

# Réseaux de neurones à graphes informés par la physique comme opérateur de localisation dans un cadre de lois hyperélastiques

M. R. Guevara Garban<sup>3</sup>, Y. Chemisky<sup>2</sup>, E. Prulière<sup>1</sup>, M. Clément<sup>4</sup>

<sup>1</sup> I2M, Arts et Métiers, [etienne.pruliere@ensam.eu](mailto:etienne.pruliere@ensam.eu)

<sup>2</sup> TIMC, Université Grenoble Alpes, [yves.chemisky@univ-grenoble-alpes.fr](mailto:yves.chemisky@univ-grenoble-alpes.fr)

<sup>3</sup> I2M, Université de Bordeaux, [manuel.guevara-garban@u-bordeaux.fr](mailto:manuel.guevara-garban@u-bordeaux.fr)

<sup>4</sup> LaBRI, Université de Bordeaux, [michael.clement@labri.fr](mailto:michael.clement@labri.fr)

---

## Résumé —

Nous proposons un cadre d'apprentissage automatique permettant la prédiction de champs mécaniques locaux à partir de l'information de la moyenne de ce champ. Notre modèle s'apparente alors à un opérateur de localisation dans le cadre de l'homogénéisation. La méthode repose sur la définition d'un domaine périodique représentatif d'une microstructure, et d'une représentation sous forme duale de maillage et de graphe est construite. Un réseau de neurones permet d'effectuer un passage de message entre les noeuds de ce graphe, permettant de prédire, par passage de messages, le champ mécanique sur tous les noeuds du graphe en se basant sur la valeur moyenne de ce champ, donnée d'entrée du modèle. Dans le cas des champs de contraintes, afin de s'assurer de l'admissibilité physique des champs prédits (composantes du tenseur des contraintes), une contrainte physique de respect de l'équilibre locale est enforced lors de l'apprentissage. Les capacités en terme de fidélité et de rapidité d'exécution de ce modèle sont évalués pour des comportement élastiques puis hyperélastiques. Les avantages d'un réseau de neurones à graphes est de permettre un apprentissage sur une base de donnée contenant des graphes différents, donc sur un ensemble de géométries. Dans le cas hyperélastique non linéaire, la méthode proposée permet d'obtenir des gains de temps de calcul significatifs par rapport à la simulation par éléments finis, la rendant particulièrement attrayante pour des applications à grande échelle.

**Mots clés** — réseau de neurones par graphe, apprentissage, localisation

---

## 1 Introduction

L'innovation technologique nécessite le développement de nouveaux outils de simulation numérique permettant d'appréhender les comportements mécaniques dans le domaine non-linéaire, en tenant compte d'aspects géométriques complexes. La détermination précise mais rapide de champs mécanique locaux, en particulier les champs de contraintes, est un enjeu majeur pour l'analyse de la durabilité des structures et engendre généralement un coût de calcul important. L'objectif de ce travail est de développer un cadre d'apprentissage permettant de réduire ce temps de calcul, sans perte d'information locale. Après une introduction sur les méthodes multi-échelles, le cadre d'apprentissage permettant d'accélérer la simulation de champs locaux sera présenté. Enfin les contraintes physiques et les principaux résultat seront analysés afin d'analyser les capacités prédictives et l'efficacité numérique de la méthode proposée.

### 1.1 Modèles d'apprentissage automatique

Les approches d'apprentissage automatique (Machine Learning ou ML) sont maintenant largement exploités pour modéliser le comportement mécanique de structures, sans se limiter au cas spécifique de la modélisation multi-échelles. Les méthodes d'apprentissage profond, telles que les réseaux de neurones convolutifs (CNN) [3], ont été utilisées pour prédire des champs de contraintes complets sur des géométries 2D, comme le montre le cadre de l'approche « StressNet » [4]. D'autres auteurs [5] ont proposé d'utiliser une architecture U-Net [6] comme cadre d'apprentissage automatique pour accélérer la simulation mécanique multi-échelles de structures hétérogènes à l'échelle macroscopique. Bien que les CNN aient démontré leur efficacité dans la prédiction des champs de contraintes en mécanique, leur applicabilité reste limitée lorsqu'il s'agit d'étendre ces approches à des géométries non structurées, en raison de

l'utilisation de grilles régulière comme donnée géométrique d'entrée pour les GNN. Pour surmonter ces limitations, les réseaux de neurones à graphes (Graph Neural Networks, GNN) [7] opèrent directement sur des données structurées en graphes, telles que des maillages. Les relations spatiales peuvent alors être prises en compte, permettant de rendre compte des champs locaux sur des géométries non structurées. Par ailleurs, de nombreux travaux s'attachent à incorporer des principes informés par la physique, également appelés réseaux de neurones informés par la physique (PINNs, Physics-Informed Neural Networks) [8]. Richter-Powell et al. [10] ont notamment exploré le concept de réseaux de neurones à divergence nulle (divergence-free neural networks) pour la mécanique des fluides en utilisant la différentiation automatique.

