

Université de Strasbourg

Rapport de Stage de Master 2 CSMI

Simulation et benchmarking de méthodes numériques de propagation d'ondes sur la paroi d'une maquette de cuve nucléaire

Rayen Tlili

Encadrant en entreprise : Thomas Legall

Encadrant à l'université : Christophe Prud'homme

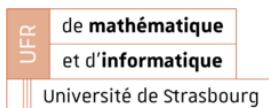


Table des matières

1 Présentation de l'entreprise	3
1.1 Historique	3
1.2 Organisation interne	3
1.3 Clients et collaborateurs	3
2 Présentation du projet	4
2.1 Contexte scientifique	4
2.1.1 Le retourneur temporel	4
2.1.2 Problématiques industrielles	4
2.2 Le projet SINUDYN	5
3 Objectifs du stage	8
4 Méthodologie	9
4.1 Outils utilisés	9
4.2 Méthodes numériques	10
4.2.1 Limites des méthodes numériques classiques pour la propagation d'ondes	10
4.2.2 Méthodes des éléments finis (FEM) et des éléments spectraux (SEM)	10
4.2.3 Stratégies d'intégration temporelle	12
4.2.4 Couplage fluide-structure	13
4.3 Modèle physique	13
4.3.1 Domaines et Notations	14
4.3.2 Équations gouvernantes	14
4.3.3 Conditions de couplage sur l'interface Γ_{fsi}	14
4.3.4 Conditions initiales et aux limites	14
4.3.5 Modélisation de l'amortissement dans le solide	15
4.4 Modèle géométrique	15
4.4.1 Description de la cuve	15
4.4.2 Stratégie de maillage avec Gmsh	15
4.4.3 Caractéristiques des maillages	17
4.5 Mesures expérimentales sur la cuve	18
4.5.1 Mise en place expérimentale	18
4.5.2 Détection des défauts : description du protocole	18
4.5.3 Choix des surfaces et points d'intérêt	20
5 Implémentation avec FEEL++	22
5.1 Utilisation des toolboxes existantes	22
5.1.1 Toolbox Solid Computational Mechanics	22
5.1.2 Post-traitement et monitoring	25
5.1.3 Passage à des ordres de discréétisation plus élevés	25
5.2 Développement d'une méthode de résolution et de couplage explicite pour le problème de vibro-acoustique	25
5.2.1 Objectifs	25
5.2.2 Formulation variationnelle	25
5.2.3 Discréétisation temporelle et schéma numérique	26
5.2.4 Algorithme : Avancement en temps pour le problème vibro-acoustique couplé	27
5.3 Ressources de calcul	28
5.3.1 Supercalculateur Gaya	28
5.3.2 Parallélisation et usage mémoire	29

6 Présentation des résultats	30
6.1 Mesures expérimentales de la phase d'apprentissage - Interprétation des signaux	30
6.2 Méthodes de validation	31
6.3 Résultats numériques	32
6.3.1 Résultats des simulations avec les toolboxes	32
6.3.2 Évaluation de la précision des simulations	33
6.3.3 Comparaison avec les mesures expérimentales	38
6.3.4 Résultats du code de couplage vibro-acoustique	41
6.4 Benchmarking	42
6.4.1 Benchmarking des simulations avec la Toolbox mécanique des solides	43
6.4.2 Benchmarking des simulations de couplage vibro-acoustique - élasticité	45
7 Conclusions et perspectives	52
7.1 Perspectives	52
7.2 Conclusions	52
7.3 Annexe	53
7.3.1 Dérivation de l'équation d'onde acoustique dans le fluide	53
7.3.2 Estimation de l'ordre de grandeur du déplacement et de l'accélération dans la cuve sous une excitation maximale de 1N	55
7.3.3 Trouver un pas de temps idéal	56
7.3.4 Estimation du temps final pour observer la propagation des ondes dans la cuve	57
7.3.5 Algorithme explicite pour la résolution du problème de vibro-acoustique transitoire	58
7.3.6 Amortissement (solide)	60
Bibliographie	61

Chapitre 1

Présentation de l'entreprise

1.1 Historique



Fondée en 2015, **AVNIR Energy** est une société d'ingénierie pour les domaines des énergies nucléaire, pétrolière, gazière, hydraulique, thermique, renouvelables et du secteur du génie civil.

L'entreprise propose des services de surveillance des vibrations (vibrations, acoustique, ultrasons, émission acoustique, rayonnements ionisants) et de détection sismique afin de garantir la sécurité et l'entretien des infrastructures.

1.2 Organisation interne

L'entreprise compte aujourd'hui plus de **30 employés** et est structurée autour de trois pôles principaux :

- **Le pôle R&D**, en charge du développement de méthodes de simulation et expérimentation avancées ;
- **Le pôle opérationnel**, bureau d'études et de calculs dédié à la réalisation des projets pour les clients industriels ;
- **Le pôle commercial**, en lien direct avec les besoins du marché et le développement des partenariats.

L'entreprise **AVNIR**, intervenant dans le domaine du nucléaire, est soumise à des exigences réglementaires strictes, notamment la certification **ISO 19443** relative à la qualité et à la sûreté nucléaire. Dans ce cadre, des causeries **QSSSE** (**Qualité, Sûreté, Santé, Sécurité, Environnement**) sont organisées tous les trois mois afin de rappeler les procédures internes et les mesures de prévention. Par ailleurs, chaque collaborateur suit une formation obligatoire à la sécurité numérique via le dispositif en ligne **SecNumAcadémie**.

Les locaux d'**AVNIR Energy** sont situés au Centre d'Entreprises et d'Innovation (CEI) sur le campus Lyon-Tech la Doua, à Villeurbanne.

1.3 Clients et collaborateurs

On retrouve plus de 200 clients depuis la création. Parmi ses **principaux clients et partenaires** :

- **SYSCOM Instruments**(les équipements de surveillance vibratoire et de détection sismique)
- **EDF** (énergie nucléaire et hydraulique),
- **CEA** (recherche appliquée en ingénierie nucléaire),
- **Sonorhc** (acoustique et contrôle non destructif),
- Des laboratoires académiques comme l'**INSA de Lyon**, l'**École Centrale de Lyon**, et le **CEMOSIS** (Université de Strasbourg).

La collaboration entre Avnir Energy et le programme de Master CSMI se poursuit depuis près de deux ans. Elle vise à étudier et à simuler les ondes acoustiques dans une cuve de réacteur nucléaire remplie de fluide (liquide ou gazeux). L'objectif principal est de détecter d'éventuelles anomalies dans la structure de la cuve, afin de garantir l'intégrité et le suivi continu de l'infrastructure.

Chapitre 2

Présentation du projet

2.1 Contexte scientifique

2.1.1 Le retournement temporel

Le retournement temporel est un procédé physique connu dans plusieurs domaines du spectre électromagnétique (lumière visible, micro-ondes, etc.) et a été adapté aux ondes ultrasonores par Mathias Fink à l'École supérieure de physique et de chimie industrielles (ESPCI) de Paris dans les années 80. Ce principe permet à une onde de retourner vers sa source dans le cas d'une propagation non dissipative.

Dans le cadre de la dynamique vibratoire, le système est modélisé par l'équation du mouvement suivante :

$$M\ddot{x}(t) + D\dot{x}(t) + Kx(t) = F(t)$$

où M , D et K désignent respectivement les matrices de masse, d'amortissement, et de raideur. $x(t)$ est le vecteur des déplacements et $F(t)$ le vecteur des forces appliquées. On considère les conditions initiales :

$$\begin{cases} x(t_0) = d_0 \\ \dot{x}(t_0) = v_0 \end{cases}$$

Pour effectuer un retournement temporel, on introduit un changement de variable : $\tau = -t$. Ce changement modifie l'équation du mouvement qui devient :

$$M\ddot{x}(\tau) - D\dot{x}(\tau) + Kx(\tau) = F(\tau)$$

On observe ici l'inversion du terme d'amortissement : le signe de $D\dot{x}$ change, ce qui implique que, pour un système faiblement amorti, cette inversion a peu d'impact et le retournement reste valable.

Les nouvelles conditions initiales pour la résolution dans le temps retourné sont les conditions finales du système direct :

$$\begin{cases} x(\tau_{initial}) = d_{final} \\ \dot{x}(\tau_{initial}) = v_{final} \end{cases}$$

Si l'amortissement est négligeable, alors le système retourné reconverge vers les conditions initiales du système direct. Cela permet en pratique de localiser une source d'excitation : en appliquant les vitesses et déplacements finaux mesurés en bout de chaîne comme nouvelles conditions initiales du système inversé, on peut faire "remonter" le signal jusqu'à son origine.

Ainsi pour des sources ponctuelles, les ondes créées par retournement temporel se rétropropagent et se focalisent en ces sources. En théorie les conditions finales retournées temporellement doivent être appliquées en tous les points du système vibrant.

2.1.2 Problématiques industrielles

Le retournement temporel peut être appliqué à plusieurs usages industriels :

- Surveillance d'intégrité de structure (SHM) à faible nombre de transducteurs
- Contrôle qualité en production
- Système de localisation de source
- Transmission de données sans fil par voie structurale

La surveillance de structure par RT se déroule en 2 phases principales :

- Une première phase d'apprentissage sur la structure dans son état initial. Cet apprentissage peut être expérimental, à la condition de pouvoir accéder aux zones à surveiller. Dans le cas contraire AVNIR travaille pour réaliser cet apprentissage par simulation numérique
- Une seconde phase de surveillance par injection des signaux retournés (environ 0.5 secondes de temps de signal) et contrôle de l'évolution des réponses du système

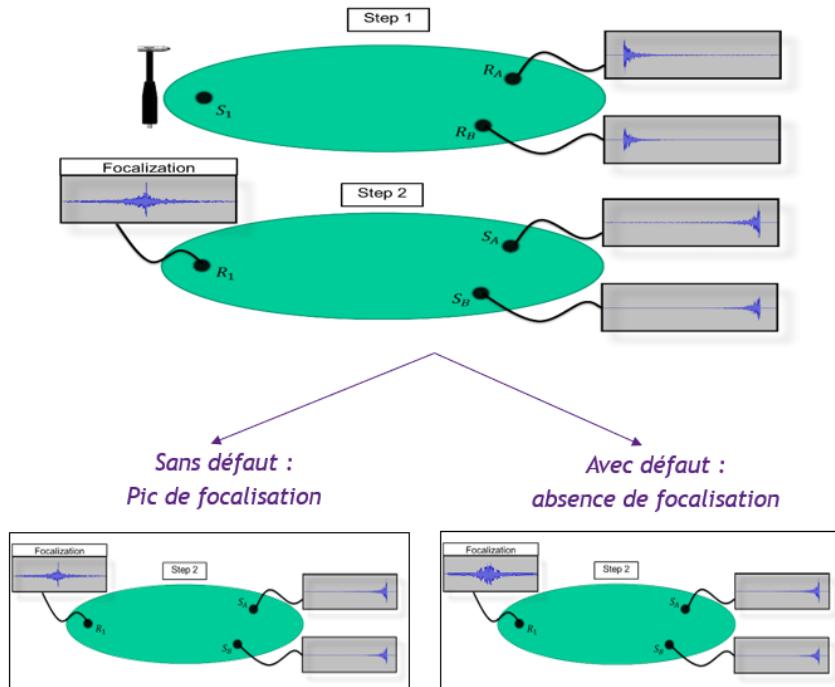


FIGURE 2.1 – Démonstration des deux phases du retournement temporel.

De nos jours, de nombreux projets de recherche en mécanique dynamique (acoustique, vibrations, ultrasons, etc.) reposent sur des modèles numériques. Pour faire face aux exigences de précision imposées par la complexité des problèmes d'ingénierie à résoudre, ces modèles numériques nécessitent des innovations dans divers domaines, parmi lesquels : le contrôle de santé des structures (SHM), notamment dans le cas de structures dont la géométrie est complexe. C'est dans ce sens qu'**AVNIR Energy** suit de près les évolutions dans le domaine de la simulation numérique notamment à travers ses projets autour du contrôle de santé des structures et de la surveillance acoustique.

2.2 Le projet SINUDYN

Le projet **SINUDYN** (SImulation NUMérique DYNamique) en collaboration avec l'entreprise **SONORHC**, est basé sur le développement d'une méthode de surveillance des structures industrielles grâce à la technique de retournement temporel.



Les calculs sur la CAO (Conception assistée par ordinateur) sont nécessaires pour réaliser un apprentissage numérique, sachant que l'état dans lequel on voudrait éventuellement surveiller les structures industrielles (à haute température, radiation) ne permet pas d'accéder à certaines zones pour réaliser le coup de marteau qui servira de force extérieure sur le système dans l'application de la méthode de retournement temporel.

Pour cela **SONORHC** dispose d'un brevet baptisé 3S comme Système de Surveillance de Structure ou encore Structural Survey System.

Technique 3S et retournement temporelle sur cuve

La structure qui sera étudiée durant ce stage est une maquette de cuve de réacteur nucléaire.

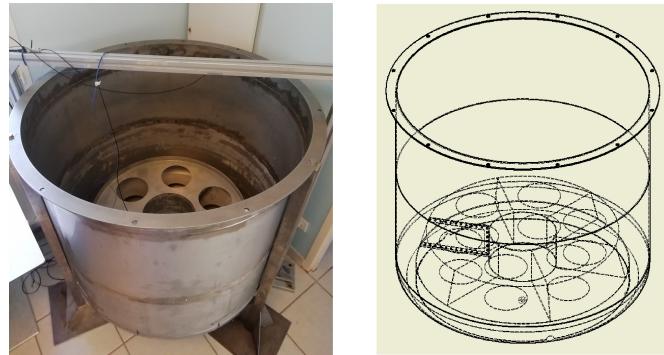
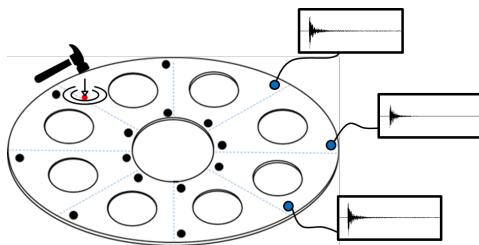


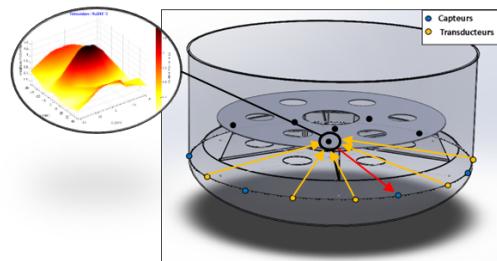
FIGURE 2.2 – Maquette de la cuve

La technique 3S, basée sur le retournement temporel, permet de focaliser des ondes mécaniques en plusieurs points d'une structure afin de créer des impacts virtuels, là où les ultrasons classiques ne peuvent accéder. L'analyse de la réponse vibratoire révèle d'éventuels défauts (fissures, dépôts, corrosion). Développée avec AREVA puis le CEA, elle s'applique aujourd'hui aux Small Modular Reactors (SMR) pour surveiller l'intégrité de structures internes critiques et inaccessibles.

Apprentissage

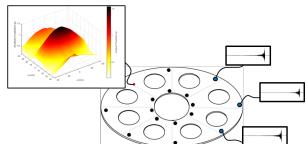


Apprentissage (*expérimental ou numérique*) des fonctions de transfert entre **capteurs** et **points de surveillance**. Impacts en zone de contrôle.

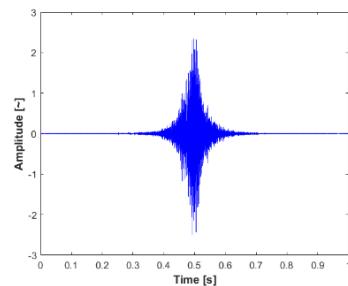


Focalisation de l'énergie

Focalisation



La réémission du signal **temporellement inversé** par les **transducteurs** induit une propagation en sens opposé, permettant une **focalisation** au point d'excitation initial.



Réponse de la structure

Surveillance

Mesure de la réponse structurelle : détection et localisation de défauts potentiels grâce à des indicateurs dédiés.

Les travaux d'**AVNIR Energy** se sont trouvés limités par le manque de précision des solutions classiques existantes (notamment la méthode des éléments-finis) dans le domaine de l'ingénierie mécanique vibratoire sur la résolution numérique des équations de la dynamique. En effet, même en utilisant des modèles classiques très raffinés en simulation, les résultats obtenus présentent des erreurs encore trop importantes par rapport aux résultats expérimentaux.

Face à ce constat, **AVNIR Energy** a considéré que, pour faire progresser de nombreux sujets de R&D, un effort supplémentaire devait être consacré à la modélisation numérique. La stratégie consiste à identifier des méthodes numériques avancées issues du monde académique, puis à les adapter ou à développer des extensions afin de répondre aux problématiques en mécanique vibratoire. Ainsi, intégrer efficacement la propagation des ondes mécaniques à haute fréquence, de traiter des géométries complexes et d'améliorer la précision, tout en optimisant le temps de calcul.

Partant du constat fait précédemment, l'objectif du projet **SINUDYN** est le suivant :

« Identifier et prendre en main des modèles numériques issus du monde de la recherche afin de les adapter aux problématiques de l'entreprise dans le domaine de la mécanique dynamique ».

Cependant, la transposition au contexte de propagation des ondes soulèvent des incertitudes liées à plusieurs paramètres, notamment :

- Effet d'échelle sur la propagation des ondes : sensibilité à la dispersion et aux effets non linéaires.
- Géométrie et conditions aux limites complexes : structure cylindrique à paroi fine (structure élancée), présence de plaquette internes avec orifices.
- Environnement : effets de la température, de la pression ou de la présence de fluides sur la propagation des ondes.

Chapitre 3

Objectifs du stage

L'objectif principal du stage est de contribuer à la modélisation numérique de phénomènes de propagation d'ondes dans des structures couplées fluide/solide et analyser la performance de la méthode numérique utilisée, dans le cas d'une géométrie de cuve nucléaire. Il s'agit de :

- Développer une chaîne de simulation utilisant les capacités multiphysiques de Feel++ ;
- Implémenter et comparer des schémas temporels explicites et implicites ;
- Analyser le jeu de paramètres (Ordre polynomiale, taille et ordre du maillage, pas de temps, solveur, etc) permettant un calcul plus précis et moins coûteux ;
- Étudier les conditions aux limites et les stratégies de couplage faible entre domaines ;
- Valider les résultats par analyse physique et post-traitement.

