

# Construction de modèles réduits pour des EDPs fortement non-linéaires avec des PINNs

Vincent Italiano

Encadrants: E.Franck (INRIA), V. Michel-Dansac (INRIA), V. Vigon (IRMA) et H. Baty (ObAS)

Stage de fin de Master

- Succès des méthodes de deep learning (traitement automatique du langage, computer vision, etc.) [[LeCun et al., 2015](#)].
- Peu d'utilisation dans le domaine des méthodes numériques jusqu'à l'arrivée des PINNs [[Raissi et al., 2017](#)].
- Nécessité d'avoir des modèles réduits (optimisation, analyse de sensibilité, etc.).
- Étude des phénomènes de reconnection magnétique [[Baty, 2019](#)].

- Compréhension et prise en main d'un code PINN. ✓
- Essais sur l'équation de Burgers en 1D. ✓
- Cas test en 2D. ✓
- Implémentation de l'échantillonnage préférentiel. ✓
- Réduction de modèle sur l'équation de Burgers. ✓
- Compréhension et implémentation de la MHD réduite. ✓
- Résolution de la MHD avec des PINNs. ✓
- Réduction de modèle sur la MHD. ✗

# Physics-Informed-Neural-Networks (PINNs)

On considère des EDP génériques

$$\begin{aligned}\mathcal{N}[t, x, u] &= 0 && \text{dans } ]0, T] \times \Omega \\ \mathcal{I}[x, u] &= 0 && \text{dans } \Omega \\ \mathcal{B}[t, x, u] &= 0 && \text{dans } [0, T] \times \partial\Omega\end{aligned}$$

sur un domaine  $\Omega \in \mathbb{R}^d$  de bord  $\partial\Omega$ .

- $\mathcal{N}$  est un opérateur différentiel qui représente l'EDP.
- $\mathcal{I}$  représente la condition initiale.
- $\mathcal{B}$  représente les conditions aux bords (Dirichlet, Neumann, Robin).

Un PINN est un réseau de neurones  $u_\theta$  qui approxime la solution de l'EDP ( $\theta$  est l'ensemble des paramètres du réseau)

Entraînement des PINNs: algorithme de gradient stochastique. On tire de manière uniforme :

- $N_r$  points  $\{(t_i^r, x_i^r)\}_{i=1}^{N_r}$  dans  $]0, T] \times \Omega$
- $N_0$  points  $\{(t_i^0, x_i^0)\}_{i=1}^{N_0}$  dans  $\{0\} \times \Omega$
- $N_b$  points  $\{(t_i^b, x_i^b)\}_{i=1}^{N_b}$  dans  $]0, T] \times \partial\Omega$

Puis on minimise la loss

$$J(\boldsymbol{\theta}) = J_{\mathcal{N}}(\boldsymbol{\theta}) + \lambda_1 J_{\mathcal{I}}(\boldsymbol{\theta}) + \lambda_2 J_{\mathcal{B}}(\boldsymbol{\theta})$$

sur un batch de  $m$  points de collocation tirés uniformément à chaque itération.

- MSE sur le résidu de l'EDP

$$J_{\mathcal{N}}(\boldsymbol{\theta}) = \frac{1}{m} \sum_{j=1}^m \mathcal{N}[t_j^r, x_j^r, u_{\boldsymbol{\theta}}(t_j^r, x_j^r)]^2$$

- MSE sur les conditions initiales et conditions aux bords

$$J_{\mathcal{I}}(\boldsymbol{\theta}) = \frac{1}{N_0} \sum_{j=1}^{N_0} \mathcal{I}[x_j^0, u_{\boldsymbol{\theta}}(0, x_j^0)]^2 \quad \text{et} \quad J_{\mathcal{B}}(\boldsymbol{\theta}) = \frac{1}{N_b} \sum_{j=1}^{N_b} \mathcal{B}[t_j^b, x_j^b, u_{\boldsymbol{\theta}}(t_j^b, x_j^b)]^2$$

