

Méthodes d'apprentissage pour les problèmes de fermeture et stabilisation en EDP

L. Bois^{1,2} M. Bestard^{1,2} E. Franck^{1,2}
L. Navoret^{1,2} V. Vigon^{1,2}

¹ Institut de Recherche Mathématique Avancée, UMR 7501,
Université de Strasbourg et CNRS, 7 rue René Descartes,
67000 Strasbourg, France

² INRIA Nancy-Grand Est, équipe-projet TONUS, Strasbourg, France

Séminaire Math@HPC, École Polytechnique

① Contexte physique et mathématique

② Modèles réduits pour l'équation de Vlasov

Génération des données

Réseaux de neurones

Résultats

Modèle fluide

Difficultés : stabilité et coût calcul

③ Fermeture et bases réduites

Bases réduites pour les équations hyperboliques

Fermeture et contrôle optimal

④ Stabilisation de schéma LBM

Schéma LBM

Stabilisation et contrôle optimal

Problème de fermeture I

- On considère une EDP du type :

$$\partial_t u(t, x_1, \dots, x_n) + \nabla_{x_1, \dots, x_n} f(u; x_1, \dots, x_n) = 0$$

- Afin de réduire le coût de calcul, on utilise des méthodes de réduction de dimension qui **projetten sur une base de fonction dépendant d'une partie des variables**.
- Deux formalismes** : Galerkin ou **méthodes aux moments**.
- Méthode de Galerkin** :

- on propose

$$u(t, x_1, \dots, x_n) = u_N(t, x_1, \dots, x_n) = \sum_1^N \alpha_i(t, x_1, \dots, x_k) \phi(x_{k+1}, \dots, x_n)$$

avec

$$\alpha_i = \int \phi_i(x_{k+1}, \dots, x_n) u(t, x_1, \dots, x_n) dx_{k+1}, \dots, dx_n$$

- on obtient un modèle réduit sur les α_i en résolvant :

$$\min_{(\alpha_1, \dots, \alpha_N)} \|\partial_t u_N(t, x_1, \dots, x_n) + \nabla_{x_1, \dots, x_n} f(u_N; x_1, \dots, x_n)\|^2$$

- si le modèle est linéaire on obtient une équation linéaire :

$$\frac{d}{dt} \alpha(t, x_1, \dots, x_k) + \nabla_{x_1, \dots, x_k} (A \alpha(t, x_1, \dots, x_k)) = 0$$

où les matrices dépendront de la base obtenue et du modèle ;

- si le modèle est non-linéaire, on obtient une équation non-linéaire

$$\frac{d}{dt} \alpha(t, x_1, \dots, x_k) + P(\nabla_{x_1, \dots, x_n} f(u_N; x_1, \dots, x_n)) = 0,$$

avec P le projecteur sur la base.

- Une fermeture est une fonction $\hat{f}(\alpha)$ qui approche $P(\nabla_{x_1, \dots, x_n} f(u_N; x_1, \dots, x_n))$.**

Problème de fermeture II

- **Méthode aux moments :**

① on se donne **N moments** :

$$\alpha_i = \langle \phi_i, u \rangle_{L^2} = \int \phi_i(x_{k+1}, \dots, x_n) u(t, x_1, \dots, x_n) dx_{k+1}, \dots, dx_n$$

② on écrit un modèle sur les moments :

$$\partial_t \int (v \phi_i(x_{k+1}, \dots, x_n)) dx_{k+1}, \dots, dx_n + \int (v \cdot \nabla \phi_i(x_{k+1}, \dots, x_n)) dx_{k+1}, \dots, dx_n = 0$$

$$\partial_t \alpha(t) + \int (v \nabla_{x_1, \dots, x_n} \phi_i(x_{k+1}, \dots, x_n)) dx_{k+1}, \dots, dx_n = 0.$$

③ Comment obtenir le modèle ? On choisit un ansatz de solution de la forme $v(\alpha_1, \dots, \alpha_N, x_{k+1}, \dots, x_n)$. Ensuite, on incorpore cette distribution dans l'équation et on calcule les moments, ce qui donne **le modèle fermé** :

$$\partial_t \alpha + \hat{f}(\nabla_{x_1, \dots, x_n}; \alpha) = 0$$

④ Pour déterminer l'ansatz, on peut le construire par analyse asymptotique ou par un processus d'optimisation :

$$\min_v \eta(v), \quad \text{sous contrainte } \langle \phi_i, v \rangle_{L^2} = \alpha_i, \forall 1 \leq i \leq N$$

⑤ La méthode de Galerkin et des moments coïncident pour $\eta(v) = \frac{1}{2} v^2$ et des EDP linéaires.

① Contexte physique et mathématique

② Modèles réduits pour l'équation de Vlasov

Génération des données

Réseaux de neurones

Résultats

Modèle fluide

Difficultés : stabilité et coût calcul

③ Fermeture et bases réduites

Bases réduites pour les équations hyperboliques

Fermeture et contrôle optimal

④ Stabilisation de schéma LBM

Schéma LBM

Stabilisation et contrôle optimal

Équation de Vlasov

- En physique des plasmas on résout l'équation de **Vlasov-Poisson** sur la distribution de particules $f(t, x, v)$:

$$(V) \quad \partial_t f + v \partial_x f + E(x) \partial_v f = \frac{1}{\varepsilon} (M(\rho, u, T) - f),$$

avec $M(\rho, u, T)$ la fonction Maxwellienne, ε le nombre de Knudsen et E le champ électrique, donné par

$$(P) \quad E = -\partial_x \phi, \quad \partial_{xx} \phi = \rho - \int \rho dx.$$

- On utilise souvent des méthodes de réduction sur l'espace des vitesses afin de se ramener à un système spatial.
- On choisit comme moments $\left\{ 1, v, \frac{1}{2} v^2 \right\}$, et on écrit l'équation sur les moments :

$$\begin{cases} \partial_t \rho + \partial_x(\rho u) = 0, \\ \partial_t(\rho u) + \partial_x(\rho u^2 + \rho T) = -E \rho, \\ \partial_t w + \partial_x(w u + \rho u T + q) = -E \rho u, \end{cases}$$

$$\text{avec } \rho = \int_{\mathbb{R}} f dv, \rho u = \int_{\mathbb{R}} f v dv, w = \frac{1}{2} \rho u^2 + \frac{1}{2} p = \int_{\mathbb{R}} f \frac{v^2}{2} dv,$$

$$\text{et } p = \rho T = \frac{1}{2} \int_{\mathbb{R}} f (u - v)^2 dv \text{ et } q = \frac{1}{2} \int_{\mathbb{R}} f (u - v)^3 dv.$$

Fermeture

- Le **flux de chaleur** q , donné par

$$q(x, t) = \frac{1}{2} \int f(x, v, t) (v - u(x, t))^3 dv,$$

reste indéterminé. On a besoin d'une **fermeture qui donne** $q = C(\rho, u, T)$ afin d'obtenir un système résoluble.

- Stratégie classique** : on choisit un ansatz.

1er cas $f(t, x, v) = M(\rho, u, T)$ est obtenu par l'asymptotique $\varepsilon = 0$ ou par minimisation de l'entropie

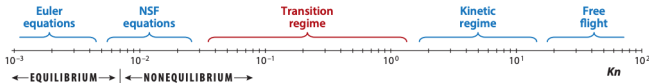
$$\eta(f) = f \ln f - f,$$

On obtient **$q = 0$ et les équations d'Euler.**

2ème cas $f(t, x, v) = M(\rho, u, T) + \varepsilon g(t, x, v)$ (obtenu par l'asymptotique $\varepsilon \rightarrow 0$). On obtient **$q = -\frac{3}{2} p \partial_x T$ et les équations de Navier-Stokes.**

3ème cas fermeture non locale en espace pour des configurations particulières.

- Idée** : proposer une fermeture en utilisant **l'apprentissage profond** et des simulations cinétiques.
- Régime et validité des fermetures (papier de M. Torrilhon et al.) :



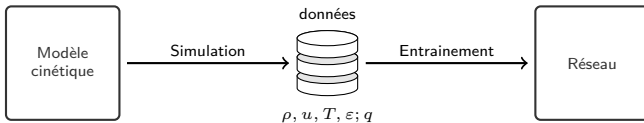
Fermeture basée sur les données

- Deux phases de calcul :

Fermeture basée sur les données

- Deux phases de calcul :

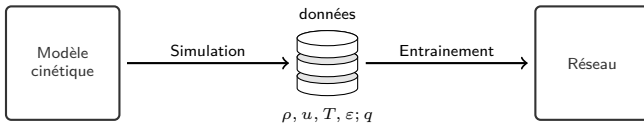
Phase hors-ligne



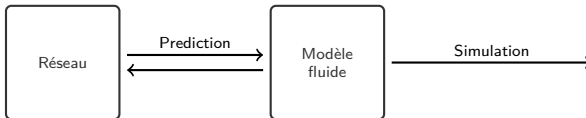
Fermeture basée sur les données

- Deux phases de calcul :

Phase hors-ligne



Phase en ligne



- 1 Contexte physique et mathématique
- 2 Modèles réduits pour l'équation de Vlasov
 - Génération des données
 - Réseaux de neurones
 - Résultats
 - Modèle fluide
 - Difficultés : stabilité et coût calcul
- 3 Fermeture et bases réduites
 - Bases réduites pour les équations hyperboliques
 - Fermeture et contrôle optimal
- 4 Stabilisation de schéma LBM
 - Schéma LBM
 - Stabilisation et contrôle optimal

Génération des données

- On considère une **fermeture globale** : $\hat{q}(\rho(.), u(.), T(.), \varepsilon)$

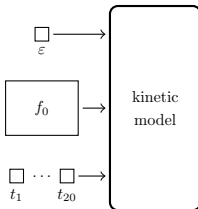
Les données sont calculées avec le modèle cinétique

kinetic
model

Génération des données

- On considère une **fermeture globale** : $\hat{q}(\rho(.), u(.), T(.), \varepsilon)$

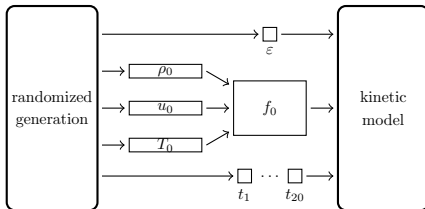
On prend un nombre de Knudsen ε , une condition initiale f_0 , et 20 temps $t_1, \dots, t_{20}$.



Génération des données

- On considère une **fermeture globale** : $\hat{q}(\rho(\cdot), u(\cdot), T(\cdot), \varepsilon)$

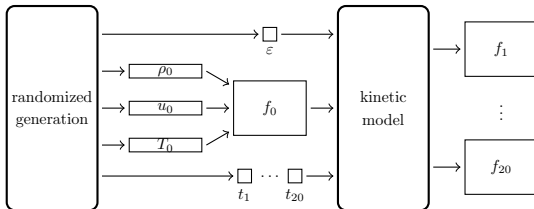
Les quantités sont générés aléatoirement. En particulier, $\varepsilon \in [0.01, 1]$.



Génération des données

- On considère une **fermeture globale** : $\hat{q}(\rho(\cdot), u(\cdot), T(\cdot), \varepsilon)$

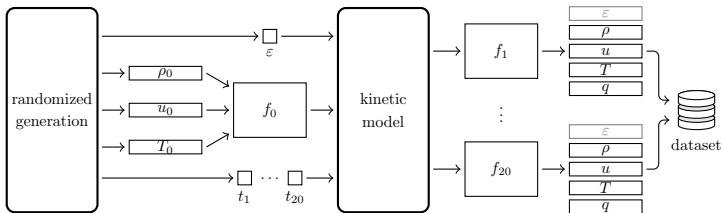
Le modèle donne $f_1, \dots, f_{20}$ aux temps $t_1, \dots, t_{20}$ respectivement.



Génération des données

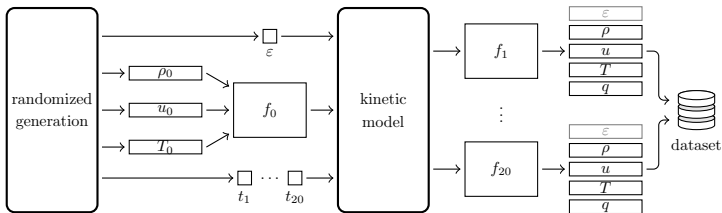
- On considère une **fermeture globale** : $\hat{q}(\rho(\cdot), u(\cdot), T(\cdot), \varepsilon)$

Finalement les fonctions ρ , u , T et q sont calculées et stockées avec ε .



