

# PINNs et Méthodes numériques

## 1 Introduction

On considère une équation elliptique de la forme

$$\begin{cases} L(u) = -\nabla \cdot (A(x)\nabla u(x)) + c(x)u(x) = f(x) & \text{dans } \Omega \\ u(x) = g(x) & \text{sur } \partial\Omega \end{cases} \quad (1)$$

**Formulation faible :**

On multiplie par une fonction test  $v$  et on intègre par parties pour obtenir

$$a(u, v) = l(v) \quad (2)$$

avec

$$\begin{aligned} a(u, v) &= \int_{\Omega} (A(x)\nabla u(x)) \cdot \nabla v(x) + c(x)u(x)v(x) dx \\ l(v) &= \int_{\Omega} f(x)v(x) dx \end{aligned}$$

**Existence et unicité :** Lax-Milgram

**A compléter !**

**Problème de minimisation équivalent :**

Comme  $a$  est symétrique, le théorème de Lax-Milgram implique le problème de minimisation suivant équivalent :

$$u(x) = \arg \min_{v \in V} J(v) \quad (3)$$

avec

$$\begin{aligned} J(v) &= \frac{1}{2}a(v, v) - l(v) \\ &= \frac{1}{2} \int_{\Omega} L(v)v - \int_{\Omega} f v \\ &= \frac{1}{2} \int_{\Omega} (A\nabla v) \cdot \nabla v + cv^2 - \int_{\Omega} f v \end{aligned}$$

On considère un problème en dimension infinie puisqu'on cherche une fonction. L'enjeu des méthodes numériques est de discrétiser ce problème et de le résoudre dans un espace de dimension finie. La résolution se décompose alors en les étapes suivantes :

- **Encodage** : on encode le problème dans un espace de dimension finie
- **Approximation** : on résout le problème dans l'espace de dimension finie
- **Décodage** : on ramène la solution dans l'espace de dimension infinie

Encodeur	Approximateur	Décodeur
$f \rightarrow \theta_f$	$\theta_f \rightarrow \theta_u$	$\theta_u \rightarrow u_\theta$

## 2 Méthodes Eléments Finis Standard

### 2.1 Encodeur/Décodeur

En pratique, on commence par définir le décodeur car c'est lui qui va nous fournir l'espace de dimension finie dans lequel on va travailler. On définit ensuite l'encodeur et l'approximateur en fonction de l'espace de dimension finie choisi.

- **Décodeur** : Dans le cadre des méthodes éléments finis, on choisit un espace de dimension finie  $V_N$  de fonctions polynomiales par morceaux  $\{\varphi_i\}_{i=1}^N$ . On définit alors le décodeur comme la combinaison linéaire de ces fonctions de base :

$$D_\theta(x) = \sum_{i=1}^N \theta_i \varphi_i(x)$$

- **Encodeur** : Comme l'espace  $V_N$  est un espace vectoriel, il existe une définition naturelle de l'encodeur qui consiste à considérer le projecteur orthogonal sur  $V_N$  :

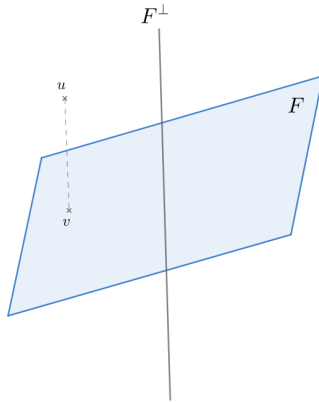
$$\theta_f = E(f) = M^{-1}b(f)$$

avec

$$M_{ij} = \langle \varphi_i, \varphi_j \rangle = \int_{\Omega} \varphi_i(x) \varphi_j(x) dx$$

$$b_i(f) = \langle f, \varphi_i \rangle = \int_{\Omega} f(x) \varphi_i(x) dx$$

**Rappel.**



**Projection orthogonale :**

$$v = p_F(u) \iff \begin{cases} v \in F \\ u - v \in F^\perp \end{cases}$$

Si  $F = \text{Vect}\{e_1, \dots, e_n\}$ , alors

$$u \in F^\perp \iff \forall i \in \{1, \dots, k\}, \quad \langle u, e_i \rangle = 0$$

**Preuve.**

...