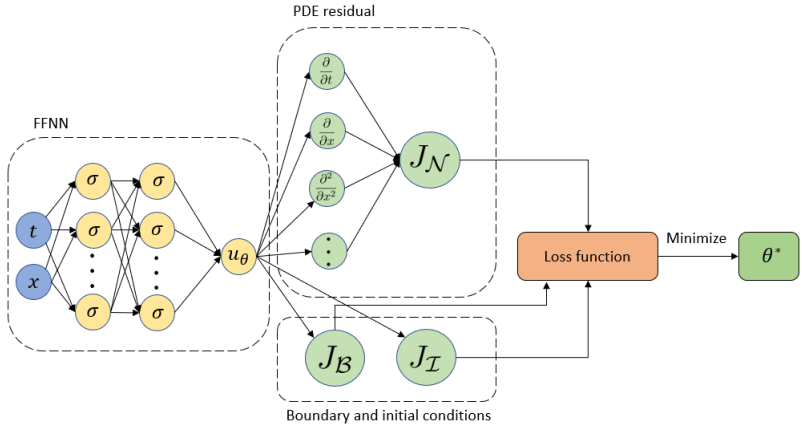


Figure: Récapitulatif PINNs

# 1er exemple : équation de Burgers en 1D

1er cas test :  $\Omega = [-1, 1]$  et  $\nu$  est la viscosité

$$\begin{cases} \partial_t u + \frac{1}{2} \partial_x u^2 = \nu \partial_{xx} u & \text{sur } [0, 1] \times \Omega \\ u(0, x) = \sin(\pi x) & \forall x \in \Omega \\ u(t, -1) = u(t, 1) = 0 & \forall t \in [0, 1] \end{cases}$$

PINN prediction VS exact solution for  $\nu = 3.00e-03$

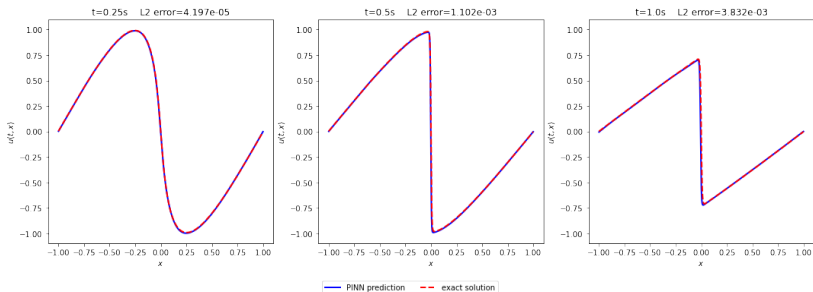


Figure: Solution de l'équation de Burgers. Prédiction du PINN vs solution exacte

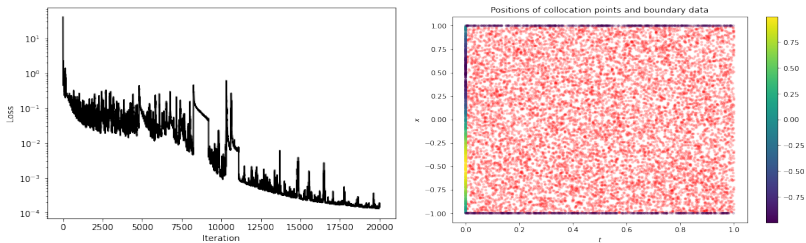


Figure: Évolution de la loss pendant l'entrainement (gauche) répartition des points de collocations (droite)



## 2ème exemple : équation de Burgers 2d

EDP en 2 variables d'espace sur le domaine  $\Omega = [0, 1] \times [0, 1]$  et  $\nu$  est la viscosité.

$$\begin{cases} \partial_t u + \frac{1}{2} \partial_x u^2 + \partial_y u = \nu \Delta u & \text{sur } [0, T] \times \Omega \\ u(0, x, y) = 1.5 - 2x & \forall (x, y) \in [0, 1] \times [0, 1] \\ u(t, 0, y) = 1.5, u(t, 1, y) = -0.5, u(t, x, 0) = 1.5 - 2x \end{cases}$$

Résultats numériques :  $\nu = 10^{-3}$ , avec un réseau de 8 couches et 150 neurones par couche. 4h d'entrainement :

