July 15, 2025

Enriching continuous Lagrange finite element approximation spaces using neural networks

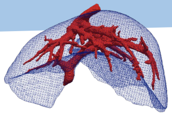
Michel Duprez[1], Emmanuel Franck[2], **Frédérique Lecourtier**[1] and Vanessa Lleras[3]

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

Joint work with:
H. Barucq, F. Faucher, N. Victorion and V. Michel-Dansac.

ICOSAHOM
MONTRÉAL 2025

Scientific context



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

Objective : Develop an hybrid finite element / neural network method.

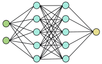
finite element: accurate

neural network: quick + parameterized

OFFLINE : PINN training

Inputs
Ω : space domain
$\mathcal{M}$: parameter domain

→



Parametric PINN

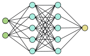
→

Output
u_θ : prediction of the PDE solution u

ONLINE : PINN evaluation + Enriched FEM resolution

Inputs
Ω : space domain
One given $\boldsymbol{\mu} \in \mathcal{M}$

→

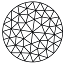


Trained PINN

→

PINN evaluation
u_θ : prediction for the parameter μ

→



FEM solver enriched by u_θ

→

Output
enriched FEM solution (depending on the mesh size h)

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 $\boldsymbol{u} = (u_1, u_2)$, 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: Simulation on a range of parameters $\boldsymbol{\mu} = (\mu, k_f) \in \mathcal{M} = [0.01, 0.1]^2$.

Stationary incompressible Navier-Stokes equations (with buoyancy and gravity) :
We consider $\boldsymbol{x} = (x, y) \in \Omega$ and $\boldsymbol{e}_y = (0, 1)$.
Find $\boldsymbol{U} = (\boldsymbol{u}, p, T) = (u_1, u_2, p, T)$ 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: Simulation on a range of parameters $\boldsymbol{\mu} = (\mu, k_f) \in \mathcal{M} = [0.01, 0.1]^2$.

Stationary incompressible Navier-Stokes equations (with buoyancy and gravity) :
We consider $\boldsymbol{x} = (x, y) \in \Omega$ and $\boldsymbol{e}_y = (0, 1)$.
Find $\boldsymbol{U} = (\boldsymbol{u}, p, T) = (u_1, u_2, p, T)$ such that

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

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 ($y = \pm 1$, denoted by Γ_{ad})

Evaluate quality of solutions

In the following, we are interested in three parameters (rising in complexity) :

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

We evaluate the quality of solutions by comparing them to a reference solution.[1]

[1] Computed on a over-refined mesh ($h = 7.10^{-3}$) on a $\mathbb{P}_3^2 \times \mathbb{P}_2 \times \mathbb{P}_3$ continuous Lagrange FE space.

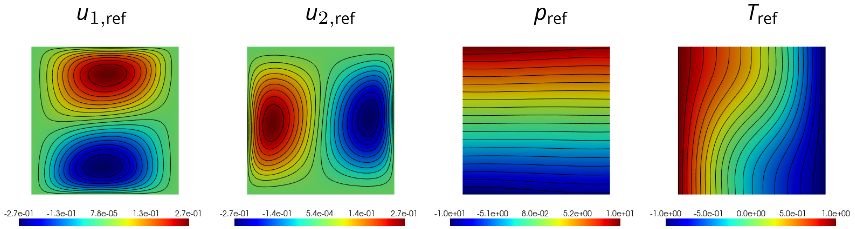
Evaluate quality of solutions

In the following, we are interested in three parameters (rising in complexity) :

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

We evaluate the quality of solutions by comparing them to a reference solution.[1]

Reference solution - Rayleigh number : $Ra = 1\,569.6$



[1] Computed on a over-refined mesh ($h = 7.10^{-3}$) on a $\mathbb{P}_3^2 \times \mathbb{P}_2 \times \mathbb{P}_3$ continuous Lagrange FE space.

