Rapport de stage

Équations aux dérivées partielles, réseaux de neurones et groupes de Lie

Timothée Rocquet

Résumé et remerciements

Ce document est le compte rendu d'un stage de deux mois (17 avril - 30 juin 2023) de recherche, effectué au SAM (Seminar for applied Mathematics) dans le département de mathématiques de l'École Polytechnique Fédérale de Zürich (ETH Zürich), sous la supervision du professeur Siddhartha Mishra, que je tiens à remercier chaleureusement pour son accueil. Je veux aussi particulièrement remercier Emmanuel de Bézenac, post-doctorant au SAM, pour m'avoir aidé et guidé tout au long de ce stage.

La simulation de phénomènes physiques est un enjeu crucial dans de nombreux domaines, autant dans la recherche que dans l'industrie. Celle-ci passe souvent par la modélisation des phénomènes au moyens de lois physiques, exprimées mathématiquement sous la forme d'équations aux dérivées partielles (EDP), puis par la résolution numérique de celles-ci. Cependant, les méthodes de résolution traditionnelles (éléments finis, volumes finis) sont assez lentes et coûteuses en calcul. Mon objet d'étude était la résolution numérique des EDP issues de la physique au moyen de réseaux de neurones. Plus précisément, il s'agissait d'améliorer les performances des réseaux existants en tenant compte des symétries sous-jacentes aux EDP étudiées.

1) Contexte

La résolution numérique efficace d'équations aux dérivées partielles à des fins de simulations est un enjeu crucial dans de nombreux domaines de recherche : physique, biologie, chimie, médecine, finance... Les algorithmes d'approximation numérique traditionnels étant extrêmement lents, de nouvelles méthodes émergent, dont plusieurs reposant sur les réseaux de neurones. L'objectif du stage était d'améliorer les réseaux de neurones existants de manière à prendre en compte les symétries des EDP issues de la physique, l'intérêt étant d'obtenir de meilleurs propriétés de généralisation pour le réseau entraîné. C'est particulièrement utile quand on a peu de données sur lesquelles entraîner le réseau.

Les EDP On désigne par **équation aux dérivées partielles** toute expression de la forme $F(x,u) = 0$ où F est une fonction de $x = (x_1, ..., x_p)$, de $u(x)$ et des dérivées partielles $\frac{\partial^{|\alpha|}u}{\partial x_1^{\alpha_1}\partial x_2^{\alpha_2}...\partial x_p^{\alpha_p}}(x) = \frac{\partial^\alpha u}{\partial x^\alpha}$ avec $\alpha \in \mathbb{N}^p$ et $u : \Omega \to \mathbb{R}^q$ une fonction lisse inconnue, où $\Omega \subset \mathbb{R}^p$ est ouvert. Plus fromellement, on note $u^{(n)}$ la fonction

$$u^{(n)} : x \mapsto \left(\frac{\partial^\alpha u}{\partial x^\alpha}\right)_{|\alpha|\leq n} = \left(u(x), \frac{\partial u}{\partial x_1}, ..., \frac{\partial u}{\partial x_p}, \frac{\partial^2 u}{\partial x_1^2}, \frac{\partial^2 u}{\partial x_1 \partial x_2}, ..., \frac{\partial^n u}{\partial x_p^n}\right).$$

Une EDP d'ordre n est alors une expression de la forme

$$F(x, u^{(n)}(x)) = 0.$$

Par exemple :

$$\text{Équation de Burgers : } \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$$

$$\text{Équation de la chaleur : } \frac{\partial u}{\partial t} - \Delta u = 0$$

$$\text{Équation de la chaleur avec terme source : } \frac{\partial u}{\partial t} - \Delta u - S = 0 \text{ (S est une fonction fixée)}$$

$$\text{Équation d'onde : } \frac{\partial^2 u}{\partial t^2} - \Delta u = 0$$

où $\Delta u = \sum_i \frac{\partial^2 u}{\partial x_i^2}$. On notera parfois pour simplifier $\frac{\partial u}{\partial x} = \partial_x u = u_x$.

La résolution d'une équation aux dérivées partielles consiste généralement à trouver une fonction u, de préférence lisse, vérifiant l'égalité $F(x, u^{(n)}(x)) = 0$ en tout point $x \in \Omega$ et vérifiant en plus des conditions de bord, par exemple $u|_{\partial\Omega} = g$ où g est une fonction donnée. Les méthodes classiques de résolution numérique des EDP consistent à discrétiser Ω en le recouvrant d'un maillage $\{x^i\}_{i=1}^N$ puis à reformuler l'équation $F(x, u^{(n)}(x)) = 0$ sous la forme de relations (souvent linéaires) entre les $u(x^i)$ et enfin à résoudre ce système discret d'équations. Il est aussi possible de choisir une base de fonctions $\{\varphi_i\}_{i=1}^N$, d'exprimer u comme combinaison linéaire de ces fonctions $u = \sum_i c_i \varphi_i$ puis de traduire l'EDP en système d'équations sur les coefficients c_i pour les calculer.

Quelle que soit la méthode utilisée, c'est en général long et coûteux en calculs. Pour pallier ce problème, une idée possible est d'utiliser l'apprentissage automatique via des réseaux de neurones. L'avantage de cette approche est que l'on effectue l'entraînement, très coûteux, en amont puis on utilise le réseau entraîné pour résoudre notre équation rapidement.

L'apprentissage automatique Un **réseau de neurones** est une famille de fonctions $\mathcal{G}_\theta : \mathbb{R}^{d_e} \to \mathbb{R}^{d_s}$, indexées par un paramètre $\theta \in \Theta \subset \mathbb{R}^\nu$ qui prend en entrée un vecteur $\mathbf{e}$ et qui en sort un autre $\mathbf{s} = \mathcal{G}_\theta(\mathbf{e})$. En général, le réseau est lui-même la composition de plusieurs fonctions (appelées couches) :

$$\mathcal{G}_\theta = C_{\theta_1} \circ C_{\theta_2} \circ ... \circ C_{\theta_m}$$

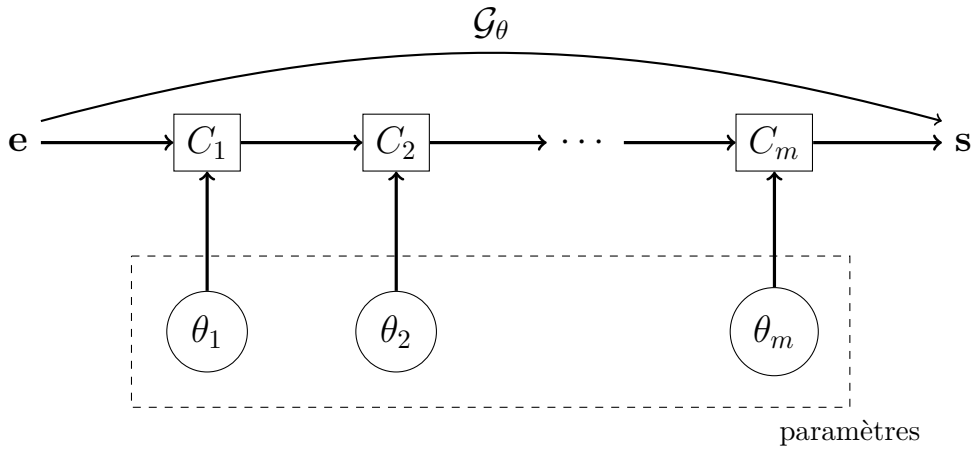

FIGURE 1 – Schéma d'un réseau de neurones

Le principe de l'apprentissage automatique (ou machine learning) est le suivant : on cherche à approcher une fonction cible donnée $\mathcal{G}$ au moyen d'une fonction $\mathcal{G}_\theta$, ie. trouver un paramètre θ qui rende l'écart entre $\mathcal{G}$ et $\mathcal{G}_\theta$ le plus faible possible. La cible $\mathcal{G}$ peut par exemple être la fonction qui à une image $\mathbf{e} \in [0,1]^{3\times 64\times 64}$ associe le vecteur $(1,0)$ si elle représente un chien, $(0,1)$ si elle représente un chat. Pour trouver le meilleur paramètre θ, on définit une fonction de perte $\ell : \mathbb{R}^{d_s} \times \mathbb{R}^{d_s} \to \mathbb{R}_+$ telle que $l(\mathbf{s}_1, \mathbf{s}_2)$ est d'autant plus petit que $\mathbf{s}_1$ et $\mathbf{s}_2$ sont proches.

L'entraînement du réseau de neurones consiste à prendre un certain nombre de couples entrée-sortie $\{(\mathbf{e}_i, \mathbf{s}_i)\}_{i=1}^M$ où $\mathbf{s}_i = \mathcal{G}(\mathbf{e}_i)$ et à minimiser la quantité

$$l(\theta) = \sum_{i=1}^{M} \ell(\mathcal{G}_\theta(\mathbf{e}_i), \mathbf{s}_i).$$

La minimisation se fait par descente de gradient : on itère la relation $\theta_{n+1} = \theta_n - \varepsilon \nabla l(\theta_n)$ de sorte que θ_n converge vers un minimum local (on utilise en réalité des versions améliorées de cet algorithme).

Dans le cas qui nous intéresse, l'entrée $\mathbf{e}$ représente les paramètres de l'EDP (conditions de bord, terme source...) et la sortie $\mathbf{s}$ représente la solution de l'EDP, les deux étant discrétisées.