Chapitre 4

Méthodologie

4.1 Outils utilisés

Le travail repose sur l'écosystème suivant :

Feel++ (Finite Element Embedded Language in C++) est une bibliothèque logicielle scientifique en C++ dédiée à la résolution numérique d'équations aux dérivées partielles (EDP) par les méthodes de Galerkin, notamment les éléments finis (FEM) et les éléments spectraux (SEM) en 1D, 2D et 3D.

Elle est développée par le centre Cemosis (Université de Strasbourg) sous la direction de Christophe Prud'homme, et s'adresse à un public de chercheurs, ingénieurs, enseignants et étudiants travaillant sur la modélisation mathématique, la simulation multi-phénomène et le calcul haute performance.



Architecture modulaire. **Feel++** se compose de trois grands ensembles :

- **Le noyau **Feel++**** : une bibliothèque mathématique générique permettant de formuler et résoudre des problèmes variationnels avec gestion des maillages, espaces fonctionnels, intégration numérique, assemblage, solveurs linéaires, etc.
- **Les Toolboxes** : ensembles de modules applicatifs pour des domaines physiques spécifiques (mécanique des fluides, élasticité, transfert thermique, électromagnétisme, interaction fluide-structure...), exploitables sans écrire de code C++.
- **La réduction de modèle (MOR)** : outils et exemples pour construire des modèles réduits (par exemple méthode de base réduite), facilitant les simulations rapides en contexte paramétrique.

Principales caractéristiques.

- **Langage embarqué Galerkin** : formulation des problèmes directement en C++ sous forme proche des équations mathématiques.
- **Compatibilité multiphysique** : couplage naturel de plusieurs phénomènes physiques au sein d'un même cadre variationnel.
- **Parallélisme natif** : exécution distribuée (MPI) possible sur plusieurs milliers de coeurs sans modification de code utilisateur.
- **Interopérabilité** : interface Python via **pyFeel++**, intégration facile avec Docker, Gmsh, ParaView.

Toolboxes **Feel++** :

Les Toolboxes de **Feel++** fournissent :

- des bibliothèques permettant de manipuler des modèles physiques et de les coupler ;
- un ensemble d'applications mono- et multiphysiques prêtes à l'emploi dont :
 - mécanique des fluides (**FluidMechanics**) ;
 - mécanique des solides (**SolidMechanics**) ;
 - interaction fluide-structure (**FSI**) ;
- des interfaces Python (**pyFeel++ Toolboxes**) pour accéder aux modèles physiques et reproduire les principales fonctionnalités des toolboxes en Python.



Gmsh : Génération de maillages 1D/2D/3D avec balises physiques exploitables dans **Feel++** ;



ParaView : Visualisation et analyse des champs simulés ;



GitHub : Gestion de version et collaboration ;



Docker : Environnement reproductible pour le développement et les tests ;

4.2 Méthodes numériques

4.2.1 Limites des méthodes numériques classiques pour la propagation d'ondes

Dans le domaine de la dynamique des structures et de la sismique, les méthodes numériques classiques telles que la méthode des éléments finis (FEM) ou les différences finies (FD) sont largement utilisées. Cependant, ces approches présentent des limitations importantes pour modéliser la propagation d'ondes dans des structures 3D complexes.^[1]

Erreur relative de période :

Liée à une estimation incorrecte des périodes propres de vibration, aussi appelée *dispersion numérique*. Elle se traduit par une mauvaise évaluation de la vitesse de propagation des ondes selon la fréquence, engendrant des erreurs croissantes avec la distance de propagation.

L'erreur de dispersion dépend fortement de la taille caractéristique des éléments d_e , du schéma d'intégration, du type d'élément et du schéma temporel employé. Pour limiter cette erreur, la taille des éléments doit être suffisamment petite par rapport à la plus petite longueur d'onde λ_{\min} considérée, selon la relation :

$$d_e = \frac{\lambda_{\min}}{R}$$

où R est un facteur de résolution, généralement compris entre 10 et 20, voire jusqu'à 40 pour éliminer quasiment toute erreur numérique.

Par exemple, pour une structure métallique où la vitesse des ondes volumétriques est $c_p \approx 6000 \text{ m/s}$, la longueur d'onde associée à une fréquence $f = 1 \text{ MHz}$ vaut :

$$\lambda = \frac{c_p}{f} = \frac{6000}{10^6} = 6 \text{ mm.}$$

Avec un facteur de résolution $R = 10$, la taille des éléments doit être inférieure à 0.6 mm , conduisant à un maillage extrêmement fin (plusieurs milliards d'éléments pour un volume industriel), ce qui est prohibitif en temps de calcul et en ressources informatiques.

En outre, les méthodes FEM classiques génèrent souvent des approximations géométriques inadéquates, particulièrement pour les géométries courbes complexes, ce qui dégrade la qualité des simulations dynamiques.

Pour ces raisons, des méthodes numériques avancées, notamment les méthodes d'ordre élevé comme la méthode des éléments spectraux (SEM), sont préférées. La SEM fournit une meilleure précision avec moins d'éléments, grâce à sa convergence spectrale et sa matrice de masse diagonale, permettant des calculs plus efficaces et précis ^[2].

Cependant, la SEM présente aussi des limites : elle est moins adaptée aux milieux fortement hétérogènes ou aux géométries complexes nécessitant des maillages tétraédriques en 3D, ce qui restreint son application dans certains contextes industriels.

4.2.2 Méthodes des éléments finis (FEM) et des éléments spectraux (SEM)

Méthode des éléments finis (FEM)

Principe et motivation La méthode des éléments finis ^[4] est une technique de discrétisation spatiale pour la résolution numérique d'équations aux dérivées partielles. Elle repose sur :

- **Subdivision du domaine** : Découpage du domaine Ω en éléments simples (triangles, tétraèdres, hexaèdres), sur lesquels les fonctions de forme sont définies localement, ayant un support restreint à quelques éléments adjacents.
- **Approximation locale** : Sur chaque élément, la solution est approchée par des fonctions polynomiales de bas degré (ordinairement jusqu'à 2 ou 3).
- **Formulation variationnelle faible** : Transformation du problème différentiel en un problème variationnel qui est ensuite discrétisé.
- **Assemblage global** : Construction d'un système algébrique creux via l'assemblage des contributions élémentaires.

Formulation mathématique Soient $V = H^1(\Omega)$ et $V_0 = H_0^1(\Omega)$ les espaces de Sobolev usuels. Pour un problème elliptique linéaire donné par

$$\mathcal{L}u = f \quad \text{dans } \Omega, \quad u = g \quad \text{sur } \partial\Omega,$$

la formulation variationnelle consiste à trouver $u \in V$ telle que

$$a(u, v) = l(v) \quad \forall v \in V_0,$$

où $a(\cdot, \cdot)$ est une forme bilinéaire coercive et $l(\cdot)$ une forme linéaire continue.

La discrétisation introduit un sous-espace de dimension finie $V_h \subset V$:

$$u_h(\mathbf{x}) = \sum_{i=1}^N u_i \phi_i(\mathbf{x}),$$

avec $\{\phi_i\}_{i=1}^N$ les fonctions de forme à support local et $\{u_i\}_{i=1}^N$ les degrés de liberté.

Éléments de Lagrange Les éléments utilisés sont principalement des polynômes de Lagrange avec continuité C^0 :

- P1 : fonctions linéaires par morceaux, noeuds aux sommets,
- P2 : fonctions quadratiques, noeuds aux sommets et milieux d'arêtes,
- Ordres supérieurs (P3, P4, ...) possibles mais peu courants en FEM.
conformément à la standardisation des toolboxes.

Méthode des éléments spectraux (SEM)

Principe et motivation La méthode des éléments spectraux [2, 6] allie la flexibilité géométrique de la FEM à la précision spectrale des méthodes globales. Elle est particulièrement adaptée aux phénomènes de propagation d'ondes, notamment en dynamique vibratoire, grâce à :

- **Haute précision spectrale** : Réduction drastique des erreurs de dispersion numérique sur des ondes haute fréquence [8],
- **Convergence exponentielle** : Pour des solutions régulières, l'erreur diminue exponentiellement avec l'ordre polynomial p ,
- **Matrice de masse diagonale** : Permettant des schémas temporels explicites rapides, réduisant la complexité de résolution.

Formulation spectrale Les bases sont construites avec des polynômes de Lagrange d'ordre élevé, définis sur les points de Gauss-Lobatto-Legendre (GLL), incluant systématiquement les bornes $\xi = \pm 1$, ce qui facilite la continuité inter-élémentaire et l'imposition des conditions aux limites.

La solution discrète s'écrit :

$$u_h(\mathbf{x}) = \sum_{i=0}^N u_i L_i(\xi),$$

où $L_i(\xi)$ sont les polynômes d'interpolation de Lagrange sur les points GLL $\{\xi_i\}_{i=0}^N$, racines de

$$(1 - \xi^2) P'_N(\xi) = 0,$$

avec $P_N(\xi)$ le polynôme de Legendre de degré N .

Avantages pour la dynamique vibratoire

- La quadrature GLL garantit une matrice de masse diagonale, optimisant les calculs explicites dans le temps,
- La précision spectrale diminue significativement les erreurs de phase (dispersion numérique), critère crucial pour la fidélité des simulations d'ondes.

Comparaison entre FEM et SEM

Philosophies de discrétisation

- FEM : raffinement spatial par réduction de la taille caractéristique du maillage h , avec polynômes de faible degré (généralement $p \leq 3$),
- SEM : raffinement par élévation de l'ordre polynomial p , avec polynômes d'ordre élevé (souvent $5 \leq p \leq 15$).

Propriétés de convergence

- FEM : convergence algébrique, erreur typiquement en $O(h^p)$ [4],
- SEM : convergence exponentielle $O(e^{-cp})$ sous hypothèse de régularité suffisante de la solution [2], [9].

Complexité numérique

- FEM : assemblage et factorisation de matrices creuses dont la complexité est approximativement $O(N^{3/2})$ en 2D et $O(N^2)$ en 3D [7],
- SEM : matrice de masse diagonale, ce qui permet une inversion directe en $O(N)$ [8] et des schémas explicites très efficaces.

TABLE 4.1 – Comparaison FEM vs SEM

Critère	FEM	SEM
Fonctions de base	Polynômes de Lagrange sur points équidistants	Polynômes de Lagrange sur points Gauss-Lobatto-Legendre
Quadrature	Gauss standard	Gauss-Lobatto-Legendre
Matrice de masse	Pleine (factorisation coûteuse)	Diagonale (inversion directe, schémas explicites)
Précision	Modérée	Très élevée (précision spectrale)
Dispersion numérique	Importante à haute fréquence	Faible grâce à la convergence spectrale

Différences d'implémentation FEM vs SEM dans Feel++

- **Quadrature** : FEM utilise des règles de Gauss standard, SEM la quadrature GLL spécifique pour garantir l'orthogonalité des bases et la diagonale de la matrice de masse.
- **Assemblage** : FEM assemble des matrices creuses, SEM peut exploiter des structures plus denses mais avec des optimisations spectrales.
- **Schémas temporels** : FEM privilégie souvent des schémas implicites à cause des matrices pleines, SEM peut utiliser des schémas explicites plus rapides grâce à la diagonalisation.

L'entreprise a choisi d'utiliser la bibliothèque Feel++ principalement pour sa prise en charge de la méthode des éléments spectraux (SEM). En effet, les méthodes classiques des éléments finis (FEM), bien que robustes et largement standardisées, présentent des limites pour la modélisation précise des phénomènes de propagation d'ondes à haute fréquence que nous étudions. De plus, les toolboxes Feel++ sont historiquement orientés FEM, avec :

- une intégration numérique basée sur la quadrature de Gauss standard,
- des schémas numériques principalement implicites,
- une discrétisation des espaces de fonctions et un maillage limités à des ordres faibles (typiquement jusqu'à P2),
- une structure logicielle optimisée pour ce cadre.

En revanche, la méthode SEM offre une meilleure précision, notamment grâce à l'utilisation de polynômes d'ordre élevé et de points de quadrature Gauss-Lobatto-Legendre (GLL), ainsi qu'une matrice de masse diagonale permettant des schémas explicites efficaces.

4.2.3 Stratégies d'intégration temporelle

Les deux schémas qui seront utilisés principalement sont le schéma de Newmark et le schéma BDF à l'ordre 2. La version implicite de ces schémas est celle qui est utilisée dans les toolboxes FEEL++ et permettent une stabilité relative même pour des pas de temps assez grand. Cependant, plus tard les termes liés à la résolution seront reformulés d'une manière explicite en utilisant les fonctionnalités des classes Newmark et BDF définis dans FEEL++.

Schéma de Newmark

Pour la résolution du déplacement du solide, le schéma de Newmark est utilisé en raison de sa stabilité inconditionnelle et de sa capacité à gérer efficacement les équations du second ordre typiques du solide. Avec les paramètres classiques $\gamma = \frac{1}{2}$ et $\beta = \frac{1}{4}$, il permet une intégration temporelle implicite assurant une bonne précision et stabilité numérique même pour des pas de temps relativement grands [13].

Les mises à jour des déplacements \mathbf{u} , vitesses \mathbf{v} et accélérations \mathbf{a} sont données par :

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \mathbf{v}^n + \frac{\Delta t^2}{2} [(1 - 2\beta)\mathbf{a}^n + 2\beta \mathbf{a}^{n+1}], \quad (4.1)$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \Delta t [(1 - \gamma)\mathbf{a}^n + \gamma \mathbf{a}^{n+1}]. \quad (4.2)$$

Ici, $\mathbf{u}^n, \mathbf{v}^n, \mathbf{a}^n$ sont respectivement le déplacement, la vitesse et l'accélération au temps t_n , et Δt le pas de temps.

Méthodes BDF

Pour la pression du fluide, les schémas Backward Differentiation Formula (BDF) d'ordre 2 (BDF2) [11] sont privilégiés. La discréétisation temporelle de la pression p s'écrit :

$$\frac{3p^{n+1} - 4p^n + p^{n-1}}{2\Delta t} = \mathbf{f}^{n+1} \quad (4.3)$$

où \mathbf{f}^{n+1} représente un terme source évalué au temps t_{n+1} .

4.2.4 Couplage fluide-structure

Algorithme de Picard

Le couplage fluide-structure est résolu par un algorithme itératif de type Picard [12], adapté aux problèmes où les deux milieux interagissent fortement mais peuvent être traités par solveurs distincts.

À chaque itération k du couplage à l'instant t_{n+1} :

1. **Prédiction** : extrapolation des variables d'interface - déplacement $\mathbf{u}_{\Gamma_{fsi}}^{(k)}$ et pression $p_{\Gamma_{fsi}}^{(k)}$ - à partir des itérations précédentes.
2. **Résolution fluide** : calcul de la pression $p^{(k+1)}$ dans le fluide avec le déplacement imposé $\mathbf{u}_{\Gamma_{fsi}}^{(k)}$ sur l'interface Γ_{fsi} .
3. **Résolution solide** : calcul du déplacement $\mathbf{u}^{(k+1)}$ dans le solide avec la pression $p_{\Gamma_{fsi}}^{(k+1)}$ imposée sur Γ_{fsi} .
4. **Critère de convergence** : vérification que les écarts aux itérations précédentes sont inférieurs à la tolérance ϵ_{fsi} :

$$\|\mathbf{u}_{\Gamma_{fsi}}^{(k+1)} - \mathbf{u}_{\Gamma_{fsi}}^{(k)}\|_2 < \epsilon_{FSI} \quad \text{et} \quad \|p_{\Gamma_{fsi}}^{(k+1)} - p_{\Gamma_{fsi}}^{(k)}\|_2 < \epsilon_{fsi}.$$

Cette boucle Picard assure la cohérence physique et numérique entre les champs fluides et solides à chaque pas de temps, permettant d'intégrer indépendamment les solveurs tout en garantissant la convergence du couplage.

4.3 Modèle physique

Pour décrire l'interaction dynamique entre un solide élastodynamique et un fluide acoustique, nous considérons les équations couplées gouvernant les deux milieux. Cette modélisation permet de capturer les échanges de forces et les déplacements sur l'interface fluide-structure, dans un cadre temps-dépendant.

Pour modéliser le système, on regarde le matériau et la géométrie de la cuve, la sollicitation, les conditions aux limites, les conditions de couplage entre la structure et le fluide ainsi que l'amortissement dans le solide.

Matériau et sollicitation Le domaine solide est modélisé comme un matériau isotrope élastique linéaire, caractérisé par :

- Module de Young : $E = 2.1 \times 10^{11}$ Pa
- Coefficient de Poisson : $\nu = 0.3$
- Masse volumique : $\rho_s = 7870$ kg/m³
- Contrainte limite élastique : $\sigma_y = 2.5 \times 10^8$ Pa

Une force surfacique harmonique transitoire est appliquée sur la surface Γ_N selon la condition de Neumann suivante :

$$f_N(t) = A \sin(2\pi f_c t) \exp(-5(f_c t - 2)^2)$$

avec :

- Amplitude $A = 1 \times 10^5$ Pa utilisé pour
- Fréquence centrale $f_c = 5000$ Hz

Cette sollicitation correspond à une onde sinusoïdale amortie par une gaussienne centrée en $t = 2/f_c$, représentant une excitation transitoire localisée en fréquence. Onde de type Ricker pour simuler un coup de marteau virtuel.

4.3.1 Domaines et Notations

Soit :

- $\Omega_s \subset \mathbb{R}^3$: domaine solide (structure)
- $\Omega_f \subset \mathbb{R}^3$: domaine fluide (acoustique)
- $\Gamma_{fsi} = \partial\Omega_s \cap \partial\Omega_f$: interface fluide–structure
- $\mathbf{u}_s(\mathbf{x}, t)$: déplacement du solide
- $p(\mathbf{x}, t)$: pression acoustique dans le fluide
- $f_s(\mathbf{x}, t)$: force extérieure appliquée sur le solide
- $f_f(\mathbf{x}, t)$: force extérieure appliquée sur le fluide

4.3.2 Équations gouvernantes

Solide : Élastodynamique

Trouver $\mathbf{u}_s : \Omega_s \times [0, T] \rightarrow \mathbb{R}^3$ tel que :

$$\rho_s \frac{\partial^2 \mathbf{u}_s}{\partial t^2} - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}_s) = \mathbf{f}_s(\mathbf{x}, \mathbf{t}) \quad \text{dans } \Omega_s, \quad (4.4)$$

avec la relation contrainte-déformation linéaire élastique :

$$\boldsymbol{\sigma}(\mathbf{u}_s) = \lambda(\nabla \cdot \mathbf{u}_s)\mathbf{I} + 2\mu\varepsilon(\mathbf{u}_s), \quad \varepsilon(\mathbf{u}_s) = \frac{1}{2}(\nabla \mathbf{u}_s + \nabla \mathbf{u}_s^T).$$

où $\varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ est le tenseur de déformation infinitésimale (petites déformations)

Fluide : Équation d'onde acoustique

Trouver $p : \Omega_f \times [0, T] \rightarrow \mathbb{R}$ tel que :

$$\frac{\partial^2 p}{\partial t^2} - c^2 \Delta p = \mathbf{f}_f(\mathbf{x}, \mathbf{t}) \quad \text{dans } \Omega_f, \quad (4.5)$$

où c est la vitesse du son dans le fluide et ρ_f sa densité.