Evaluate quality of solutions

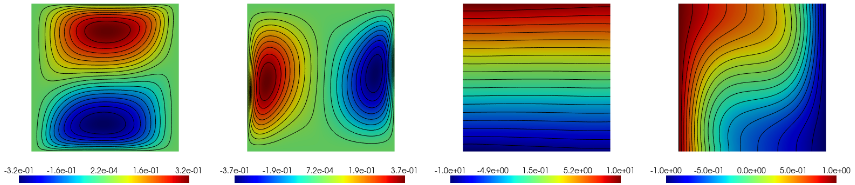
In the following, we are interested in three parameters (rising in complexity) :

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

We evaluate the quality of solutions by comparing them to a reference solution.[1]

Reference solution - Rayleigh number : $Ra = 6\,278.4$

$u_{1,\text{ref}}$ $u_{2,\text{ref}}$ p_{ref} T_{ref}



[1] Computed on a over-refined mesh ($h = 7.10^{-3}$) on a $\mathbb{P}_3^2 \times \mathbb{P}_2 \times \mathbb{P}_3$ continuous Lagrange FE space.

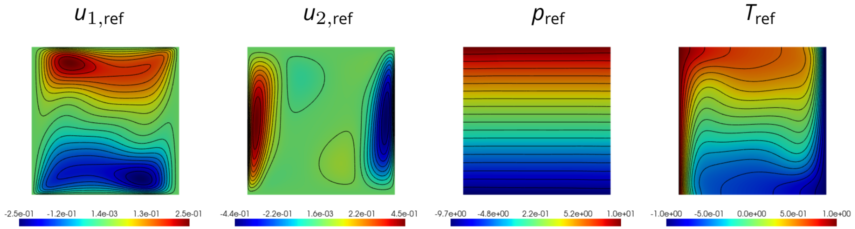
Evaluate quality of solutions

In the following, we are interested in three parameters (rising in complexity) :

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

We evaluate the quality of solutions by comparing them to a reference solution.[1]

Reference solution - Rayleigh number : $Ra = 156\,960$



[1] Computed on a over-refined mesh ($h = 7.10^{-3}$) on a $\mathbb{P}_3^2 \times \mathbb{P}_2 \times \mathbb{P}_3$ continuous Lagrange FE space.

Physics-informed neural network (PINN)

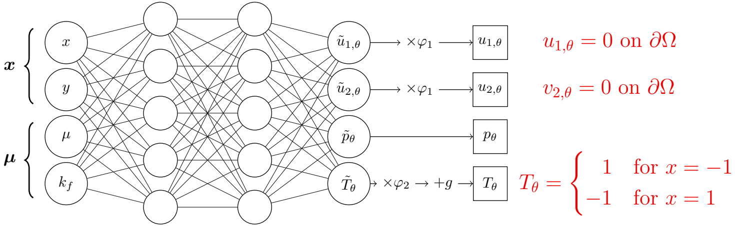
The PINN is parametrized by the $\boldsymbol{\mu}$ parameter.

Neural Network considered

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

$$U_\theta(\boldsymbol{x}, \boldsymbol{\mu}) = \big(u_{1,\theta}, u_{2,\theta}, p_\theta, T_\theta\big)(\boldsymbol{x}, \boldsymbol{\mu}).$$

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

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

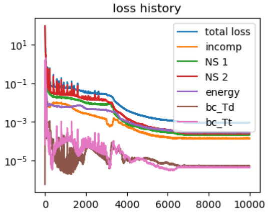
3 residual losses

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

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

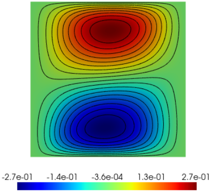


[1] Discretized by a random process using Monte-Carlo method.

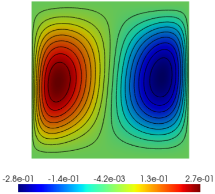
Prediction on $\mu^{(1)} = (0.1, 0.1)$

Prediction :

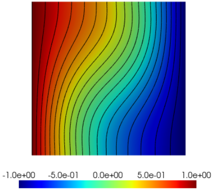
$u_{1,\theta}$



$u_{2,\theta}$

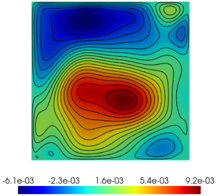


T_θ

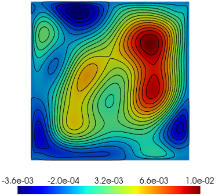


Error map :

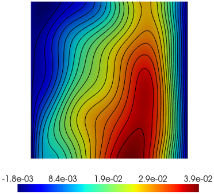
$u_{1,\text{ref}} - u_{1,\theta}$



$u_{2,\text{ref}} - u_{2,\theta}$



$T_{\text{ref}} - T_\theta$



L^2 error : (relative)

2.98×10^{-2} 3.17×10^{-2} 3.90×10^{-2}

Finite element method (FEM)

The μ parameter is fixed in the FE resolution.