Génération des données

- On considère une **fermeture globale** : $\hat{q}(\rho(\cdot), u(\cdot), T(\cdot), \varepsilon)$

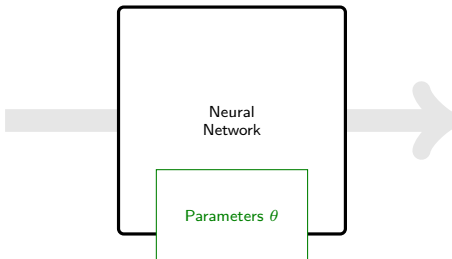


Deux jeux de données ($500 \times 20 = 10\,000$) sont générés :
un pour l'entraînement, et un pour la validation.

- 1 Contexte physique et mathématique
- 2 Modèles réduits pour l'équation de Vlasov
 - Génération des données
 - Réseaux de neurones
 - Résultats
 - Modèle fluide
 - Difficultés : stabilité et coût calcul
- 3 Fermeture et bases réduites
 - Bases réduites pour les équations hyperboliques
 - Fermeture et contrôle optimal
- 4 Stabilisation de schéma LBM
 - Schéma LBM
 - Stabilisation et contrôle optimal

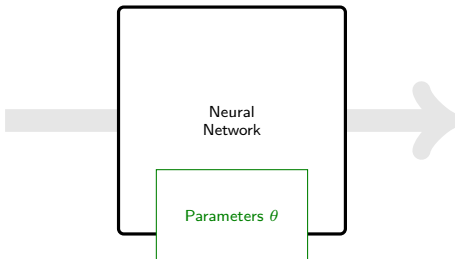
Entrainement

Un *réseau de neurones* une fonction particulière
non-linéaire dépendante d'un grand nombre de **paramètres**.



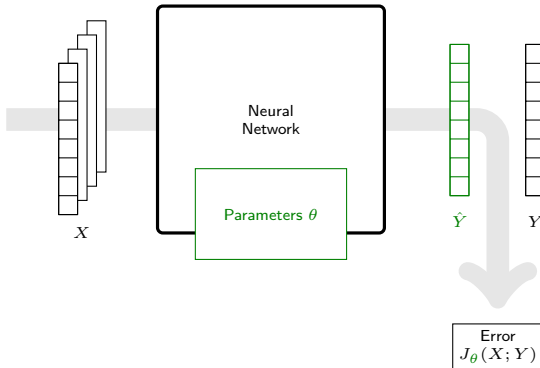
Entrainement

L'*entrainement* d'un réseau est un processus d'optimisation
qui permet d'ajuster les paramètres.



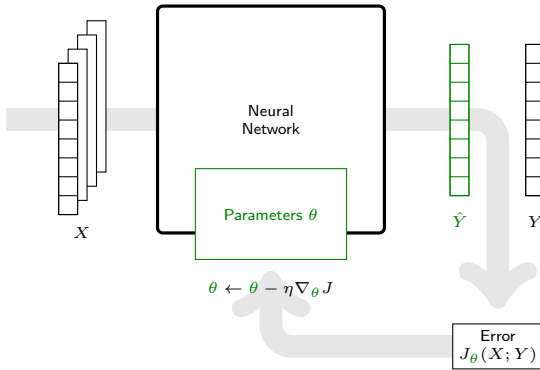
Entrainement

Pour chaque couple (X, Y) dans l'ensemble de données,
une erreur est calculée entre la prédiction $\hat{Y}$ et la solution Y .



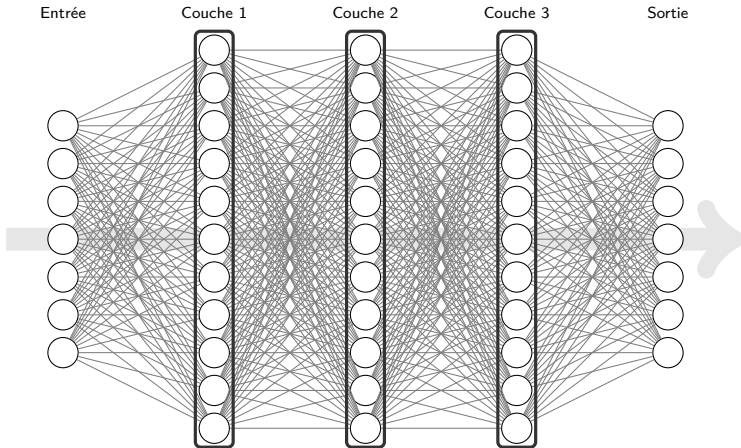
Entrainement

L'erreur permet d'ajuster les paramètres avec un algorithme de gradient.



Structure générale

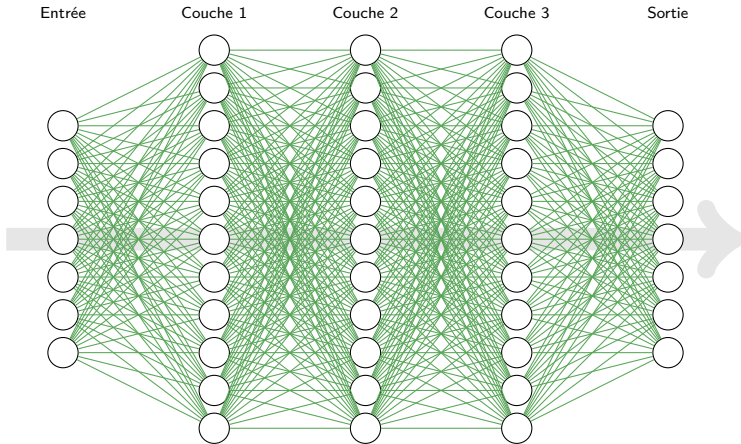
L'entrée passe successivement dans des *couches* cachées.



- En analyse numérique, on utilise en général des fonctions paramétriques obtenues en sommant des fonctions simples ou données : $f(x) = \sum_{i=1}^N \theta_i \phi_i(x)$.
- Un réseau de neurone est obtenu par **composition de fonctions simples**.

Structure générale

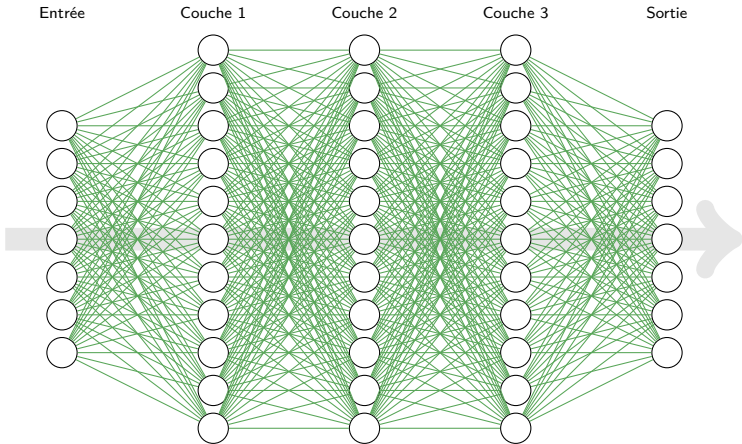
Chaque arête contient un **paramètre**
qui peut être ajusté dans l'entraînement.



- En analyse numérique, on utilise en général des fonctions paramétriques obtenues en sommant des fonctions simples ou données : $f(x) = \sum_{i=1}^N \theta_i \phi_i(x)$.
- Un réseau de neurone est obtenu par **composition de fonctions simples**.

Structure générale

Chaque neurone applique une fonction
d'activation non-linéaire à son résultat.



- En analyse numérique, on utilise en général des fonctions paramétriques obtenues en sommant des fonctions simples ou données : $f(x) = \sum_{i=1}^N \theta_i \phi_i(x)$.
- Un réseau de neurone est obtenu par **composition de fonctions simples**.

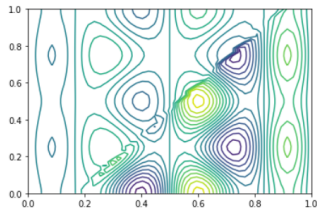
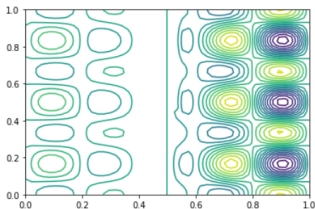
Réseaux convolutifs I

- **Réseaux convolutifs** : étape linéaire basée sur des **convolutions contre des petits noyaux**.
- **Avantages** : **Réseau de neurones creux**. Très efficace pour les **données structurées** (données spatiales sur grilles cartésiennes, images, signaux temporelles etc).
- **Exemple** : détection de discontinuités dans des fonctions.
On génère 1000 fonctions avec des modes de Fourier choisis aléatoirement. On ajoute une discontinuité aléatoire. 890 (entraînement), 110 (pour monitorer le sur-apprentissage).
- **But** : classifier les fonctions avec et sans discontinuité :

$$f(x, y) = \left(\sum_{i=1}^5 \alpha_i \cos(2\pi k_i x) \sin(2\pi l_i y) \right) Disc(x, y)$$

avec k_i, l_i aléatoire entre 0 et 4, $\alpha_i \in [-1, 1]$ et $Disc$ une discontinuité aléatoire. Maillage : 48×48 .

Réseaux convolutifs II



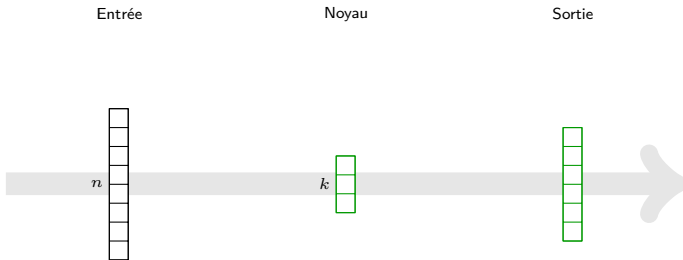
Test : 200 fonctions supplémentaires.

- Réseau totalement connecté (4 couches) 1 : 566 000 poids. Efficacité $\approx 0.45 - 0.6$.
- Réseau totalement connecté (4 couches) 2 : 230 000 poids. Efficacité $\approx 0.45 - 0.6$.
- CNN (3 couches Conv, 3 couches denses) : 41 300 poids. Efficacité $\approx 0.9 - 0.95$.

Réseaux convolutifs III

Base

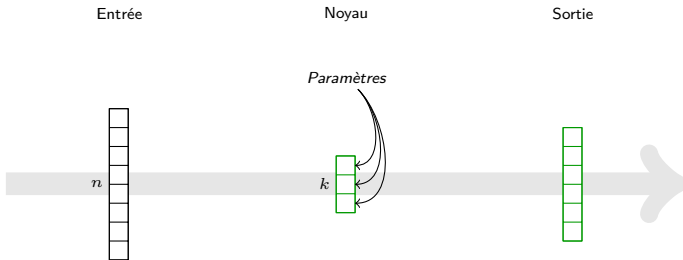
Couche convolutive :



Réseaux convolutifs III

Base

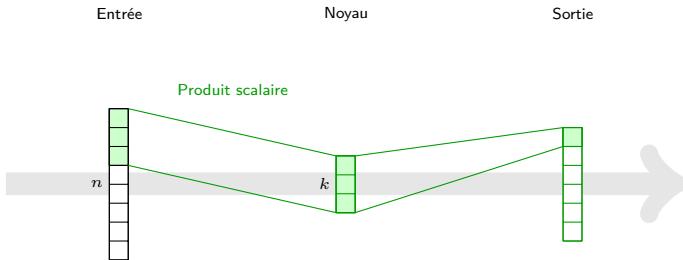
Les paramètres sont localisés dans le noyau.



