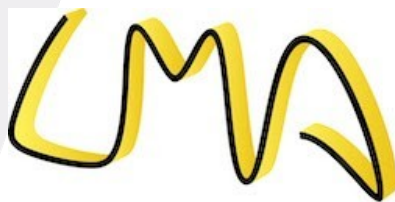


Simulation numérique de la propagation d'ondes sismiques appliquée aux barrages

RAPPORT DE STAGE STG2

Laboratoire de Mécanique et d'Acoustique
4 Impasse Nikola Tesla



Laboratoire de Mécanique et d'Acoustique

GLOSSAIRE

SEM : Spectral Element Method = méthode des éléments spectraux

FEM : Finite Element Method = méthode des éléments finis

FDM : Finite Diffrencial Method = méthode des différences finis

LMA: Laboratoire de Mécanique et d'Acoustique

PML : Perfect Match Layers

CFL : nombre de Courant–Friedrichs–Lewy

ICOLD : International Commission Of Large DAM =Commission internationale des Grands Barrage

BUNCOLD: Bulgarian National Committee on Large Dams = Comité Nationale Bulgare des Grands Barrages

RÉSUMÉ

Ce rapport présente les travaux effectués durant mon stage au Laboratoire de Mécanique et d'Acoustique (LMA) de Marseille, portant sur la simulation numérique de la propagation des ondes sismiques appliquée à l'étude de la résistance de barrages. Mon objectif principal était de générer un maillage hexaédrique du barrage de Tsankov Kamak et de son environnement, en tenant compte de la topographie réelle. Pour cela, j'ai utilisé GMSH pour la création du maillage Python, pour y faire des modifications difficiles à faire sur GMSH. J'ai pu aussi lancer sur le mésocentre d'Aix-Marseille certains calculs, que je n'ai pas fait mention sur ce rapport.

En parallèle, j'ai mené une étude de stabilité pour des schémas numériques d'ordres supérieurs en temps, en utilisant une approche par équation modifiée en temps et une discrétisation à l'aide de la méthode des éléments spectraux en espace, pour évaluer les gains en stabilité suivant l'ordre d'approximation en temps. Cette analyse a permis de démontrer que l'augmentation de l'ordre d'approximation permet une meilleure stabilité, bien que les gains deviennent de plus en plus faibles pour des ordres très élevés.

Sommaire

Glossaire	2
Résumé.....	3
1 Introduction.....	5
1.1 Contexte et sujet du stage:.....	5
2 Présentation du cadre du stage.....	5
2.1 À propos du Laboratoire.....	5
2.1.1 Missions et domaines de recherche du laboratoire.....	5
3 Ma Mission.....	6
3.1 Sujet de mon travail.....	6
3.1.1 Sujet.....	6
3.1.2 Logiciels et matériels utilisés.....	7
3.2 Objectifs.....	8
3.2.1 Simulation.....	8
3.2.2 Étude de stabilité pour des ordre en temps supérieurs.....	8
3.3 Déroulement.....	9
3.3.1.Organisation.....	9
3.3.2 Premiers pas.....	9
4 Travail Effectué.....	11
4.1 Modélisation du barrage.....	11
4.1.1 Géométrie du barrage.....	11
4.1.2 Ajout de Perfect Match Layers (PML).....	14
4.1.3 Importation de la courbure du barrage et de la topographie avoisinante.....	14
4.2 Étude de stabilité.....	16
4.2.1 Apprentissage des bases théoriques préliminaires.....	16
4.2.2 Définition du problème et implémentation solution exacte.....	18
4.2.3 Implémentation méthode des différences finies (FDM).....	21
4.2.4 Implémentation méthode des éléments spectraux (SEM) en espace et différences finies en temps.....	24
4.2.5 Gains apportés par avec une approximations en temps d'ordre supérieurs : Méthode d'approche par équation modifiés.....	27
5.Conclusion.....	30
6 Tableaux des figures.....	31
7 Références bibliographiques.....	32

1 Introduction

1.1 CONTEXTE ET SUJET DU STAGE:

La simulation numérique de propagation d'ondes mécanique permet dans certains contextes de pouvoir analyser comment les ondes se comportent en présence de structures spécifique et inversement de pouvoir analyser le comportement de ces structures lorsqu'elles sont traversées par ces ondes. Cependant, dans des milieux où l'on est susceptible d'avoir des éléments avec des tailles d'ordre de grandeur différents, on peut rencontrer certains problèmes pour mettre en œuvre ces simulations. Une application concrète est l'étude des effets d'un séisme sur un barrage et la zone environnantes, ainsi le barrage, faisant 130 m de hauteur et 340 m de largeur, vis à vis de la topographie d'une dizaine de kilomètres a une taille très petite. Le laboratoire de Mécanique et d'Acoustique de Marseille faisant des recherches dans le domaine de la propagation des ondes mécaniques dans des milieux complexe, le développement d'un code de simulation dans ce contexte s'inscrit alors dans son activité de recherches.

2 Présentation du cadre du stage

2.1 À PROPOS DU LABORATOIRE

2.1.1 Missions et domaines de recherche du laboratoire.

Le LMA est une unité mixte de recherche AMU-CNRS-Centrale Méditerranée, UMR 7031. Ses principaux domaines de compétences sont la mécanique du solide (structures, matériaux, interfaces) et l'acoustique (propagation des ondes dans des milieux fluides et solides complexes).

Le laboratoire est structuré en trois équipes de recherche. Les actions de l'équipe **Matériaux & Structures** concernent l'étude théorique des milieux solides, leur caractérisation expérimentale et l'établissement de modèles de comportement. L'équipe **Ondes et Imagerie** est spécialisée dans l'étude de la propagation des ondes mécaniques afin d'imager, caractériser et/ou contrôler de manière non-invasive les milieux fluides et solides complexes. L'équipe **Sons** fédère des travaux menés sur les sons audibles, du point de vue de leur production, de leur contrôle (actif ou passif) et de leur perception (subie ou souhaitée).

Le LMA à un personnel d'environ 140 personnes et bénéficie d'un ensemble exceptionnel (plus de 5 000 m²) de plateformes et centres de ressources expérimentales organisés en 5 champs disciplinaires thématiques. Il dispose alors d'infrastructure telle que des chambres anéchoïques et réverbérantes pour des tests acoustiques, des bancs d'essais pour des tests mécaniques, et des plateformes de calcul pour faire tourner des simulations numériques.

2.1.2 Équipe Ondes et Imagerie

L'équipe Ondes et imagerie est spécialisée dans l'étude de la propagation des ondes mécaniques dans les milieux complexes. Elle développe des recherches visant à modéliser, caractériser et imager de manière non-invasive les milieux naturels, biologiques et manufacturés au moyen des ondes. Le spectre de ses activités de recherche est large, allant de la modélisation phénoménologique à l'étude expérimentale, en passant par le développement de codes de simulation numérique et le calcul haute performance.

Les domaines d'application concernent l'environnement (imagerie sismique, réduction de la pollution sonore sous-marine, matériaux biosourcés pour la réduction du bruit), le génie civil et nucléaire (contrôle et suivi de l'intégrité des structures), le transport (réduction des vibrations ferroviaires), la santé (imagerie diagnostique et thérapie par ultrasons) et l'archéologie (imagerie de statues antiques).

3 Ma Mission

3.1 SUJET DE MON TRAVAIL

3.1.1 Sujet

Le sujet de mon stage porte sur la simulation numérique de propagation d'ondes, dans des géométries à des degrés de raffinements différents, tels que la simulation de tremblement de terre appliqué à la résistance de barrages. Le barrage étudié lors de mon stage est le barrage de tsankov en Bulgarie.



Figure 1: Vue de la zone avoisinant le barrage de tsankov
(Google Maps)



Figure 2: Vue du Barrage de Tskankov

Ainsi alors que l'on a une zone d'étude de l'ordre d'une dizaine de kilomètres comme on peut le voir sur l'échelle à gauche de la figure 1, on est obligé de prendre un petit pas d'espace du fait de la présence du barrage qui a une taille de l'ordre du mètre. Cela constitue un premier « obstacle », car pour des raisons de stabilité, on est alors contraint de prendre un pas de temps plus petit. Le coût en ressource de calcul est alors très grand.

Pour pouvoir simuler un tremblement de terre, le laboratoire utilise un logiciel de calculs mettant en œuvre la méthode des éléments spectraux pour une discrétisation en espace et une méthode de Newmark prédicteur- correcteur en temps. La première difficulté étant que la topographie autour d'un barrage est généralement complexe. Ainsi utiliser une méthode de différences finies en espace serait beaucoup plus difficiles d'implémentation qu'en utilisant les éléments finis ou les éléments spectraux .

Le type d'équation que l'on doit résoudre :

$$\frac{\partial^2 u}{\partial t^2} - c^2 \Delta u = f$$

En mettant en œuvre SEM et FEM on a alors :

$$M \ddot{U} + KU = F$$

Avec M la matrice de masse et K la matrice de rigidité.

Il est préférable d'utiliser la méthode des éléments spectraux, car la matrice de masse est diagonale et le produit matrice-vecteur à une meilleure complexité qu'avec les éléments finis[2], lorsque on utilise des polynôme d'approximation de haut degrés. Par conséquent, dans un but de réduire, le temps de calcul et la quantité de ressources machine, on utilisera la méthode des éléments spectraux avec des polynômes d'approximation de hauts degrés[1] .

3.1.2 Logiciels et matériels utilisés

Tout au long de mon stage, je serai amené à utiliser différents logiciels et outils.

SEM3D : C'est l' algorithme de simulation numérique de tremblement de terre SEM3D issue d'une coopération de l'Institut de Physique du Globe de Paris (IPGP), CEA ET Central Supélec, et du CNRS.

GMSH : c'est un logiciel qui permet de générer des maillage

Paraview: logiciel permettant de visualiser certains résultats

Mésocentre AMU : Ensemble d'ordinateur pour mener des calcul haute performance.

Dans ce rapport, je vais surtout détailler mon travail sur la production d'un maillage sur GMSH et sur la mise en place de petite simulation pour une étude de stabilité à l'aide de Python.

3.2 OBJECTIFS

3.2.1 Simulation

Le premier objectif est d'effectuer la simulation d'un tremblement de terre pour le barrage de Tsankov pour cela il faudra :

- Dans un premier produire un maillage avec uniquement des éléments hexaédrique du barrage respectant la forme du barrage de Tsankov Kamak et la topologie avoisinante.
- Dans un seconde temps faire une simulation précise, lors d'un tremblement de terre, grâce aux logiciels de calcul SEM3D et au mésocentre.

L'objectif au-delà de mon stage étant que l'on puisse réutiliser mon travail pour pouvoir réitéré le processus plus simplement pour d'autres barrages. La méthode de discrétisation en espace utilisée et la méthode des éléments spectraux (SEM), et une méthode de différence finies en temps. Les calculs vont être effectués aux mésocentre d'Aix-Marseille Université, car du fait de la taille totale du domaine (une dizaine de kilomètres) les calculs ne pourront tourner assez rapidement dans un ordinateur ou un simple serveur de calcul.

