Simulations multi-échelle d'électromagnétisme sur des domaines complexes

avec les PINNs

Marie Sengler

Faculty of Mathematics
University of Strasbourg

August 19, 2024

Université de Strasbourg

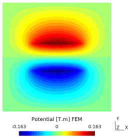
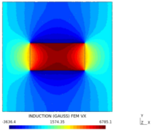
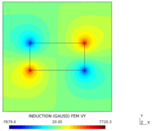
Rappel et objectifs

- Lors du semestre, on avait travaillé sur le thème des PINNs (Physics Informed Neural Network) [2].
- Les problèmes de physiques sont présentés dans un article de fin d'étude avec l'entreprise Thales [1].

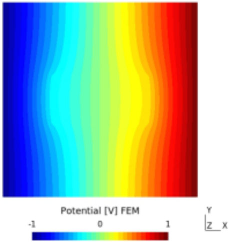
Objectifs généraux

- Le premier objectif du stage est d'améliorer les résultats obtenus en modifiant les différents paramètres et poids.
- Le second grand objective est d'implémenter un réseau greedy.

Rappel et premier objectif

Potentiel A $(T.m)$	Champ d'induction B_x (Gauss)	Champ d'induction B_y (Gauss)
		

(a) Problème de Magnétostatique avec Bimatériaux



(b) Problème d'électrostatique avec Bimatériaux

Figure: Solution calculée par un logiciel élément finis de Thales

Université de Strasbourg

Scimba

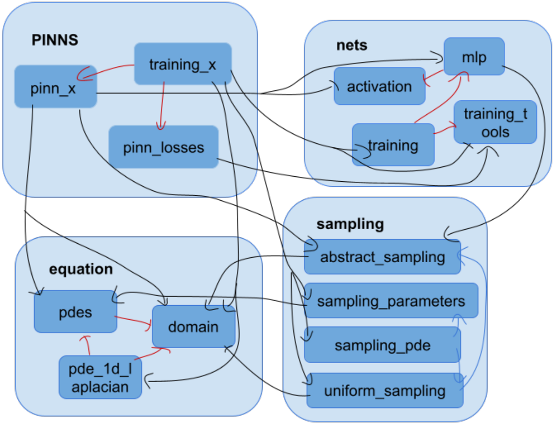


Figure: Diagramme représentant les différents liens entre les fichiers pour la partie PINNs de scimba

Université de Strasbourg

Réseau glouton

Definition

Suivant le principe d'un algorithme glouton, dans le cas des PINNs, on utilise à chaque étape un réseau de neurone différent pour affiner notre prédiction. On s'est basé sur l'étude "Greedy training algorithms for neural networks and applications to PDEs" du Journal of Computational Physics [3]

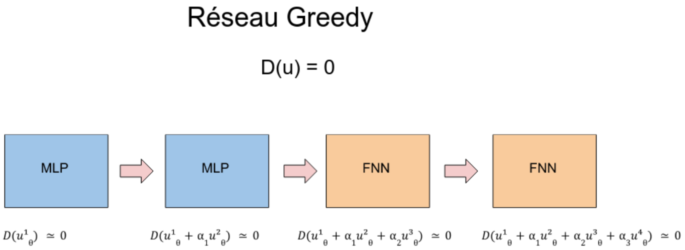


Figure: Schéma du fonctionnement d'un réseau greedy

Ajout de nouvelles fonctionnalités

- Nouveaux graphiques et modifications des anciens
  - Graphiques de prédiction intermédiaire à chaque sous-réseau prise de manière globale ou isolée
  - Graphiques Bx et By: Changement sur la manière d'interpoler et sur l'échelle de valeur
- Pouvoir évaluer une prédiction sans refaire un entraînement
- Pouvoir reprendre un entraînement à partir d'un réseau donné

Université de Strasbourg

Ajout de nouvelles fonctionnalités

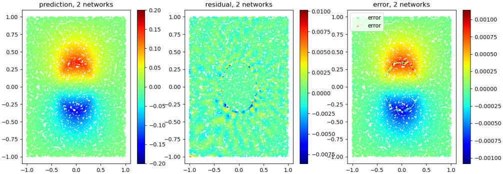


Figure: Prédiction aux réseaux n°2 du problème d'Electrostatique avec Bimatériaux

