

Résolution de l'Équation de Poisson par Méthodes Hybrides : Réseaux Neuronaux Informés par la Physique et Apprentissage Supervisé

Idriss MORTADI
idriss.mortadi@student-cs.fr
 [Github Repository](#)

Avril 2025

Abstract

Ce rapport présente une comparaison entre deux approches pour résoudre l'équation de Poisson : les Réseaux Neuronaux Informés par la Physique (Physics-Informed Neural Networks, PINNs) et l'apprentissage supervisé traditionnel. Les PINNs intègrent directement les équations différentielles dans la fonction de perte du réseau, tandis que l'apprentissage supervisé utilise des solutions numériques préexistantes comme données d'entraînement. Nous analysons les performances de ces deux méthodes et discutons de leurs avantages et inconvénients respectifs.

Contents

1	Introduction	2
2	Formulation du Problème	2
3	Génération de données	2
3.1	Points pour le PINN	3
3.2	Points pour l'apprentissage supervisé	3
4	Méthodes	3
4.1	Réseau Neuronal Informé par la Physique (PINN)	3
4.2	Apprentissage Supervisé avec Solution de Référence	4
4.3	Méthode des Différences Finies	4
4.4	Environnement de Calcul	4
5	Résultats	4
5.1	Solution PINN	4
5.2	Solution par Différences Finies	6
5.3	Comparaison PINN vs Différences Finies	6
5.4	Solution par Apprentissage Supervisé	7
5.5	Comparaison Apprentissage Supervisé vs Différences Finies	9
6	Comparaison des Méthodes	10
6.1	Discussion	10
7	Conclusion et Perspectives	12
7.1	Conclusion	12
7.2	Perspectives	13

1 Introduction

L'équation de Poisson est une équation aux dérivées partielles (EDP) fondamentale qui apparaît dans de nombreux domaines des sciences et de l'ingénierie, notamment l'électromagnétisme, la mécanique des fluides, et la diffusion de chaleur. Dans sa forme la plus simple, elle s'écrit :

$$\nabla^2 u + f = 0 \quad (1)$$

où ∇^2 est l'opérateur laplacien, u est la fonction inconnue, et f est un terme source connu.

Traditionnellement, cette équation est résolue numériquement par des méthodes telles que les différences finies, les éléments finis ou les volumes finis. Cependant, ces dernières années, les approches basées sur l'apprentissage automatique ont gagné en popularité pour la résolution d'EDPs.

Dans ce rapport, nous comparons deux approches d'apprentissage automatique pour résoudre l'équation de Poisson :

1. **Réseaux Neuronaux Informés par la Physique (PINNs)** : Ces réseaux incorporent directement les équations différentielles dans leur fonction de perte, permettant un apprentissage sans données de solution.
2. **Apprentissage Supervisé Classique** : Cette approche utilise des solutions numériques préexistantes (obtenues par différences finies dans notre cas) comme données d'entraînement.

Notations

- ∇^2 : opérateur laplacien.
- u : fonction inconnue (solution de l'équation de Poisson).
- f : terme source.
- Ω : domaine de calcul.
- $\partial\Omega$: frontière du domaine.

2 Formulation du Problème

Nous considérons l'équation de Poisson bidimensionnelle sur le domaine carré unitaire $\Omega = [0, 1]^2$ avec des conditions aux limites de Dirichlet homogènes :

$$\nabla^2 u(x, y) + f(x, y) = 0, \quad (x, y) \in \Omega \quad (2)$$

$$u(x, y) = 0, \quad (x, y) \in \partial\Omega \quad (3)$$

Le terme source $f(x, y)$ est défini comme :

$$f(x, y) = x \sin(a\pi y) + y \sin(b\pi x) \quad (4)$$

avec $a = 0.3$ et $b = 0.5$.

3 Génération de données

La génération de données est une étape cruciale pour les deux approches étudiées. Pour le PINN et les différences finies, nous générons une grille uniforme de points sur le domaine $\Omega = [0, 1]^2$ avec n points dans chaque dimension, résultant en $n \times n$ points au total. Pour notre étude, nous avons utilisé $n = 100$, ce qui fournit une résolution suffisante pour capturer la dynamique de la solution.

3.1 Points pour le PINN

Pour l'entraînement du PINN, nous séparons les points de la grille en deux catégories :

- **Points intérieurs** : Ces points, situés à l'intérieur du domaine Ω , sont utilisés pour calculer le terme de perte associé à l'équation de Poisson ($\mathcal{L}_{\text{équation}}$).
- **Points frontières** : Ces points, situés sur le bord du domaine $\partial\Omega$, sont utilisés pour le terme de perte des conditions aux limites ($\mathcal{L}_{\text{limite}}$).

Pour les points intérieurs, nous calculons directement le résidu de l'équation différentielle $\nabla^2 u + f = 0$ en utilisant la différentiation automatique de PyTorch, qui permet d'obtenir précisément les dérivées partielles de la sortie du réseau par rapport à ses entrées.

