

UFR de mathématique et d'informatique

Université de Strasbourg

MASTER 2 DE CALCUL SCIENTIFIQUE ET
MATHÉMATIQUES DE L'INFORMATION

Rapport de stage présenté par

Sacha ALIDADI HERAN
sacha.alidadi-heran@etu.unistra.fr

ÉTUDES DE LA PROPAGATION DES ONDES MÉCANIQUES SUR FEEL++

27 août 2024

Stage encadré par

Thomas LE GALL
t.legall@avnir-energy.fr

Au sein de
AVNIR ENERGY



Table des matières

I	Introduction	5
0.1	Présentation de l'entreprise	6
0.2	Contexte du projet SYNUDIN	6
0.3	Limites des méthodes actuelles	8
0.4	Objectifs du stage	8
II	Contributions	9
1	Prérequis	10
1.1	Méthodes des éléments spectraux	10
1.2	Discretisation en temps de Newmark	11
1.3	Équation d'élasticité	11
1.4	Méthodes de résolution d'équations différentielles ordinaires	13
1.5	Préconditionneur	16
1.6	Accélération	17
1.7	Feel++	17
2	Résultats obtenus	18
2.1	Méthode de comparaison des résultats	18
2.2	Efficacité de la méthode	19
2.2.1	Cas 2D	19
2.2.2	Cas 3D	22
2.3	Comparaison des résultats avec les méthodes précédentes	29
2.4	Comparaison des temps de calcul	30
2.5	Cas pratique	32
2.5.1	Retournement temporel en 2D	32
2.5.2	Déformation d'une plaque 3D	38
III	Conclusion	42
2.6	Travail à venir	43
2.7	Bilan	43
2.8	Remerciements	43
A	Calcul de la solution analytique	45
A.1	Plaque	45
A.1.1	Équation mécanique de la plaque	45
A.1.2	Schema modal	46
A.1.3	Résolution de l'équation forcée	48
A.1.4	estimation de l'erreur	49

Table des figures

1	Phase d'apprentissage du retournement temporel	6
2	Phase de focalisation du retournement temporel	7
3	Détection d'un défaut dans la structure	8
1.1	Comparaison de la représentation d'un trou dans une plaque par la méthode des éléments finis et des éléments spectraux	10
1.2	Méthode de Runge-Kutta d'ordre 4	15
2.1	Plaque carrée encastree	19
2.2	Effort appliqué sur la plaque	19
2.3	Video de la solution. Il faut cliquer sur l'image pour lancer la vidéo	20
2.4	Emplacement des capteurs	21
2.5	Erreur L^2 en fonction de la taille des éléments	22
2.6	Plaque rectangulaire, les bords rouges sont simplement supportés	23
2.7	Plaque cylindrique, les bords rouges sont simplement supportés	23
2.8	Solution analytique sur les 3 capteurs	26
2.9	Emplacement des capteurs et du choc	27
2.10	Solution numérique sur la plaque	27
2.11	Signaux obtenus aux niveaux des 3 capteurs lors de la résolution numérique	28
2.12	Comparaison des erreurs entre la méthode par élément finis et élément spectraux	29
2.13	Évolution de l'accélération en fonction du nombre de coeurs	31
2.14	Illustration d'un maillage partagé entre 2 coeurs. Les éléments verts sont traités par le premier coeur, les éléments rouges par le second.	32
2.15	Maillage avec des éléments plus petits partagé entre 2 coeurs. Il y a 9 noeuds partagés.	32
2.16	Phase d'apprentissage sur la plaque 2D	33
2.17	Emplacement des capteurs	33
2.18	Signaux obtenus par la méthode des éléments finis (En orange) et des éléments spectraux (en bleu)	34
2.19	Phase de focalisation sur la plaque 2D	35
2.20	Signal obtenu lors de la phase de focalisation	35
2.21	Plaque 2D avec un défaut (Trou en cercle de centre (-0.5;0) et de rayon 0.25)	36
2.22	Phase de focalisation sur la plaque 2D	37
2.23	Signal obtenu lors de la phase de focalisation sur la structure endommagé (en bleu) et la structure saine (en orange)	37
2.24	Maillage de la plaque trouée sur Gmsh	38
2.25	Maillage contenant des défauts. Les faces grises montrent des faces qui se coupent entre elles.	39
2.26	Maillage avec les défauts supprimés.	39
2.27	Emplacement des capteurs et des points de chocs	40
2.28	Phase d'apprentissage sur la plaque trouée	40
2.29	Phase d'apprentissage sur la plaque cylindrique	41

Liste des tableaux

1.1	Condition de stabilité de la méthode de Newmark	11
2.1	Paramètres utilisés pour obtenir le résultat de 2.3	21
2.2	Erreur entre la solution analytique et la solution numérique	28
2.3	Paramètres utilisés pour obtenir le résultat de 2.10	29
2.4	Erreur commise par les solutions sur les 3 capteurs par rapport à la solution de référence . .	30
2.5	Comparaison du temps de calcul entre les deux méthodes à iso taille d'élément	30
2.6	Comparaison du temps de calcul entre les deux méthodes à iso degré de liberté	30
2.7	Comparaison du temps de calcul entre les deux méthodes à iso erreur	30
2.8	Comparaison des temps de calculs en utilisant différents nombres de coeurs	31
2.9	Comparaison des temps de calcul en fonction du nombre de coeurs	31
2.10	Paramètres utilisés pour obtenir le résultat de 2.28	41
2.11	Paramètres utilisés pour obtenir le résultat de 2.29	41

Première partie

Introduction

0.1 Présentation de l'entreprise

Avnir Energy est une entreprise créée en 2015 spécialisée dans les problèmes d'ingénierie mécaniques. Elle a comme clientèle les grands comptes de l'industrie énergétique (EDF, CEA) et accompagne les industriels suivants :

- Énergie nucléaire
- Pétrolière
- Gazière
- Hydroélectrique
- Thermique
- Renouvelable

Composé de 25 employés, Avnir Energy est constitué de 3 pôles :

- Pôle R&D
- Pôle Opérationnel
- Pôle Commercial

Le projet lié au stage se situe dans le pôle R&D. De plus, l'entreprise compte plusieurs partenaires dont l'INSA de Lyon, l'École Centrale de Lyon, Sonorhc, etc.

0.2 Contexte du projet SYNUDIN

Le projet SYNUDIN est un projet de R&D en collaboration avec l'entreprise Sonorhc. L'objectif de ce projet est de développer une méthode de surveillance des cuves de réacteurs nucléaires grâce à une technique appelée *retournement temporel*.

Le retournement temporel est composé de deux phases, premièrement une onde est envoyée à partir d'un point source dans la structure à surveiller, c'est la phase d'apprentissage. La figure 1 illustre cette phase.

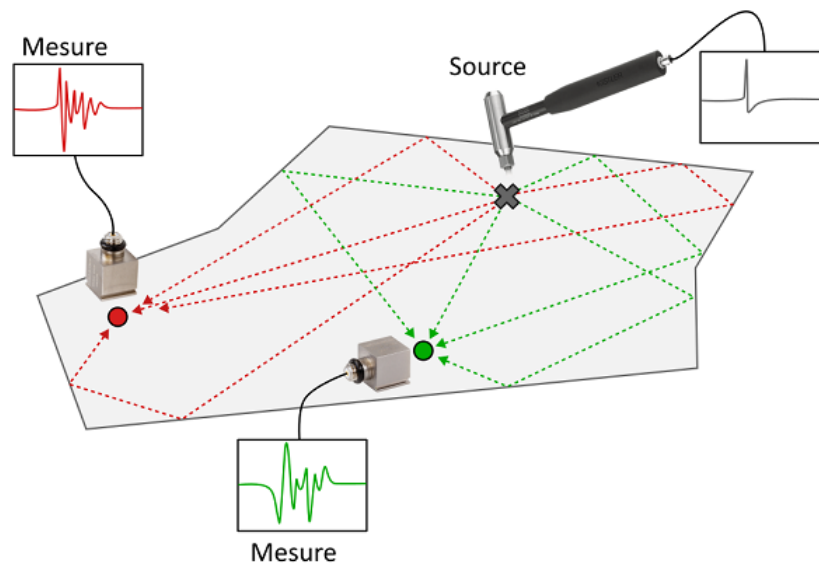


FIGURE 1 – Phase d'apprentissage du retournement temporel

Durant cette phase, des accéléromètres sont placés à différents points de la structure pour mesurer le déplacement au cours du temps. Les points vert et rouge de la figure 1 correspondent à ces accéléromètres.

Ensuite, les signaux enregistrés sont inversés dans le temps et émis dans la structure aux emplacements des accéléromètres. L'onde se propage alors dans la structure et se focalise sur le point d'origine de l'onde. C'est la phase de focalisation, comme expliqué dans la figure 2.

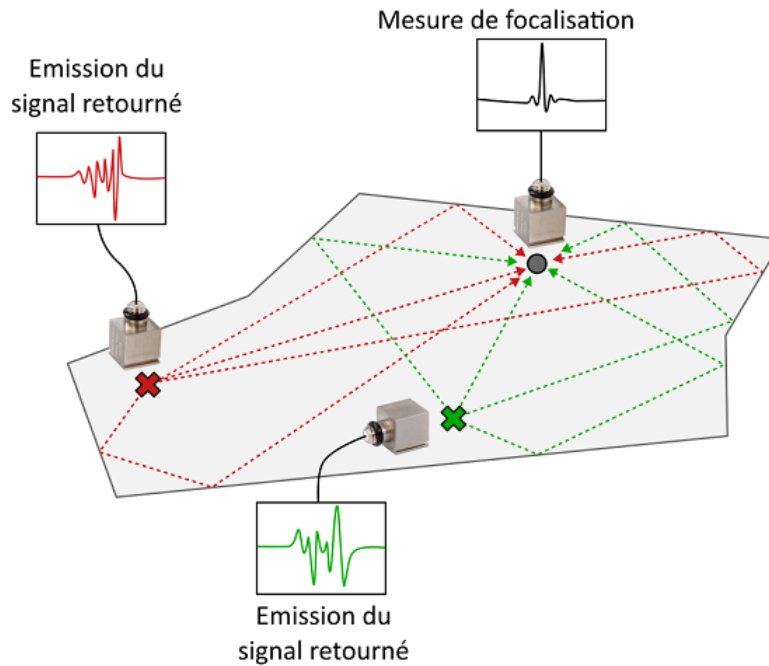


FIGURE 2 – Phase de focalisation du retournement temporel

Sur la structure sans défaut de la figure 2, la focalisation permet de créer un pic. Tandis que si un défaut apparaît entre la phase d'apprentissage et de focalisation, ce pic n'apparaîtra pas ou aura une amplitude plus faible, ce qui donne un signal plus diffus. La figure 3 illustre ce cas.

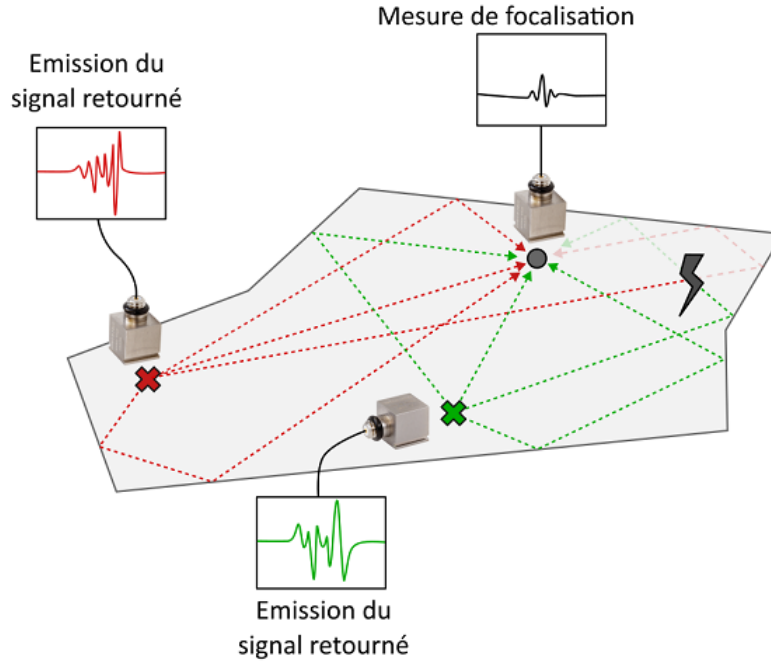


FIGURE 3 – Détection d'un défaut dans la structure

Dans le cas d'une cuve de réacteur nucléaire, le retournement temporel est difficile voire impossible à appliquer en pratique. Parce que les cuves contiennent un liquide à haute température ainsi que des barres de combustibles. C'est pourquoi le projet SYNUDIN vise à développer une méthode numérique pour effectuer la phase d'apprentissage sur un jumeau numérique de la cuve.

0.3 Limites des méthodes actuelles

Faire ces simulations implique de résoudre des équations aux dérivées partielles. La méthode numérique de référence est la méthode des éléments finis (FEM). Cependant, cette méthode, bien que rapide, est très imprécise. En effet, les éléments finis demandent d'approximer la géométrie de la structure par des éléments simples (triangles, quadrangles), ce qui nécessite un maillage très fin pour obtenir une solution précise. L'objectif de ce projet est donc de développer une méthode de résolution numérique qui soit plus précise que la méthode des éléments finis tout en restant peu coûteuse en temps de calcul. Pour cela, la méthode isogéométrique (IGA) a été implémentée chez Avnir Energy. Cette méthode permet de résoudre des équations aux dérivées partielles en utilisant des courbes et des surfaces de Bézier pour décrire la géométrie de la structure. En outre, cette méthode utilise des fonctions d'interpolation NURBS (Non-Uniform Rational Basis Splines) comme fonction d'interpolation, là où la méthode par éléments finis utilise des fonctions d'interpolation de Lagrange. De plus, une autre méthode appelée méthode par éléments spectraux doit être implémentée pour tester sa précision. Cette méthode sera décrite dans la section 1.1.

0.4 Objectifs du stage

Le stage a pour objectif de développer la méthode des éléments spectraux grâce à la librairie Feel++ développé par le CEMOSIS (Centre de MODélisation et de SIMulation à Strasbourg). Une fois la méthode implémentée, il faudra lancer des calculs et comparer les résultats obtenus avec ceux des méthodes précédemment mentionnées (FEM,IGA) ainsi que des solutions analytiques si elles existent.

Deuxième partie

Contributions

Chapitre 1

Prérequis

1.1 Méthodes des éléments spectraux

Comme mentionné dans l'introduction, la méthode des éléments spectraux est la méthode de résolution qui sera utilisé durant ce stage. Il s'agit d'une variante de la méthode des éléments finis (FEM) qui utilise comme fonction de base des polynômes de degré élevé (3,4 voire plus), là où la FEM utilise généralement des polynômes de degré 1 ou au plus 2. L'intérêt de cette méthode est le gain de précision pour la solution, car les polynômes de degré élevé permettent de donner une déformation des éléments plus fidèle à la réalité. Également, cette méthode offre un gain de précision dans la représentation de la géométrie de la structure. Contrairement à la méthode par élément finis, qui nécessite un maillage très fin pour obtenir une géométrie précise, la méthode des éléments spectraux permet de représenter la géométrie de la structure avec moins d'éléments. La figure 1.1 compare un trou dans une plaque représenté par des éléments finis et par des éléments spectraux.

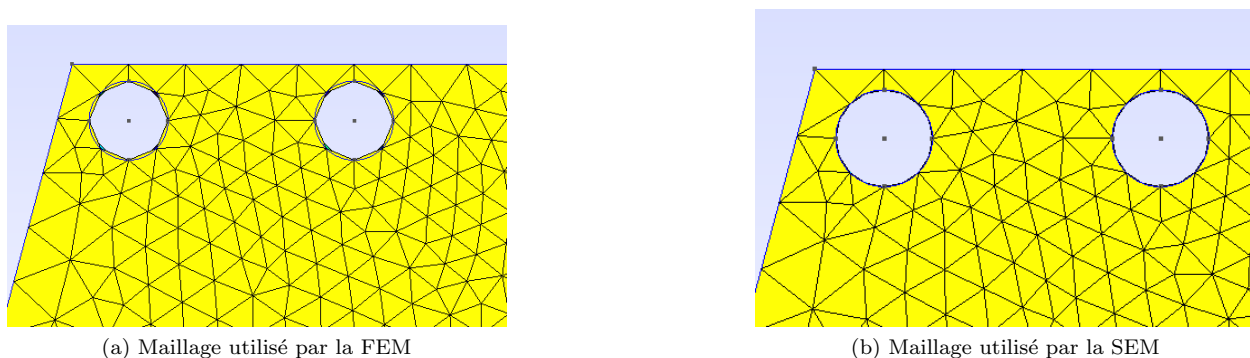


FIGURE 1.1 – Comparaison de la représentation d'un trou dans une plaque par la méthode des éléments finis et des éléments spectraux

Cependant, utiliser des polynômes de degré élevé implique un coût de calcul plus important, car lors de l'assemblage de la matrice de masse et de raideur, ces dernières seront plus denses en termes de coefficients non nuls. En conséquence, le temps de calcul pour inverser la matrice sera long et le stockage en mémoire sera plus important. Pour résoudre ce problème, il est possible de passer par un maillage quadrangulaire (en 2D) ou hexaédrique (en 3D) et d'utiliser des points de Gauss-Lobatto-Legendre pour intégrer les termes de la matrice. Ce faisant, les coefficients non nuls de la matrice seront réduits. Il est également possible de rester sur un maillage triangulaire ou tétraédrique et d'utiliser des points de Dubiner pour intégrer les termes de la matrice. Cette méthode conserve les mêmes propriétés que la méthode précédente.