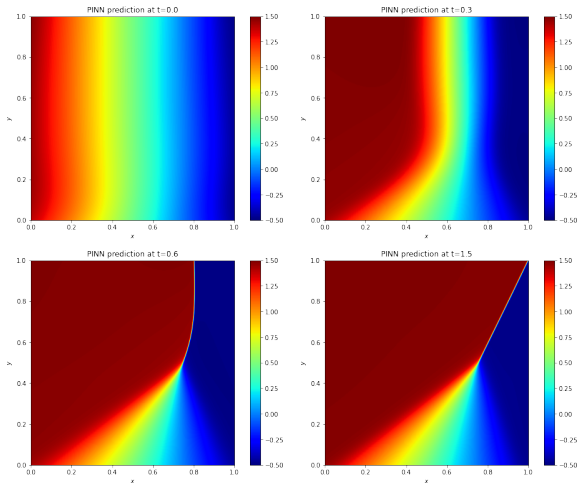


Figure: Solution de l'EDP pour  $\nu = 10^{-3}$

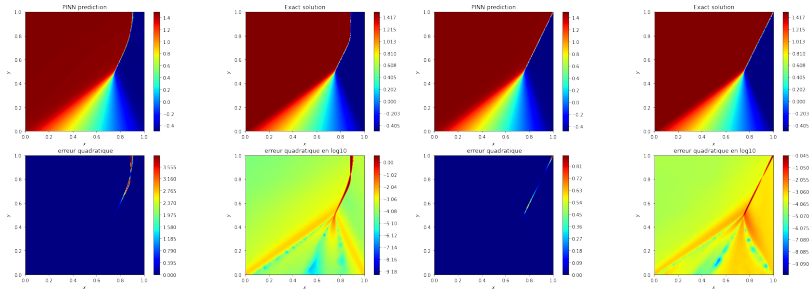


Figure: Comparaison avec la solution exacte sur tout le domaine

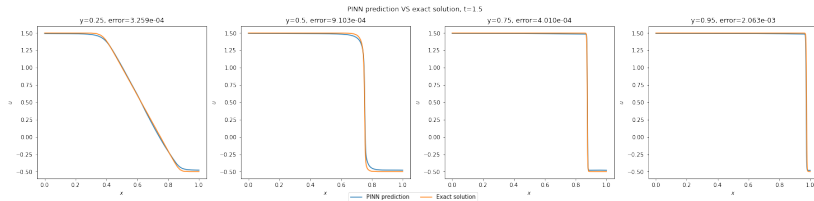


Figure: Comparaison avec la solution exacte selon des tranches horizontales

Idee : modifier le tirage des points utilisés lors de l'entrainement pour accélérer la convergence

Deux critères envisagés :

- Le résidu : on choisit les points où le résidu est le plus grand.
- Un critère spécifique à l'EDP : par exemple choisir les points où l'entropie est la plus grande pour l'équation de Burgers.

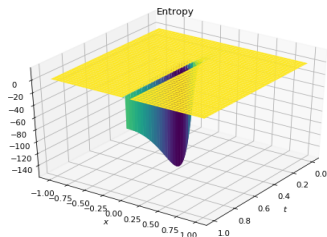
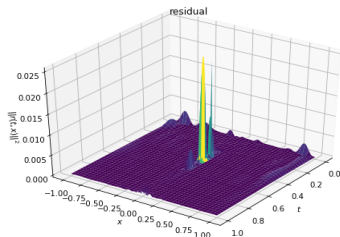


Figure: Résidu (gauche) Entropie (droite) pour une solution de l'équation de Burgers

---

**Algorithme 1** : Echantillonnage préférentiel

---

Générer  $N_r$  points de collocation tirés uniformément dans  $]0, T] \times \Omega$ .

**tant que**  $J(\theta^i) > \varepsilon$  **faire**

    Calculer les probabilités  $q_j^i$  selon le critère choisi

    Tirer  $(1 - \alpha)m$  points de collocation selon la loi multinomiale de paramètre  $p^i = (q_j^i)$

    Tirer  $\alpha m$  points de collocation selon une loi uniforme.

    Effectuer la descente de gradient sur le batch obtenu.