3.2 Points pour l'apprentissage supervisé

Pour l'entraînement du modèle supervisé, nous utilisons la solution par différences finies comme référence. Les données générées sont organisées comme suit :

- **Entrées** : Coordonnées (x, y) des points sur la grille.
- **Sorties cibles** : Valeurs de la solution $u(x, y)$ calculées par la méthode des différences finies.

Ces données sont ensuite divisées en trois ensembles :

- Ensemble d'entraînement (80% des données) : Utilisé pour ajuster les paramètres du réseau.
- Ensemble de validation (10% des données) : Utilisé pour surveiller la généralisation pendant l'entraînement.
- Ensemble de test (10% des données) : Utilisé pour l'évaluation finale des performances.

Le temps nécessaire pour générer les données est négligeable (environ 0.0004 secondes) comparé au temps d'entraînement des modèles, mais cette étape est fondamentale pour l'approche supervisée qui repose entièrement sur ces données pré-calculées.

4 Méthodes

4.1 Réseau Neuronal Informé par la Physique (PINN)

Un PINN consiste en un réseau neuronal profond entraîné à satisfaire l'équation différentielle et les conditions aux limites. La fonction de perte comprend deux termes :

$$\mathcal{L} = \mathcal{L}_{\text{équation}} + \mathcal{L}_{\text{limite}} \quad (5)$$

où $\mathcal{L}_{\text{équation}}$ représente l'erreur quadratique moyenne du résidu de l'équation différentielle, et $\mathcal{L}_{\text{limite}}$ représente l'erreur quadratique moyenne sur les conditions aux limites.

Pour l'équation de Poisson :

$$\mathcal{L}_{\text{équation}} = \frac{1}{N_{\text{int}}} \sum_{i=1}^{N_{\text{int}}} \left| \nabla^2 u(x_i, y_i) + f(x_i, y_i) \right|^2 \quad (6)$$

$$\mathcal{L}_{\text{limite}} = \frac{1}{N_{\text{bd}}} \sum_{j=1}^{N_{\text{bd}}} |u(x_j, y_j)|^2 \quad (7)$$

où (x_i, y_i) sont des points à l'intérieur du domaine et (x_j, y_j) sont des points sur la frontière.

L'architecture du réseau PINN utilisée dans cette étude est constituée de quatre couches entièrement connectées avec fonction d'activation tangente hyperbolique :

- Couche d'entrée : 2 neurones (pour les coordonnées x et y)
- Couches cachées : 3 couches de 64 neurones chacune avec activation tanh
- Couche de sortie : 1 neurone (pour la valeur de u)

La dérivation des gradients nécessaires pour calculer le laplacien est effectuée par différentiation automatique grâce à PyTorch.

4.2 Apprentissage Supervisé avec Solution de Référence

L'approche supervisée utilise une solution numérique obtenue par différences finies comme données d'entraînement. Le réseau apprend simplement une correspondance entre les coordonnées (x, y) et les valeurs de la solution $u(x, y)$ à ces points.

Nous utilisons une architecture de réseau identique à celle du PINN mais avec une fonction de perte différente, à savoir l'erreur quadratique moyenne classique :

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=1}^N |u_{pred}(x_i, y_i) - u_{ref}(x_i, y_i)|^2 \quad (8)$$

Les données sont divisées en ensembles d'entraînement (80%), de validation (10%) et de test (10%).

4.3 Méthode des Différences Finies

Pour obtenir une solution de référence, nous utilisons la méthode des différences finies. L'opérateur laplacien $\nabla^2 u$ est discrétisé en utilisant le schéma à 5 points :

$$\nabla^2 u(x_i, y_j) \approx \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2} \quad (9)$$

où h est le pas de discrétisation dans les deux directions. La solution est obtenue en résolvant le système linéaire résultant.

4.4 Environnement de Calcul

Les expériences ont été réalisées sur la configuration matérielle suivante :

- GPU : NVIDIA GeForce RTX 3050 (4096 MiB de mémoire)
- Driver NVIDIA : version 570.86.16
- CUDA : version 12.8

Tous les modèles ont été implémentés avec PyTorch et entraînés sur le même environnement matériel pour assurer une comparaison équitable des performances de calcul.

5 Résultats

5.1 Solution PINN

Les figures 1a et 1b montrent la solution obtenue par le réseau PINN après entraînement.

