A new unfitted finite element method: ϕ-FEM

27/09/2023

Killian Vuillemot

Univ Montpellier (IMAG) & Inria (MIMESIS)

Context

Construction of digital twins, in real-time, for surgical interventions.

Tools

- Simulation of the deformations of organs : PDEs $\longrightarrow$ FEMs,
- Complex geometries $\longrightarrow$ Unfitted FEMs,
- Real-time constructions $\longrightarrow$ machine learning techniques.

New method : ϕ-FEM, unfitted method, precise, easy to implement.

- FEM : Finite Element Method

- FEM : Finite Element Method
- Idea : from continuous to discrete,
  - Strong formulation :

    Find $u \in H^2(\Omega)$ s.t : $-\Delta u = f$ in Ω, $u = 0$ on $\partial\Omega$.
  - Weak formulation :
    multiplication by a "test function" + integration by parts,

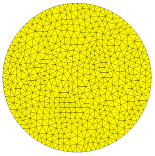
    Find $u \in H_0^1(\Omega)$ s.t. :

$$\int_\Omega \nabla u \cdot \nabla v - \underbrace{\int_\Gamma \frac{\partial u}{\partial n} v}_{=0} = \int_\Omega f v \quad \forall v \in H_0^1(\Omega).$$

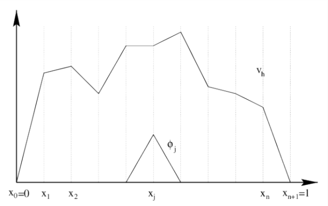
  - FEM formulation :

    Find $u_h \in V_h$ s.t. :

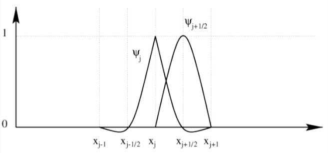
$$\int_\Omega \nabla u_h \cdot \nabla v_h = \int_\Omega f v_h \quad \forall v \in V_h.$$



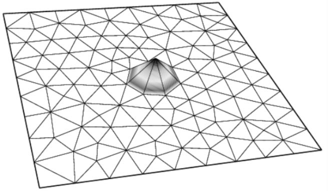
Finite element space : $V_h = \{$ cont. piecewise pol. functions on a regular mesh $\}$.



(a) $\mathbb{P}^1$ shape function in dimension 1.

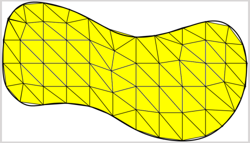


(b) $\mathbb{P}^2$ shape function in dimension 1.

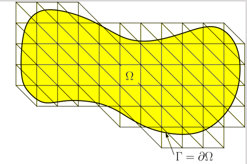


(c) $\mathbb{P}^1$ shape function in dimension 2.

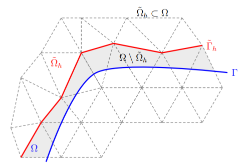
Previous works



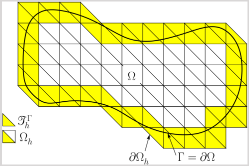
(a) Standard FEM (Clough 60s).



(b) XFEM (Moes and al., 2006), CutFEM (Burman, Hansbo, 2010-2014).



(c) Shifted Boundary method (Atallah and al., 2021).



(d) ϕ-FEM (Duprez and Lozinski, 2020).

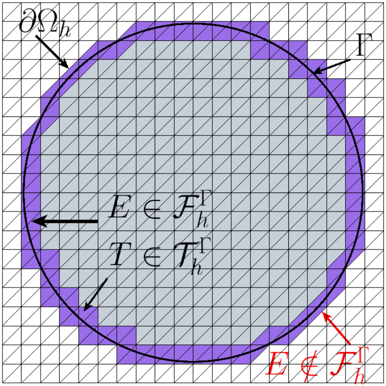
Problems on complex shapes $\longrightarrow$ unfitted FEMs

Level-set function

$$\Omega = \{\phi < 0\} \text{ et } \Gamma = \{\phi = 0\}\,.$$

The spaces

- $\mathcal{T}_h$: ϕ-FEM mesh,
- $\mathcal{T}_h^\Gamma$: cells of $\mathcal{T}_h$ cut by the boundary (purple triangles),
- $\mathcal{F}_h^\Gamma$: internal facets of $\mathcal{T}_h^\Gamma$.



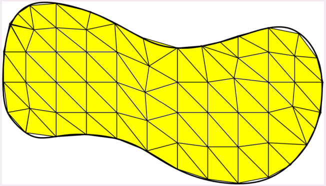
Example with $\phi(x, y) = -1 + x^2 + y^2$.