**fin**

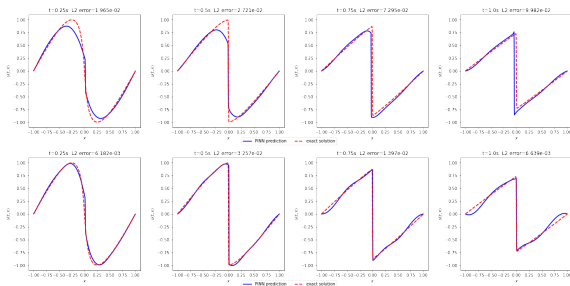
---

Avec

$$q_j^i = \frac{\mathcal{C}(t_j^r, x_j^r; \theta^i)}{\sum_{j=1}^N \mathcal{C}(t_j^r, x_j^r; \theta^i)} \quad \forall j = 1, \dots, N_r \quad (1)$$

où  $\mathcal{C}$  désigne le critère choisi (résidu ou entropie).

Le temps d'entraînement avec un échantillonnage préférentiel est plus long (évaluation de la loss en chaque points) mais permet de converger plus vite (en itérations).



**Figure:** Comparaison précision du PINN sur une viscosité faible  $\nu = 10^{-4}$ . Avec un tirage uniforme (haut) avec un tirage préférentiel selon le résidu (bas)

Il devient inutile lorsqu'on a beaucoup de points de collocations.

# Modèle réduit pour l'équation de Burgers 1d

Idée : Les PINNs sont peu sensibles à la dimension. On considère la viscosité comme une variable, on obtient un problème à 3 variables ( $t, x, \nu$ ).

- Réseau de taille 16 couches et 256 neurones par couche.
- 2h d'entraînement ( $\simeq$  autant de temps que pour juste  $\nu = 10^{-4}$ )
- Plage de viscosité allant de  $10^{-2}$  à  $10^{-4}$
- Loss finale obtenue:  $10^{-5}$

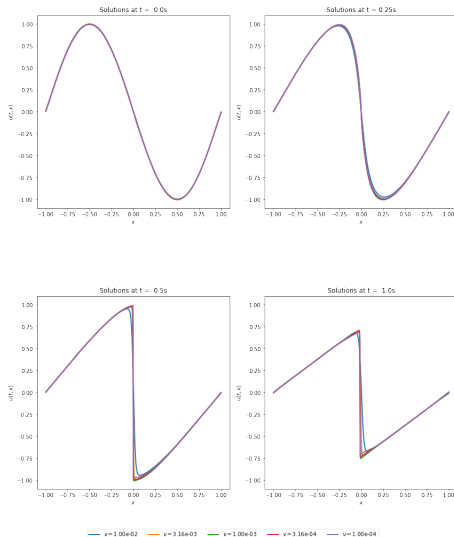
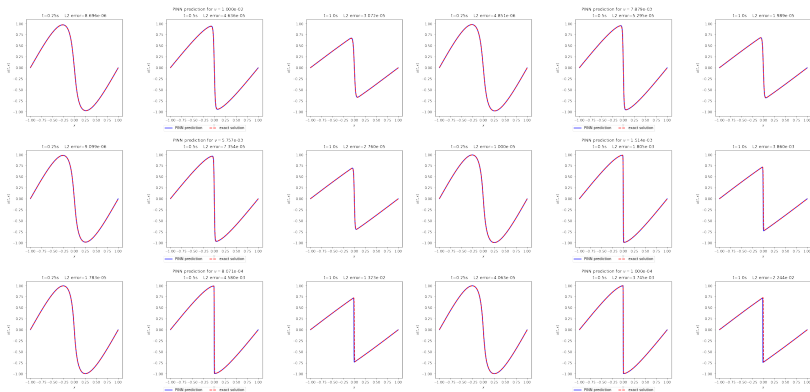


Figure: Prédiction du PINN pour plusieurs viscosités





# Application à la magnétohydrodynamique

Magnétohydrodynamique (MHD) = dynamique des fluides conducteurs du courant électrique en présence de champs électromagnétiques (plasmas).