## 2 Cadre d'apprentissage automatique

Ce travail se place dans le contexte d'une analyse multi-échelles, en définissant deux sous-problèmes : microscopique (cellule unitaire) et macroscopique, couplés au travers de la définition de champs moyens. Dans une analyse multi-échelle, les deux problèmes sont résolus simultanément à l'aide d'un schéma itératif : à partir de l'analyse macroscopique, les déformations macroscopiques sont utilisées dans le problème de la cellule unitaire afin de calculer les variables microscopiques nécessaires à la détermination de la contrainte macroscopique inconnue. Le problème mécanique traité ici est l'évaluation numérique de la réponse d'une structure périodique, en supposant une séparation des échelles — c'est-à-dire que la taille caractéristique de la cellule locale est plusieurs ordres de grandeur inférieure à celle de la structure — et en appliquant la théorie de l'homogénéisation périodique qui permet de déterminer une réponse effective de la cellule représentative (RUC). Cette méthode est connue sous le nom de méthode FE<sup>2</sup> [1, 2]. À l'intérieur d'une telle cellule, l'équilibre statique local des contraintes, en l'absence de forces volumiques, s'écrit :

$$\operatorname{div} \boldsymbol{\sigma} = 0, \quad \text{qui en 2D se réduit à} \quad \operatorname{div} \boldsymbol{\sigma} = \begin{pmatrix} \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} \\ \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} \end{pmatrix} = 0, \quad (1)$$

où  $\boldsymbol{\sigma}$  est le tenseur des contraintes et  $\operatorname{div}$  l'opérateur divergence. Dans ce qui suit, la loi de comportement est considérée hyperélastique, le cadre des matériaux quasi-incompressible étant utilisé, à partir d'une séparation entre l'énergie de déformation volumique et isochore :

$$\Psi = U(J) + W(\bar{I}_1, \bar{I}_2). \quad (2)$$

Dans l'équation ci-dessus,  $J = \operatorname{Det} \mathbf{F}$  est le déterminant du gradient de la transformation  $\mathbf{F}$ , et  $\bar{I}_1, \bar{I}_2$  sont respectivement les premiers et second invariants du tenseur de Cauchy–Green isochore  $\bar{\mathbf{C}}$  :

$$\begin{aligned} \bar{I}_1 &= J^{-2/3} I_1 & I_1 &= \operatorname{tr}(\mathbf{C}) \\ \bar{I}_2 &= J^{-4/3} I_2 & I_2 &= \frac{1}{2} (\operatorname{tr}(\mathbf{C})\operatorname{tr}(\mathbf{C}) - \operatorname{tr}(\mathbf{C}^2)). \end{aligned} \quad (3)$$

La contrainte de Cauchy  $\boldsymbol{\sigma}$  est également séparée en partie volumique  $p$  et déviatorique  $\mathbf{s}$  :

$$\begin{aligned} \boldsymbol{\sigma} &= p\mathbf{1} + \mathbf{s} & p &= \frac{\partial U}{\partial J} \\ J\mathbf{s} &= 2 \frac{\partial W}{\partial \bar{I}_1} \operatorname{dev} \bar{\mathbf{b}} + 2 \frac{\partial W}{\partial \bar{I}_2} (\operatorname{tr} \bar{\mathbf{b}} \operatorname{dev} \bar{\mathbf{b}} - \operatorname{dev} \bar{\mathbf{b}}^2). \end{aligned} \quad (4)$$

$\bar{\mathbf{b}} = J^{-2/3} \mathbf{b}$  est la partie isochore du tenseur de Cauchy–Green gauche  $\mathbf{b} = \mathbf{F}\mathbf{F}^T$ .