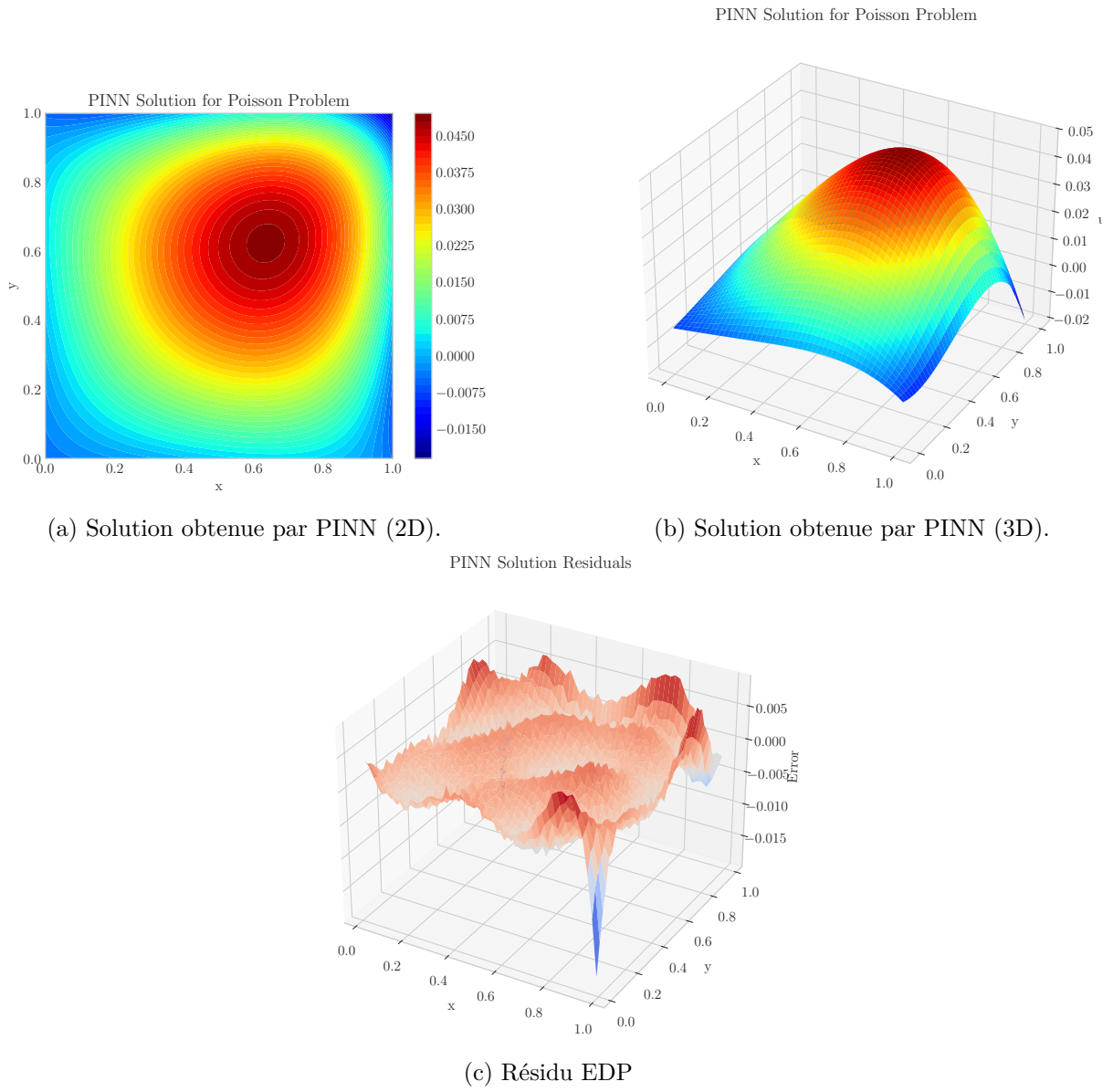


Figure 1: Solutions obtenues par PINN après 10 000 itérations : (a) vue 2D, (b) représentation 3D et (c) résidu de l'EDP.

La figure 2 montre l'évolution de la perte pendant l'entraînement du PINN.