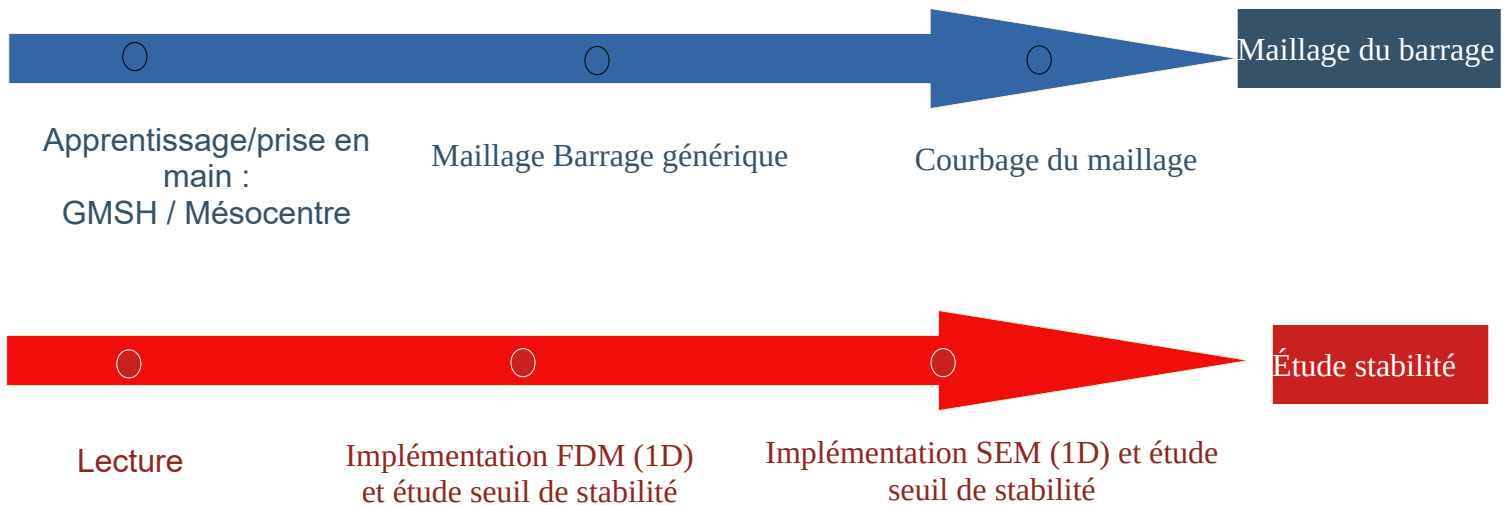
Je ne vais pas détailler les simulations que j'ai pu lancer sur le mésocentre en utilisant notamment SEM3D , car n'ayant pas complètement fini la production du maillage, ces simulations ont été faites dans la plupart des cas pour apprendre à utiliser SEM3D et comment lancer des calculs dans un cluster de calcul.

3.2.2 Étude de stabilité pour des ordre en temps supérieurs

Le second objectif, parallèlement à cela, est d'étudier la stabilité et les performances d'autres méthodes numériques sur des cas simples (1D), principalement la méthode des éléments spectraux avec une discrétisation en temps à l'aide de l'approche par équation modifiés pour des ordres supérieurs. Le but, au-delà de mon stage, c'est de voir jusqu'à quel ordre on gagne significativement en stabilité (un CFL qui augmente). On souhaite rester stable est respecté la condition de stabilité, car pour un schéma explicite, si l'on ne respecte pas cette condition de stabilité au fil des itérations l'erreur croit de manière incontrôlée. Nous verrons dans la partie 4 que la condition de stabilité s'exprime grâce aux $CFL = c \cdot \Delta t / \Delta h$ et on est stable tant que $CFL < \alpha_{max}$ (Δt pas de temps, Δh taille d'un éléments, c célérité de l'onde). Dans ce type de problème où l'on est confronté à des éléments de tailles différentes, tout les éléments doivent avoir le α_{max} du plus petit élément pour rester stable. Ainsi, avoir une méthode d'approximation en temps qui a le α_{max} plus grand, permettrait de pouvoir augmenter le pas de temps sur toute la simulation tout en restant stable. Dès lors, après-avoir implémenté cette méthode d'approximation en temps dans SEM3D que l'on puisse ainsi augmenter le pas de temps, toujours dans un souci d'économiser des ressources de calculs.

3.3 DÉROULEMENT

3.3.1. Organisation



J'ai alors deux objectifs principaux :

- Faire un maillage hexaédrique du barrage et effectuer les calculs
- Faire une étude de stabilité avec la méthode des éléments spectraux pour des hauts degrés d'approximation en temps.

Les premières semaines, j'ai consacré mon temps à apprendre en autonomie à utiliser différents logiciels.

3.3.2 Premiers pas

J'ai consacré les premières semaines à me familiariser avec les différents logiciels que j'étais amenés à utiliser, le plus important étant GMSH pour que je puisse commencer à réaliser le maillage du barrage. GMSH est un logiciel open source permettant de générer des maillages à partir de codes. Il existe différentes bibliothèques qui permettent de l'utiliser avec d'autres langages comme Python, mais j'ai jugé qu'il était plus facile d'utiliser directement le langage (.geo) propre aux logiciels.

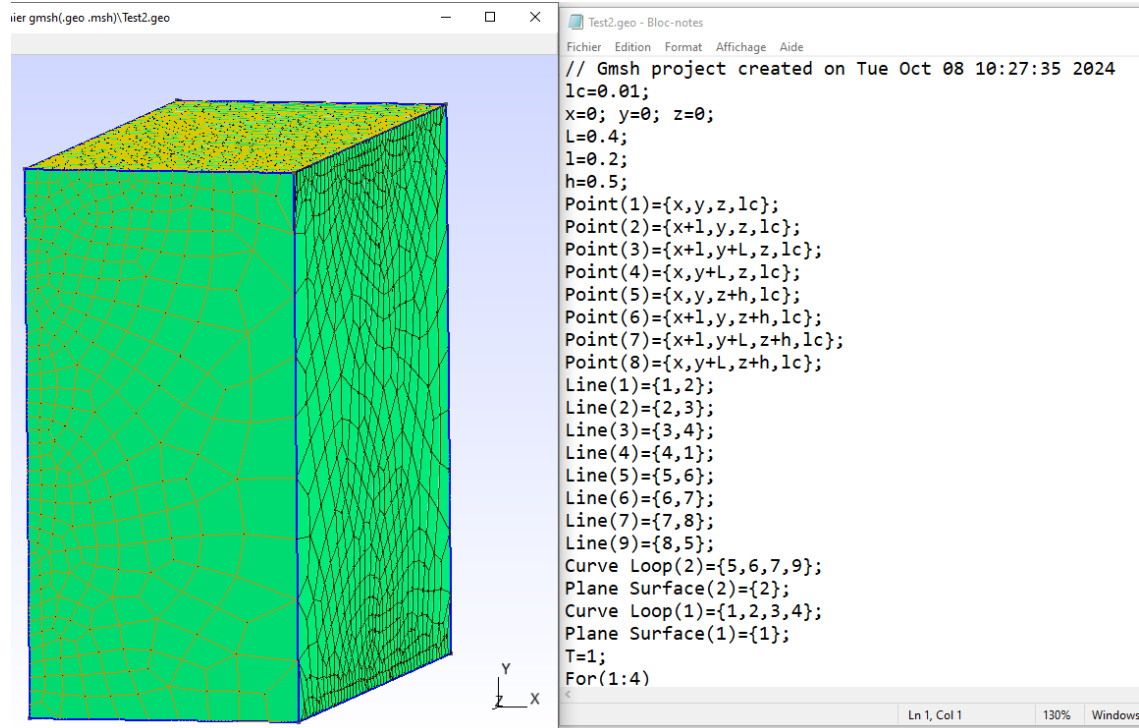


Figure 3: Maillage d'un cube et son code (maillage d'entraînement)

Voici un exemple de maillage simple que j'ai pu réaliser lors de ma phase d'apprentissage et une partie du code que j'ai écrit pour le générer. J'ai pu alors me rendre compte que le logiciel est assez fastidieux à manipuler, puisqu'il faut détailler point par point, ligne par ligne en tenant compte de l'ordre pour générer une surface ou un volume.

Parallèlement à la prise en main de GMSH, mon tuteur m'a donné à lire [1 Higher Order Numericals Method for Transient Wave Équations de Gary C.Cohen dans le but de voir et de mieux comprendre les différentes méthodes numériques utilisées pour résoudre des problèmes de propagations d'ondes, leurs avantages inconvénients à différents niveaux (stabilité, dispersion, complexité...).

4 Travail Effectué

4.1 MODÉLISATION DU BARRAGE

4.1.1 Géométrie du barrage

Le barrage de Tsankov Kamak est un barrage à voûte à double courbure, c'est-à-dire arqué selon le plan horizontal et le plan vertical. Le barrage de rétention a une hauteur de 130 m et une longueur de près 400 m. Il permet, grâce à la topographie du terrain et la taille imposante du mur de rétention de contenir plus de 111 000 000 m³ d'eau. La géométrie a réalisé pour la modélisation est alors de prime abord assez simple comme on peut le voir sur cette figure.

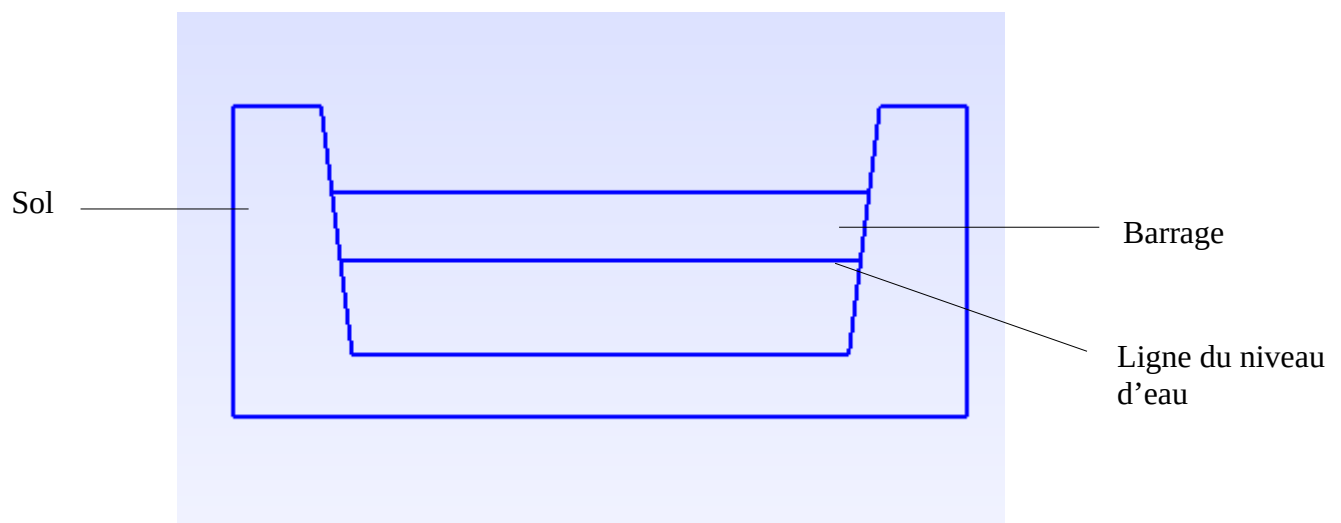


Figure 4: Modèle du domaine du barrage simple (avant extrusion)

Pour pouvoir réaliser le maillage des trois éléments, sol, barrage, eau, on trace la face du barrage puis réalise un maillage 2D de l'ensemble puis on extrude les différentes surfaces pour avoir notre maillage en 3D. Cependant, avec un tel modèle, même en utilisant les différentes fonctions de GMSH, on ne peut pas garantir que l'on puisse mailler le barrage avec des éléments hexaédriques uniquement. Il faut alors rajouter, des lignes par nous-même pour pouvoir « forcer » le maillage avec des éléments hexaédrique, quitte à ce que générer des surfaces et donc des volumes supplémentaires, ce qui ne pose pas de problème, car il suffira de rassembler les volumes dans des groupes correspondant à nos éléments réels (eau, barrages, sol). Tout le travail que j'ai effectué est disponible sur mon Github¹.