Discrete weak formulation

We consider a mixed finite element space $\boxed{M_h = [V_h^0]^2 \times Q_h \times W_h}$ and

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

$$\begin{array}{lllllll} T_h & \in & W_h & \subset & W & : & \mathbb{P}_2 \end{array}$$

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

Discrete weak formulation

We consider a mixed finite element space $\boxed{M_h = [V_h^0]^2 \times Q_h \times W_h}$ and

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

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

Weak problem : Find $U_h = (\boldsymbol{u}_h, p_h, T_h) \in M_h$ s.t., $\forall(\boldsymbol{v}_h, q_h, w_h) \in M_h^0$,

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

where $M_h^0 = [V_h^0]^2 \times Q_h \times W_h^0$ with $W_h^0 \subset \{w \in H^1[\Omega],\ w|_{x=\pm 1} = 0\}$.

Newton method

We consider the following three parameters:

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

Denoting N_h the dimension of M_h, we want to solve the non linear system:

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

with $F : \mathbb{R}^{N_h} \to \mathbb{R}^{N_h}$ a non linear operator and $\vec{U}_k \in \mathbb{R}^{N_h}$ the unknown vector associated to the k-th parameter $\boldsymbol{\mu}^{(k)}$ ($k = 1, 2, 3$). Appendix 1

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

Initialization step: set $\vec{U}_k^{(0)} = \vec{U}_{k,0}$;

for $n \geq 0$ **do**

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

Update $\vec{U}_k^{(n+1)} = \vec{U}_k^{(n)} + \delta_k^{(n+1)}$;

end

Newton method

We consider the following three parameters:

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

Denoting N_h the dimension of M_h, we want to solve the non linear system:

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

with $F : \mathbb{R}^{N_h} \to \mathbb{R}^{N_h}$ a non linear operator and $\vec{U}_k \in \mathbb{R}^{N_h}$ the unknown vector associated to the k-th parameter $\boldsymbol{\mu}^{(k)}$ ($k = 1, 2, 3$). Appendix 1

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

Initialization step: set $\vec{U}_k^{(0)} = \vec{U}_{k,0}$;

for $n \geq 0$ **do**

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

Update $\vec{U}_k^{(n+1)} = \vec{U}_k^{(n)} + \delta_k^{(n+1)}$;

end

How to initialize the Newton solver?

3 types of initialization

- **Natural initialization :**
- **DeepPhysics initialization :**
- **Incremental initialization.**

3 types of initialization

- **Natural initialization :** Using constant or linear function.
  Considering a fixed parameter with $k \in \{1, 2, 3\}$, we can use the following initialization:
  $$\vec{U}_{k,0} = (\vec{0}, \vec{0}, \vec{0}, \vec{T}_0)$$
  where for $i = 1, \dots, \dim(W_h)$,
  $$(\vec{T}_0)_i = g(\boldsymbol{x}^{(i)}) = 1 - (x^{(i)} + 1)$$
  with $\boldsymbol{x}^{(i)} = \left(x^{(i)}, y^{(i)}\right)$ the i-th dofs coordinates of W_h.
- **DeepPhysics initialization :**
- **Incremental initialization.**

3 types of initialization

- **Natural initialization :** Using constant or linear function.
- **DeepPhysics initialization :** Using PINN prediction [Odot et al., 2021].
  Considering a fixed parameter with $k \in \{1,2,3\}$, we can use the following initialization for $i = 1, \dots, N_h$,

  $$\left(\vec{U}_{k,0}\right)_i = U_\theta(\boldsymbol{x}^{(i)}, \boldsymbol{\mu}^{(k)})$$

  with $\boldsymbol{x}^{(i)} = \left(x^{(i)}, y^{(i)}\right)$ the i-th dofs coordinates of M_h and U_θ the PINN.
- **Incremental initialization.**