Groupes de symétrie Les EDP issues de la physique possèdent généralement une certain nombre de symétries, par exemple l'invariance par rotation ou translation, qui sont représentées par des groupes de Lie. Formellement, on considère une EDP $F(x, u^{(n)}(x)) = 0$ sur une fonction lisse $u : \Omega \subset \mathbb{R}^p \to \mathbb{R}^q$. À une solution u, on peut associer son graphe

$$\Gamma_u = \{(x, u(x)), x \in \Omega\}.$$

Un **groupe de Lie** est une variété lisse munie d'une structure de groupe et dont les opérations sont lisses. Étant donné un groupe de Lie G d'élément neutre e_G et une action lisse de G sur $\Omega \times \mathbb{R}^q$, pour tout point $(x, u(x)) \in \Gamma_u$, il existe un voisinage V de ce point est un voisinage W de e_G tel que $g \cdot (\Gamma_u \cap V)$ soit aussi le graphe d'une certaine fonction $\tilde{u}$, quel que soit $g \in W$. On dit que G est une **symétrie** de l'EDP si $\tilde{u}$ est également une solution, et ce quel que soit $g \in W$.

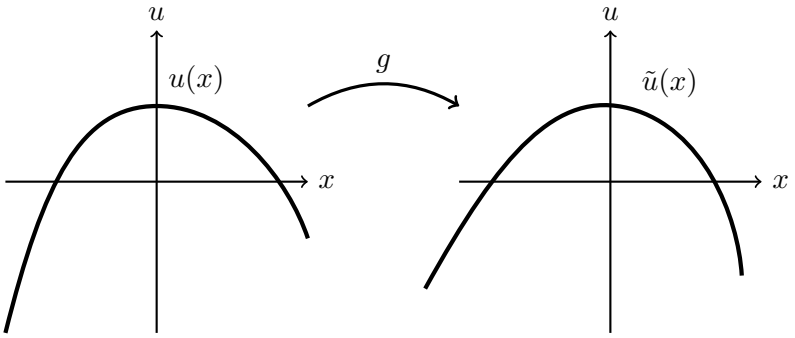

FIGURE 2 – Action de $SO(2)$ sur les fonctions de $\mathbb{R}$ dans $\mathbb{R}$

Exemple On se place dans les cas simple où $\mathbb{R}^p = \mathbb{R}^q = \mathbb{R}$, l'équation considérée est $u_{xx} = 0$, le groupe G est le groupe des rotations $G = SO(2)$ qui agit naturellement sur le plan $\mathbb{R} \times \mathbb{R}$. Les fonctions réelles qui vérifient $u_{xx} = 0$ sont les fonctions affines $u(x) = ax + b$, dont le graphe est une droite non verticale. Or si l'on prend un segment de droite non vertical et qu'on lui applique une rotation d'angle suffisamment faible, cela reste un segment de droite non vertical donc $\tilde{u}$ est également une fonction affine et vérifie $\tilde{u}_{xx} = 0$: $SO(2)$ est un groupe de symétrie de $u_{xx} = 0$.

Symétrie infinitésimale On peut associer à chaque groupe de Lie G un objet fondamental, son **algèbre de Lie** $\mathfrak{g}$. Il s'agit d'un espace vectoriel de dimension finie muni d'une stucture supplémentaire, le crochet de Lie, et qui a l'avantage d'essentiellement caractériser la géométrie du groupe. À chaque vecteur de $\mathfrak{g}$ on peut associer un champ de vecteurs $\mathbf{v}$ sur $\Omega \times \mathbb{R}^q$ de sorte que l'action de G se ramène au flot du champ de vecteurs $\mathbf{v}$. Formellement, si $\mathbf{g} \in \mathfrak{g}$, on définit $\mathbf{v}$ de sorte que si $\Phi_{\mathbf{v}}(t, x)$ est le flot associé, on ait

$$\exp(t\mathbf{g}) \cdot (x, u) = \Phi_{\mathbf{v}}(t, (x, u)).$$

Prenons par exemple l'action de $SO(2)$ sur $\mathbb{R}^2$: $G = SO(2) \cong \mathbb{S}^1$, $\mathfrak{g} \cong \mathbb{R}$ et avec $\mathbf{g} = \partial_x$, $\exp(t\mathbf{g}) = R(\theta)$ la rotation d'angle θ donc $\mathbf{v}(x,y) = \frac{d}{d\theta}\big|_{\theta=0} R(\theta)\cdot(x,y) = x\partial_y - y\partial_x$.

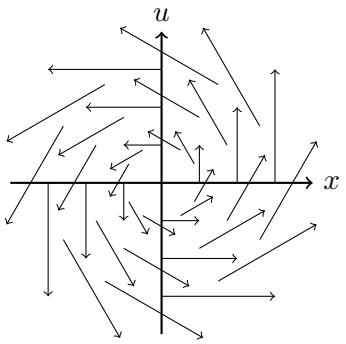

FIGURE 3 – Représentation du champ $x\partial_y - y\partial_x$ associé à l'action de $SO(2)$ sur $\mathbb{R}\times\mathbb{R}$

Le résultat clé est que la condition de symétrie peut alors être reformulée de manière plus simple sous la forme $F(x,u) = 0 \implies \mathbf{v}.F(x,u) = 0$ (voir le théorème 2 pour l'énoncé exact) ce qui permet, avec un peu de calcul, de dresser la liste de toutes les symétries de manière exhaustive.

Lien avec le réseau de neurones Une fois les symétries connues, le problème est de les incorporer au réseau de neurones. L'option choisie a été de traduire la condition précédente comme une condition sur $D\mathcal{G}_u(v)$ et de rajouter un terme

$$\|D(\mathcal{G}_\theta)_u(v) - D\mathcal{G}_u(v)\|^2$$

à la fonction de perte ℓ (voir plus bas pour les détails). Cela a été expérimenté sur l'exemple suivant : on considère l'équation de la chaleur sur le tore, l'entrée est la fonction à l'instant initial, la sortie est le profil de la chaleur à l'instant $t = 0.1$. On optimise $\mathcal{G}_\theta$ sur une seule fonction, avec et sans terme de perte lié à l'invariance par translation dans le temps et dans l'espace. On observe effectivement une amélioration du réseau de neurones en ajoutant le terme dû à la symétrie (voir la fin du rapport pour les résultats numériques).

2) Détails mathématiques et résultats

2.1) Groupes de Lie locaux et symétries

Le développement qui suit est tiré des deux premiers chapitres de l'ouvrage [Olv93].

De manière générale, une symétrie peut être locale sans être globale, par exemple une EDP invariante par rotation sans que le domaine de définition ne soit lui-même invariant par rotation. C'est pourquoi on peut se contenter, pour la définition d'une symétrie, d'un goupe local agissant de manière locale.

Définition 1. *Un **groupe de Lie local** de dimension r est la donnée d'un ouvert $0 \in G \subset \mathbb{R}^r$ et de deux opérations lisses*

$$\begin{cases} G\times G \to \mathbb{R}^r \\ (x,y) \mapsto x\cdot y \end{cases} \qquad \begin{cases} G \to G \\ x \mapsto x^{-1} \end{cases}$$

de sorte que

$$\begin{cases} \forall (x,y,z)\in G^3, (x\cdot y)\cdot z = x\cdot(y\cdot z) \text{ si les deux membres sont définis} \\ \forall x\in G, x\cdot 0 = 0\cdot x = x \\ \forall x\in G, x\cdot x^{-1} = x^{-1}\cdot x = 0 \end{cases}.$$

On le notera simplement G, et $e = 0$ sont élément neutre.

Remarque. Avec cette définition, on peut montrer qu'un groupe de Lie local est toujours isomorphe à la restriction d'un groupe de Lie à un voisinage de l'élément neutre.

Soit G un groupe de Lie local. On peut définir son algèbre de Lie de la même manière qu'avec un groupe classique : pour tout $g \in G$, l'application $R_g : h \mapsto h \cdot g$ définit un homéomorphisme d'un voisinage de e_G vers un voisinage de g, donc $dR_g : TG_e \to TG_g$ est un isomorphisme. Les champs de vecteurs $\mathbf{v}$ tels que $\mathbf{v}(g) = dR_g(\mathbf{v}(e))$ sont dits invariants à droite, et l'ensemble de ces champs invariants à droite munis du crochet de Lie forment l'**algèbre de Lie** $\mathfrak{g} \cong TG_e$ de G. De plus, à tout champs $\mathbf{g} \in \mathfrak{g}$ on peut associer son flot $\Phi_{\mathbf{g}}$ et donc définir l'application exponentielle :

$$\exp(t\mathbf{g}) = \Phi_{\mathbf{g}}(t, e) = \Phi_{t\mathbf{g}}(1, e).$$

Définition 2. *Étant donné une variété M et un groupe de Lie local G, une **action locale** de G sur M est la donnée d'un ouvert $\{e\} \times M \subset V \subset G \times M$ et d'une application lisse*

$$\begin{cases} U \to M \\ (g, x) \mapsto g \cdot x \end{cases}$$

telle que

$$\begin{cases} \forall (g,h) \in G^2, \forall x \in M, h \cdot (g \cdot x) = (h \cdot g) \cdot x \text{ si les deux membres sont bien définis} \\ \forall x \in M, e \cdot x = x \\ \forall (g,x) \in U, (g^{-1}, g \cdot x) \in U \text{ (donc } g^{-1} \cdot (g \cdot x) = x) \end{cases}.$$

