

Présentation du sujet de simulation et contrôle d'écoulements instables par intelligence artificielle

julien VALENTIN

Juin 2022

Introduction

- ▶ CNAM
- ▶ ONERA
- ▶ programme Artificial Intelligence for Health, Physics, Transportation and Defense
 - ▶ 4 axes de recherche
 - ▶ 10 doctorants
 - ▶ durée de 3 ans
- ▶ cofinancé par l'ANR

Ma formation initiale (1/2)

- ▶ lycée et C.P.G.E en sciences de l'ingénieur
- ▶ un an à CentraleSupélec, campus de Metz
- ▶ bifurcation licence de maths/physique à Sorbonne-Université
- ▶ projet de fin de licence sur l'algorithme DeepTexture de Léon Gatys

Ma formation initiale (2/2)

- ▶ ouverture à l'Observatoire de Paris (12 ECTS en Master 1)
- ▶ master de mathématique spécialité mathématiques de la modélisation, Sorbonne-Université.
- ▶ stage de fin d'étude à l'Institut de Mathématique de Marseille : analyse numérique par volumes finis sur maillage M.A.C structuré des équations de Stokes, Darcy et Brinkman auprès de Philippe Angot

Depuis l'obtention du diplôme

Ingénieur R&D

- ▶ TENSYL SARL, PME à Périgny (17)
- ▶ Modélisation et simulation de procédés de fabrications de matériaux composites
- ▶ Problèmes multiphysiques (écoulements, poreux, structure, thermique)
- ▶ Problèmes multi-échelle : matériau, mèche, fibre
- ▶ FreeFEM++

Enseignant vacataire bac +3

- ▶ EPITA, Paris
- ▶ 12×2 heures
- ▶ Travaux Pratiques d'algèbre numérique
- ▶ Python : Pandas, Numpy, Scipy
- ▶ PCA, Resolution de systemes lineaires, conditionnement et pré-conditionnement, SVD...

Présentation générale du sujet de thèse

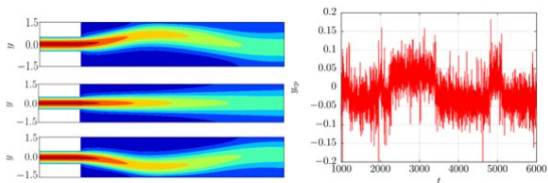
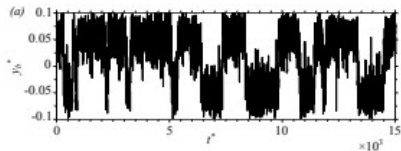
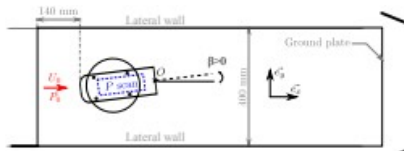


Figure 1. (Left): top, symmetric and bottom exact solutions existing for diffuser flow. The symmetric solution is unstable and in the absence of noise, the flow converges either to the top or bottom solutions. (Right): in the presence of upstream noise, the flow is erratically switching between the top and bottom solutions. The higher the noise, the more often the flow switches.



Simulation des écoulements (1/5)

Stokes après obstacle

$$\begin{cases} \nabla \cdot u &= 0 \\ -\nu \Delta u + \nabla p &= 0 \end{cases}$$

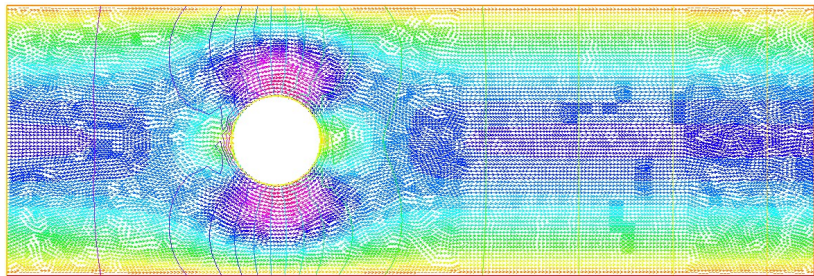


Figure – Setup point-selle d'un lagrangien, éléments finis de Taylor-Hood P2/P1, conditions aux limites de Poiseuille et stabilisation de la pression

Simulation des écoulements (2/5)