1.2 Discrétisation en temps de Newmark

Les équations à résoudre dans le cadre de la propagation des ondes sont des équations dépendantes du temps. Ce qui implique de discrétiser ces équations dans le temps et d'approximer les dérivées en temps. Une approximation d'ordre 2 d'une dérivée est donnée par :

$$\frac{\partial^2 u}{\partial t^2} \approx \frac{u_{n+1} - 2u_n + u_{n-1}}{\Delta t^2}$$

Où u_n est la solution à l'instant n et Δt est le pas de temps. Cependant, il faut assurer la stabilité de la méthode. C'est pour ça que la méthode de Newmark est utilisée. Il s'agit de résoudre le système d'équation ci-dessous pour u_{n+1} et $\partial_t u_{n+1}$

$$\begin{cases} u_{n+1} = u_n + \Delta t \partial_t u_n + \frac{\Delta t^2}{2} ((1 - 2\beta) \partial_{t^2} u_n + 2\beta \partial_{t^2} u_{n+1}) \\ \partial_t u_{n+1} = \partial_t u_n + \Delta t ((1 - \gamma) \partial_{t^2} u_n + \gamma \partial_{t^2} u_{n+1}) \end{cases}$$

Où β et γ sont des paramètres de la méthode de Newmark. Le choix de ces paramètres est important pour assurer la stabilité du schéma. Le tableau ?? vient de [5] et donne la stabilité du schéma en fonction de β et γ :

Coefficients	Stabilité
$\gamma \leq \frac{1}{2}$	Instable
$\gamma \geq \frac{1}{2}$ et $2\beta \leq \gamma$	Conditionnellement stable
$\frac{1}{2} \leq \gamma \leq 2\beta$	Inconditionnellement stable

TABLE 1.1 – Condition de stabilité de la méthode de Newmark

De plus, l'article [4] garantit une stabilité du schéma pour $\beta = \frac{1}{4}$ et $\gamma = \frac{1}{2}$, c'est pourquoi ces valeurs ont été choisis tout au long du stage.

1.3 Équation d'élasticité

Le déplacement d'une structure soumise à des forces est régi par l'équation d'élasticité. Il existe deux types d'équation, l'équation d'élasticité statique et l'équation d'élasticité dynamique. La première équation est donnée par :

$$\begin{cases} -\nabla \cdot \sigma(u) = f & \text{dans } \Omega \end{cases} \quad (1.1)$$

Où f désigne la force appliquée sur la structure, u est le déplacement de la structure et σ le tenseur des contraintes. Par la suite, σ sera donné par :

$$\sigma(u) = \lambda \nabla \cdot u \mathbf{I} + 2\mu \epsilon(u)$$

Ici λ et μ sont les coefficients de Lamé, qui dépendent des propriétés du matériau. \mathbf{I} est la matrice identité. Enfin $\epsilon(u)$ est le tenseur des déformations, donné par :

$$\epsilon(u) = \frac{1}{2} (\nabla u + \nabla u^T)$$

L'équation 1.1 ne dépend pas du temps, c'est pourquoi elle est appelée équation d'élasticité statique. L'équation d'élasticité dynamique, qui dépend du temps, est donnée par :

$$\begin{cases} \rho \frac{\partial^2 u}{\partial t^2} - \nabla \cdot \sigma(u) = f & \text{dans } \Omega \\ \sigma = \lambda (\nabla \cdot u) \mathbf{I} + \mu (\nabla u + \nabla u^T) & \\ \sigma \cdot n = g & \text{sur } \Gamma \\ u(x, 0) = 0 & \text{dans } \Omega \\ \frac{\partial u}{\partial t}(x, 0) = 0 & \text{dans } \Omega \end{cases} \quad (1.2)$$

L'équation 1.2 décrit le déplacement de la plaque. Où ρ est la densité de l'acier, u , σ , λ et μ désignent les mêmes variables. Comme le domaine est homogène, les coefficients de Lamé sont constants. De plus, f désigne la force externe appliqué sur la plaque et g dans tous les cas qui seront abordés est la fonction nulle. Ω désigne le milieu dans lequel est étudié la propagation des ondes et Γ son bord.

Pour résoudre cette équation, il faut d'abord écrire sa formulation variationnelle. Soit v une fonction test, l'équation variationnelle est donnée par :

$$\int_{\Omega} \rho \frac{\partial^2 u}{\partial t^2} \cdot v \, dx - \int_{\Omega} \sigma : \varepsilon(v) \, dx = \int_{\Omega} f \cdot v \, dx + \int_{\Gamma} g \cdot v \, ds \quad (1.3)$$

Où $\varepsilon(v) = \frac{1}{2} (\nabla v + \nabla v^T)$ est le tenseur des déformations. Où $\varepsilon(v) = \frac{1}{2} (\nabla v + \nabla v^T)$ est le tenseur des déformations. Et $:$ désigne le produit matriciel scalaire défini par :

$$A : B = \sum_{i,j} A_{ij} B_{ij}$$

Ensuite, il faut discrétiser l'équation 1.3 dans le temps. Pour cela, Feel++ propose un schéma d'ordre 2 en différences finies. Le schéma est donné par :

$$\frac{\partial^2 u}{\partial t^2} \approx \frac{u_{n+1} - 2u_n + u_{n-1}}{\Delta t^2}$$

Où Δt est le pas de temps, u_n et u_{n-1} sont les solutions aux instants t_n et t_{n-1} respectivement. u_0 est déjà connu, cependant u_1 ne l'est pas. Pour le déterminer, il faut utiliser l'approximation de Taylor d'ordre 2 :

$$u_1 = u_0 + \Delta t \frac{\partial u_0}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 u_0}{\partial t^2} + \mathcal{O}(\Delta t^3)$$

Finalement, la formulation variationnelle pour u_1 s'écrit :

$$\begin{aligned} \int_{\Omega} u_1 \cdot v \, dx &= \int_{\Omega} u_0 v \, dx + \Delta t \int_{\Omega} \partial_t u_0 \cdot v \, dx + \frac{\Delta t^2}{2} \int_{\Omega} \partial_t^2 u_0 \cdot v \, dx + \mathcal{O}(\Delta t^3) \\ &= \frac{\Delta t^2}{2} \int_{\Omega} \partial_t^2 u_0 \cdot v \, dx + \mathcal{O}(\Delta t^3) \end{aligned}$$

De plus, u_0 est solution de l'équation ?? à $t = 0$, donc $\partial_t u_0 = 0$. En remplaçant dans l'équation précédente, on obtient :

$$\begin{aligned} \int_{\Omega} \partial_t^2 u^0 \cdot v \, dx - \int_{\Omega} \nabla \cdot \sigma(u_0) \cdot v \, dx &= \int_{\Omega} f_0 \cdot v \, dx \\ \int_{\Omega} \partial_t^2 u^0 \cdot v \, dx + \int_{\Omega} \sigma(u_0) : \nabla v \, dx - \int_{\Gamma} \sigma(u_0) \cdot n \cdot v \, ds &= \int_{\Omega} f_0 \cdot v \, dx \\ \int_{\Omega} \partial_t^2 u^0 \cdot v \, dx + \int_{\Omega} \sigma(u_0) : \varepsilon(v) \, dx &= \int_{\Omega} f_0 \cdot v \, dx + \int_{\Gamma} g_0 \cdot v \, ds \end{aligned}$$

En remplaçant $\sigma(u_0)$ dans l'équation ci-dessus, on obtient :

$$\int_{\Omega} \partial_t^2 u^0 \cdot v \, dx = \int_{\Omega} f_0 \cdot v \, dx - \int_{\Omega} \lambda(\nabla u_0)(\nabla v) + 2\mu \varepsilon(u_0) : \varepsilon(v) \, dx + \int_{\Gamma} g_0 \cdot v \, ds$$

Ce qui donne la forme finale de l'équation variationnelle pour u_1 :

$$\begin{aligned} \int_{\Omega} \partial_t^2 u^1 \cdot v \, dx &= \frac{\Delta t^2}{2} \left(\int_{\Omega} f_0 \cdot v \, dx - \int_{\Omega} \lambda(\nabla u_0)(\nabla v) + 2\mu \varepsilon(u_0) : \varepsilon(v) \, dx + \int_{\Gamma} g_0 \cdot v \, ds \right) \\ &= \frac{\Delta t^2}{2} \left(\int_{\Omega} f_0 \cdot v \, dx - \int_{\Omega} \lambda(\nabla u_0)(\nabla v) + 2\mu \mathbf{Tr}(\varepsilon(u_0)\varepsilon(v)^T) \, dx + \int_{\Gamma} g_0 \cdot v \, ds \right) \end{aligned}$$

Une fois que u_1 est défini, il est possible de trouver une équation à résoudre pour u_{n+1} à partir de u_n et u_{n-1} . Il suffit, de remplacer la dérivée d'ordre 2 en temps par notre approximation en différences finies, ce qui donne l'équation suivante :

$$\int_{\Omega} \rho \frac{u_{n+1} - 2u_n + u_{n-1}}{\Delta t^2} \cdot v \, dx + \int_{\Omega} \sigma(u_n) : \varepsilon(v) \, dx - \int_{\Gamma} g \cdot v \, ds = \int_{\Omega} f_n \cdot v \, dx \quad (1.4)$$

Pour avoir la formulation variationnelle du problème à résoudre, il faut isoler les termes u_{n+1} à gauche de l'équation :

$$\begin{aligned} \int_{\Omega} u_{n+1} \cdot v \, dx &= \frac{\Delta t^2}{\rho} \left(\int_{\Omega} f \cdot v \, dx - \int_{\Omega} \sigma(u_n) : \varepsilon(v) \, dx + \int_{\Gamma} \sigma(u_n) \cdot n \cdot v \, ds \right) \\ &\quad + 2 \int_{\Omega} u_n \cdot v \, dx - \int_{\Omega} u_{n-1} \cdot v \, dx \\ \int_{\Omega} u_{n+1} \cdot v \, dx &= \frac{\Delta t^2}{\rho} \left(\int_{\Omega} f \cdot v \, dx - \int_{\Omega} \lambda(\nabla u_n)(\nabla v) + 2\mu \varepsilon(u_n) : \varepsilon(v) \, dx + \int_{\Gamma} g \cdot v \, ds \right) \\ &\quad + 2 \int_{\Omega} u_n \cdot v \, dx - \int_{\Omega} u_{n-1} \cdot v \, dx \end{aligned}$$

Enfin, le double produit scalaire $\varepsilon(u_n) : \varepsilon(v)$ peut se réécrire comme la trace du produit matriciel $\varepsilon(u_n) \times \varepsilon(v)$, ce qui donne la forme finale de l'équation variationnelle :

$$\begin{aligned} \int_{\Omega} u_{n+1} \cdot v \, dx &= \frac{\Delta t^2}{\rho} \left(\int_{\Omega} f \cdot v \, dx - \int_{\Omega} \lambda(\nabla u_n)(\nabla v) + 2\mu \mathbf{Tr}(\varepsilon(u_n)\varepsilon(v)^T) \, dx + \int_{\Gamma} g \cdot v \, ds \right) \\ &\quad + 2 \int_{\Omega} u_n \cdot v \, dx - \int_{\Omega} u_{n-1} \cdot v \, dx \end{aligned}$$

1.4 Méthodes de résolution d'équations différentielles ordinaires

Les équations d'élasticité linéaire ne possèdent pas de solutions analytiques générales. Cependant, sur des cas simples (Plaques rectangulaires, cylindriques), il est possible de trouver des solutions analytiques. Ces dernières sont données par plusieurs fonctions solutions d'équations différentielles ordinaires (EDO). Il est possible de les résoudre à la main, ou bien d'utiliser des méthodes numériques comme la méthode d'Euler ou Runge-Kutta. Ces deux méthodes permettent d'approximer les solutions d'EDO du premier ordre de la forme :

$$\begin{cases} \dot{y}(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases}$$

La méthode d'Euler (implicite) utilise une approximation simple de la dérivée. Elle est donnée ci-dessous :

$$y(t_i) \approx \frac{y(t_{i+1}) - y(t_i)}{t_{i+1} - t_i}$$

Comme le temps est discrétisé de manière uniforme, on peut écrire $t_{i+1} - t_i = \Delta t$. En remplaçant injectant cette approximation dans l'EDO, on obtient :

$$y(t_{i+1}) = y(t_i) + \Delta t f(t_i, y(t_i))$$

Autrement dit, cela revient à calculer les termes de la suite :

$$\begin{cases} y_{i+1} = y_i + \Delta t f(i, y_i) \\ y_0 = y(t_0) \end{cases}$$

La méthode de Runge-Kutta en particulier la méthode d'ordre 4, utilise une approximation plus précise de la dérivée. Elle est donnée par :

$$y_{i+1} = y_i + \frac{h}{6} (k_1 + k_2 + k_3 + k_4)$$

Où :

$$\begin{cases} k_1 = f(t_i, y_i) \\ k_2 = f(t_i + \frac{h}{2}, y_i + \frac{h}{2} k_1) \\ k_3 = f(t_i + \frac{h}{2}, y_i + \frac{h}{2} k_2) \\ k_4 = f(t_i + h, y_i + h k_3) \end{cases}$$

Et h est le pas de temps. La figure [1.2](#) illustre la méthode de Runge-Kutta d'ordre 4.

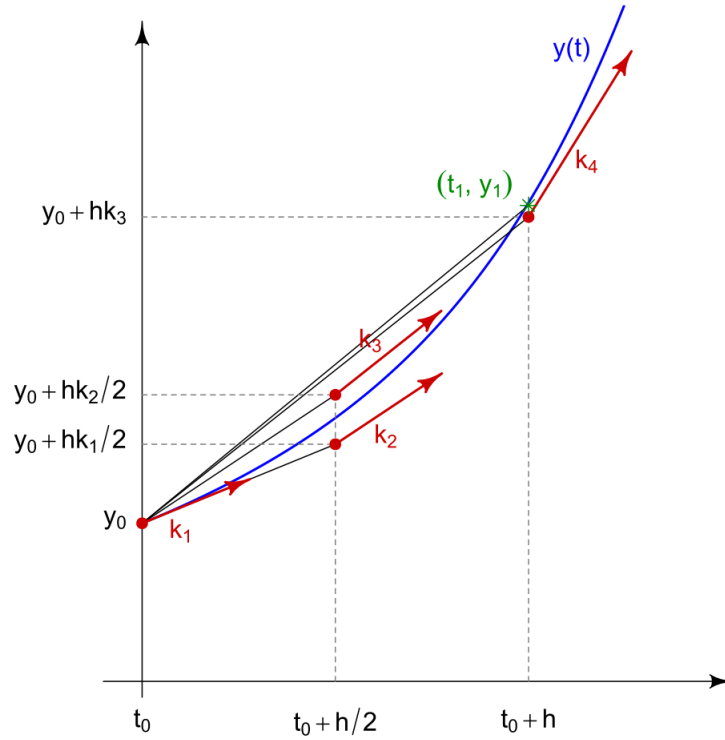


FIGURE 1.2 – Méthode de Runge-Kutta d'ordre 4

Cette méthode est plus efficace que la méthode d'Euler, car l'erreur de consistance relative notée e_n est convergente d'ordre 4, c'est-à-dire que :

$$e_n = \mathcal{O}(\Delta t^5)$$

De manière générale, une méthode numérique est dite convergente d'ordre p si :

$$e_n = \mathcal{O}(\Delta t^{p+1}) \quad (1.5)$$

Dans le cas de la méthode d'Euler, l'erreur de consistance est d'ordre 1.

Cependant, ces deux méthodes sont des méthodes explicites, c'est-à-dire que la solution à l'instant t_{i+1} dépend de la solution à l'instant t_i . Contrairement aux méthodes implicites, qui dépendent de la solution à l'instant t_{i+1} . Par exemple, la méthode de Crank-Nicolson est une méthode implicite qui est donnée par :

$$y_{i+1} = y_i + \frac{\Delta t}{2} (f(t_i, y_i) + f(t_{i+1}, y_{i+1})) \quad (1.6)$$

Ces méthodes sont plus stables dans le cas d'équation différentielles dites raides, c'est à dire une EDO telle que l'équation s'écrit sous la forme :

$$\frac{du}{dt} = F(u)$$

Où F est une matrice, alors il existe deux valeurs propres λ_1 et λ_2 telles que :

$$|\lambda_1| \gg |\lambda_2|$$

L'exemple le plus simple est donné par :

$$\frac{du}{dt} = \begin{pmatrix} -1000 & 0 \\ 0 & -1 \end{pmatrix} u$$

Ce genre d'équation est difficile à résoudre avec une méthode explicite, car le pas de temps doit être très petit pour assurer la stabilité de la méthode. C'est pourquoi les méthodes implicites sont préférées dans ce cas. Un exemple de problème raide sera abordé dans la section ??.