*L'action locale est dite **connexe** si G, M, V et $\{g \in G | (g, x) \in U\}$ sont connexes, quel que soit $x \in M$. Par simplicité, toutes les actions seront implicitement considérées comme connexes.*

Pour tout $x \in M$, l'application $a_x : G \to M, g \mapsto g \cdot x$ est lisse et va d'un voisinage de e dans un voisinage de x donc la différentielle est une application linéaire $da_x : \mathfrak{g} \to TM_x$. On peut alors associer à tout $\mathbf{g} \in \mathfrak{g}$ un champ de vecteurs sur M comme $\psi(\mathbf{g}) : x \mapsto da_x(\mathbf{g})$. On obtient alors une application

$$\psi : \begin{cases} \mathfrak{g} \to \Gamma(M) \text{ (champs de vecteurs lisses sur } M) \\ \mathbf{g} \mapsto \left(x \mapsto da_x(\mathbf{g}) = \left.\dfrac{d}{dt}\right|_{t=0} \exp(t\mathbf{g}) \cdot x \right) \end{cases}$$

qui est un morphisme d'algèbres de Lie et qui vérifie

$$\exp(t\mathbf{g}) \cdot x = \Phi_{\psi(\mathbf{g})}(t, x).$$

Or, si $\mathbf{g}_1, ..., \mathbf{g}_r$ est une base de $\mathfrak{g}$, tout élément de G s'écrit sous la forme $\exp(t_i \mathbf{g}_{i_1}) ... \exp(t_k \mathbf{g}_{i_k})$ (car G est connexe et la relation d'équivalence $g \sim h \iff h = \exp(t_i \mathbf{g}_{i_1}) ... \exp(t_k \mathbf{g}_{i_k}) \cdot g$ n'a que des classes ouvertes) donc l'action de G est entièrement déterminée par les flots des champs de vecteurs $\psi(\mathbf{g}_1), ..., \psi(\mathbf{g}_r)$. On confondra parfois abusivement $\psi(\mathbf{g})$ et $\mathbf{g}$.

Formalisons maintenant les propriétés de symétrie esquissées en introduction. On définit $X = \mathbb{R}^p$, $U = \mathbb{R}^q$ et on considère un système d'EDP dont l'inconnue $u = (u^1, ..., u^q)$ est une fonction lisse de $x = (x^1, ..., x^p)$, donc de X dans U. Soit G un groupe de Lie local agissant de manière locale sur un ouvert $M \subset X \times U$. Soit $\Omega \subset X$ un ouvet, $f : \Omega \to U$ une fonction lisse dont le graphe $\Gamma_f = \{(x, f(x)), x \in \Omega\}$ vérifie $\Gamma_f \subset M$. Pour tout $\Omega' \Subset \Omega$ (partie ouverte d'adhérence compacte), il existe un voisinage G' de e dans G tel que pour tout $g \in G'$, l'ensemble $g \cdot \Gamma_f = \{g \cdot (x, u), (x, u) \in \Gamma_f\}$ est le graphe d'une fonction $\tilde{f} : X \to U$:

$$g \cdot \Gamma_f = \Gamma_{\tilde{f}}.$$

On définit alors $\tilde{f}$ comme l'image de f par l'action de g :

$$\tilde{f} = g \cdot f$$

(ou pour être parfaitement rigoureux : $\tilde{f} = g \cdot (f|_{\Omega'})$). La construction est illustrée figure 3.

Formellement, si on définit l'action du groupe comme

$$g \cdot (x, u) = (\Xi_g(x, u), \Psi_g(x, u))$$

alors la nouvelle fonction est définie par

$$\tilde{f} = g \cdot f = (\Psi_g \circ (\mathbb{1} \times f)) \circ (\Xi_g(x) \circ (\mathbb{1} \times f))^{-1} \tag{1}$$

où $\mathbb{1} \times f : x \mapsto (x, f(x))$.

Définition 3. *Avec les définitions précédentes, G est une **symétrie** du système d'EDP si pour toute solution f, si $g \cdot f$ est définie, alors c'est également une solution.*

L'objectif serait de pouvoir énumérer les symétries d'une EDP donnée. Pour ce faire, nous allons étendre l'espace d'arrivée U de sorte qu'in contienne également les dérivées de u. La fonction F peut alors être vue comme une fonction partant de cet espace prolongé et le système d'EDP $F(x, u) = 0$ est une variété dans cet espace prolongé. Une symétrie sera alors une action de groupe laissant invariante cette variété.

On a toujours $X = \mathbb{R}^p$ l'espace de départ, $U = \mathbb{R}^q$ l'espace d'arrivé. Pour $n \in \mathbb{N}$, le nombre de $\alpha \in \mathbb{N}^p$ tels que $|\alpha| \leq n$ (où $|\alpha| = \alpha_1 + ... + \alpha_p$) est $\binom{p+n}{p}$. On note $U^{(n)} = \mathbb{R}^{q \times \binom{p+n}{p}}$ l'espace étendu et pour toute fonction $f = (f^1, ..., f^q) : \Omega \to U$, on peut définir le **prolongement** de f comme

$$\mathsf{pr}^{(n)} f : \begin{cases} \Omega \to U^{(n)} \\ x \mapsto (\partial_J f^\alpha)_{1 \leq \alpha \leq q, J \in \mathbb{N}^p, |J| \leq n} \end{cases}.$$

La coordonnée dans $U^{(n)}$ correspondant à $\partial_J f^\alpha$ sera notée u^α_J. Les coordonnées $(x^1, ...x^p)$ sont les **variables indépendantes** et les coordonnées $(u^\alpha_J)_{J \in \mathbb{N}^n}$ sont les **coordonnées dépendantes**.

Pour $M \subset X \times U$ ouvert, on définit l'**espace des jets** $M^{(n)} = M \times \mathbb{R}^{q \times \binom{p+n}{p} - p - q} \subset X \times U^{(n)}$. Un **système de l équations différentielles** est alors une équation $F(x, (\mathsf{pr}^{(n)} u)(x)) = 0$ où $F = (F_1, ..., F_l) : M^{(n)} \to \mathbb{R}^l$ est une fonction lisse. Cette fonction F s'annule sur une variété

$$\mathscr{S}_F = \{(x, u^{(n)}) \in M^{(n)} | F(x, u^{(n)}) = 0\} \subset X \times U^{(n)}$$

et f est solution si et seulement si le graphe de $\mathsf{pr}^{(n)} f$ est inclus dans $\mathscr{S}_F$.

Le point clé réside dans le fait suivant : soit $(x_0, u_0^{(n)}) \in M^{(n)}$ et f une fonction définie sur un voisinage de x_0 telle que $\mathsf{pr}^{(n)} f(x_0) = u_0^{(n)}$ (on peut par exemple prendre le polynôme de Taylor d'ordre n). Quitte à restreindre le domaine de f, la fonction $g \cdot f$ est bien définie pour tout $g \in G$ dans un certain voisinage de e donc on peut poser $\tilde{f} = g \cdot f$, $(\tilde{x}_0, \tilde{u}_0) = g \cdot (x_0, u_0)$ et $\tilde{u}_0^{(n)} = \mathsf{pr}^{(n)} \tilde{f}(\tilde{x}_0)$. La valeur de $\tilde{u}_0^{(n)}$ ne dépend que des dérivées de $\tilde{f}$ en $\tilde{x}_0$ donc, d'après la formule (1), elle ne dépend que des dérivées de f en x_0 et des dérivées de Ξ_g, Φ_g. Elle ne dépend dont *pas* de la fonction f choisie, ce qui justifie la définition suivante.

Définition 4. *Avec les définitions précédentes, on appelle **prolongement de l'action** de G sur M l'action de G sur $M^{(n)}$ définie par*

$$\mathsf{pr}^{(n)} g \cdot (x_0, u_0^{(n)}) = (\tilde{x}_0, \tilde{u}_0^{(n)}).$$

Ainsi, G est une symétrie du système d'EDP si et seulement si $\mathscr{S}_F$ est stable par G (c'est en réalité un petit peu plus subtil car tous les points de $\mathscr{S}_F$ n'appartiennent pas nécessairement au graphe d'une solution du système d'EDP donc cette condidion est suffisante mais il faut rajouter des contraintes sur F pour qu'elle soit nécessaire).

On l'a vu, l'action d'un groupe se résume au flot d'une base de son algèbre de Lie. Mais un champ de vecteurs $\mathbf{v}$ sur M est équivalent à une action locale de $\mathbb{R}$ sur M (l'action associée étant $t \cdot x = \Phi_{\mathbf{v}}(t,x) = \exp(t\mathbf{v}) \cdot x$) donc en la prolongeant à une action sur $M^{(n)}$ comme décrit précédemment, on obtient un champ de vecteurs sur $M^{(n)}$: le prolongement de $\mathbf{v}$.

Définition 5. *Soit* $\mathbf{v}$ *un champ de vecteurs sur* M*. On définit* $\mathsf{pr}^{(n)}\mathbf{v}$ *le* ***prolongement*** *du champ* $\mathbf{v}$ *par, pour tout* $(x, u^{(n)}) \in M^{(n)}$*,*

$$\mathsf{pr}^{(n)}\mathbf{v}(x,u^{(n)}) = \frac{d}{dt}\Big|_{t=0} \mathsf{pr}^{(n)} \exp(t\mathbf{v}) \cdot (x,u^{(n)}).$$

L'opérateur de prolongement ainsi défini est un morphisme d'algèbres de Lie des champs de vecteurs sur M vers les champs de vecteurs sur $M^{(n)}$.