Réseaux convolutifs III

Base

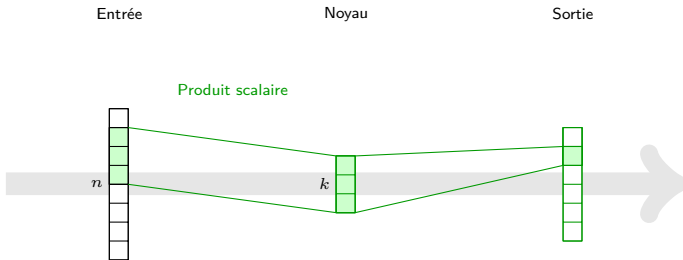
Le produit avec le noyau applique un filtre.



Réseaux convolutifs III

Base

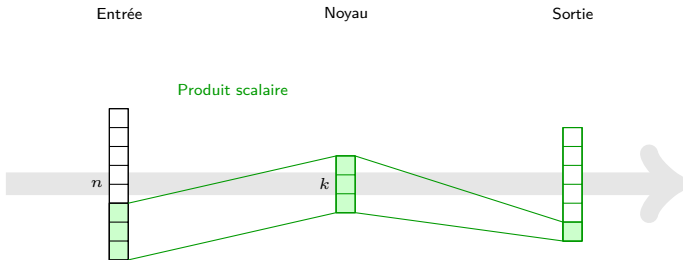
Le produit avec le noyau applique un filtre.



Réseaux convolutifs III

Base

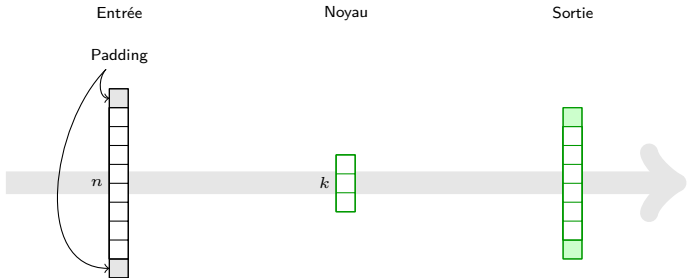
Le produit avec le noyau applique un filtre.



Réseaux convolutifs III

Padding

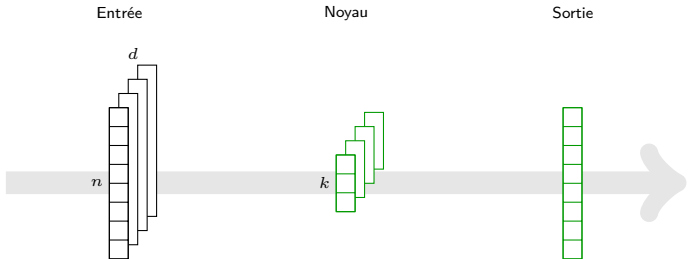
Pour obtenir une sortie de même taille : "padding".



Réseaux convolutifs III

Entrée à plusieurs vecteurs

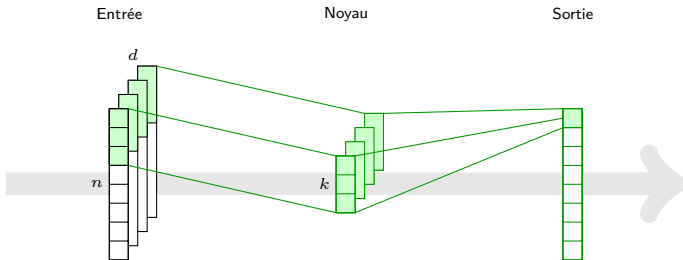
Entrées multiples : couleurs dans une image, ...



Réseaux convolutifs III

Entrée à plusieurs vecteurs

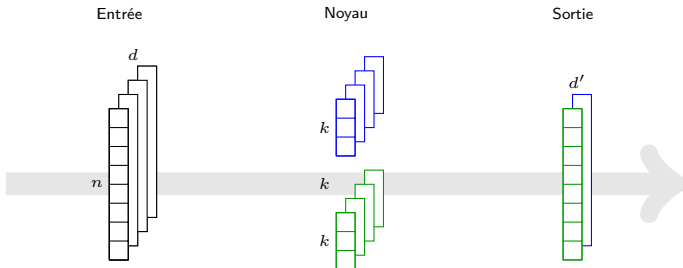
La sortie est donnée par la somme des produits scalaires.



Réseaux convolutifs III

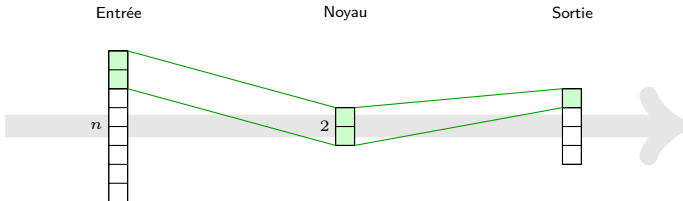
Sortie à plusieurs vecteurs

On peut appliquer plusieurs filtres (noyau).



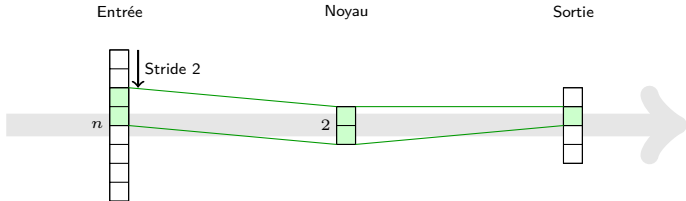
"Sous- et sur-échantillonnage"

On utilise une convolution différentielle pour diminuer la taille du signal.



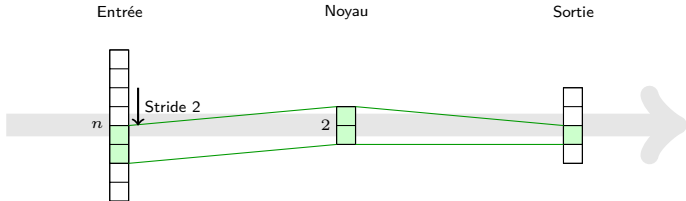
"Sous- et sur-échantillonnage"

Noyau de taille 2 avec un *pas* de 2



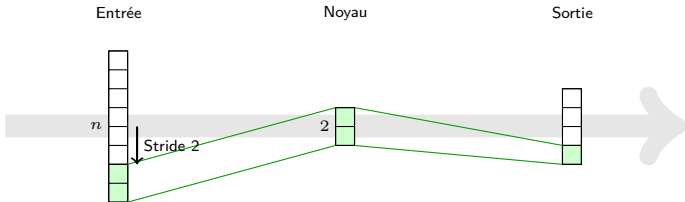
“Sous- et sur-échantillonnage”

En négligeant les bords, sortie = entrée / pas



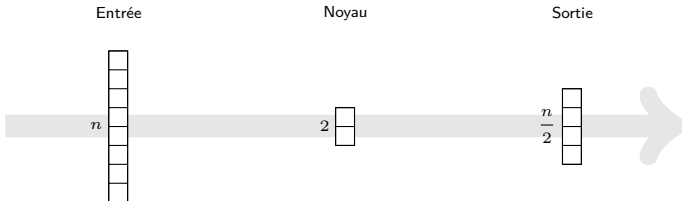
“Sous- et sur-échantillonnage”

En négligeant les bords, sortie = entrée / pas



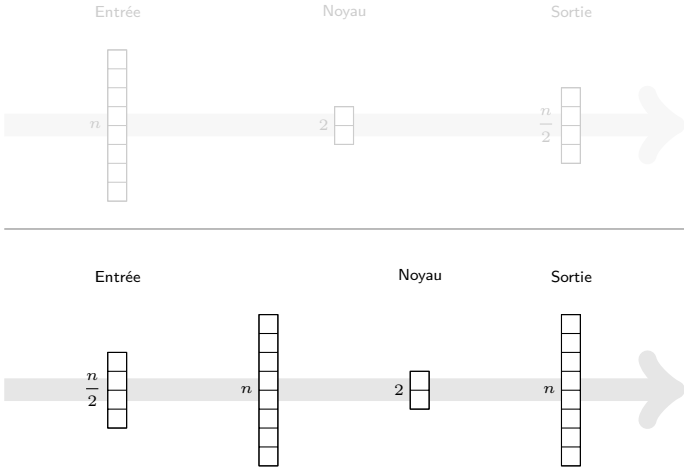
"Sous- et sur-échantillonnage"

Ici la sortie est deux fois moins échantillonnée que l'entrée.



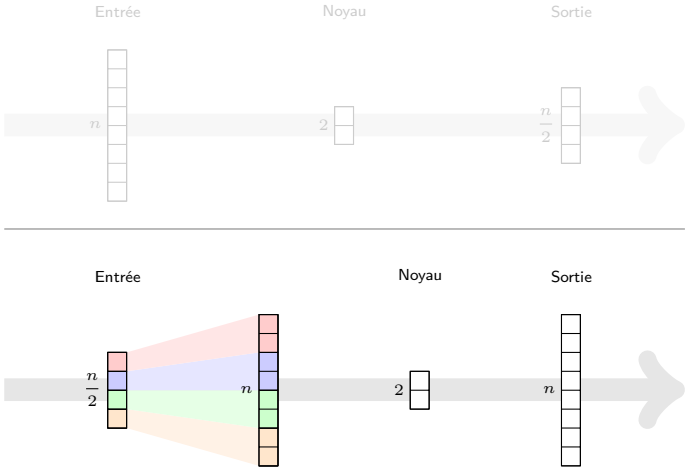
“Sous- et sur-échantillonnage”

L'opération inverse :



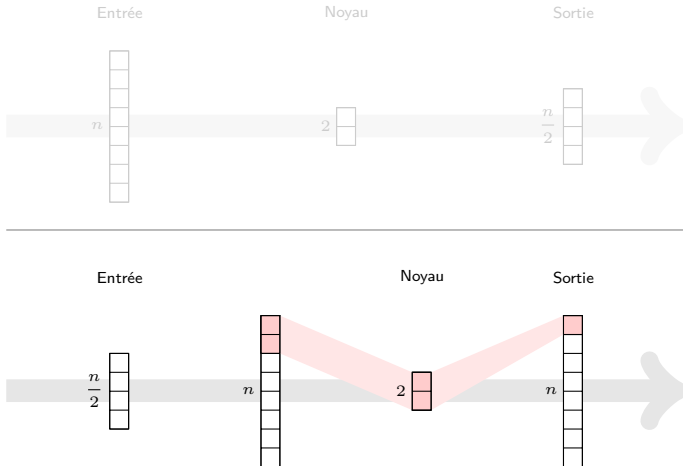
“Sous- et sur-échantillonnage”

on duplique les coefficients.



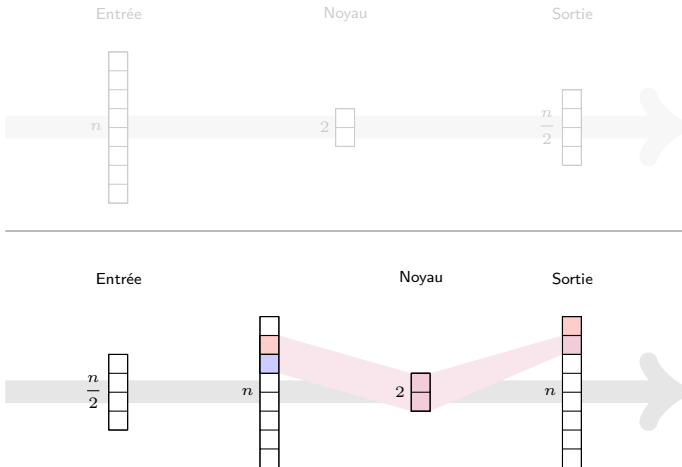
“Sous- et sur-échantillonnage”

On applique une convolution de taille 2 pour régulariser.



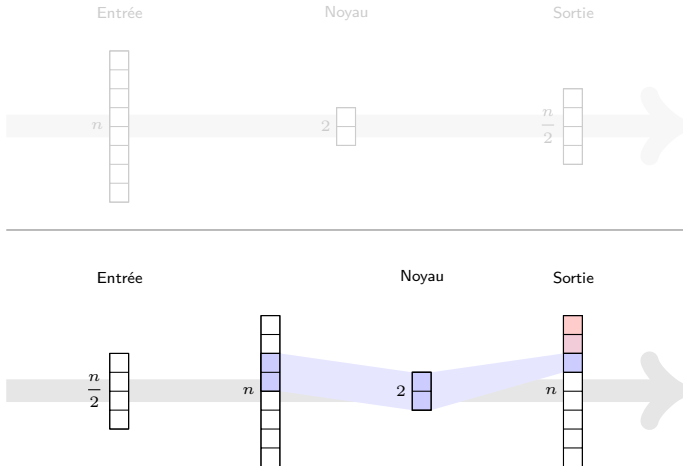
“Sous- et sur-échantillonnage”

On applique une convolution de taille 2 pour régulariser.