4.3.3 Conditions de couplage sur l'interface Γ_{fsi}

- **Condition dynamique** (équilibre des forces) :

$$\boldsymbol{\sigma}(\mathbf{u}_s) \cdot \mathbf{n}_s = -p \mathbf{n}_s,$$

avec \mathbf{n}_s la normale extérieure au solide.

- **Condition cinématique** (continuité des vitesses normales) :

$$\frac{\partial \mathbf{u}_s}{\partial t} \cdot \mathbf{n}_s = -\frac{1}{\rho_f} \frac{\partial p}{\partial \mathbf{n}_f},$$

où \mathbf{n}_f est la normale extérieure au fluide.

4.3.4 Conditions initiales et aux limites

Système initialement au repos

$$\mathbf{u}_s(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \quad \frac{\partial \mathbf{u}_s}{\partial t}(\mathbf{x}, 0) = \mathbf{v}_0(\mathbf{x}).$$

$$p(\mathbf{x}, 0) = p_0(\mathbf{x})$$

$$\mathbf{u}_0 = \mathbf{0}, \quad \mathbf{v}_0 = \mathbf{0}, \quad p_0 = 0.$$

Condition aux limites (solide) Encastrement sur $\Gamma_{s_fixed} \subset \partial\Omega_s$:

$$\mathbf{u}_s(\mathbf{x}, t) = \mathbf{0} \quad \text{pour } \mathbf{x} \in \Gamma_{fixed}, \quad t \in [0, T].$$

4.3.5 Modélisation de l'amortissement dans le solide

Pour modéliser l'amortissement des hautes fréquences, on utilise un amortissement de Rayleigh défini par :

$$\sigma_d = \alpha\sigma(\mathbf{u}_s) + \beta\dot{\epsilon}(\mathbf{u}_s),$$

où les coefficients α et β sont choisis pour cibler des taux d'amortissement modal spécifiques.

4.4 Modèle géométrique

4.4.1 Description de la cuve

La géométrie d'étude consiste en une cuve cylindrique dont les dimensions principales sont :

— **Hauteur** : 0.964 m

— **Diamètre** : 0.944 m

— **Épaisseur de paroi** : 3 mm

La cuve est maintenue par un système de support constitué de 3 poutres vissées à la collerette. La totalité de la collerette sera définie comme surface encastrée, plus tard nous pourrons rajouter des conditions d'enca斯特ment plus compatibles avec le système de support de la cuve.

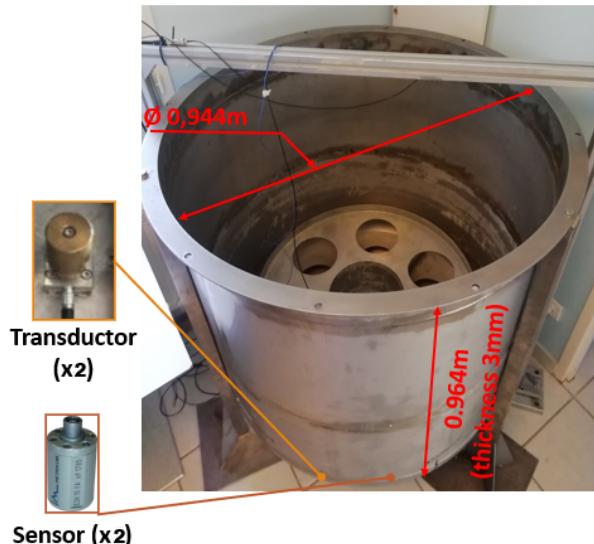


FIGURE 4.1 – Maquette de la cuve

4.4.2 Stratégie de maillage avec Gmsh

AVNIR a fourni deux fichiers STEP (format standardisé pour l'utilisation de CAO sur différentes plateformes, ici générés avec ANSYS) pour différents types de simulations :

- **Cuve simplifiée** : Version sans les plaquettes internes
 - Dédiée aux simulations d'interaction fluide-structure (FSI)
 - Géométrie optimisée pour réduire la complexité du maillage fluide
- **Cuve complète** : Version avec toutes les plaquettes internes
 - Modèle complet pour les simulations structurelles
 - Prise en compte de tous les détails géométriques

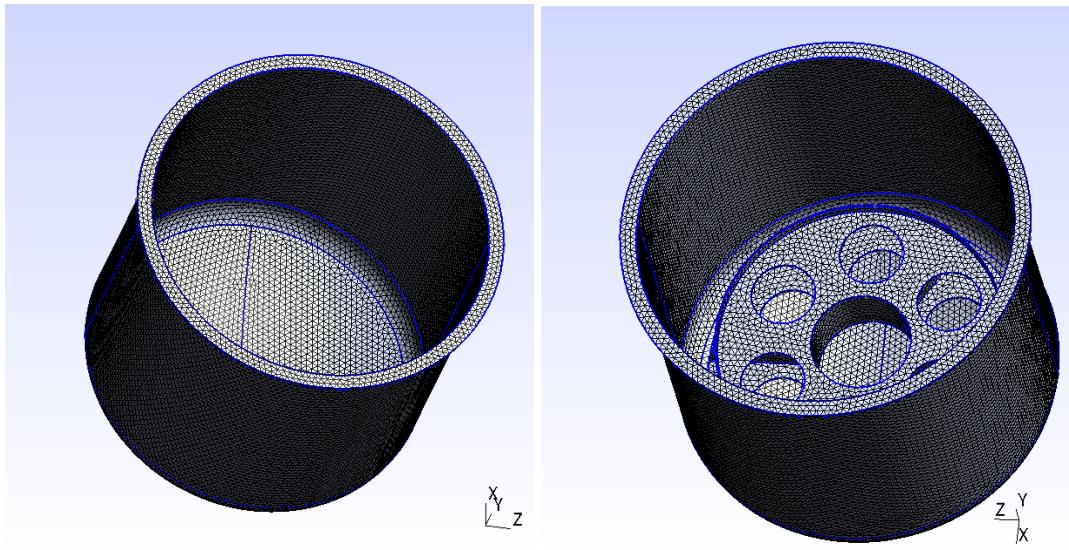


FIGURE 4.2 – Maillages 3D : (gauche) cuve simplifiée, (droite) cuve complète

Les fichiers STEP sont importés dans Gmsh via l'option `merge` permettant :

- Import et fusion des géométries CAO
- Conservation des entités géométriques d'origine
- Définition des régions physiques pour Feel++

Un troisième maillage est également créé :

- **Coupe 2D** : Réalisée directement dans Gmsh
- Utilisée pour les études préliminaires et la validation
- Réduction significative du coût calcul

Nous allons principalement utiliser des maillages tétraédriques issus de l'algorithme de maillage Delaunay par défaut de GMSH. Il est néanmoins possible d'utiliser un maillage hexaédrique ce qui est préférable pour la méthode des éléments spectraux.

Stratégie de raffinement adaptatif : Selon le type de simulation, deux approches de maillage sont adoptées :

- **Simulations FSI** (cuve simplifiée et 2D) :
 - Maillage adaptatif avec raffinement local autour de l'interface fluide-structure
 - Optimisation du rapport précision/coût calcul en concentrant les éléments fins sur `Gamma_fsi`
 - Transition progressive vers un maillage plus grossier loin de l'interface
- **Simulations solides** (cuve vide complète) :
 - Maillage uniforme avec précision maximale sur toute la structure

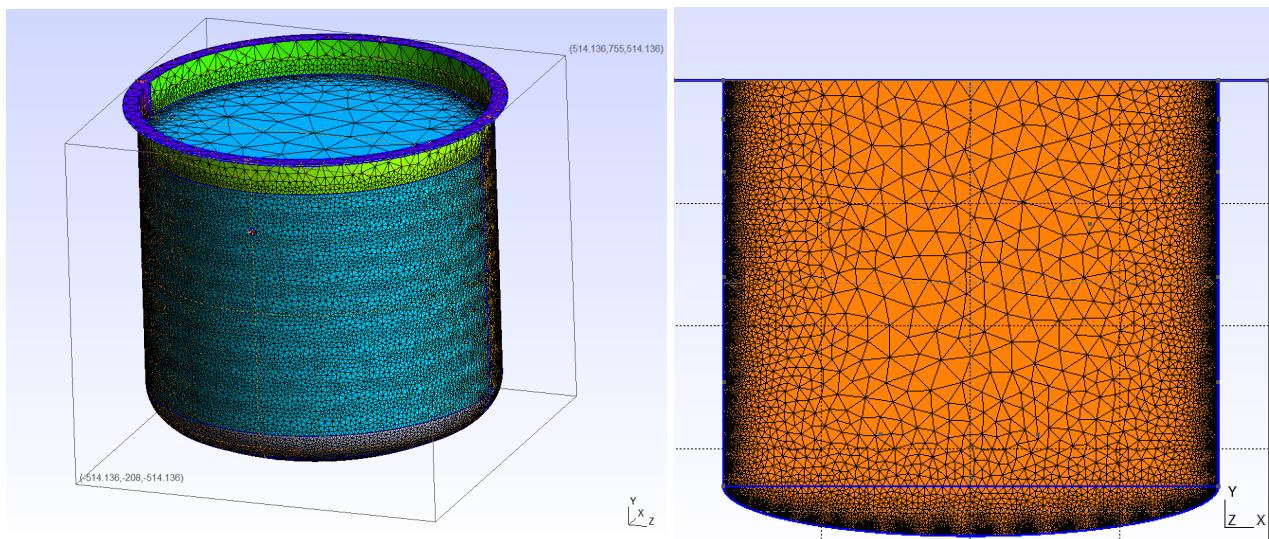


FIGURE 4.3 – Maillages avec raffinement adaptatif

Définition des régions physiques

Les entités physiques suivantes sont définies dans Gmsh pour l'utilisation dans Feel++ :

Volumes physiques :

- "Omega_f" : Domaine fluide
- "Omega_s" : Domaine solide

Surfaces physiques :

- "Gamma_s" : Surfaces libres du solide
- "Gamma_s_fixed" : Surface d'encastrement (collerette)
- "Gamma_N" : Surface de Neumann (application des efforts)
- "Gamma_fsi" : Interface fluide-structure
- "Gamma_top" : Surface supérieure du fluide (pression nulle)

Points de mesure : Des points physiques supplémentaires sont définis comme capteurs pour l'extraction des mesures :

- Extraction des déplacements, vitesses et accélérations
- Positionnement stratégique selon les besoins expérimentaux

4.4.3 Caractéristiques des maillages

Trois tailles caractéristiques de maillage sont utilisées pour l'étude de convergence :

- **m1 = 3 mm** : Maillage fin (haute précision)
- **m2 = 6 mm** : Maillage moyen
- **m3 = 12 mm** : Maillage grossier

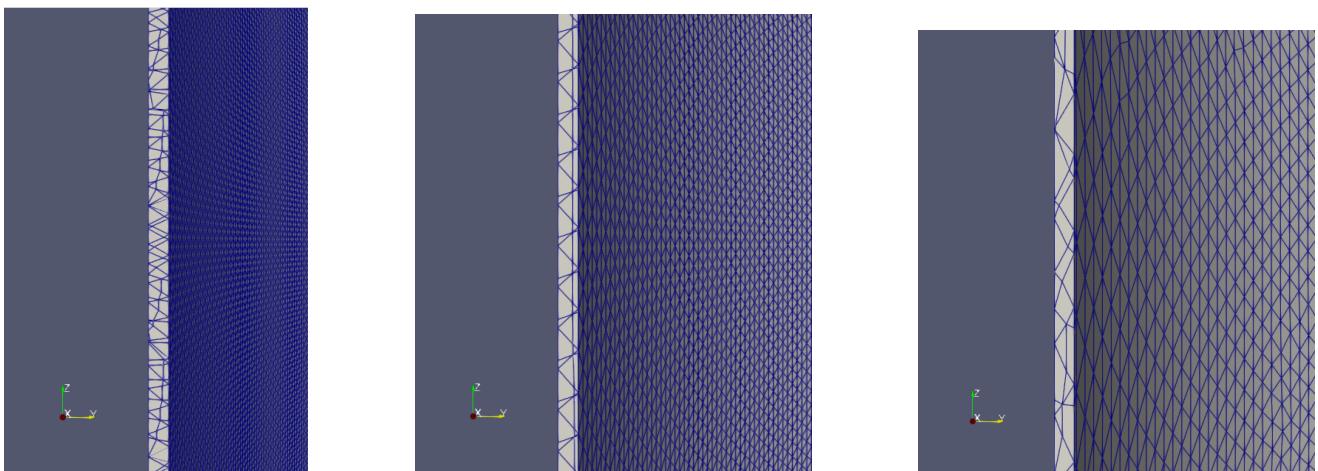


FIGURE 4.4 – Comparaison des tailles de maillage : (gauche) $h=3\text{mm}$, (centre), $h=6\text{mm}$ (droite) $h=12\text{mm}$

Les éléments peuvent être générés avec des ordres géométriques supérieurs dans Gmsh que l'on identifiera par Gn :

- **Ordre G1** : Éléments linéaires (tétraèdres/hexaèdres droits)
- **Ordre G2** : Éléments quadratiques (faces courbes)
- **Ordre > 2** : Éléments à haute précision géométrique pour l'iso-paramétrisation avec l'ordre des polynômes des espaces de fonctions utilisés pour la méthode SEM

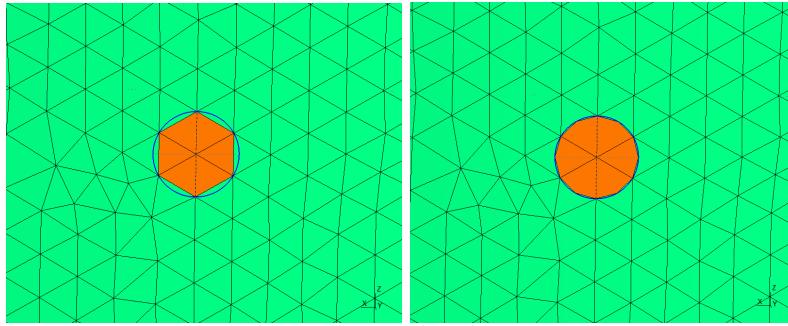


FIGURE 4.5 – Comparaison des ordre de discréétisation géométriques : (gauche) ordre G1, (droite) ordre G2

Le tableau suivant résume les caractéristiques de chaque configuration de maillage Pour la géométrie de la cuve 3D complète sans contenu avec GMSH :

TABLE 4.2 – Caractéristiques des différents maillages

Taille	Ordre	Nb. éléments	Taille disque	Temps calcul
m1 (3mm)	G1	5927252	262.65 MB	5 :58.78
	G2		840.96 MB	8 :22.26
m2 (6mm)	G1	1519288	64.24 MB	1 :08.64
	G2		207.35 MB	1 :37.85
m3 (12mm)	G1	396689	15.50 MB	0 :16.34
	G2		51.96 MB	0 :22.47

4.5 Mesures expérimentales sur la cuve

Durant le stage j'ai pu participer aux prises de mesures expérimentales sur la structure dans les locaux de SONORHC).

4.5.1 Mise en place expérimentale

Les tests ont été effectués sur une maquette réduite d'une cuve de réacteur nucléaire en acier inoxydable vidée de son contenu (eau, sodium, plomb..). Des transducteurs piézoélectriques sont attachés sur la paroi inférieure de la cuve, agissant respectivement comme émetteur et récepteur de signaux vibratoires.



Montage expérimental de la cuve



Transducteurs

4.5.2 Détection des défauts : description du protocole

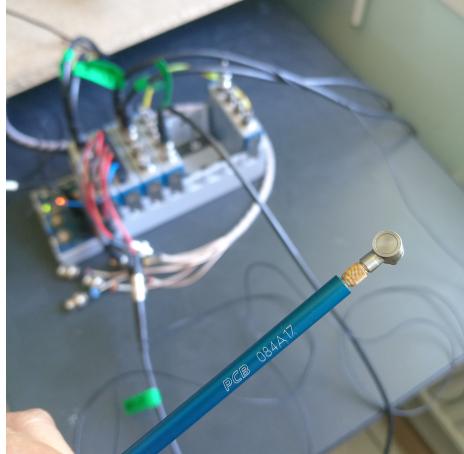
Le processus de détection de défauts sur la cuve s'organise en deux grandes phases : une phase d'apprentissage (learning) et une phase de focalisation/surveillance (monitoring).

Phase d'apprentissage (learning)

L'apprentissage consiste à exciter mécaniquement la cuve sur les zones à surveiller. L'excitation est réalisée à l'aide d'un petit marteau électronique. La force d'impact doit dépasser un seuil précis pour que les capteurs s'activent. Il est crucial que le coup soit à la fois *rapide* et *souple*, afin d'éviter :

- une double excitation (rebond),

— un contact prolongé avec la cuve.



Marteau électronique



Tige métallique pour zones internes

Pour les zones difficilement accessibles, notamment les plaques internes, particulièrement sujettes à l'usure et aux dommages, une tige métallique est utilisée pour transmettre l'impact.

Chaque impact génère une vibration, captée par les transducteurs. Le signal est enregistré pendant une durée d'environ une seconde, à une fréquence d'échantillonnage de 51 200 Hz, soit un pas de temps de $\Delta t = \frac{1}{51200} \approx 2 \cdot 10^{-5}$ s.

Cette opération est répétée pour chaque point d'impact identifié comme une future zone de surveillance. La taille des zones dépend directement de la gamme fréquentielle utilisée dans le programme de traitement, entre 800 et 5000Hz dans notre cas.

Construction de l'état de référence

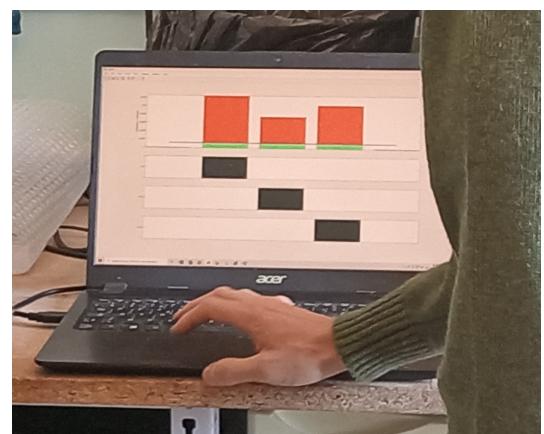
Les signaux obtenus durant l'apprentissage sont ensuite rejoués en sens inverse (*retournement temporel*) un certain nombre de fois (correspondant au nombre de moyennages souhaités). Le signal ainsi reconstruit est considéré comme l'**état de référence** du système, c'est-à-dire son état « sain », en l'absence de tout défaut.