1: [Mon github](#)

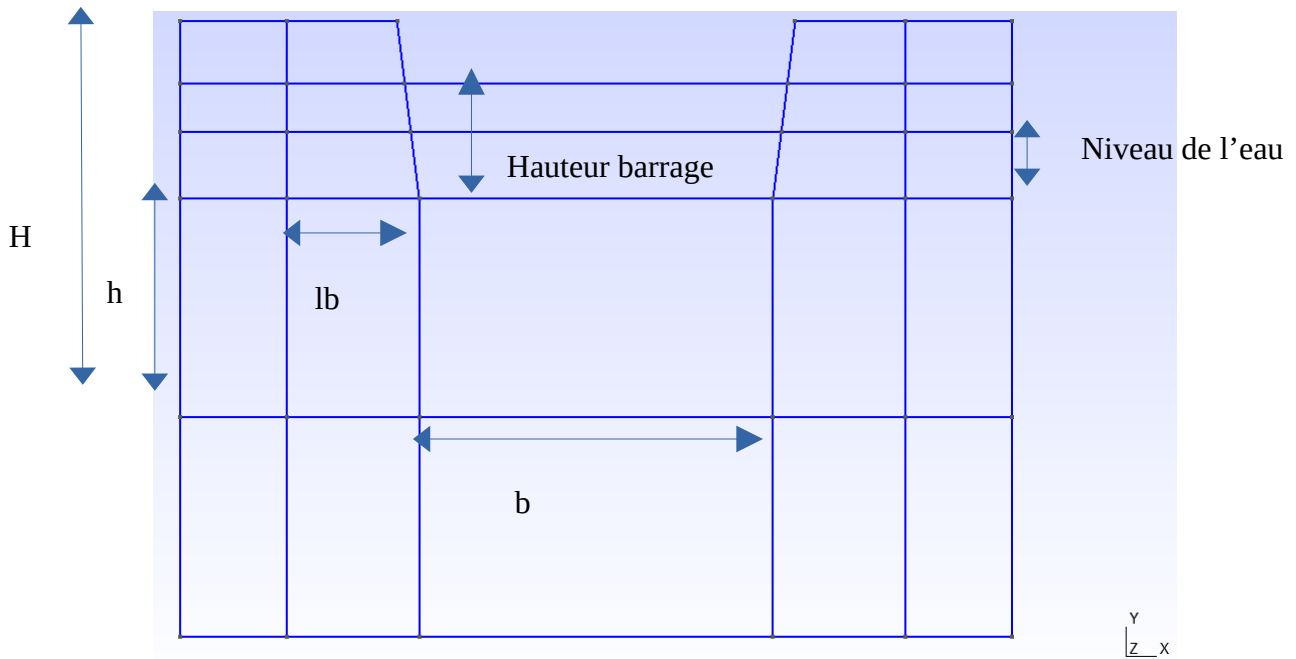
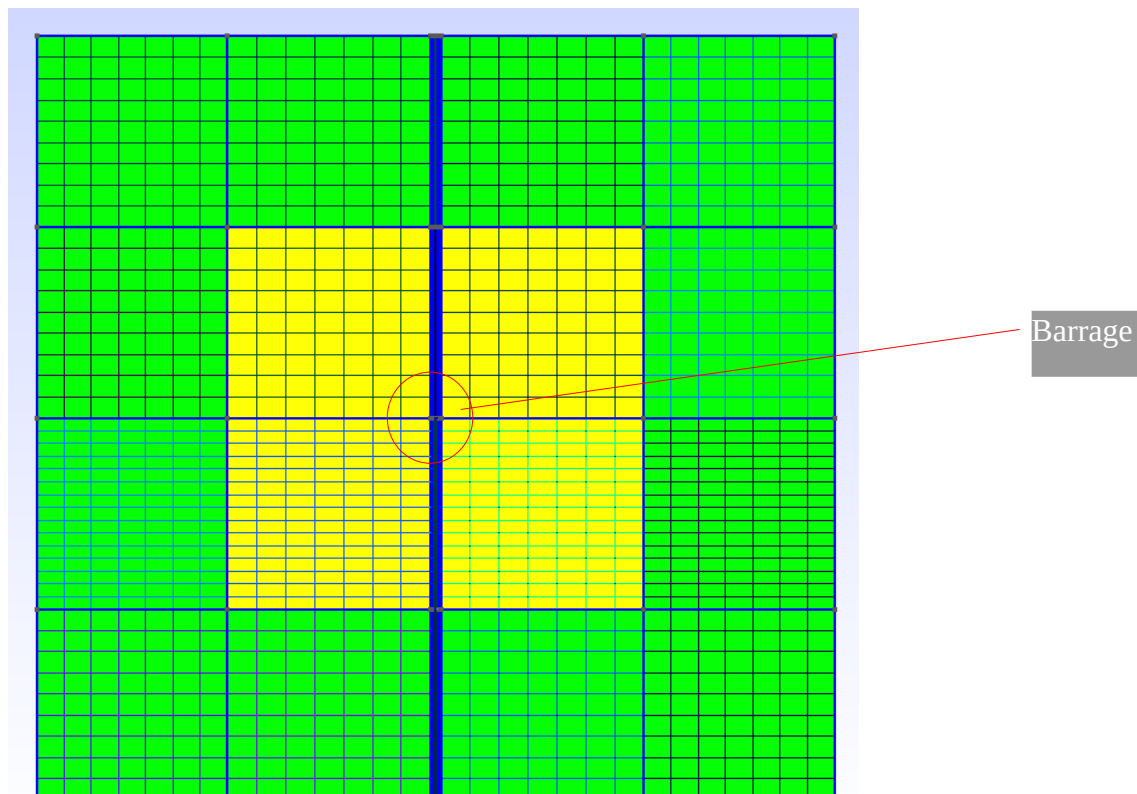


Figure 5: Modèle du domaine final

Ainsi sur la figure 5 voici le modèle final du barrage² avec les noms des variables que j'ai utilisés dans mon fichier .geo. Pour fixer les valeurs des grandeurs je me suis servie du site de la Commission Internationale des Grands Barrages (ICOLD)³.



2: [Mon github\stage\fichier.geo](#) Barage_V3.geo

3 [Site ICOLD](#)

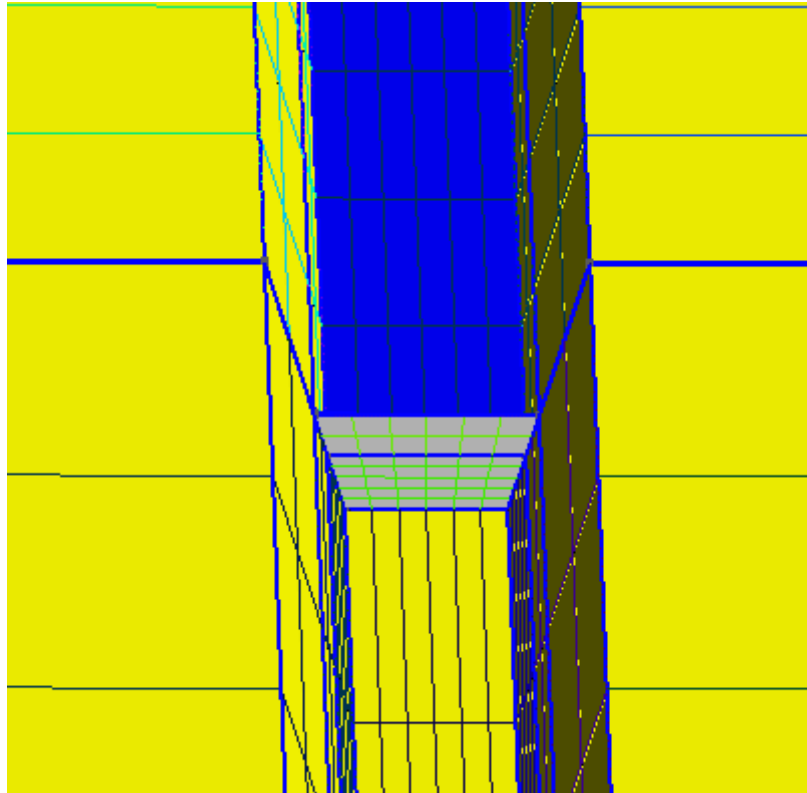


Figure 6: Vue du maillage du barrage

Sol en jaune ; Barrage en gris ; PML en vert

Sur ces vues du maillage, on peut bien se rendre compte de la différence de tailles entre le barrage et la taille du domaine. Pour déterminer un ordre de grandeur du nombre d'éléments qu'il faut pour avoir une bonne simulation (avec assez d'éléments pour la précision, sans en mettre trop pour éviter d'économiser des ressources de calculs), je me suis aidé des vitesses des ondes de cisaillement et des ondes transversales (ondes p et ondes s) dans les différents matériaux. En considérant que les milieux sont homogène, isotrope, linéaire et qu'on se place sous de petites déformations. Je les ai calculées à partir du module de Young E^4 , du coefficient de Poisson⁵ et de la masse volumique de chaque matériau à savoir l'eau, le sol, et le béton. J'ai repris les valeurs qu'avait trouvées un ancien stagiaire travaillant sur un sujet similaire. Les informations du Barrages (géométrie, maillage, composition béton) sont fourni par ICOLD.

	Vp (m/s)	Vs (m/s)
Béton	3992	2370
Sol	6000	2650
Eau	1400	0

$$v_p = \sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}}$$

$$v_s = \sqrt{\frac{E}{2\rho(1+\nu)}}$$

4: [Source Module de Young Sol](#)

5: [Source coefficient de Poisson](#)

4.1.2 Ajout de Perfect Match Layers (PML)

On souhaite simuler la propagation d'ondes sismiques dans un domaine qui est fini, si nous effectuons une simulation avec uniquement un domaine fini, les ondes rebondiront aux limites du domaine ce qui ne correspond à une situation physique. Les Perfect Match Layers (PML) sont alors ajoutées à un maillage pour simuler des conditions aux limites absorbantes, en particulier pour les simulations d'ondes (comme les ondes électromagnétiques ou acoustiques). Ils permettent de réduire les réflexions des ondes aux bords du domaine de simulation. En introduisant des couches PML autour du domaine, les ondes qui atteignent ces frontières sont progressivement atténuées, ce qui évite les artefacts de réflexion non-physiques qui pourraient fausser les résultats. Cela permet de simuler un espace infini sans avoir à étendre indéfiniment le domaine de calcul. Une fois ajouter au maillage, il faut indiquer générer un fichier texte (material.input) qui indiquera aux logiciel SEM 3D où se trouve chaque bloc de PML et avec quel matériau il est en contact.

4.1.3 Importation de la courbure du barrage et de la topographie avoisinante

Maintenant que le maillage hexaédrique du domaine est généré, mon prochain objectif est de le déformer afin qu'il prenne la courbure réelle du barrage, et que le sol et l'eau prenne également la topographie réelle autour du barrage de tsankov. Pour cela, on dispose des données fournis par BOUNDCLOUD à savoir un fichier de maillage (.msh) du barrage uniquement et des données de la surface topographique de la zone avoisinante contenu dans un fichier(.stl). Pour appliquer la topographie et la courbure du barrage sur mon maillage, j'ai réutilisé le travail d'un ancien stagiaire qui a travaillé au LMA sur cette problématique et l'ai adapté à mon maillage.

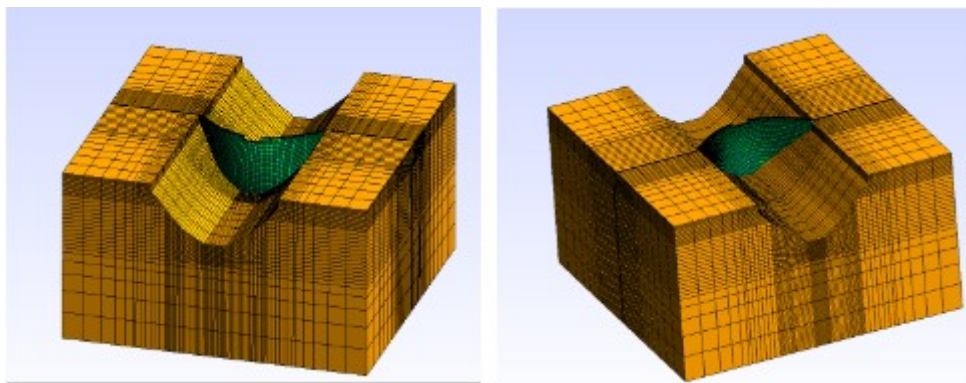


Figure 7: Fichier fournis par BOUNDCLOUD

À ce stade GMSH ne permet pas de déformer les nœuds d'un fichier de maillage, il va alors falloir directement écrire dans le fichier .msh pour appliquer les déformation. Je me suis donc appuyé sur le travail de Vitaly, l'ancien stagiaire qui a développé un ensemble d'outil pour modifier et créer des fichier contenant un maillage (ce sont des fichier texte) grâce à des programmes python et à l'API python de GMSH. Le principe pour déformer le barrage étant de sélectionner les points sur les deux faces à partir du maillage du barrage réel fournis, puis à partir de ces deux ensembles de points, créer deux fonctions d'interpolation qui vont nous permettre de récupérer la quantité dont il faut déplacer chaque point du barrage droit selon un axe pour lui donner la forme et la courbure souhaitée. L'importation de la topographie suit cette même logique mais à été faites d'une autre manière par l'ancien stagiaire, grâce aux

github⁶ qu'il a laissé j'ai pu réutiliser l'ensemble de ses codes pour pouvoir appliquer la topographie sur mon maillage.