Stokes en cavité entraînée

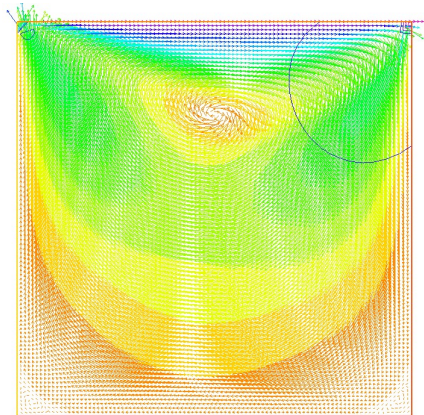
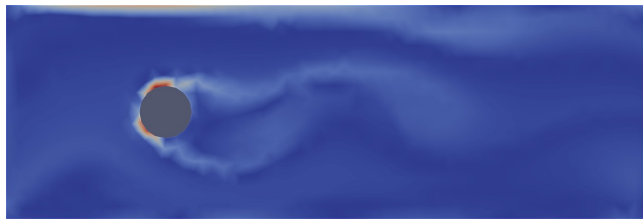
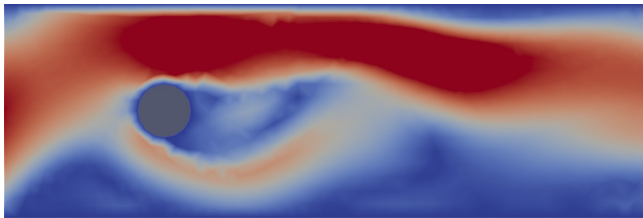


Figure – Schéma d'Uzawa pour le lagrangien augmenté, éléments finis de Taylor-Hood P2/P1, conditions aux limites de vitesse pour la cavité entraînée

Simulation des écoulements (3/5)

Navier-Stokes après obstacle

$$\begin{cases} \nabla \cdot u &= 0 \\ \partial_t u + (u \cdot \nabla) u &= \nu \Delta u - \nabla p \end{cases} \quad w := \partial_y u^x - \partial_x u^y$$



Simulation des écoulements 4/5

Méthode basée sur l'apprentissage statistique

- méthode de Décomposition Orthogonale aux valeurs Propres (POD) : on considère une suite finie de solutions (U_i), appelées *snapshots*, et on cherche une base orthogonale (ψ_j) d'interpolation qui minimise l'erreur sur l'ensemble des U_i .
Pour d modes :

$$\operatorname{argmin}_{\{\psi_j\}} \frac{1}{L} \sum_{i=1}^L \left\| U_i - \sum_{j=1}^d (U_i | \psi_j) \psi_j \right\|^2$$

Simulation des écoulements 5/5

Autres méthodes d'apprentissage statistique

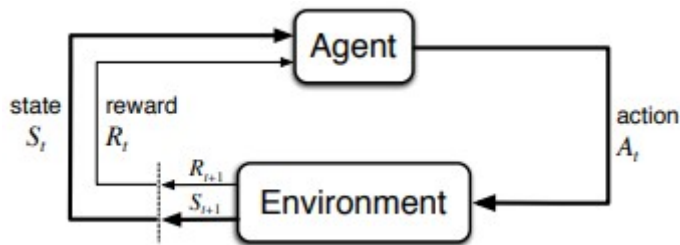
- ▶ méthodes de Décomposition Dynamique en Modes : trouver un opérateur linéaire A tel que la suite $(A^i U_0)_i$ minimise l'erreur des moindres carrés
- ▶ méthodes de Physics-informed Neural Networks : on ajoute à la fonction de coût usuelle la contribution de la norme du résidu

$$J : W, b \mapsto \frac{1}{L_u} \sum_{i=1}^{L_u} \|u(t_u^i, x_u^i) - u^i\|^2 + \frac{1}{L_f} \sum_{i=1}^{L_f} \|f(t_f^i, x_f^i)\|^2$$

où f est un schéma pour le membre de gauche.

Apprentissage par renforcement

Framework



$$g := \nabla_{\theta} \mathbb{E} [\gamma^t r_t]$$

$$\mathcal{L}_{\theta} := -\frac{1}{L} \sum_{i=1}^L A^{\pi}(s_i, a_i) \ln(\pi^{\theta}(a_i | s_i))$$






Conclusion

- ▶ Langage Python3, avec API MPI, PETSc, FEniCs, frameworks de machine-learning...
- ▶ simulations par éléments finis : FEniCs/dolfinx (API PETSc)
- ▶ framework pour l'apprentissage statistique en général : Scikit-learn
- ▶ implémentation de l'environnement : openAI/Gym
- ▶ framework pour le deep-learning : Tensorflow, Pytorch

Autre langage pour le machine-learning : Julia.

Autre langage de haut niveau pour les éléments finis : FreeFEM++.

Sur l'étagère...

-  Pierre Saramito (2016) *Complex fluids Modeling and Algorithms*, [site internet de Rheolef](#)
-  N. Thuerey, P. Holl, M. Mueller, P. Schnell, F. Trost, K. Um (2021) *Physics-based Deep Learning*, [site internet du livre](#)
-  Richard S. Sutton, Andrew G. Barto (2018) *Reinforcement Learning : An Introduction*, The MIT Press
-  Tarik Fahlaoui (2020) *Réduction de modèles et apprentissage de solutions spatio-temporelles paramétrées à partir de données : application à des couplages EDP-EDO*, PhD Thesis.
-  Antoine Dumon (2011) *Réduction dimensionnelle de type PGD pour la résolution des écoulements incompressibles*, PhD Thesis.