Phase de surveillance (monitoring)

Dans cette phase, des défauts sont simulés artificiellement en vissant des objets de 20 grammes sur les zones ciblées durant l'apprentissage. Les mêmes signaux que dans la phase d'apprentissage sont ensuite rejoués. Les signaux capturés par les transducteurs sont comparés à l'état de référence.



Exemple de défaut simulé fixé sur une zone d'impact.



Détection des défauts dans les zones de monitoring.

Critère de détection de défaut

Un défaut est détecté lorsqu'un écart significatif avec l'état de référence est observé. Ce seuil de détection est défini dans le programme de traitement. Si l'amplitude de la différence dépasse le seuil dans une des zones surveillées, un défaut y est déclaré.

Remarques : Cette méthode de détection non destructive permet :

- d'apprendre l'état sain d'une structure sans l'endommager,
- de détecter la présence ultérieure de défauts de manière localisée.
- d'automatiser le processus d'apprentissage et monitoring ce qui permet de suivre l'évolution de l'usure au cours du temps.

4.5.3 Choix des surfaces et points d'intérêt

Retrouver les coordonnées

Points importants pour mesures expérimentales :

- Zones de surveillances (points d'excitations numérotés, 1,2,3 sur la plaquette interne supérieure et 4 sur la paroi de la cuve)
- Capteurs (accéléromètres)
- Transducteurs



FIGURE 4.6 – Positions des points d'excitation, capteurs et transducteurs

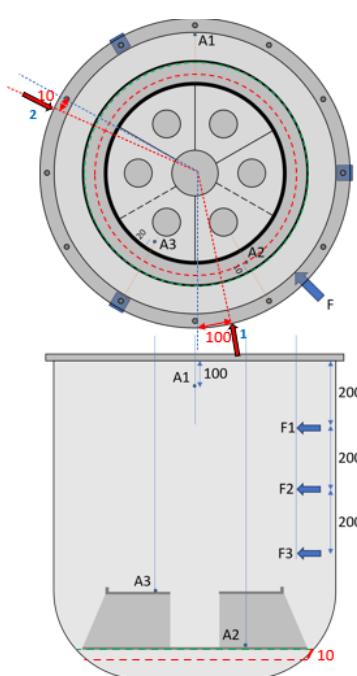


FIGURE 4.7 – Positions des accéléromètres

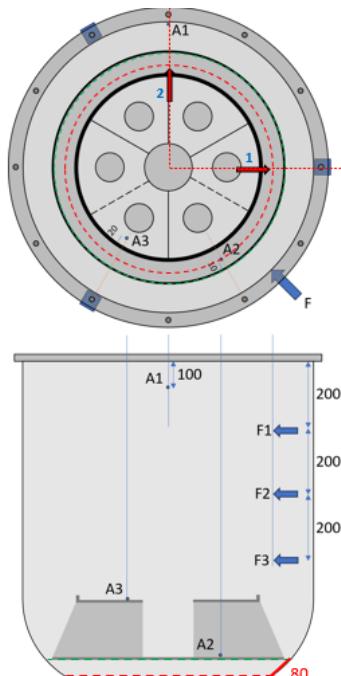


FIGURE 4.8 – Positions des transducteurs

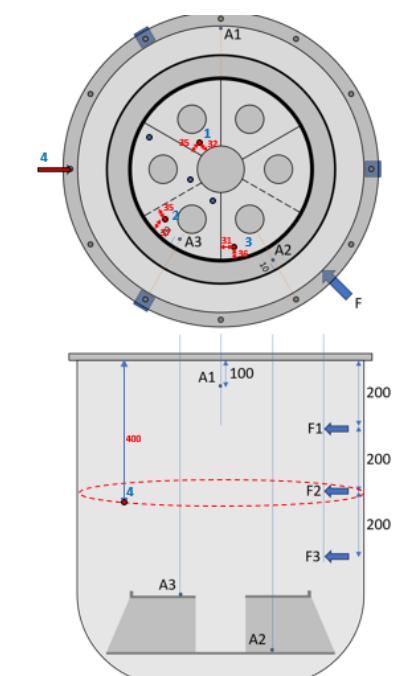


FIGURE 4.9 – Positions des points d'excitation

Placer les coordonnées sur la CAO

Pour la phase d'apprentissage, nous devons évaluer les accélérations dans les capteurs lorsque nos zones de surveillance sont excitées.

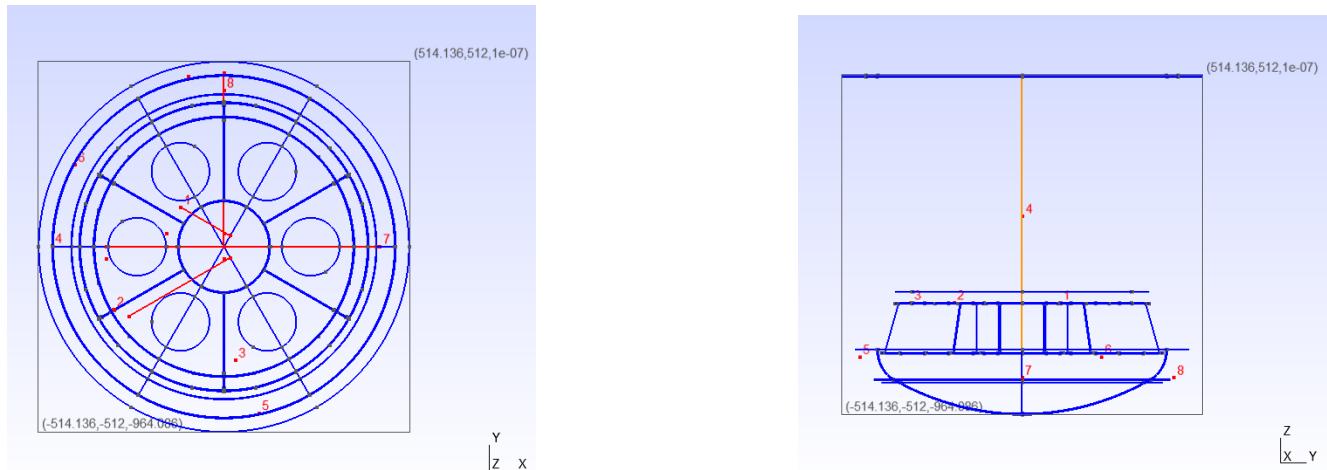


FIGURE 4.10 – Positions des points de mesure sur gmsh

On place les points dans le maillage : Les points 1, 2, 3 et 4 représentent les points d'excitation. Les points 5 et 6 les accéléromètres et les points 7 et 8 les transducteurs.

Chapitre 5

Implémentation avec FEEL++

5.1 Utilisation des toolboxes existantes

Les toolboxes Feel++ utilisent un système de configuration modulaire basé sur deux types de fichiers complémentaires.

Les fichiers **.cfg** permettent de configurer les aspects numériques de la simulation : répertoires de sortie, paramètres de maillage, schémas d'intégration temporelle, choix des solveurs linéaires et non-linéaires, ainsi que leurs tolérances de convergence.

Les fichiers **.json** définissent quant à eux le modèle physique complet : nomenclature des domaines, équations gouvernantes, propriétés des matériaux, conditions initiales et aux limites, et stratégies de post-traitement. Cette séparation claire entre la physique du problème (**json**) et sa résolution numérique (**cfg**) facilite la réutilisation des configurations et permet une adaptation rapide des paramètres selon les besoins de calcul.

5.1.1 Toolbox Solid Computational Mechanics

Théorie

La **toolbox SolidMechanics** de Feel++ implémente la résolution numérique des équations de la mécanique des milieux continus déformables.

Elle repose sur l'équation fondamentale de Newton appliquée aux milieux solides :

$$\rho_s^* \frac{\partial^2 \eta_s}{\partial t^2} - \nabla \cdot (F_s \Sigma_s) = f_s^t$$

où η_s représente le déplacement, ρ_s^* la densité, F_s le gradient de déformation et Σ_s le tenseur de contrainte de Piola-Kirchhoff.

La cinématique des déformations est caractérisée par plusieurs tenseurs fondamentaux : le gradient de déformation $F_s = I + \nabla \eta_s$, le tenseur de Cauchy-Green $C_s = F_s^T F_s$, et le tenseur de Green-Lagrange défini par :

$$E_s = \frac{1}{2}(C_s - I) = \frac{1}{2}(\nabla \eta_s + (\nabla \eta_s)^T) + \frac{1}{2}((\nabla \eta_s)^T \nabla \eta_s)$$

où le premier terme correspond aux déformations linéaires ε_s et le second aux déformations non-linéaires γ_s , E_s étant le module de Young du matériau.

La toolbox supporte plusieurs modèles constitutifs selon la nature des déformations : l'élasticité linéaire avec la loi de Hooke ($\Sigma_s = \lambda_s \text{tr}(\varepsilon_s)I + 2\mu_s \varepsilon_s$) pour les petites déformations, et l'hyperélasticité pour les grandes déformations avec les modèles Saint-Venant-Kirchhoff et Neo-Hookeen.

Implémentation

La toolbox est configuré pour la méthode FEM et donc utilise des intégrations Gauss standards et non GLL et un schéma en temps implicite. L'utilisation de la toolbox SolidMechanics vise principalement à :

- **Benchmarker les simulations** : évaluer les performances selon différentes configurations
- **Optimiser le ratio coût/précision** : identifier les paramètres offrant le meilleur compromis
- **Valider la stabilité numérique** : tester la robustesse des schémas d'intégration

Configuration numérique testée

Préconditionneur GAMG (Geometric Algebraic MultiGrid) :

- Méthode multigrid algébrique adaptée aux problèmes d'élasticité

- Convergence rapide pour les systèmes issus de discrétisations EF
- Scalabilité excellente en parallèle
- GASM** également testé :
- Décomposition de domaine avec recouvrement
- Alternative pour certaines géométries complexes

Schémas d'intégration temporelle :

Newmark (schéma principal)

- Schéma implicite d'ordre 2
- Stabilité inconditionnelle pour $\beta = 1/4, \gamma = 1/2$

Schémas alternatifs testés :

- **BDF** : méthodes multi-pas, robustes pour problèmes raides
- **Theta** : paramètre $\theta = 0.6, 0.7$ pour introduire de l'amortissement numérique

Autres options :

```
jacobian-linear-update=false
linearsystem-cst-update=false
```

Ces options exploitent la linéarité du problème :

- **jacobian-linear-update=false** : évite le recalcul de la matrice jacobienne (constante)
- **linearsystem-cst-update=false** : maintient la factorisation du système linéaire
- **Gain** : réduction significative du temps d'assemblage et de résolution

Décomposition par composantes (Fieldsplit) :

```
fieldsplit-use-components=1
fieldsplit-fields=0->(0),1->(1),2->(2)
fieldsplit-type=multiplicative
```

Cette approche décompose le système vectoriel 3D en trois sous-systèmes scalaires correspondant aux composantes X, Y et Z du déplacement. Chaque composante utilise un solveur GAMG indépendant avec préconditionnement optimisé :

- **fieldsplit-use-components=1** : active la décomposition par composantes vectorielles
- **fieldsplit-fields** : mapping des composantes (0→X, 1→Y, 2→Z)
- **fieldsplit-type=multiplicative** : résolution séquentielle des sous-systèmes
- **Avantage** : parallélisation efficace et convergence améliorée pour les problèmes vectoriels
- **Coût** : overhead de communication entre les sous-systèmes

Configuration physique du modèle

Paramètres d'excitation

```
"Parameters": {
    "fc": "5000",
    "force_hammer": "-sin(2*pi*fc*t)*exp(-5*(fc*t-2)*(fc*t-2)):fc:t"
}
```

- **fc = 5000 Hz** : fréquence caractéristique de l'excitation
- **force_hammer** : onde sinusoïdale modulée par gaussienne (onde évanescante)

L'excitation tente de reproduire l'impact d'un marteau expérimental :

$$F(t) = -\sin(2\pi f_c t) \times \exp(-5(f_c t - 2)^2)$$

```
"Materials": {
    "Omega_s": {
        "E": "2.1e11",
        "nu": "0.3",
        "rho": "7870",
        "yield_stress": "2.5e8"
    }
}
```

Propriétés de l'acier inoxydable :

- E : Module de Young [Pa]
- ν : Coefficient de Poisson
- ρ : Densité de l'acier [kg/m^3]

Conditions aux limites

Encastrement de la structure

```
"displacement": {
    "Gamma_s_fixed": {
        "expr": "{0,0,0}"
    }
}
```

- **Gamma_s_fixed** : surface d'encastrement (surface au niveau de la collerette)
- **Condition** : déplacement nul dans toutes les directions

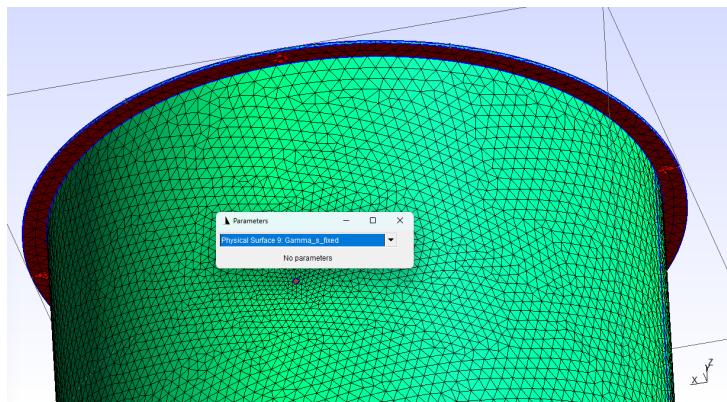


FIGURE 5.1 – Surface d'encastrement Gamma_s_fixed

Excitation par contrainte normale

```
"normal_stress": {
    "Gamma_N": {
        "expr": "-force_hammer/Area:Area:force_hammer"
    }
}
```

- **Gamma_N** : zone d'impact du marteau
- **Contrainte appliquée** : "normal_stress" est une force par unité de surface, on a donc $\sigma_n = F(t)/\text{Area}$.
- **Direction** : normale à la surface (vers l'extérieur)

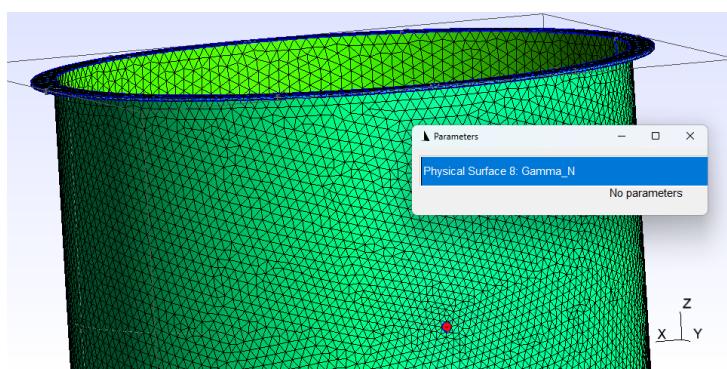


FIGURE 5.2 – Zone d'excitation Gamma_N

5.1.2 Post-traitement et monitoring

```
"PostProcess": {
    "solid": {
        "Exports": {
            "fields": [ "displacement", "acceleration" ]
        },
        "Measures": {
            "Points": {
                "Sensor1": {
                    "coord": "{0,0.475,-0.400}",
                    "fields": [ "displacement", "acceleration" ]
                }
            }
        }
    }
}.
```

Configuration de sortie :

— **Champs exportés** : déplacement et accélération pour visualisation ParaView.

On utilise le déplacement pour valider la réponse de la cuve selon son ordre de grandeur approximé (en annexe) et l'accélération qui nous sera utile pour comparer les mesures expérimentales avec les accéléromètre avec les résultat numériques.

— **Points de mesure** : Sensor1 au centre de la zone d'excitation

On pourra aussi observer les mesures sur les autres points physiques définis dans le maillage pour la comparaison avec les mesures expérimentales.

Scripts de post-traitement plusieurs scripts python ont aussi été développés pour analyser les résultats des mesures aux points capteurs des simulations entamées dont :

- Extraire les paramètres d'une simulation (Ordre de discréétisation, taille de maillage, pas de temps ..)
- Tracer les signaux extraits des différents points de mesure (déplacement et accélération)
- Parcourir les simulations existantes et comparer résultats et temps de calcul.
- Tracer les mesures expérimentales et extraire les données d'amortissement.

5.1.3 Passage à des ordres de discréétisation plus élevés

Dans l'utilisation de la **toolbox Solid**, l'étude des simulations a conduit à tester des ordres polynomiaux supérieurs à ceux initialement disponibles dans le code. L'accès à ces ordres a nécessité une recompilation manuelle de la **toolbox** après modification des définitions des espaces de fonctions afin d'autoriser des degrés plus élevés.

Cette opération augmente fortement le nombre de degrés de liberté et, par conséquent, le coût en calcul et en mémoire. Pour limiter ces surcoûts, il s'est avéré plus pertinent d'opter pour une méthode de résolution explicite, mieux adaptée au traitement de grands systèmes avec des polynômes d'ordre élevé.

5.2 Développement d'une méthode de résolution et de couplage explicite pour le problème de vibro-acoustique

5.2.1 Objectifs

Étant donné les limites atteintes en utilisant les toolboxes ; D'une part, la toolbox d'intéraction fluide-structure qui résoud les équations de Navier-Stokes alors que l'équation des ondes serait plus adaptée à notre cas.

Et d'autre part, la volonté d'améliorer le coût de calcul et la précision des résultats avec une méthode de résolution spectrale.

On s'intéresse donc à développer une application de la résolution de ce problème vibro-acoustique utilisant les outils FEEL++.

On voudrait avoir une résolution d'un système vibro-acoustique explicite en temps. Pour adopter une méthode d'éléments spectraux, on sera aussi amenés à utiliser des quadrature de Gauss-Lobatto-Legendre, et un maillage d'ordre élevé avec des éléments hexaédriques.

Il est important de souligner que la carte ALE (Annexe 7.3.5) et l'amortissement dans le solide étaient pris en compte lors du développement théorique de la méthode. Néanmoins, dû aux contraintes de temps ainsi que la possibilité de rajouter ces éléments post-facto à travers des outils spécialisés de FEEL++, le travail est centré sur la résolution et le couplage des système d'élastodynamique et d'ondes acoustique.