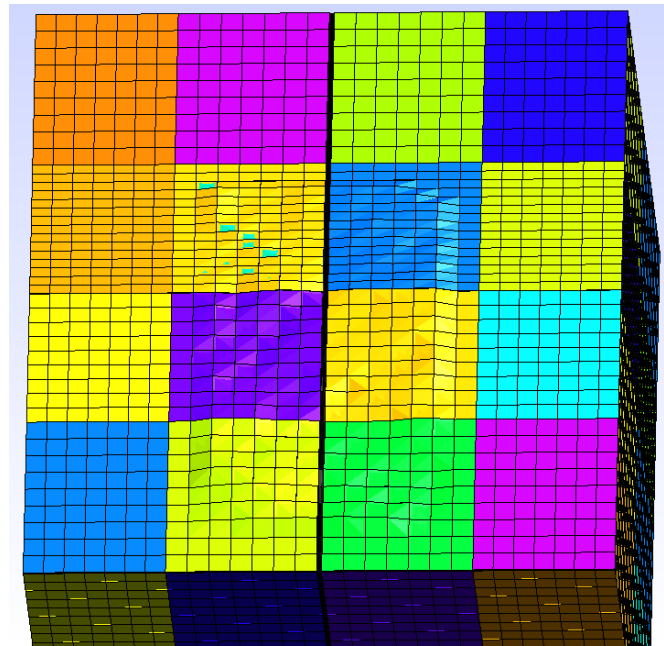


Figure 8: Maillage après application de la topographie avoisinante

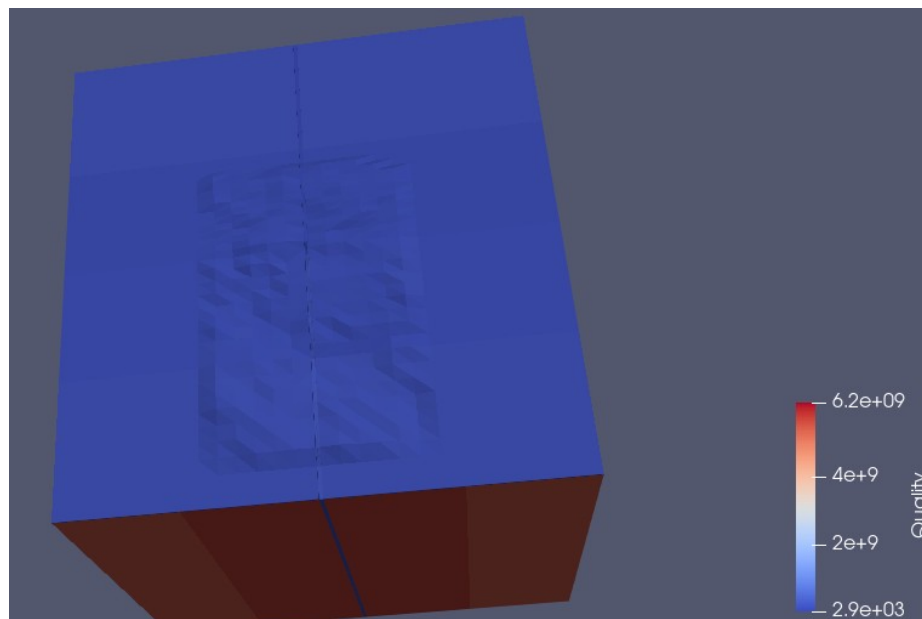


Figure 9: Vue du déterminant Jacobien des éléments hexaédrique du maillage sur Paraview

6: [Github Vitaly \(ancien stagiaire\)](#)

On peut voir sur la figure 8 que le maillage a bien été déformés. Un des problèmes qu'avait rencontré l'ancien stagiaire était que certains éléments s'inversaient, sur la figure 9, j'ai appliqué un filtre pour voir le déterminant de la matrice jacobienne des éléments hexaédrique, il n'y a aucune valeur négative donc aucun élément n'a été inversé. Cependant, une partie du PML a été déformée, ce qui montre que la topographie n'a pas été correctement importée.

4.2 ÉTUDE DE STABILITÉ

4.2.1 Apprentissage des bases théoriques préliminaires

À mon arrivée au Laboratoire de Mécanique et d'Acoustique (LMA), je ne disposais pas des connaissances nécessaires pour mener une étude approfondie de la stabilité des équations d'onde. Afin de combler cette lacune, j'ai entrepris une lecture approfondie de l'ouvrage Higher-Order Numerical Methods for Transient Wave Equations [1]. Ce travail m'a permis d'acquérir les bases théoriques nécessaires pour comprendre les concepts fondamentaux liés à la stabilité numérique.

Dans un premier temps, j'ai choisi de m'entraîner et d'expérimenter avec des schémas de différences finies en temps et en espace d'ordre 2. Ces tests m'ont permis de vérifier et d'explorer les limites de stabilité associées à ces méthodes. Une fois ces résultats confirmés, j'ai pu aborder les méthodes des éléments spectraux en espace avec une meilleure compréhension des enjeux et des défis liés à la stabilité numérique. Dans la partie qui suit, je vais reprendre les principaux résultats [1.chap 5-6] sur la stabilité des schémas aux différences finis.

Considérons une équation d'onde scalaire en une dimension donnée par :

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 \quad (1) \text{ où } c \text{ est la vitesse de propagation de l'onde. Soit un maillage}$$

régulier de \mathbb{R} avec un pas d'espace $h > 0$, composé des points $x_j = jh$, $j \in \mathbb{Z}$. La solution $u(x, t)$ est approchée par une fonction discrète $u_j(t)$, représentant $u(x_j, t)$. Pour discrétiser l'équation en espace on utilise une approximation d'ordre 2 pour le terme de la dérivée spatial :

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{x=x_j} \approx \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2}$$

Ainsi l'équation devient :

$$\frac{\partial^2 u_j}{\partial t^2} - c^2 \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} = 0$$

Supposons une solution sous la forme d'une onde plane sous la forme $u = e^{i(\omega t - kx)}$, où ω est la pulsation de l'onde et k le nombre d'onde, on a alors :

$$u_j(t) = e^{i(\omega t - kx_j)} = e^{i\omega t} e^{ikjh}$$

$$-\omega^2 e^{i(\omega t - kx)} + c^2 \frac{e^{i(\omega t - kx_{j+1})} - 2e^{i(\omega t - kx_j)} + e^{i(\omega t - kx_{j+1})}}{h^2} = 0$$

en utilisant les relations $x_{j+1} = x_j + h$ et $x_{j+1} = x_j - h$ on peut simplifier :

$$-\omega^2 + c^2 \frac{e^{-ikh} + e^{ikh} - 2}{h^2} = 0$$

$$\text{Avec } e^{-ikh} + e^{ikh} = 2 \cos(kh) = 2 \left(1 - 2 \sin^2\left(\frac{kh}{2}\right)\right)$$

$$\omega^2 = \frac{4c^2}{h^2} \sin^2\left(\frac{kh}{2}\right) \quad (1.1) \quad \text{Relation de dispersion pour l'équation discrète en espace}$$

On peut obtenir une relation pour un schéma en temps et en espace de la même manière :

$$\frac{u_j^{n+1} - 2u_j^n + u_j^{n+1}}{\Delta t^2} - c^2 \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2} = 0 \quad \forall (j, n) \in \mathbb{Z} \times \mathbb{Z}$$

avec $u_j^n = e^{i(\omega n \Delta t - kjh)}$

on trouve la relation suivante d'une manière similaire

$$\frac{4}{\Delta t^2} \sin^2\left(\frac{\omega \Delta t}{2}\right) = \frac{4c^2}{h^2} \sin^2\left(\frac{kh}{2}\right) \quad (1.2)$$

$$\sin^2\left(\frac{\omega \Delta t}{2}\right) = \frac{\Delta t^2 c^2}{h^2} \sin^2\left(\frac{kh}{2}\right) = \alpha^2 \sin^2\left(\frac{kh}{2}\right) \quad (1.3)$$

on peut remarquer dans la relation (1.3) que lorsque Δt tends vers 0 alors on retrouve la relation (1.1).

L'approche pour l'étude de stabilité est basée sur des solutions d'ondes planes, qui peuvent être considérés comme des solutions élémentaires des équations d'ondes. Considérons la solution d'onde plane d'une équation d'onde : $u = e^{i(\omega t + kx)}$. Puisque les deux célérités de l'onde sont $\pm \sqrt{\omega/k}$, il faut que ω appartienne à \mathbb{R}^{+*} . Maintenant, si nous fixons k dans la relation de dispersion d'un schéma, ω devient une fonction de k , h et Δt qui pourrait, pour certaines valeurs de h et Δt , être à valeurs complexes. Si nous prenons la relation de dispersions du second ordre en temps et en espace (1.3) :

$$\sin^2\left(\omega \frac{\Delta t}{2}\right) = \alpha^2 \sin^2\left(\frac{kh}{2}\right) \text{ Évidemment pour } \alpha=2 \text{ on a } \sin^2(\omega \Delta t/2) \text{ qui est plus grand que 1}$$

cela implique que ω est complexe, ω perd sa signification physique et on a alors :

$$\omega = \omega_r + i\omega_i \quad \text{En fonction du signe de } \omega_i, \text{ on obtient des ondes évanescentes } (\omega_i > 0), \text{ ou}$$

$$u = e^{-i\omega_i t} e^{i(\omega_r t + kx)}$$

des ondes qui croissent de façon exponentielle ($\omega_i < 0$). Ainsi pour un schéma aux différences finies centrés d'ordre 2 en temps et en espace, on a :

$$4 \sin^2\left(\frac{\omega \Delta t}{2}\right) = \alpha^2 4 \sin^2\left(\frac{kh}{2}\right) \text{ Pour que } \omega \text{ soit réel il faut que } 0 < \sin^2(\omega \Delta t/2) < 1, \text{ on écrit le}$$

CFL de telle sorte que : $\alpha < \alpha_{\max} = 1$

Mon tuteur m'a alors encouragé à mettre en œuvre un algorithme de différences finies de même ordre en temps et en espace pour pouvoir mettre en œuvre une étude de stabilité pour un cas simple et de pouvoir alors vérifier mes résultats.

4.2.2 Définition du problème et implémentation solution exacte

Dans un premier temps, je vais résoudre un problème de propagation d'ondes ayant une solution exacte pour pouvoir, dans un second temps, comparer avec un algorithme de différences finies que j'aurais implémenté. Le problème que l'on va résoudre et le suivant (conditions de Neumann) :

$$\left\{ \begin{array}{l} \frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 \quad x \in [0, L] \\ \frac{\partial u}{\partial t}(x, 0) = 0 \\ \frac{\partial u}{\partial x}(L, t) = 0 \\ \frac{\partial u}{\partial x}(0, t) = 0 \\ u(x, 0) = 0 \\ u(a, t) = H(T - t) A \sin(\omega t) \delta(x - a), \quad \text{avec } T = \frac{2\pi}{\omega} \end{array} \right.$$

avec $u(x, t)$ le déplacement suivant la verticale le long d'un milieu 1D homogène isotropes, c la célérité de l'onde. À noter que au lieu de résoudre directement un problème avec un terme source au niveau de l'équation d'onde (imposé une force en $x=a$), nous résolvons cette équation avec une condition sur le point en $x=a$. En effet, j'ai rencontré des problèmes avec l'autre formulation au niveau de l'implémentation avec les différences finies Monsieur De-abreu-Correa l'un de mes tuteurs m'a alors suggéré de plutôt résoudre le problème sous cette formulation.