1.5 Préconditionneur

Un preconditionneur est utilisé en algèbre linéaire pour résoudre un système linéaire de la forme $Ax = B$ avec $A \in \mathcal{M}_n(\mathbb{R})$, $x \in \mathbb{R}^n$ et $b \in \mathbb{R}^n$. Concrètement, il s'agit d'une matrice P telle que en multipliant l'équation de base par P^{-1} , on obtient un système plus simple à résoudre. Plus simple veut dire :

- Conditionnement de la matrice $P^{-1}A$ plus petit (Moins de sensibilité aux erreurs d'arrondi de la machine)
- Moins d'itérations pour converger vers la solution

La librairie Feel++ contient plusieurs preconditionneurs, durant ce stage, deux preconditionneurs ont été utilisés :

- Préconditionneur de Jacobi
- Factorisation LU incomplète

Avant de décrire le preconditionneur, il faut remarquer que A peut s'écrire comme une somme de trois matrices $A = L + D + U$, où L contient les coefficients strictement inférieurs à la diagonale de A , D contient les coefficients diagonaux de A et U contient les coefficients strictement supérieurs à la diagonale de A . Le preconditionneur de Jacobi consiste à isoler la matrice diagonale de A et à l'inverser. Le preconditionneur de Jacobi est donné par :

$$P = D^{-1} \tag{1.7}$$

Ce preconditionneur est très simple à implémenter, est fonctionne bien pour les matrices à diagonale dominante, c'est à dire les matrices où les coefficients diagonaux sont plus grands que la somme des coefficients non-diagonaux, concrètement :

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \tag{1.8}$$

De plus, la matrice P est simple à paralléliser, ce qui sera très utile pour nos calculs qui sont nécessairement parallélisés.

Le deuxième preconditionneur utilisé est la factorisation LU incomplète. La factorisation LU consiste à écrire une matrice A comme le produit de deux matrices L et U où L est une matrice triangulaire inférieure et U est une matrice triangulaire supérieure. La factorisation LU incomplète consiste à ne pas calculer tous les coefficients de L et U mais seulement ceux qui sont non nuls. Cela permet de réduire le temps de calcul de la factorisation LU. Le preconditionneur de factorisation LU incomplète est donné par :

$$P = LU \tag{1.9}$$

1.6 Accélération

Pour un temps de calcul optimal, il faut paralléliser les calculs. C'est-à-dire, utiliser plusieurs coeurs de processeurs pour exécuter les calculs. Pour mesurer l'efficacité de la parallélisation, on utilise l'accélération (speedup en anglais) dont la formule est donnée par :

$$S(p) = \frac{T_p}{T_1}$$

où p désigne le nombre de coeurs utilisés, T_p le temps de calcul avec p coeurs et T_1 le temps de calcul avec un seul processeur. En théorie l'accélération devrait être proportionnelle au nombre de coeurs utilisés, mais en pratique, elle ne l'est pas. C'est du au nombre communications entre les coeurs qui augmentent avec le nombre de coeurs utilisés. De plus, l'accélération dépend également de la taille des éléments de notre maillage. Un maillage trop épais donnera des noeuds situés sur plusieurs coeurs, et donc des communications plus fréquentes. Il y a donc un nombre optimal de coeurs à utiliser pour un maillage donné.

1.7 Feel++

Feel++ est une librairie en C++ contenant des méthodes de résolution d'EDP dont méthodes des éléments finis et éléments spectraux. Elle a été développée par le Cemosis dont le directeur est Christophe Prud'homme mon tuteur de stage. Avec cette librairie, j'ai résolu deux types de cas :

Le cas 2D, avec une structure en 2 dimension et dont les éléments ne peuvent se mouvoir que dans 2 directions.

Puis le cas 3D, avec une structure en 3 dimensions et dont les éléments se meuvent dans les 3 directions de l'espace \mathbb{R}^3

Chapitre 2

Résultats obtenus

2.1 Méthode de comparaison des résultats

Comme expliqué dans l'introduction, les solutions calculés avec Feel++ doivent être comparées avec des solutions analytiques si elles existent. Et si elles n'existent pas, il faut les comparer avec une solution obtenue sur un maillage très fin ainsi que les solutions obtenues par éléments finis ou IGA. De plus, un maillage de plaque trouée existe dans les serveurs d'Avnir Energy, ce maillage vient d'une vraie plaque sur laquelle l'entreprise fait des expériences avec des accéléromètres. Les signaux enregistrés par ces capteurs pourront être utilisés pour valider les solutions obtenus avec Feel++.

Quelque soit la méthode de référence, il convient d'établir une mesure permettant de comparer les résultats SEM aux résultats de référence. Pour cela, la norme L^2 mesure l'erreur entre la solution numérique et de référence. Elle est donnée par :

$$\|u - u_h\|_{L^2} = \sqrt{\int_{\Omega} (u - u_h)^2 dx} \quad (2.1)$$

Où u représente la solution de référence (analytique, IGA, FEM), u_h la solution numérique (SEM). De plus, les solutions sont discrète, il faut donc discrétiser la norme L^2 , ce qui donne la formule ci-dessous :

$$\|u - u_h\|_{L^2} \approx \sqrt{\sum_{i=1}^n (u(t_i) - u_h(t_i))^2} \quad (2.2)$$

Où t_i est un instant donné.

Enfin pour comparer la solution expérimentale avec la solution numérique, il faut calculer la valeur moyenne de la solution sur un disque de rayon r autour du point de mesure. C'est dû au fait que les accéléromètres mesurent la valeur moyenne du déplacement sur une certaine surface. La valeur moyenne est donnée par :

$$\bar{u}_{x_0} = \frac{1}{\pi r^2} \int_{B_r(x_0)} u dx \quad (2.3)$$

Où $B_r(x)$ est le disque de rayon r autour du point x_0 . Comme pour la norme L^2 , il faut discrétiser la valeur moyenne, ce qui donne la formule suivante :

$$\bar{u}_{x_0} \approx \frac{1}{\pi r^2} \sum_{x_i \in B_r(x_0)} u(x_i) \quad (2.4)$$

Où x_i est un point du disque de rayon r .

Ce qui donne un signal de déplacement moyen en fonction du temps. Pour comparer les deux signaux, on utilise la norme L^2 discrétisée ou bien une autocorrélation.

2.2 Efficacité de la méthode

2.2.1 Cas 2D

Avant le stage, les employés d'Avnir Energy ont implémenté la méthode des éléments finis et la méthode isogéométrique sur plaque carrée. Cette dernière est encastree sur tout le bord droit et un effort est appliqué sur le milieu du bord gauche. La figure 2.1 illustre le modèle.

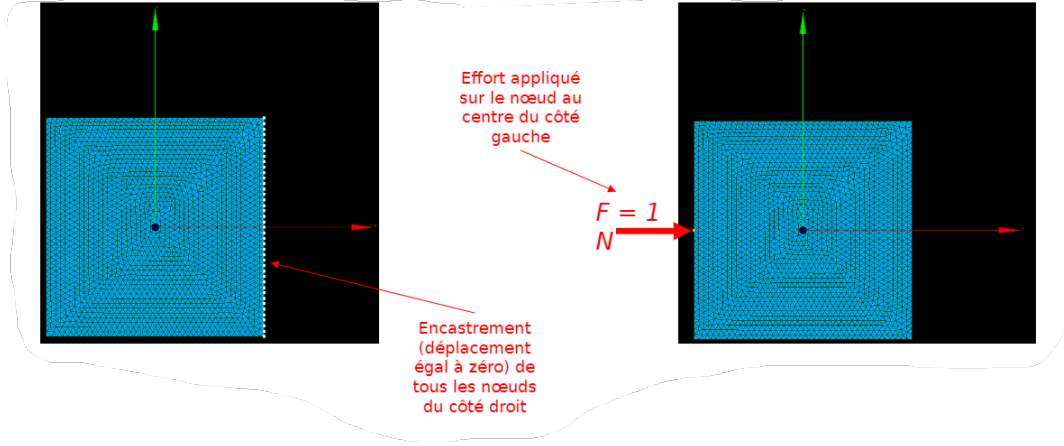


FIGURE 2.1 – Plaque carrée encastree

Ce modèle représente une structure en acier, il faut donc adapter les coefficients de Lamé. λ et μ sont donnés par :

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$$

$$\mu = \frac{E}{2(1+\nu)}$$

Dans le cas de l'acier, $E = 210 \text{ GPa}$ et $\nu = 0.3$. La densité de l'acier est aussi comprise dans les équations et vaut $\rho = 7800 \text{ kg/m}^3$. Enfin, la force appliquée sur la plaque varie dans le temps, elle est donnée par :

$$\begin{cases} h(t) = \sin(2\pi f_c t) e^{-5(f_c t - 2)^2} \\ f_c = 20000 \text{ Hz} \end{cases} \quad (2.5)$$

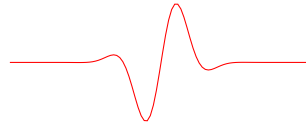


FIGURE 2.2 – Effort appliqué sur la plaque

La figure 2.2 montre le graphe de l'effort appliqué sur la plaque.

Le pas de temps maximal peut se calculer à partir de la fréquence f_c . Ce pas de temps est donné par :

$$\Delta t \leq \frac{1}{10f_c} \quad (2.6)$$

De plus, pour observer la flexion de la plaque, il faut une taille d'élément suffisamment petite. Elle peut être calculée à partir de la longueur d'onde λ , qui est donnée par :

$$h \leq \frac{\lambda}{10} \quad (2.7)$$

Ici, $\lambda \approx 0.15m$, les paramètres de la Δt et h sont alors donnés par :

$$\begin{aligned} \Delta t &\leq 5 \times 10^{-6} s \\ h &\leq 0.015m \end{aligned}$$

J'ai donc choisi $\Delta t = 2 \times 10^{-6} s$ et $h = 0.01m$.

Avec ces informations, j'ai lancé les calculs sur une plaque carrée de côté 2m et généré le maillage. Les résultats obtenus sont présentés dans la figure 2.3.



FIGURE 2.3 – Video de la solution. Il faut cliquer sur l'image pour lancer la vidéo

Le résultat obtenu utilise des polynômes de degré 3 et un maillage triangulaire.

Le tableau ?? montre les différents paramètres utilisés pour obtenir le résultat 2.3

Paramètres	Valeurs
Taille des éléments	0.01 m
Nombre d'éléments	92556
Nombre de noeuds	46680
Nombre de degré de liberté	835406
E	210000000000 Pa
ν	0.3
ρ	7800 kg/m ³
Préconditionneur	LU

TABLE 2.1 – Paramètres utilisés pour obtenir le résultat de 2.3

Où E désigne le module de Young, ν le coefficient de Poisson et ρ la densité de l'acier

Pour s'assurer de la convergence de la méthode, j'ai lancé un calcul sur maillage très fin et des polynômes de degré 4. Ce calcul prend plus de temps que les calculs à comparer (5 fois plus long dans le cas 2D), mais une fois fait, il sera utilisé comme solution de référence pour comparer les résultats obtenus avec les méthodes précédentes.

Comme mentionné dans la section précédente, j'ai extrait les valeurs de déplacement en trois points de la plaque (nommé S1, S2 et S3). La figure 2.4 montre les emplacements choisis.

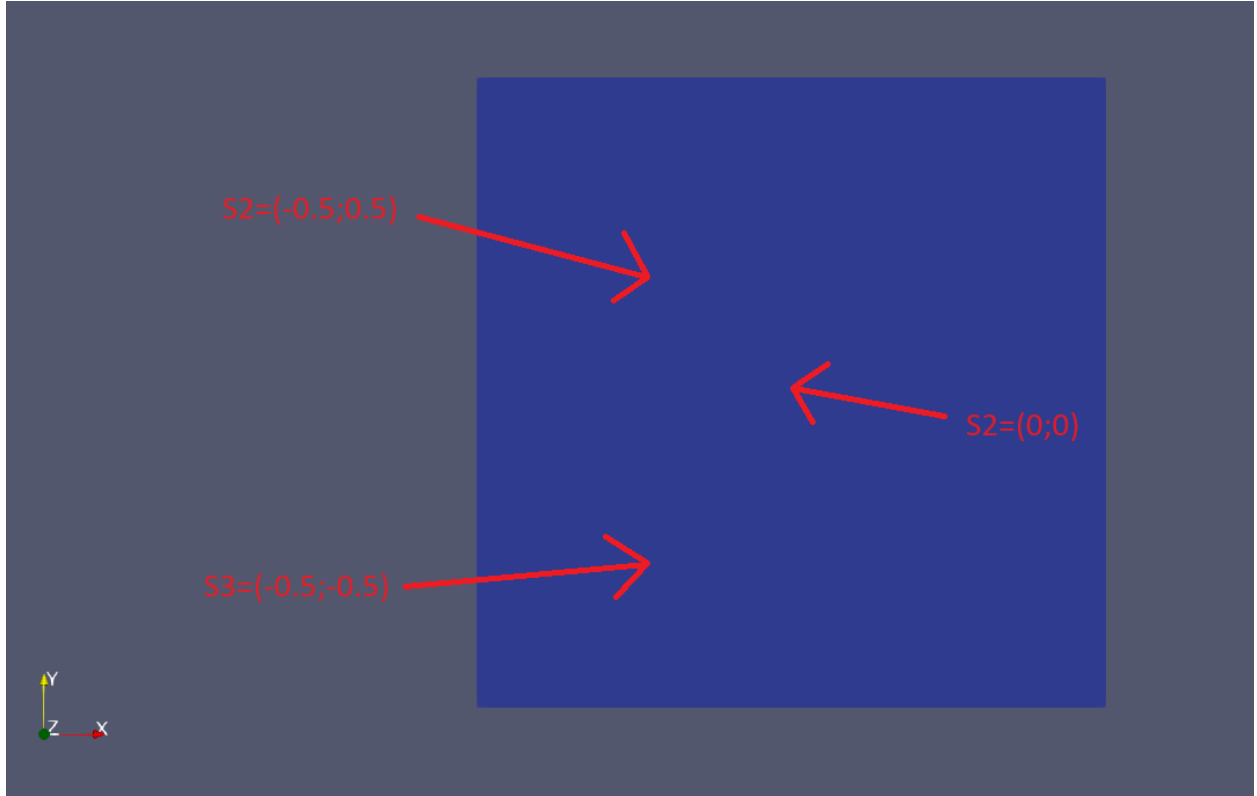


FIGURE 2.4 – Emplacement des capteurs

J'ai donc utilisé ces points pour comparer les résultats obtenus sur le maillage fin avec des résultats obtenus avec des polynômes de degré 3 et une taille d'éléments qui varie. La figure 2.5 montre l'erreur L^2 entre la solution de référence et la solution approchée en fonction de la taille des éléments.

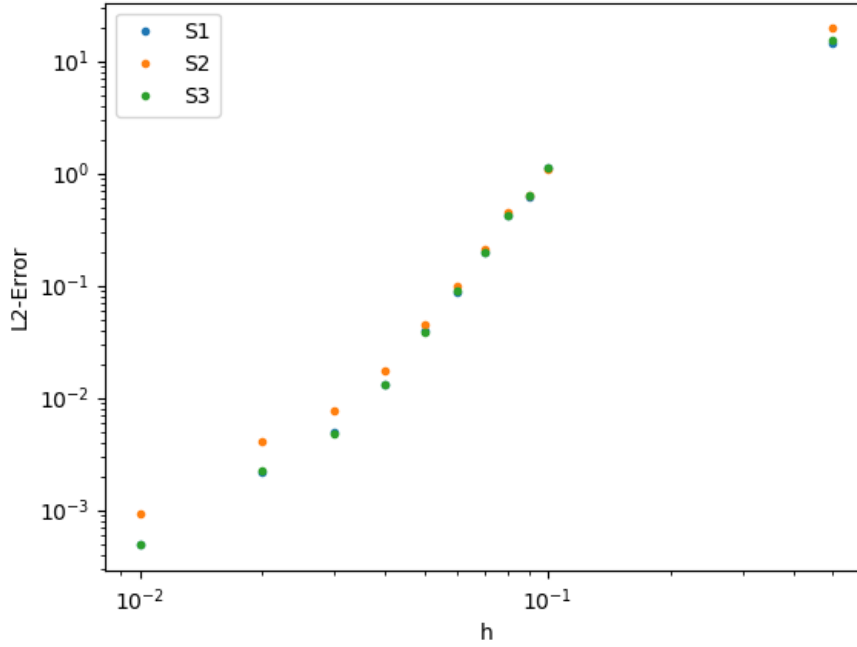


FIGURE 2.5 – Erreur L^2 en fonction de la taille des éléments

Le graphique montre que l'erreur diminue avec la taille des éléments avec une erreur de l'ordre de 10^{-3} pour des éléments de taille $0.01m$. Cela montre que la méthode est convergente en maillage.