et quelques réalisations et recherches personnelles sur
[ma page github](#)

Annexe : Singular Value Decomposition

Soit une matrice de données X telle que le nombre d'attributs est grand devant le nombre d'observations

$$X := \left(\begin{array}{cccc} \vdots & \vdots & \dots & \vdots \\ x^1 & x^2 & \dots & x^m \\ \vdots & \vdots & \dots & \vdots \end{array} \right) \Bigg\} n \text{ features}$$

On appelle décomposition en valeurs singulières de X l'unique triplet (U, Σ, V) tel que $U \in \mathcal{M}_{n \times n}(\mathbb{R})$, $\Sigma \in \mathcal{M}_{n \times m}$ diagonale par blocs et $V \in \mathcal{M}_{m \times m}$ tel que

$$X = \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ u^1 & u^2 & \dots & u^n \\ \vdots & \vdots & \dots & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_m \\ \hline & & & (0) \end{bmatrix} \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ v^1 & v^2 & \dots & v^m \\ \vdots & \vdots & \dots & \vdots \end{bmatrix}^T$$

Annexe : décomposition D.M.D

On obtient un premier exemple de décomposition dynamique en modes en considérant les deux matrices de données

$$X := \begin{pmatrix} \vdots & \vdots & \dots & \vdots \\ x^1 & x^2 & \dots & x^{N_t-1} \\ \vdots & \vdots & \dots & \vdots \end{pmatrix} \quad X' := \begin{pmatrix} \vdots & \vdots & \dots & \vdots \\ x^2 & x^3 & \dots & x^{N_t} \\ \vdots & \vdots & \dots & \vdots \end{pmatrix}$$

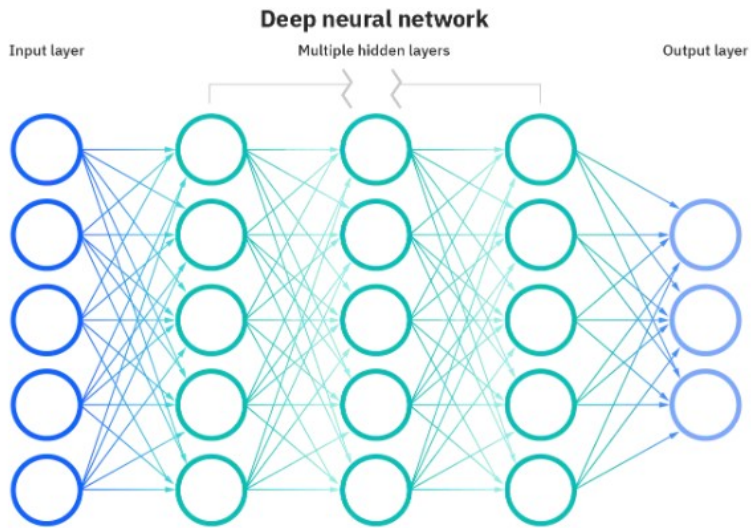
On cherche A telle que $X' \sim AX$. La matrice

$$A := X'V\Sigma^{-1}U^T$$

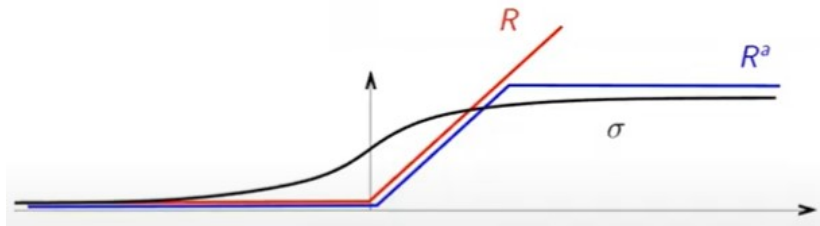
où apparaît la S.V.D de X convient.

Annexe : Réseau de neurones profonds

dense, séquentiel

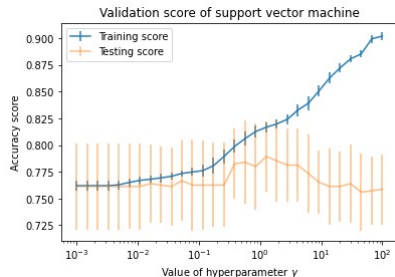
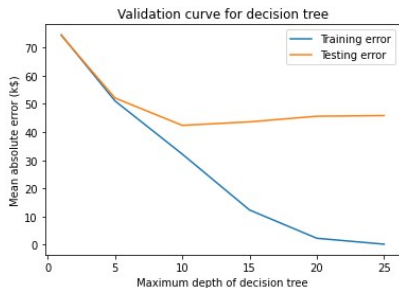


Annexe : exemples de fonctions d'activations



En rouge, RELU, en bleu : RELU saturée, en noir la sigmoïde avec norme inf. de la dérivée inférieure à 1.

Annexe : exemples de courbes d'apprentissage



Annexe : exemple de rapport d'apprentissage pour la sélection de modèles