3 types of initialization

- **Natural initialization :** Using constant or linear function.
- **DeepPhysics initialization :** Using PINN prediction [Odot et al., 2021].
- **Incremental initialization.** Using a coarse FE solution of a simpler parameter.
  - We consider a fixed parameter with $k \in \{2,3\}$.
  - We consider a coarse grid (16×16 grid) and compute the FE solution of ($\mathcal{P}_h$) for the parameter $\boldsymbol{\mu}^{(k-1)}$.
  - We interpolate the coarse solution to the current mesh.
  - We use it as an initialization for the Newton method, i.e.

$$\vec{U}_{k,0} = \left(\vec{u}_{k-1}, \vec{v}_{k-1}, \vec{p}_{k-1}, \vec{T}_{k-1}\right)$$

where $\vec{u}_{k-1}$, $\vec{v}_{k-1}$, $\vec{p}_{k-1}$ and $\vec{T}_{k-1}$ are the FE solutions for the parameter $\boldsymbol{\mu}^{(k-1)}$.

Enriched finite element method using PINN

Enriched space using PINN

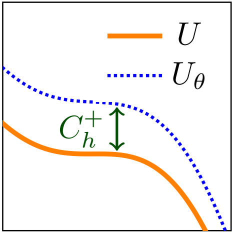
Considering the PINN prior $U_\theta = (\boldsymbol{u}_\theta, p_\theta, T_\theta)$, we define the mixed finite element space additively enriched by the PINN as follows:

$$\boxed{M_h^+ = \left\{ U_h^+ = U_\theta + C_h^+, \quad C_h^+ \in M_h^0 \right\}}$$

with $M_h^0 = [V_h^0]^2 \times Q_h \times W_h^0$, $U_h^+ = (\boldsymbol{u}_h^+, p_h^+, T_h^+) \in M_h^+$ and $C_h^+ = (\boldsymbol{C}_{h,\boldsymbol{u}}^+, C_{h,p}^+, C_{h,T}^+)$.

We can then define the three finite element subspaces of M_h^+ as follows:

$$\begin{aligned}
\boldsymbol{V}_h^+ &= \left\{ \boldsymbol{u}_h^+ = \boldsymbol{u}_\theta + \boldsymbol{C}_{h,\boldsymbol{u}}^+, \ \boldsymbol{C}_{h,\boldsymbol{u}}^+ \in [V_h^0]^2 \right\}, \\
Q_h^+ &= \left\{ p_h^+ = p_\theta + C_{h,p}^+, \ C_{h,p}^+ \in Q_h \right\}, \\
W_h^+ &= \left\{ T_h^+ = T_\theta + C_{h,T}^+, \ C_{h,T}^+ \in W_h^0 \right\},
\end{aligned}$$



where $\boldsymbol{C}_{h,\boldsymbol{u}}^+$, $C_{h,p}^+$ and $C_{h,T}^+$ becomes the unknowns.

à ajouter : dans quoi vit U_θ ?

Weak formulation - Additive approach

Weak problem : Find $C_h^+ = (C_{h,\boldsymbol{u}}^+, C_{h,p}^+, C_{h,T}^+) \in M_h^0$ s.t., $\forall(\boldsymbol{v}_h, q_h, w_h) \in M_h^0$,