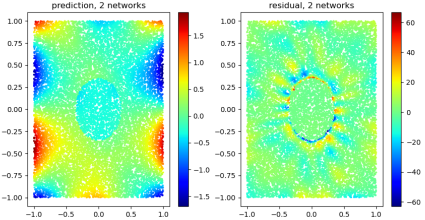


Figure: Prédiction isolée aux réseaux n°2 du problème d'Electrostatique avec Bimatériaux

Magnétostatique sur un cercle

$$
\left\{
\begin{aligned}
\Delta A_m &= 0 \in \Omega_m \\
\Delta A_v &= 0 \in \Omega_v \\
(B_{xv} - B_{xm}) \cdot n_x + (B_{yv} - B_{ym}) \cdot n_y &= 0 \in \Gamma \\
(H_{xv} - H_{xm}) \cdot n_y - (H_{yv} - H_{ym}) \cdot n_x &= 0 \in \Gamma \\
A_m - A_v &= 0 \in \Gamma \\
A_v &= 0 \in \delta\Omega
\end{aligned}
\right.
$$

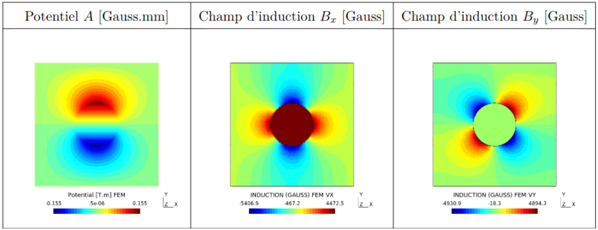


Figure: Solution du problème de Magnétostatique avec Bimatériaux sur un cercle calculé par éléments finis de Thales

Calcul des alphas et fréquences

Pour calculer les alphas optimaux de la pondération de la somme des prédiction, on se base sur l'article *Multi-stage neural networks: Function approximator of machine precision* [4].

$$\phi(x) \approx \sum_i \alpha_i \phi_i(x)$$

Fréquence dominante

Si la fonction source a une certaine fréquence dominante (par exemple, elle oscille avec une certaine fréquence), alors la solution $u(x)$ doit osciller avec la même fréquence dominante pour que l'équation différentielle soit équilibrée.

$$f_d \approx f_0 \epsilon^{-\frac{1}{6}}$$

$$\epsilon_1 = \frac{RMS(residu(x, u_0))}{(2\pi f x_d)(2\pi f y_d) RMS(\beta_m)}$$

avec $\beta_m = \frac{\delta \mathcal{N}}{\delta u^{(m)}}|_{u=u_0}$ et $\mathcal{N}(u) = \Delta u$

Recherche d'une configuration optimale

- Evaluation des alphas
- Evaluation des fréquences f0x et f0y
- Modification de l'architecture des réseaux
  - Type du réseau : MLP, FNN
  - Nombres de couches et neurones par couche
  - Fonction d'activation
  - Nombres de features, et variance pour les FNN
- Taux d'apprentissage et schedulder
- Nombre de points de collocation
- Choix de l'optimizer (Adam ou LBFGS)

Université de Strasbourg

Automatisation du calcul des alphas

Point de divergence avec l'article

- Nos fonctions sources sont nulles
- Nous avons des conditions de bord et d'interface qui peuvent compliquer la satisfaction de l'équation dans le domaine des fréquences.

La méthode de l'article n'est donc pas applicable pour nos problèmes.
Pour calculer nos alphas successifs, nous avons besoin de déterminer les fréquences dominantes. On fait alors une **fast fourier transform** (fft) avec NumPy de nos solutions intermédiaires.