## 2.2 Approximateur

La stratégie utilisée précédemment consiste à faire une projection d'une certaine forme de l'équation sur l'espace de dimension finie  $V_N$ . Par unicité de la projection orthogonale on va donc se ramener à un problème sur les degrés de liberté qui lui aussi aura une unique solution. Par contre, on peut obtenir plusieurs variantes pour chaque méthode selon la forme du problème qu'on utilise pour la projection. On va introduire deux projections (Galerkin et Galerkin moindre carrés) et montrer à quel problème sur les degrés de liberté elles se ramènent.

### 2.2.1 Projection de Galerkin

→ **Problème de minimisation considéré :**

Dans ce cas, on considère en fait le même problème de minimisation que celui défini par (3). Pour l'unicité du problème, on va considérer un résidu qui inclut les conditions au bord (condition de Dirichlet). On définit alors le résidu par

$$R(v) = R_i(v)\mathbb{1}_\Omega + R_{bc}(v)\mathbb{1}_{\partial\Omega}$$

avec

$$R_i(v) = L(v) - f \quad \text{et} \quad R_{bc}(v) = v - g$$

qui définissent respectivement les résidus à l'intérieur de  $\Omega$  et sur le bord  $\partial\Omega$ .  
Ainsi on peut définir le problème à minimiser par

$$u(x) = \arg \min_{v \in V} J(v) \quad (4)$$

où

$$J(v) = J_i(v) + J_{bc}(v)$$

avec

$$J_i(v) = \frac{1}{2} \int_{\Omega} L(v)v - \int_{\Omega} f v \quad \text{et} \quad J_{bc}(v) = \frac{1}{2} \int_{\Omega} R_{bc}(v)^2$$

On peut vérifier que la résolution du problème de minimisation (4) est équivalent à la résolution de l'EDP (1).

**Preuve.**

...

#### → Résolution numérique :

La projection de Galerkin consiste à projeter l'EDP sur l'espace  $V_N$  de sorte à ce que  $\forall v \in V_N$  :

$$\langle R(u_{\theta}(x)), v \rangle_{L^2} = 0$$

avec  $R$  le résidu de l'équation défini par :

$$R(v) = L(v) - f$$

On cherche à montrer ici que la projection de Galerkin est équivalente à la minimisation de  $J$  sur  $V_N$ . Plus précisément, comme dans le cadre de la méthode des éléments finis, on a construit une base de  $V_N$  à partir de fonctions de base polynomiale  $\{\varphi_i\}_{i=1}^N$ , on peut écrire la projection de Galerkin sous la forme suivante :

$$\langle R(u_{\theta}(x)), \varphi_i \rangle_{L^2} = 0 \quad \forall i \in \{1, \dots, N\}$$

et le problème de minimisation s'écrit sous la forme d'un problème discret :

$$\theta_u = \arg \min_{\theta \in \mathbb{R}^N} J(\theta) \quad (5)$$

avec

$$J(\theta) = \frac{1}{2} \int_{\Omega} L(v_{\theta})v_{\theta} - \int_{\Omega} f v_{\theta}$$

**Remarque.** On peut remarquer ici que le problème de minimisation définit ici n'inclue pas explicitement les conditions de bord. Dans la pratique, on utilise deux types de méthodes pour imposer les conditions au bord soit par pénalisation soit en le incluant directement dans les fonctions de bases. De ce fait, ici on ne va s'intéresser qu'au problème sur  $\Omega$  sans tenir compte des conditions de bord.

**Preuve.**

...