5.2.2 Formulation variationnelle

Soit :

— $\mathbf{v}_s \in [H^1(\Omega_s)]^3$: fonction test pour le déplacement solide,

- $q \in H^1(\Omega_f)$: fonction test pour la pression fluide,
- $f_N(t)$: fonction source appliquée sur $\Gamma_N \subset \Omega_s$.

Solide : Élastodynamique (forme faible)

Trouver $\mathbf{u}_s \in [H^1(\Omega_s)]^3$ tel que :

$$\int_{\Omega_s} \rho_s \frac{\partial^2 \mathbf{u}_s}{\partial t^2} \cdot \mathbf{v}_s dx + \int_{\Omega_s} \boldsymbol{\sigma}(\mathbf{u}_s) : \nabla \mathbf{v}_s dx = \int_{\Gamma_{fsi}} p \mathbf{n}_s \cdot \mathbf{v}_s ds + \int_{\Gamma_N} f_N(t) \mathbf{n}_s \cdot \mathbf{v}_s ds$$

Fluide : Onde acoustique (forme faible)

Trouver $p \in H^1(\Omega_f)$ tel que :

$$\int_{\Omega_f} \frac{\partial^2 p}{\partial t^2} q dx + c^2 \int_{\Omega_f} \nabla p \cdot \nabla q dx = \rho_f \int_{\Gamma_{fsi}} \frac{\partial \mathbf{u}_s}{\partial t} \cdot \mathbf{n}_s q ds$$

5.2.3 Discrétisation temporelle et schéma numérique

Pour résoudre le système couplé dans le domaine temporel, on utilise des schémas d'ordre 2 compatibles.

Domaine solide

On discrétise l'équation d'élastodynamique (Annexe 7.3.5) en démarrant avec la méthode de Newmark-beta [4.1,4.2] :

$$\begin{aligned} \mathbf{u}_s^{n+1} &= \mathbf{u}_s^n + \Delta t \dot{\mathbf{u}}_s^n + \frac{\Delta t^2}{2} (1 - 2\beta) \ddot{\mathbf{u}}_s^n + \beta \Delta t^2 \ddot{\mathbf{u}}_s^{n+1} \\ \dot{\mathbf{u}}_s^{n+1} &= \dot{\mathbf{u}}_s^n + \Delta t [(1 - \gamma) \ddot{\mathbf{u}}_s^n + \gamma \ddot{\mathbf{u}}_s^{n+1}] \end{aligned}$$

avec des valeurs usuelles $\beta = 1/4$ et $\gamma = 1/2$ assurant la stabilité inconditionnelle. \mathbf{u} étant le déplacement du solide et $\dot{\mathbf{u}}, \ddot{\mathbf{u}}$ les dérivées temporelles, la vitesse et l'accélération. Si l'on prend en compte que le système est à petites déformations, on peut approximer $\boldsymbol{\sigma}(u^{n+1}) = \boldsymbol{\sigma}(u^n)$. On obtiens le terme explicité :

$$\begin{aligned} \frac{1}{\beta \Delta t^2} \int_{\Omega_s} \rho_s \mathbf{u}_s^{n+1} \cdot \mathbf{v}_s dx &= \int_{\Omega_s} \rho_s \left[\frac{1}{\beta \Delta t^2} \mathbf{u}_s^n + \frac{1}{\beta \Delta t} \dot{\mathbf{u}}_s^n + \frac{1 - 2\beta}{2\beta} \ddot{\mathbf{u}}_s^n \right] \cdot \mathbf{v}_s dx + \int_{\Omega_s} f_N^n \cdot \mathbf{v}_s dx - \int_{\Omega_s} \boldsymbol{\sigma}(\mathbf{u}_s^n) : \nabla \mathbf{v}_s dx \\ (5.1) \end{aligned}$$

Domaine fluide

L'équation d'onde est discrétisée (Annexe 7.3.5) avec un schéma en différences centrales en temps :

$$\frac{p^{n+1} - 2p^n + p^{n-1}}{\Delta t^2} - c^2 \Delta p^n = \rho_f \int_{\Gamma_{fsi}} \frac{\partial \mathbf{u}_s}{\partial t} \cdot \mathbf{n}_s q ds$$

on obtient :

$$\int_{\Omega_f} p^{n+1} q dx = \int_{\Omega_f} (2p^n - p^{n-1}) q dx - \Delta t^2 c^2 \int_{\Omega_f} \nabla p^n \cdot \nabla q dx + \Delta t^2 \rho_f \int_{\Gamma_{fsi}} \dot{\mathbf{u}}_s^n \cdot \mathbf{n}_f q ds \quad (5.2)$$

Stratégie de couplage

À chaque pas de temps :

1. Appliquer la pression fluide p^{n+1} comme chargement de Neumann sur la structure

$$\int_{\Gamma_{fsi}(t^n)} p^{n+1} \mathbf{n}_f \cdot \mathbf{v}_s ds$$

2. Mettre à jour la structure avec le schéma de Newmark.

5.2.4 Algorithme : Avancement en temps pour le problème vibro-acoustique couplé

Donné :

- Le déplacement structural $\mathbf{u}_s(t)$ dans Ω_s ,
- La pression acoustique $p(t)$ dans $\Omega_f(t)$,

Procéder comme suit :

1. Initialisation :

$$\mathbf{u}_s^0, \quad \dot{\mathbf{u}}_s^0, \quad \ddot{\mathbf{u}}_s^0, \quad p^0, \quad \dot{p}^0 = 0$$

2. Pour chaque pas de temps $n = 0, 1, \dots, N - 1$:

- (a) Résoudre le problème fluide :

Trouver $p^{n+1} \in H^1(\Omega_f(t^n))$ dans [5.2.3](#)

- (b) Calculer la charge de pression sur la structure :

Interpoler p^{n+1} sur $\Gamma_{fsi}^s \subset \partial\Omega_s$.

$$\int_{\Gamma_{fsi}(t^n)} p^{n+1} \mathbf{n}_f \cdot \mathbf{v}_s \, ds$$

- (c) Résoudre l'élastodynamique dans la structure :

Trouver $\mathbf{u}_s^{n+1} \in [H^1(\Omega_s)]^3$ [5.2.3](#)

Utiliser le schéma d'intégration temporelle Newmark-beta pour mettre à jour : $\mathbf{u}_s^{n+1}, \dot{\mathbf{u}}_s^{n+1}, \ddot{\mathbf{u}}_s^{n+1}$.

3. Incrémenter le temps : $t^{n+1} = t^n + \Delta t$

Cet algorithme a été implémenté avec des classes C++ développées avec les outils de résolution d'EDP de FEEL++.

Les classes Elastic et Wave

Ces deux classes résolvent le système physique associé respectivement au solide et au fluide suivant les marqueurs présents dans le maillage.

Comme pour les toolboxes, on utilise un système de configuration, .cfg + .json pour les spécifications de la simulation.

Le but étant d'abord de résoudre ces systèmes linéaires avec une méthode explicite. En utilisant les classes de Newmark et BDF, on résout, à chaque pas de temps, le système :

$$\begin{cases} A_{disp} \cdot u^{n+1} = l(u^n, \dot{u}^n, \ddot{u}^n) \\ A_{pres} \cdot p^{n+1} = l(p^n, p^{n-1}) \end{cases}$$

Il est important de noter qu'à ce stade du projet certaines difficultés à expliciter la résolution de l'équation d'élasticité sont apparues. En effet, l'explicitation se base sur une approximation sur le terme de raideur, on suppose de petites déformations et on prend $\sigma(u^{n+1}) = \sigma(u^n)$.

La résolution explicite est instable même avec des petits pas de temps (jusqu'à $10^{-9}s$). Il sera peut-être possible d'y remédier lorsque sera implanté la diagonalisation de la matrice de masse et l'amortissement.

Couplage

Le couplage est fait à partir d'une troisième classe qui applique les conditions de couplage sur les deux systèmes à travers la boucle de picard [5.2.4](#). On peut lancer les simulations mono-physiques à partir de cette classe.

Algorithme de Picard

```

Input:  $\varepsilon$  tolérance,  $k_{\max}$  itérations max
Output:  $u^{n+1}, p^{n+1}$ 
 $u^{n,0} \leftarrow \text{Elastic.GetCurrentDisp}();$ 
 $v^{n,0} \leftarrow \text{Elastic.GetCurrentVelocity}();$ 
 $p^{n,0} \leftarrow \text{Wave.getCurrentPressure}();$ 
 $k \leftarrow 0;$ 
 $converged \leftarrow \text{faux};$ 
while  $\neg converged$  et  $k < k_{\max}$  do
     $k \leftarrow k + 1;$ 
     $\text{Elastic.Solve}(p^{n,k-1});$ 
     $\text{Wave.Solve}(v^{n,k-1});$ 
     $u^{n,k} \leftarrow \text{Elastic.GetCurrentDisp}();$ 
     $p^{n,k} \leftarrow \text{Wave.getCurrentPressure}();$ 
    if  $\|u^{n,k} - u^{n,k-1}\| < \varepsilon$  et  $\|p^{n,k} - p^{n,k-1}\| < \varepsilon$  then
        |  $converged \leftarrow \text{vrai};$ 
    end
end
Elastic.Update();
Wave.Update();

```

Algorithm 1: Algorithme de Picard (schéma de couplage itératif)

Principe des mesures sur Γ_N .

Pour caractériser la réponse vibro-acoustique du système, différentes quantités intégrées sont calculées sur le domaine complet Ω et sur la surface de Neumann Γ_N où l'excitation est appliquée. Ces mesures permettent de quantifier le comportement global du système sous sollicitation.

On définit les quantités suivantes :

— **Quantité totale dans le domaine** : Intégrale du déplacement sur tout le volume

$$Q_\Omega(t) = \int_\Omega u_h(t, \mathbf{x}) \, d\mathbf{x}$$

— **Déplacement moyen dans le domaine** : Valeur moyenne spatiale

$$\bar{u}_\Omega(t) = \frac{1}{|\Omega|} \int_\Omega u_h(t, \mathbf{x}) \, d\mathbf{x} = \frac{Q_\Omega(t)}{|\Omega|}$$

— **Quantité totale sur la surface** : Intégrale du déplacement sur Γ_N

$$Q_{\Gamma_N}(t) = \int_{\Gamma_N} u_h(t, \mathbf{x}) \, dS$$

— **Déplacement moyen sur la surface** : Valeur moyenne sur Γ_N

$$\bar{u}_{\Gamma_N}(t) = \frac{1}{|\Gamma_N|} \int_{\Gamma_N} u_h(t, \mathbf{x}) \, dS = \frac{Q_{\Gamma_N}(t)}{|\Gamma_N|}$$

— **Flux normal de référence** : Flux total à travers la surface de Neumann

$$\Phi_{\Gamma_N}(t) = \int_{\Gamma_N} \nabla u_h(t, \mathbf{x}) \cdot \mathbf{n} \, dS$$

— **Extrêmes du déplacement** : Valeurs minimale et maximale dans le domaine

$$u_{\min}(t) = \min_{\mathbf{x} \in \Omega} u_h(t, \mathbf{x}), \quad u_{\max}(t) = \max_{\mathbf{x} \in \Omega} u_h(t, \mathbf{x})$$

Le suivi temporel de ces quantités fournit une caractérisation de la réponse dynamique du système, permettant d'analyser le comportement local sous excitation.

Cependant, pour accéder à des mesures locales plus représentatives du déplacement (et, par différentiation temporelle, de l'accélération) réellement induit par l'excitation, il serait nécessaire d'implémenter un suivi en un point physique précis, identifié dans le fichier `.geo`. Cette stratégie permettrait d'obtenir des signaux ponctuels directement comparables à des mesures expérimentales.

5.3 Ressources de calcul

5.3.1 Supercalculateur Gaya

Les simulations numériques ont été effectuées sur le supercalculateur *Gaya* de l'établissement, géré par l'ordonnanceur `Slurm`.

Configuration du cluster

- **Noeuds disponibles** : 6 noeuds CPU (gaya1–gaya6) et 1 noeud GPU (gaya-gpu).
- **Partitions** :
 - **production** (par défaut) : 4 noeuds CPU hautes performances.
 - **public et debug** : accès à l'ensemble des noeuds CPU.
 - **gpu** : 1 noeud GPU dédié.

Spécifications des noeuds CPU (gaya1–gaya6)

- Architecture : x86_64, double socket NUMA.
- Processeurs : 2 × AMD EPYC (famille 23, modèle 49).
- Cœurs : 64 cœurs physiques par socket, 2 threads par cœur, soit 256 threads par noeud.
- Mémoire vive : 515 Go par noeud.
- Système d'exploitation : Linux 5.15.

Spécifications du noeud GPU (gaya-gpu)

- Cœurs : 32 threads (2 × 16 cœurs physiques, SMT désactivé).
- Mémoire vive : 257 Go.
- Accélérateurs : 3 × AMD Instinct MI210.

5.3.2 Parallélisation et usage mémoire

Le supercalculateur *Gaya* est géré par l'ordonnanceur **Slurm** (*Simple Linux Utility for Resource Management*). Slurm permet de planifier et d'exécuter des tâches de calcul intensif sur un ensemble de noeuds en allouant dynamiquement les ressources (CPU, mémoire, GPU). L'utilisateur soumet un fichier de script contenant les directives **#SBATCH**, qui précisent notamment :

- le nombre de noeuds et de tâches MPI à utiliser ;
- la partition ciblée (**production**, **debug**, etc.) ;
- la durée maximale d'exécution (**-time**) ;
- allouer l'intégralité du noeud avec l'option **-exclusif** ;
- les paramètres d'affinité et de liaison aux cœurs (maillage avec nombre de partitions égal au nombre de tâches).

Partitionnement du maillage

Dans ce projet, les simulations utilisent des maillages volumineux, générés et partitionnés en amont pour optimiser l'exécution parallèle. Le maillage est découpé en un nombre de sous-domaines égal au nombre de processus MPI (**mpirun -n**). Chaque processus gère alors uniquement la portion du maillage qui lui est assignée, ce qui réduit la charge mémoire individuelle et permet une exécution simultanée sur plusieurs cœurs ou noeuds.

Suivi de l'usage mémoire

L'usage mémoire a été suivi à l'aide de l'interface **PETSc**. La fonction :

`PetscMemoryGetCurrentUsage()`

permet de récupérer, pour chaque processus MPI, la quantité de mémoire actuellement utilisée (en octets). Cette mesure, enregistrée durant la simulation, permet d'identifier les phases les plus consommatrices en mémoire et d'évaluer l'impact du partitionnement du maillage sur la consommation totale.

Par ailleurs, les blocs de code correspondant aux étapes clés de la simulation :

- La construction du système algébrique.
 - La résolution du système algébrique.
 - Le post-traitement (exporteurs de fichiers de simulations, de mesures physiques et de temps de calculs)
- sont entourés de fonctions *timer* afin de mesurer le temps de calcul à chaque pas de temps. Les résultats de ces mesures sont ensuite exportés dans des fichiers **.data**, destinés à l'exploitation et à l'analyse dans le cadre du benchmarking.

Résumé

L'utilisation combinée de Slurm pour l'allocation des ressources, du partitionnement de maillage pour la distribution du calcul, et du suivi mémoire via les timer et PETSc, garantit un usage efficace des ressources du cluster et un contrôle précis des performances.

Chapitre 6

Présentation des résultats

6.1 Mesures expérimentales de la phase d'apprentissage - Interprétation des signaux

Pour comprendre les résultats des mesures prises durant [4.5](#) de la phase d'apprentissage, on observe les sorties pour les comparer aux mesures numériques.

Monsig : Accéléromètres

Les signaux **Monsig** sont enregistrés à l'aide de deux accéléromètres par point d'excitation (OEM KS90B / KS901B, sensibilité 100 mV/g).

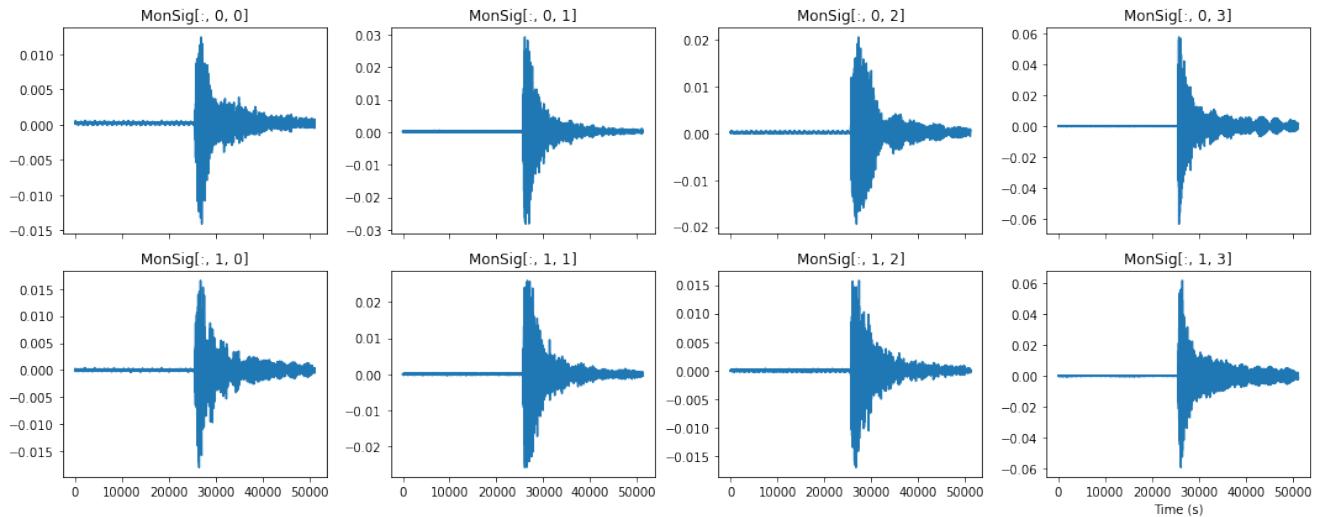


FIGURE 6.1 – Signaux **Monsig** enregistrés lors de la phase d'apprentissage. Colonnes : zones de surveillance ; lignes : accéléromètres associés.

L'axe vertical des graphiques est exprimé en volts. En tenant compte de la sensibilité, une tension de V [V] correspond à :

$$a = V \times \frac{1000}{100} \times 9.81 \quad [\text{m/s}^2]$$

Ainsi, 0.01 V correspond à environ 0.98 m/s^2 . Les mesures observées varient généralement entre 1 et 6 m/s^2 .