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho V) = 0 \\ \partial_t V + (V \cdot \nabla) V = \frac{J \times B - \nabla P}{\rho} + \nu \Delta V \\ \partial_t B = \nabla \times (V \times B - J) \\ \partial_t P + V \cdot \nabla P = -\Gamma P \nabla \cdot V \end{cases}$$

avec  $J = \nabla \times B$  et  $\nabla \cdot B = 0$ .

$B$  est le champ magnétique,  $J$  le courant,  $V$  la vitesse du fluide,  $\rho$  sa densité,  $P$  sa pression,  $\nu$  sa viscosité, et  $\Gamma$  le coefficient adiabatique.

Modèle réduit de la MHD : fluide incompressible, dynamique restreinte à un plan, avec viscosité  $\nu$  et résistivité  $\eta$  uniformes et  $\rho = \mu_0 = 1$ .

Les équations se réduisent à un système de 4 variables scalaires

$$\begin{cases} \partial_t \omega + (V \cdot \nabla) \omega = (B \cdot \nabla) J + \nu \Delta \omega \\ \partial_t \psi + (V \cdot \nabla) \psi = \eta \Delta \psi \\ \Delta \phi = -\omega \\ \Delta \psi = -J \end{cases}$$

$\omega$  est la vorticité et  $\psi$  le potentiel vecteur du champ magnétique

Configuration d'équilibre initiale :  $\omega_e = \phi_e = 0$

$$\psi_e(x, y) = \begin{cases} \left(\frac{1}{r} - r\right)\frac{y}{r} & \text{si } r > 1 \\ -\frac{2}{j_0(k)}j_1(kr)\frac{y}{r} & \text{si } r \leq 1 \end{cases} \quad J_e(x, y) = \begin{cases} 0 & \text{si } r > 1 \\ -\frac{2k}{j_0(k)}j_1(kr)\frac{y}{r} & \text{si } r \leq 1 \end{cases}$$

avec  $r = \sqrt{x^2 + y^2}$ ,  $j_0$  et  $j_1$  les première et seconde fonctions de Bessel de première espèce et  $k = 3.83170597$

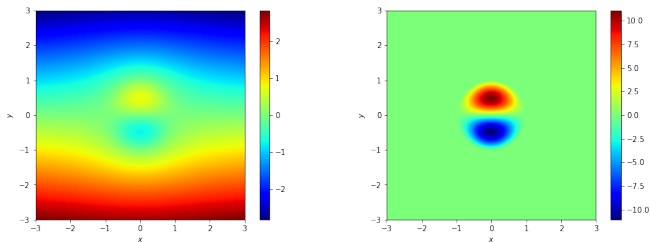
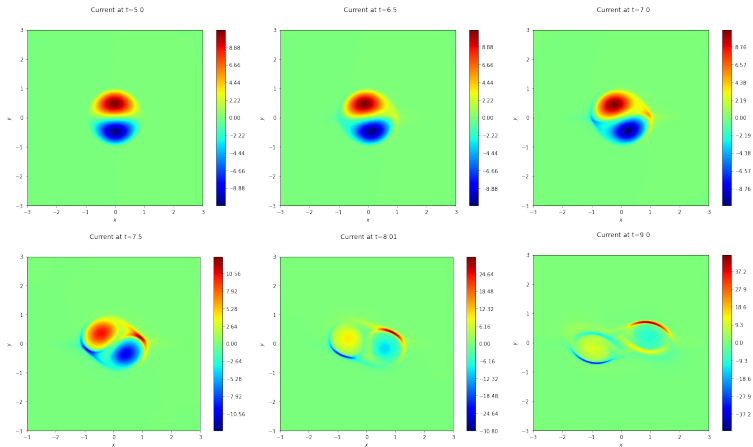


Figure: Configuration initiale :  $\psi_e$  à gauche et  $J_e$  à droite



**Figure:** Évolution du courant  $J$  durant le développement de l'instabilité de tilt pour une résistivité  $\eta = 3 \times 10^{-3}$ .