Dans les exemples qui suivent, le comportement quasi-incompressible de type Neo-Hookéen est utilisé :

$$\Psi = \kappa(J \ln J - J + 1) + \frac{\mu}{2} (\bar{I}_1 - 3), \quad (5)$$

En grandes transformations, les conditions aux limites périodiques découlent directement de la discretisation du gradient du déplacement sur la cellule unitaire, soit :

$$\mathbf{u}(\mathbf{x}^+) - \mathbf{u}(\mathbf{x}^-) = \overline{\nabla u} (\mathbf{x}^+ - \mathbf{x}^-). \quad (6)$$

La résolution par éléments finis nécessite alors de résoudre un système non linéaire, traité avec un algorithme de Newton–Raphson, dans un cadre lagrangien mis à jour. Le modèle de comportement utilisé est implémenté dans la bibliothèque open-source simcoon, et le solveur non linéaire dans la bibliothèque éléments finis fedoo, tous deux appartenant au projet 3MAH [11]. Notons que dans la méthode EF, l'équilibre local (Éq. (1)) est imposé au sens faible, ce qui implique que la divergence des contraintes n'est pas exactement nulle. Cette divergence peut même être utilisée comme indicateur d'erreur d'approximation du calcul par éléments finis. Nous utilisons ici la forme forte de l'équilibre dans le cadre d'apprentissage, ce qui nécessite la définition d'un opérateur différentiel de divergence, en prenant certaines précautions.

## 2.1 Opérateur divergence

Nous définissons l'opérateur de divergence comme une matrice  $\mathbf{D}$  qui, appliquée aux valeurs nodales d'un champ vectoriel (ici 2D), permet le calcul de la divergence de ce champ en utilisant les contributions des dérivées des fonctions de forme à chaque nœud :

$$\mathbf{D}(\mathbf{x}) = (\mathbf{D}_x(\mathbf{x}) \quad \mathbf{D}_y(\mathbf{x})), \quad (7)$$

où  $\mathbf{D}_x$  et  $\mathbf{D}_y$  sont des matrices contenant respectivement les dérivées des fonctions de forme par rapport à  $x$  et  $y$  :

$$\mathbf{D}_x(\mathbf{x}) = \begin{pmatrix} \frac{\partial N_1(\mathbf{x})}{\partial x} & \frac{\partial N_2(\mathbf{x})}{\partial x} & \dots & \frac{\partial N_n(\mathbf{x})}{\partial x} \end{pmatrix} \quad (8)$$

et

$$\mathbf{D}_y(\mathbf{x}) = \begin{pmatrix} \frac{\partial N_1(\mathbf{x})}{\partial y} & \frac{\partial N_2(\mathbf{x})}{\partial y} & \dots & \frac{\partial N_n(\mathbf{x})}{\partial y} \end{pmatrix}. \quad (9)$$

En considérant  $\boldsymbol{\sigma}$  le vecteur des composantes nodales de contrainte, construit en empilant les valeurs nodales de chaque composante de contrainte :

$$\boldsymbol{\sigma} = \begin{pmatrix} [\sigma_{xx}] \\ [\sigma_{yy}] \\ [\sigma_{xy}] \end{pmatrix} \quad \text{et} \quad \mathbf{D}_\sigma(\mathbf{x}) = \begin{pmatrix} \mathbf{D}_x(\mathbf{x}) & 0 & \mathbf{D}_y(\mathbf{x}) \\ 0 & \mathbf{D}_y(\mathbf{x}) & \mathbf{D}_x(\mathbf{x}) \end{pmatrix}, \quad (10)$$

nous obtenons alors les valeurs de la divergence du champ de contraintes en chaque nœud d'un maillage

$$\text{div}(\boldsymbol{\sigma}(\mathbf{x})) = \mathbf{D}_\sigma(\mathbf{x}) \cdot \boldsymbol{\sigma}. \quad (11)$$

Notons que la divergence calculée avec cet opérateur n'est pas continue entre les éléments et le champ nodal de divergence utilisé dans ce travail est calculé en moyennant les valeurs des éléments adjacents.

## 2.2 Réseau de neurone à graphe

Le modèle d'apprentissage profond utilisé dans cette étude est dénommé réseau de neurones à graphe (Graph Neural Networks, ou GNN). Il utilise une représentation abstraite de données définies comme un ensemble de sommets (ou nœuds du graphe) noté  $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$  et un ensemble de paires non ordonnées de sommets noté  $\mathcal{E}$  appelé arêtes, représentant les connexions ou relations entre les sommets. Chaque arête  $e \in \mathcal{E}$  est un élément de la forme  $(v_i, v_j)$ . On peut associer à chaque nœud  $v_i \in \mathcal{V}$  un vecteur de caractéristiques  $\mathbf{v}_i \in \mathbb{R}^d$  de dimension  $d$ , et à chaque arête  $(v_i, v_j) \in \mathcal{E}$  un vecteur de caractéristiques  $\mathbf{e}_{ij} \in \mathbb{R}^k$  de dimension  $k$ . Les GNN reposent sur un mécanisme appelé couches de passage de message (Message Passing Layers, MPL), où chaque opération consiste à propager et agréger de l'information à travers les nœuds d'un graphe. L'objectif de ce mécanisme est de mettre à jour de manière itérative les caractéristiques de chaque nœud en échangeant des « messages » avec ses voisins. Une couche de passage de message est alors une opération locale réalisée sur un nœud donné  $v_i$ , prenant en compte l'information de son ensemble de voisins  $j \in \mathcal{N}(i)$ , potentiellement complétée par l'information portée par les arêtes