Outsig : Marteau d'impact

Les signaux **Outsig** proviennent d'un marteau d'impact (ICP Impact Hammer, sensibilité 22.5 mV/N).

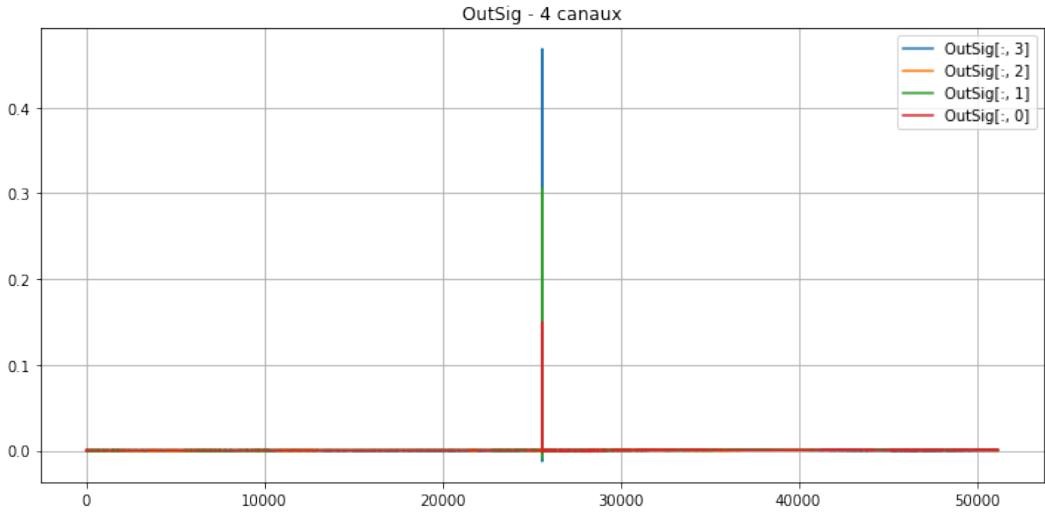


FIGURE 6.2 – Signaux **Outsig** de la phase d’apprentissage numérique. Chaque canal représente la force exercée lors de l’impact sur une zone de surveillance.

La conversion tension-force se fait par :

$$F = V \times \frac{1000}{22.5} \quad [\text{N}]$$

Par exemple, 0.1 V correspond à environ 4.44 N. Les impacts mesurés se situent entre 4.44 et 17.76 N.

Lien force–accélération

On observe qu’un impact d’environ 1 N sur la paroi génère une accélération d’environ $\frac{6}{17.76} \approx 0.34 \text{ m/s}^2$ mesurée au bas de la cuve.

6.2 Méthodes de validation

Compte tenu de la complexité physique et géométrique du problème, il est difficile d’obtenir une solution analytique. La validation repose donc sur trois approches :

- trouver l’ordre de grandeur du déplacement par analogie avec un problème plus simple (Annexe 7.3.2) ;
- confrontation avec des données expérimentales ;
- comparaison avec une simulation numérique de référence à ordre de précision plus élevé.

Erreur L^2

Soient $u_{\text{num}}(t)$ la solution numérique et $u_{\text{ref}}(t)$ la solution de référence. L’erreur relative en norme L^2 est définie par :

$$E_{L^2} = \frac{\|u_{\text{num}} - u_{\text{ref}}\|_{L^2}}{\|u_{\text{ref}}\|_{L^2}} = \frac{\left(\sum_{i=1}^N |u_{\text{num}}(t_i) - u_{\text{ref}}(t_i)|^2\right)^{1/2}}{\left(\sum_{i=1}^N |u_{\text{ref}}(t_i)|^2\right)^{1/2}}$$

où t_i sont les instants d’échantillonnage.

- Mesure l’écart quadratique moyen entre deux signaux ou champs.
- Plus E_{L^2} est petit, plus les signaux sont proches point à point en amplitude.
- Pour comparer la précision d’une simulation à une mesure ou à une solution de référence.

Intercorrélation de Pearson

Pour deux signaux discrets x_i et y_i , le coefficient d’intercorrélation de Pearson mesure leur similarité (corrélation linéaire) :

$$r = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}}$$

où \bar{x} et \bar{y} sont les moyennes des signaux.

- Mesure le degré de similarité de forme entre deux signaux, indépendamment de l'échelle ou de l'offset.
- $r \approx 1$: formes identiques (corrélation parfaite positive)
- $r \approx -1$: formes identiques mais inversées (corrélation parfaite négative)
- $r \approx 0$: absence de corrélation linéaire

Pour vérifier si deux signaux ont la même évolution temporelle ou spatiale, même avec des amplitudes différentes.

6.3 Résultats numériques

6.3.1 Résultats des simulations avec les toolboxes

Tendances de l'évolution des taux d'erreurs pour les simulations

Pour appliquer la technique du retournement temporel, il nous faut une durée de signal de 0.5 s. En prenons en compte l'amortissement et en observant les mesures expérimentales, les vibrations de la structure tombent en dessous du seuil des capteurs vers 0.25 s (plus courte pour les tests avec la cuve remplie d'eau comme vu dans la figure 6.3) .

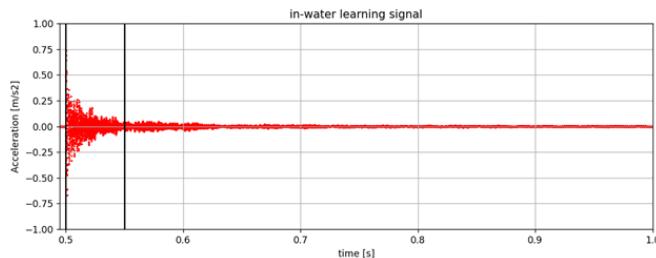


FIGURE 6.3 – Coda mesurée lors des tests expérimentaux : cuve avec eau.

Comme on peut voir sur la figure 6.4 les erreurs se comportent d'une manière similaire. Un saut lors du premier pic du déplacement puis une stabilité. La corrélation de Pearson se dégrade avec le temps de simulation, proche de 1 pour les plus courtes durées.

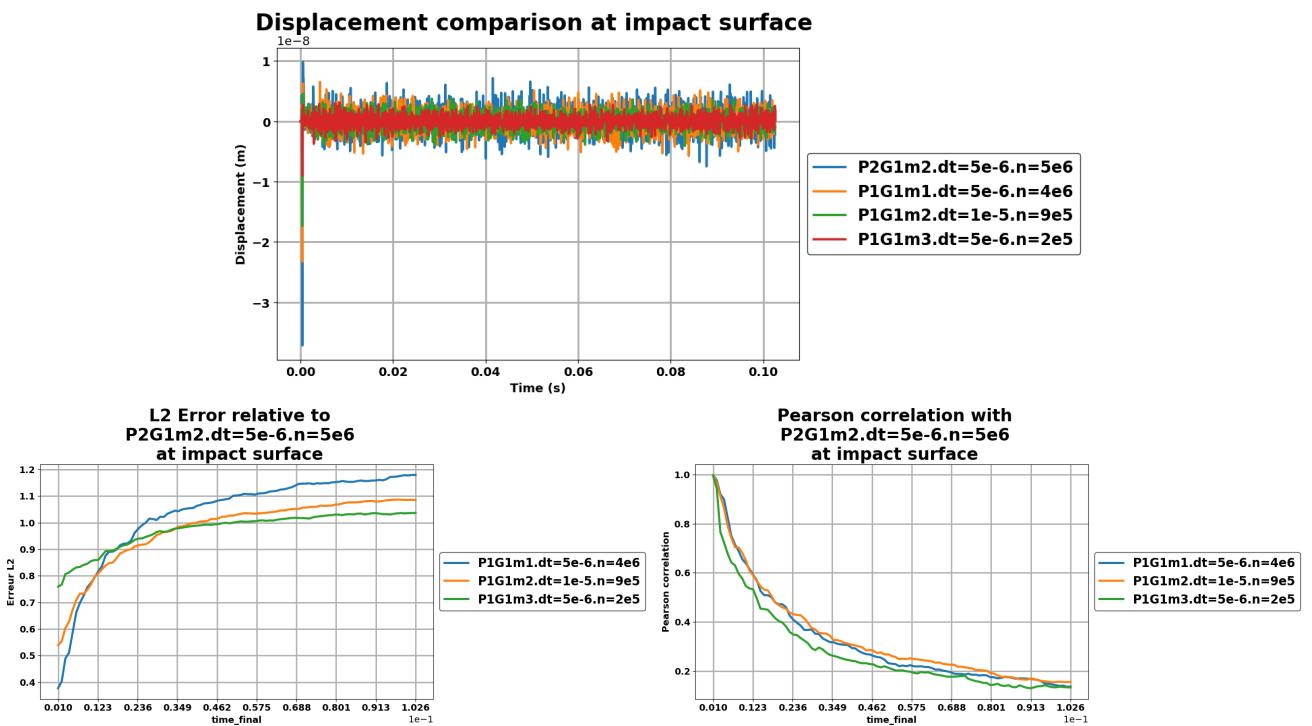


FIGURE 6.4 – Comparaison de simulations.

Dû aux tailles des problèmes et des énormes temps de calcul la simulation de référence principale est restreinte à $t = 2.5$ ms, assez de temps pour que la première onde traverse la cuve (Annexe 7.3.4).

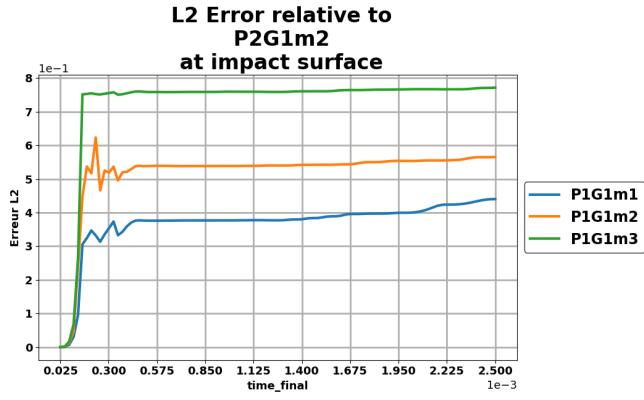


FIGURE 6.5 – Comparaison de simulations sur une plus courte durée.

Comprendre les graphes :

- $P_n G_m$ = ordre de discréttisation polynomial et géométrique
- m1, m2, m3 = tailles des maillages (3, 6, 12 mm)
- dt = pas de temps
- n = nombre de degrés de liberté

On compare les mesures extraites des signaux sur un même point du maillage des différentes configurations de simulations en les superposant.

On regarde l'évolution des erreurs L2 et de l'intercorrélation sur la durée commune entre les simulations. On peut observer les simulations sur d'autres points définis dans le fichiers de configuration. On remarque que plus on s'éloigne de la zone d'impact plus les écarts entre les simulations sont importants, mais le comportement reste le même.

Naturellement, plus on a de temps de simulation, plus l'erreur augmente. Mais aussi, plus il y a d'écart lors du pic initial du déplacement plus l'erreur par rapport à la référence sera grande.

On peut donc observer les premières oscillations et comparer l'évolution de l'erreur pour cette durée, pour les différentes configurations. Ce là montrerait que plus l'ordre d'approximation est grand, moins il y a de différences entre les simulations.

6.3.2 Évaluation de la précision des simulations

L'évaluation compare les mesures aux points capteurs selon :

- Variation de la taille de maillage pour un ordre de discréttisation donné.
- Variation de l'ordre de discréttisation pour un maillage donné.
- Évolution de l'erreur L^2 et de l'intercorrélation de Pearson.

Analyse :

Trois graphiques montrent, pour chaque ordre P_1, P_2, P_3 , l'impact des tailles de maillage m_1, m_2, m_3 .

Inversement, trois autres graphiques comparent P_1, P_2, P_3 pour chaque maillage fixe.

Passage à l'ordre de maillage G_2 et les gains en précision par rapport à G_1 .

Regard général sur les simulations les plus fines pour trouver le meilleur compromis coût de calcul vs précision.

Paramètres fixés des simulations

- 128 partitions sur un noeud *gaya* -exclusive(5.3.2), ou 256 partitions sur deux noeuds pour les plus grands cas.
- Pas de temps : D'après les critères de stabilité (Annexe 7.3.3), un pas de temps entre 10^{-7} et 10^{-5} s pour bien capturer le comportement du système.
- Schéma temporel : Newmark d'ordre 2.
- Temps de calcul : 1 s par $\Delta t \approx 1.1$ jours pour 0.5 s de signal avec $\Delta t = 5 \times 10^{-6}$ s.
- Référence : $P4G2m2$ (plus grand nDOF), 198.7 s par pas de temps, 0.00256 s de signal. Très coûteux et inefficace (33 min d'initialisation, swapping mémoire).
- Erreurs : L^2 relatives à la référence, intercorrélation proche de 1 pour les signaux courts.

Impact de la taille de maillage selon l'ordre de discréttisation

Cette analyse présente trois graphiques distincts, chacun correspondant à un ordre de discréttisation fixe (P_1, P_2, P_3). Pour chaque graphique, l'effet de la variation de la taille de maillage (m_1, m_2, m_3) est examiné sur la qualité des résultats. Des graphiques complémentaires illustrent l'évolution des erreurs associées à chaque configuration.

	P1	P2	P3	P4	P5
3mm (m1)	3.51M	21.42M	64.05M	-	-
6mm (m2)	897k	5.45M	16.44M	36.46M	-
12mm (m3)	230k	1.38M	4.28M	9.528M	17.87M

TABLE 6.1 – Nombre de degrés de liberté selon P_n et la taille de maillage

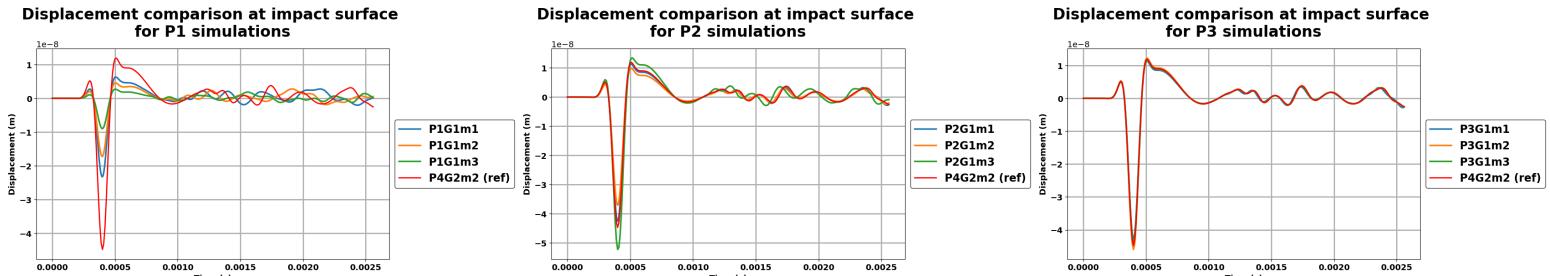


FIGURE 6.6 – Comparaison des signaux pour différentes tailles de maillage selon $P1$, $P2$, et $P3$.

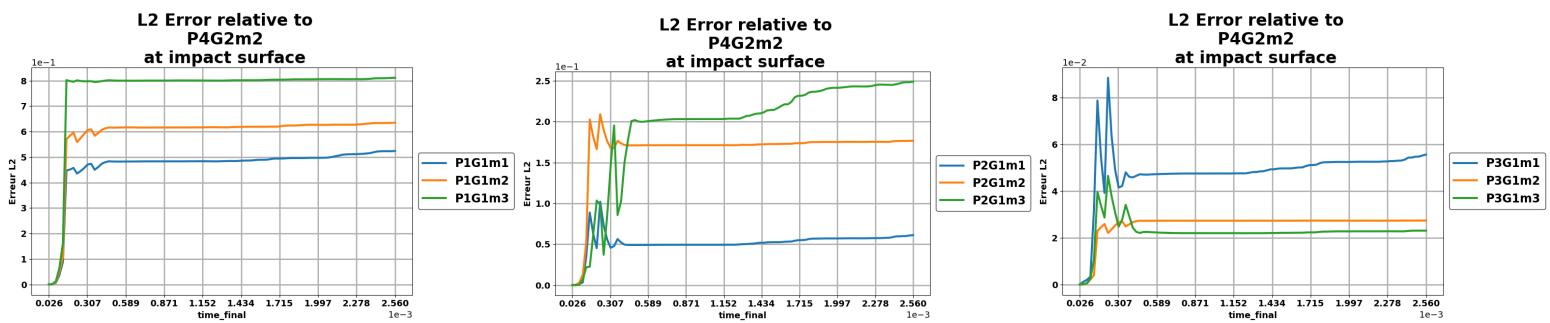


FIGURE 6.7 – Évolution de l'erreur L^2 associée aux tailles de maillage pour $P1$, $P2$, et $P3$.

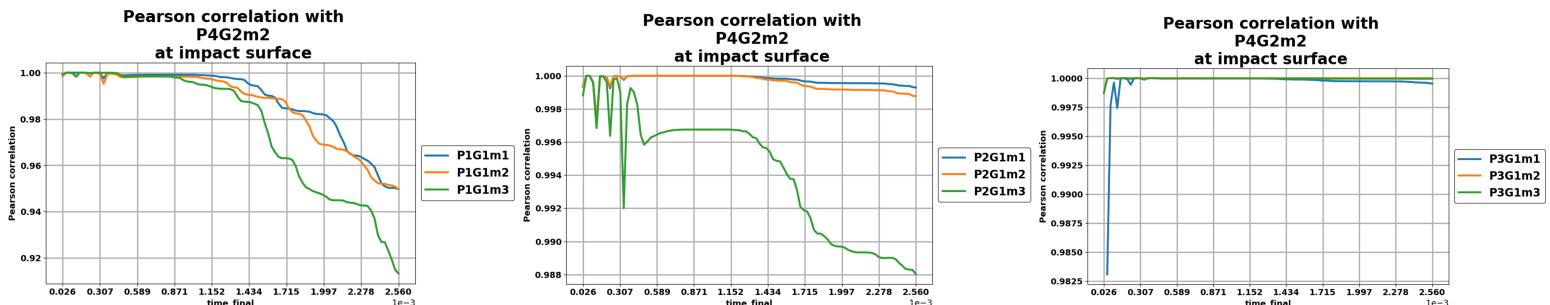


FIGURE 6.8 – Évolution de l'intercorrélation de Pearson associée aux tailles de maillage pour $P1$, $P2$, et $P3$.