$$\epsilon_1 = \frac{RMS(residu(x, u_0))}{(2\pi fx_d)(2\pi fy_d)RMS(\beta_m)}$$

Université de Strasbourg

Résultats électrostatique

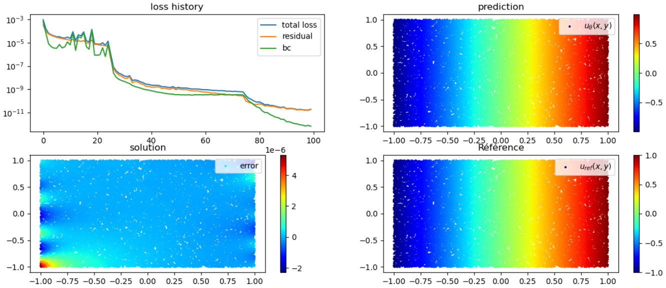


Figure: Prédiction du problème d'électrostatique

	sol
Erreur max	10^{-5}
MSE	10^{-8}
MAE	10^{-6}

Table: Tableau des erreurs

Résultats électrostatique bimatériaux

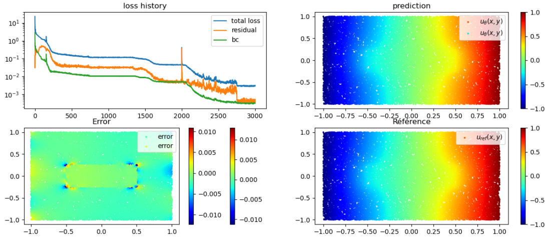


Figure: Prédiction du problème d'électrostatique avec bimatériaux

	sol
Erreur max	$1 * 10^{-2}$
MSE	$3 * 10^{-6}$
MAE	$1 * 10^{-4}$

Table: Tableau des erreurs

Résultats magnétostatique rectangle

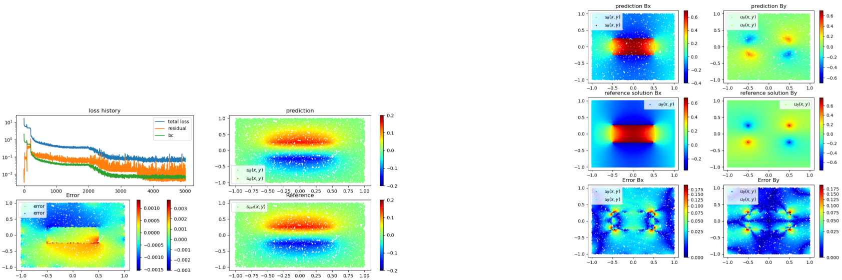


Figure: Prédiction de A, Bx, By du problème de magnétostatique sur un rectangle

	A	Bx	By
Erreur max	$2 * 10^{-3}$	0.14	0.08
MSE	$3 * 10^{-6}$	$1 * 10^{-4}$	$8 * 10^{-5}$
MAE	$3 * 10^{-4}$	$3 * 10^{-3}$	$1 * 10^{-3}$

Table: Tableau des erreurs pour A, Bx, et By

Résultats magnétostatique cercle

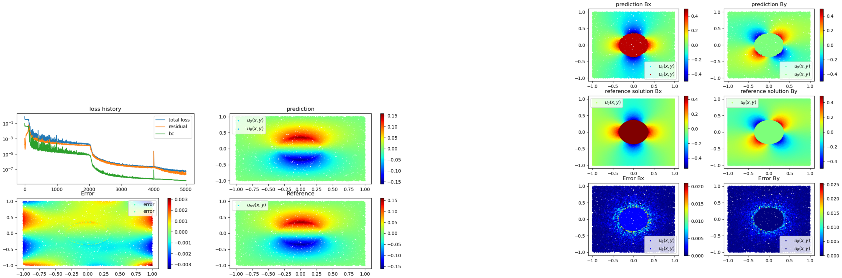


Figure: Prédiction de A, Bx, By du problème de magnétostatique sur un cercle

	A	Bx	By
Erreur max	$9.9 * 10^{-4}$	0.02	0.018
MSE	$7 * 10^{-6}$	$1 * 10^{-5}$	$6 * 10^{-6}$
MAE	$2 * 10^{-4}$	$1 * 10^{-3}$	$5.8 * 10^{-4}$

Table: Tableau des erreurs pour A, Bx, et By

Université de Strasbourg

Conclusion: Comparaison

- Greedy donne des meilleurs résultats sur les erreurs que sur le classique
- L'entraînement est plus complexe, plus long, et plus subsceptible d'avoir des problèmes de mémoires
- Comme dans notre cas classique la fonction de perte stagne à partir d'un certain nombre d'epochs

Université de Strasbourg

Reference I

TRYOEN Julie.
Mise en œuvre d'un métamodèle pinn pour la simulation de champ magnétique.
Master's thesis, INSA Toulouse, 2022-2023.

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, 378:686–707, 2019.

Jonathan W. Siegel, Qingguo Hong, Xianlin Jin, Wenrui Hao, and Jinchao Xu.
Greedy training algorithms for neural networks and applications to pdes.
Journal of Computational Physics, 484:112084, 2023.

Yongji Wang and Ching-Yao Lai.
Multi-stage neural networks: Function approximator of machine precision.
Journal of Computational Physics, 504:112865, 2024.

Université de Strasbourg