### 2.2.2 Projection de Galerkin moindre-carré :

#### → Problème de minimisation considéré :

On considère maintenant le problème de minimisation moindre-carré en considérant un résidu qui inclut les conditions au bord (condition de Dirichlet). On définit alors le résidu de la même manière que précédemment :

$$R(v) = R_i(v)\mathbb{1}_\Omega + R_{bc}(v)\mathbb{1}_{\partial\Omega}$$

avec

$$R_i(v) = L(v) - f \quad \text{et} \quad R_{bc}(v) = v - g$$

qui définissent respectivement les résidus à l'intérieur de  $\Omega$  et sur le bord  $\partial\Omega$ .

On peut alors définir le problème moindre-carré de la façon suivante

$$u(x) = \arg \min_{v \in V} J(v) \tag{6}$$

où

$$J(v) = J_i(v) + J_{bc}(v)$$

avec

$$J_i(v) = \frac{1}{2} \int_{\Omega} R_i(v)^2 \quad \text{et} \quad J_{bc}(v) = \frac{1}{2} \int_{\Omega} R_{bc}(v)^2$$

On peut vérifier que la résolution du problème de minimisation (6) est équivalent à la résolution de l'EDP (1).

**Remarque.** A noter que  $J_{bc}$  est identique que dans le cas de la projection par Galerkin.

**Preuve.**

...

#### → Résolution numérique :

La projection de Galerkin moindre carré consiste à projeter l'EDP sur l'espace  $W_N = Vect(\nabla_\theta R(u_\theta(x)))$  de sorte à ce que :

$$\langle R(u_\theta(x)), (\nabla_\theta R(u_\theta(x)))_i \rangle_{L^2} = 0, \quad \forall i \in \{1, \dots, N\}$$

avec  $R$  le résidu de l'équation défini par :

$$R(v) = L(v) - f$$

On cherche à montrer ici que la projection de Galerkin moindre-carré est équivalente au problème de minimisation moindre-carré défini précédemment. Plus précisément, le problème de minimisation s'écrit sous la forme d'un problème discret :

$$\theta_u = \arg \min_{\theta \in \mathbb{R}^N} J(\theta) \tag{7}$$

avec

$$J(\theta) = \frac{1}{2} \int_{\Omega} R(v_\theta)^2$$

**Remarque.** De la même manière, on ne s'intéresse pas aux conditions de bord ici.

**Preuve.**

...

## 2.3 Application à $\phi$ -FEM

Dans le cadre de  $\phi$ -FEM, on considère en fait le décodeur défini par

$$D_\theta(x) = \left( \sum_{i=1}^N \theta_i \varphi_i(x) \right) \phi(x)$$

avec  $\phi$  la fonction level-set représentant notre domaine (et nulle au bord).

Ici, nous ne précisons pas les calculs...

**A compléter !**

**Modifs.**

Pour  $\phi$ -FEM : dire que globalement c'est pareil, on utilise le même décodeur que l'on multiplie par  $\phi$ . Ici, on veut juste expliquer que l'idée est la même : Ne pas refaire les calculs. On doit tout de même préciser qu'on doit rajouter des termes de stabilisation pour assurer la coercivité de la matrice.

## 3 Réseaux de neurones

A présent, nous allons nous intéresser aux réseaux de neurones dit "Physiquement informés". Nous considérerons 2 types de méthodes : les méthodes Deep-Ritz et les méthodes PINNs standard.

**Idée centrale.**

Les méthodes FEM dont on a parlé précédemment utilise un décodeur linéaire par rapport aux paramètres, ce qui implique qui revient à se restreindre à un espace vectoriel de dimension finie. Les méthodes dont on va parler maintenant utilise un **décodeur non linéaire** ce qui revient à se restreindre à des variétés. En plus d'utiliser un décodeur non linéaire, on utilise des modèles réguliers plusieurs fois dérivables dont on calcule les dérivées par **différentiation automatique**

### 3.1 Encodeur/Décodeur