

Sujet de stage de recherche

Impl mentation d'un r seau de neurones interpr table pour la simulation et la r duction de mod les magn tothermiques

Pr sentation et contexte

  l' re de l'industrie 4.0, la simulation num rique occupe une place centrale, que ce soit pour la conception, l'optimisation, la propagation d'incertitudes, les  tudes param triques, ou l' tablissement de jumeaux num riques [1]. Ce recours massif   la simulation s'accompagne d'une exigence d'efficacit  des calculs num riques, qui doivent   la fois  tre les plus rapides possible, mais  galement les moins consommateurs en  nergie. La simulation des ph nom nes physiques repose aujourd'hui quasi-exclusivement sur la m thode des  l ments finis [2], d velopp e depuis les ann es 1950. Cette m thode, dont les fondations th oriques et l'impl mentation sont parfaitement ma tris es, reste cependant gourmande en puissance de calcul et peut manquer de flexibilit  vis- -vis des usages de l'industrie, notamment les larges  tudes param triques lors de la phase de conception.

D'autres formalismes de simulation se d veloppent du c t  du machine learning, comme les « Physics-Informed Neural Networks » (PINNs) [3]. Leur principal inconv nient est de s'appuyer sur une architecture non-interpr table, ce qui complique leur g n ralisation   d'autres grandeurs que le champ physique : il est alors difficile d'inclure explicitement dans le mod le des param tres variables (g om triques, mat riaux, chargement, etc.). C'est pr cis ment ce que propose l'architecture « Hierarchical Deep-Learning Neural Network representing Finite Element Method » (HiDeNN-FEM), qui propose une r -impl mentation de la m thode des  l ments finis sur un mod le de r seau de neurones [4,5]. Par rapport   un code  l ments finis classique, la structure tensorielle des r seaux de neurones permet une impl mentation tr s efficace sur GPU, et les nombreux param tres suppl mentaires rendus explicitement accessibles (position des n uds, propri t s mat riaux, etc.) ouvrent des possibilit s naturelles d'optimisation de forme, d'adaptation de maillage et de r duction de mod le, ce qui offre davantage de polyvalence et d'efficacit .

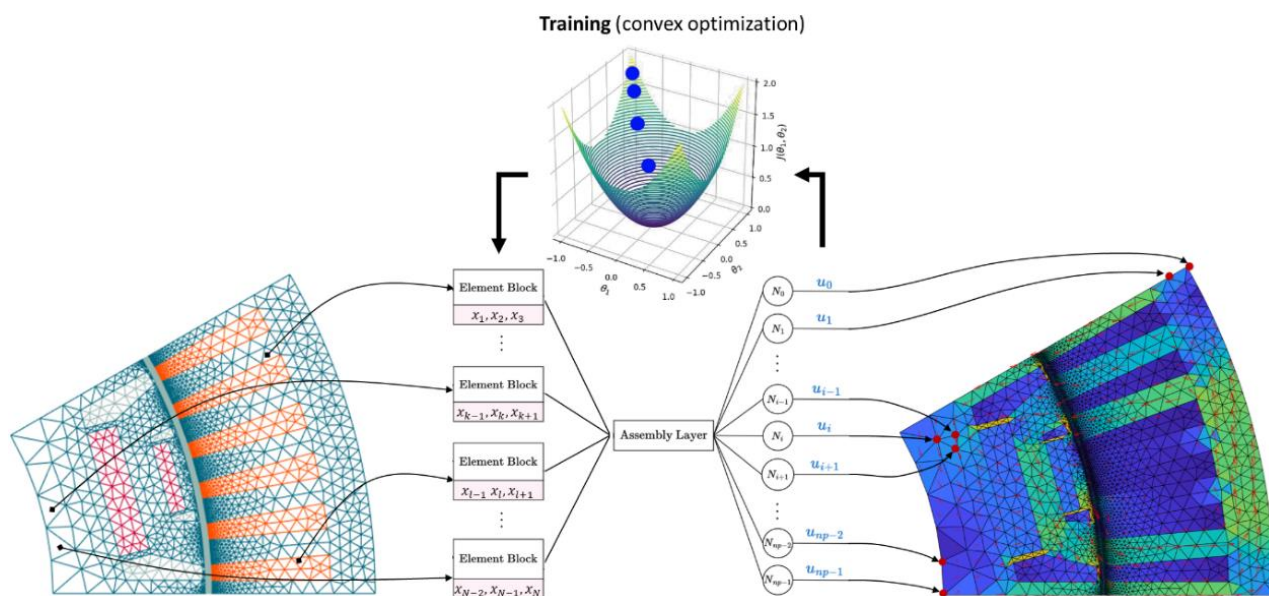


Figure 1 : Sch matisation d'une architecture HiDeNN-FEM appliqu e   la simulation magn tostatique d'un p le de machine  lectrique

Dans ce contexte, le Laboratoire de M canique des Solides (LMS,  cole Polytechnique) d veloppe actuellement un jumeau num rique de poumon pour l'aide au diagnostic de maladies respiratoires avec l'approche HiDeNN-FEM, et a r cemment publi  un code open-source appliqu    la m canique fortement non-lin aire, baptis  NeuROM [6].