$e_{ij}$ . Ce mécanisme peut s'apparenter à une opération de convolution, cependant, les GNN ne sont pas restreints aux données structurées en grille comme c'est le cas des réseaux de neurones convolutifs (CNN). Les GNN peuvent en effet être appliqués à des structures de données très générales, pouvant être représentées comme des graphes. Les maillages de modèles par éléments finis sont particulièrement adaptés à une représentation en graphe, puisqu'ils partagent des caractéristiques structurales : ils impliquent tous deux des nœuds (sommets) et des connexions (arêtes). Ce modèle est appliqué comme un opérateur de localisation, permettant de reconstruire le champ de contraintes local  $\sigma$  d'une microstructure périodique (représenté par un maillage périodique  $\mathcal{M}$ ), en ne disposant que de la réponse mécanique moyenne  $\bar{\sigma}$ . On réplique alors un opérateur  $\mathcal{B}(r,t)$  tel que :

$$\sigma(r,t) = \mathcal{B}(r,t)\bar{\sigma}(t), \quad (12)$$

où  $r$  est un vecteur position et  $t$  le temps. Les caractéristiques des nœuds et des arêtes du graphe d'entrée sont projetées dans une représentation latente à l'aide d'un encodeur de type Multi-Layer Perceptron (MLP). Le graphe latent obtenu est ensuite traité par un cadre GNN à passage de message, puis le graphe latent est décodé par un décodeur MLP, produisant en sortie les valeurs locales de contrainte. Cette architecture GNN, communément appelée *MeshGraphNet*, a été appliquée avec succès à des prédictions de champs scalaires sur des données basées sur des maillages[?, ?, ?], et représente l'état de l'art dans ce domaine. Comme notre méthode d'apprentissage repose sur la comparaison avec une simulation numérique servant de vérité terrain, nous rappelons ici que l'équilibre mécanique est satisfait au travers de la formulation faible, c'est-à-dire en considérant l'intégrale volumique du domaine. Nous proposons ici de compléter la fonction de perte par l'ajout d'une information portant sur la formulation forte afin de mieux décrire la condition d'équilibre statique local décrite dans la Section 2. Le modèle qui implémente cette perte informée par la physique est désigné comme *P-DivGNN*.

### 2.3 Formulation forte de l'équilibre dans la fonction de perte

Pour guider le processus d'entraînement, nous introduisons une fonction de perte informée par la physique qui impose la condition d'équilibre décrite dans la Section 2.1. La fonction de perte totale est composée de deux termes : (1) une erreur quadratique moyenne normalisée (NMSE) entre les prédictions du GNN et la vérité terrain, et (2) un terme de régularisation basé sur la physique qui pénalise les écarts à la condition d'équilibre statique, en négligeant les forces de volume  $\text{div} \sigma = 0$ . Soit  $\sigma$  le champ de contraintes de référence obtenu à partir des simulations par éléments finis,  $\hat{\sigma}$  le champ de contraintes prédit par le GNN, et  $n$  le nombre total de nœuds du maillage  $\mathcal{M}$ . La fonction de perte est définie comme :

$$\mathcal{L}(\sigma, \hat{\sigma}) = \text{NMSE}(\sigma, \hat{\sigma}) + \lambda (\mathbf{div}(\hat{\sigma}))^2. \quad (13)$$

Le terme informé par la physique  $(\mathbf{div}(\hat{\sigma}))^2$  est calculé en utilisant l'opérateur de divergence discret et est défini comme la norme quadratique moyenne de la divergence sur tous les indices  $i$  :

$$(\mathbf{div}(\hat{\sigma}))^2 = \frac{1}{n} \sum_{i=1}^n \|\mathbf{div}(\hat{\sigma})|_{\mathbf{x}=\mathbf{x}_i}\|^2, \quad (14)$$

La NMSE est utilisée à la place de l'erreur quadratique moyenne (MSE) afin de garantir une importance égale pour chaque composante de contrainte,  $c \in \{xx, yy, xy\}$ . Elle est définie comme :

$$\text{NMSE}(\sigma, \hat{\sigma}) = \frac{1}{n_c} \sum_{c=1}^{n_c} \frac{\sum_{i=1}^n (\sigma_{c,i} - \hat{\sigma}_{c,i})^2}{\sum_{i=1}^n (\sigma_{c,i} - \frac{1}{n} \sum_{j=1}^n \sigma_{c,j})^2}, \quad (15)$$

La NMSE garantit que toutes les composantes de contrainte contribuent de manière équivalente à la perte d'entraînement, et le terme de régularisation basé sur la physique encourage davantage le GNN à satisfaire la condition d'équilibre.