$$
\begin{aligned}
&\int_\Omega \left[(\boldsymbol{u}_\theta \cdot \nabla)\boldsymbol{u}_\theta + (\boldsymbol{u}_\theta \cdot \nabla)C_{h,\boldsymbol{u}}^+ + (C_{h,\boldsymbol{u}}^+ \cdot \nabla)\boldsymbol{u}_\theta + (C_{h,\boldsymbol{u}}^+ \cdot \nabla)C_{h,\boldsymbol{u}}^+\right] \cdot \boldsymbol{v}_h \, d\boldsymbol{x} \\
&\quad + \mu\left(\int_\Omega \nabla \boldsymbol{u}_\theta : \nabla \boldsymbol{v}_h \, d\boldsymbol{x} + \int_\Omega \nabla C_{h,\boldsymbol{u}}^+ : \nabla \boldsymbol{v}_h \, d\boldsymbol{x}\right) + \left(\int_\Omega \nabla p_\theta \cdot \boldsymbol{v}_h \, d\boldsymbol{x} - \int_\Omega C_{h,p}^+ \nabla \cdot \boldsymbol{v}_h \, d\boldsymbol{x}\right) \\
&\qquad - g\int_\Omega (1 + \beta(T_\theta + C_{h,T}^+))\boldsymbol{e}_y \cdot \boldsymbol{v}_h \, d\boldsymbol{x} = 0, \ \text{(momentum)} \\
&\int_\Omega q_h \left[\nabla \cdot \boldsymbol{u}_\theta + \nabla \cdot C_{h,\boldsymbol{u}}^+\right] d\boldsymbol{x} + 10^{-4}\int_\Omega q_h\,(p_\theta + C_{h,p}^+)\, d\boldsymbol{x} = 0, \ \text{(incompressibility + penal)} \\
&\int_\Omega \left[\boldsymbol{u}_\theta \cdot \nabla T_\theta + \boldsymbol{u}_\theta \cdot \nabla C_{h,T}^+ + C_{h,\boldsymbol{u}}^+ \cdot \nabla T_\theta + C_{h,\boldsymbol{u}}^+ \cdot \nabla C_{h,T}^+\right] w_h \, d\boldsymbol{x} \\
&\quad + k_f\left(\int_\Omega \nabla T_\theta \cdot \nabla w_h \, d\boldsymbol{x} + \int_\Omega \nabla C_{h,T}^+ \cdot \nabla w_h \, d\boldsymbol{x}\, w_h \, d\boldsymbol{s}\right) = 0, \ \text{(energy)}
\end{aligned}
\qquad (\mathcal{P}_h^+)
$$

with $U_\theta = (\boldsymbol{u}_\theta, p_\theta, T_\theta)$ the PINN prior and some modified boundary conditions.

Newton method - Additive approach

We want to solve the non linear system:

$$F_\theta(\vec{C}) = 0$$

with $F_\theta : \mathbb{R}^{N_h} \to \mathbb{R}^{N_h}$ the non linear operator associated to the weak problem ($\mathcal{P}_h^+$) and $\vec{C} \in \mathbb{R}^{N_h}$ the correction vector (unknown).

Algorithm 2: Newton algorithm [Aghili et al., 2025]

Initialization step: set $\vec{C}^{(0)} = 0$;
for $n \geq 0$ **do**
  Solve the linear system $F_\theta(\vec{C}^{(n)}) + F'_\theta(\vec{C}^{(n)})\delta^{(n+1)} = 0$ for $\delta^{(n+1)}$;
  Update $\vec{C}^{(n+1)} = \vec{C}^{(n)} + \delta^{(n+1)}$;
end

Advantage compared to DeepPhysics[1]: Appendix 2

u_θ is not required to live in the same discrete space as C_h^+.

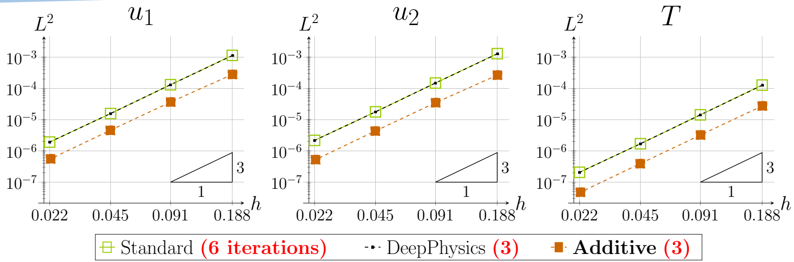
[1] The additive approach is exactly the same as DeepPhysics if we take U_θ in the same space as C_h^+.

Numerical results

- Results obtained with a laptop GPU.
- The newton solver is the same for all methods (rtol$= 10^{-10}$, atol$= 10^{-10}$, max_it$= 30$).
- Additive approach : we consider u_θ in a $\mathbb{P}_3^2 \times \mathbb{P}_2 \times \mathbb{P}_3$ continuous Lagrange FE space (defined on the current mesh).