Solution exacte pour un domaine infini :

$$u(x, t) = \frac{A}{2c} H\left(t - \frac{|x-a|}{c}\right) \sin\left(\omega\left(t - \frac{|x-a|}{c}\right)\right) H\left(T - t + \frac{|x-a|}{c}\right)$$

En revanche, pour simuler la propagation de l'onde dans un domaine finie il faudra alors prendre en compte les réflexion qui s'effectuent sur les bords en $x=0$ et en $x=L$. On a alors

$$u(x, t) = \sum_{n=-\infty}^{+\infty} (-1)^n \frac{A}{2c} H\left(t - \frac{|x-x_n|}{c}\right) \sin\left(\omega\left(t - \frac{|x-x_n|}{c}\right)\right) H\left(T - t + \frac{|x-x_n|}{c}\right)$$

x_n est la position de la source après n réflexions :

$$x_n = a + (-1)^n (x - a + nL)$$

Il faut alors que j'implémente un algorithme permettant de calculer les déplacements sur un milieu fini discrétisée en espace et en temps. Ainsi, on pourra comparer la matrice obtenue pour la solution exacte et les matrices obtenues avec les différentes méthodes de simulation en des temps et lieux identiques.

J'ai alors écrit deux fonctions :

```
# Fonction de déplacement u(x, t) dans un domaine fini
def deplacement_finite_domain(t, x, a, c, A, omega, T, L):
    delay = np.abs(x - a) / c # Temps de retard
    u = np.zeros_like(t)

    # Onde seulement si t > retard et t < retard + T, avec réflexions aux bords
    for n in range(-5, 6): # Réflexions multiples (n = nombre de réflexions)
        x_reflected = a + (-1)**n * (x - a + n * L) # Position réfléchie
        delay_reflected = np.abs(x_reflected - a) / c
        mask = (t >= delay_reflected) & (t <= delay_reflected + T)
        u[mask] += A * np.sin(omega * (t[mask] - delay_reflected))

    return u
```

Figure 10: fonction calculant le déplacement $u(t)$ pour une position

```
#fonction renvoyant la matrice de deplacement (ligne=espace colone=temps)
def matrice_depl_exact(t,X,a,c,A,omega,T,L):
    U=np.zeros((len(X),len(t)))
    for j,x in enumerate(X):
        U[j,:]=deplacement_finite_domain(t,x,a,c,A,omega,T,L)
    return U
```

Figure 11: fonction calculant le déplacement $u(x,t)$

La fonction **deplacement_finite_domain** (t,x,a,c,A,ω,L)⁷ en figure 10 prend en argument :

t, la liste des instants pour lesquels on souhaite calculer les valeurs jusqu'à la durée total de simulations

x, la position à laquelle on souhaite calculer le déplacement

a, la position de la source,

omega, la pulsation de la source

T, la période d'émission du signal

L, la longueur du domaine.

Et renvoie une liste de valeurs de déplacements à la position x donné. Pour modéliser les réflexions ne pouvant pas faire une somme infinie, j'ai augmenté le nombre de réflexion n jusqu'à avoir une solution « satisfaisante », que l'on peut visualiser en dans la figure 13 à savoir ici $n=5$ (borne de la boucle for). La fonction **matrice_depl_exact**⁸ en figure 11 prenant les mêmes arguments que celle en figure 10 nonobstant qu'au lieu de prendre une position elle prend alors la liste des différentes positions où calculer les valeurs d'amplitude de

7:Présent dans le fichier : [solution_exacte_bis.py](#)

8:Présent dan le fichier : [solution_exacte_bis.py](#)

déplacement. Ainsi, elle renvoi la matrice **U** contenant **u(x,t)** pour une grille de temps et d'espace discrétisé :

$$(U)_{j,n} = U_j^i = u(j * dx, i * dt) \quad \text{pour un espace discrétisé de façon régulière}$$

```
# Paramètres physiques
x_0 = 1.0      # Point d'observation
instant=20     # instant d'observation
L = 10.0      # Longueur du domaine fini
a = L/2        # Position de la source
c = 1.0       # Vitesse de propagation de l'onde
A = 10.0      # Amplitude de l'onde
f=0.5         # frequence
omega = 2.0 * np.pi * f # Fréquence angulaire (ajustée)
T = 1/f       # Durée d'émission qui corespond à une periode
```

Figure 12: Paramètres de simulation

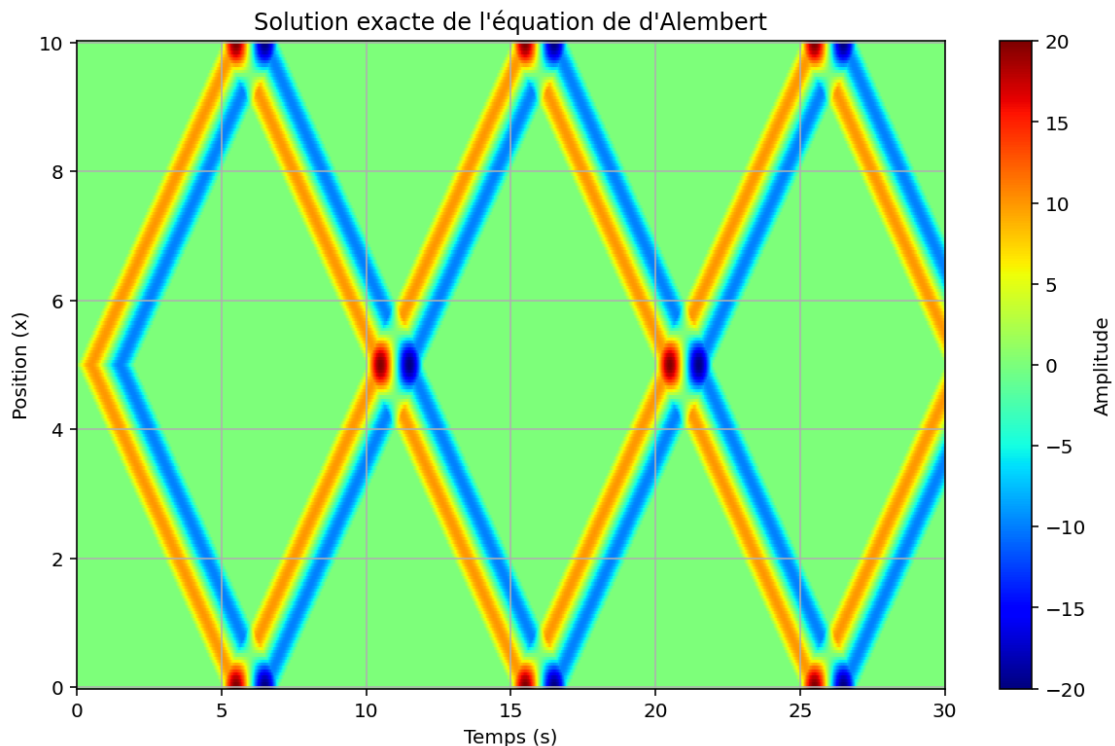


Figure 13: Diagramme de densité de la solution exacte $u(x,t)$

Avec les paramètre en figure 12 (tous exprimé dans les unités du système international) j'ai alors tracé un diagramme avec en abscisses le temps, en ordonnée la position et en couleur l'amplitude de déplacements. On constate que l'on a bien une onde qui initialement se propage dans les deux direction en $x=5$ puis une fois qu'elle atteint les bords, rebondit.

4.2.3 Implémentation méthode des différences finies (FDM)

Je me suis lancé dans l'implémentation d'un algorithme de différences finies⁹ maintenant que je dispose d'un ensemble de fonctions me permettant de visualiser la solution exacte et de calculer ses valeurs aux différents points. Il faut alors que je résolve ce problème grâce à la méthode des différences finies.

Pour résoudre ce problème on discrétise l'espace et le temps de tels sorte que :

$x \in [0, L]$ que l'on découpe en intervalle de longueur dx on alors $n_{\text{espace}} = L/dx$ donc $n_{\text{espace}} + 1$ points

$x_j = j \times dx$, avec $j \in \llbracket 0 ; n_{\text{espace}} \rrbracket$

$t \in [0, \text{Duree}]$ que l'on divise en intervalle de longueur dt ainsi, $n_{\text{temps}} = \text{Duree}/dt$ $n_{\text{temps}} + 1$ points en espace

$t_i = i \times dt$, avec $t \in \llbracket 0 ; n_{\text{temps}} \rrbracket$

On note $u(x_j, t_i) = u_j^i$

On cherche à résoudre numériquement l'équation de d'Alembert à une dimension d'espace.

Il faut alors trouver une approximation de la dérivée spatiale d'ordre 2 et de la dérivée temporelle d'ordre 2. On peut prendre alors prendre le schéma saute mouton (le plus connue) en temps et en espace (schéma d'ordre 2) que j'ai pris dans le livre [1chap 4.4].

$$\frac{\partial^2 u}{\partial x^2}(x_j, t_i) = \frac{1}{dx^2} (u_{j+1}^i - 2u_j^i + u_{j-1}^i) + O(dx^2)$$

$$\frac{\partial^2 u}{\partial t^2}(x_j, t_i) = \frac{1}{dt^2} (u_j^{i+1} - 2u_j^i + u_j^{i-1}) + O(dt^2)$$

En remplaçant dans l'équation de d'Alembert on a alors

$$\frac{u_j^{i+1} - 2u_j^i + u_j^{i-1}}{dt^2} - c^2 \frac{u_{j+1}^i - 2u_j^i + u_{j-1}^i}{dx^2} = 0$$

$$u_j^{i+1} = 2u_j^i - u_j^{i-1} + \frac{c^2 dt^2}{dx^2} (u_{j+1}^i - 2u_j^i + u_{j-1}^i)$$

$$u_j^{i+1} = 2u_j^i - u_j^{i-1} + \alpha^2 (u_{j+1}^i - 2u_j^i + u_{j-1}^i) \text{ pour } i \in \llbracket 2, n_{\text{temps}} - 1 \rrbracket, j \in \llbracket 1 ; n_{\text{espace}} - 1 \rrbracket$$

Traitements des conditions aux limites / initiales :

$$u(0, x_j) = u_j^0 = 0$$

$$u(0+dt, x_j) = u_j^1 = u_j^0 + dt \frac{\partial u}{\partial t}(0, x_j) = 0$$

$$u_0^i = 0$$

$$u_{n_{\text{espace}}}^i = 0$$

$$u_0^i = u_1^i$$

$$u_{n_{\text{espace}}}^i = u_{n_{\text{espace}}-1}^i$$

$$u(a, t_i) = u_{a/dx}^i = A \sin(\omega t_i), \text{ pour } t_i < T$$

9: Fichier: [FDM_bis](#)

Comme pour la solution exacte j'ai implémenter le schéma aux différences finies ci-contre avec les mêmes paramétrés physiques et j'ai alors tracé un diagramme de densité.

```
def matrice_depl_fdm(c,duree,L,a,f,Amplitude,dt,dx):
    alpha=c*dt/dx
    lvecteur=int(L/dx)
    cvecteur=int(duree/dt)
    a_=int(a/dx)+1
    gamma=alpha**2
    T=1/f
    ua=[condition(i*dt,Amplitude,f) for i in range(int(duree/dt)+1)]
    U=np.zeros((lvecteur+1,cvecteur+1))

    #Resolution du prblm
    for n in range(2,cvecteur+1):
        if n*dt<=T:
            for j in range(1,lvecteur):
                U[j,n]=2*U[j,n-1]-U[j,n-2] + gamma*(U[j+1,n-1]-2*U[j,n-1]+U[j-1,n-1])
                if j==a_:
                    U[a_,n]=ua[n]
            else:
                for j in range(1,lvecteur):
                    U[j,n]=2*U[j,n-1]-U[j,n-2] + gamma*(U[j+1,n-1]-2*U[j,n-1]+U[j-1,n-1])

    U[0, n] = U[1, n]
    U[lvecteur, n] = U[lvecteur-1, n]

    return U
```