### 3 Résultats - comportement hyperélastique

#### 3.1 Génération du jeu de données

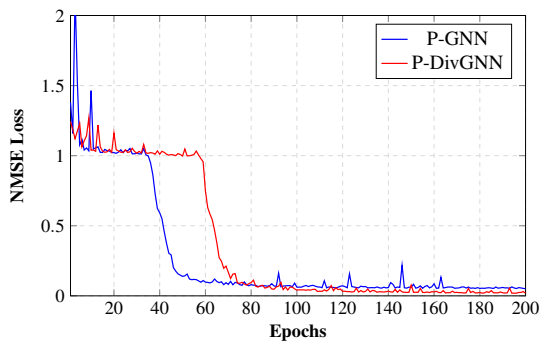
Un total de 9795 maillages périodiques bidimensionnels d'une plaque carrée perforée, présentant une variabilité dans la géométrie du trou et dans le raffinement du maillage, a été généré à l'aide de la même procédure d'échantillonnage aléatoire. Le jeu de données est séparé en un ensemble d'entraînement (70%) et un ensemble de test (30%). Tous les résultats présentés dans cette étude sont issus de l'ensemble de test. Les simulations par Éléments Finis ont été réalisées sous conditions périodiques imposées sur les bords de la plaque carrée en déformation plane. Afin de couvrir une gamme plus large de réponses mécaniques, les composantes moyennes de la déformation logarithmique dans le plan,  $\bar{h}_{xx}$ ,  $\bar{h}_{yy}$  et  $\bar{h}_{xy}$ , ont été échantillonnées uniformément dans l'intervalle  $[-0.15, 0.15]$ . Dans ce modèle non linéaire,  $\bar{h}$  correspond au tenseur de déformation logarithmique. Comme indiqué dans l'équation (6), pour les problèmes en grandes transformations, les conditions périodiques sont exprimées à partir du gradient de déplacement  $\bar{\nabla}u$ . Les gradients de déplacement sont obtenus à partir des tenseurs de déformation logarithmique échantillonnés en supposant une décomposition polaire de  $\mathbf{F}$  et en imposant une matrice de rotation identité (c'est-à-dire sans rotation de corps rigide). Cela donne :

$$\bar{\nabla}u = \mathbf{F} - \mathbf{1} = \exp \bar{\epsilon} - \mathbf{1} \quad (16)$$

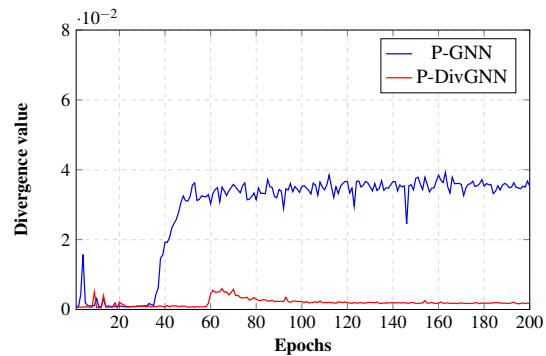
Le comportement constitutif est modélisé à l'aide de la loi hyperélastique de Neo-Hooke, telle que définie dans la Section 2 avec les paramètres suivants : module de cisaillement  $\mu = 3.0$  MPa et module de compressibilité volumique  $\kappa = 10$  MPa.

#### 3.2 Entraînement du modèle

Pour évaluer le comportement de convergence des modèles dans le cadre hyperélastique non linéaire, les modèles *P-GNN* et *P-DivGNN* sont tous deux entraînés pendant 200 époques à l'aide de l'optimiseur Adam, un taux d'apprentissage de  $10^{-2}$  et des taille de lots (batch size) de 16 graphes. La principale différence dans ce cas réside dans le choix du coefficient de régularisation physique  $\lambda$ . En raison des échelles de contraintes différentes présentes dans le jeu de données hyperélastique par rapport au cas linéaire, le poids de régularisation est ajusté à  $\lambda = 10^{-2}$  afin de conserver un équilibre approprié entre le terme de fidélité aux données et le terme physique dans la fonction de perte.