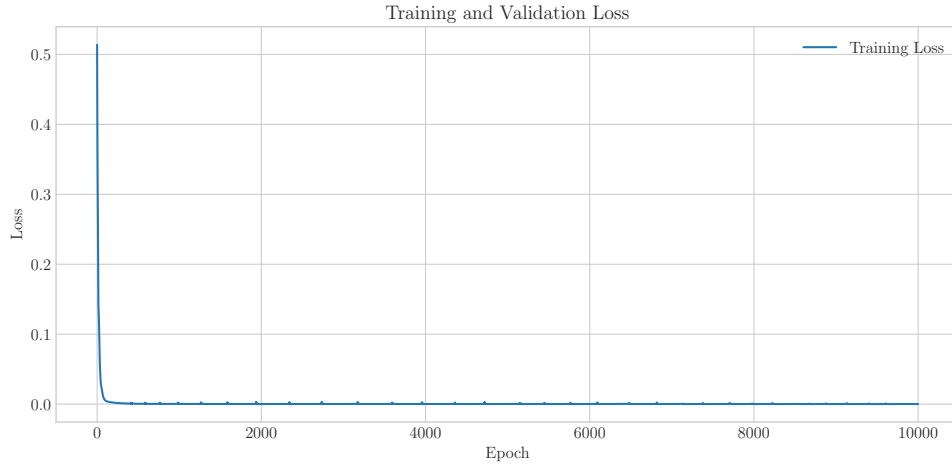
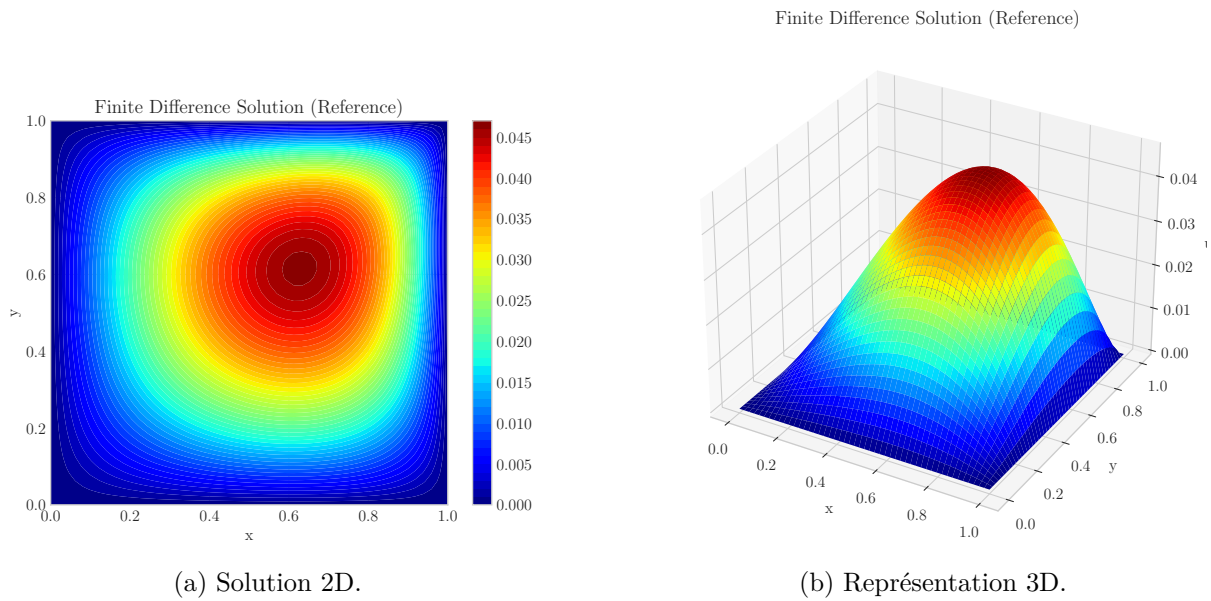


Figure 2: Évolution de la perte pendant l'entraînement du PINN.

5.2 Solution par Différences Finies

Les figures 3a et 3b présentent la solution de référence obtenue par la méthode des différences finies.



(a) Solution 2D.

(b) Représentation 3D.

Figure 3: Solutions obtenues par la méthode des différences finies : (a) vue 2D et (b) représentation 3D.

5.3 Comparaison PINN vs Différences Finies

Les figures 4a et 4b montrent l'erreur entre la solution PINN et la solution par différences finies.

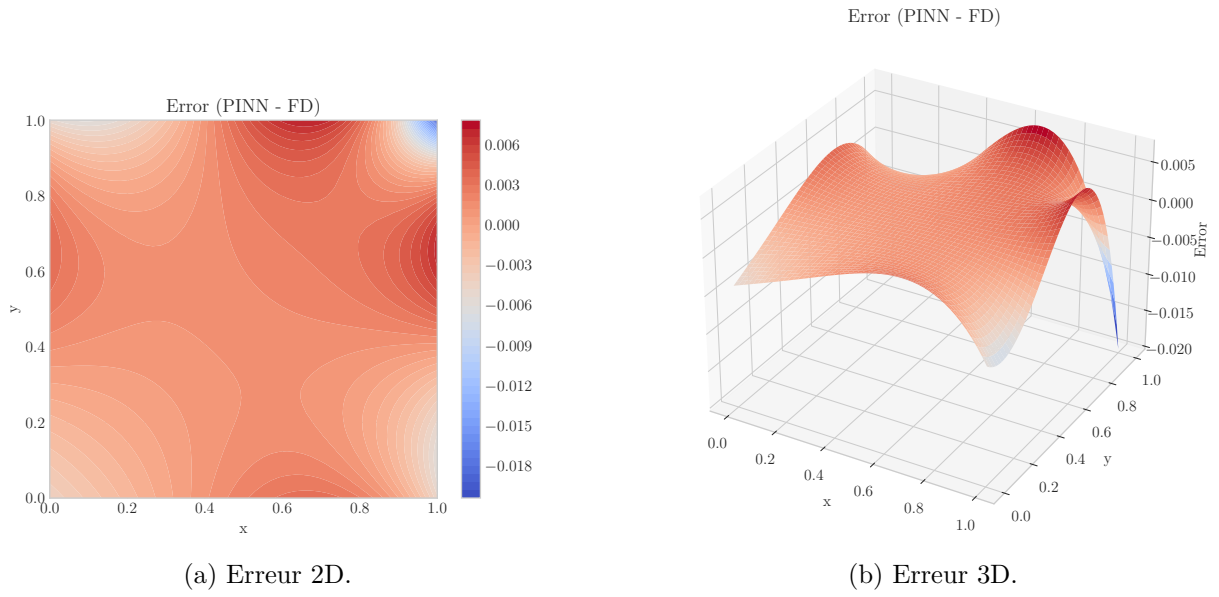


Figure 4: Erreur entre la solution PINN et la solution par différences finies : (a) vue 2D et (b) représentation 3D.