Objectifs du stage

Ce stage vise à appliquer la méthode HiDENN-FEM aux physiques et problèmes du génie électrique. Une première étape sera de l'appliquer à un code thermique linéaire 2D. On implémentera notamment un raffinement de maillage automatique basé sur la minimisation de l'énergie totale. Une fois cette prise en main effectuée, on appliquera la méthode à un problème magnétostatique non-linéaire (pôle de machine synchrone à aimant). Selon l'avancement des travaux, une extension à un problème nécessitant des éléments d'arêtes spécifiques pourra être envisagée. Dans le cadre de la collaboration GeePs – LMS, les développements effectués lors de ce stage ont vocation à enrichir le package NeuROM. Il est donc attendu que le code soit correctement documenté et illustré de cas tests fonctionnels.

Profil souhaité

Ce stage s'adresse à des étudiants de niveau M1/M2 (ou 2^e-3^e année d'école d'ingénieur), intéressés par la modélisation numérique et par les architectures de réseaux de neurones. Une aisance dans le langage Python est souhaitée. Sans être requise, une expérience avec la bibliothèque PyTorch serait appréciée. Une connaissance générale de la méthode des éléments finis est un plus.

Déroulement du stage

Ce stage se déroulera au laboratoire GeePs (11 Rue Joliot Curie, 91192 Gif-sur-Yvette, France), au sein d'une équipe spécialisée dans la modélisation électromagnétique basse-fréquence. Des déplacements occasionnels au laboratoire LMS de (École Polytechnique, Route de Saclay, 91120 Palaiseau) sont à prévoir. Le stage durera au moins 8 semaines, à compter de février 2025 (dates flexibles). Le/la stagiaire aura à sa disposition un bureau, un ordinateur fixe équipé de Python et des bibliothèques nécessaires. Il lui sera aussi possible d'utiliser son ordinateur personnel. Le/la stagiaire aura également accès à un grand nombre de ressources bibliographiques (IEEE, Springer, Elsevier...) pour l'accompagner dans son travail.

Contacts

- *Encadrant GeePs*: T. Chèrière (theodore.cherriere@centralesupelec.fr, theodore.cherriere@ricam.oeaw.ac.at)
 - Site personnel : tcherrie.github.io
- *Encadrants LMS*:
 - A. Daby-Seesaram (alexandre.daby-seesaram@polytechnique.edu)
 - Site personnel : alexandredabyseesaram.github.io
 - K. Skardova (katerina.skardova@polytechnique.edu)
 - M. Genet (martin.genet@polytechnique.edu)
 - Site personnel : mgenet.github.io

Références

- [1] Rubmann, M., Lorenz, M., Gerber P. et. al. (2015), « Industry 4.0—The Future of Productivity and Growth in Manufacturing Industries », The Boston Consulting Group (consulté le 05/11/2024)
- [2] Manet, V. (2018). « Méthode des Éléments Finis: vulgarisation des aspects mathématiques et illustration de la méthode ». DEA. Éléments finis pour l'ingénieur, ViM2, Lyon, France. 2018, pp.429. ([cel-00763690v8](https://doi.org/10.1016/j.jcp.2018.10.045))
- [3] Raissi M., Perdikaris P., Karniadakis G.E. (2019). « 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. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [4] Zhang, L., Cheng, L., Li, et. al. (2021). « Hierarchical deep-learning neural networks: Finite elements and beyond. » Computational Mechanics, 67(1), 207–230. <https://doi.org/10.1007/s00466-020-01928-9>
- [5] Liu, Y., Park, C., Lu, et. al. (2023). « HiDeNN-FEM: A seamless machine learning approach to nonlinear finite element analysis. » Computational Mechanics, 72(1), 173–194. <https://doi.org/10.1007/s00466-023-02293-z>
- [6] Daby-Seesaram, A., Škardová, K., & Genet, M. (2024). « Neurom » (Version v3.1.17) [Logiciel]. Zenodo. <https://doi.org/10.5281/zenodo.13907063>