(a) Comparaison de la convergence de la perte NMSE sur l'ensemble de validation lorsque la pénalisation de divergence est appliquée durant l'entraînement



(b) Divergence evolution in validation dataset during training

FIGURE 1 – Comparaison de la perte NMSE et de l'évolution de la divergence durant l'entraînement du modèle *P-GNN* (en bleu) et du modèle *P-DivGNN* (en rouge) sur le jeu de données hyperélastique.

L'évolution de la perte NMSE durant l'entraînement sur le jeu de données hyperélastique non linéaire est présentée dans la Figure 1a. Le modèle *P-GNN* présente une diminution rapide de la NMSE autour de l'époque 40, tandis que *P-DivGNN* montre une réduction plus tardive mais plus progressive, devenant notable vers l'époque 60. Malgré une convergence initiale plus lente, *P-DivGNN* atteint finalement des valeurs de NMSE légèrement plus faibles, suggérant que l'ajout du terme de régularisation fondé sur la

physique améliore la capacité du modèle à reconstruire les champs de contraintes. La Figure 1b illustre l'évolution de la divergence des champs de contraintes prédits au cours de l'entraînement. Comme dans le cas linéaire, les valeurs de divergence sont initialement proches de zéro en raison de l'application uniforme du champ de contraintes moyen sur tous les nœuds du graphe. Cependant, dans ce scénario non linéaire, les profils de divergence des deux modèles se distinguent davantage à mesure que l'entraînement progresse. En particulier, *P-DivGNN* maintient systématiquement des valeurs de divergence plus faibles que *P-GNN*, indiquant une meilleure conformité aux contraintes physiques sous-jacentes. Cette différence est plus marquée que dans le cas linéaire et met en évidence l'intérêt de la pénalisation de la divergence, notamment dans des régimes plus complexes où des non-linéarités géométriques sont présentes.