On veut alors savoir à quelle condition une variété est stable par le flot d'un champ de vecteurs. C'est l'objet du théorème suivant.

Théorème 1. *Soit* M *une variété lisse de dimension* m*,* $\mathbf{v}$ *un champ de vecteurs sur* M*,* $F : M \to \mathbb{R}^l$ *une fonction lisse, où* $l \leq m$*. On considère l'ensemble* $\mathscr{S}$ *des solutions du système d'équations* $F(x) = 0$ *et on suppose que* F *est de rang plein sur* $\mathscr{S}$*, ie. pour tout* $x \in \mathscr{S}$*,* dF_x *est de rang* l*. Alors* $\mathscr{S}$ *est stable par le flot de* $\mathbf{v}$ *si et seulement si*

$$\forall x \in M, F(x) = 0 \implies \mathbf{v}.F(x) = 0.$$

Démonstration. Si $\mathscr{S}$ est stable par le flot de $\mathbf{v}$ alors pour tout $x \in \mathscr{S}$ et $t \in \mathbb{R}$ suffisemment petit, $\exp(t\mathbf{v}) \cdot x \in \mathscr{S}$ donc $F(\exp(t\mathbf{v}) \cdot x) = 0$ donc en dérivant en $t = 0$, on obtient $\mathbf{v}.F(x) = 0$.

Réciproquement, si $\mathbf{v}.F(x) = 0$ pour tout $x \in \mathscr{S}$ alors par théorème du rang constant, on peut en tout point trouver une carte local au voisinage de tout point x de $\mathscr{S}$ dans laquelle $F(y^1, ..., y^m) = (y^1, ..., y^l)$. Avec $\mathbf{v}(y) = \sum_k \xi^k(y)\partial_{y^k}$, la condition devient

$$y^1 = ... = y^l = 0 \implies \xi^1(y) = ... = \xi^l(y) = 0.$$

Par conséquent, on peut restreindre le champ de vecteurs au sous-espace $\{y | y^1 = ... = y^l = 0\}$ et par unicité du flot, celui-ci reste dans ce sous-espace, au moins dans un voisinage de x. On a donc montré que pour tout $x \in \mathscr{S}$, $\exp(t\mathbf{v}) \cdot x \in \mathscr{S}$ pour tout t suffisemment petit. Puisque $\mathscr{S}$ est fermée et le flot est continu, $\{t \in \mathbb{R} | \exp(t\mathbf{v}) \cdot x \in \mathscr{S}\}$ est à la fois ouvert et fermé donc par connexité, $\exp(t\mathbf{v}) \cdot x \in \mathscr{S}$ pour tout t.

Corollaire 1. *Soit* M *une variété lisse de dimension* m*,* G *un groupe de Lie agissant localement sur* M*,* $F : M \to \mathbb{R}^l$ *une fonction lisse, où* $l \leq m$*. On considère l'ensemble* $\mathscr{S}$ *des solutions du système d'équations* $F(x) = 0$ *et on suppose que* F *est de rang plein sur* $\mathscr{S}$*, ie. pour tout* $x \in \mathscr{S}$*,* dF_x *est de rang* l*. Soit* $\mathbf{g}_1, ..., \mathbf{g}_r$ *une base de* $\mathfrak{g}$*. Alors les conditions suivantes sont équivalentes :*

- $\mathscr{S}$ *est stable par l'action de* G*,*
- $\forall x \in M, F(x) = 0 \implies (\forall \mathbf{g} \in \mathfrak{g}, \mathbf{g}.F(x) = 0)$*,*
- $\forall x \in M, F(x) = 0 \implies (\forall k \in [\![1, r]\!], \mathbf{g}_k.F(x) = 0)$*.*

Démonstration. C'est une conséquence directe du théorème précédent et du fait que tout élément de G s'écrive $\exp(t_i\mathbf{g}_{i_1})...\exp(t_k\mathbf{g}_{i_k})$.

En appliquant ce théorème à la variété $\mathscr{S}_F$, on obtient une condition suffisante (et souvent nécessaire) pour qu'un groupe G soit une symétrie d'un système d'EDP.

Théorème 2. *Soit $X = \mathbb{R}^p$, $U = \mathbb{R}^q$, $M \subset X \times U$, $M^{(n)}$ l'espace des jets associé, $F : M^{(n)} \to \mathbb{R}^l$ une fonction lisse, G une groupe de Lie local agissant localement sur M. On pose $\mathscr{S}_F = \{(x, u^{(n)}) \in M^{(n)} | F(x, u^{(n)}) = 0\}$ et on suppose que dF est de rang maximal sur $\mathscr{S}_F$. Si*

$$\forall \mathbf{g} \in \mathfrak{g}, \forall (x, u^{(n)}) \in \mathscr{S}_F, (\mathsf{pr}^{(n)}\mathbf{g}).F(x, u^{(n)}) = 0$$

alors G est un groupe de symétrie du système d'EDP $F(x, \mathsf{pr}^{(n)} f(x)) = 0$.

Remarque. En fait cette condition nécessaire est presque suffisante. Pour qu'elle le soit effectivement, il faut rajouter une contrainte sur F : il faut que tout point de $\mathscr{S}_F$ appartienne au graphe du prolongement d'une solution lisse de l'EDP.

Il reste donc, pour pouvoir énumérer les symétrie d'une EDP, à calculer $\mathsf{pr}^{(n)}\mathbf{v}$ pour un champ de vecteurs quelconque. Il existe heureusement une formule explicite, la **formule de prolongement**.

Définition 6. *Soit $P : M^{(n)} \to \mathbb{R}^l$ une fonction lisse. Pour tout $1 \leq k \leq p$, la **k-ième dérivée totale** de P est l'unique fonction $D_k P : M^{(n+1)} \to \mathbb{R}^l$ telle que, pour toute fonction lisse $f : \Omega \subset X \to U$ telle que $\Gamma_f \subset M$,*

$$\forall x \in \Omega, \mathsf{D}_k P(x, \mathsf{pr}^{(n+1)} f(x)) = \frac{\partial}{\partial x^k}\left[P(x, \mathsf{pr}^{(n)} f(x))\right].$$

Explicitement,

$$\mathsf{D}_k P = \frac{\partial P}{\partial x^k} + \sum_{\alpha=1}^{q} \sum_{J \in \mathbb{N}^p} u^\alpha_{J,k} \frac{\partial P}{\partial u^\alpha_J} \tag{2}$$

où $u_{J,k}$ se rapporte à la coordonnée dans $U^{(n)}$ liée à la dérivée $\partial_k \partial_J$.

Théorème 3. *Soit*

$$\mathbf{v}(x, u) = \sum_{k=1}^{p} \xi^k(x, u)\partial_{x^k} + \sum_{\alpha=1}^{q} \varphi_\alpha(x, u)\partial_{u^\alpha}$$

un champ de vecteurs lisse sur un ouvert $M \subset X \times U$. Alors son prolongement est donné par

$$\mathsf{pr}^{(n)}\mathbf{v} = \mathbf{v} + \sum_{\alpha=1}^{q} \sum_{\substack{J \in \mathbb{N}^p \\ |J| \leq n}} \varphi^J_\alpha(x, u^{(n)})\partial_{u^\alpha_J}$$

avec les relations

$$\varphi^{J,k}_\alpha = \mathsf{D}_k \varphi^J_\alpha - \sum_{i=1}^{p} u^\alpha_{J,i} \mathsf{D}_k \xi^i, \tag{3}$$

$$\varphi^J_\alpha = \mathsf{D}_J \left(\varphi_\alpha - \sum_{i=1}^{p} u^\alpha_i \xi^i\right) + \sum_{i=1}^{p} u^\alpha_{j,i} \xi^i. \tag{4}$$

Démonstration. Voir [Olv93], théorème 2.36, page 110. Il s'agit essentiellement de montrer la formule à l'ordre 1 en utilisant l'expression explicite (1). On en déduit ensuite par récurrence les formules (3) et (4).

Comme exemple, appliquons les théorèmes 2 et 3 à un cas très simple : prenons $p = q = 1$ et l'équation $u_{xx} = 0$, dont les solutions sont les fonctions affines.