Métriques d'erreur pour PINN:

- Erreur quadratique moyenne (MSE): $6.114357e - 06$
- Erreur L2 relative: 0.106320

Commentaires sur les résultats PINN

- La solution PINN n'arrive pas à reproduire la solution de référence sur les bordures du domaine.
- On peut se demander si l'architecture du réseau est suffisante pour capturer la complexité de la solution. Ou si on augmente le coefficient dans la fonction de perte on peut espérer que le modèle approxime bien la solution sur les bordures.

5.4 Solution par Apprentissage Supervisé

Les figures 5a, 5b et 5c montrent respectivement la solution obtenue par le réseau supervisé en vue 2D, en représentation 3D, et le résidu de l'EDP.

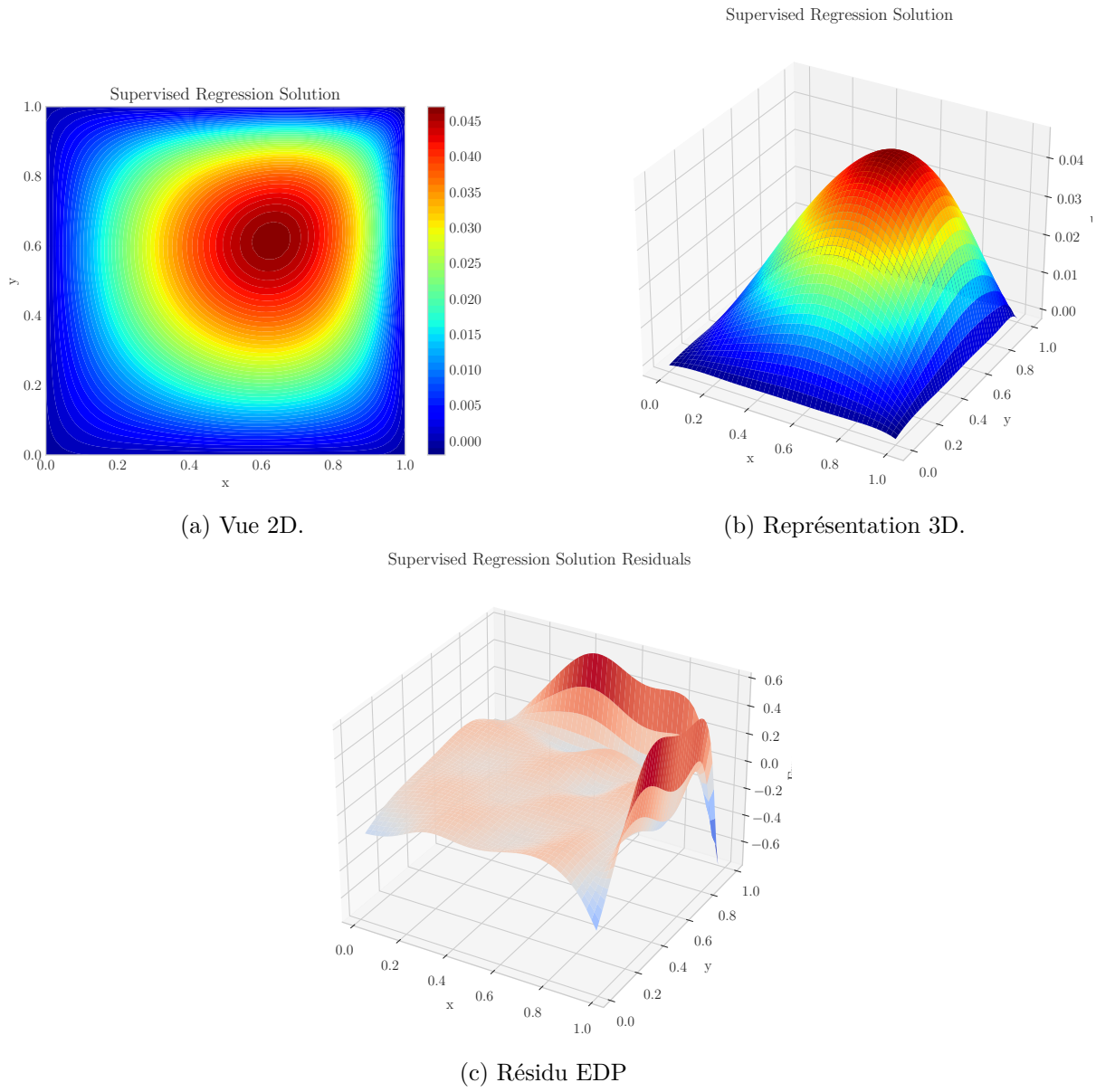


Figure 5: Solutions obtenues par apprentissage supervisé : (a) vue 2D, (b) représentation 3D et (c) résidu de l'EDP.

La figure 6 montre l'évolution des pertes d'entraînement et de validation pendant l'apprentissage supervisé.