2.2.2 Cas 3D

Pour vérifier si la méthode SEM fonctionne en 3D, mes collègues et moi avons décidé de la comparer avec une solution analytique. Ce choix a été fait car il était possible de comparer le résultat avec une expérimentation. Cependant, cette option n'est pas envisageable car la présence d'accéléromètres lors des expérimentations risque de donner des signaux différents de la solution numérique. Nous avons donc choisi la première option avec deux cas simples en 3D. Le premier est une plaque rectangulaire, le second une plaque cylindrique. Dans les deux cas, la plaque est simplement supporté (i.e déplacement nul) sur les bords comme illustré dans les figures 2.6 et 2.7.

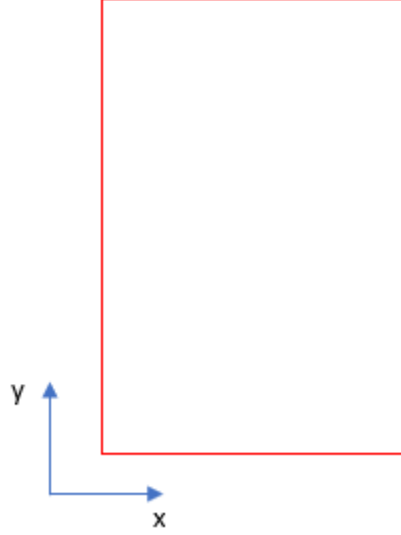


FIGURE 2.6 – Plaque rectangulaire, les bords rouges sont simplement supportés

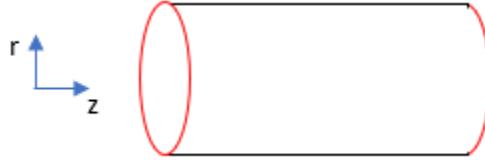


FIGURE 2.7 – Plaque cylindrique, les bords rouges sont simplement supportés

Dans le cas de la plaque rectangulaire, les dimensions sont $a = 0.4m$, $b = 0.6m$ et $h = 0.003m$. Où a est la longueur de la plaque, b la largeur et h l'épaisseur. La solution analytique a été calculé par un collègue de l'entreprise, et est donnée par :

$$W(x_1, x_2, t) = \sum_{n=1}^{+\infty} \sum_{m=1}^{+\infty} a_{n,m}(t) f_{n,m}(x_1, x_2) \quad (2.8)$$

W désigne le déplacement normal à la plaque (i.e dans la direction z) en (x_1, x_2) , $a_{n,m}$ est le mode solution de l'EDO :

$$\begin{cases} \ddot{a}_{n,m}(t) + \omega_{n,m}^2 a_{n,m}(t) = \frac{\tilde{p}(t) f_{n,m}(X_1, X_2)}{M_{n,m}} \\ a_{n,m}(0) = \dot{a}_{n,m}(0) = 0 \end{cases}$$

Où $\omega_{n,m}$ s'écrit :

$$\omega_{n,m} = \sqrt{\frac{D}{\rho h} \left(\left(\frac{n\pi}{a} \right)^2 + \left(\frac{m\pi}{b} \right)^2 \right)}$$

\tilde{p} désigne l'ondelette dans l'équation 2.5 mais avec $f_c = 161107Hz$. Le choix de f_c est donné en annexe. De plus $M_{n,m}$ est donné par

$$M_{n,m} = \rho h \frac{a}{2} \frac{b}{2} = \rho \frac{hab}{4}$$

Enfin $f_{n,m}$ est la déformée modale donnée par :

$$f_{n,m}(x_1, x_2) = \sin\left(\frac{n\pi}{a}x_1\right) \sin\left(\frac{m\pi}{b}x_2\right)$$

Les calculs qui ont amenés à l'équation 2.2.2 sont détaillés dans l'annexe A

(X_1, X_2) désigne le point d'impact de la force sur la plaque. Comme la solution analytique est donnée par une série, il faut tronquer la série à un certain rang N .

De plus, l'EDO est du second ordre, or, les méthodes de RK et d'Euler ne permettent de calculer que des EDO du premier ordre. Pour passer au premier ordre, il faut introduire la variable $b_{p,k}(t) = \dot{a}_{p,k}(t)$ et écrire le vecteur $X = (a_{p,k}(t), b_{p,k}(t))^T$. Il suffit alors de dériver X pour obtenir le système d'EDO suivant :

$$\dot{X} = \begin{pmatrix} \dot{a}_{p,k}(t) \\ \dot{b}_{p,k}(t) \end{pmatrix} = \begin{pmatrix} b_{p,k}(t) \\ \frac{\tilde{p}(t)f_{p,k}(X_1, X_2)}{M_{n,m}} - \omega_{p,k}^2 a_{p,k}(t) \end{pmatrix}$$

En résolvant ce système avec Runge-Kutta inclus dans le module SciPy, la méthode ne converge pas vers une solution stable. Une solution serait de baisser le paramètre de tolérance de la méthode, mais cela augmenterait le temps de calcul. Au vu du temps qu'il restait, j'ai décidé de résoudre à la main l'EDO, en utilisant une variation de la constante. Cela revient à trouver deux solution particulières de l'EDO homogène suivante :

$$\ddot{a}_{p,k}(t) + a_{p,k}(t)\omega_{p,k}^2 = 0$$

Il y a deux solutions triviales qui sont données par :

$$\begin{aligned} \lambda^{(1)}(t) &= \cos(\omega_{p,k}t) \\ \lambda^{(2)}(t) &= \sin(\omega_{p,k}t) \end{aligned}$$

En effet, $\ddot{\lambda}^{(1)}(t) = -\omega_{p,k}^2 \cos(\omega_{p,k}t)$ et $\ddot{\lambda}^{(2)}(t) = -\omega_{p,k}^2 \sin(\omega_{p,k}t)$

La résolution de l'EDO avec le membre de droite revient à chercher une solution $a_{p,k}$ de la forme :

$$\begin{cases} a_{p,k}(t) &= \lambda^{(1)}(t) \sin(\omega_{p,k}t) + \lambda^{(2)}(t) \cos(\omega_{p,k}t) \\ \dot{a}_{p,k}(t) &= \lambda^{(1)}(t) \omega_{p,k} \cos(\omega_{p,k}t) - \lambda^{(2)}(t) \omega_{p,k} \sin(\omega_{p,k}t) \end{cases}$$

D'après l'expression de $\dot{a}_{p,k}$, on peut déduire que :

$$\dot{\lambda}^{(1)} \sin(\omega_{p,k}t) + \dot{\lambda}^{(2)} \cos(\omega_{p,k}t) = 0 \quad (2.9)$$

De plus, en dérivant une seconde fois $a_{p,k}$, on obtient :

$$\begin{aligned} \ddot{a}_{p,k}(t) &= \dot{\lambda}^{(1)} \omega_{p,k} \cos(\omega_{p,k}t) - \lambda^{(1)}(t) \omega_{p,k} \sin(\omega_{p,k}t) \\ &\quad - \dot{\lambda}^{(2)} \omega_{p,k} - \lambda^{(2)}(t) \omega_{p,k}^2 \cos(\omega_{p,k}t) \end{aligned} \quad (2.10)$$

En injectant l'équation 2.10 dans l'EDO, on obtient :

$$\begin{aligned} &\dot{\lambda}^{(1)}(t) \omega_{p,k} \cos(\omega_{p,k}t) - \lambda^{(1)}(t) \omega_{p,k} \sin(\omega_{p,k}t) - \dot{\lambda}^{(2)} \omega_{p,k} \sin(\omega_{p,k}t) - \lambda^{(2)} \omega_{p,k}^2 \cos(\omega_{p,k}t) \\ &+ \omega_{p,k}^2 \lambda^{(1)}(t) \sin(\omega_{p,k}t) + \omega_{p,k}^2 \lambda^{(2)} \cos(\omega_{p,k}t) = \frac{\tilde{p}(t)f_{p,k}(X_1, X_2)}{M_{p,k}} \end{aligned} \quad (2.11)$$

Quatres termes s'annulent dans l'équation 2.11, ce qui donne :

$$\dot{\lambda}^{(1)}(t)\omega_{p,k} \cos(\omega_{p,k}t) - \dot{\lambda}^{(2)}\omega_{p,k} \sin(\omega_{p,k}t) = \frac{\tilde{p}(t)f_{p,k}(X_1, X_2)}{M_{p,k}} \quad (2.12)$$

Finalement, en utilisant l'équation 2.9 et 2.12, $a_{p,k}$ s'écrit comme une combinaison linéaire de $\lambda^{(1)}$ et $\lambda^{(2)}$. Ces dernière sont solutions du système suivant :

$$\begin{cases} \dot{\lambda}^{(1)}(t)\omega_{p,k} \cos(\omega_{p,k}t) - \dot{\lambda}^{(2)}\omega_{p,k} \sin(\omega_{p,k}t) = \frac{\tilde{p}(t)f_{p,k}(X_1, X_2)}{M_{p,k}} \\ \dot{\lambda}^{(1)} \sin(\omega_{p,k}t) + \dot{\lambda}^{(2)} \cos(\omega_{p,k}t) = 0 \end{cases} \quad (2.13)$$

La deuxième ligne peut se réécrire comme :

$$\dot{\lambda}^{(1)}(t) = -\dot{\lambda}^{(2)}(t) \frac{\cos(\omega_{p,k}t)}{\sin(\omega_{p,k}t)} \quad (2.14)$$

En injectant la relation 2.14 dans la première ligne de l'équation 2.13, on obtient :

$$\begin{aligned} \frac{-\dot{\lambda}^{(2)}(t)\omega_{p,k} \cos^2(\omega_{p,k}t)}{\sin^2(\omega_{p,k}t)} - \frac{-\dot{\lambda}^{(2)}(t)\omega_{p,k} \sin^2(\omega_{p,k}t)}{\sin^2(\omega_{p,k}t)} &= \frac{\tilde{p}(t)f_{p,k}(X_1, X_2)}{M_{p,k}} \\ \frac{\dot{\lambda}^{(2)}(t)\omega_{p,k}}{\sin(\omega_{p,k}t)} &= \frac{\tilde{p}(t)f_{p,k}(X_1, X_2)}{M_{p,k}} \\ \dot{\lambda}^{(2)}(t) &= \frac{\tilde{p}(t)f_{p,k}(X_1, X_2) \sin(\omega_{p,k}t)}{\omega_{p,k}M_{p,k}} \end{aligned}$$

En intégrant des deux côtés, on obtient :

$$\lambda^{(2)}(t) = \frac{f_{p,k}(X_1, X_2)}{\omega_{p,k}M_{p,k}} \int \tilde{p}(t) \sin(\omega_{p,k}t) dt \quad (2.15)$$

En injectant l'équation 2.15 dans l'équation 2.14, $\lambda^{(1)}$ s'écrit comme :

$$\lambda^{(1)}(t) = -\frac{f_{p,k}(X_1, X_2)}{\omega_{p,k}M_{p,k}} \int \tilde{p}(t) \cos(\omega_{p,k}t) dt \quad (2.16)$$

Ces deux primitives ne peuvent avoir d'expression de forme fermée dû au théorème de Liouville (en algèbre différentielle). Cependant, elle peut être calculé avec une méthode d'approximation d'intégrale (Trapèzes, Simpson ...). En écrivant $G(t) = \int_0^t \tilde{p}(t) \cos(\omega_{p,k}t) dt$ et $H(t) = \int_0^t \tilde{p}(t) \sin(\omega_{p,k}t) dt$, $a_{p,k}(t)$ s'écrit ainsi :

$$a_{p,k}(t) = (G(t) + C_1) \sin(\omega_{p,k}t) + (H(t) + C_2) \cos(\omega_{p,k}t), \quad C_1, C_2 \in \mathbb{R} \quad (2.17)$$

De plus, $a_{p,k}(0) = 0$ et $\dot{a}_{p,k}(0) = 0$, donc d'après le théorème de Cauchy-Lipschitz, il existe un unique couple de variables C_1 et C_2 qui vérifie ces conditions. Il n'y a plus qu'à déterminer ces constantes.

Comme $a_{p,k}(0) = 0$, on a :

$$\begin{aligned} a_{p,k}(0) &= (G(0) + C_1) \sin(0) + (H(0) + C_2) \cos(0) = 0 \\ &= H(0) + C_2 = 0 \\ \iff C_2 &= -H(0) \end{aligned}$$

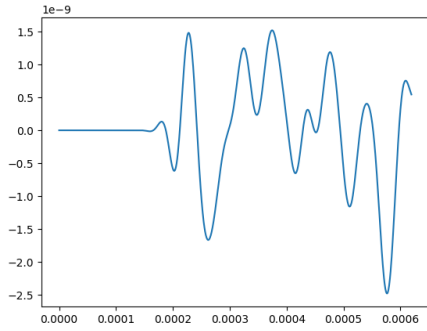
Or, $H(0) = \int_0^0 \tilde{p}(t) \sin(\omega_{p,k}t) dt = 0$, d'où $C_2 = 0$. Enfin, comme $\dot{a}_{p,k}(0) = 0$, on a :

$$\begin{aligned}
\dot{a}_{p,k}(t) &= \omega_{p,k}(G(t) + C_1) \cos(\omega_{p,k}t) + \dot{G}(t) \sin(\omega_{p,k}t) \\
&\quad - \omega_{p,k}(H(t) + C_2) \sin(\omega_{p,k}t) + \dot{H}(t) \cos(\omega_{p,k}t) \\
\iff \dot{a}_{p,k}(0) &= \omega_{p,k}(G(0) + C_1) \cos(0) + \dot{G}(0) \sin(0) \\
&\quad - \omega_{p,k}H(0) \sin(0) + \dot{H}(0) \cos(0) = 0 \\
\iff \dot{a}_{p,k}(0) &= \omega_{p,k}(G(0) + C_1) = 0 \\
\iff C_1 &= -G(0)
\end{aligned}$$

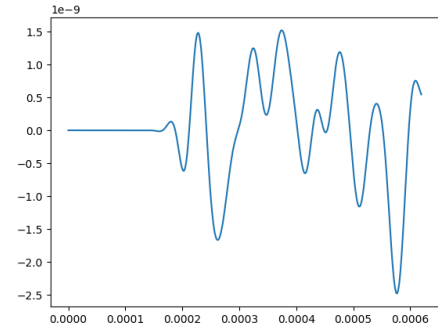
De la même manière que pour C_2 , $C_1 = 0$. Finalement, $a_{p,k}$ s'écrit comme :

$$a_{p,k}(t) = G(t) \sin(\omega_{p,k}t) + H(t) \cos(\omega_{p,k}t) \quad (2.18)$$

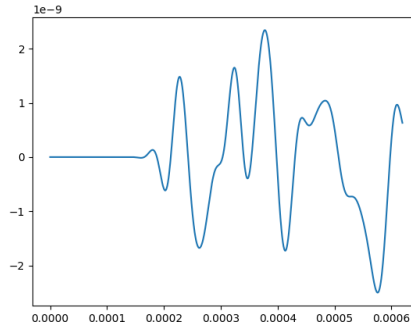
En calculant, G et H avec la méthode des trapèzes, on obtient la solution de l'EDO. Enfin, injecter cette solution dans la solution analytique donne le déplacement normal à la plaque en fonction du temps. La figure 2.8 montre la solution analytique obtenue.



(a) S1



(b) S2



(c) S3

FIGURE 2.8 – Solution analytique sur les 3 capteurs

Pour mesurer l'erreur entre la solution analytique et la solution numérique, j'ai choisi trois points de la plaque. La figure 2.9 montre les emplacements choisis.

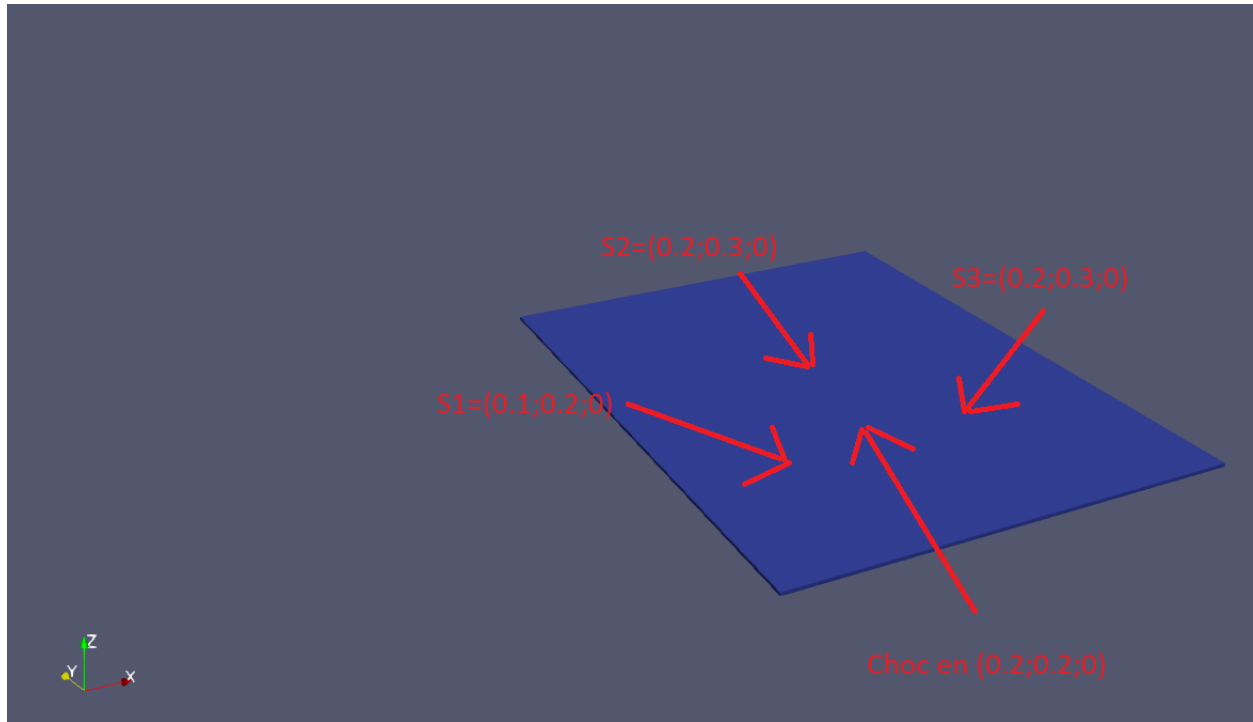


FIGURE 2.9 – Emplacement des capteurs et du choc

Enfin, j'ai lancé les calculs sur une plaque rectangulaire de dimensions $a = 0.4m$, $b = 0.6m$ et $h = 0.003m$. Les résultats obtenus sont présentés dans la figure ??.