Soit $\mathbf{v} = \xi\partial_x + \varphi\partial_u$ un champ de vecteurs lisse, $\mathsf{pr}^{(2)}\mathbf{v} = \xi\partial_x + \varphi\partial_u + \varphi^x\partial_{u_x} + \varphi^{xx}\partial_{u_{xx}}$ sa prolongation. D'après (3) et (2),

$$\begin{aligned}
\mathsf{D}_x\xi &= \partial_x\xi + u_x\partial_u\xi = \xi_x + u_x\xi_u \\
\mathsf{D}_x\varphi &= \partial_x\varphi + u_x\partial_u\varphi = \varphi_x + u_x\varphi_u \\
\varphi^x &= \mathsf{D}_x\varphi - u_x\mathsf{D}_x\xi = \varphi_x + u_x\varphi_u - u_x\xi_x - u_x^2\xi_u \\
\mathsf{D}_x\varphi^x &= \partial_x\varphi^x + u_x\partial_u\varphi^x + u_{xx}\partial_{u_x}\varphi^x \\
&= \varphi_{xx} + u_x\varphi_{xu} - u_x\xi_{xx} - u_x^2\xi_{xu} + u_x\varphi_{xu} + u_x^2\varphi_{uu} \\
&\quad - u_x^2\xi_{xu} - u_x^3\xi_{uu} + u_{xx}\varphi_u - u_{xx}\xi_x - 2u_{xx}u_x\xi_u \\
\varphi^{xx} &= \mathsf{D}_x\varphi^x - u_{xx}\mathsf{D}_x\xi \\
&= \varphi_{xx} + u_x\varphi_{xu} - u_x\xi_{xx} - u_x^2\xi_{xu} + u_x\varphi_{xu} + u_x^2\varphi_{uu} - u_x^2\xi_{xu} \\
&\quad - u_x^3\xi_{uu} + u_{xx}\varphi_u - u_{xx}\xi_x - 2u_{xx}u_x\xi_u - u_{xx}\xi_x - u_{xx}u_x\xi_u \\
&= \varphi_{xx} + 2u_x\varphi_{xu} - u_x\xi_{xx} - 2u_x^2\xi_{xu} + u_x^2\varphi_{uu} \\
&\quad - u_x^3\xi_{uu} + u_{xx}\varphi_u - 2u_{xx}\xi_x - 3u_{xx}u_x\xi_u.
\end{aligned}$$

Puisque $F(x,u,u_x,u_{xx}) = u_{xx}$, on a $\mathsf{pr}^{(2)}\mathbf{v}.F = \varphi^{xx}$ et la condition du théorème 2 s'écrit

$$u_{xx} = 0 \implies \varphi^{xx}(x,u,u_x,u_{xx}) = 0$$

donc pour tout $(x,u,u_x) \in \mathbb{R}^3$,

$$\varphi_{xx}(x,u) + 2u_x\varphi_{xu}(x,u) - u_x\xi_{xx}(x,u) - 2u_x^2\xi_{xu}(x,u) + u_x^2\varphi_{uu}(x,u) - u_x^3\xi_{uu}(x,u) = 0.$$

On en déduit successivement

$$\varphi_{xx} = 0, \quad \xi_{uu} = 0, \quad 2\varphi_{xu} = \xi_{xx}, \quad 2\xi_{xu} = \varphi_{uu}$$

$$\varphi(x,u) = \mu(u) + x\nu(u), \quad \xi(x,u) = \pi(x) + u\rho(x)$$

$$2\nu'(u) = \pi''(x) + u\rho''(x), \quad 2\rho'(x) = \mu''(u) + x\nu''(u)$$

$$2\nu'(u) = \pi''(x) = 2c_1, \quad 2\rho'(x) = \mu''(u) = 2c_2$$

$$\xi(x,u) = c_1x^2 + c_3x + c_4 + c_2xu + c_5u, \quad \varphi(x,u) = c_2u^2 + c_6u + c_7 + c_1xu + c_8x$$

On obtient donc 8 champs de vecteurs dont le flot préserve les solutions.

$$\mathbf{v}_1 = \partial_x, \quad \mathbf{v}_2 = \partial_u, \quad \mathbf{v}_3 = x\partial_x, \quad \mathbf{v}_4 = u\partial_x, \quad \mathbf{v}_5 = x\partial_u, \quad \mathbf{v}_6 = u\partial_u,$$

$$\mathbf{v}_7 = x^2\partial_x + xu\partial_u, \quad \mathbf{v}_8 = xu\partial_x + u^2\partial_u.$$

En intégrant ces champs de vecteurs, on peut vérifier que leurs actions transforment bien les droites en droites. La composition de ces actions correspond en fait à l'action du groupe projectif linéaire $PGL(3)$ (de dimension 8) sur le plan (projectif).

2.2) Apprentissage d'opérateurs par réseau de neurones

Le problème de la discrétisation Comme dit en introduction, un réseau de neurones est une famille de fonctions $(\mathcal{G}_\theta)_{\theta\in\Theta}$ qui prennent en entrée un vecteur $\mathbf{e} \in \mathbb{R}^{d_e}$ et sortent un vecteur $\mathbf{s} \in \mathbb{R}^{d_s}$, paramétrées continuement par $\theta \in \Theta \subset \mathbb{R}^\nu$ que l'on choisit de sorte que $\mathcal{G}_\theta$ approche une fonction donnée. Généralement, il s'agit d'une alternance de couches linéaires $\mathbf{x} \mapsto W\mathbf{x}$ et de couches d'activation $\mathbf{x} = (x_i)_{i\in I} \mapsto (x_i^+)_{i\in I} = (\max(0,x_i))_{i\in I}$. La difficulté dans notre cas est que les entrées et les sorties ne sont pas des vecteurs mais des fonctions. Il faut donc trouver

une manière de les représenter informatiquement et trouver un réseau de neurones capable de les traiter.

La représentation des fonctions peut se faire de deux manières : soit par discrétisation, soit par projection sur une base. La première méthode est plus facile à utiliser car dans un contexte physique, on possède toujours une version échantillonnée du signal que l'on veut traiter. La deuxième méthode a l'avantage de fournir une expression exacte de la fonction qu'on veut traiter, ce qui peut être utile si on a besoin de calculer sa dérivée par exemple. La notion mathématique qui permet de combiner les deux est celle d'espace de Hilbert à noyau reproduisant.

Définition 7. *Soit un ensemble E. Un* ***espace de Hilbert à noyau reproduisant*** *sur E est un ensemble de fonctions de E dans $\mathbb{C}$ muni d'une structure d'espace de Hilbert telle que pour tout $x \in E$, la forme linéaire $f \mapsto f(x)$ est continue.*

Lintérêt d'un tel espace est le suivant : par théorème de Riesz, la forme linéaire $f \mapsto f(x)$ est représentée par un vecteur k_x, ie. $f(x) = (f|k_x)$ donc si on trouve une base Hilbertienne formée de vecteurs $(k_{x_n})_n$, les coefficients seront les évaluations de la fonction.

Un exemple de tel espace est l'ensemble des **fonctions à bande limitée**. Avec $\mathbb{T} = \mathbb{R}/2\pi\mathbb{Z}$, l'ensemble

$$B_K^d = \{f \in \mathrm{L}^2(\mathbb{T}^d, \mathbb{C}) | \mathrm{Supp}(\hat{f}) \subset [\![-K, K]\!]^d\},$$

où $K \in \mathbb{N}$, est de dimension finie donc à noyau reproduisant. Pour $n = 1$, on vérifie que

$$k_x(y) = \frac{1}{2\pi} \sum_{m=-K}^{K} e^{im(y-x)} = \frac{1}{2\pi} \frac{\sin\left((K+\frac{1}{2})(y-x)\right)}{\sin\left(\frac{1}{2}(y-x)\right)} = \frac{1}{2\pi} D_K(y-x) = k(y-x).$$

On a alors, en posant $x_p = \frac{2\pi p}{2K+1}$, pour tout $p, q \in [\![-K, K]\!]$,

$$(k_{x_p}|k_{x_q}) = k_{x_p}(x_q) = k(x_q - x_p) = \begin{cases} \frac{2K+1}{2\pi} & \text{si } p = q \\ 0 & \text{sinon} \end{cases}$$

donc $\left(\sqrt{\frac{2\pi}{2K+1}} k_{x_p}\right)_{p \in [\![-K,K]\!]}$ est une base orthonormée de B_K^1. Ainsi, pour tout $f \in B_K^1$,

$$f(x) = \frac{2\pi}{2K+1} \sum_{m=-K}^{K} f(x_m) k_{x_m}(x) = \frac{1}{2K+1} \sum_{m=-K}^{K} f(x_m) D_K(x - x_m). \tag{5}$$

Le raisonnement précédent se généralise à la dimension n : on a alors $k_x(y) = \prod_j k(y_j - x_j)$.

Un autre exemple intéressant est la version non périodique du précédent : soit

$$\mathcal{B}_K^d = \{f \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{C}) | \mathrm{Supp}(\hat{f}) \subset [-K, K]^d\},$$

où $K > 0$. On a alors, por tout $f \in \mathcal{B}_K^1$,

$$\begin{aligned} f(x) &= \frac{1}{2\pi} \overline{\mathcal{F}} \mathcal{F} f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} \mathcal{F} f(\omega) e^{i\omega x} d\omega = \frac{1}{2\pi} \int_{\mathbb{R}} \mathcal{F} f(\omega) \mathbb{1}_{[-K,K]} e^{i\omega x} d\omega \\ &= \frac{1}{2\pi} \int_{\mathbb{R}} \mathcal{F} f(\omega) \overline{\mathcal{F} k_x(\omega)} d\omega = (f|k_x) \end{aligned}$$

où $\mathcal{F} k_x(\omega) = \mathbb{1}_{[-K,K]} e^{-i\omega x}$ donc

$$\begin{aligned} k_x(y) &= \frac{1}{2\pi} \int_{\mathbb{R}} \mathbb{1}_{[-K,K]} e^{-i\omega x} e^{i\omega y} d\omega = \frac{1}{2\pi} \int_{-K}^{K} e^{-i\omega x} e^{i\omega y} d\omega \\ &= \frac{1}{\pi} \frac{\sin(K(y-x))}{y-x} = \frac{K}{\pi} \mathrm{sinc}(K(y-x)) = k(y-x) \end{aligned}$$

donc $\mathcal{B}_K^1$ est un espace de Hilbert à noyau reproduisant.

