec{U}^{(k+1)} = \vec{U}^{(k)} + \delta^{(k+1)}$;

end

Newton method

We want to solve the non linear system:

$$F(\vec{U}) = 0$$

with $F : \mathbb{R}^{N_h} \to \mathbb{R}^{N_h}$ a non linear operator and $\vec{U} \in \mathbb{R}^{N_h}$ the unknown vector.

Algorithm 1: Newton's algorithm [Aghili et al., 2025]

Initialization step: set $\vec{U}^{(0)} = \vec{U}_0$;
for $k \geq 0$ **do**
  Solve the linear system $F(\vec{U}^{(k)}) + F'(\vec{U}^{(k)})\delta^{(k+1)} = 0$ for $\delta^{(k+1)}$;
  Update $\vec{U}^{(k+1)} = \vec{U}^{(k)} + \delta^{(k+1)}$;
end

Natural initialization :

$$\vec{U}_0 = \left(\mathbf{0}_{N_u}, \mathbf{0}_{N_u}, \mathbf{0}_{N_p}, \vec{T}_0\right)$$

where for $i = 1, \dots, N_T$,

$$(\vec{T}_0)_i = g(\boldsymbol{x}^{(i)}) = 1 - (x^{(i)} + 1)$$

with $\boldsymbol{x}^{(i)} = \left(x^{(i)}, y^{(i)}\right)$ the i-th dofs coordinates.

Initialization

Enriched finite element method using PINN

Newton method - Additive approach

TODO

Numerical results

Numerical results

TODO

Conclusion

TODO

References

J. Aghili, E. Franck, R. Hild, V. Michel-Dansac, and V. Vigon. Accelerating the convergence of newton's method for nonlinear elliptic pdes using fourier neural operators. 2025.

Guillaume Coulaud, Maxime Le, and Régis Duvigneau. Investigations on Physics-Informed Neural Networks for Aerodynamics, 2024.

N. Sukumar and A. Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. 2022.