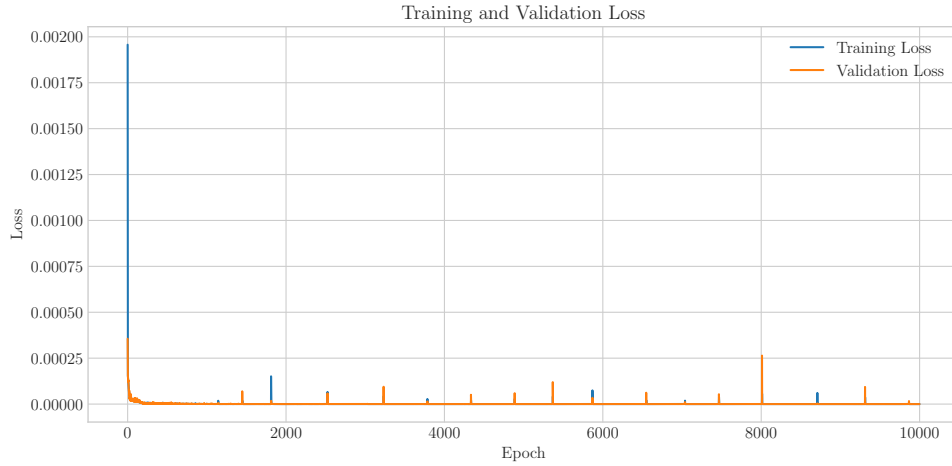


Figure 6: Évolution des pertes pendant l'entraînement du modèle supervisé.

5.5 Comparaison Apprentissage Supervisé vs Différences Finies

Les figures 7a et 7b montrent l'erreur entre la solution par apprentissage supervisé et la solution par différences finies.

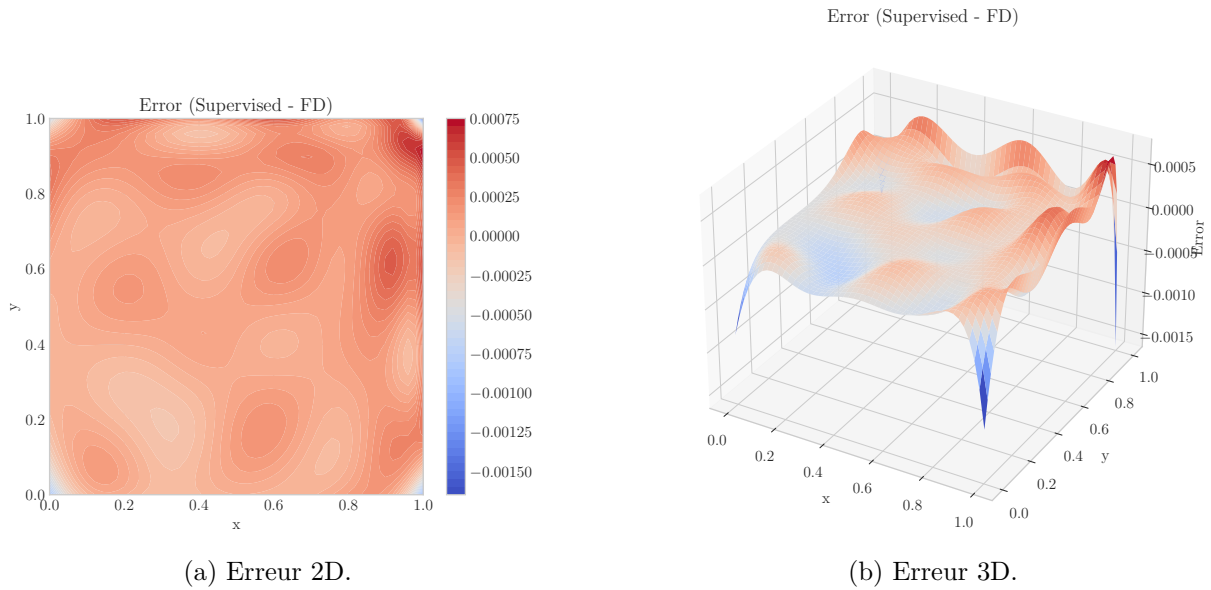


Figure 7: Erreur entre apprentissage supervisé et différences finies : (a) vue 2D et (b) représentation 3D.

Métriques d'erreur pour l'apprentissage supervisé:

- Sur la grille complète:
 - Erreur quadratique moyenne (MSE): $2.867895e - 08$
 - Erreur L2 relative: 0.007281
- Sur l'ensemble de test:
 - Erreur quadratique moyenne (MSE): $2.916618e - 08$
 - Erreur L2 relative: 0.006990

Commentaires sur les résultats de l'apprentissage supervisé

- La méthode de régression supervisée utilisant un réseau de neurones présente une erreur faible par rapport à la méthode précédente.
- On remarque aussi une bonne généralisation sur l'ensemble de test.
- Il est important de noter que les performances peuvent varier en fonction des hyperparamètres choisis.
- Le choix de la taille de batch (128) peut influencer la stabilité et la vitesse de convergence lors de l'entraînement.
- On peut également envisager d'optimiser l'architecture du réseau pour améliorer les résultats.