Difficulté : problème multi-échelle où les solutions prennent des formes complexes

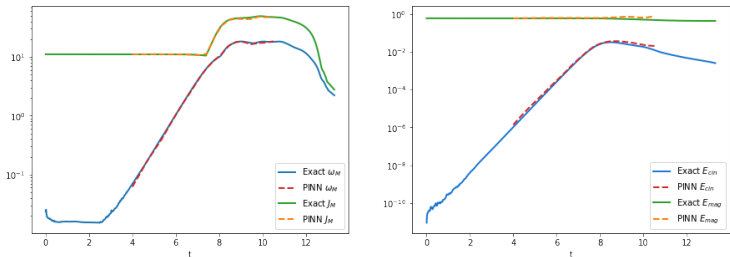
- Modification de la loss pour intégrer des données

$$J(\theta) = \frac{1}{m} \sum_{j=1}^m \mathcal{N}[t_j^r, x_j^r, u_{\theta}(t_j^r, x_j^r)]^2 + \underbrace{\frac{\lambda}{N_d} \sum_{j=1}^{N_d} \left[ u_j^d - u_{\theta}(t_j^d, x_j^d) \right]^2}_{=J_{\text{data}}(\theta)}$$

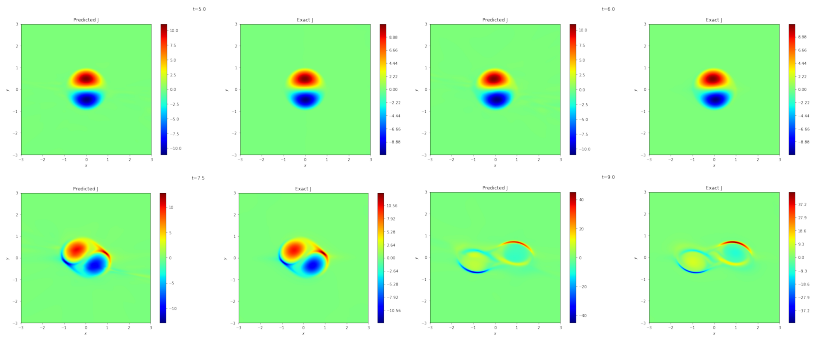
- On utilise plusieurs snapshots pour guider l'apprentissage
- Plusieurs réseaux de neurones composent le PINNs

Cela augmente beaucoup le nombre d'hyperparamètres à régler.

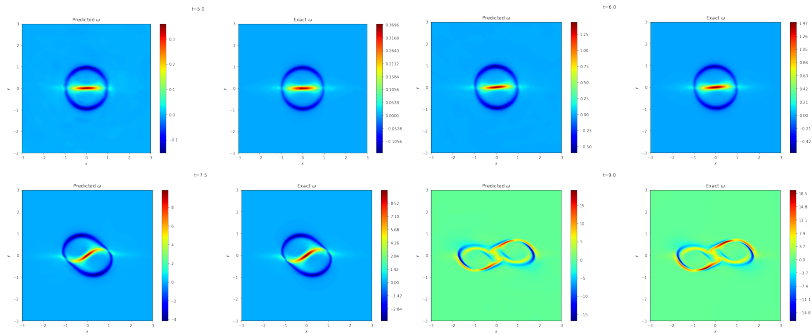
- résistivité  $\eta = \nu = 3 \times 10^{-3}$
- 2 réseaux de 16 couches et 256 neurones par couche.
- 10 snapshots (donc 44 hyperparamètres pour l'équilibrage de la loss)
- 4h d'entrainement



**Figure:** Prédiction PINNs vs solutione référence : amplitude de courant et de vorticité maximale (gauche) énergie cinétique et magnétique du système (droite)