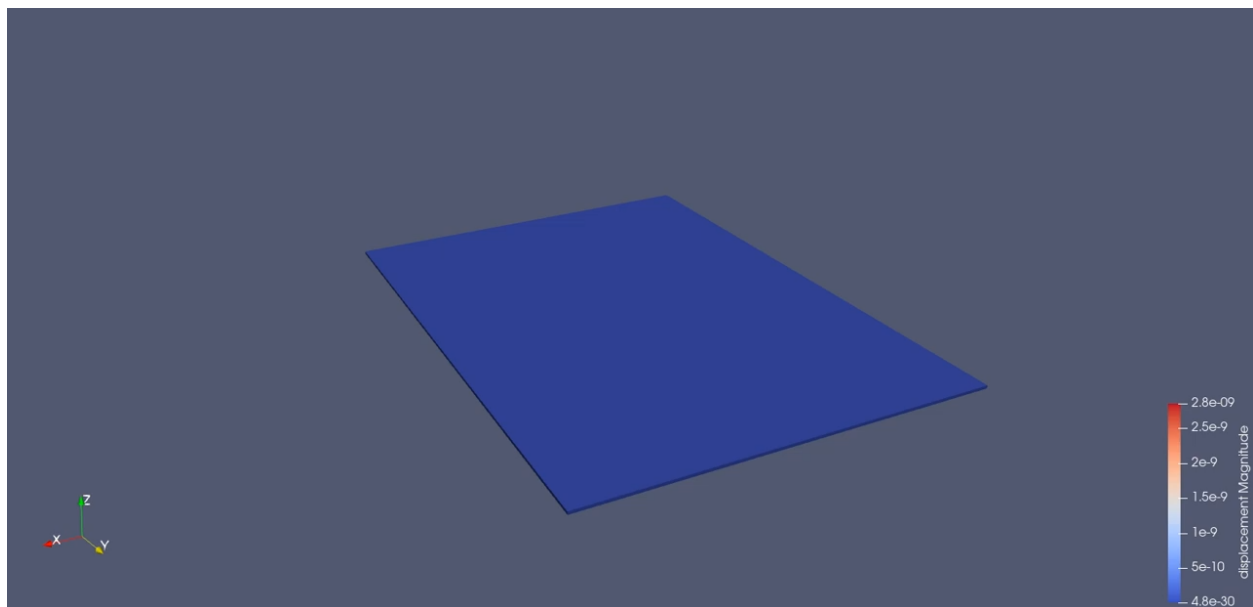
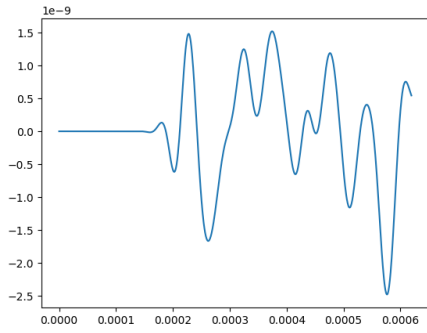


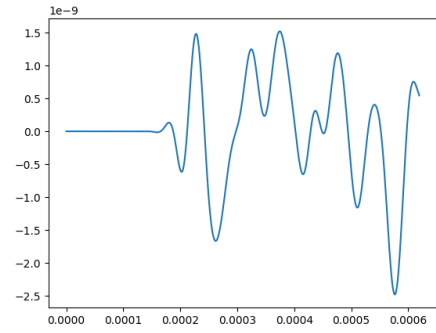
FIGURE 2.10 – Solution numérique sur la plaque

La figure 2.10 montre la solution sur le maillage de la plaque au cours du temps. Et la figure 2.11 montre

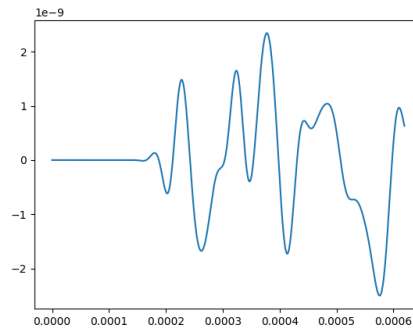
les signaux enregistrés sur les trois capteurs



(a) S1



(b) S2



(c) S3

FIGURE 2.11 – Signaux obtenus aux niveaux des 3 capteurs lors de la résolution numérique

En utilisant la norme L^2 discrétisée, j'ai comparé les résultats obtenus avec la solution analytique. Le tableau 2.2 montre les erreurs obtenues.

Capteurs	Erreur L2
S1	2.4687e-4
S2	2.4687e-4
S3	2.9935e-4

TABLE 2.2 – Erreur entre la solution analytique et la solution numérique

Le tableau 2.3 montre les différents paramètres utilisés pour obtenir le résultat de 2.10

Paramètres	Valeurs
Taille des éléments	0.003 m
Nombre d'éléments	104143
Nombre de noeuds	52071
Nombre de degré de liberté	1041143
E	210000000000 Pa
ν	0.3
ρ	7800 kg/m^3
Préconditionneur	Jacobi
Degré des polynômes	3

TABLE 2.3 – Paramètres utilisés pour obtenir le résultat de 2.10

2.3 Comparaison des résultats avec les méthodes précédentes

Pour résoudre des problèmes impliquant une équation d'élasticité, les employés d'Avnir Energy utilisent un logiciel qui fait les calculs à leur place comme Code_Aster. C'est sur ce logiciel que les résultats obtenus sur la plaque 2D avec Feel++ ont été obtenus. Code_Aster utilise la méthode des éléments finis classique pour faire ses simulations. De plus, Feel++ permet aussi de résoudre des équations d'élasticité avec la méthode des éléments finis. J'ai donc comparé les résultats obtenus avec la méthode SEM avec ceux obtenus avec la méthode des éléments finis que ce soit avec Code_Aster ou avec Feel++. La figure 2.12 montre les résultats obtenus.

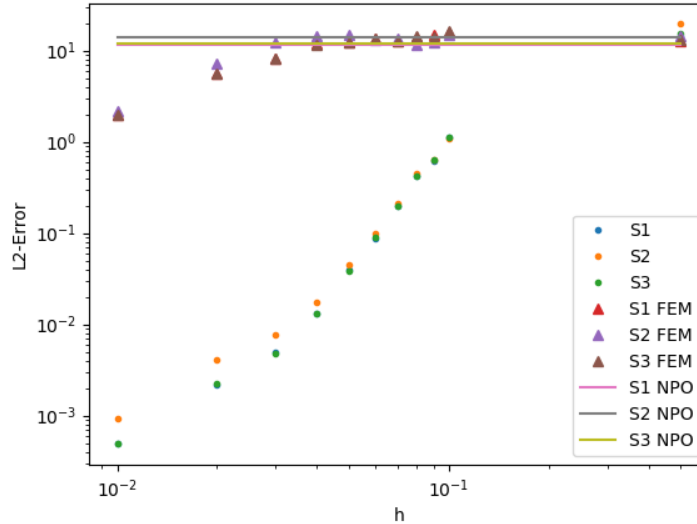


FIGURE 2.12 – Comparaison des erreurs entre la méthode par élément finis et élément spectraux

Où FEM désigne la méthode FEM sur Feel++, NPO désigne la méthode FEM sur Code_Aster.

Le graphique montre que la méthode SEM converge vers la solution de référence, là où l'erreur liée à la FEM converge plus lentement avec la taille des éléments. Cela montre que la méthode SEM est plus précise que la méthode FEM.

Pour comparer encore plus les méthodes, j'ai comparé la méthode FEM et SEM à iso-degré de liberté

(Environ 94000 au total). Le tableau 2.4 montre l'erreur de chaque capteur entre la solution numérique et la solution de référence.

Capteur	Méthodes	Erreur L2
S1	FEM	1.9758
	SEM	0.0050
S2	FEM	2.1534
	SEM	0.0078
S3	FEM	1.9744
	SEM	0.0048

TABLE 2.4 – Erreur commise par les solutions sur les 3 capteurs par rapport à la solution de référence

2.4 Comparaison des temps de calcul

D'après les résultats précédents, la méthode SEM est plus précise que la méthode FEM. Cependant, il faut aussi s'assurer de la rapidité de cette méthode. Pour éviter des calculs trop longs, il faut lancer les calculs en parallèle. J'ai donc comparé les temps de calculs de la méthode SEM avec la méthode FEM. Le tableau 2.5 montre les temps de calculs obtenus pour les deux méthodes sur une plaque carrée de côté 2m et un maillage de taille 0.01m.

Méthode	Temps de calcul (s)
SEM	1024
FEM	629

TABLE 2.5 – Comparaison du temps de calcul entre les deux méthodes à iso taille d'élément

Comme pour le tableau 2.4, j'ai comparé le temps de calcul entre les deux méthodes à iso degré de liberté (94000) et iso erreur (10^0). Le tableau 2.6 montre le temps de calcul sur les deux méthodes avec le même nombre de degré de liberté et le tableau 2.7 fait la même chose mais avec la même erreur entre les deux méthodes.

Méthode	Temps de calcul (s)
SEM	712
FEM	618

TABLE 2.6 – Comparaison du temps de calcul entre les deux méthodes à iso degré de liberté

Méthode	Temps de calcul (s)
SEM	41
FEM	629

TABLE 2.7 – Comparaison du temps de calcul entre les deux méthodes à iso erreur

Le tableau montre que la méthode FEM est à peine plus rapide que la méthode SEM à iso degré de liberté et iso taille d'élément. En revanche, la méthode SEM est plus rapide à iso erreur.

De plus, le temps de calcul est encore plus rapide en optimisant le nombre de coeurs utilisés.

J'ai donc lancé les calculs sur le même maillage avec la même taille d'éléments, mais en utilisant entre 1 et 3 coeurs. Le tableau 2.8 montre les temps de calculs obtenus.

Coeurs	Temps de calcul (s)
1	2192
2	1024
3	2751

TABLE 2.8 – Comparaison des temps de calculs en utilisant différents nombres de coeurs

Ce tableau montre le gain de temps du parallélisme. En utilisant 2 coeurs, le temps de calcul est divisé par 2. Cependant, en utilisant 3 coeurs, le temps de calcul est plus long.

Pour le cas de la plaque 3D, j’ai aussi lancé des calculs avec des configurations identiques (même maillage, taille d’éléments ...), mais en utilisant entre 1 et 6 coeurs. Le tableau 2.9 montre les temps de calculs obtenus.

Coeurs	Temps de calcul (s)
1	869
2	504
3	341
4	404
5	477
6	414

TABLE 2.9 – Comparaison des temps de calcul en fonction du nombre de coeurs

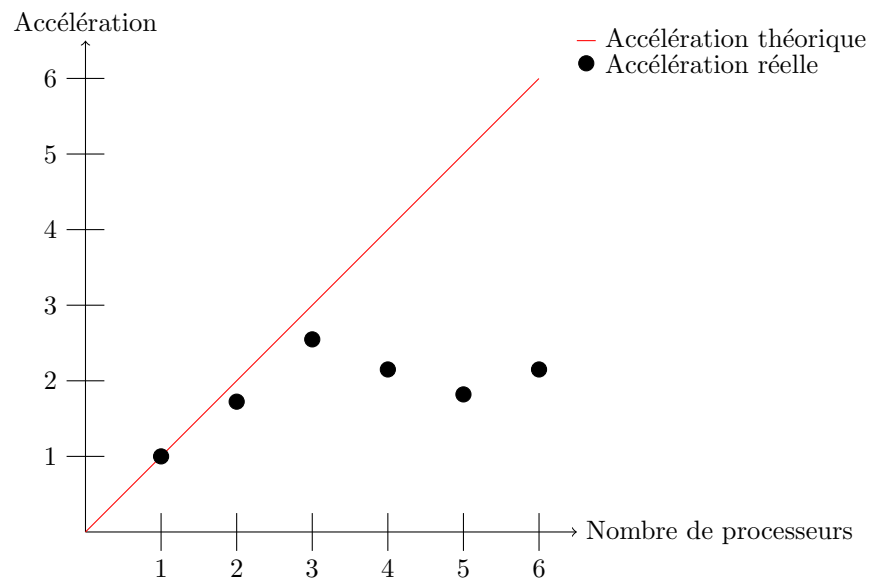


FIGURE 2.13 – Évolution de l’accélération en fonction du nombre de coeurs

La figure 2.13 montre l’évolution de l’accélération en fonction du nombre de coeurs. Cette fois-ci, 3 coeurs permettent de diviser le temps de calcul par 2,5.

La baisse de performance en utilisant plus de coeurs est due à la taille du maillage. Un maillage trop gros implique des communications trop fréquentes entre les coeurs, comme illustré dans la figure 2.14.

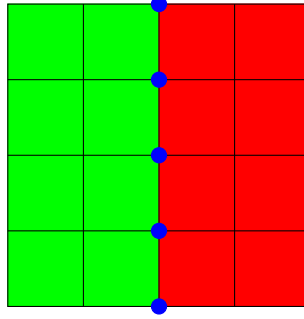


FIGURE 2.14 – Illustration d’un maillage partagé entre 2 coeurs. Les éléments verts sont traités par le premier coeur, les éléments rouges par le second.

Dans la figure 2.14, les noeuds bleus représentent les noeuds partagés entre les deux coeurs. Ces noeuds doivent être mis à jour à chaque itération, ce qui implique des communications fréquentes entre les coeurs. Avec cette taille d’éléments, il y a 5 noeuds partagés entre les deux coeurs sur un total de $5 \times 5 = 25$ noeuds. Cela fait 20% de noeuds partagés, ce qui est trop élevé. Pour améliorer les performances, il faut réduire la taille des éléments. La figure 2.15 montre un maillage avec des éléments plus petits.

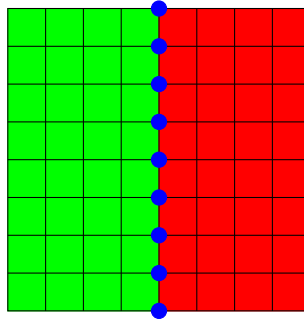


FIGURE 2.15 – Maillage avec des éléments plus petits partagé entre 2 coeurs. Il y a 9 noeuds partagés.

Cette fois-ci, il y a un total de 81 noeuds pour 9 noeuds partagés, ce qui représente environ 11% de noeuds partagés. Il y a moins de communications entre les coeurs par rapport à la taille du problème, ce qui améliore les performances comparé au même problème géré par un seul coeur.

2.5 Cas pratique

2.5.1 Retournement temporel en 2D

Pour tester la méthode SEM sur un cas pratique, j’ai utilisé un cas de retournement temporel en 2D. Il s’agit de la même plaque que dans la section 2.2.1. La phase d’apprentissage a déjà été calculé lors de la validation de la méthode SEM. La vidéo 2.16 montre la phase d’apprentissage sur un maillage fin de la plaque et des polynômes de degré 4.



FIGURE 2.16 – Phase d'apprentissage sur la plaque 2D

Le programme qui va faire les calculs, extrait également le déplacement dans les deux axes sur trois points de la plaque. L'emplacement de ces points est montré dans la figure 2.17.