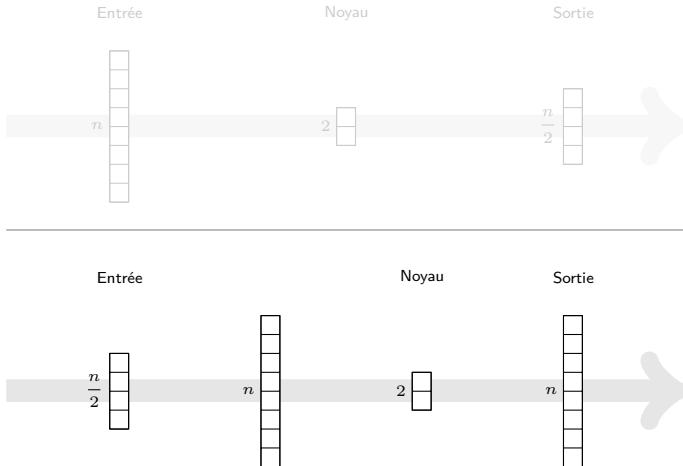
“Sous- et sur-échantillonnage”

On applique une convolution de taille 2 pour régulariser.



“Sous- et sur-échantillonnage”

On a, au final, doublé la taille.



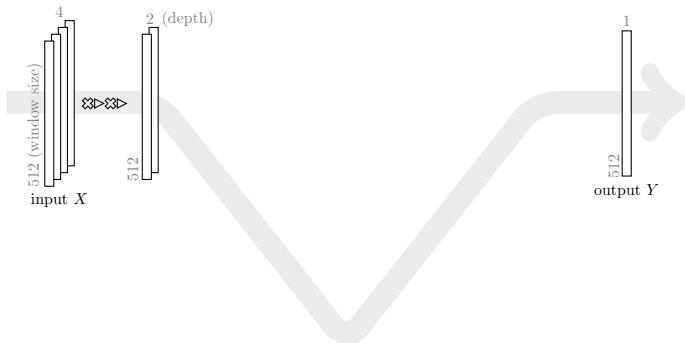
Réseau V-Net

On utilise une version 1D de l'architecture V-Net,
pour la fermeture non-locale.



Réseau V-Net

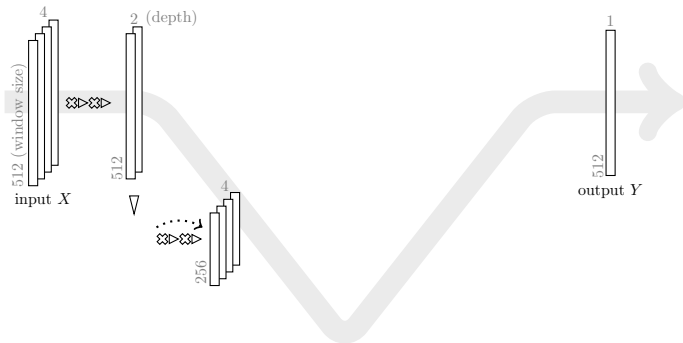
On applique deux couches de convolution
pour obtenir d (profondeur) signaux de sortie. Activation : softplus.



⊗ Convolution
▷ Softplus

Réseau V-Net

On sous-échantillonne :
on applique deux convolutions en multipliant par deux la profondeur.



⊗ Convolution

▷ Softplus



Down-sampling

⋯▷ Small shortcut

Réseau V-Net

L'opération est répétée sur ℓ niveaux.

Contexte physique
et mathématique

Modèles réduits
pour l'équation de
Vlasov

Génération des données

Réseaux de neurones

Résultats

Modèle fluide

Difficultés : stabilité et
coût calcul

Fermeture et bases
réduites

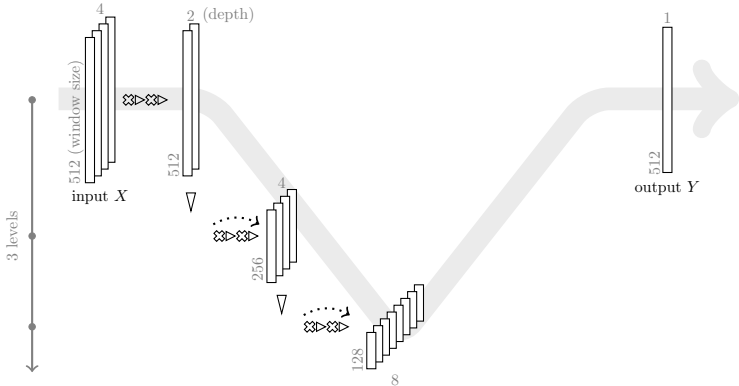
Bases réduites pour les
équations hyperboliques

Fermeture et contrôle
optimal

Stabilisation de
schéma LBM

Schéma LBM

Stabilisation et contrôle
optimal



⊗ Convolution

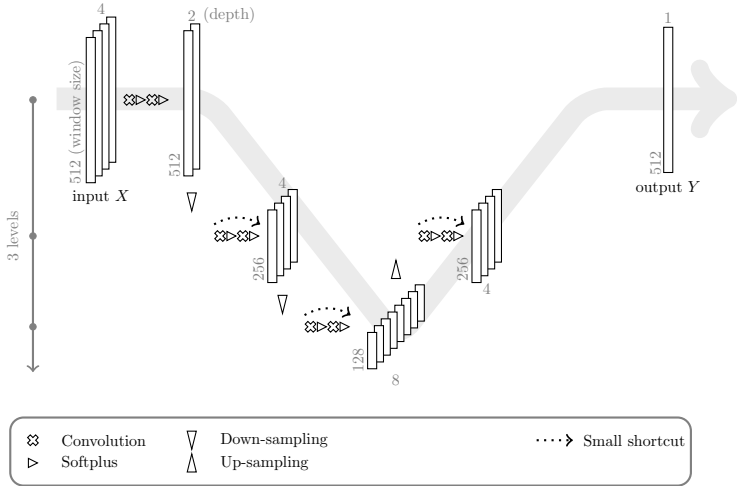
▽ Down-sampling

⋯➤ Small shortcut

▷ Softplus

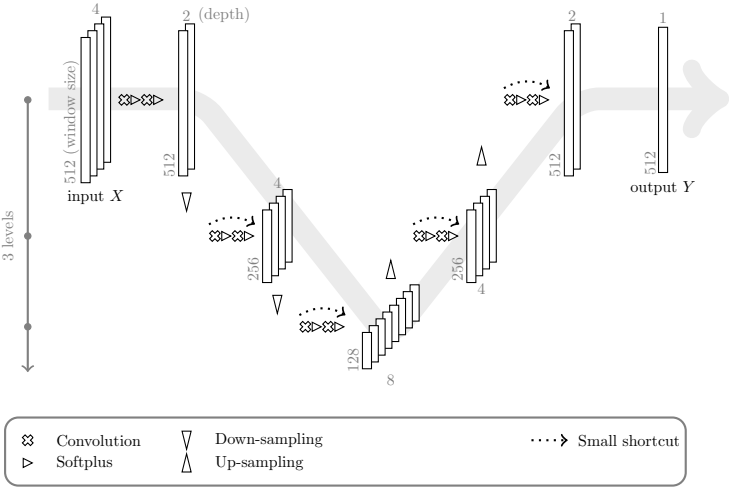
Réseau V-Net

On applique les opérations inverses pour obtenir la résolution d'origine.



Réseau V-Net

On applique les opérations inverses pour obtenir la résolution d'origine.



Contexte physique
et mathématique

Modèles réduits
pour l'équation de
Vlasov

Génération des données

Réseaux de neurones

Résultats

Modèle fluide

Difficultés : stabilité et
coût calcul

Fermeture et bases
réduites

Bases réduites pour les
équations hyperboliques

Fermeture et contrôle
optimal

Stabilisation de
schéma LBM

Schéma LBM

Stabilisation et contrôle
optimal

Réseau V-Net

À l'aide d'une moyenne pondérée pour obtenir la sortie.

Contexte physique
et mathématique

Modèles réduits
pour l'équation de
Vlasov

Génération des données

Réseaux de neurones

Résultats

Modèle fluide

Difficultés : stabilité et
coût calcul

Fermeture et bases
réduites

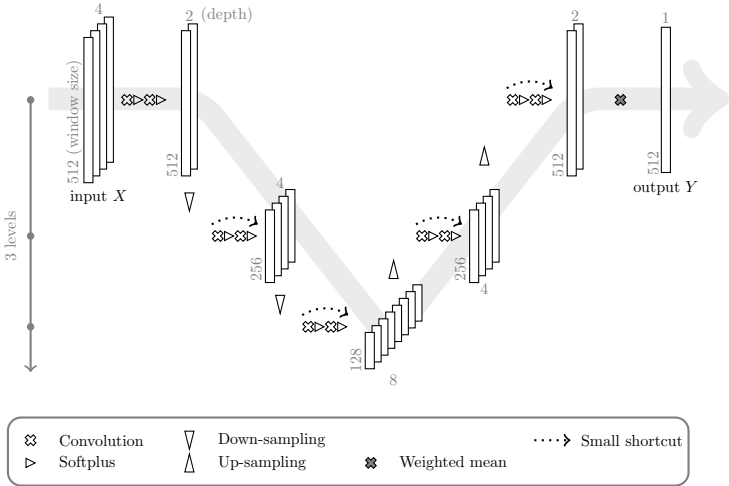
Bases réduites pour les
équations hyperboliques