Figure 14: Fonction FDM renvoyant la matrice de déplacement

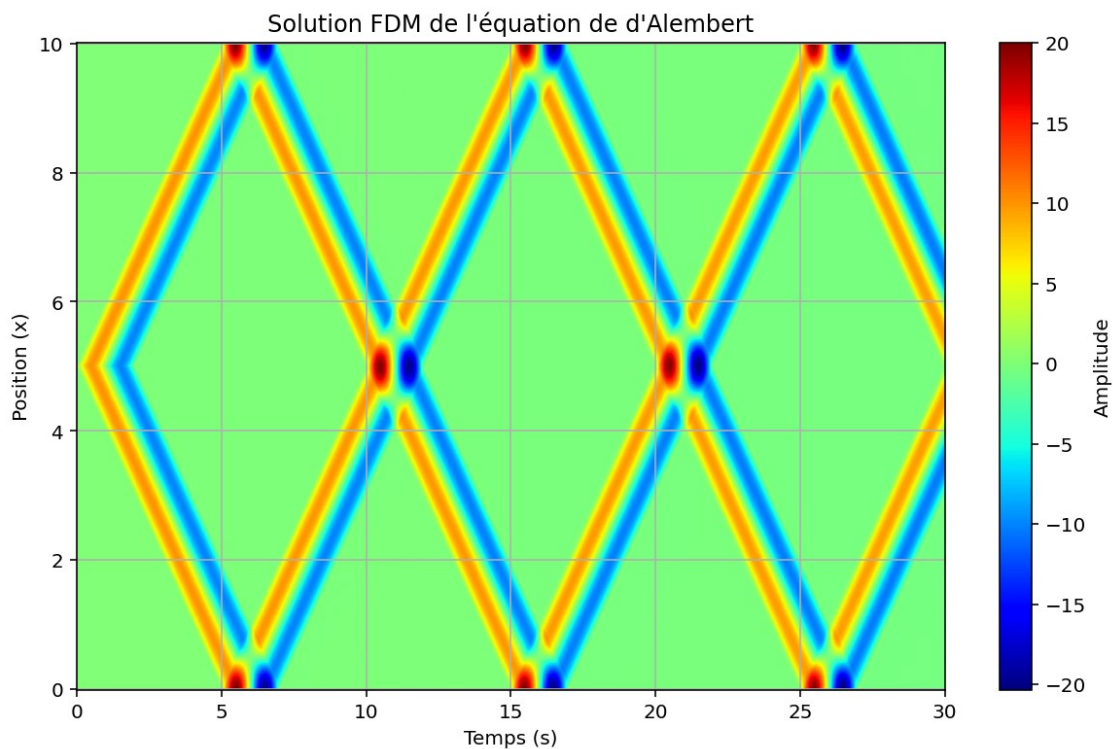


Figure 15: Diagramme de densité FDM ($dt=0.005$, $dx=0.01$)

On peut ainsi constater que tout comme dans la figure 15 on a bien une onde se propageant dans les deux sens et qui rebondit et s'inverse lorsqu'elle atteint les bords.

On va vérifier que ce schéma reste stable pour un α inférieur à 1. Un schéma est instable lorsque les erreurs numériques croient de manière exponentielle, cela se caractérise par la

croissance incontrôlée de la norme $\|u^n\|$ de la solution. Afin de pouvoir évaluer et comparée l'erreur d'un schéma pour un α donnée on va alors déterminer la grandeur :

$$\epsilon(t) = \frac{\int (u(x,t) - u_{exact}(x,t))^2 dx}{\int u_{exact}(x,t)^2 dx} \text{ avec } \epsilon(t) \text{ étant l'erreur normalisée sur tout le domaine à}$$

un instant donnée. J'ai choisi arbitrairement de regarder l'erreur en $T=20s$ car mon domaine faisant 10m et la source étant positionnée en $x=5m$, au bout de 20s l'onde à parcourus 2 fois tout mon domaine ainsi mesurer l'erreur à cette instant s'est retrouvé être assez pertinent. J'ai choisi de calculer les intégrales avec la méthode des trapèze qui est assez commode à implémenter. On cherche à chaque fois à calculer une intégrale entre 0 et L en $t=T=20S$:

$$\int_0^L u(x,T) dx \approx \frac{L}{n_{espace}} \left(\frac{u(0,T) + u(L,T)}{2} + \sum_{k=1}^{n-1} u\left(0 + k \frac{L}{n_{espace}}, T\right) \right)$$

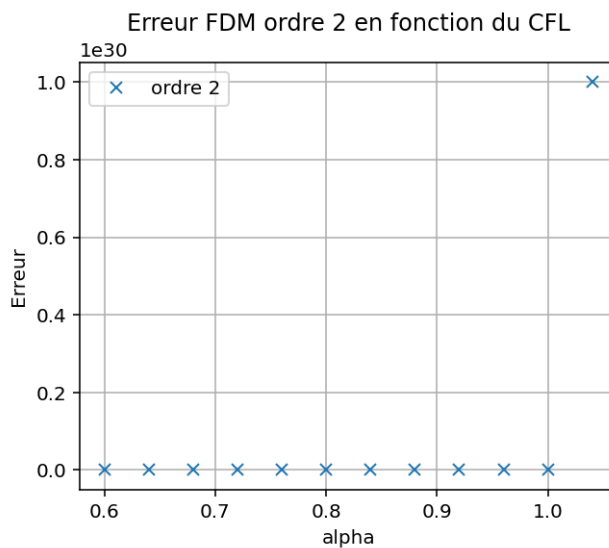


Figure 16: $\epsilon(t=20)$ en fonction de α ($dx=0.05$, dt entre 0.03 et 0.052)

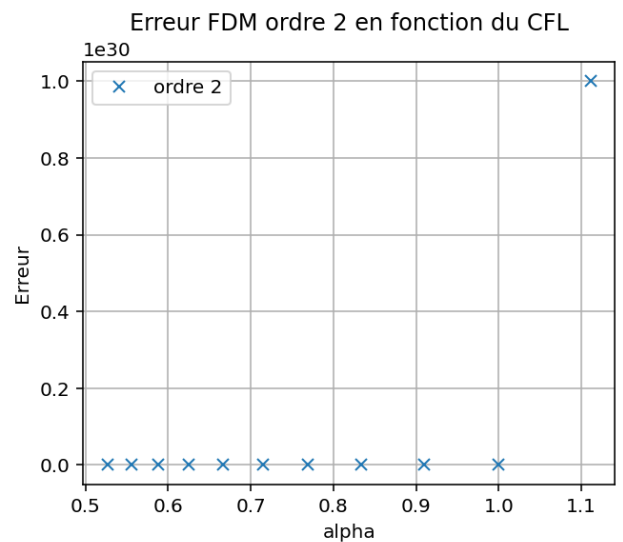


Figure 17: $\epsilon(t=20)$ en fonction de α (dx entre 0.019 et 0.008, $dt=0.01$)

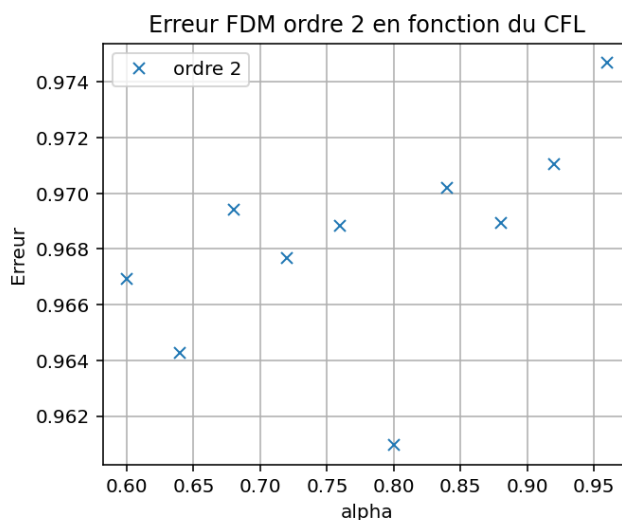


Figure 18: Figure 16 zoom

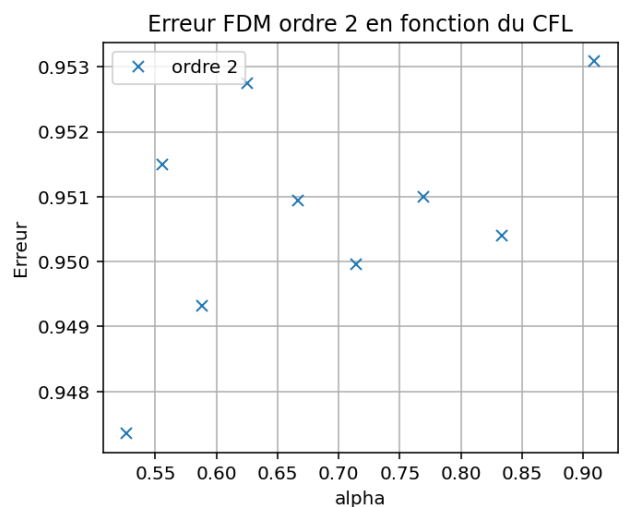


Figure 19: Figure 17 zoom

Les figure 16 et 17 montrent bien que la limite de stabilité est atteinte pour $\alpha_{\max}=1$. Les figures 18 et 19 montre que l'erreur croit lorsque α augmente. Par conséquent, ces figures me permettent bien de vérifier que la stabilité de mon schéma dépend bien de la valeur de α , ce qui confirme d'une part ma bonne implémentation de mon schéma aux différences finies : je pourrais alors réutiliser le schéma en temps pour la méthode des éléments spectraux. D'une autre part, ces résultats me confirment que ma méthode, pour étudier la stabilité d'une méthode de résolution est pertinente.

4.2.4 Implémentation méthode des éléments spectraux (SEM) en espace et différences finies en temps

La discrétisation en espace grâce à la méthode des éléments spectraux et une méthode que je n'ai pas étudié à Supméca mais le principe de cette méthode est le même que pour la méthode des éléments finis. J'ai suivi la méthode de discrétisation de Galerkin reprise dans le livre, on a alors la formulation du problème sous cette forme :

$$\rho \frac{\partial^2 u}{\partial t^2} - E \frac{\partial^2 u}{\partial x^2} = 0 \quad x \in [0, L] = \Omega, \quad t \in [0, \text{Durée}]$$

avec les conditions de Neuman et E le module de young et ρ la masse volumique

$$\left\{ \begin{array}{l} \text{Trouver } u(x, t) \in H_0^1(\Omega), t \in]0; T[, \text{ tels que} \\ \frac{d^2}{dt^2} \rho \int_{\Omega} u v \, dx + E \int_{\Omega} \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} \, dx = 0 \quad \forall v \in H_0^1(\Omega) \\ u(x, 0) = 0 \\ \frac{\partial u}{\partial t}(x, 0) = 0 \\ u(a, t) = H(T - t) A \sin(\omega t) \delta(x - a), \quad \text{avec } T = \frac{2\pi}{\omega} \end{array} \right.$$

$$\text{avec } H^1(\Omega) = \{ u \in L^2(\Omega) \text{ et } \frac{\partial u}{\partial x} \in L^2 \}$$

$$\text{et } H_0^1(\Omega) = \{ u \in L^2(\Omega) \text{ et } \frac{\partial u}{\partial t} = 0 \text{ sur } \partial \Omega \} \text{ avec } L^2(\Omega) \text{ l'espace des fonctions de carré intégrable}$$

on a alors :

$$M \ddot{U} + KU = 0$$

$$\text{avec } \left\{ \begin{array}{l} (M)_{i,j} = \sum_{k=0}^r \rho \omega_j \omega_i f(\epsilon_j) f(\epsilon_i) \det(J) \\ (K)_{i,j} = \sum_{k=0}^r \rho \omega_j \omega_i \frac{\partial f}{\partial x}(\epsilon_j) \frac{\partial f}{\partial x}(\epsilon_i) \det(J) \\ U^i = \text{matrice colonne de } u \text{ à l'instant } t_i \end{array} \right.$$