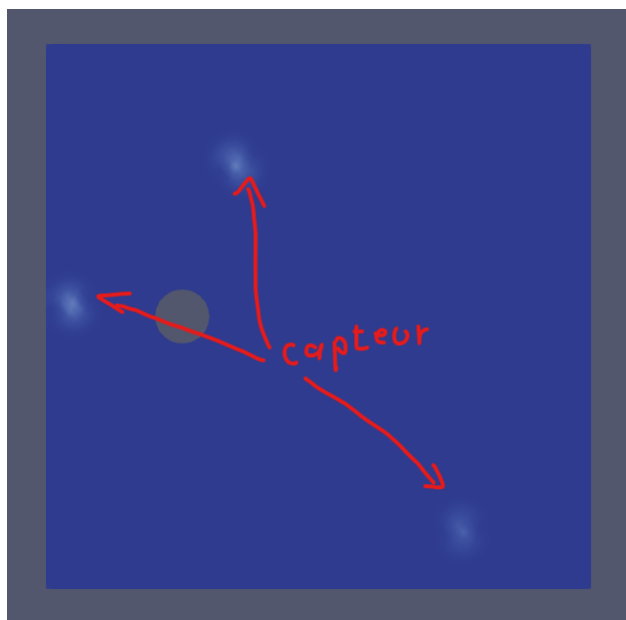


FIGURE 2.17 – Emplacement des capteurs

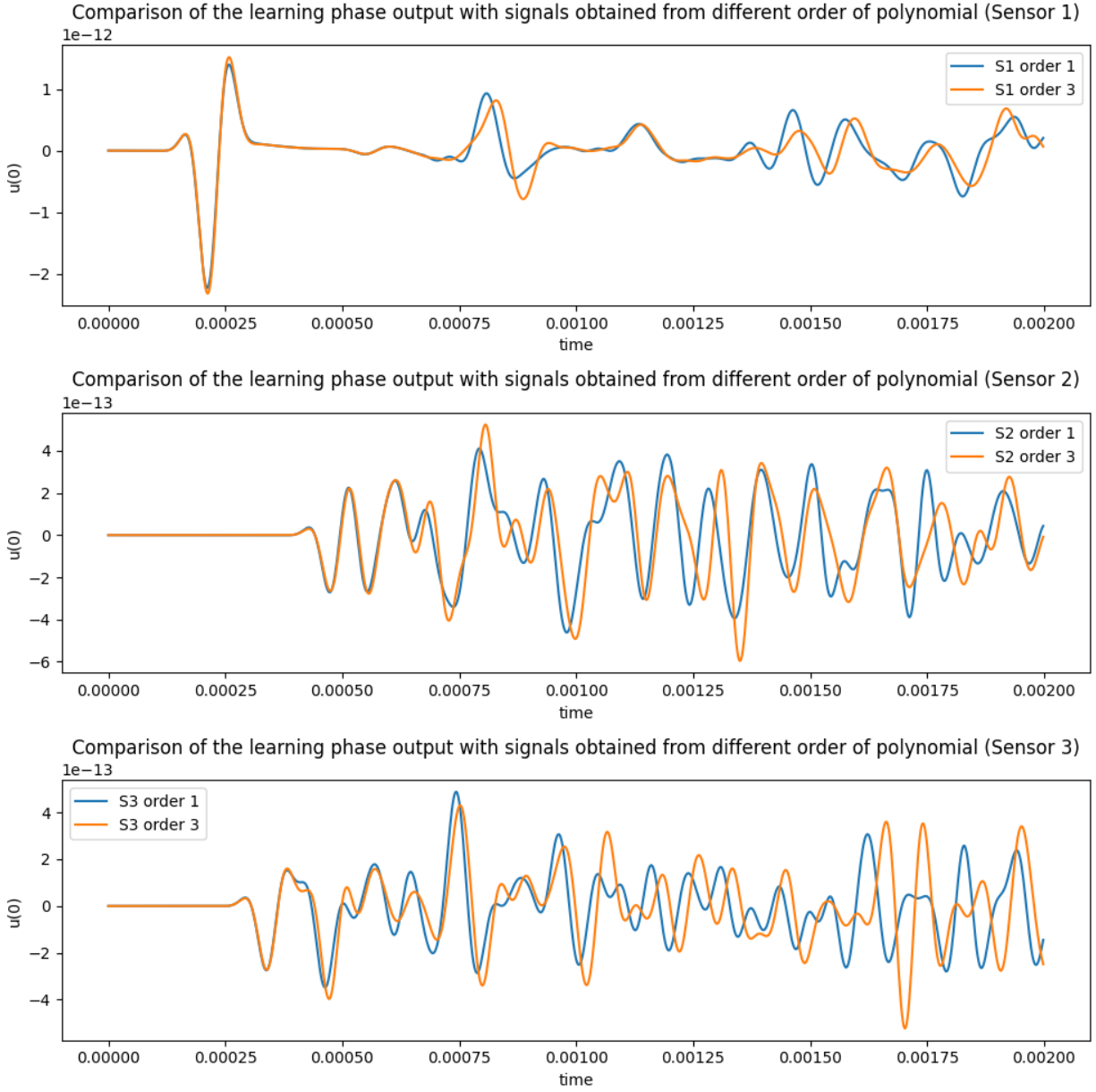


FIGURE 2.18 – Signaux obtenus par la méthode des éléments finis (En orange) et des éléments spectraux (en bleu)

La figure 2.18 montre les signaux obtenus par la méthode FEM (en orange) et la méthode SEM (en bleu). Il faut maintenant réinjecter les signaux enregistrés dans la plaque et observer la focalisation de l'onde sur le point d'origine. La vidéo 2.19 montre la focalisation de l'onde sur la plaque.

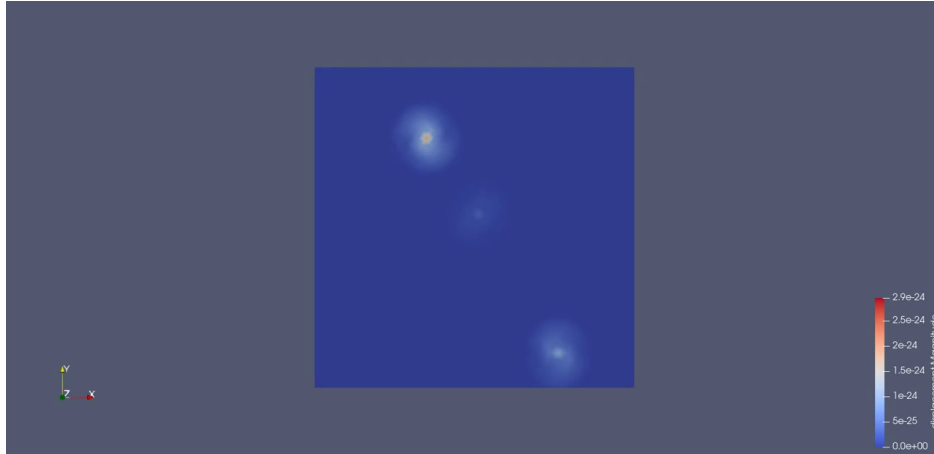


FIGURE 2.19 – Phase de focalisation sur la plaque 2D

Le déplacement du point d'origine est enregistré et montré dans la figure 2.20.

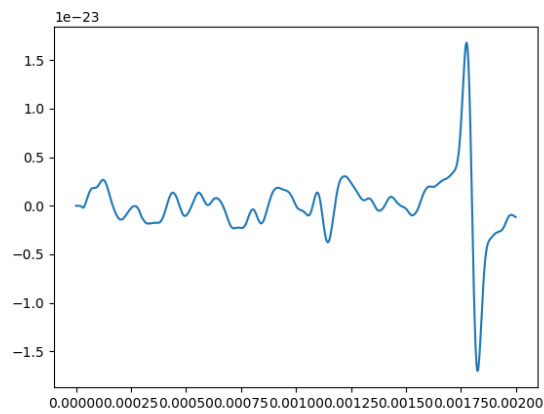


FIGURE 2.20 – Signal obtenu lors de la phase de focalisation

La vidéo et la figure montrent que l'onde se focalise sur le point d'origine au bout d'un moment puis se disperse. De plus, le pic attendu sur une structure saine est bien présent.

J'ai également lancé les calculs sur la même plaque mais avec un défaut. Celle-ci est affiché dans la figure 2.21.

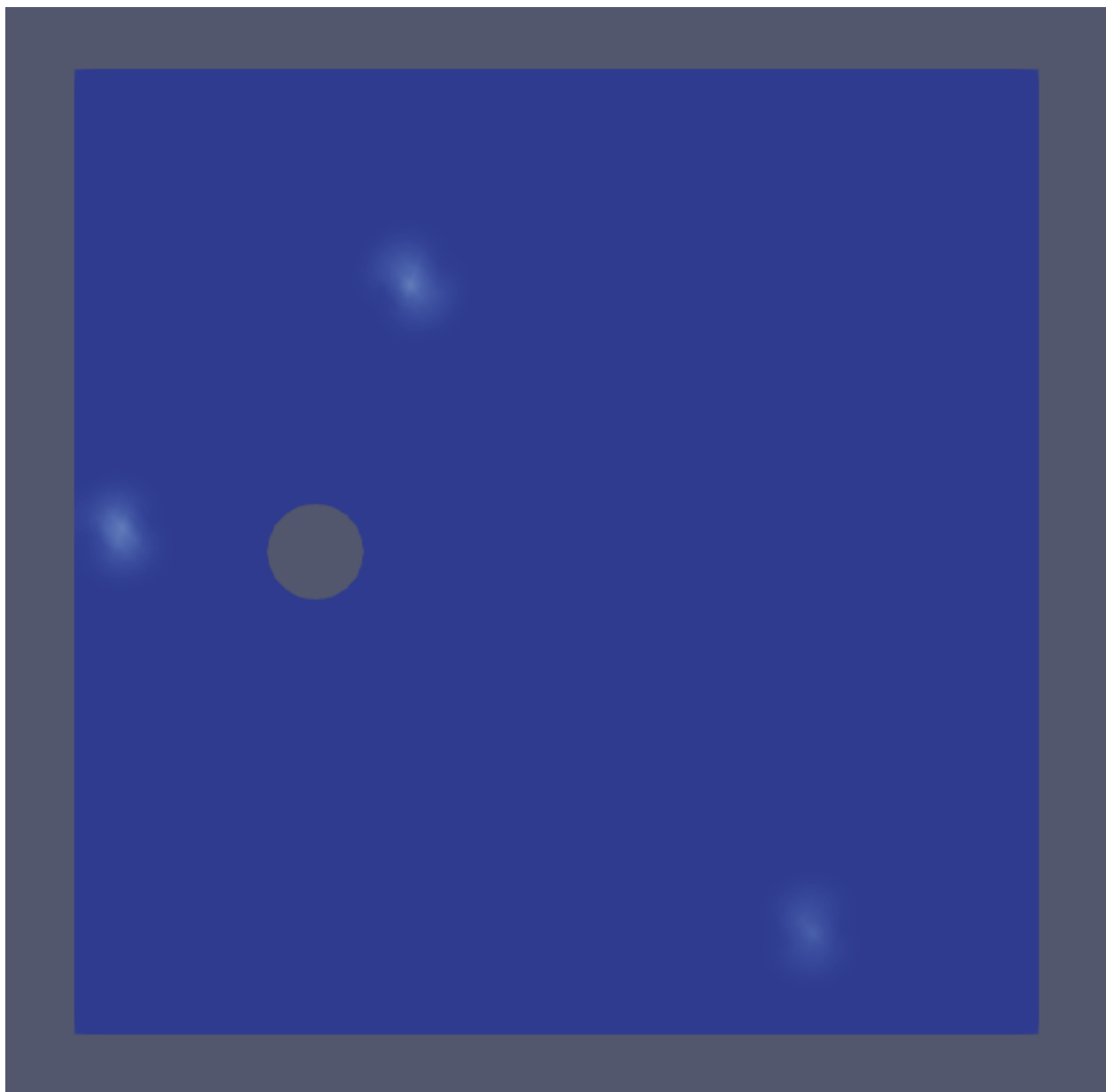


FIGURE 2.21 – Plaque 2D avec un défaut (Trou en cercle de centre $(-0.5;0)$ et de rayon 0.25)

Sur cette plaque, il y a trou en cercle de centre $(-0.5;0)$ et de rayon 0.25. La figure [2.22](#) montre la phase de focalisation sur cette plaque

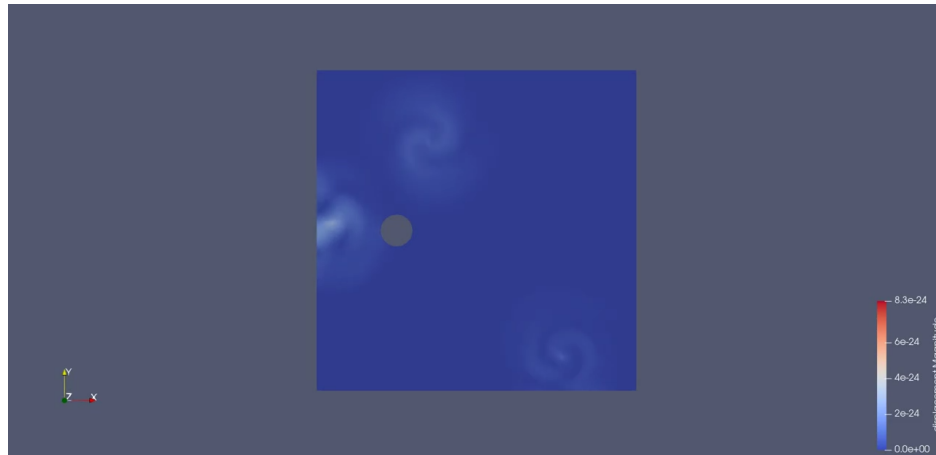


FIGURE 2.22 – Phase de focalisation sur la plaque 2D

Le signal enregistré est montré dans la figure 2.23.

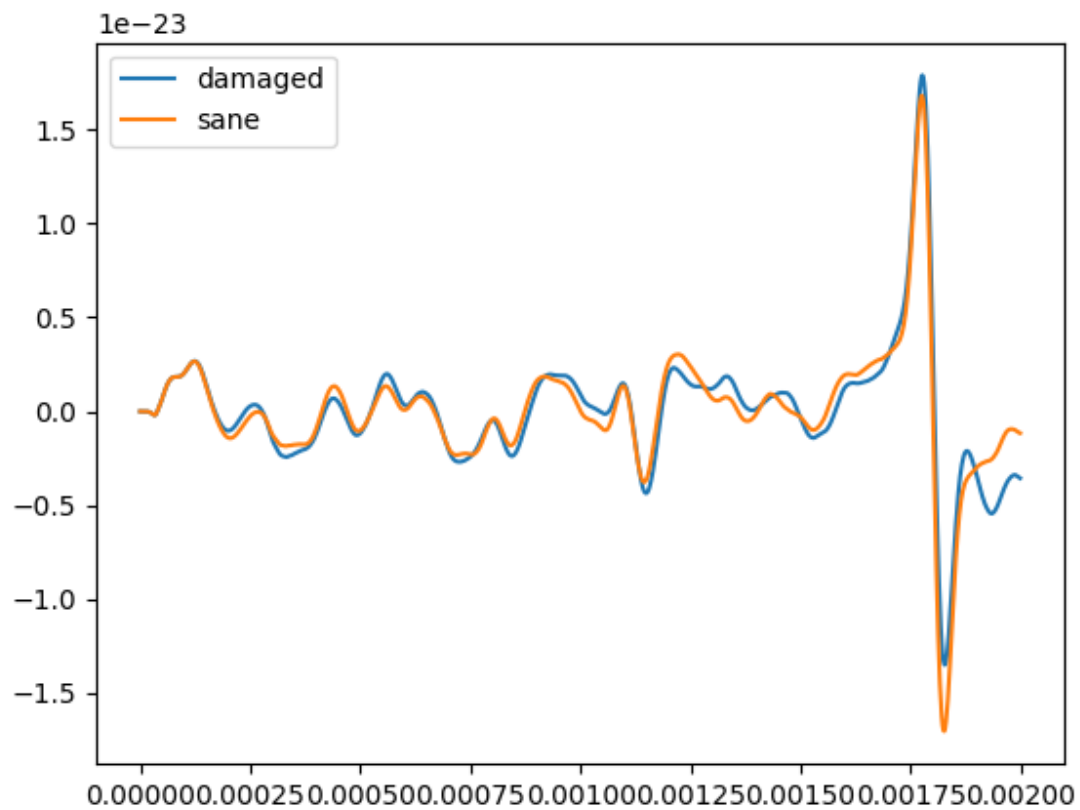


FIGURE 2.23 – Signal obtenu lors de la phase de focalisation sur la structure endommagé (en bleu) et la structure saine (en orange)

Le signal montre que l'onde ne se focalise pas sur le point d'origine, mais se disperse. De plus, l'amplitude est plus faible qu'attendu sur une structure saine.

2.5.2 Déformation d'une plaque 3D

Pour le cas 3D, j'ai utilisé une plaque trouée coupé en biais. Cette plaque provient d'une maquette de cuve de réacteur nucléaire. La figure 2.24 montre la plaque.