On pose $x_p = \frac{p\pi}{K}$. On remarque alors que $(\mathcal{F}k_{x_p})_{p\in\mathbb{Z}}$ est une base orthonormée de $\mathrm{L}^2([-K,K],\mathbb{C})$ et $\mathcal{F} : \mathcal{B}_K^1 \to \mathrm{L}^2([-K,K],\mathbb{C})$ est un isomorphisme isométrique (à un facteur multiplicatif près) donc $(k_{x_p})_{p\in\mathbb{Z}}$ est une base orthogonale de $\mathcal{B}_K^1$. Le même raisonnement que celui conduit sur le tore donne la formule bien connue (de Shannon)

$$f(x) = \sum_{m=-\infty}^{\infty} f(x_m)\,\mathrm{sinc}(K(x-x_m)) = \sum_{m=-\infty}^{\infty} f\left(\frac{m\pi}{K}\right)\mathrm{sinc}(Kx - m\pi).$$

Structure du réseau de neurones On pourrait échantillonner la fonction, appliquer un réseau de neurones classique à cet échantillonnage puis reconstruire la fonction à partir du vecteur de sortie, mais cela impose que toutes les fonctions traitées sont échantillonnées sur le même ensemble de points. Il est cependant très difficile de traiter des fonctions échantillonnées sur n'importe quel ensemble de points (cela néressiterait de trouver une manière canonique d'interpoler les fonctions, quelle que soit le maillage de points de l'échantillonnage). On va donc se restreindre aux fonctions échantillonnées sur des grilles mais on aimerait pouvoir traiter n'importe quelle résolution. Autrement dit, on veut un opérateur qui ne dépend pas de la manière dont est discrétisée la fonction. C'est ce que formalise la définition suivante, tirée de [BdBR$^+$23].

Définition 8. *Soit E, F deux espaces de Hilbert, $\mathcal{U} : E \to F$ un opérateur (par forcément linéaire), $\{\Phi^k\}_{k\in K} = \{(\varphi_j^k)_{j\in J}\}_{k\in K}$, $\{\Psi^k\}_{k\in K} = \{(\psi_j^k)_{j\in J}\}_{k\in K}$ des ensembles de familles de vecteurs de E et F respectivement. Soit des opérateurs linéaire*

$$T_{\Phi^k} : E \to \ell^2(J), \qquad T_{\Psi^k} : F \to \ell^2(J),$$

$$T_{\Phi^k}^\dagger : \ell^2(J) \to E, (c_j) \mapsto \sum_j c_j\varphi_j^k, \qquad T_{\Psi^k}^\dagger : \ell^2(J) \to F, (c_j) \mapsto \sum_j c_j\psi_j^k,$$

$$\mathfrak{u}_k : \ell^2(J) \to \ell^2(J),$$

*tels que $T_{\Phi^k}^\dagger T_{\Phi^k} = \mathrm{Id}$, $T_{\Psi^k}^\dagger T_{\Psi^k} = \mathrm{Id}$. Alors $\mathcal{U}$ est **invariant par changement de résolution** si pour tout $k \in K$,*

$$\mathcal{U} = T_{\Psi^k}^\dagger \circ \mathfrak{u}_k \circ T_{\Phi^k}.$$

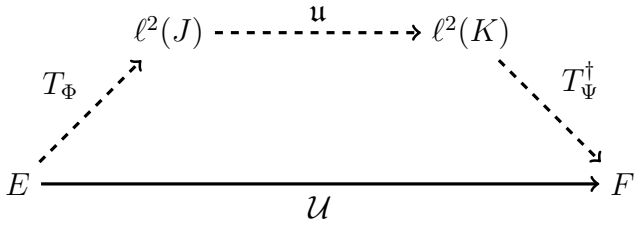

FIGURE 4 – Propriété d'invariance par changement de résolution

L'opérateur $\mathfrak{u}$ est la version discrétisée de l'opérateur $\mathcal{U}$. Il dépend de la façon dont est discrétisée la fonction.

On aimerait donc un réseau de neurones dont les couches vérifient l'invariance par changement de résolution. Un manière de faire est d'utiliser les réseaux de neurones de Fourier (FNO, introduits dans [LKA$^+$20]).

Dans de tels réseaux, une couche agit sur un espace de fonctions à bande limitée sur le tore. C'est une hypothèse raisonnable en faisant l'heuristique suivante : les fonction à traiter sont régulières donc leur transformée de Fourier décroit rapidement à l'infini. On peut donc

raisonnablement supposer que celle-ci est à support compact inclus dans $[-K,K]^{d_e}$ avec K à déterminer. L'expression d'une couche de Fourier est

$$C_R = \mathcal{F}^{-1} \circ R \circ \mathcal{F}, \text{ où } R \text{ est la multiplication par une fonction } R : \mathbb{Z}^{d_e} \to \mathbb{R}.$$

Cela revient à faire une convolution par $\hat{R}$. Le paramètre que l'on va faire varier ici est $\theta = R$. La version discrète est la même, en remplaçant la transformée de Fourier par une transformée de Fourier discrète $\mathcal{F}_n : \mathbb{R}^n \to \mathbb{R}^n$, appliquée la fonction discrétisée $(f(\frac{2\pi k}{n}))_{0\leq k\leq n-1}$:

$$C_{R,n} = \mathcal{F}_n^{-1} \circ R \circ \mathcal{F}_n, \text{ où } R \text{ est la multiplication par une fonction } R : \mathbb{Z}^{d_e} \to \mathbb{R}.$$

Théorème 4. *Sur $\mathbb{T}^d$, une couche de Fourier est invariante par changement de résolution au sens suivant : pour tout $f \in B_K^d$, pour tout $n \geq K$, $C_R(f) = C_{R,2n+1}(f)$.*

Démonstration. Il s'agit essentiellement de montrer qu'appliquer la transformée de Fourier revient à discrétiser et appliquer la transformée de Fourier discrète (et de même pour la transformation inverse).

Soit $f(x) = \sum_{k=-K}^{K} c_k e^{ikx} = \sum_{k=-\infty}^{\infty} c_k e^{ikx}$ où $c_k = 0$ si $|k| > K$. La transformée de Fourier est

$$\mathcal{F}f(k) = (f|e^{ik\cdot}) = 2\pi c_k.$$

La transformée de Fourier discrète de f est

$$\begin{aligned}
\mathcal{F}_n f(k) &= \sum_{l=-n}^{n} f\left(\frac{2\pi l}{2n+1}\right) e^{-2i\pi k\frac{l}{2n+1}} = \sum_{l=-n}^{n}\sum_{m=-\infty}^{\infty} c_m e^{im\frac{2\pi l}{2n+1}} e^{-2i\pi k\frac{l}{2n+1}} \\
&= \sum_{m=-\infty}^{\infty}\sum_{l=-n}^{n} c_m e^{\frac{2i\pi}{2n+1}l(m-k)} = \sum_{m=-\infty}^{\infty} c_m(2n+1)\mathbb{1}_{m\equiv k[2n+1]} \\
&= (2n+1)c_k \text{ car } 2n+1 \geq 2K+1
\end{aligned}$$

donc on obtient le même résultat dans les deux cas, à un facteur $\frac{2\pi}{2n+1}$ près.

De même, soit $(c_k)_{k\in\mathbb{Z}} \in \mathbb{C}^{\mathbb{Z}}$ où $c_k = 0$ si $|k| > K$. Alors

$$\begin{aligned}
\mathcal{F}^{-1}c(x) &= \frac{1}{2\pi}\sum_{k=-\infty}^{\infty} c_k e^{ikx} = \frac{1}{2\pi}\sum_{k=-K}^{K} c_k e^{ikx}, \\
\mathcal{F}_n^{-1}c(x) &= \frac{1}{2n+1}\sum_{l=-n}^{n}\left(\frac{1}{2n+1}\sum_{m=-\infty}^{\infty} c_m e^{2i\pi l\frac{m}{2n+1}}\right) D_n\left(x - \frac{2\pi l}{2n+1}\right) \\
&= \frac{1}{(2n+1)^2}\sum_{l=-n}^{n}\left(\sum_{m=-\infty}^{\infty} c_m e^{2i\pi l\frac{m}{2n+1}}\right)\left(\sum_{k=-n}^{n} e^{ik(x-\frac{2\pi l}{2n+1})}\right) \\
&= \frac{1}{(2n+1)^2}\sum_{m=-\infty}^{\infty}\sum_{k=-n}^{n}\sum_{l=-n}^{n} c_m e^{2i\pi l\frac{m-k}{2n+1}} e^{ikx} \\
&= \frac{1}{2n+1}\sum_{k=-n}^{n}\sum_{m=-\infty}^{\infty} c_m \mathbb{1}_{m\equiv k[2n+1]} e^{ikx} \\
&= \frac{1}{2n+1}\sum_{m=-\infty}^{\infty} c_m e^{ikx}
\end{aligned}$$

donc les deux donnent le même résultat, à un facteur $\frac{2n+1}{2\pi}$ près. Au total, les deux facteurs se compensent donc on a bien $C_R(f) = C_{R,2n+1}(f)$.