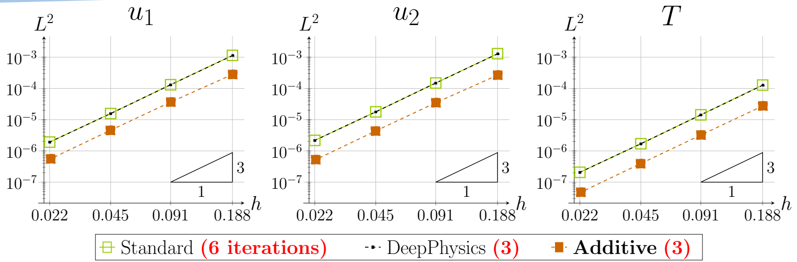
Error estimates I

$\boldsymbol{\mu}^{(1)}$:

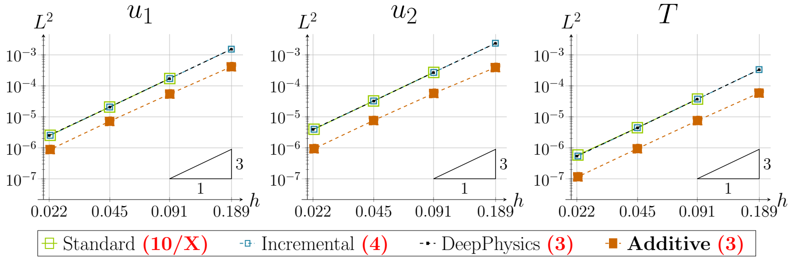


Error estimates I

$\boldsymbol{\mu}^{(1)}$:

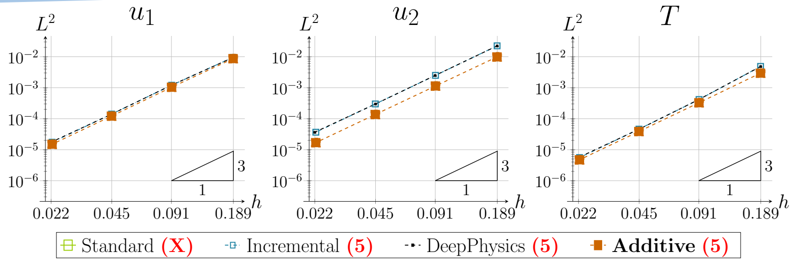


$\boldsymbol{\mu}^{(2)}$:



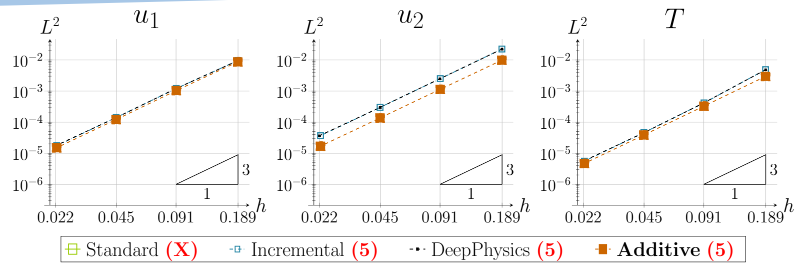
Error estimates II

$\boldsymbol{\mu}^{(3)}$:

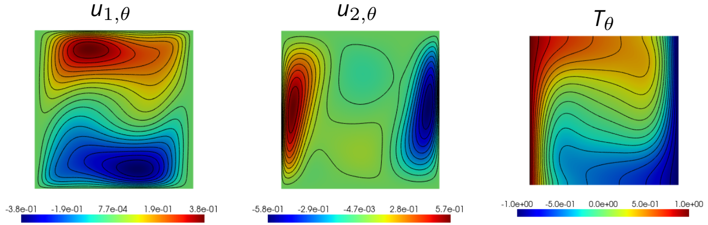


Error estimates II

$\boldsymbol{\mu}^{(3)}$:

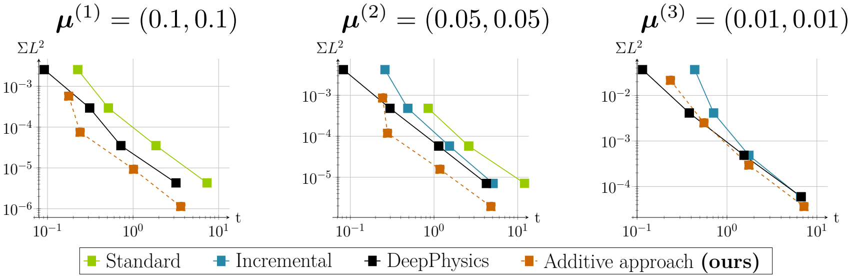


Prediction :