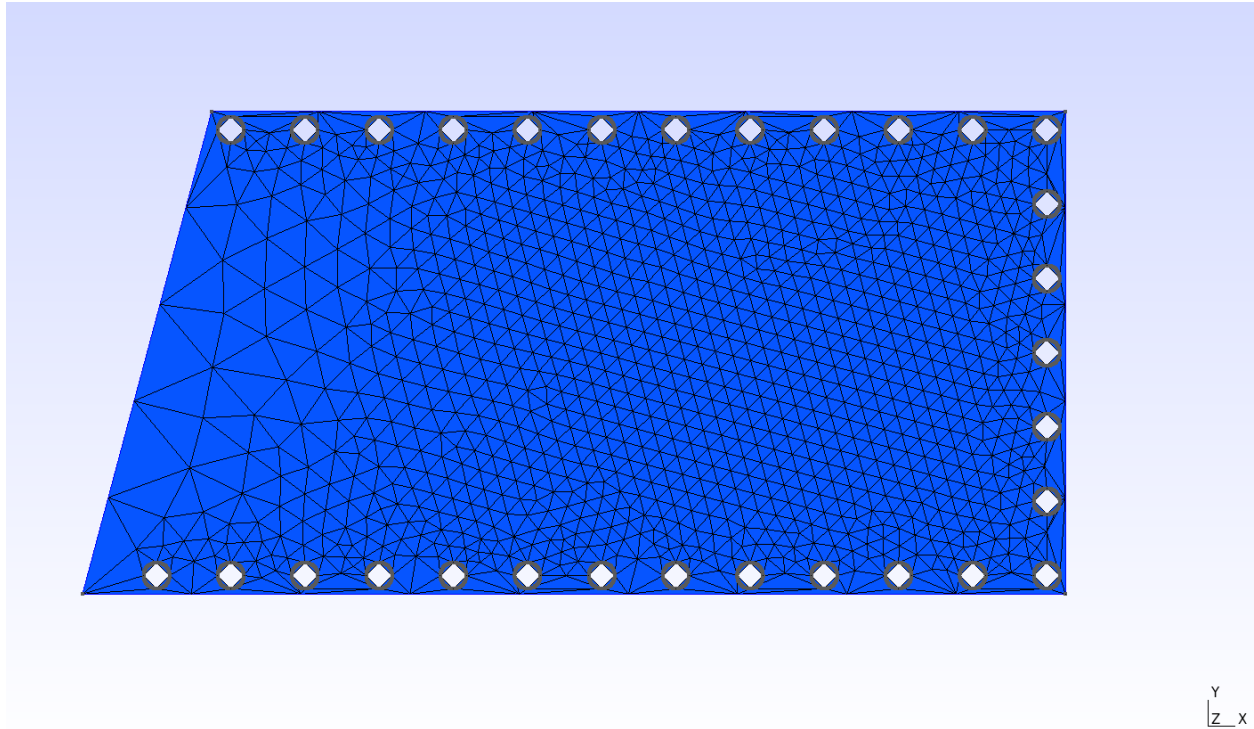


FIGURE 2.24 – Maillage de la plaque trouée sur Gmsh

Cependant, ce maillage n'est pas à l'échelle et présente des défauts lors de la génération de maillage. Un exemple de défaut est présenté dans la figure 2.25.

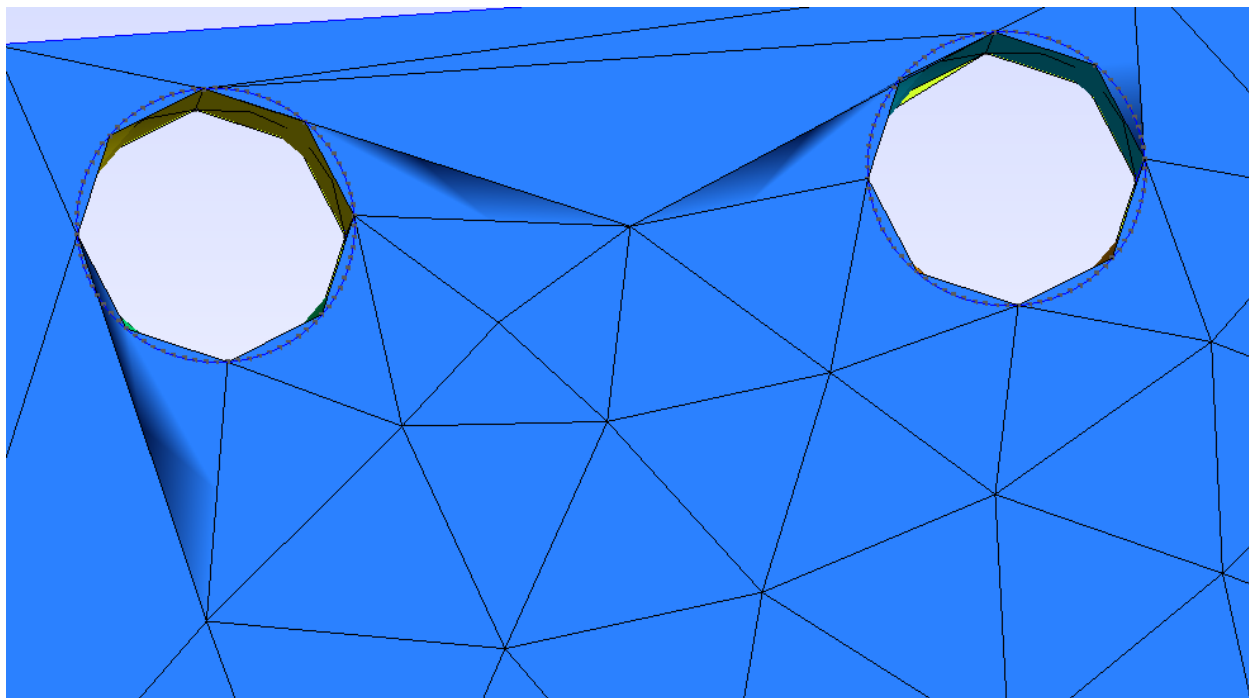


FIGURE 2.25 – Maillage contenant des défauts. Les faces grises montrent des faces qui se coupent entre elles.

J'ai donc du refaire le maillage de la plaque. La figure 2.26 montre le maillage de la plaque.

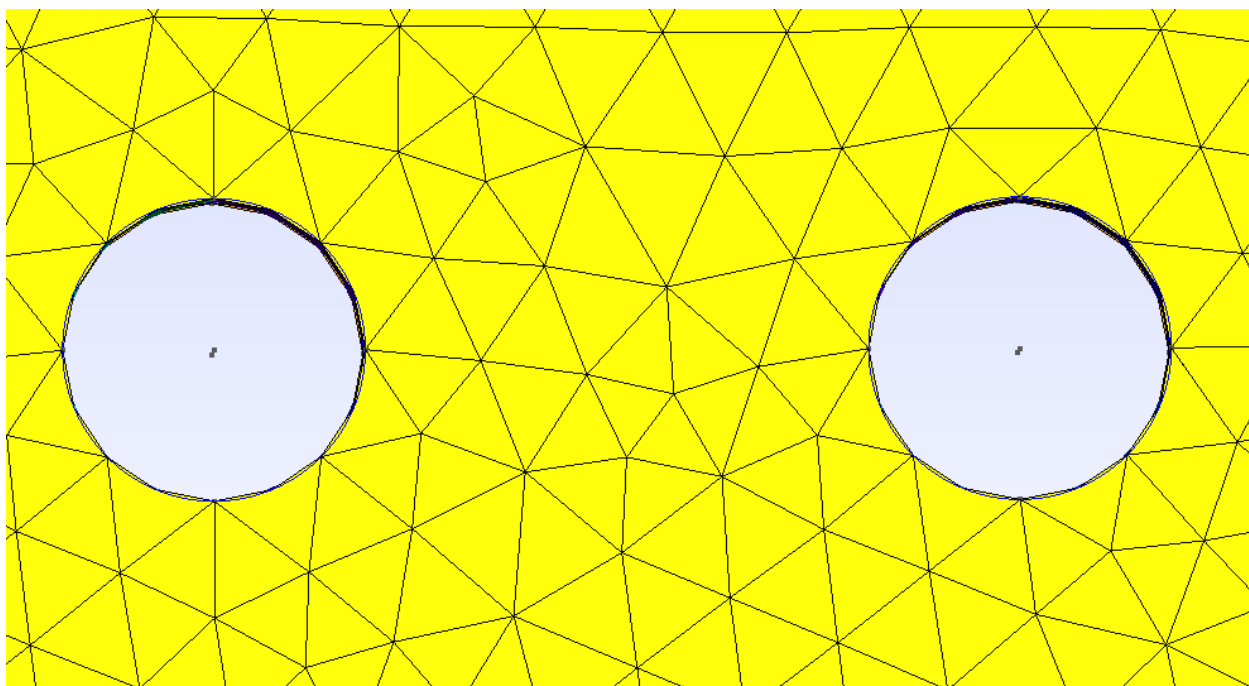


FIGURE 2.26 – Maillage avec les défauts supprimés.

Pour l'emplacement des capteurs et de la source, j'ai repris les emplacements utilisés lors d'expérience

sur la vraie plaque comme expliqué dans la figure 2.27.

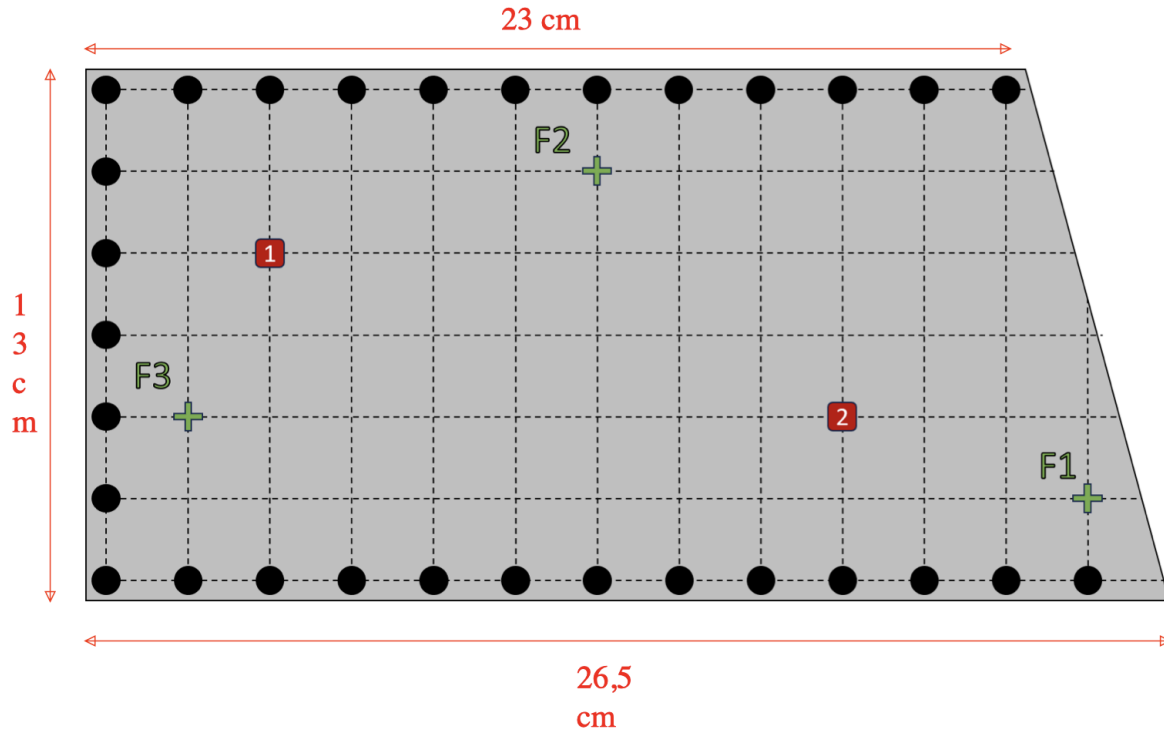


FIGURE 2.27 – Emplacement des capteurs et des points de chocs

Sur la figure 2.27, les points rouges désignent l'emplacement des capteurs (le capteur 1 est placé à $(-0.185, 0.045, -0.0015)$ et le capteur 2 est placé à $(-0.045, 0.085, -0.0015)$), et les points F désignent les points de choc. F1 est à $(-0.245, 0.025, -0.0015)$, F2 est à $(-0.125, 0.105, -0.0015)$ et F3 est à $(-0.025, 0.045, -0.0015)$

La vidéo 2.28 montre la phase d'apprentissage sur la plaque 3D.

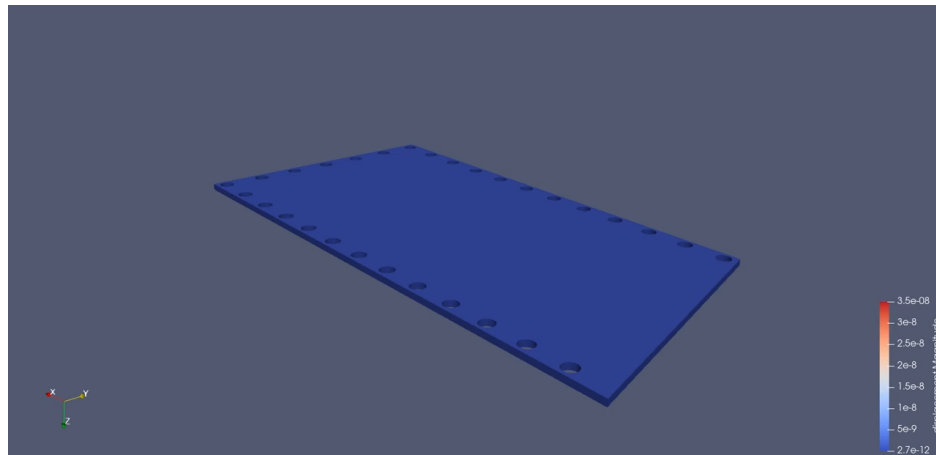


FIGURE 2.28 – Phase d'apprentissage sur la plaque trouée

Le tableau 2.11 montre les différents paramètres utilisés pour obtenir le résultat 2.28

Paramètres	Valeurs
Taille des éléments	0.001 m
Nombre d'éléments	14153
Nombre de noeuds	654512
Nombre de degré de liberté	6516
E	210000000000 Pa
ν	0.3
ρ	7800 kg/m^3
Préconditionneur	Jacobi
Emplacement du choc	F1
Degré des polynômes	3

TABLE 2.10 – Paramètres utilisés pour obtenir le résultat de 2.28

Paramètres	Valeurs
Taille des éléments	0.002 m
Nombre d'éléments	14153
Nombre de noeuds	654512
Nombre de degré de liberté	1588389
E	210000000000 Pa
ν	0.3
ρ	7800 kg/m^3
Préconditionneur	Jacobi
Emplacement du choc	F1
Degré des polynômes	3

TABLE 2.11 – Paramètres utilisés pour obtenir le résultat de 2.29

Enfin, j'ai lancé les calculs sur une plaque cylindrique simplement supportée. Les résultats obtenus sont montrés dans la vidéo 2.29.

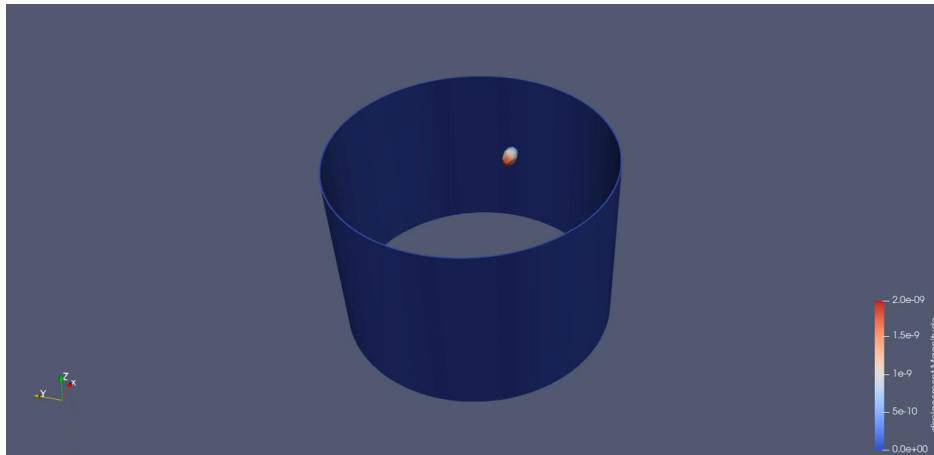


FIGURE 2.29 – Phase d'apprentissage sur la plaque cylindrique

Troisième partie

Conclusion

2.6 Travail à venir

Malgré ce travail, il reste encore des choses à faire. D’abord, il faut trouver une solution analytique pour la plaque cylindrique. Cette solution devrait être trouvée avant la fin du stage. Ensuite, il faut améliorer les performances des cas 3D. Le passage au cas 3D implique un plus grand nombre d’éléments et donc un calcul plus coûteux. L’idée serait de passer à un cas coque, c’est-à-dire de négliger l’épaisseur de la plaque 3D, de façon à avoir un maillage 2D capable de se mouvoir en 3D. Enfin, pour détecter des défauts sur un jumeau numérique de cuve de réacteur nucléaire, il faut prendre en compte la présence de liquide et de différents matériaux. Il faut donc ajouter un couplage fluide-structure et un couplage multi-matériaux dans le solveur.

Comme il me reste encore 1 mois avant la fin de mon stage, un collègue d’Avnir Energy va reprendre mon travail. Et durant le mois de Septembre, je vais l’aider à prendre en main le code et à utiliser la librairie Feel++ pour qu’il puisse continuer le travail.