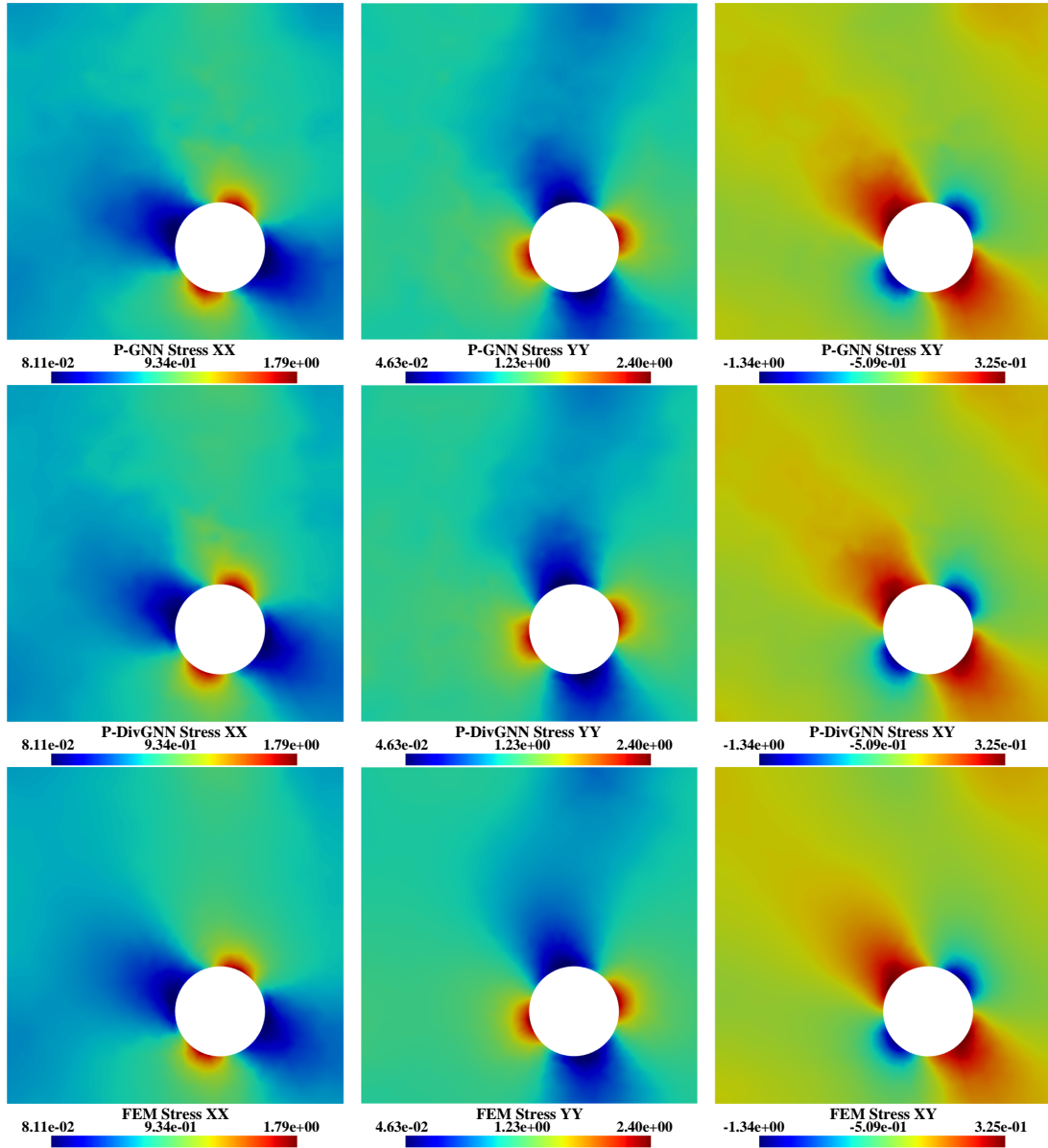


FIGURE 2 – Comparaison des reconstructions du champ de contraintes local pour *P-GNN* (ligne du haut), *P-DivGNN* (ligne du milieu) et FEM (ligne du bas) pour les trois composantes de contrainte  $(\sigma_{xx}, \sigma_{yy}, \sigma_{xy})$ . Contraintes moyennes en entrée :  $\bar{\sigma}_{xx} = 0.59$  MPa,  $\bar{\sigma}_{yy} = -1.01$  MPa,  $\bar{\sigma}_{xy} = 0.33$  MPa. L'échelle de la barre de couleur est définie pour correspondre à la plage de la solution FE.

La Figure 2 présente un exemple représentatif issu du jeu de test hyperélastique, avec une prédiction des champs de contraintes (composantes planes). Les champs d'erreur correspondantes (erreur sur la norme du vecteur divergence) au niveau des nœuds sont représenté en figure 3. Les résultats indiquent que *P-GNN* produit des écarts à l'équilibre locaux plus faibles que *P-DivGNN*, en particulier dans les zones où ce dernier génère des prédictions de contraintes plus irrégulières ou bruitées.

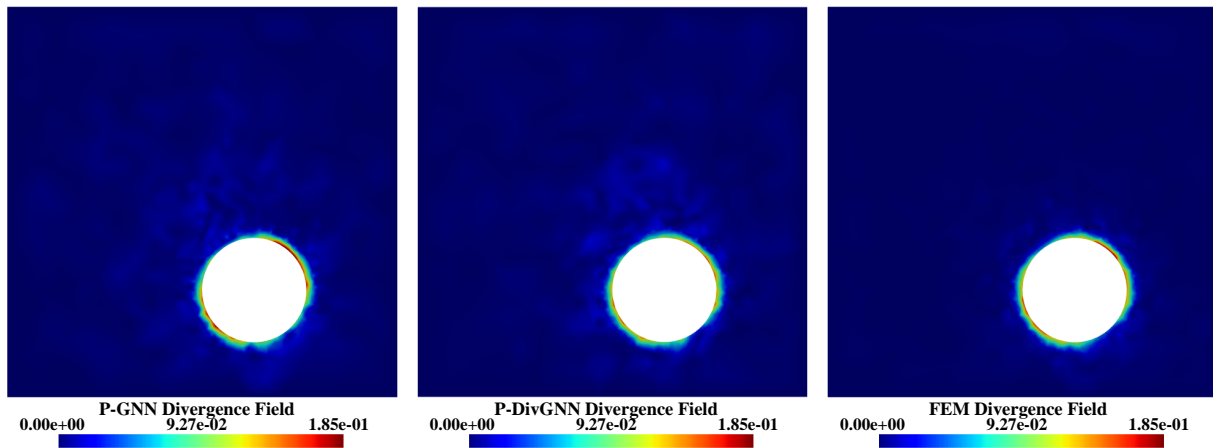


FIGURE 3 – Comparison of norm of divergence field of local stress field between  $P$ -GNN,  $P$ -DivGNN and FEM predictions for a non-linear hyperelastic case.

TABLE 1 – Comparaison des valeurs moyenne de la partie erreur en divergence et NMSE sur tout le jeu de données

	<b>Divergence</b>	<b>NMSE</b>
P-GNN	$3.55 \times 10^{-2}$	$5.36 \times 10^{-2}$
<b>P-DivGNN</b>	<b><math>1.56 \times 10^{-3}</math></b>	<b><math>2.18 \times 10^{-2}</math></b>
FEM	$1.69 \times 10^{-3}$	–

Le Tableau 1 présente une comparaison quantitative de la divergence moyenne et de l’erreur quadratique moyenne normalisée (NMSE) sur l’ensemble des données de test. Les résultats confirment l’avantage d’intégrer une pénalisation de divergence nulle durant l’entraînement. Le modèle proposé  $P$ -DivGNN atteint la plus faible valeur de divergence, surpassant à la fois le modèle de base  $P$ -GNN et même la solution FEM. Il s’agit d’un résultat notable, car il indique que le modèle n’approche pas seulement la vérité de terrain, mais applique également la contrainte physique d’équilibre local de manière plus efficace que la méthode des éléments finis, à nombre de degrés de libertés.