J la matrice Jacobienne ,

ω les poids associés aux points de Gauss – Lobatto ϵ et f les polynôme de Legendre d'ordre r

En appliquant un schéma aux différences finies en temps d'ordre 2 on a :

$$M \frac{U^{i+1} - 2U^i + U^{i-1}}{\Delta^2} + KU^i = 0$$

$$U^{i+1} = M^{-1}(-1) KU^i + \Delta t^2 (2U^i - U^{i-1}) = -NU^i + \Delta t^2 (2U^i - U^{i-1})$$

$$\left\{ \begin{array}{l} \text{Trouver } u(x, t) \in H_0^1(\Omega), t \in]0; T[, \text{ tels que} \\ \frac{d^2}{dt^2} \rho \int_{\Omega} u v \, dx + E \int_{\Omega} \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} \, dx = 0 \quad \forall v \in H_0^1(\Omega) \\ u(x, 0) = 0 \\ \frac{\partial u}{\partial t}(x, 0) = 0 \\ u(a, t) = H(T - t) A \sin(\omega t) \delta(x - a), \quad \text{avec } T = \frac{2\pi}{\omega} \end{array} \right.$$

avec $H^1(\Omega) = \{u \in L^2(\Omega) \text{ et } \frac{\partial u}{\partial x} \in L^2\}$

et $H_0^1(\Omega) = \{u \in L^2(\Omega) \text{ et } \frac{\partial u}{\partial t} = 0 \text{ sur } \partial\Omega\}$ avec $L^2(\Omega)$ l'espace des fonctions de carré intégrable

on a alors :

$$M \ddot{U} + KU = 0$$

$$\text{avec } \left\{ \begin{array}{l} (M)_{i,j} = \sum_{k=0}^r \rho \omega_j \omega_i f(\epsilon_j) f(\epsilon_i) \det(J) \\ (K)_{i,j} = \sum_{k=0}^r \rho \omega_j \omega_i \frac{\partial f}{\partial x}(\epsilon_j) \frac{\partial f}{\partial x}(\epsilon_i) \det(J) \\ U^i = \text{matrice colonne de } u \text{ à l'instant } t_i \end{array} \right.$$

J la matrice Jacobienne ,

ω les poids associés aux points de Gauss – Lobatto ϵ et f les polynôme de Legendre d'ordre r

En appliquant un schéma aux différences finies en temps d'ordre 2 on a :

$$M \frac{U^{i+1} - 2U^i + U^{i-1}}{\Delta^2} + KU^i = 0$$

$$U^{i+1} = M^{-1}(-1)KU^i + \Delta t^2(2U^i - U^{i-1}) = -NU^i + \Delta t^2(2U^i - U^{i-1})$$

La matrice M est diagonale, car les polynômes d'approximations utilisés sont orthogonaux, ce qui constitue un premier avantage significatif par rapport à la méthode des éléments finies où en général M n'est pas diagonal. Ainsi, on peut inverser très simplement M . Dans des problèmes comme en 3D sur des vastes domaines, comme pour les simulations autour du barrage, on peut atteindre le 10^{12} degrés de liberté. Dans la pratique, la matrice M n'est pas assemblée sous la forme d'une seule matrice, ce qui nous contraint à inverser la matrice M pour chaque tour de boucle en temps. Comme pour la partie précédente, je vais implémenter un algorithme de résolution avec une discrétisation en espace grâce à la méthode des éléments spectraux et une discrétisation en temps grâce à un schéma aux différences finies d'ordre 2. Je me propose alors de vérifier que $\alpha_{\max} = 0.40$ comme démontré dans le livre. Pour être cohérents à la formulation du problème initial, je vais non pas considérer ρ et E mais

plutôt $c = \sqrt{\frac{E}{\rho}}$ (dans mon algorithme, on peut tout de même les paramétrer si besoin, mais ils

sont fixés à 1). J'ai alors :

$$M \frac{U^{i+1} - 2U^i + U^{i-1}}{\Delta^2 t} + c^2 KU^i = 0 \quad (2.1)$$

$$U^{i+1} = -c^2 M^{-1} K + 2U^i - U^{i-1}$$

$$(U)_{L/dx \times r, i} = u(a, t_i) = A \sin(\omega t_i)$$

Avec toutes les conditions aux bords et initial que pour l'algorithme aux différences finies précédents. À noter que pour ma résolution, j'ai assemblé les matrices de rigidité et de masse, car je n'ai pas beaucoup d'éléments, ainsi je n'ai qu'à inverser une fois la matrice de masse. Les fonctions de mon code¹⁰ qui me donnent les valeurs des polynômes de Legendre, les poids et les points de Gauss Lobatto m'a été fourni par Monsieur De-Abreus Correa. J'ai alors tracer le diagramme de densité en figure 20:

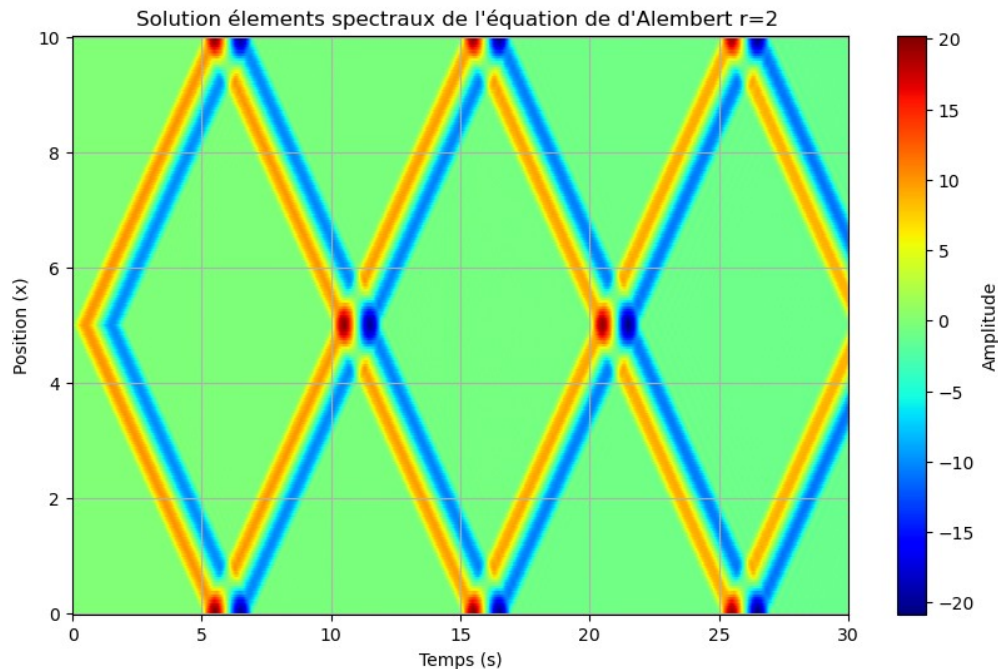


Figure 20: Diagramme de densité SEM $dt=0.01$ $dx=0.1$ $r=2$

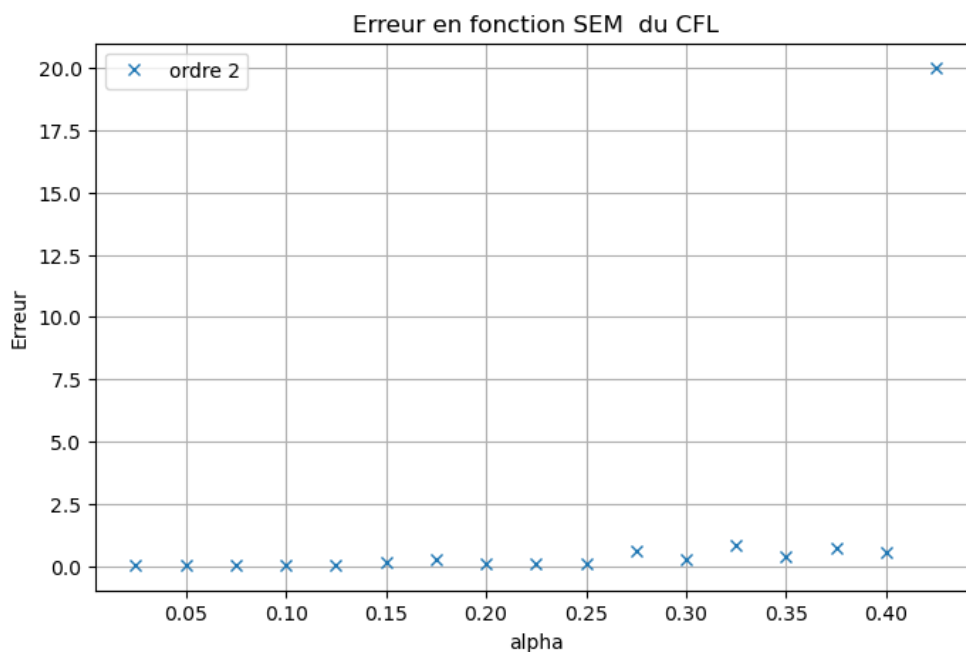


Figure 21: $\epsilon(T=20)$ SEM $r=2$ $dx=0.1$, dt entre 0.02 et 0.04

10: Fichier: [SEM_bis](#)

Comme attendu, on peut voir que la limite de stabilité α_{\max} semble être aux alentours de 0.4. À noter que bien que la méthode des éléments spectraux semble être beaucoup plus restrictif en termes de stabilité car par rapport aux différences finies on a un $\alpha_{\max}=1$, cependant prendre en compte le nombre de points et plus pertinents car pour un élément de longueur dx on a $r-1$ points à l'intérieur de un élément. Ainsi, on a un nombre de points qui vaut le nombre d'éléments $\times r+1$, si on considère α comme le rapport entre le nombre de point en temps et en espace multiplier par la célérité on a en réalité $\alpha_{\max}=0.4 \times r=0.8$ pour comparer à nombre d'éléments égaux.

4.2.5 Gains apportés par avec une approximations en temps d'ordre supérieurs : Méthode d'approche par équation modifiés.

Maintenant, que j'ai un algorithme d'éléments spectraux en espace correct et que j'ai une méthode fiable pour étudier la stabilité d'un algorithme de résolution, j'ai pu me lancer dans l'étude de stabilité pour des approximations en temps de plus hauts degrés. Pour obtenir des approximations de degré supérieur, une méthode intéressante qui figure dans le livre de Cohen[1.chap 11.3] est la méthode d'approche par équation modifiée qui consistent à utiliser l'équation d'onde pour obtenir des degrés d'approximation en temps supérieurs. On construit l'approche comme cela :

Le développement de Taylor du schéma saute – mouton est \leq suivant :

$$\frac{u^{i+1} - 2u^i + u^{i-1}}{\Delta t^2} = \left(\frac{\partial^2 u}{\partial t^2}\right)^i + \frac{\Delta t^4}{12} \left(\frac{\partial^4 u}{\partial t^4}\right)^i + O(\Delta t^4)$$

En utilisant l'équation d'onde (1)

$$\frac{\partial^4 u}{\partial t^4} = \frac{\partial^2}{\partial t^2} \left(\frac{\partial^2 u}{\partial t^2}\right) = \frac{\partial^2}{\partial t^2} (c^2 \frac{\partial^2 u}{\partial x^2}) = c^2 \frac{\partial^2}{\partial x^2} \left(\frac{\partial^2 u}{\partial t^2}\right) = c^4 \frac{\partial^4 u}{\partial x^4}$$

Donc on a,

$$\frac{u^{i+1} - 2u^i + u^{i-1}}{\Delta t^2} - \frac{c^4 \Delta t^2}{12} \frac{\partial^4 u}{\partial x^4} - c^2 \frac{\partial^2 u}{\partial x^2} = 0$$

En utilisant un approximation par éléments spectraux en espace et en utilisant la même technique on a :

$$\frac{U^{i+1} - 2U^i + U^{i-1}}{\Delta t^2} + c^2 N U^i = 0$$

avec $N = M^{-1} \nabla^2 M$ on obtient alors

$$\frac{U^{i+1} - 2U^i + U^{i-1}}{\Delta t^2} - c^4 \frac{\Delta t^2}{12} N^2 U^i + c^2 N U^i = 0$$

On a alors une approximation d'ordre 4 de la dérivée seconde. En répétant la méthode on a alors :

$$\frac{U^{i+1} - 2U^i + U^{i-1}}{\Delta t^2} + c^2 N U^i - 2 \sum_{j=2}^k (-1)^j c^{2j} \frac{\Delta t^{2j-2}}{(2j)!} N^j U^i = 0$$

$$U^{i+1} = 2U^i - U^{i-1} + 2 \sum_{j=2}^k (-1)^j c^{2j} \frac{\Delta t^{2j}}{(2j)!} N^j U^i$$

Grâce à cette méthode on obtient alors des approximations en temps d'ordre $2k$. Cette méthode permet d'avoir un ordre plus élevé d'approximation en temps en utilisant des termes correctifs faisant intervenir des puissances de N et donc de M . L'intérêt d'avoir un

ordre plus élevé en temps c'est de éventuellement gagner en stabilité pour pouvoir augmenter le pas de temps dans le cadre de notre sujet initiale (la simulation de tremblement de terre au sein d'un barrage et sa zone avoisinante) . En effet, dans la partie 4.1 ,on a vu que la présence d'un petit élément comme le barrage provoque la multiplication du nombre d'éléments nécessaire si il n'était pas présent Ainsi une manière de réduire le nombre de calculs, c'est d'augmenter le pas de temps, on doit néanmoins veiller à rester en dessous de α_{max} . Le fait d'avoir plusieurs produit entre U et une puissance de N peut être un problème dans notre soucis d'économiser des ressources de calcul, cependant les faits de ne jamais assembler K permet d'effectuer avec une meilleure complexité le produit matrice-vecteur (Je parle ici de l'algorithme de résolution SEM3D du LMA, dans mes algorithmes j'assemble toutes les matrices).