6 Comparaison des Méthodes

Méthode	MSE	Erreur L2 Relative
PINN vs. Différences Finies	6.11×10^{-6}	0.1063
Supervisé vs. Différences Finies	2.87×10^{-8}	0.0073
Supervisé (ensemble de test)	2.92×10^{-8}	0.0070

Table 1: Comparaison des métriques d'erreur entre les différentes approches.

Méthode	Génération de données (s)	Entraînement (s)	Inférence (ms)
PINN	0.0004	94.11	0.33
Supervisé	0.0004	198.57	0.40
Différences Finies	-	-	4.77

Table 2: Comparaison des temps d'exécution (en secondes pour l'entraînement et millisecondes pour l'inférence).

Paramètre	PINN	Supervisé
Architecture	2-64-64-64-1	2-64-64-64-1
Taux d'apprentissage	0.001	0.001
Nombre d'époques	10000	10000
Taille de batch	-	128
Grille de discrétisation	100×100	100×100

Table 3: Paramètres d'entraînement des modèles.

6.1 Discussion

Les tableaux 1 et 2 révèlent des différences significatives entre les approches étudiées.

La méthode d'apprentissage supervisé montre clairement une meilleure précision avec une erreur MSE de 2.87×10^{-8} et une erreur L2 relative de 0.0073 par rapport à la solution par différences finies. Cette performance supérieure est attendue puisque le modèle supervisé est

directement entraîné sur les données générées par la méthode des différences finies. La cohérence entre les erreurs sur l'ensemble complet (2.87×10^{-8}) et sur l'ensemble de test (2.92×10^{-8}) démontre également une bonne capacité de généralisation. Toutefois, cet avantage en précision s'accompagne d'un coût computationnel plus élevé, avec un temps d'entraînement de 198.57 secondes, soit plus du double que celui du PINN.

Le PINN, en revanche, présente une erreur MSE de 6.11×10^{-6} et une erreur L2 relative de 0.1063, environ 15 fois supérieures à celles de l'approche supervisée. De plus, il présente une erreur flagrante sur les bords du domaine. Une solution possible est d'augmenter le coefficient du terme de perte des bords, ce qui améliore les résultats (voir figure 8). Cependant, il est étonnant de constater que le résidu augmente dans la figure 9. En regardant l'échelle, on passe de résidus d'ordre 0.0075 à 0.02. Cette différence s'explique par la nature de son apprentissage, qui repose uniquement sur la minimisation du résidu de l'équation différentielle et des conditions aux limites, sans accès direct à la solution de référence. Malgré cette précision moindre, le PINN offre un avantage considérable : il n'a pas besoin de solution préalable pour son entraînement et s'exécute plus rapidement (94.11 secondes).

La solution PINN a un résidu plus faible comparé à la méthode d'apprentissage supervisé sur certaines régions du domaine, ce qui indique une meilleure satisfaction locale de l'équation différentielle. Cependant, cette performance est obtenue au détriment de la précision globale, notamment sur les bordures du domaine. Cela souligne l'importance de choisir des coefficients appropriés dans la fonction de perte pour équilibrer les différents termes.

En conclusion, bien que le PINN soit moins précis que l'approche supervisée, il reste une méthode prometteuse pour résoudre des problèmes où les données de solution ne sont pas disponibles. Des améliorations futures pourraient inclure l'utilisation de techniques d'optimisation avancées, des architectures de réseau plus adaptées ou des stratégies hybrides combinant les avantages des PINNs et de l'apprentissage supervisé.

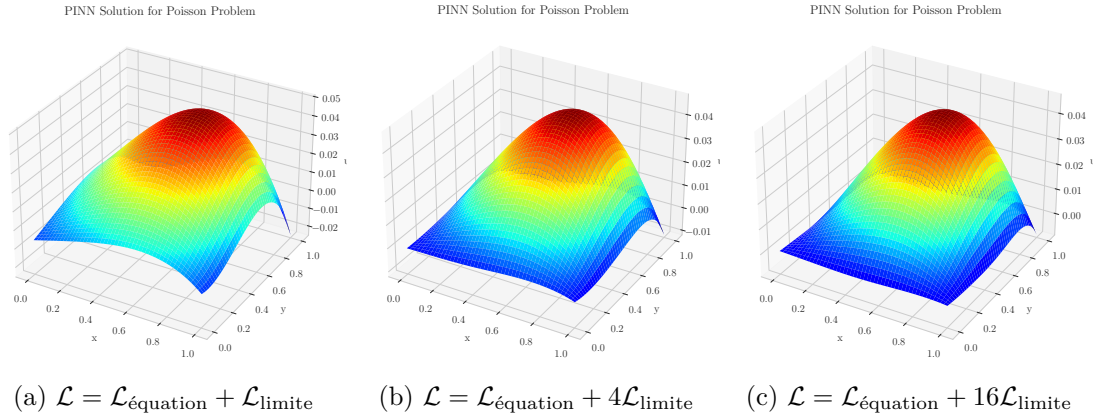


Figure 8: Comparaison des solutions obtenues par PINN (10000 itérations) avec différents coefficients pour le terme de perte des conditions aux limites.