Fermeture et contrôle
optimal

Stabilisation de
schéma LBM

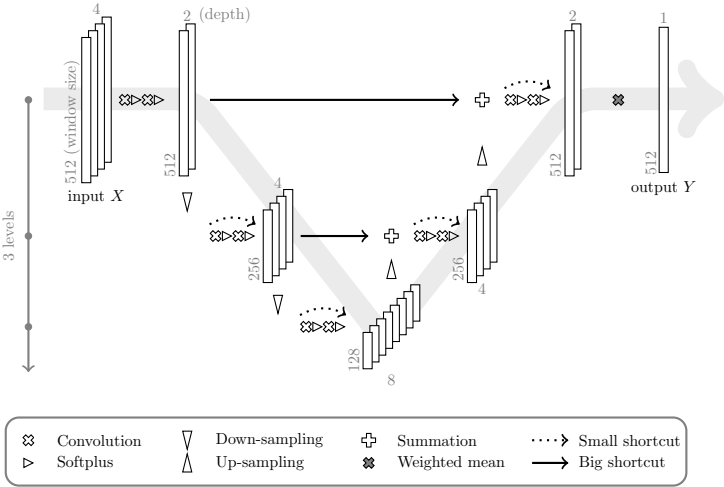
Schéma LBM

Stabilisation et contrôle
optimal



Réseau V-Net

On moyenne les signaux avant et après reconstruction



① Contexte physique et mathématique

② Modèles réduits pour l'équation de Vlasov

Génération des données

Réseaux de neurones

Résultats

Modèle fluide

Difficultés : stabilité et coût calcul

③ Fermeture et bases réduites

Bases réduites pour les équations hyperboliques

Fermeture et contrôle optimal

④ Stabilisation de schéma LBM

Schéma LBM

Stabilisation et contrôle optimal

Exemples issus du jeu de données test

Contexte physique
et mathématique

Modèles réduits
pour l'équation de
Vlasov

Génération des données
Réseaux de neurones

Résultats

Modèle fluide

Difficultés : stabilité et
coût calcul

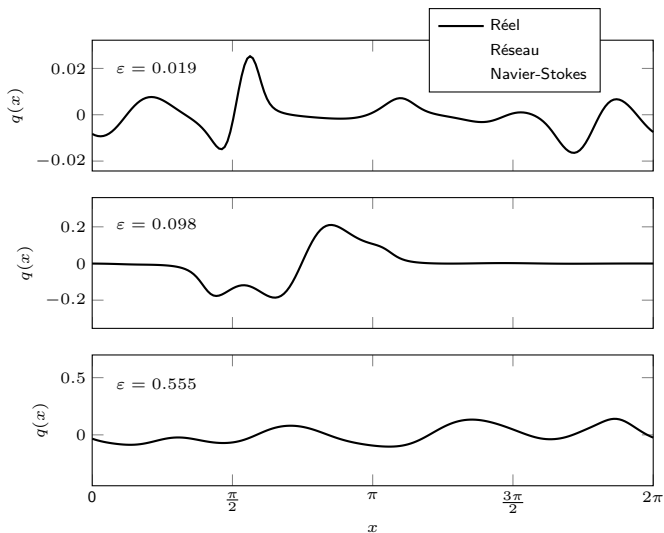
Fermeture et bases
réduites

Bases réduites pour les
équations hyperboliques

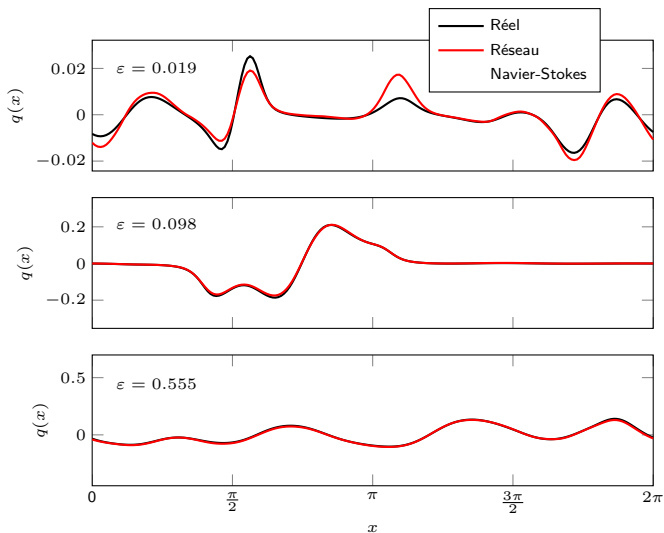
Fermeture et contrôle
optimal

Stabilisation de
schéma LBM

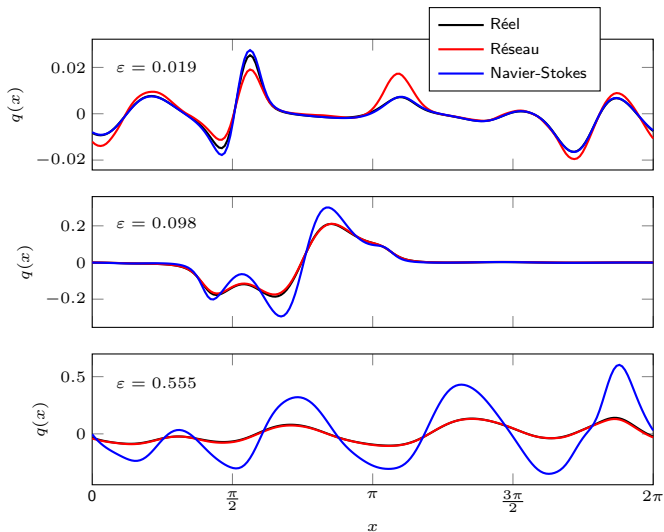
Schéma LBM
Stabilisation et contrôle
optimal



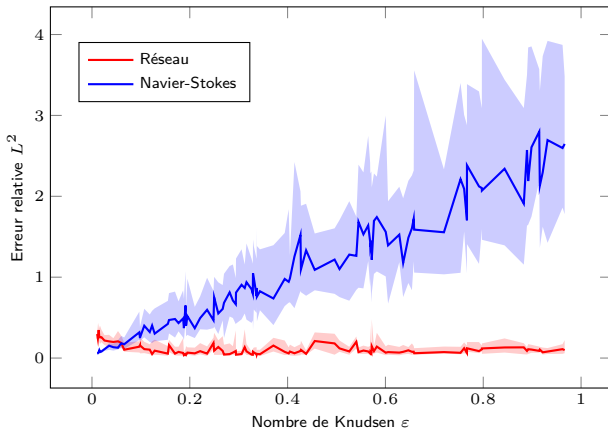
Exemples issus du jeu de données test



Exemples issus du jeu de données test



Influence de ε



Comme attendu, l'erreur explose pour Navier-Stokes quand ε augmente. L'erreur semble quasiment uniforme pour le réseau (qui est moins bon que Navier-Stokes pour les petites valeurs de ε).

① Contexte physique et mathématique

② Modèles réduits pour l'équation de Vlasov

Génération des données

Réseaux de neurones

Résultats

Modèle fluide

Difficultés : stabilité et coût calcul

③ Fermeture et bases réduites

Bases réduites pour les équations hyperboliques

Fermeture et contrôle optimal

④ Stabilisation de schéma LBM

Schéma LBM

Stabilisation et contrôle optimal

Modèle fluide

- On compare l'énergie électrique $\mathcal{E}(t) = \int E(x, t)^2 dx$ dans les modèles suivants :

Cinétique

- Dans la suite : Il y a des différences entre "le modèle cinétique" et "le modèle fluide+cinétique". Cela semble provenir des méthodes numériques.
- On peut pas faire mieux que : "fluide + cinétique".

Modèle fluide

- On compare l'énergie électrique $\mathcal{E}(t) = \int E(x, t)^2 dx$ dans les modèles suivants :

————

Cinétique

- - - -

Fluide+Cinétique : $\hat{q} = q$

- Dans la suite : Il y a des différences entre "le modèle cinétique" et "le modèle fluide+cinétique". Cela semble provenir des méthodes numériques.
- On peut pas faire mieux que : "fluide + cinétique".

Modèle fluide

- On compare l'énergie électrique $\mathcal{E}(t) = \int E(x, t)^2 dx$ dans les modèles suivants :

————

Cinétique

- - - -

Fluide+Cinétique : $\hat{q} = q$

- - - -

Fluide+Réseau : $\hat{q} = C_\theta(\varepsilon, \rho, u, T)$

- Dans la suite : Il y a des différences entre "le modèle cinétique" et "le modèle fluide+cinétique". Cela semble provenir des méthodes numériques.
- On peut pas faire mieux que : "fluide + cinétique".

Modèle fluide

- On compare l'énergie électrique $\mathcal{E}(t) = \int E(x, t)^2 dx$ dans les modèles suivants :

————

Cinétique

- - - -

Fluide+Cinétique : $\hat{q} = q$

- - - -

Fluide+Réseau : $\hat{q} = C_\theta(\varepsilon, \rho, u, T)$

- - - -

Navier-Stokes : $\hat{q} = -\frac{3}{2}\varepsilon\rho T\partial_x T$

Modèle fluide

- On compare l'énergie électrique $\mathcal{E}(t) = \int E(x, t)^2 dx$ dans les modèles suivants :

————

Cinétique

- - - - -

Fluide+Cinétique : $\hat{q} = q$

- - - - -

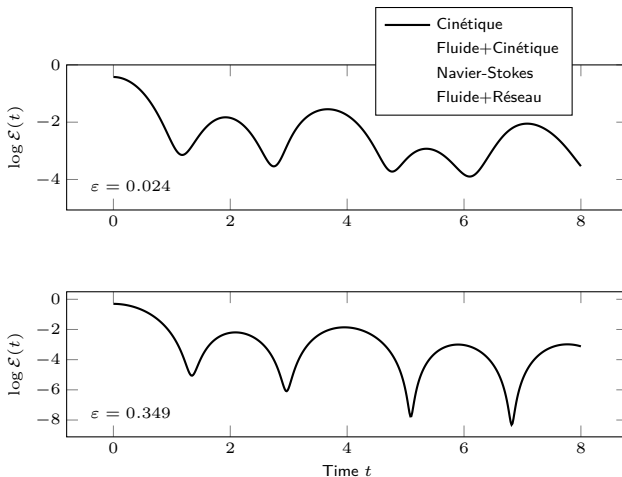
Fluide+Réseau : $\hat{q} = C_\theta(\varepsilon, \rho, u, T)$

- - - - -

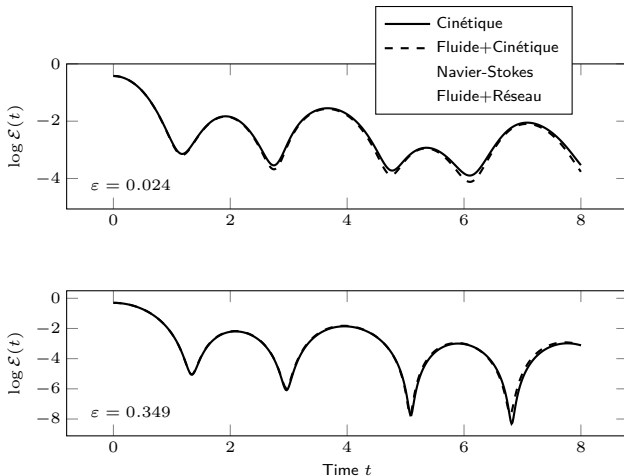
Navier-Stokes : $\hat{q} = -\frac{3}{2}\varepsilon\rho T\partial_x T$

- Dans la suite : Il y a des différences entre "le modèle cinétique" et "le modèle fluide+cinétique". Cela semble provenir des méthodes numériques.
- On peut pas faire mieux que : "fluide + cinétique".

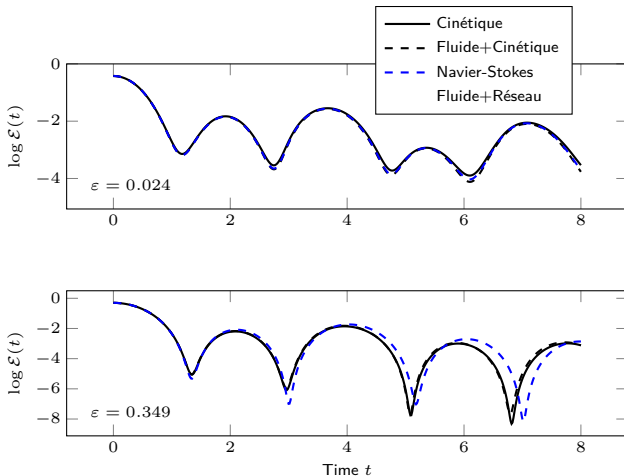
Exemples (1/2)



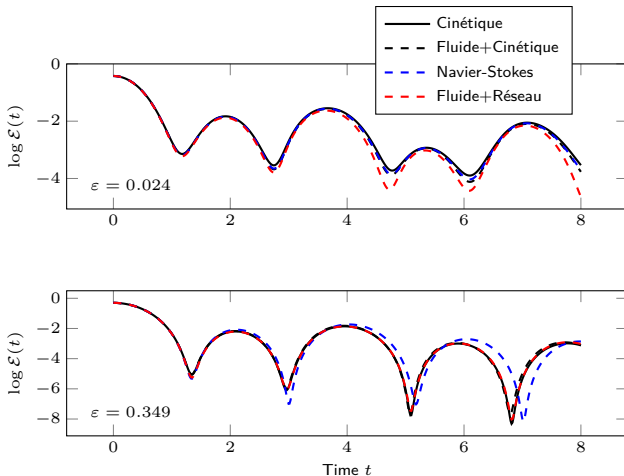
Exemples (1/2)



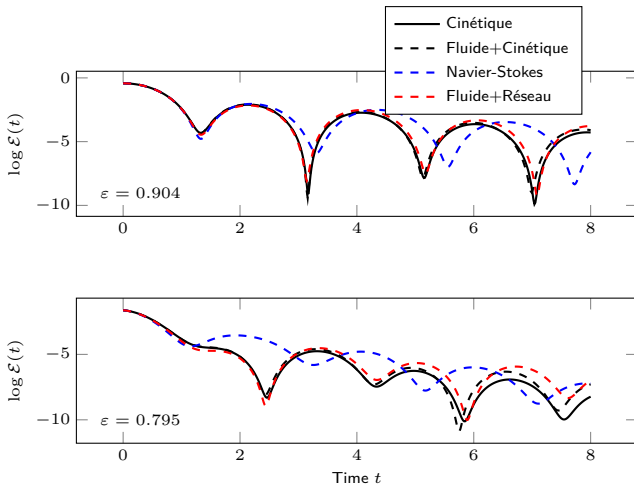
Exemples (1/2)