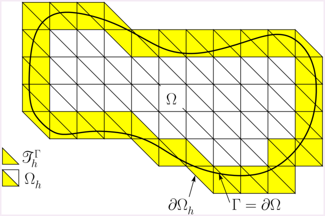
Example (Poisson-Dirichlet equation)

$$-\Delta u = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \Gamma. \tag{1}$$

Find u s.t. $\begin{cases} -\Delta u & = f, \text{ in } \Omega, \\ u & = 0, \text{ on } \Gamma. \end{cases}$

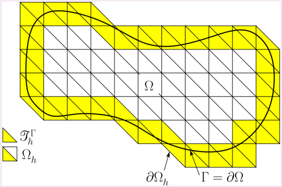
Find w s.t. $\begin{cases} -\Delta(\phi w) & = f, \text{ in } \Omega_h, \\ \text{with } u & = \phi w. \end{cases}$





Example (Poisson-Dirichlet equation)

$$-\Delta u = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \Gamma\,. \quad (1)$$



- Extend (1) to Ω_h with no b.c. : $-\Delta u = f$ in Ω_h,
- Impose b.c. by using the level-set and additional variables : $u = \phi w$,
- Go to discrete spaces using Lagrange interpolations and finite elements : $\phi \to \phi_h, w \to w_h$ and $u \to u_h$,
- Find w_h such that for all v_h,

$$\int_{\Omega_h} \nabla(\phi_h w_h) \cdot \nabla(\phi_h v_h) - \int_{\partial\Omega_h} \frac{\partial}{\partial n}(\phi_h w_h)\phi_h v_h + \text{stabs} = \int_{\Omega_h} f\phi_h v_h - \text{stabs}\,.$$

Interests of the method

- Optimal convergence in L^2 and H^1 norms,
- Easy to implement : standard shape functions, no cut cells $\longrightarrow$ standard quadrature rules,
- Acceptable conditioning of the finite element matrix.

Interests of the method

- Optimal convergence in L^2 and H^1 norms,
- Easy to implement : standard shape functions, no cut cells $\longrightarrow$ standard quadrature rules,
- Acceptable conditioning of the finite element matrix.

Other schemes

- Mixed boundary conditions : Dirichlet and Neumann conditions,
- Linear elasticity problems,
- Hyperelastic materials,
- Stokes problem,
- Heat equation.

When will my pan be cold?

$$\begin{cases} \partial_t u - \Delta u &= f \text{ in } \Omega \times (0,T), \\ u &= 0 \text{ on } \Gamma \times (0,T), \\ u_{|t=0} &= u^0 \qquad\qquad \text{in } \Omega\,. \end{cases}$$

When will my pan be cold?

$$\begin{cases} \partial_t u - \Delta u &= f \text{ in } \Omega \times (0,T), \\ u &= 0 \text{ on } \Gamma \times (0,T), \\ u_{|t=0} &= u^0 \qquad\quad \text{in } \Omega\,. \end{cases}$$

First step : time discretization

Implicit Euler scheme : given $u^n = \phi w^n$, find $u^{n+1} = \phi w^{n+1}$ such that

$$\frac{\phi w^{n+1} - \phi w^n}{\Delta t} - \Delta(\phi w^{n+1}) = f^{n+1}\,.$$

When will my pan be cold?

$$\begin{cases} \partial_t u - \Delta u &= f \text{ in } \Omega \times (0,T), \\ u &= 0 \text{ on } \Gamma \times (0,T), \\ u_{|t=0} &= u^0 \qquad \text{ in } \Omega\,. \end{cases}$$

Time discretization

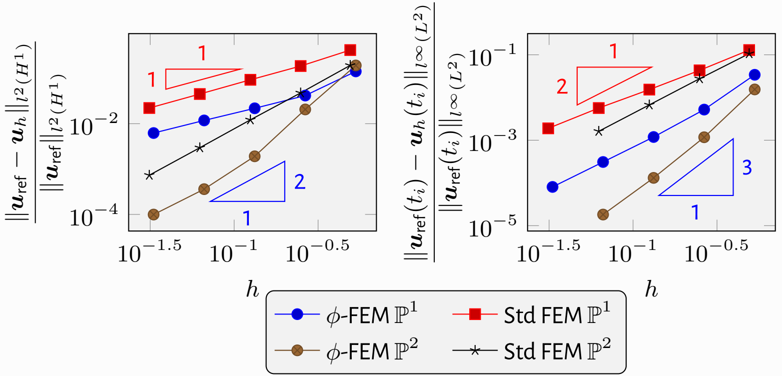
$$\frac{\phi w^{n+1} - \phi w^n}{\Delta t} - \Delta(\phi w^{n+1}) = f^{n+1}\,.$$