Couche d'activation et problèmes de repliement du spectre En ne mettant que des couches comme celle décrite précédemment, on obtient un réseau linéaire, qui ne peut représenter que des opérateurs linéaires. Il faut donc rajouter des couches non linéaires, dites d'activation : $f \mapsto f^+ = \max(0, f)$. Cependant, ces couches ne sont pas invariantes par changement de résolution car si f est à bande limitée, f^+ ne l'est pas forcément. L'opération $f \mapsto f^+$ fait apparaître de nouvelles fréquences donc la fonction f^+ est sous-échantillonnée : on observe un phénomène de **repliement du spectre** (ou **aliasing** en anglais).

Mathématiquement, on le voit en reprenant le calcul de la démonstration du théorème 4 : on a montré que

$$\mathcal{F}_n f(k) = (2n+1) \sum_{m=-\infty}^{\infty} c_m \mathbb{1}_{m \equiv k[2n+1]}$$

donc si $c_m \neq 0$ avec $|m| > K$, alors $m \equiv m'[2n+1]$ avec $|m'| \leq K$ et le coefficient c_m est interprété comme étant associé à la fréquence m' ce qui fausse la fonction que l'on croit échantillonner.

La solution pour atténuer le problème consiste à augmenter la résolution avant d'appliquer l'opération d'activation puis à la diminuer après. Cela permet de supprimer les coefficients de Fourier associés à des fréquences trop grandes plutôt que de les compter comme associés à des fréquences plus petites. Concrètement, la couche d'activation est

$$\sigma = \mathcal{D} \circ \sigma_{dis} \circ \mathcal{U}$$

où l'entrée est une fonction échantillonnée en $2K+1$ points et

- $\mathcal{U}$ est l'augmentation de la résolution : on sur-échantillonne la fonction à l'aide de la formule d'interpolation (5) ;
- σ_{dis} est la couche d'activation discrète : $(u_i)_i \mapsto (u_i^+)_i$;
- $\mathcal{D}$ est la diminution de la résolution : on applique une transformée de Fourier, on supprime les grandes fréquences (positives et négatives) puis on inverse la transformée de Fourier, ce qui revient à multiplier la transformée de Fourier par un filtre $\mathbb{1}_{[-K,K]}$, donc à convoler la fonction par D_K.

Le réseau effectivement implémenté pour les tests numériques est

$$C_{R_4} \circ \sigma \circ C_{R_3} \circ \sigma \circ C_{R_2} \circ \sigma \circ C_{R_1}$$

et le paramètre optimisé est $\theta = (R_1, R_2, R_3, R_4)$.

Ajout de la symétrie On veut maintenant améliorer les réseau de neurones décrit ci-dessus pour améliorer l'entraînement en tenant compte des symétries. On l'a vu, pour une EDP portant sur une fonction $u : X \to U$, une symétrie est l'action lisse d'un groupe de Lie G sur $X \times U$. Généralement, si $\mathcal{G}$ est l'opérateur qui aux paramètres associe la solution, la symétrie se traduit par une **relation d'équivariance**

$$\forall g \in G, \forall u, \mathcal{G}(g \cdot u) = g \cdot \mathcal{G}(u)$$

Que l'on peut reformuler en

$$\mathcal{G}(\exp(\varepsilon \mathbf{g}) \cdot u) = \exp(\varepsilon \mathbf{g}) \cdot \mathcal{G}(u)$$

c'est-à-dire, en dérivant en $\varepsilon = 0$ (ce qui permet d'avoir une relation linéaire),

$$D\mathcal{G}_u(\mathbf{g}(u)) = \mathbf{g}.\mathcal{G}(u) \,. \tag{6}$$

L'ajout des symétries peut se faire de plusieurs manières : en entrée, à l'intérieur ou en sortie.

Une première possibilité est d'augmenter la quantité des données fournies en entrée pendant l'entraînement. C'est ce qui est fait dans [BWW22]. L'idée est d'agrandir l'ensemble des données

$(\mathbf{e}_i, \mathbf{s}_i)_{i\in I}$ avec lequel notre réseau de neurones est entraîné en $(g_j \cdot \mathbf{e}_i, g_j \cdot \mathbf{s}_i)_{i\in I, j\in J}$ où les $g_j \in G$ sont choisis ou tirés au hasard. On espère qu'en entraînant sur $g \cdot u$ pour suffisemment de $g \in G$, le réseau sera bon quel que soit g.

Une deuxième possibilité est de construire directement des réseaux de neurones $\mathcal{G}_\theta$ équivariants. Ce n'est en fait pas évident. Il existe des constructions théoriques, comme décrit dans [CW16] et [KT18], qui ressemblent à des convolutions sur des groupes mais elles sont difficiles à mettre en pratique car elles dépendent beaucoup du groupe G et parce qu'il n'existe pas de manière canonique de discrétiser un groupe de Lie général (par exemple $SO(3)$).

La troisième méthode, adoptée ici, est d'ajouter un terme de perte dû à l'équivariance. Le plus simple est d'utiliser la version linéarisée (6) et d'ajouter un terme $\|D\mathcal{G}_u(\mathbf{g}(u)) - \mathbf{g}.\mathcal{G}(u)\|_2$ à ℓ pour chaque vecteur $\mathbf{g}$ d'une base de $\mathfrak{g}$.

Exemple Nous allons prendre un exemple simple : on considère l'équation de la chaleur $\partial_t u - \Delta u = 0$ où $u : \mathbb{R}^+ \times \mathbb{T}^2 \to \mathbb{R}$ et on considère l'opérateur $\mathcal{G} : u(t = 0) \to u(t = 0.1)$.

La théorie précédente fonctionne si on considère un champ de vecteurs dont l'action est une symétrie et dont le flot laisse stable $\{0\} \times \mathbb{T}^2 \times \mathbb{R}$ et $\{0.1\} \times \mathbb{T}^2 \times \mathbb{R}$. C'est le cas de la translation dans l'espace ∂_x. On peut observer figure 5 la différence de comportement du réseau de neurones avec et sans terme dû à l'invariance par translation dans l'espace de l'équation.

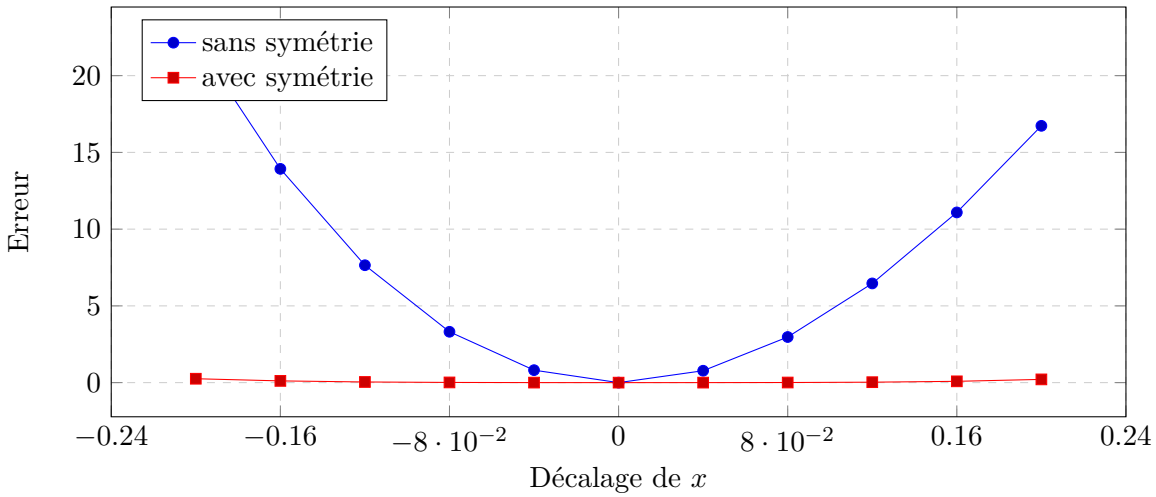

FIGURE 5 – Performance du réseau en fonction du décalage spatial de la fonction initiale. Il est meilleur avec ajout du terme de symétrie (100 fois meilleur avec un décalage de ± 0.2).

Cependant, avec cette condition, on restreint beaucoup les symétries qu'on s'autorise. On peut en fait s'en sortir même quand G agit sur $\mathbb{R}^+ \times \mathbb{T}^2 \times \mathbb{R}$ en entier. On la transforme en une action sur $\{0\} \times \mathbb{T}^2 \times \mathbb{R}$ de la manière suivante : on part de $u_0 : \mathbb{T}^2 \to \mathbb{R}$, on considère $u : \mathbb{R}^+ \times \mathbb{T}^2 \to \mathbb{R}$ solution de $u(t = 0) = u_0, \partial_t u - \Delta u = 0$, on fait agir G sur le graphe de u ce qui donne une nouvelle fonction $\tilde{u}$ et enfin on restreint $\tilde{u}$ à $\{0\} \times \mathbb{T}^2$. La version infinitésimale d'une telle action, si G correspond au flot d'un champ de vecteurs, est donnée par le théorème suivant.