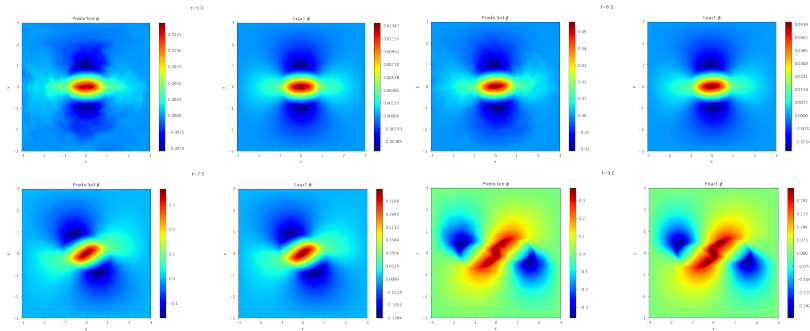


**Figure:** Comparaison du courant  $J$  prédit par les PINNs vs la solution de référence

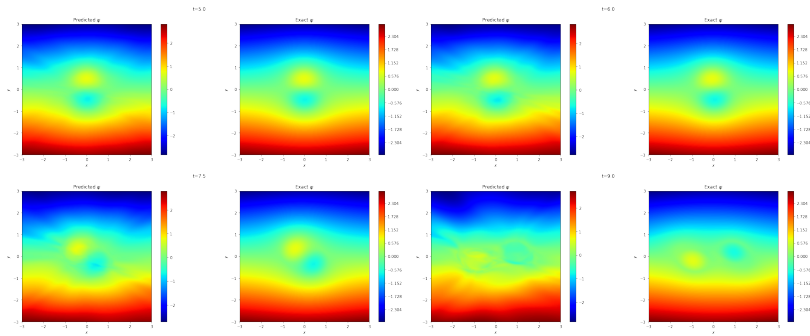


**Figure:** Comparaison de la vorticité  $\omega$  prédite par les PINNs vs la solution de référence





**Figure:** Comparaison de la fonction  $\phi$  prédite par les PINNs vs la solution de référence



**Figure:** Comparaison de la fonction  $\psi$  prédite par les PINNs vs la solution de référence

- Les PINNs peuvent résoudre des EDPs complexes sans maillage
- Difficulté pour les systèmes multi-échelle
- Entraînement nécessite de régler beaucoup d'hyperparamètres

# Bibliographie I



Baty, H. (2019).

FINMHD: An adaptive finite-element code for magnetic reconnection and formation of plasmoid chains in magnetohydrodynamics.

<https://arxiv.org/abs/1904.11173>.



Baty, H. (2022).

Reconnexion magnétique et instabilités magnétohydrodynamiques dans les plasmas.

<https://hal.archives-ouvertes.fr/hal-03711796>.

working paper or preprint.



Chen, W., Wang, Q., Hesthaven, J. S., and Zhang, C. (2021).

Physics-informed machine learning for reduced-order modeling of nonlinear problems.

<https://www.sciencedirect.com/science/article/pii/S0021999121005611>.



Cuomo, S., di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., and Piccialli, F. (2022).

Scientific machine learning through physics-informed neural networks: Where we are and what's next.

<https://arxiv.org/abs/2201.05624>.



Haghighat, E., Raissi, M., Moure, A., Gomez, H., and Juanes, R. (2021).

A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics.

<https://www.sciencedirect.com/science/article/pii/S0045782521000773>.

## Bibliographie II



Krishnapriyan, A. S., Gholami, A., Zhe, S., Kirby, R. M., and Mahoney, M. W. (2021).  
Characterizing possible failure modes in physics-informed neural networks.  
<https://arxiv.org/abs/2109.01050>.



LeCun, Y., Bengio, Y., and Hinton, G. (2015).  
Deep learning.  
<https://www.nature.com/articles/nature14539>.



Lu, L., Meng, X., Mao, Z., and Karniadakis, G. (2021).  
Deepxde: A deep learning library for solving differential equations.  
<https://epubs.siam.org/doi/pdf/10.1137/19M1274067>.



Moseley, B., Markham, A., and Nissen-Meyer, T. (2021).  
Finite basis physics-informed neural networks (fbpinns): a scalable domain decomposition  
approach for solving differential equations.  
<https://arxiv.org/abs/2107.07871>.



Nabian, M. A., Gladstone, R. J., and Meidani, H. (2021).  
Efficient training of physics-informed neural networks via importance sampling.  
<https://arxiv.org/abs/2104.12325>.



Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017).

Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations.

<https://arxiv.org/abs/1711.10561>.



Rao, C., Sun, H., and Liu, Y. (2020).

Physics informed deep learning for computational elastodynamics without labeled data.

<https://arxiv.org/abs/2006.08472>.