Ces métriques apportent une information importante sur la robustesse du modèle d’apprentissage automatique proposé, et en particulier de la formulation de la fonction de perte tenant compte de l’équilibre local.

## 4 Conclusion et perspectives

Le cadre d’apprentissage développé permet de reconstruire des champs de contraintes locaux à partir de valeurs moyennes de contraintes homogénéisées et agit comme un opérateur de localisation non-linéaire. Un modèle hyperélastique non-linéaire a permis d’évaluer les performances de ce modèle, donc sa capacité à reconstruire des champs locaux avec précision, sur une large gamme de géométries (position et rayon du trou percé), en considérant différents raffinements de maillage. Les résultats montrent que le cadre développé permet de représenter avec une bonne précision les champs locaux, en apportant un facteur d’accélération de l’ordre de 500. Ces résultats doivent tenir compte des différences d’architecture (CPU vs GPU) mais permettent cependant d’apprécier les bénéfices de ce type de modèles. Notons que les simulations par éléments finis dans un contexte non-linéaire utilisent un schéma itératif (typiquement de Newton–Raphson), entraînant un coût de calcul non négligeable. À l’inverse, le modèle GNN proposé reconstruit directement le champ local de contraintes à partir du champ moyen de contraintes à l’échelle de la RVE, sans recourir à des solveurs itératifs.

La prise en compte des équations d’équilibre locales dans l’apprentissage permet d’obtenir des résultats plus précis et plus cohérents, et outrepassent même la solution de référence par éléments finis, ce qui renforce la nécessité d’intégrer des contraintes physiques dans les démarches d’apprentissage

automatique.

Le code source permettant de générer les jeux de données utilisés et de reproduire les expériences présentées dans cet article est disponible à l'adresse suivante :

<https://github.com/ricardo0115/p-div-gnn>

## Références

- [1] Feyel, F. (1999). Multiscale FE2 elastoviscoplastic analysis of composite structures. *\*Computational Materials Science\**, 16(1), 344–354.
- [2] Feyel, F. (2003). A multilevel FE2 method for nonlinear structures. *\*CMAME\**, 192(28), 3233–3244.
- [3] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *\*Proceedings of the IEEE\**, 86(11), 2278–2324.
- [4] Nie, Z., Jiang, H., & Kara, L. B. (2019). Stress Field Prediction in Cantilevered Structures Using CNNs. *\*Journal of Computing and Information Science in Engineering\**, 20(1), 011002.
- [5] Gupta, A., Bhaduri, A., & Graham-Brady, L. (2023). Accelerated multiscale mechanics modeling in a deep learning framework. *\*Mechanics of Materials\**, 184, 104709.
- [6] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net. In *\*MICCAI\**.
- [7] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The Graph Neural Network Model. *\*IEEE Transactions on Neural Networks\**, 20(1), 61–80.
- [8] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks. *\*JCP\**, 378, 686–707.
- [9] Hernández, Q., Badías, A., Chinesta, F., & Cueto, E. (2024). Thermodynamics-Informed Graph Neural Networks. *\*IEEE Transactions on Artificial Intelligence\**, 5(3), 967–976.
- [10] Richter-Powell, J., Lipman, Y., & Chen, R. T. Q. (2022). Neural Conservation Laws : A Divergence-Free Perspective. In *\*NeurIPS 2022\**.
- [11] Prulière, E., & Chemisky, Y. (2023). 3MAH : Un ensemble de bibliothèques pour analyser le comportement complexe de matériaux hétérogènes. In *\*Colloque National en Calcul des Structures (CSMA)\**.
- [12] Maurizi, M., Gao, C., & Berto, F. (2022). Predicting stress/strain fields with GNNs. *\*Scientific Reports\**, 12.
- [13] Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., & Battaglia, P. W. (2021). Learning Mesh-Based Simulation with Graph Networks. In *\*ICLR 2021\**.
- [14] Fortunato, M., Pfaff, T., Wirnsberger, P., Pritzel, A., & Battaglia, P. W. (2022). MultiScale MeshGraphNets. In *\*AI4Science Workshop, ICML 2022\**.