Théorème 5. *Soit $X \subset \mathbb{R}^p$ ouvert. On considère un champ de vecteurs lisse sur $\mathbb{R}^p \times \mathbb{R}^q$*

$$\mathbf{v} = \sum_i \xi_i(x, u)\partial_{x^i} + \sum_\alpha \varphi_\alpha(x, u)\partial_{u^\alpha} = \xi(x, u) \cdot \nabla_x + \varphi(x, u) \cdot \nabla_u.$$

Le flot de $\mathbf{v}$ induit une action locale de $\mathbb{R}$ sur $\mathbb{R}^p \times \mathbb{R}^q$ donc une action locale sur l'ensemble des fonctions lisses $f : X \to \mathbb{R}^q$. Pour tout $x \in X$, on a alors

$$\left.\frac{d}{d\varepsilon}\right|_{\varepsilon=0} (\varepsilon \cdot f)\,(x) = \varphi(x, f(x)) - Jac_f(x)\xi(x, f(x)).$$

Démonstration. Soit le flot $\varepsilon \cdot (x,u) = (\Xi_\varepsilon(x,u), \Phi_\varepsilon(x,u))$. Alors

$$\varepsilon \cdot f\ (x) = (\Phi_\varepsilon \circ (\mathbb{1} \times f)) \circ (\Xi_\varepsilon \circ (\mathbb{1} \times f))^{-1}$$

On note

$$F(\varepsilon,x) = (\Phi_\varepsilon \circ (\mathbb{1} \times f))(x) = \Phi_\varepsilon(x, f(x)),$$
$$G(\varepsilon,x) = (\Xi_\varepsilon \circ (\mathbb{1} \times f))(x) = \Xi_\varepsilon(x, f(x)),$$
$$H(\varepsilon,x) = (\Xi_\varepsilon \circ (\mathbb{1} \times f))^{-1}(x).$$

On a $\varepsilon \cdot f\ (x) = F(\varepsilon, H(\varepsilon,x))$ donc

$$\frac{d}{d\varepsilon}\ \varepsilon \cdot f\ (x) = \partial_\varepsilon F(\varepsilon, H(\varepsilon,x)) + Jac_{F,x}(\varepsilon, H(\varepsilon,x))\partial_\varepsilon H(\varepsilon,x).$$

Par définition, $F(0,x) = f(x)$, $G(0,x) = x$, $H(0,x) = x$. De plus $G(\varepsilon, H(\varepsilon,x)) = x$ donc en dérivant selon ε,

$$\partial_\varepsilon G(\varepsilon, H(\varepsilon,x)) + Jac_{G,x}(\varepsilon, H(\varepsilon,x))\partial_\varepsilon H(\varepsilon,x) = 0$$
$$\partial_\varepsilon H(\varepsilon,x) = -Jac_{G,x}(\varepsilon, H(\varepsilon,x))^{-1}\partial_\varepsilon G(\varepsilon, H(\varepsilon,x))$$

donc

$$\frac{d}{d\varepsilon}\varepsilon \cdot f\ (x) = \partial_\varepsilon F(\varepsilon, H(\varepsilon,x)) - Jac_{F,x}(\varepsilon, H(\varepsilon,x))Jac_{G,x}(\varepsilon, H(\varepsilon,x))^{-1}\partial_\varepsilon G(\varepsilon, H(\varepsilon,x)).$$

Finalement, avec $\varepsilon = 0$,

$$\begin{aligned}\frac{d}{d\varepsilon}\Big|_{\varepsilon=0}(\varepsilon \cdot f)\ (x) &= \partial_\varepsilon F(0,x) - Jac_{F,x}(0,x)Jac_{G,x}(0,x)^{-1}\partial_\varepsilon G(0,x) \\ &= \varphi(x, f(x)) - Jac_f(x)\xi(x, f(x)).\end{aligned}$$

En appliquant ce théorème aux points tels que $t = 0$ et $t = 0.1$ et en utilisant le fait que $\partial_t u = \Delta u$, on obtient sur notre opérateur $\mathcal{G}$ la relation

$$D\mathcal{G}_{u_0}(\Delta u_0) = \Delta(\mathcal{G}(u_0))$$

donc on peut rajouter un terme $\|D\mathcal{G}_{u_0}(\Delta u_0) - \Delta(\mathcal{G}(u_0))\|_2$ à ℓ. On peut observer figure 6 la différence de comportement du réseau de neurones avec et sans terme dû à l'invariance par translation dans le temps de l'équation.

3) Déroulement du stage

Le stage, de deux mois, s'est déroulé au SAM (Seminar for applied Mathematics), à l'École polytechnique fédérale de Zürich (ETH Zürich), sous la supervision du professeur Siddhartha Mishra. J'ai surtout interagi avec Emmanuel de Bezenac, post-doctorant en informatique dans ce même laboratoire.

J'ai passé les trois voire quatre premières semaines à lire et comprendre le livre de Olver sur les symétries des EDP [Olv93]. En réalité le livre va bien plus loin que ce qui est décrit ici : il explique concrètement comment simplifier voir résoudre des EDP grâce aux symétries exhibées et donne une formulation rigoureuse de théorème de Noether dans le cadre des EDP d'Euler-Lagrange obtenues par minimisation de l'intégrale d'un lagrangien. Je me suis surtout attardé sur les deux premiers chapitres qui m'étaient utiles pour le stage.

J'ai ensuite passé les semaines suivantes à lire un certain nombre de papiers sur les symétries des réseaux de neurones, pas forcément en lien avec les EDP. La première manifestation de l'utilisation des symétries a été l'invention des réseaux de neurones convolutifs, pour l'analyse

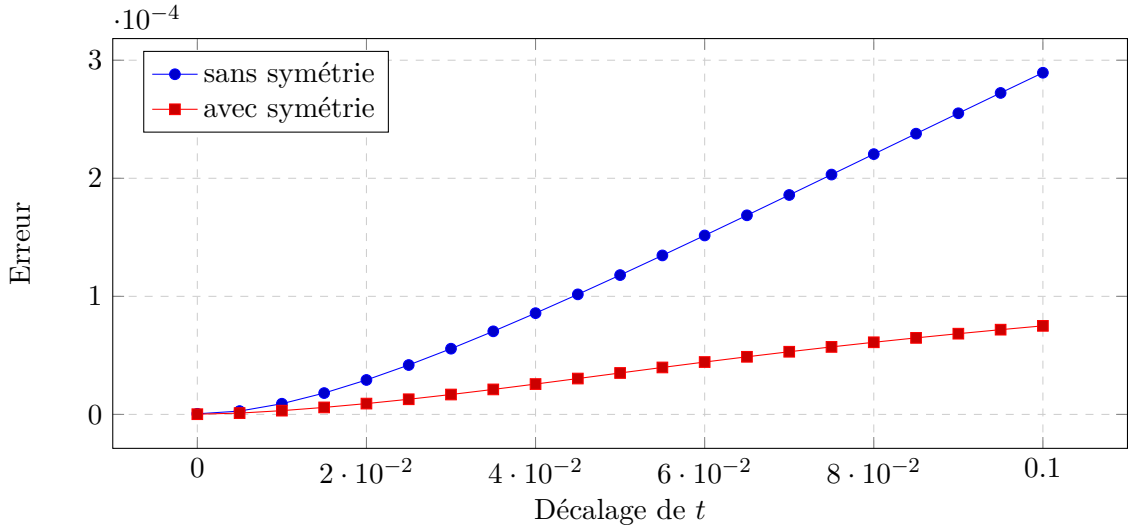

FIGURE 6 – Performance du réseau en fonction du décalage temporel de la fonction initiale. Il est meilleur avec ajout du terme de symétrie (4 fois meilleur avec un décalage de 0.1).

d'image. En effet, les convolutions commutent avec les translations (ce sont les seules opérations linéaires à respecter cette propriété) et a priori, pour un réseau analysant des images, deux images liées par une tranlsation représentent le même chose. Certaines symétries liées à des groupes de permutations interviennent également dans la structure des réseaux faits pour le traitement du langage naturel, l'échange de certains mots ne changeant pas le sens de la phrase.

Enfin, j'ai passé les dernières semaines à comprendre et améliorer un code écrit par Emmanuel pour tester l'efficacité effective de l'ajout d'un terme de perte lié à l'équivariance infinitésimale du réseau de neurones. Les tests étaient concluants dans le cas où le réseau était entraîné sur une seule fonction. Nous n'avons pas eu le temps de tester avec plusieurs fonctions à la fois.

Références

[BdBR+23] F. BARTOLUCCI, E. DE BÉZENAC, B. RAONIĆ, R. MOLINARO, S. MISHRA & R. ALAIFARI – « Are neural operators really neural operators ? frame theory meets operator learning », **arXiv preprint arXiv :2305.19913** (2023).

[BWW22] J. BRANDSTETTER, M. WELLING & D. E. WORRALL – « Lie point symmetry data augmentation for neural pde solvers », in **International Conference on Machine Learning**, PMLR, 2022, p. 2241–2256.

[CW16] T. COHEN & M. WELLING – « Group equivariant convolutional networks », in **International conference on machine learning**, PMLR, 2016, p. 2990–2999.

[KT18] R. KONDOR & S. TRIVEDI – « On the generalization of equivariance and convolution in neural networks to the action of compact groups », in **International Conference on Machine Learning**, PMLR, 2018, p. 2747–2755.

[LKA+20] Z. LI, N. KOVACHKI, K. AZIZZADENESHELI, B. LIU, K. BHATTACHARYA, A. STUART & A. ANANDKUMAR – « Fourier neural operator for parametric partial differential equations », **arXiv preprint arXiv :2010.08895** (2020).

[Olv93] P. J. OLVER – **Applications of lie groups to differential equations**, Springer, 1993.