

Fiche Explicative

Table des matières

Dérivation automatique	1
Le vecteur d'entrée : \mathbf{x}	3
La fonction d'activation : σ	3
Les lois physiques : PDE, BC, IC	3
Les collocation points : Tf et Tb	4
Les données mesurées : χ^*	4
La fonction de perte totale (Loss).....	4

Dérivation automatique

C'est une technique qui permet à l'ordinateur de calculer automatiquement des dérivées exactes, même pour des fonctions très complexes, en utilisant la structure interne du calcul. Ce n'est pas de la dérivation symbolique (style "comme en maths") et ce n'est pas de la dérivation numérique (avec des différences finies).

C'est un système qui calcule des dérivées par construction, en utilisant la chaîne d'opérations effectuées.

AD dérive le programme tout seul.

Quand un réseau neuronal calcule une sortie, tout est fait en petites opérations simples :

- Multiplication
- Addition
- Activation (tanh, relu, sigmoid...)
- Combinaison linéaire

AD enregistre toutes ces opérations dans un graphe computationnel. Puis, quand une dérivée, AD remonte dans ce graphe en appliquant la règle de dérivation en chaîne (chain rule) partout automatiquement. On obtient $\partial\psi/\partial x$ ou $\partial^2\psi/\partial x^2$ exactement, sans approximation numérique.

Pourquoi c'est révolutionnaire pour les PINNs ?

Dans un PINN, on intègre la PDE directement dans la fonction de perte.

Par exemple pour Richards, on doit calculer :

- $\partial\psi/\partial t$
- $\partial\psi/\partial z$
- $\partial^2\psi/\partial z^2$
- $k(\psi), \theta(\psi)$ et leurs dérivées
- etc.

Sans AD, il faudrait soi-même coder toutes les dérivées, ce qui est difficile car ψ est donnée par un réseau neuronal très complexe !

Grâce à l'AD

- tu écris juste la PDE telle quelle, comme une équation mathématique
- le Framework (DeepXDE, TensorFlow, PyTorch) calcule toutes les dérivées lui-même
- et les dérivées sont précises, pas approximées

Différence avec les autres méthodes

Méthode	Qualité	Vitesse	Utilité
Dérivée numérique	Approximative (bruit)	Rapide	FDM, FEM
Dérivée symbolique	Exacte	Lente	CAS (Sympy, Maple)
Automatic Differentiation	Exacte + rapide	Rapide	Machine learning & PINNs

L'AD combine donc la précision du symbolique ainsi que la rapidité du numérique

Le vecteur d'entrée : \mathbf{x}

Ce vecteur sert d'entrée au réseau neuronal. Chaque point \mathbf{x} est un endroit où le PINN doit apprendre la solution physique.

Le réseau neuronal : $\hat{\mathbf{u}} = \mathcal{N}(\mathbf{x}, \boldsymbol{\lambda})$:

Il approxime la **solution inconnue** de la PDE :

- soit la pression d'eau $\psi(x, t)$
- soit la teneur en eau $\theta(x, t)$

Pour l'équation de Richards en fonction de l'ensemble de l'espace $\boldsymbol{\lambda}$.

La fonction d'activation : σ

σ est une transformation non linéaire appliquée à chaque couche du réseau (ReLU, tanh, sigmoid...).

Elle permet au réseau d'approximer des comportements complexes, comme les transitions abruptes de saturation observées dans l'infiltration.

Les lois physiques : PDE, BC, IC

Ces trois éléments forment les contraintes physiques que le réseau doit satisfaire.

PDE – l'équation différentielle (Richards)

Elle décrit l'écoulement de l'eau dans les sols non saturés :

$$f(x, t; u) = 0$$

Le PINN calcule les dérivées de $\hat{\mathbf{u}}$ via la différentiation automatique pour vérifier que la PDE est respectée.

BC – Boundary Conditions

Conditions imposées aux frontières (surface du sol, fond de la colonne, etc.).

IC – Initial Conditions

État initial (profil de pression ou d'humidité au début de l'expérience).

La PDE et les conditions aux limites et initiales sont incorporées directement dans la fonction de perte.

Les collocation points : T_f et T_b

T_f – Points dans le domaine pour la PDE

Ce sont des points virtuels où le réseau doit satisfaire la PDE.

Le PINN «apprend la physique» en réduisant l'erreur de la PDE en ces points.

T_b – Points situés sur la frontière

Ils définissent là où les conditions aux limites (BC) doivent être imposées.

Les données mesurées : χ^*

Ce sont les valeurs mesurées sur le terrain ou lors d'un test (capteurs de teneur en eau ou de pression).

Elles ajoutent une contrainte expérimentale : la prédiction du PINN doit coller aux observations.

La fonction de perte totale (Loss)

Le PINN minimise une somme pondérée :

$$L = w_f L_f + w_b L_b + w_i L_i$$

Avec :

- $L_f \rightarrow$ erreur de la PDE (physique)
- $L_b \rightarrow$ erreur des conditions aux limites
- $L_i \rightarrow$ erreur entre prédictions et données mesurées