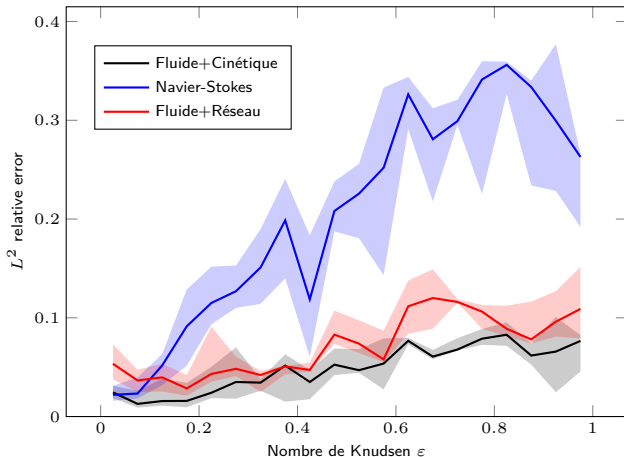
Exemples (1/2)



Exemples (2/2)



Influence de ε



*L'erreur du modèle "Fluide+Réseau" augmente de façon similaire à celle du modèle "Fluide+Cinétique".
Dérive de l'erreur expliquée par des problèmes numériques ?*

① Contexte physique et mathématique

② Modèles réduits pour l'équation de Vlasov

Génération des données

Réseaux de neurones

Résultats

Modèle fluide

Difficultés : stabilité et coût calcul

③ Fermeture et bases réduites

Bases réduites pour les équations hyperboliques

Fermeture et contrôle optimal

④ Stabilisation de schéma LBM

Schéma LBM

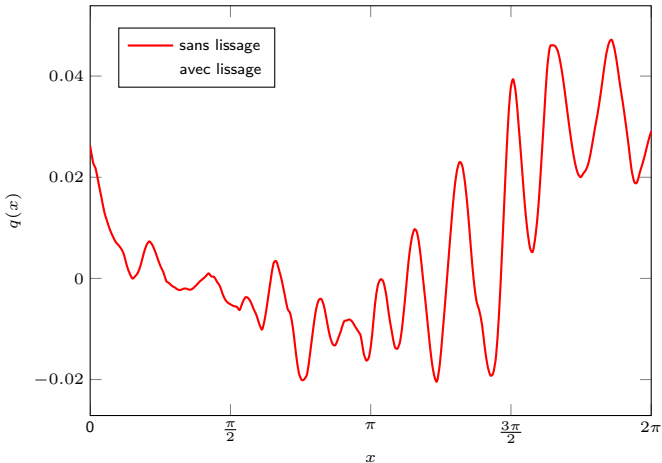
Stabilisation et contrôle optimal

Hyper-paramètres	Valeur
rule taille de l'entrée	512
nombre de niveau (ℓ)	5
profondeur initiale (d)	4
taille des noyaux (p)	11
fonction d'activation	softplus

- Hyper-paramètres du réseau :
- Nombre total de paramètres : 161 937
- Préparation des données (standardisation, découpage, normalisation) en annexe.
- **Schéma** :
 - On calcule Q_θ^n avec ρ^n , u^n et T^n
 - On calcule l'approximation de $\partial_x Q_\theta^n$
 - On calcule le modèle fluide avec un schéma explicite $\partial_x Q_\theta^n$ traité comme un terme source.
- On peut pas assurer que Q_θ^n et $\partial_x Q_\theta^n$ ne génèrent pas des oscillations numériques.
- Les oscillations peuvent détruire **la stabilité du schéma complet**. C'est le cas en pratique.
- **1ère approche (simple) : Régularisation Gaussienne.**

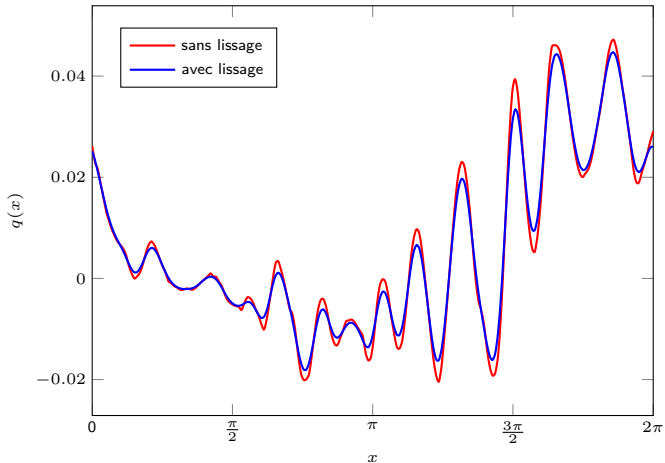
Exemple

Exemple de flux de chaleur particulièrement oscillant.



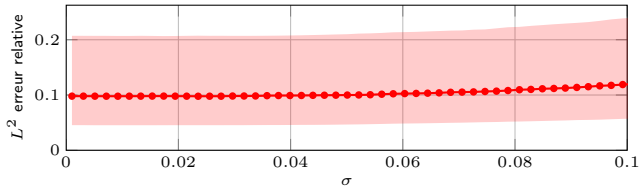
Exemple

On régularise la prédiction avec une convolution
et un noyau Gaussien de paramètre σ .



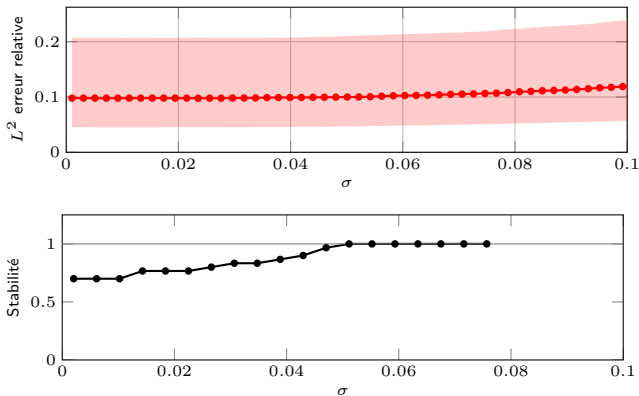
Analyse de stabilité

Première figure : erreur de prédiction en fonction du lissage.



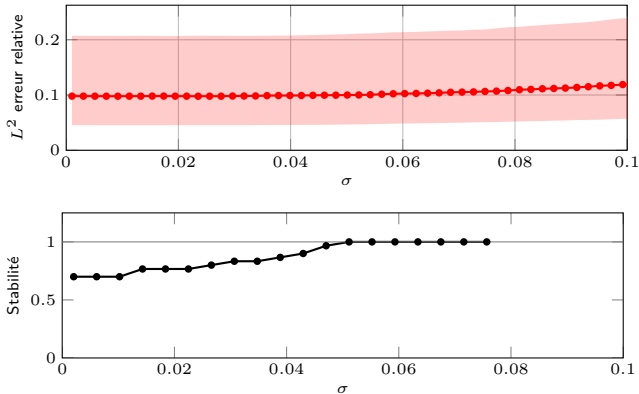
Analyse de stabilité

Cette deuxième figure montre
la proportion croissante de simulations réussies
à mesure que le lissage augmente.



Analyse de stabilité

Nous avons choisi $\sigma \simeq 0.06$,
car cela semble être suffisant pour assurer la stabilité
sans pour autant diminuer la précision.



Conclusion

- **Conclusion** : Le processus d'apprentissage et le CNN permettent d'obtenir **une nouvelle fermeture avec une erreur presque uniforme en ε .**
- La fermeture obtenue est plus précise que celle de Navier-Stokes pour des $\varepsilon \in [0, 1]$.
- **Défaut** : on ne peut pas garantir la stabilité.
- Travail actuel : Extension $2D \times 2D$ au modèle BGK. Problème de stabilité.
- **Futur travail** : pour assurer la stabilité, on souhaite **apprendre la base aux moments et utiliser la méthode de fermeture basée sur la minimisation d'entropie, avec apprentissage de l'entropie.**

- 1 Contexte physique et mathématique
- 2 Modèles réduits pour l'équation de Vlasov
 - Génération des données
 - Réseaux de neurones
 - Résultats
 - Modèle fluide
 - Difficultés : stabilité et coût calcul
- 3 Fermeture et bases réduites
 - Bases réduites pour les équations hyperboliques
 - Fermeture et contrôle optimal
- 4 Stabilisation de schéma LBM
 - Schéma LBM
 - Stabilisation et contrôle optimal

- 1 Contexte physique et mathématique
- 2 Modèles réduits pour l'équation de Vlasov
 - Génération des données
 - Réseaux de neurones
 - Résultats
 - Modèle fluide
 - Difficultés : stabilité et coût calcul
- 3 Fermeture et bases réduites
 - Bases réduites pour les équations hyperboliques
 - Fermeture et contrôle optimal
- 4 Stabilisation de schéma LBM
 - Schéma LBM
 - Stabilisation et contrôle optimal

EDP hyperbolique et bases réduites

- **Objectif** : introduire une **autre méthode de fermeture**. On introduit pour cela un autre problème.
- On considère les équations du type :

$$\partial_t \rho + \partial_x F(\rho) = \frac{1}{R_e} \partial_{xx} \rho$$

- Après discrétisation, on obtient une ODE de taille n_x : $\frac{d}{dt} \rho(t) = F(\rho(t); R_e)$
- Méthode des bases réduites : on cherche une base du type

$$\rho \approx \sum_{k=1}^N \alpha_k(t) \phi_k(x)$$

avec $N \ll n_x$, pour représenter les solutions de cette équation dans une zone de paramètres.

- **Méthode POD (Proper Orthogonal Decomposition)** :
 - on fait des simulations avec différents paramètres pour obtenir des snapshots ρ_i ,
 - on construit la matrice de covariance

$$C = \frac{1}{m} \sum_{i=1}^m \rho_i \otimes \rho_i,$$

- on obtient une base Φ à N colonnes en prenant les N vecteurs propres associés aux plus grandes valeurs propres.
- On teste avec une méthode de Galerkin :

$$\frac{d}{dt} \mathbf{z}(t) = \Phi^t F(\Phi \mathbf{z}(t))$$

avec $\mathbf{z}(t) = \Phi \rho(t)$.

EDP hyperbolique et bases réduites

- La résolution de l'ODE

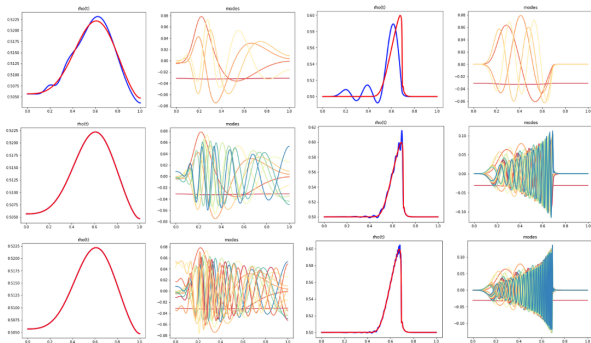
$$\frac{d}{dt}\mathbf{z}(t) = \Phi^t F(\Phi\mathbf{z}(t); R_e)$$

avec $\mathbf{z}(t) = \Phi\rho(t)$ nécessite de repasser en grande dimension.

- On veut construire une **fermeture** du type :

$$\frac{d}{dt}\mathbf{z}(t) = F_r(\mathbf{z}(t); R_e)$$

- Méthode classique : DEIM.** Problème : la POD, et surtout DEIM, fonctionne mal en régime hyperbolique.
- Exemple : équation de Burgers $\rightarrow F(\rho) = \frac{1}{2}\rho^2$.



- À gauche : $R_e = 40$ (5, 10, 20 modes); à droite : $R_e = 40000$ (5, 20, 40 modes)
- Conclusion** : la POD crée une base spectrale associée aux données. On retrouve l'effet de Gibbs associé.

- 1 Contexte physique et mathématique
- 2 Modèles réduits pour l'équation de Vlasov
 - Génération des données
 - Réseaux de neurones
 - Résultats
 - Modèle fluide
 - Difficultés : stabilité et coût calcul
- 3 Fermeture et bases réduites
 - Bases réduites pour les équations hyperboliques
 - Fermeture et contrôle optimal
- 4 Stabilisation de schéma LBM
 - Schéma LBM
 - Stabilisation et contrôle optimal

Apprentissage de la fermeture

- On souhaite apprendre la fermeture.

