# Rapport du Projet

MTH8211

Oussama Mouhtal, Yasmine Amami

## Description du projet

### Assimilation de données

L'assimilation de données consiste à ajuster un modèle physique dépendant d'un paramètre, souvent l'estimation initiale, en utilisant les données et des informations a priori sur le problème. Mathématiquement, on cherche à déterminer le paramètre du modèle physique en minimisant l'écart entre les sorties du modèle et les observations, tout en tenant compte des informations a priori.

$$\min_{x_0 \in \mathbb{R}^n} f(x_0) = \min_{x_0 \in \mathbb{R}^n} \|x_0 - x_b\|^2_{B^{-1}} + \sum_{t_i \in \mathcal{T}} \|y_i - \mathcal{H}_i(\mathcal{M}_{t_0,t_i}(x_0))\|^2_{R_i^{-1}}. \tag{1}$$

Ici, $x_0 = x(t_0)$ est l'état au temps initial $t_0$, par exemple la valeur de la température, $x_b \in \mathbb{R}^n$ représente les informations a priori au temps $t_0$ et $y_i \in \mathbb{R}^{m_i}$ est le vecteur d'observation au temps $t_i$ pour $t_i \in \mathcal{T}$ avec $\mathcal{T} = t_1, t_2, \ldots, t_{nt}$ représente l'ensemble des instants où les observations sont collectés. $\mathcal{M}_{t_0,t_i}(\cdot)$ est un modèle dynamique physique non linéaire qui propage l'état $x_0$ au temps $t_0$ vers l'état $x_i$ au temps $t_i$ en résolvant les équations aux dérivées partielles. $\mathcal{H}_i(\cdot)$ transforme le vecteur d'état $x_i$ en un vecteur de dimension $m_i$ représentant l'état dans l'espace d'observation. $B \in \mathbb{R}^{n \times n}$, $R_i \in \mathbb{R}^{m_i \times m_i}$ sont des matrices symétriques définies positives de covariance d'erreur correspondant aux erreurs du modèle a priori et des observations, respectivement.

### Modèle utilisé

Le modèle que nous utilisons est `LinShallowWater1DModel`. Ce modèle repose sur deux variables principales : la hauteur de la surface de l'eau $\zeta$ et la vitesse de l'eau $u$, avec les équations linéarisées.

Les équations discrétisées sont les suivantes :

1. Mise à jour de la hauteur de la surface de l'eau $\zeta$ à une position donnée indexée par $j$ :

$$\zeta_j^{(new)} = \zeta_j - \frac{\mathrm{dt} \cdot h}{x_u[j+1] - x_u[j]}(u[j+1] - u[j])$$

   Où dt est le pas de temps, $h$ est la profondeur de l'eau, et $x_u$ est l'espace discret pour les vitesses de l'eau.

2. Mise à jour de la vitesse de l'eau $u_j$ :

$$u_j^{(new)} = u_j - \frac{\mathrm{dt} \cdot g}{x_r[j] - x_r[j-1]}(\zeta_i - \zeta_{j-1})$$

   Où $g$ est la constante gravitationnelle et $x_r$ est l'espace discret pour les hauteurs de l'eau.

Le modèle $\mathcal{M}$ utilisé est linéaire, de sorte que :

$$\mathcal{M}_{t_0,t_i}(x_0) = \mathcal{M}(\mathcal{M}(...(\mathcal{M}(x_0)))) = M^i x_0$$

avec $x = [\zeta, u]$.

### Opérateur d'observation

L'opérateur d'observation utilisé ici consiste à extraire les $m_i$ premier élément du vecteur d'état $\mathcal{M}_{t_0,t_i}(x_0)$ à l'instant $t_i$, qui est une opération linéaire.

### Génération des observations

Le modèle est exécuté avec les conditions initiales exactes $xit$, et les observations réelles $y_i$ sont générées par l'opérateur d'observation $\mathcal{H}$ tel que :

$$y_i = \mathcal{H}(\mathcal{M}_{t_0,t_i}(xt)) + \epsilon_{\text{obs}}$$

où $\epsilon_{\text{obs}} \sim \mathcal{N}(0, R_i)$ est un bruit d'observation. Dans ce projet les matrice $R_i = \sigma I_{m_i}$, et $m_i = m.

### Information a priori (background)

En assimilation de données, l'information a priori correspons au vecteur d'état du sytème prédit par l'ancien modèle à l'instant $t_0$,l'information à priori $x_b$ est généré tel que :

$$x_b = xit + \epsilon_{\text{b}}$$

où $\epsilon_{\text{b}} \sim \mathcal{N}(0, B)$ est un bruit. Dans ce projet les matrice $B = \sigma I_n$.

### Résolution du problème d'optimisation

En général, pour résoudre ce problème de moindres carrés non linéaires, on utilise la méthode de Gauss-Newton, avec l'utilisation du gradient conjugué pour résoudre les sous-problèmes quadratiques convexes. Dans notre cas, puisque le modèle est quadratique, nous n'utiliserons qu'une itération de Gauss-Newton et nous utiliserons un solveur itératif pour résoudre le problème de moindres carrés linéaires.

## Problèmatique

La problématique de ce projet consiste à résoudre le problème de moindres carrés linéaires en utilisant plusieurs méthodes itératives de la librairie Krylov et à comparer la performance des différents solveurs avec une implémentation sur GPU.

## Les objectifs : Les algorithmes à tester

Pour l'instant, les algorithmes que nous voulons explorer sont les suivants :

— 'krylov.cg'
— 'Krylov.cglanczos'
— 'Krylov.cgls'
— 'Krylov.cglslanczosshift'
— 'Krylov.lsqr'

De même, nous souhaitons utiliser des préconditionneurs déterministes ou aléatoires existants dans la littérature dans le contexte du gradient conjugué.

## Résultats préliminaires

— Nous avons implémenté le jacobien du ‘LinShallowWater1DModel‘ et, de même, l’adjoint. Dans ‘../examples/example.ipynb‘, nous avons testé l’adjoint et le tangent d’un modèle linéaire ainsi que celui du ‘LinShallowWater1DModel‘. L’implémentation des jacobiens et de l’adjoint est correcte.
— Nous avons utilisé la méthode CG de Krylov pour résoudre le problème quadratique. L’implémentation de la matrice est basée sur le gradient de l’objectif et non sur celui du jacobien du résidu (déjà implémenté). La matrice obtenue à partir de cette implémentation n’est pas symétriquement définie, ce qui peut être dû à des problèmes d’arithmétique flottante. Nous envisageons d’implémenter le jacobien du résidu et son adjoint.

## Plan d’action

— Les données dans ce projet, comme déjà expliqué, seront simulées à partir du modèle ‘LinShallowWater1DModel‘.
— Pour utiliser les algorithmes itératifs déjà mentionnés, nous nous baserons sur la documentation de la librairie Krylov.
— Pour la partie préconditionnement, nous envisageons de tester des préconditionneurs basés sur l’information spectrale estimée par des méthodes aléatoires, et d’utiliser des préconditionneurs basés sur une décomposition incomplète de Cholesky.
— Si nous n’obtenons pas de résultats intéressants avec ce modèle, nous envisageons de résoudre ce problème en utilisant des opérateurs existants en Julia que le professeur pourra nous recommander.

## Lien Github

https ://github.com/Oussamamouhtal/DataAssim.jl