The proposed scheme

$$\int_{\Omega_h} \frac{\phi_h w_h^{n+1}}{\Delta t} \phi_h v_h + \int_{\Omega_h} \nabla(\phi_h w_h^{n+1}) \cdot \nabla(\phi_h v_h)$$
$$- \int_{\partial\Omega_h} \frac{\partial}{\partial n}(\phi_h w_h^{n+1})\phi_h v_h + \text{stabs} = \int_{\Omega_h} \left(\frac{u_h^n}{\Delta t} + f^{n+1} \right) \phi_h v_h - \text{stabs}\,.$$

Theorem (Duprez, Lleras, Lozinski, Vuillemot, 2023)

- $\mathcal{E}_{l^2(H^1)} \sim \mathcal{O}(h^k)$
- $\mathcal{E}_{l^\infty(L^2)} \sim \mathcal{O}(h^{k+\frac{1}{2}})$



In the context of real-time simulations, we need
quasi-instantaneous results.

- ϕ-FEM : precise but slow $\longrightarrow$ Not real-time ☹️
- How to obtain fast results $\longrightarrow$ Neural Networks 🚀
- ϕ-FEM $+$ Neural Networks $\longrightarrow$ precise and real-time method 🫡

A new problem :

How to combine ϕ-FEM and neural networks to obtain fast and precise results ?

$\longrightarrow$ the Fourier Neural Operator.

A new problem :

How to combine ϕ-FEM and neural networks to obtain fast and precise results ?

$\longrightarrow$ the Fourier Neural Operator.

Why the FNO ?

- uses FFT $\longrightarrow$ requires Cartesian grid, as ϕ-FEM,
- according to the authors : more accurate than other ML-methods,
- multi-resolution abilities,
- No need to change the underlying architecture when changing the governing PDE.

- Parametric application :

$$\mathcal{G}_\theta : \mathbb{R}^{ni \times nj \times 3} \xrightarrow{P} \mathbb{R}^{ni \times nj \times nk'} \xrightarrow{\mathcal{H}_\theta^1} \mathbb{R}^{ni \times nj \times nk'} \xrightarrow{\mathcal{H}_\theta^2} \dots \xrightarrow{\mathcal{H}_\theta^4} \mathbb{R}^{ni \times nj \times nk'} \xrightarrow{Q} \mathbb{R}^{ni \times nj \times 1},$$

with ni (resp nj) the number of pixels in the height (resp width) and nk' a hidden dimension.

- Parametric application :

$$\mathcal{G}_\theta : \mathbb{R}^{ni\times nj\times 3} \xrightarrow{P} \mathbb{R}^{ni\times nj\times nk'} \xrightarrow{\mathcal{H}_\theta^1} \mathbb{R}^{ni\times nj\times nk'} \xrightarrow{\mathcal{H}_\theta^2} \dots \xrightarrow{\mathcal{H}_\theta^4} \mathbb{R}^{ni\times nj\times nk'} \xrightarrow{Q} \mathbb{R}^{ni\times nj\times 1},$$

with ni (resp nj) the number of pixels in the height (resp width) and nk' a hidden dimension.

- Each layer is defined by :

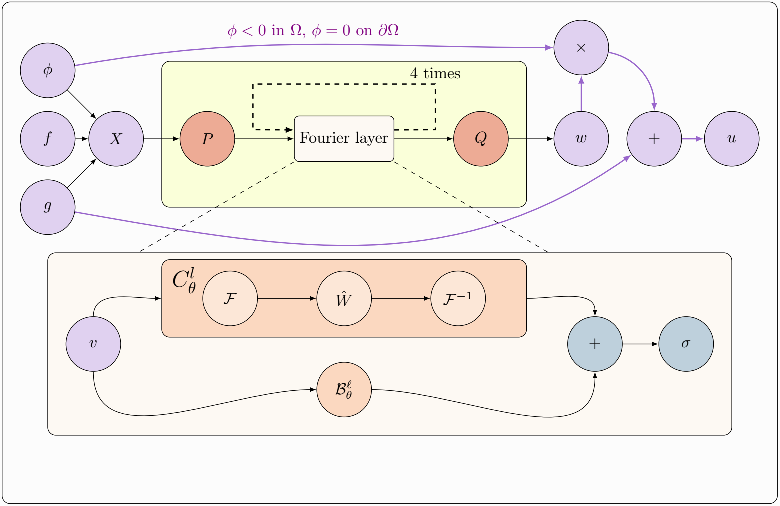
$$\mathcal{H}_\theta^\ell(X) = \sigma\left(\mathcal{C}_\theta^\ell(X) + \mathcal{B}_\theta^\ell(X)\right),$$