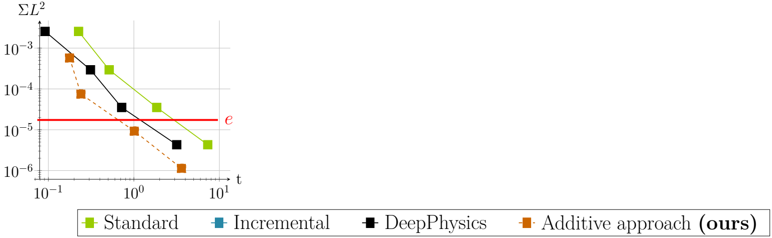
L^2 error : (relative) 5.75×10^{-1} 4.89×10^{-1} 2.57×10^{-1}

Numerical costs



Numerical costs

$\boldsymbol{\mu}^{(1)} = (0.1, 0.1)$



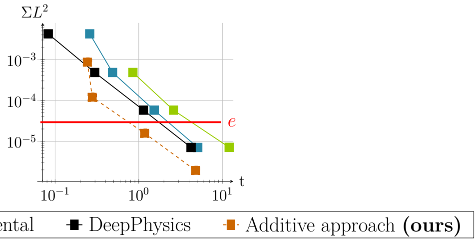
N_{dofs} and execution time required to reach the same global L^2 relative error[1] e :

	Number of DoFs		Execution times		
e	**Std/DPhy**	**Add**	**Std**	**DPhy**	**Add**
$1 \cdot 10^{-3}$	6,031	2,044	0.32	0.16	0.16
$1 \cdot 10^{-4}$	26,959	10,588	0.99	0.48	0.23
$1 \cdot 10^{-5}$	121,156	49,231	4.21	1.75	0.96

[1] Defined as the sum of the L^2 relatives errors on $\boldsymbol{u}$ and T.

Numerical costs

$\mu^{(2)} = (0.05, 0.05)$



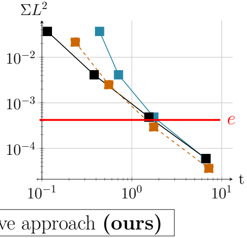
N_{dofs} and execution time required to reach the same global L^2 relative error[1] e :

	Number of DoFs		Execution times			
e	**Std/Inc/DPhy**	**Add**	**Std**	**Inc**	**DPhy**	**Add**
$1 \cdot 10^{-3}$	7,828	2,748	0.58	0.39	0.19	0.24
$1 \cdot 10^{-4}$	35,884	14,623	1.95	1.14	0.8	0.32
$1 \cdot 10^{-5}$	167,583	70,303	9.39	4.16	3.4	1.59

[1] Defined as the sum of the L^2 relatives errors on $\boldsymbol{u}$ and T.

Numerical costs

$\mu^{(3)} = (0.01, 0.01)$



N_{dofs} and execution time required to reach the same global L^2 relative error[1] e :

	Number of DoFs			Execution times			
e	Std	Inc/DPhy	Add	Std	Inc	DPhy	Add
$1 \cdot 10^{-3}$	X	33,204	23,524	X	1.29	0.96	0.91
$1 \cdot 10^{-4}$	X	150,339	108,931	X	4.76	4.67	3.65
$1 \cdot 10^{-5}$	X	690,924	502,156	X	20.34	23.3	17.23

[1] Defined as the sum of the L^2 relatives errors on $\boldsymbol{u}$ and T.

Conclusion

TODO

Parler du papier en linéaire et dire que dans ce cadre on a des résultats théoriques de convergence.

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.

A. Odot, R. Haferssas, and S. Cotin. Deepphysics: a physics aware deep learning framework for real-time simulation, 2021.

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

Appendix 1 : Finite element method (FEM)

A1 – Construction of the unknown vector

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

$$\vec{u} = (u_i)_{i=1}^{N_u} \in \mathbb{R}^{N_u}, \quad \vec{v} = (v_i)_{i=1}^{N_u} \in \mathbb{R}^{N_u},$$
$$\vec{p} = (p_j)_{j=1}^{N_p} \in \mathbb{R}^{N_p} \text{ and } \vec{T} = (T_k)_{k=1}^{N_T} \in \mathbb{R}^{N_T}.$$

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 $(\mathcal{P}_h)$.

Appendix 2 : DeepPhysics / Additive approach

A2 – ??