- Par régression de modèle :

- Stratégie 1 (hyper-réduction) :

$$\min_{\theta} \left(\sum_{i=1}^m \sum_{t_n=1}^{n_t} \|\Phi^t F(\Phi \mathbf{z}_i(t_n)) - F_{\theta}(\mathbf{z}_i(t_n))\|_2^2 \right)$$

avec i le numéro de la condition initiale et $\mathbf{z}_i(t_n) = \Phi \rho_i(t_n)$.

- Stratégie 2 (fermeture) :

$$\min_{\theta} \left(\sum_{i=1}^m \sum_{t_n=1}^{n_t} \|\Phi^t F(\rho_i(t_n)) - F_{\theta}(\mathbf{z}_i(t_n))\|_2^2 \right)$$

avec i le numéro de la condition initiale.

- Par régression de trajectoire :

L'idée est de tenir compte de portions entières de trajectoire pour apprendre. On résout donc

$$\min_{\theta} \left(\sum_{i=1}^m \int_0^T \|\Phi^t \rho_i(t) - \mathbf{z}_i(t)\|_2^2 \right)$$

sous contrainte que $\mathbf{z}_i(t)$ soit solution de

$$\frac{d}{dt} \mathbf{z}(t) = F_{\theta}(\Phi \mathbf{z}(t)), \quad \mathbf{z}_i(t) = \mathbf{z}_i^0$$

On parle alors de contrôle optimal.

Apprentissage par régression de trajectoire

- On discrétise

$$\min_{\theta} J(\theta) = \min_{\theta} \left(\sum_{i=1}^m \sum_{t_n=1}^{n_t} \|\rho_i(t_n) - \mathbf{z}_i(t_n)\|_2^2 \right)$$

avec

$$\mathbf{z}_i(t_n) = S_{\Delta t}^l(\mathbf{z}_i(t_{n-1}))$$

avec $S_{\Delta t}$ votre schéma numérique en temps favori.

- Comment apprendre ? **Descente de gradient** :

$$\theta = \theta - \gamma \nabla_{\theta} J(\theta)$$

- Difficulté** : le gradient dépend de nombreuses compositions du réseau et de $S_{\Delta t}$.
- Solution** : les frameworks d'apprentissage profond sont capables de différencier au travers de **nombreuses compositions de fonctions**.
- On fait l'approximation suivante :

$$\sum_{t_n=1}^{n_t} \|\rho_i(t_n) - \mathbf{z}_i(t_n)\|_2^2 \approx \sum_{k=0}^K \sum_{t_n=k}^{(k+1)\Delta t} \|\rho_i(t_n) - \mathbf{z}_i(t_n)\|_2^2$$

avec ΔT tel que $K\Delta t = n_t$ et avec $\mathbf{z}_i^k(0) = \Phi \rho_i^k(k\Delta t)$.

- Cette approximation revient à supposer que le coût sur l'ensemble de la trajectoire est très proche de la somme des coûts sur des sous-trajectoires.

Contexte physique
et mathématique

Modèles réduits
pour l'équation de
Vlasov

Génération des données

Réseaux de neurones

Résultats

Modèle fluide

Difficultés : stabilité et
coût calcul

Fermeture et bases
réduites

Bases réduites pour les
équations hyperboliques

Fermeture et contrôle
optimal

Stabilisation de
schéma LBM

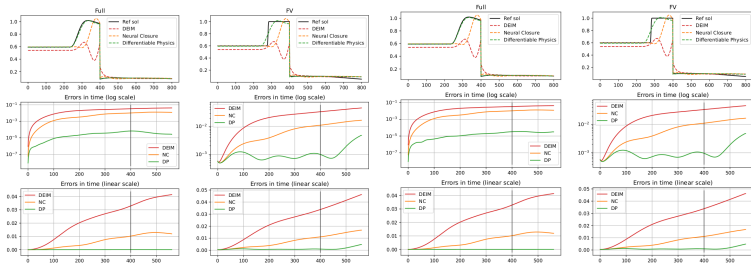
Schéma LBM

Stabilisation et contrôle
optimal

- On résout le problème suivant de trafic routier

$$\begin{cases} \partial_t \rho_1 + \partial_x \left(\rho_1 \left(1 - \frac{\rho_1}{\rho_{1,max}} \right) \right) = 0, & x \in [0, 1] \\ \partial_t \rho_2 + \partial_x \left(\rho_2 \left(1 - \frac{\rho_2}{\rho_{2,max}} \right) \right) = 0, & x \in [1, 2] \\ \rho_1(t, 0) = \rho_l, & \rho_1(t, 1) = \rho_2(t, 1) = g(\rho_1(t, 1), \rho_2(t, 0)), & \rho_2(t, 2) = \rho_r \end{cases}$$

- On couple donc la propagation du trafic sur deux routes avec des densités maximales $\rho_{i,max}$.



- 5 modes.** À gauche : on apprend sur des tranches de 50 pas de temps. À droite : on apprend sur des tranches de 200 pas de temps.
- On observe un léger gain de précision avec 200 pas de temps dans l'apprentissage.

Contexte physique
et mathématique

Modèles réduits
pour l'équation de
Vlasov

Génération des données

Réseaux de neurones

Résultats

Modèle fluide

Difficultés : stabilité et
coût calcul

Fermeture et bases
réduites

Bases réduites pour les
équations hyperboliques

Fermeture et contrôle
optimal

Stabilisation de
schéma LBM

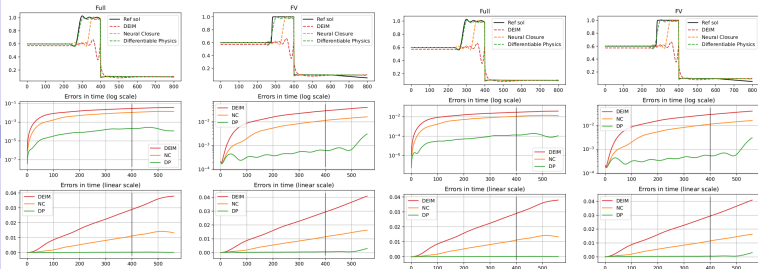
Schéma LBM

Stabilisation et contrôle
optimal

- On résout le problème suivant de trafic routier

$$\begin{cases} \partial_t \rho_1 + \partial_x \left(\rho_1 \left(1 - \frac{\rho_1}{\rho_{1,max}} \right) \right) = 0, & x \in [0, 1] \\ \partial_t \rho_2 + \partial_x \left(\rho_2 \left(1 - \frac{\rho_2}{\rho_{2,max}} \right) \right) = 0, & x \in [1, 2] \\ \rho_1(t, 0) = \rho_l, & \rho_1(t, 1) = \rho_2(t, 1) = g(\rho_1(t, 1), \rho_2(t, 0)), & \rho_2(t, 2) = \rho_r \end{cases}$$

- On couple donc la propagation du trafic sur deux routes avec des densités maximales $\rho_{i,max}$.



- 10 modes.** À gauche : on apprend sur des tranches de 50 pas de temps. À droite : on apprend sur des tranches de 200 pas de temps.
- On observe un léger gain de précision avec 200 pas de temps dans l'apprentissage.

Contexte physique
et mathématique

Modèles réduits
pour l'équation de
Vlasov

Génération des données

Réseaux de neurones

Résultats

Modèle fluide

Difficultés : stabilité et
coût calcul

Fermeture et bases
réduites

Bases réduites pour les
équations hyperboliques

Fermeture et contrôle
optimal

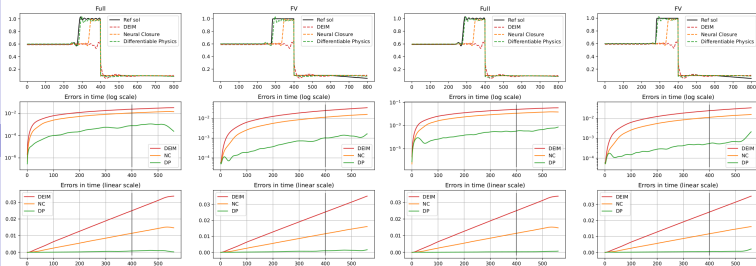
Stabilisation de
schéma LBM

Schéma LBM
Stabilisation et contrôle
optimal

- On résout le problème suivant de trafic routier

$$\begin{cases} \partial_t \rho_1 + \partial_x \left(\rho_1 \left(1 - \frac{\rho_1}{\rho_{1,max}} \right) \right) = 0, & x \in [0, 1] \\ \partial_t \rho_2 + \partial_x \left(\rho_2 \left(1 - \frac{\rho_2}{\rho_{2,max}} \right) \right) = 0, & x \in [1, 2] \\ \rho_1(t, 0) = \rho_l, & \rho_1(t, 1) = \rho_2(t, 1) = g(\rho_1(t, 1), \rho_2(t, 0)), & \rho_2(t, 2) = \rho_r \end{cases}$$

- On couple donc la propagation du trafic sur deux routes avec des densités maximales $\rho_{i,max}$.



- 20 modes.** À gauche : on apprend sur des tranches de 50 pas de temps. À droite : on apprend sur des tranches de 200 pas de temps.
- On observe un léger gain de précision avec 200 pas de temps dans l'apprentissage.

Conclusion

- **Conclusion** : apprendre la fermeture sur plusieurs pas de temps permet d'éviter les accumulations dramatiques d'erreur, et même de corriger des défauts dans la base.
- Dans ce cas, on peut même apprendre sur des centaines de pas de temps.
- **Défaut** : coût en mémoire. Pour des gros problèmes, on doit n'apprendre que sur **peu de pas de temps**.
 - Apprentissage par renforcement,
 - Neural ODE (gradient sur les pas de temps calculé par contrôle optimal).

- 1 Contexte physique et mathématique
- 2 Modèles réduits pour l'équation de Vlasov
 - Génération des données
 - Réseaux de neurones
 - Résultats
 - Modèle fluide
 - Difficultés : stabilité et coût calcul
- 3 Fermeture et bases réduites
 - Bases réduites pour les équations hyperboliques
 - Fermeture et contrôle optimal
- 4 Stabilisation de schéma LBM
 - Schéma LBM
 - Stabilisation et contrôle optimal

- 1 Contexte physique et mathématique
- 2 Modèles réduits pour l'équation de Vlasov
 - Génération des données
 - Réseaux de neurones
 - Résultats
 - Modèle fluide
 - Difficultés : stabilité et coût calcul
- 3 Fermeture et bases réduites
 - Bases réduites pour les équations hyperboliques
 - Fermeture et contrôle optimal
- 4 Stabilisation de schéma LBM
 - Schéma LBM
 - Stabilisation et contrôle optimal

Schéma LBM

- On considère des **équations hyperboliques**.
- On introduit le cas scalaire, mais tout ce qui suit fonctionne de la même façon pour le cas vectoriel.

$$\partial \rho + \nabla \cdot (f(\rho)) = 0.$$

- Schéma LBM [D2Q4] :

- On introduit 4 variables $f_1(t, x), \dots, f_4(t, x)$
- On introduit 4 vitesses :

$$\left[v_1 = \begin{pmatrix} \lambda \\ 0 \end{pmatrix}, v_2 = \begin{pmatrix} -\lambda \\ 0 \end{pmatrix}, v_3 = \begin{pmatrix} 0 \\ -\lambda \end{pmatrix}, v_4 = \begin{pmatrix} 0 \\ \lambda \end{pmatrix} \right]$$

- On résout à chaque pas de temps :

$$f_i^*(t_n, \mathbf{x}_j) = f_i(t_n, \mathbf{x}_j - \Delta t v_i), \quad \forall 1 \leq i \leq 4$$

- On applique une **étape de collision**

$$f_i(t_{n+1}, \mathbf{x}_j) = f_i^*(\mathbf{x}_j) + W(\rho(\mathbf{x}_j))(f_i^{eq}(\rho(\mathbf{x}_j)) - f_i^*(\mathbf{x}_j))$$

avec $\rho(\mathbf{x}_j) = \sum_{i=1}^4 f_i^*(\mathbf{x}_j)$, M une matrice de passage et

$$W(\rho) = M^{-1} \left(\begin{array}{c|cc|c} 1 & 0 & 0 & 0 \\ \hline 0 & \Omega(\rho) & & 0 \\ 0 & & & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right) M.$$