where σ is an activation function (here GELU), $\mathcal{C}_\theta^l$ is a convolution layer and

$$\mathcal{B}_\theta^\ell(X)_{ijk} = \sum_{k'} X_{ijk} W_{k'k} + B_k,$$

with $W_{k'k}$ and B_k the kernels and trainable biases which constitute θ.

ϕ-FEM + FNO : THE PIPELINE



First test case

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

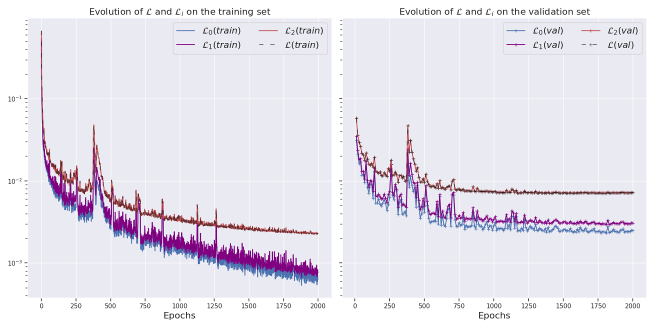
where Ω is a random rotated ellipse.

First test case

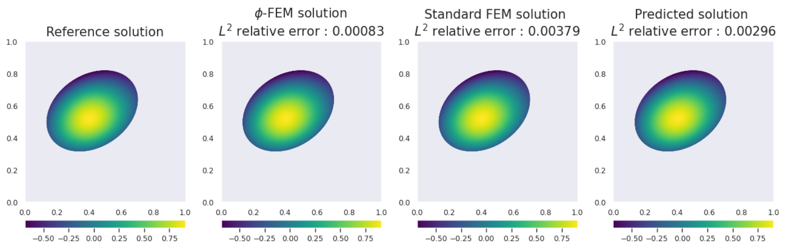
$$-\Delta u = f \text{ , in } \Omega, \; u = g \text{ , on } \Gamma \text{ ,}$$

where Ω is a random rotated ellipse.

Convergence of the loss function

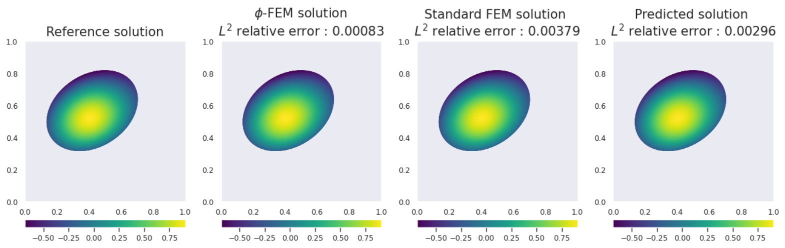


(ϕ-FEM + FNO) VS STANDARD FEM VS ϕ-FEM

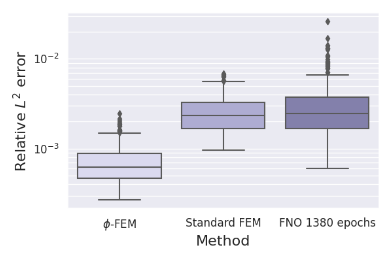


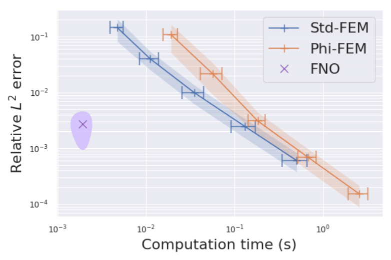
Outputs of the three methods.

(ϕ-FEM + FNO) VS STANDARD FEM VS ϕ-FEM



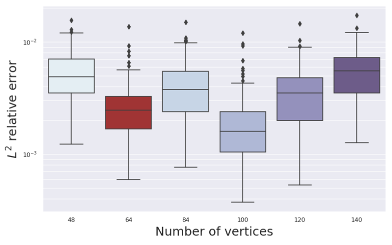
Outputs of the three methods.