Pour chaque ordre P_n , l'augmentation de la finesse de maillage réduit systématiquement l'erreur L^2 et maintient une corrélation plus élevée avec la référence. Le maillage le plus fin ($m1$) reste globalement le plus proche du cas $P4$ pour les cas $P1$ et $P2$, mais conserve un écart d'amplitude initial notable. La discrétisation $P3$ a la dégradation la moins importante du taux d'erreur et de corrélation, mais on remarque que raffiner le maillage n'est plus synonyme de meilleure précision.

Comparaison des ordres de discrétisation pour maillage fixe

Pour chaque taille de maillage fixée ($m1$, $m2$, $m3$), trois graphiques comparant les performances des ordres de discrétisation $P1$, $P2$, et $P3$. Cette approche permet d'évaluer l'influence de l'augmentation de l'ordre sur la précision des simulations.

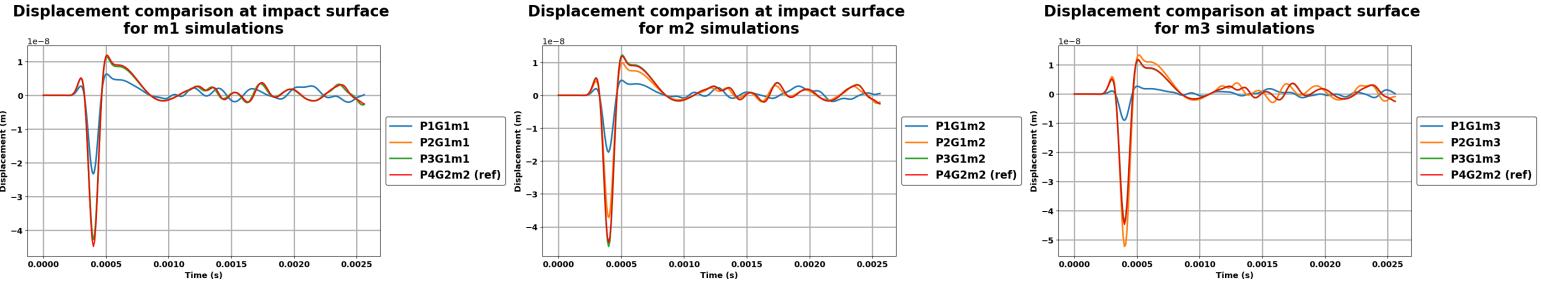


FIGURE 6.9 – Comparaison des signaux selon les tailles de maillage $m1$, $m2$, $m3$ pour chaque ordre $P1$, $P2$, et $P3$

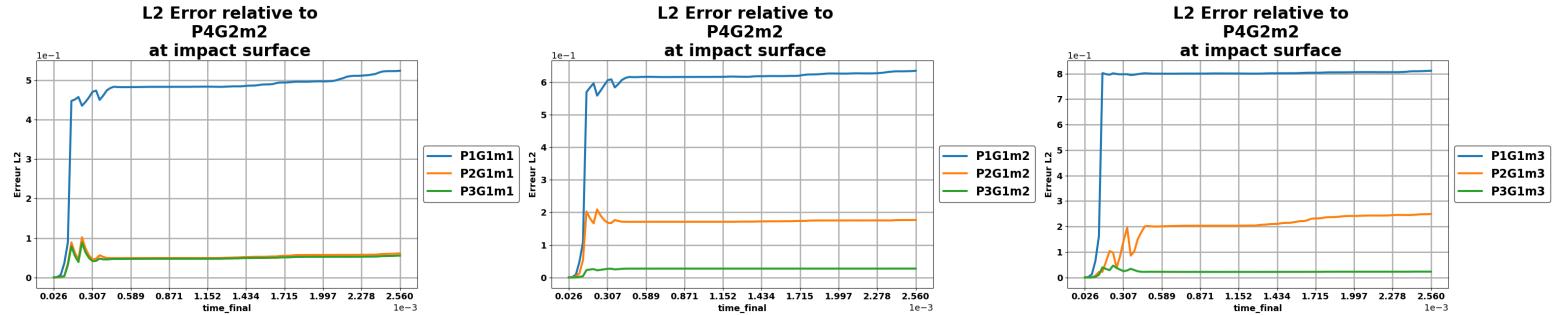


FIGURE 6.10 – Évolution de l'erreur L^2 associée aux ordres de discréétisation pour $m1$, $m2$, et $m3$.

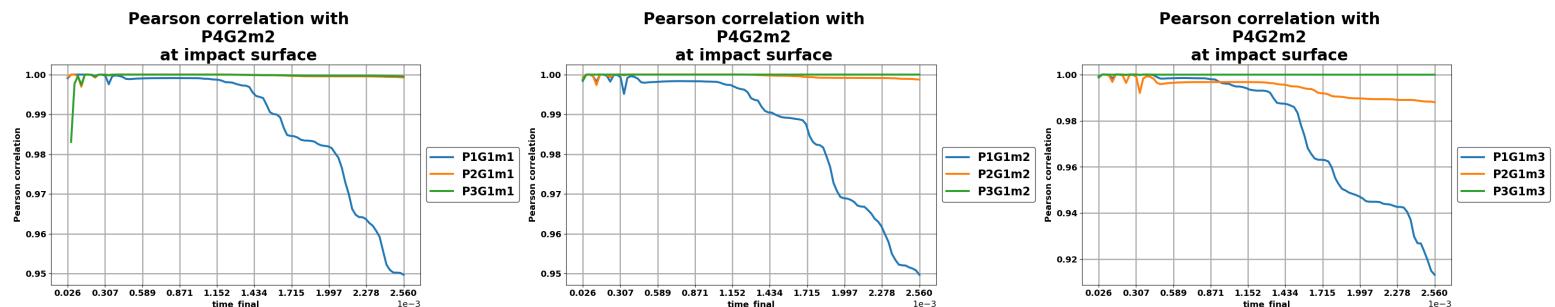


FIGURE 6.11 – Évolution de l'intercorrélation de Pearson associée aux ordres de discréétisation pour $m1$, $m2$, et $m3$.

Dans les 3 cas on peut observer que les courbes des erreurs suivent la même structure. Plus le maillage est grossier plus l'erreur pour la discréétisation $P1$ est grande. Cet impact est moins important plus en monte en ordre de discréétisation.

Plus l'ordre de discréétisation est haut moins il y a de gains en précision par raffinement du maillage. Supposément, augmenter l'ordre de discréétisation du maillage pour avoir des éléments plus compatibles avec l'ordre de discréétisation des espaces de fonctions, offrirait une meilleure précision avec un maillage à finesse modérée qu'un maillage de bas ordre plus fin.

Analyse comparée des discréétisations $G1$ et $G2$

Un graphique illustre les signaux de sortie obtenus avec la discréétisation $G1$, tandis qu'un autre présente ceux obtenus avec $G2$. Un graphique supplémentaire compare les erreurs relatives entre ces deux méthodes, mettant en évidence le gain en précision offert par la discréétisation quadratique ($G2$).

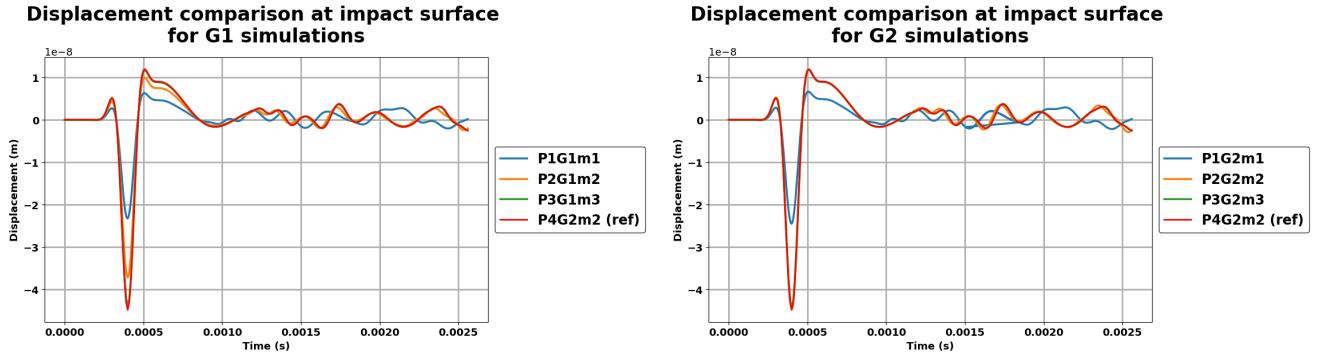


FIGURE 6.12 – Signaux simulés pour les discrétisations $G1$ (gauche) et $G2$ (droite).

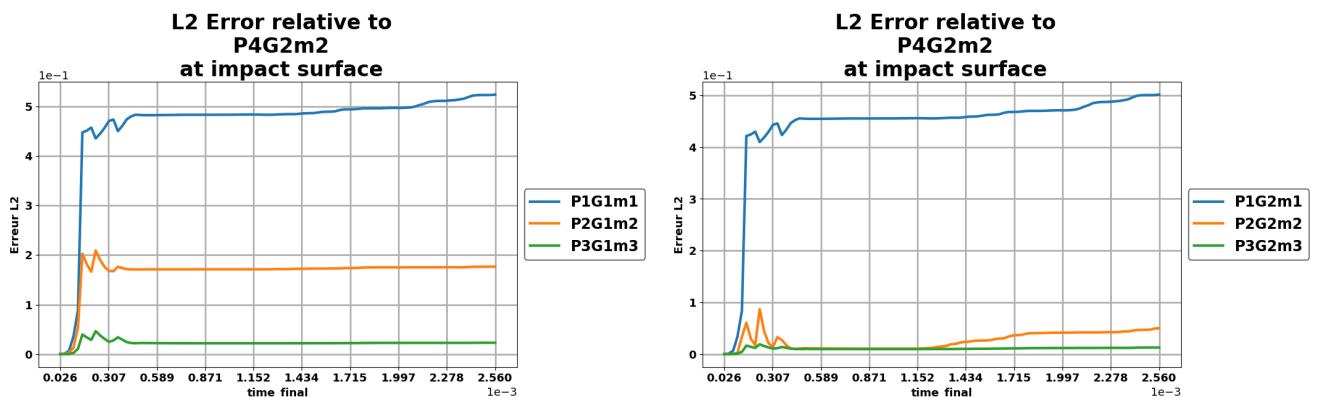


FIGURE 6.13 – Comparaison des erreurs relatives entre discrétisations $G1$ et $G2$.

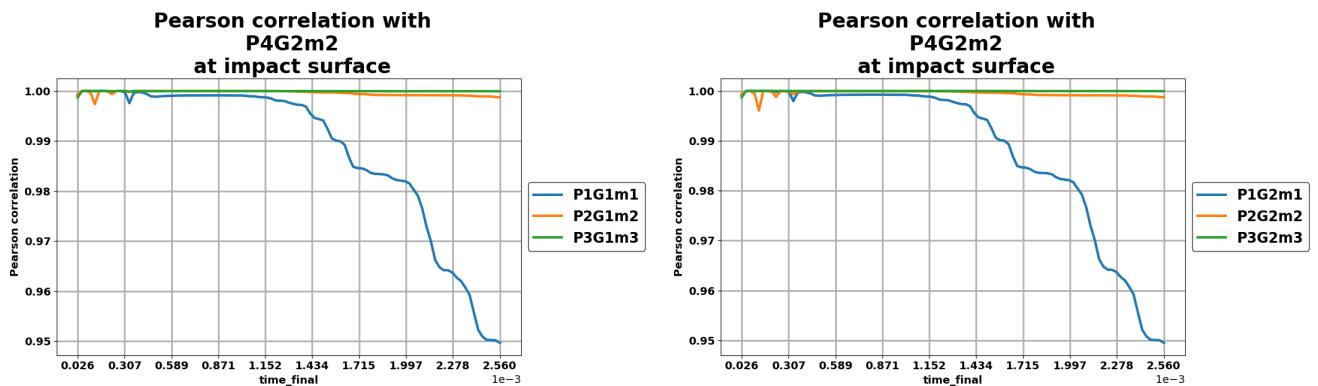


FIGURE 6.14 – Comparaison des intercorrélations entre discrétisations $G1$ et $G2$.

On remarque d'abord qu'un ordre de discrétisation du maillage plus haut n'est pas très bénéfique pour une discrétisation linéaire (P1). Par contre, il est clair que les solutions approximées dans un espace d'ordre supérieur gagnent en précision avec une iso-paramétrisation avec l'ordre du maillage.

Simulations les plus fines vs référence

Une série de simulations a été réalisée depuis le début du stage dans le cadre d'observer les effets des différentes configurations. Une gestion de ces simulations (allant à une centaine) a été nécessaire. Certaines de ces simulations consomment énormément de temps de calcul et prennent beaucoup de place mémoire.

Les simulations les plus coûteuses ont été réalisées avec 256 partitions partagées sur 2 chœurs de la machine Gaya.

Voici les simulations les plus précises réalisées comparées à la simulation de référence. Ceci servira d'étude du meilleur compromis coût de calcul / précision.

Comparaison au capteur 1 (zone d'excitation) :

— $P_n G_m$ = ordre de discrétisation

- Solver, ici GAMG
- m1, m2, m3 = tailles des maillages (3, 6, 12 mm)
- dt = pas de temps
- tol = tolérance relative du solver
- n = nombre de degrés de liberté
- perf = durée de calcul par pas de temps

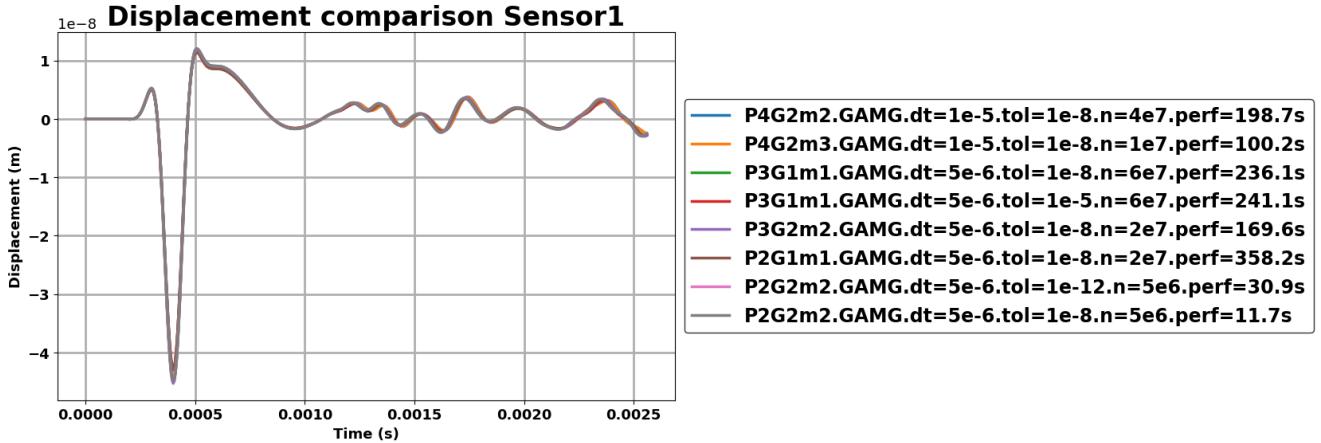


FIGURE 6.15 – Signaux simulés au capteur 1

Ce graphique montre que les différentes configurations fines reproduisent globalement les mêmes tendances temporelles du déplacement au point d'impact pour cette durée de simulation. Les déplacements initiaux sont similaires comparées aux solution de plus bas ordre.

Sensor1 L2 Error relative to P4G2m2.GAMG.dt=1e-5.tol=1e-8.n=4e7.perf=198.7s

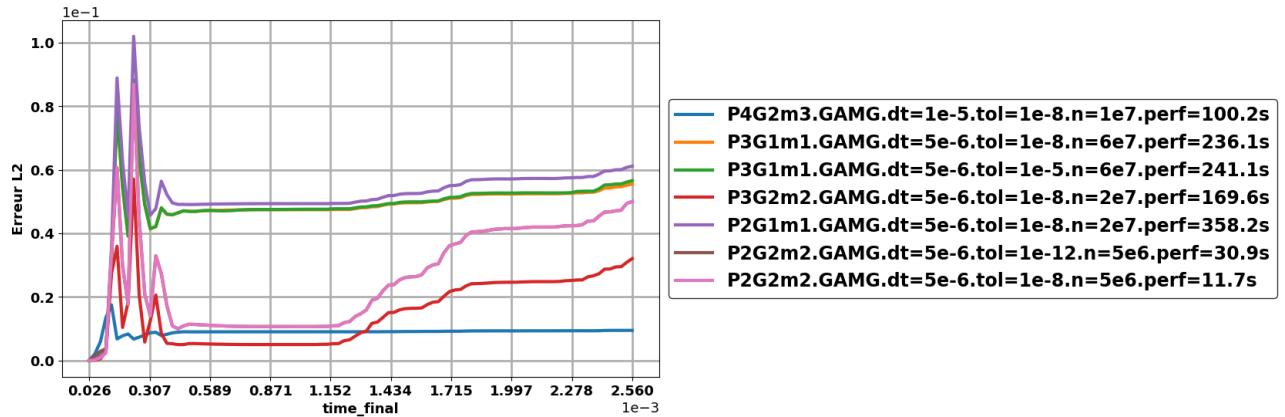


FIGURE 6.16 – Erreur relative au capteur 1

Configuration	Temps calcul / Δt	Erreur relative
P4G2m2 (256 partitions)	198.7 s	référence
P4G2m3 (256 partitions)	100.2 s	0.95%
P3G1m1 (10^{-8})	236.1 s	5.56%
P3G1m1 (10^{-5})	241.1 s	5.66%
P3G2m2	169.6 s	3.21%
P2G1m1	358.2 s	6.12%
P2G2m2 (10^{-12})	30.9 s	5.00%
P2G2m2 (10^{-8})	11.7 s	5.00%

TABLE 6.2 – Erreur relative et coût temporel par pas de temps pour différentes configurations

La table souligne le compromis entre temps de calcul et précision. Certaines configurations (ex : P4G2m3) offrent une forte réduction du temps de calcul pour une perte minime en précision. D'autres (ex : P2G1m1) sont coûteuses avec des erreurs plus élevées, soulignant l'importance du choix judicieux de la discréétisation et du maillage.

Solution la moins coûteuse pour une erreur fixée

Objectif : identifier la configuration (P_n, G_m, m_k) minimisant le coût computationnel pour un seuil d'erreur donné. Ce critère est essentiel pour optimiser les ressources et le temps.