2.7 Bilan

Pour conclure, ce stage m’a permis de découvrir le monde de la recherche et de l’entreprise. J’ai pu mettre en pratique mes connaissances en mathématiques et en informatique. J’ai aussi appris à travailler en équipe et à communiquer avec des personnes de différents horizons. J’ai aussi appris à gérer mon temps et à m’organiser pour mener à bien un projet. Enfin, j’ai pu découvrir le monde de l’énergie et de l’industrie nucléaire.

Quant à la méthode par élément spectraux, j’ai pu découvrir une méthode numérique que je n’avais pas encore implémentée. Elle a montré des résultats prometteurs en 2D dans la précision des résultats et la rapidité des calculs. Cependant, en 3D, la méthode a montré ses limites en terme de rapidité sur lesquelles il faudra travailler notamment en implémentant le cas coque.

2.8 Remerciements

Je tiens à remercier Thomas Le Gall, mon tuteur de stage et directeur R&D d’Avnir Energy pour m’avoir accompagné tout au long de ce stage malgré 2 mois d’absence du à son congé paternité. Je remercie également Nicolas Ponthus et Pascal Vouagner pour avoir encadré mon stage à la place de Thomas durant son congé et qui m’ont aidé sur la partie physique du projet. Je remercie également Jacques Charvin, président d’Avnir Energy pour m’avoir accueilli dans son entreprise. Enfin, je remercie Christophe Prud’homme, mon maître de stage pour m’avoir aidé à comprendre la méthode SEM, prendre en main le code existant et nous avoir accueilli dans le CEMOSIS.

Bibliographie

- [1] Cemosis. Solving linear elasticity using finite element method, 2024.
- [2] S. Fauqueux. Éléments finis mixtes spectraux et couches absorbantes parfaitement adaptées pour la propagation d'ondes élastiques en régime transitoire. *ENSTA ParisTech*, 2003.
- [3] J.-L. Guyader. *Vibrations des milieux continus*. Hermès-Lavoisier, 2002.
- [4] S. Krenk. Energy conservation in newmark based time integration algorithms. *HAL Science*, 2005.
- [5] Wikipedia. Méthodes de newmark.
- [6] Wikipedia. Méthodes de runge-kutta.

Annexe A

Calcul de la solution analytique

Afin d'évaluer les résultats de méthodes numériques d'ordre supérieur, on se propose de réaliser le calcul sur des objets pour lesquels la solution analytique existe. Les objets choisis sont au nombre de deux :

- Une plaque simplement supportée sur ses quatre bords
- Un cylindre creux simplement supporté sur ses deux extrémités

A.1 Plaque

Les résultats présentés ici sont dérivés des sections 7 et 9 de [3]. Soit une plaque de dimension a selon l'axe 1 et b selon l'axe 2. Les coordonnées selon ces deux axes seront notées x_1 et x_2 . Cette plaque est d'épaisseur h . Ses caractéristiques matériaux sont uniformes avec une masse volumique ρ et de module de Young E . On introduit également un coefficient de Poisson ν . On pose également $I = h^3/12$. On s'intéresse ici exclusivement aux mouvements de la plaque dans sa direction normale. On note $W(x_1, x_2, t)$ le déplacement dans cette direction.

A.1.1 Équation mécanique de la plaque

Selon les hypothèses de Love-Kirchhoff l'équation du mouvement de la plaque s'écrit :

$$-\rho h \frac{\partial^2 W}{\partial t^2} + \rho I \left(\frac{\partial^4 W}{\partial x_1^2 \partial t^2} + \frac{\partial^4 W}{\partial x_2^2 \partial t^2} \right) - D \left(\frac{\partial^4 W}{\partial x_1^4} + \frac{\partial^4 W}{\partial x_1^2 \partial x_2^2} + \frac{\partial^4 W}{\partial x_2^4} \right) = -q(x_1, x_2, t) \quad (\text{A.1})$$

avec $D = EI/(1 - \nu^2)$. Comme l'auteur de [3], on choisit d'ignorer l'inertie en rotation :

$$\rho h \frac{\partial^2 W}{\partial t^2} + D \left(\frac{\partial^4 W}{\partial x_1^4} + \frac{\partial^4 W}{\partial x_1^2 \partial x_2^2} + \frac{\partial^4 W}{\partial x_2^4} \right) = q(x_1, x_2, t) \quad (\text{A.2})$$

On écrit les conditions aux limites (simplement supporté) pour les quatre bords de la plaque :

$$\left\{ \begin{array}{l} W(x_1, 0, t) = 0 \\ \frac{\partial^2 W}{\partial x_2^2}(x_1, 0, t) = 0 \\ W(x_1, b, t) = 0 \\ \frac{\partial^2 W}{\partial x_2^2}(x_1, b, t) = 0 \\ W(0, x_2, t) = 0 \\ \frac{\partial^2 W}{\partial x_1^2}(0, x_2, t) = 0 \\ W(a, x_2, t) = 0 \\ \frac{\partial^2 W}{\partial x_1^2}(a, x_2, t) = 0 \end{array} \right. \quad (\text{A.3})$$

A.1.2 Schema modal

Réponse harmonique

Le cas de la plaque appuyée sur ses quatre bords possède une solution analytique car la séparation de variable offre de satisfaire à la fois à l'équation de mouvement et aux conditions aux limites. La résolution de l'équation homogène (A.2 avec le second membre nul) permet de calculer le schéma modal suivant :

$$\omega_{n,m} = \sqrt{\frac{D}{\rho h} \left(\left(\frac{n\pi}{a} \right)^2 + \left(\frac{m\pi}{b} \right)^2 \right)} \quad (\text{A.4})$$

La déformée modale associée est donnée par (normalisation à 1) :

$$f_{n,m}(x_1, x_2) = \sin\left(\frac{n\pi}{a}x_1\right) \sin\left(\frac{m\pi}{b}x_2\right) \quad (\text{A.5})$$

La réponse harmonique est donc donnée par :

$$W_h(x_1, x_2, t) = [A_{n,m} \cos(\omega_{n,m}t) + B_{n,m} \sin(\omega_{n,m}t)] f_{n,m}(x_1, x_2) \quad (\text{A.6})$$

Orthogonalité des déformées propres

On définit les deux opérateurs suivants :

$$M(f_{n,m}, f_{p,k}) = \int_{x_1=0}^a \int_{x_2=0}^b \rho h f_{n,m}(x_1, x_2) f_{p,k}(x_1, x_2) dx_1 dx_2 \quad (\text{A.7})$$

$$K(f_{n,m}, f_{p,k}) = \int_{x_1=0}^a \int_{x_2=0}^b D \left(\frac{\partial^4 f_{n,m}}{\partial x_1^4} + \frac{\partial^4 f_{n,m}}{\partial x_1^2 \partial x_2^2} + \frac{\partial^4 f_{n,m}}{\partial x_2^4} \right) f_{p,k} dx_1 dx_2 \quad (\text{A.8})$$

On peut vérifier qu'il s'agit là de produit scalaire : la linéarité et la positivité sont immédiates. La symétrie de M est triviale. Pour la symétrie de K , on procède à la manière de [3] par intégration par parties successives faisant intervenir les conditions aux limites. Montrons l'orthogonalité des déformées propres de l'équation

A.5 :

$$\int_{x_1=0}^a \int_{x_2=0}^b \rho h f_{n,m}(x_1, x_2) f_{p,k}(x_1, x_2) dx_1 dx_2 \quad (\text{A.9})$$

$$= \rho h \int_{x_1=0}^a \int_{x_2=0}^b \sin\left(\frac{n\pi}{a}x_1\right) \sin\left(\frac{m\pi}{b}x_2\right) \sin\left(\frac{p\pi}{a}x_1\right) \sin\left(\frac{k\pi}{b}x_2\right) dx_1 dx_2 \quad (\text{A.10})$$

$$= \rho h \int_{x_1=0}^a \sin\left(\frac{n\pi}{a}x_1\right) \sin\left(\frac{p\pi}{a}x_1\right) \int_{x_2=0}^b \sin\left(\frac{m\pi}{b}x_2\right) \sin\left(\frac{k\pi}{b}x_2\right) dx_2 dx_1 \quad (\text{A.11})$$

L'orthogonalité de la famille $\{\sin(n\pi x/l) | n \in \mathbb{N}\}$ permet de démontrer que si $n \neq p$ ou $m \neq k$ alors la norme $M(f_{n,m}, f_{p,k})$ est nulle. On définit :

$$M_{n,m} = \int_{x_1=0}^a \int_{x_2=0}^b \rho h f_{n,m}(x_1, x_2)^2 dx_1 dx_2 \quad (\text{A.12})$$

$$\begin{aligned} &= \rho h \int_{x_1=0}^a \int_{x_2=0}^b \sin^2\left(\frac{n\pi}{a}x_1\right) \sin^2\left(\frac{m\pi}{b}x_2\right) dx_1 dx_2 \\ &= \rho h \int_{x_1=0}^a \sin^2\left(\frac{n\pi}{a}x_1\right) \int_{x_2=0}^b \sin^2\left(\frac{m\pi}{b}x_2\right) dx_2 dx_1 \\ &= \rho h \int_{x_1=0}^a \sin^2\left(\frac{n\pi}{a}x_1\right) \int_{x_2=0}^b \frac{1 - \cos\left(\frac{2m\pi}{b}x_2\right)}{2} dx_2 dx_1 \\ &= \rho h \int_{x_1=0}^a \sin^2\left(\frac{n\pi}{a}x_1\right) \left[\frac{x_2}{2} - \frac{\sin\left(\frac{2m\pi}{b}x_2\right)}{m\pi} \right]_{x_2=0}^b dx_1 \\ &= \rho h \frac{b}{2} \int_{x_1=0}^a \sin^2\left(\frac{n\pi}{a}x_1\right) dx_1 \\ M_{n,m} &= \rho h \frac{a}{2} \frac{b}{2} = \rho \frac{hab}{4} \end{aligned} \quad (\text{A.13})$$

En injectant A.6 dans l'équation homogène, on obtient :

$$\begin{aligned} &-\omega_{n,m}^2 \rho h f_{n,m} + D \left(\frac{\partial^4 f_{n,m}}{\partial x_1^4} + \frac{\partial^4 f_{n,m}}{\partial x_1^2 \partial x_2^2} + \frac{\partial^4 f_{n,m}}{\partial x_2^4} \right) = 0 \\ &-\omega_{n,m}^2 \rho h f_{n,m} f_{p,k} + D \left(\frac{\partial^4 f_{n,m}}{\partial x_1^4} + \frac{\partial^4 f_{n,m}}{\partial x_1^2 \partial x_2^2} + \frac{\partial^4 f_{n,m}}{\partial x_2^4} \right) f_{p,k} = 0 \\ &-\omega_{n,m}^2 \int_{x_1=0}^a \int_{x_2=0}^b \rho h f_{n,m} f_{p,k} dx_1 dx_2 + \\ &\int_{x_1=0}^a \int_{x_2=0}^b D \left(\frac{\partial^4 f_{n,m}}{\partial x_1^4} + \frac{\partial^4 f_{n,m}}{\partial x_1^2 \partial x_2^2} + \frac{\partial^4 f_{n,m}}{\partial x_2^4} \right) f_{p,k} dx_1 dx_2 = 0 \\ &-\omega_{n,m}^2 M(f_{n,m}, f_{p,k}) + K(f_{n,m}, f_{p,k}) = 0 \end{aligned} \quad (\text{A.14})$$

On montre ainsi l'orthogonalité des déformées propres vis à vis du produit scalaire K . On définit $K_{n,m} = K(f_{n,m}, f_{n,m})$ et A.14 permet de montrer que :

$$\omega_{m,n} = \sqrt{\frac{K_{n,m}}{M_{n,m}}} \quad (\text{A.15})$$

Dont on peut également déduire l'expression de $K_{n,m}$

A.1.3 Résolution de l'équation forcée

Afin de résoudre l'équation A.2, on écrit que la solution peut s'écrire sous la forme d'une décomposition modale telle que :

$$W(x_1, x_2, t) = \sum_{n=1}^{+\infty} \sum_{m=1}^{+\infty} a_{n,m}(t) f_{n,m}(x_1, x_2) \quad (\text{A.16})$$

En injectant cette forme dans le l'équation du mouvement on obtient :

$$\begin{aligned} \rho h \sum_{n=1}^{+\infty} \sum_{m=1}^{+\infty} \ddot{a}_{n,m}(t) f_{n,m}(x_1, x_2) + \\ D \sum_{n=1}^{+\infty} \sum_{m=1}^{+\infty} a_{n,m}(t) \left(\frac{\partial^4 f_{n,m}}{\partial x_1^4} + \frac{\partial^4 f_{n,m}}{\partial x_1^2 \partial x_2^2} + \frac{\partial^4 f_{n,m}}{\partial x_2^4} \right) (x_1, x_2) = p(x_1, x_2, t) \end{aligned} \quad (\text{A.17})$$

en multipliant par $f_{p,k}(x_1, x_2)$ et en intégrant par rapport à x_1 et x_2 . On utilise l'orthogonalité des modes pour obtenir :

$$\ddot{a}_{p,k}(t) M_{p,k} + a_{p,k} K_{p,k} = \int_{x_1=0}^a \int_{x_2=0}^b p(x_1, x_2, t) f_{p,k}(x_1, x_2) dx_1 dx_2 \quad (\text{A.18})$$

On écrit que $p(x_1, x_2, t)$ est une force ponctuelle en un point (X_1, X_2) :

$$p(x_1, x_2, t) = \tilde{p}(t) \delta(X_1, X_2) \quad (\text{A.19})$$

ou δ est la fonction dirac. On défini :

$$F_{p,k}(t) = \int_{x_1=0}^a \int_{x_2=0}^b p(x_1, x_2, t) f_{p,k}(x_1, x_2) dx_1 dx_2 \quad (\text{A.20})$$

$$F_{p,k}(t) = \int_{x_1=0}^a \int_{x_2=0}^b \tilde{p}(t) \delta(X_1, X_2) f_{p,k}(x_1, x_2) dx_1 dx_2 \quad (\text{A.21})$$

$$F_{p,k}(t) = \tilde{p}(t) f_{p,k}(X_1, X_2) \quad (\text{A.22})$$

Le problème revient donc à traiter pour chaque couple (p, k) l'équation différentielle du second ordre à coefficient constant :

$$\ddot{a}_{p,k}(t) M_{p,k} + a_{p,k} K_{p,k} = \tilde{p}(t) f_{p,k}(X_1, X_2) \quad (\text{A.23})$$

ou encore :

$$\ddot{a}_{p,k}(t) + a_{p,k} \omega_{p,k}^2 = \frac{\tilde{p}(t) f_{p,k}(X_1, X_2)}{M_{p,k}} \quad (\text{A.24})$$

Afin de résoudre ce système, il convient de rajouter deux conditions initiales :

$$a_{p,k}(0) = \frac{\int_{x_1=0}^a \int_{x_2=0}^b \rho h f_{p,k}(x_1, x_2) d_0(x_1, x_2) dx_1 dx_2}{M_{p,k}} \quad (\text{A.25})$$

$$\dot{a}_{p,k}(0) = \frac{\int_{x_1=0}^a \int_{x_2=0}^b \rho h f_{p,k}(x_1, x_2) v_0(x_1, x_2) dx_1 dx_2}{M_{p,k}} \quad (\text{A.26})$$

où d_0 et v_0 sont respectivement les champs de déplacement et de vitesse à $t = 0$. En prenant une plaque non déformée et au repos à $t = 0$ il vient :

$$a_{p,k}(0) = 0 \quad (\text{A.27})$$

$$\dot{a}_{p,k}(0) = 0 \quad (\text{A.28})$$

On défini également l'ondelette suivante :

$$\tilde{p}(t) = \sin(2\pi Ft) e^{-5(Ft-2)^2} \quad (\text{A.29})$$

Cette équation différentielle pourra être résolue soit par méthode numérique classique (ode45) soit par la méthode de la variation de la constante.

A.1.4 estimation de l'erreur

La résolution de l'équation [A.24](#) ne pourra se faire que pour un nombre fini mode, c'est à dire de valeur (n, m) . Néanmoins il est possible d'estimer l'erreur produite par cette troncature (N, M) . On défini :

$$W_{N,M}(x_1, x_2, t) = \sum_{n=1}^N \sum_{m=1}^M a_{n,m}(t) f_{n,m}(x_1, x_2) \quad (\text{A.30})$$

on peut écrire

$$e_{N,M}(x_1, x_2, t) = \frac{\rho h \ddot{W}_{N,M}(x_1, x_2, t) + D \left(\frac{\partial^4 W_{N,M}}{\partial x_1^4} + \frac{\partial^4 W_{N,M}}{\partial x_1^2 \partial x_2^2} + \frac{\partial^4 W_{N,M}}{\partial x_2^4} \right) (x_1, x_2, t) - p(x_1, x_2, t)}{p(x_1, x_2, t)} \quad (\text{A.31})$$

Sur les points (x_1, x_2) de la solution rechercher, avec une discrétisation temporelle donnée, on pourra par exemple calculer la valeur rms de cette erreur.