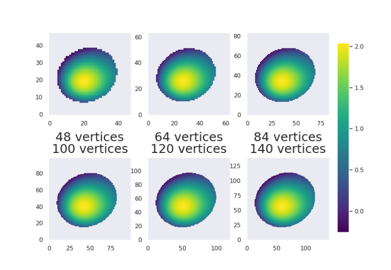




Errors of the three methods.

Multi-resolution abilities?



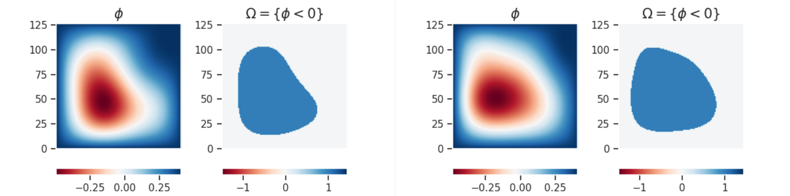


Second test case

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

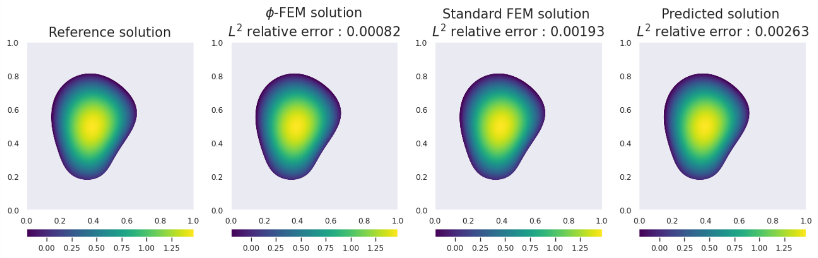
where Ω is defined using Fourier series,

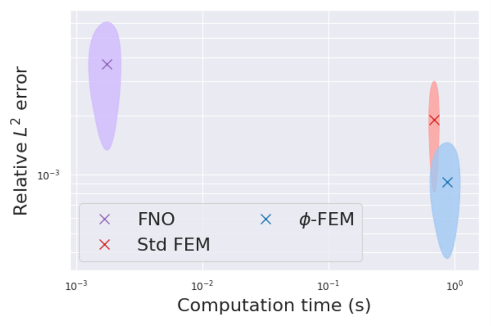
$$\phi(x,y) = 0.4 - \sum_k \sum_l \alpha_{kl} \sin(k\pi x)\sin(l\pi y),$$



Examples of level-set functions and corresponding domains.

ϕ-FEM and FNO : complex shapes





Conclusion

Everything seems to be working 😊

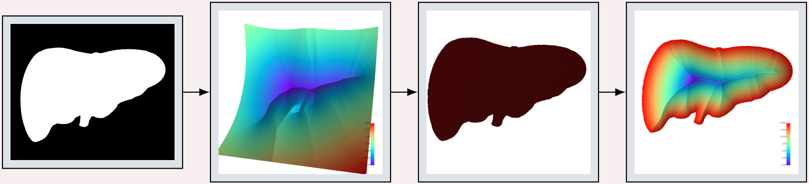
Conclusion

Everything seems to be working 😊

Ongoing works

- how to construct sufficiently smooth level-set functions from medical images ?

  ⟶ First interesting results in 2D and 3D, fast method



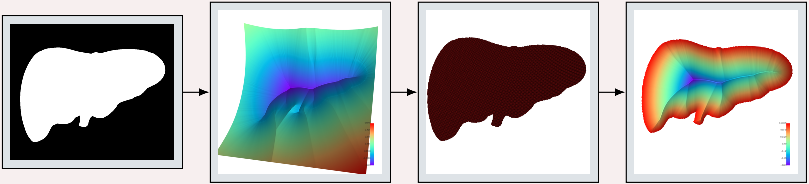
Conclusion

Everything seems to be working 😊

Ongoing works

- how to construct sufficiently smooth level-set functions from medical images ?

  $\longrightarrow$ First interesting results in 2D and 3D, fast method



- ϕ-FEM for mixed Dirichlet-Neumann boundary conditions.

Thank you for your attention!