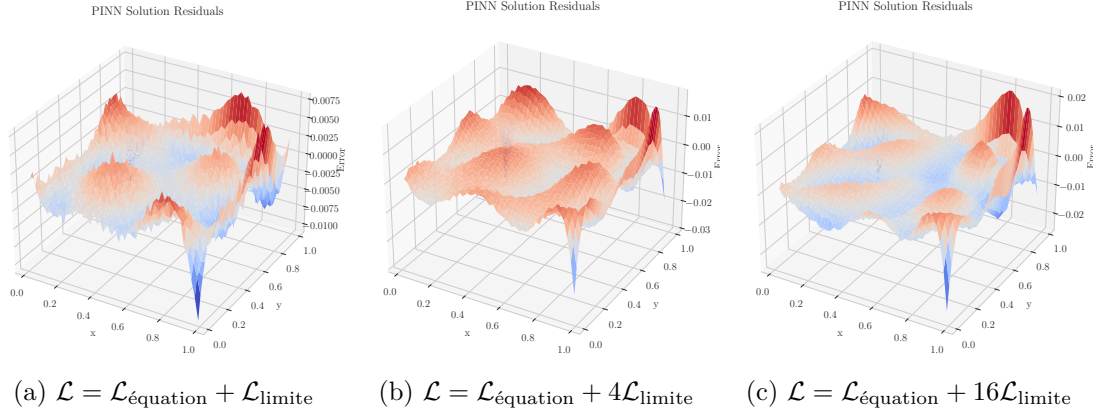


Figure 9: Comparaison des résidus obtenues par PINN (10000 itérations) avec différents coefficients pour le terme de perte des conditions aux limites.

La méthode des différences finies reste, quant à elle, significativement plus rapide à l'exécution (0.00477 secondes), mais moins flexible face à des géométries complexes ou des conditions aux limites non standards.

En termes d'architecture et d'hyperparamètres (voir tableau 3), les deux approches neuronales utilisent une configuration identique (réseau 2-64-64-64-1, taux d'apprentissage de 0.001, 10000 époques), ce qui renforce la validité de leur comparaison directe. La principale différence réside dans l'utilisation d'un mini-batch de taille 128 pour le modèle supervisé, contre un traitement global pour le PINN.

En résumé :

- **Supervisé** : Offre la meilleure précision (erreur 15 fois moindre que le PINN) mais nécessite des données de solution préexistantes et un temps d'entraînement plus long (198.57s).
- **PINN** : Présente l'avantage majeur de fonctionner sans données de solution, avec un temps d'entraînement réduit (94.11s), mais au prix d'une précision plus faible.
- **Différences Finies** : Reste la méthode la plus rapide à l'exécution, servant de référence pour l'évaluation des approches neuronales.

7 Conclusion et Perspectives

7.1 Conclusion

Cette étude comparative entre les Réseaux Neuronaux Informés par la Physique (PINNs) et l'apprentissage supervisé pour la résolution de l'équation de Poisson a révélé des caractéristiques distinctes pour chaque approche.

L'apprentissage supervisé a démontré une précision supérieure (erreur L2 relative de 0.0073 contre 0.1063 pour le PINN) mais nécessite une solution de référence préexistante et un temps d'entraînement plus long (198.57 secondes). En revanche, le PINN offre l'avantage considérable de fonctionner sans données de solution préalable et avec un temps d'entraînement réduit (94.11 secondes), au prix d'une précision plus faible.

Les résultats suggèrent que le choix entre ces méthodes dépend principalement du contexte d'application :

- Les PINNs sont préférables lorsque les solutions analytiques ou numériques sont difficiles à obtenir, ou pour des géométries complexes.

- L'apprentissage supervisé est plus adapté aux problèmes où la précision est primordiale et où des solutions de référence sont disponibles.

La méthode des différences finies reste, quant à elle, significativement plus rapide à l'exécution (0.00477 secondes), mais moins flexible face à des géométries complexes ou des conditions aux limites non standards.

7.2 Perspectives

Plusieurs pistes d'amélioration et d'extension peuvent être envisagées :

1. **Architectures hybrides** : Combiner les avantages des deux approches, par exemple en pré-entraînant un PINN avec des données supervisées puis en l'affinant avec la contrainte physique.
2. **Intégration de connaissances a priori** : Explorer comment incorporer des informations spécifiques au domaine, comme des symétries ou des invariances, pour améliorer la précision et la vitesse de convergence.
3. **Quantification d'incertitude** : Développer des méthodes pour estimer l'incertitude des prédictions, ce qui est essentiel pour les applications critiques.
4. **Apprentissage par transfert** : Étudier la possibilité d'utiliser des modèles entraînés sur certaines configurations pour accélérer l'apprentissage sur de nouvelles configurations similaires.

Ces perspectives illustrent le potentiel considérable des approches basées sur l'apprentissage automatique pour la résolution d'équations aux dérivées partielles, un domaine en pleine expansion à l'intersection de l'intelligence artificielle et de la modélisation physique.