Ainsi, on cherche à savoir jusqu'à quel ordre d'approximation il est intéressant de gagner en stabilité.

Comme pour les parties 4.2.3 et 4.2.4 je vais répondre au même problème avec les mêmes paramètres physiques pour étudier la stabilité des différents ordres d'approximation en temps.

Pour cela, j'ai écrit une fonction **approcheModifié**¹¹ qui renvoie la matrice U contenant les déplacements pour un ordre d'approximation en temps égale à $2k$, k étant un paramètres d'entrée de cette fonction, j'ai alors tous les outils nécessaires pour mener à bien les différentes études. J'ai effectué l'étude pour k les ordre : 2,4,6,8,10,12. Comme pour les précédents cas, mon algorithme calcul $\epsilon(t=20)$ pour un dx constant et des dt et donc un α qui augmentent, le calcul s'arrête lorsque un seuil a été dépasser, je l'ai fixé tout à 20 et le dernier point prend la valeur de ce seuil.

11: fonction ApprocheModifié : [mon github\stage\SEM_bis.py](#)

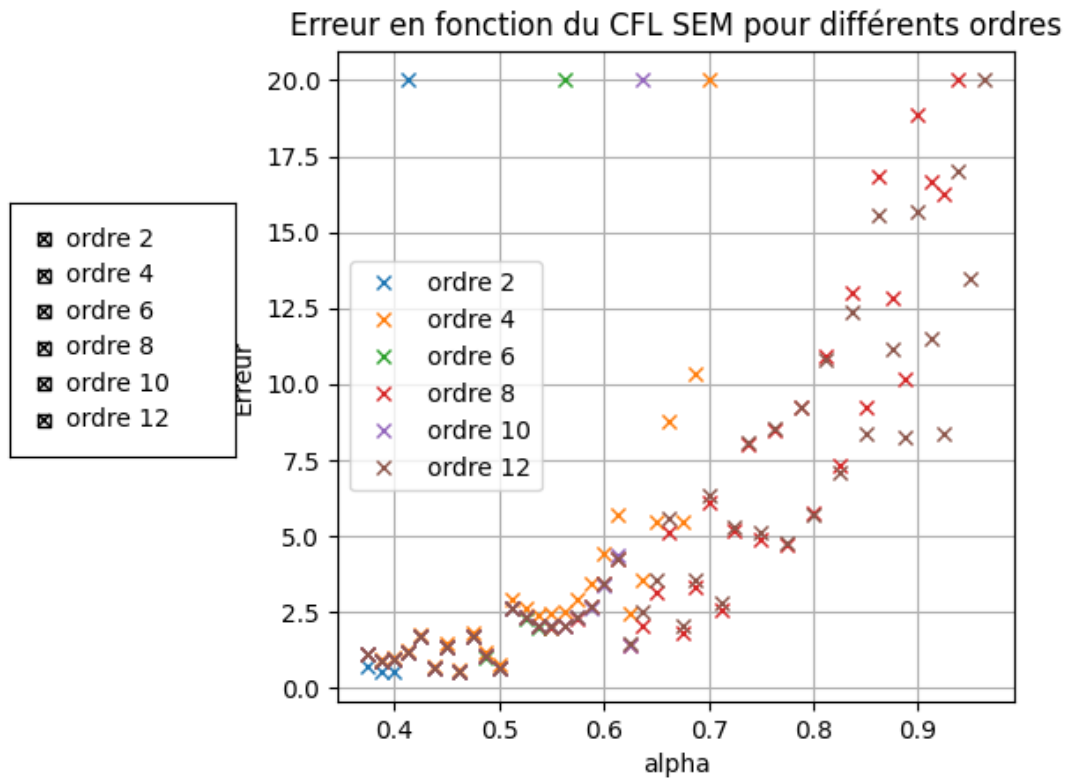


Figure 22: $\epsilon(t=20)$ SEM à différent ordre d'approximations en temps

Sur la figure 22 on s'aperçoit que pour des ordres supérieurs l'erreur croit moins vite que pour des ordres inférieurs.

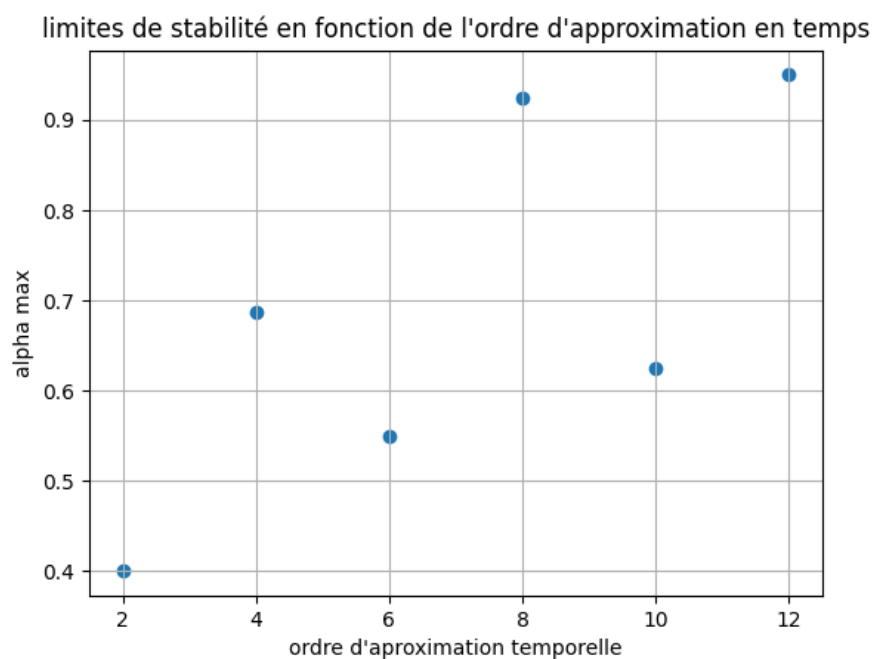


Figure 23: α max pour différents ordre d'approximations en temps

On constate bien, sur la figure 23, que pour des ordres supérieurs à 2, on a des α_{\max} supérieurs. On s'aperçoit cependant que pour les ordres 6 et 10, leur limite de stabilité est inférieurs à la limite de l'ordre précédent. On peut conjecturer que pour des ordres d'approximations 2^k avec k pairs, on a effectivement un gain par rapport aux précédents. On passe de 0.4 à 0.7 entre l'ordre 2 et 4 puis soit un gain de 0.3 en α_{\max} . Puis on a un gain de 0.2 entre l'ordre 4 et 8 et enfin on a un gain de même 0.05 entre l'ordre 8 et 12 : ainsi plus on avance plus le gain que l'on fait sur l'ordre précédent sera minime.

Néanmoins, sur la figure 22, on remarque que l'erreur augmente de moins en moins vite pour les ordres avec k pairs, mais il reste élevé s'approchant du seuil que j'ai mis initialement à 20 car lors de mes premiers tests en 4.2.2 et en 4.2.3 on avait vite des valeurs supérieures à 10^{100} parfois dépassant les limites de ce que mon ordinateur peut supporter. Ici l'erreur augmente progressivement, mais elle reste tout de même relativement élevée. Par exemple pour l'ordre 12 avec un α entre 0.6 et 0.9 $\varepsilon(t) > 2$.

5. Conclusion

Ce stage m'a permis d'acquérir des compétences en simulation numériques sur différentes méthodes, et m'a permis d'apprendre à utiliser plusieurs logiciels comme GMSH et Paraview qui me seront certainement utiles pendant ma future carrière. Le maillage de barrage que j'ai produit ouvre la voie à d'autres personnes pour la mise en place de simulation précise pour d'autres barrages. L'étude de stabilité a mis en évidence les avantages et les limitations concernant l'utilisation d'une méthode d'approximation en temps de hauts degrés couplée à une approximation en espace grâce à la méthode des éléments spectraux. À terme, ces résultats pourraient être intégrés dans des logiciels de simulation comme SEM 3D afin de pouvoir mener des études sismologiques.

6 Tableaux des figures

Figure 1: Vue de la zone avoisinant le barrage de tsankov (Google Maps).....	6
Figure 2: Vue du Barrage de Tskankov.....	6
Figure 3: Maillage d'un cube et son code (maillage d'entraînement).....	10
Figure 4: Modèle du domaine du barrage simple (avant extrusion).....	11
Figure 5: Modèle du domaine final.....	12
Figure 6: Vue du maillage du barrages.....	13
Figure 7: Fichier fournis par BOUNDCLOUD.....	14
Figure 8: Maillage après application de la topographie avoisinante.....	15
Figure 9: Vue du déterminant Jacobien des éléments hexaédrique du maillage sur Paraview.	15
Figure 10: fonction calculant le déplacement $u(t)$ pour une position.....	19
Figure 11: fonction calculant le déplacement $u(x,t)$	19
Figure 12: Paramètres de simulation.....	20
Figure 13: Diagramme de densité de la solution exacte $u(x,t)$	20
Figure 14: Fonction FDM renvoyant la matrice de déplacement.....	22
Figure 15: Diagramme de densité FDM ($dt=0.005$, $dx=0.01$).....	22
Figure 16: $\varepsilon(t=20)$ en fonction de α ($dx=0.05$, dt entre 0.03 et 0.052).....	23
Figure 17: $\varepsilon(t=20)$ en fonction de α (dx entre 0.019 et 0.008, $dt=0.01$).....	23
Figure 18: Figure 16 zoom.....	23
Figure 19: Figure 17 zoom.....	23
Figure 20: Diagramme de densité SEM $dt=0.01$ $dx=0.1$ $r=2$	26
Figure 21: $\varepsilon(T=20)$ SEM $r=2$ $dx=0.1$, dt entre 0.02 et 0.04.....	26
Figure 22: $\varepsilon(t=20)$ SEM à différent ordre d'approximations en temps.....	29
Figure 23: α max pour différents ordre d'approximations en temps.....	29

7 Références bibliographiques

[1] Garry COHEN., 2002. *Higher-Order Numerical Methods for Transient Wave Equations*,

[2] Karniadakis, G. E., & Sherwin, S. J. (2005). *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press.



Institut supérieur de mécanique de Paris
3 rue Fernand Hainaut 93407 Saint-Ouen Cedex
tél. 01 49 45 29 00 / www.isae-supmeca.fr

Ministère de l'Enseignement supérieur
et de la Recherche