Let,

$$V_h = \langle \psi_k \in H_0^1(\Omega) \, : \, k \in 1, \dots, N \rangle \, .$$

Find $u_h \in V_h$ s.t. :

$$\int_\Omega \nabla u_h \cdot \nabla \psi_k = \int_\Omega f \psi_k \, , \, \forall k$$

$\iff$ Find $U_h \in \mathbb{R}^N$ s.t. :

$$A_h U_h = F_h \, , \text{ where } \begin{cases} A_h & = \left(\int_\Omega \nabla \psi_k \cdot \nabla \psi_j \right)_{kj} \\ F_h & = \left(\int_\Omega f \psi_k \right)_k \\ U_h & = \left(U_{h,k} \right)_k \end{cases}$$

The final solution is then :

$$u_h = \sum_{k=1}^{N} U_{h,k} \psi_k \, .$$

Example (Poisson-Dirichlet equation)

Recall eq. (1) :

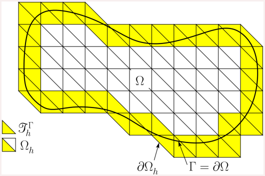
$$-\Delta u = f \quad \text{in}\, \Omega, \quad u = 0 \quad \text{on}\, \Gamma\,. \tag{1}$$

ϕ-FEM scheme

Find w_h such that for all v_h,

$$\int_{\Omega_h} \nabla(\phi_h w_h)\cdot\nabla(\phi_h v_h) - \int_{\partial\Omega_h} \frac{\partial}{\partial n}(\phi_h w_h)\phi_h v_h$$

$$+ \text{stabs} = \int_{\Omega_h} f\phi_h v_h - \text{stabs}\,.$$



Who are « stabs » ?

- First order : Ghost penalty,
- Second order : mean square imposition of (1) on $\mathcal{T}_h^\Gamma$.

- $\mathcal{F}$, 2-dimensional Discrete Fourier transform (DFT) on the $ni \times nj$ grid :

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

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

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

- $\mathcal{C}^{\ell}_{\theta}(X)$, the convolution kernel :

$$\mathcal{C}^{\ell}_{\theta}(X) = \mathcal{F}^{-1}\left(\mathcal{F}(X) \cdot \hat{W}\right).$$

$$\mathcal{L} = \frac{1}{N} \sum_{n=0}^{N} \sqrt{\frac{\mathcal{E}_0(\omega^n u^n, \omega^n \hat{u^n}) + \mathcal{E}_1(\omega^n u^n, \omega^n \hat{u^n}) + \mathcal{E}_2(\omega^n u^n, \omega^n \hat{u^n})}{\mathcal{N}_0(\omega^n u^n) + \mathcal{N}_1(\omega^n u^n) + \mathcal{N}_2(\omega^n u^n)}} \,,$$

where

$$\begin{aligned}
\mathcal{E}_0(\omega u, \omega \hat{u}) &= \mathsf{MSE}(\omega u, \omega \hat{u}) \,, \\
\mathcal{E}_1(\omega u, \omega \hat{u}) &= \mathsf{MSE}(\omega \nabla_x^h u, \omega \nabla_x^h \hat{u}) + \mathsf{MSE}(\omega \nabla_y^h u, \omega \nabla_y^h \hat{u}) \,, \\
\mathcal{E}_2(\omega u, \omega \hat{u}) &= \mathsf{MSE}(\omega \nabla_x^h \nabla_x^h u, \omega \nabla_x^h \nabla_x^h \hat{u}) \\
&\quad + \mathsf{MSE}(\omega \nabla_x^h \nabla_y^h u, \omega \nabla_x^h \nabla_y^h \hat{u}) + \mathsf{MSE}(\omega \nabla_y^h \nabla_y^h u, \omega \nabla_y^h \nabla_y^h \hat{u}) \,,
\end{aligned}$$

and

$$\mathcal{N}_0(\omega u) = \frac{1}{ni \times nj} \sum_{i=0}^{ni} \sum_{j=0}^{nj} \|\omega(i,j) u(i,j)\|^2 \,,$$

$$\mathcal{N}_1(\omega u) = \frac{1}{ni \times nj} \sum_{i=0}^{ni} \sum_{j=0}^{nj} \left(\|\omega(i,j) \nabla_x^h u(i,j)\|^2 + \|\omega(i,j) \nabla_y^h u(i,j)\|^2 \right) \,,$$

$$\begin{aligned}
\mathcal{N}_2(\omega u) &= \frac{1}{ni \times nj} \sum_{i=0}^{ni} \sum_{j=0}^{nj} \Big(\|\omega(i,j) \nabla_x^h \nabla_x^h u(i,j)\|^2 \\
&\quad + \|\omega(i,j) \nabla_x^h \nabla_y^h u(i,j)\|^2 + \|\omega(i,j) \nabla_y^h \nabla_y^h u(i,j)\|^2 \Big) \,,
\end{aligned}$$