Schéma LBM

- Erreur des schémas LBM :

$$\partial \rho + \nabla \cdot (f(\rho)) = \Delta t \nabla \cdot \left(\underbrace{\left(\Omega^{-1}(\rho) - \frac{1}{2} I \right) \left(\frac{\lambda^2}{2} I - f'(\rho) \otimes f'(\rho) \right)}_{D(\rho)} \nabla \rho \right) + O(\Delta t^2)$$

- $\Omega = I_d$ donne le schéma d'ordre 1 classique. $\Omega = 2I_d$ donne le schéma d'ordre 2.
- Stabilité approchée : $D(\rho) \geq 0$
- Le schéma d'ordre 2 **oscille notamment autour des discontinuités, ce qui génère des problèmes de stabilité.**
- Exemple ci-dessous avec $f(\rho) = \begin{pmatrix} a \\ b \end{pmatrix} \rho$:

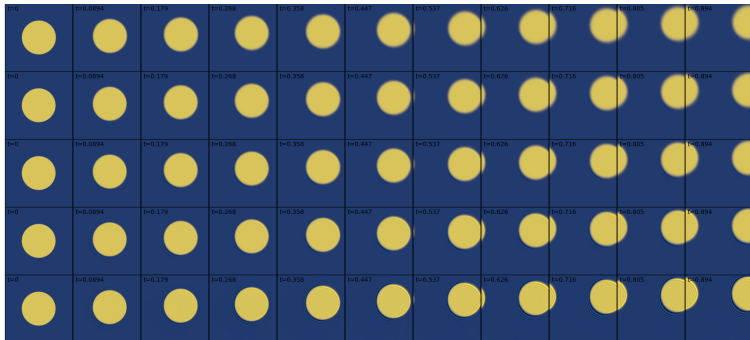
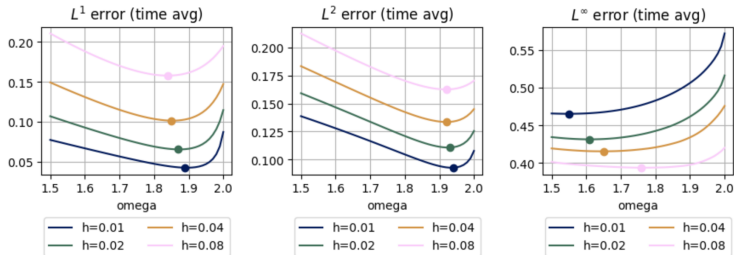


Schéma LBM

- Le choix de Ω n'est pas simple.



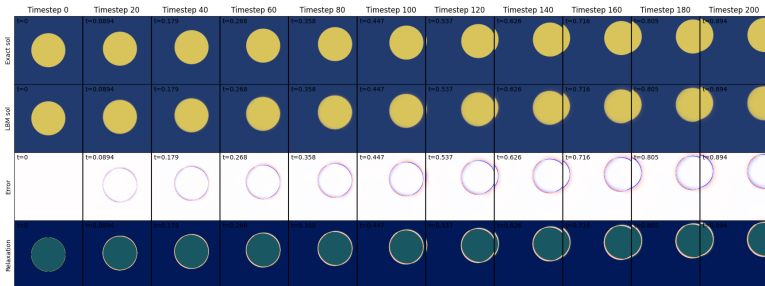
- Erreur en fonction de $\Omega = \omega I_d$.
- Il y a compétition :
 - entre l'erreur de diffusion et l'erreur de dispersion $O(\Delta t^2 \partial_{xxx} \dots)$,
 - entre l'erreur de consistance et la stabilité.
- But** : apprendre le meilleur Ω en apprenant sur des simulations en temps.
- Méthode** : même algorithme d'apprentissage que précédemment (apprentissage sur plusieurs pas de temps).
- Remarque** : c'est un travail en cours, et les résultats sont très préliminaires.
- Plusieurs réseaux :

$$\Omega(\rho(x_i)) = f_\theta(\rho(x_i))I_d, \text{ ou } f_\theta(\rho(x_{i-1}), \rho(x_i), \rho(x_{i+1}))I_d, \text{ ou } = A_\theta(\rho(x_i))$$

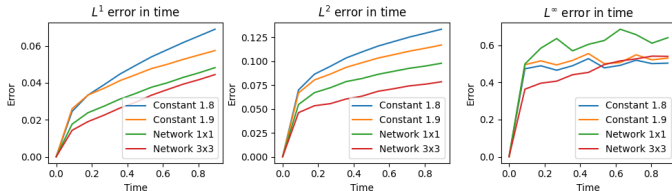
- 1 Contexte physique et mathématique
- 2 Modèles réduits pour l'équation de Vlasov
 - Génération des données
 - Réseaux de neurones
 - Résultats
 - Modèle fluide
 - Difficultés : stabilité et coût calcul
- 3 Fermeture et bases réduites
 - Bases réduites pour les équations hyperboliques
 - Fermeture et contrôle optimal
- 4 Stabilisation de schéma LBM
 - Schéma LBM
 - Stabilisation et contrôle optimal

Équation de transport I

- Équation de transport. On apprend sur 1 ou 3 conditions initiales constantes par morceaux.
- Stencil en espace du réseau 1×1 et 3×3 .

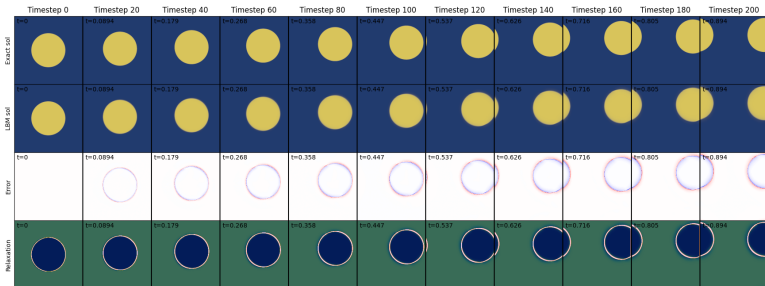


- Réseau 1×1
- Erreurs

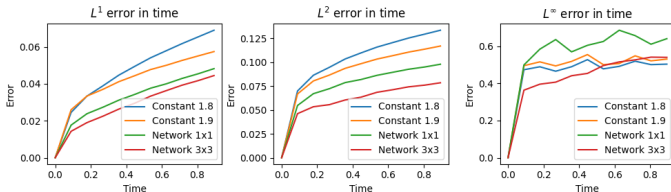


Équation de transport I

- Équation de transport. On apprend sur 1 ou 3 conditions initiales constantes par morceaux.
- Stencil en espace du réseau 1×1 et 3×3 .

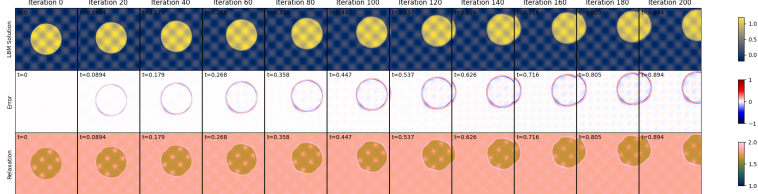


- Réseau 3×3
- Erreurs

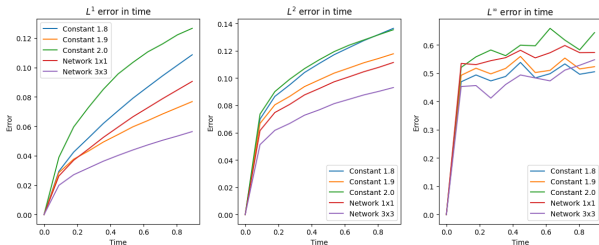


Équation de transport II

- Équation de transport. On apprend sur 1 ou 3 conditions initiales sinusoïdales par morceaux.
- Solution

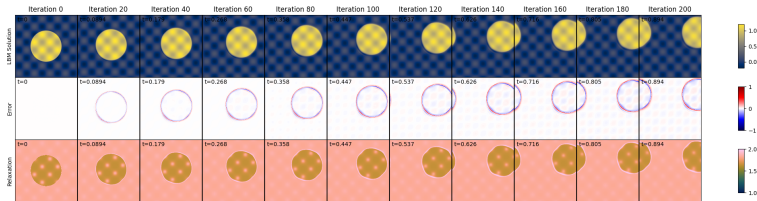


- Erreurs

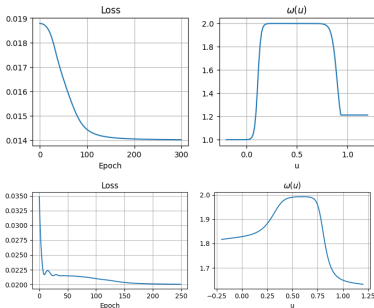


Équation de transport II

- Équation de transport. On apprend sur 1 ou 3 conditions initiales sinusoïdales par morceaux.
- Solution

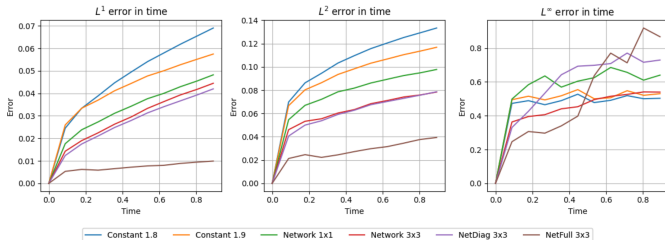


- Coefficients $\Omega(\rho)$. Gauche : cas constant. Droit : cas variable.



Équation de transport III

- Le réseaux est une matrice $\Omega_{\theta}(\rho)$
- Permet de proposer des viscosity anisotrope en espace.
- Erreurs



- Pas d'apprentissage en norme $L^{inf ty}$ ni de pénalisation des oscillations.
- On peut l'ajouter. Par exemple la semi norme $W^{2,1}$ marche bien.

Équations d'Euler

- On s'intéresse aux équations d'Euler

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p I_d) = 0, \\ \partial_t E + \nabla \cdot (E \mathbf{u} + p \mathbf{u}) = 0. \end{cases}$$

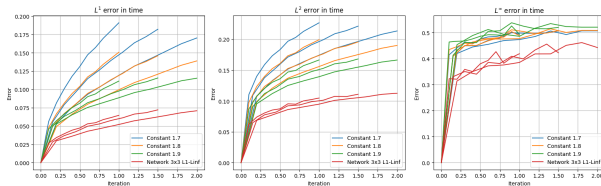
- Si $p = \text{cst}$ et $\mathbf{u} = \text{cst}$, alors on a

$$\partial_t \rho + \mathbf{u} \cdot \nabla \rho = 0$$

- Pour la stabilité $\lambda > |\mathbf{u}| + \sqrt{\frac{\gamma p}{\rho}}$.
- Erreur de diffusion standard sur la densité :

$$E = \Delta t \nabla \cdot (\lambda^2 - |\mathbf{u}|^2) \nabla \rho + O(\Delta t)$$

- L'erreur augmente avec p sans que le problème de transport de densité change.



- Trois pressions : $p = 0$, $p = 0.012$, $p = 0.5$.

Conclusion

- **Conclusion** : cette technique fonctionne bien sur des cas limités.
- **Prochaine étape** : apprendre sur plusieurs données initiales.
- Le plus intéressant est d'utiliser des matrices Ω non diagonales pour les systèmes, afin d'obtenir de meilleures viscosités.
- Des problèmes de stabilité compromettent rapidement l'entraînement. Questions :
 - comment détecter les possibles explosions et apprendre ?
 - comment caractériser/explore les matrices Ω qui assurent la stabilité ? (la dessus on galère).

Méthodes
d'apprentissage
pour les problèmes
de fermeture et
stabilisation en EDP

E. Franck

Contexte physique
et mathématique

Modèles réduits
pour l'équation de
Vlasov

Génération des données

Réseaux de neurones

Résultats

Modèle fluide

Difficultés : stabilité et
coût calcul

Fermeture et bases
réduites

Bases réduites pour les
équations hyperboliques

Fermeture et contrôle
optimal

Stabilisation de
schéma LBM

Schéma LBM

Stabilisation et contrôle
optimal

Merci de votre attention

Merci de votre attention !