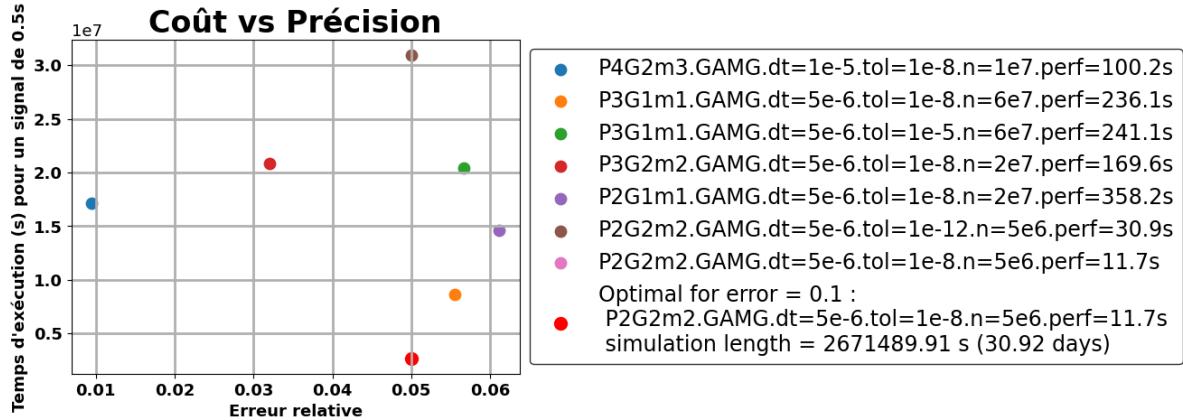


FIGURE 6.17 – Solutions optimales selon compromis coût vs précision

Cette analyse valide le choix d'une discréétisation quadratique iso-paramétrique ($P2G2$) avec un maillage moyen ($m2$) comme meilleur compromis coût/précision pour le modèle étudié.

6.3.3 Comparaison avec les mesures expérimentales

On choisit le signal sortie équivalent au point d'impact 4 (c'est-à-dire sur la paroi de la cuve). On multiplie l'amplitude de ce signal par la force d'impact du marteau pour ce teste (17.76 N) car nous sommes dans un cas linéaire et on rajoute l'amortissement évalué depuis les résultats expérimentaux (Annexe 7.3.6).

Signaux d'accélération aux capteurs

Accélération normale - Comparaison multi-capteurs

La figure 6.18 montre l'évolution temporelle de l'accélération normale mesurée simultanément par les six capteurs disposés sur la structure. Cette configuration permet d'analyser la propagation des ondes et la cohérence des mesures entre les différents points de la cuve.

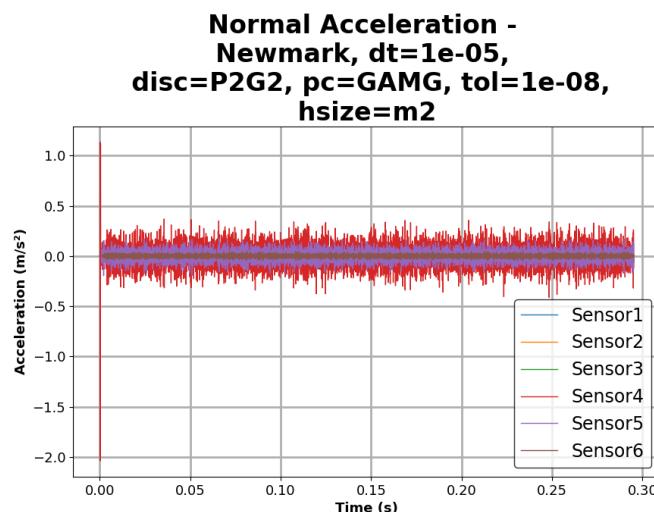


FIGURE 6.18 – Accélération normale mesurée par les six capteurs

Paramètres de simulation :

- Schéma temporel : Newmark avec $\Delta t = 10^{-5}$ s
- Discrétisation spatiale : Éléments P2G2 (polynômes d'ordre 2)
- Solveur : GAMG (Geometric Algebraic Multigrid)
- Tolérance : 10^{-8}
- Maillage : Taille caractéristique m2 (6 mm)
- Zone d'impact : Point 4

Les figures suivantes présentent les signaux d'accélération mesurés à la position correspondantes aux accéléromètres sur la cuve pendant la prise des mesures expérimentales :

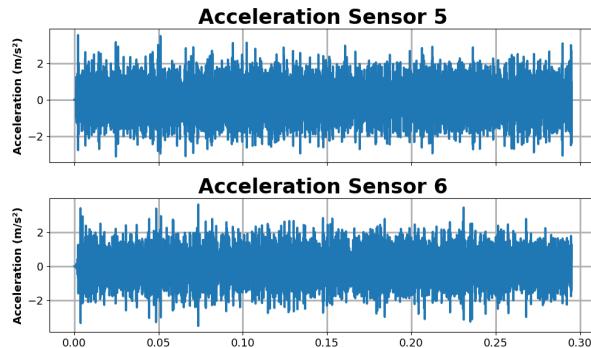


FIGURE 6.19 – Signaux d'accélération mesurés et numérique aux points 5 et 6 (positions des accéléromètres

Comparaison signal mesuré vs numérique :

La figure 6.20 présente la comparaison directe entre le signal expérimental et le signal numérique avec amortissement au capteur .

Comparaison mesures expérimentales et signal numérique amorti au Capteur 1

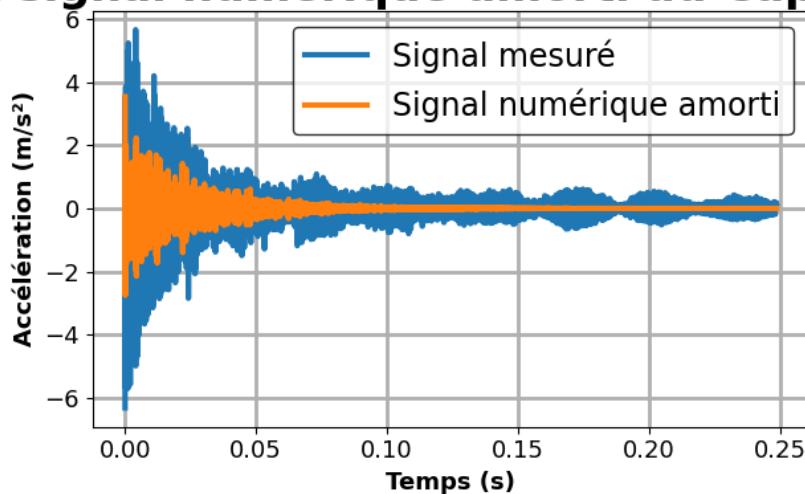


FIGURE 6.20 – Comparaison entre le signal mesuré expérimentalement et le signal numérique avec amortissement au Capteur 1

Analyse en zoom temporel : Pour une analyse plus fine de la corrélation entre les signaux, la figure 6.21 présente un zoom sur les oscillations. Les deux types d'amortissement numérique sont comparés :

- **Amortissement FFT** : Basé sur l'analyse fréquentielle du signal expérimental
- **Amortissement logarithmique** : Calculé par décrément logarithmique des pics d'oscillation

Zoom sur le signal

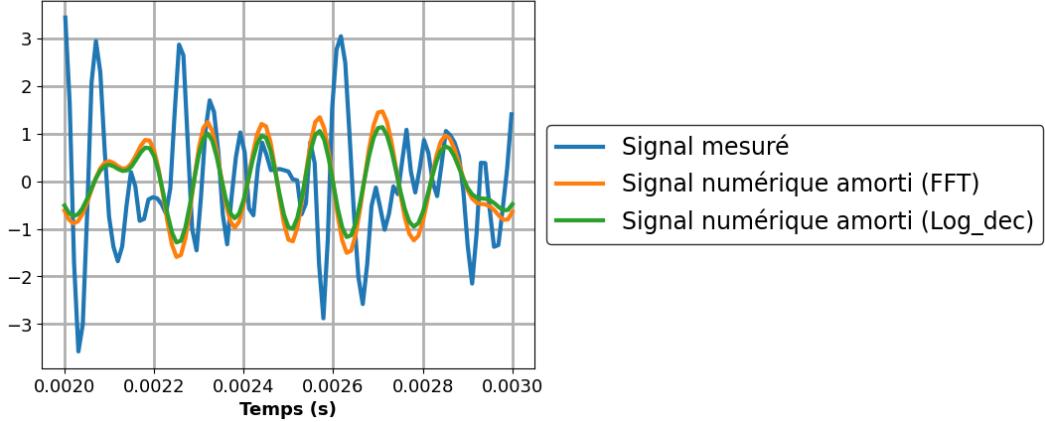


FIGURE 6.21 – Zoom sur le signal - Comparaison des méthodes d'amortissement numérique avec la mesure expérimentale

On a un ordre de grandeur similaire mais ces résultats démontrent principalement la mise en place d'un moyen de comparaison des mesures numériques avec les mesures expérimentales. Une comparaison plus poussée demande plus de travail à faire au niveau des conditions limites, la géométrie et la méthode de résolution.

Extraits d'une simulation en 3D en utilisant la toolbox

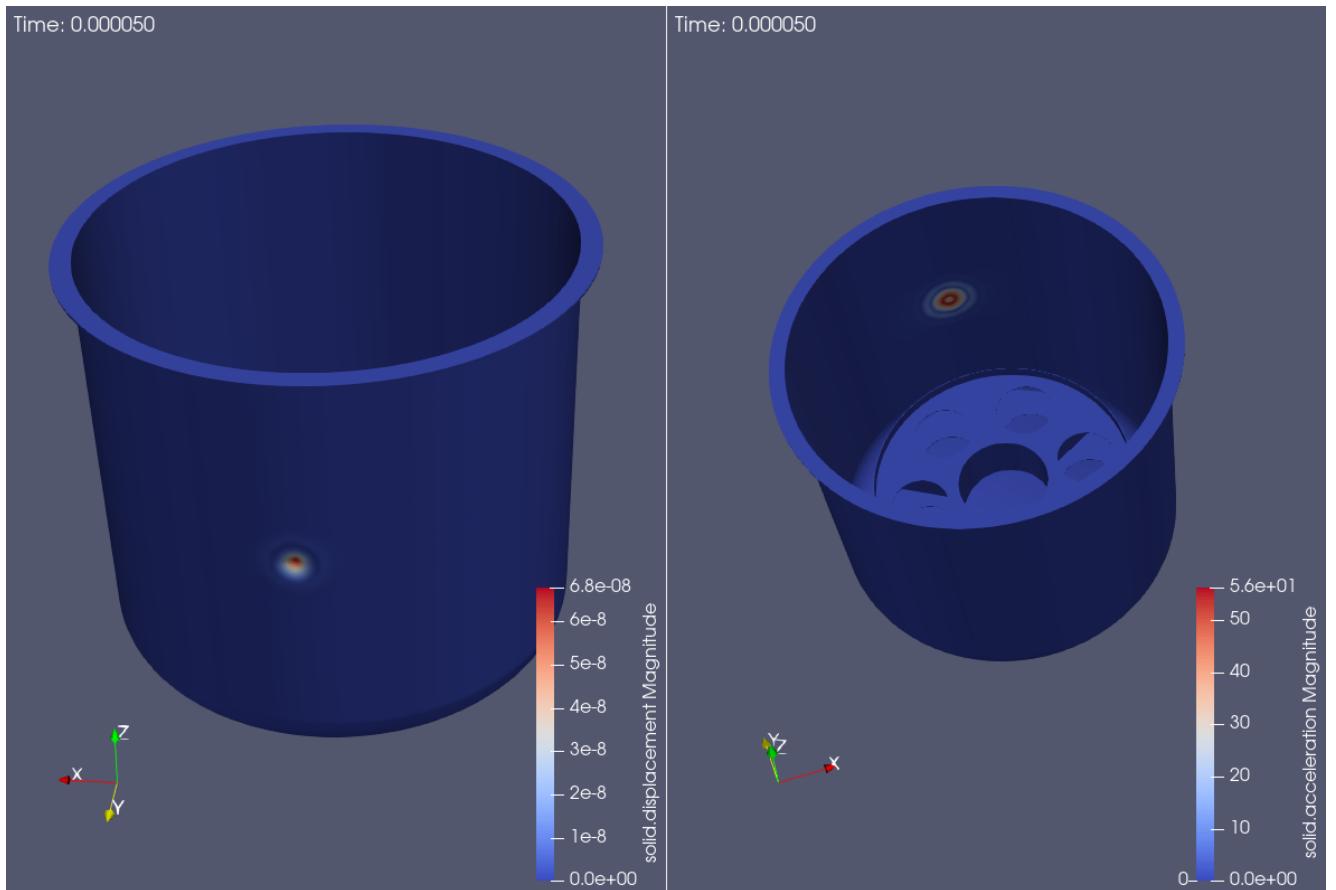


FIGURE 6.22 – Illustration de la simulation 3D avec la toolbox mécanique des solides

Paramètres utilisés : discréttisation P1G1, maillage m2 (6mm), pas de temps $\Delta t = 10^{-5}$, durée totale $T = 0.005$ s.

6.3.4 Résultats du code de couplage vibro-acoustique

Mesures extraite depuis la surface Gamma_N

Analyse des résultats. La figure 6.23 présente l'évolution temporelle de la quantité de déplacement intégrée sur la surface de Neumann Γ_N , notée $Q_{\Gamma_N}(t)$. On observe une réponse oscillatoire du solide suite à l'excitation appliquée sur cette surface.

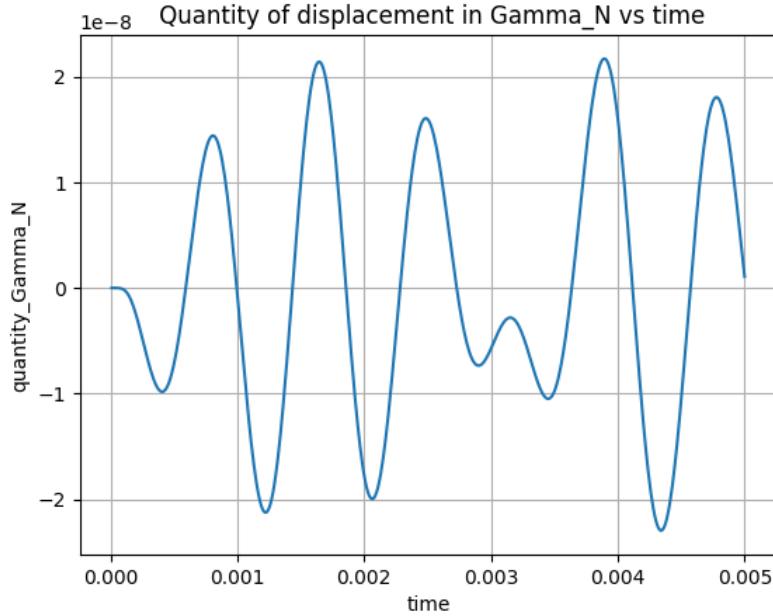


FIGURE 6.23 – Évolution temporelle de la quantité intégrée sur Γ_N , $Q_{\Gamma_N}(t)$ pour une simulation en 3D, P1G1

L'amplitude des oscillations reste de l'ordre de 10^{-8} , ce qui correspond à aux déplacements théoriques (Annexe 7.3.2). Pour une interprétation plus fine (accélération, fréquence propre dominante), il sera nécessaire d'extraire des mesures ponctuelles en des noeuds physiques du maillage, directement identifiés dans le fichier .geo.

Extraits des simulations en 2D et en 3D du couplage vibro-acoustique

Dans cette section, nous présentons quelques extraits vidéos issus des simulations numériques.

Simulation 2D. Une première vidéo illustre le couplage vibro-acoustique en deux dimensions.

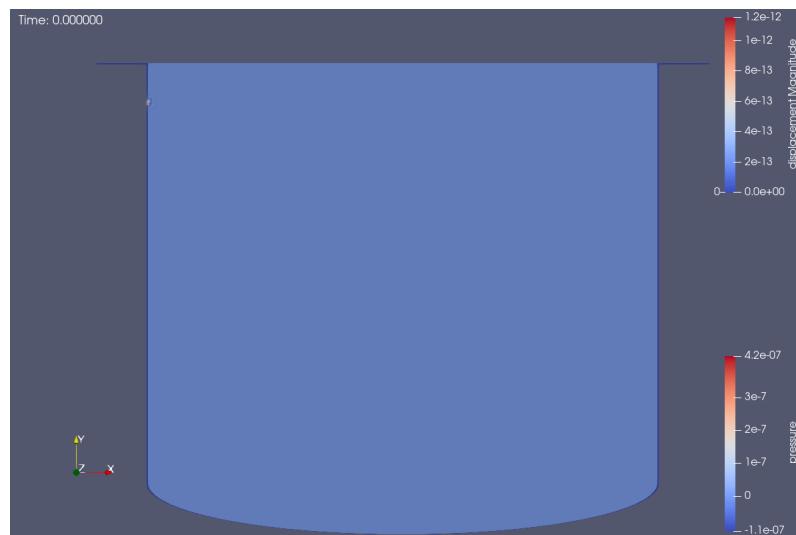


FIGURE 6.24 – Illustration de la simulation 2D de couplage vibro-acoustique

Paramètres utilisés : discréétisation P1G1, pas de temps $\Delta t = 5.10^{-7}$, durée totale $T = 0.002$ s.

Simulation 3D. Une deuxième vidéo montre la simulation en trois dimensions.

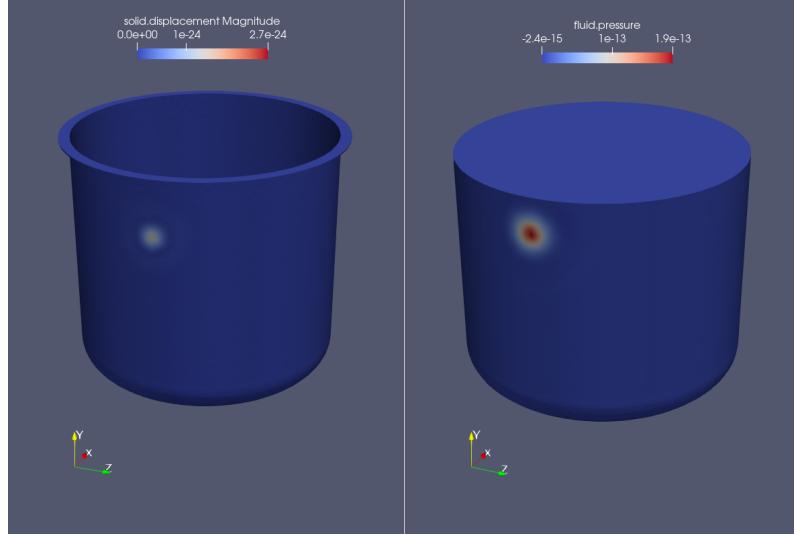


FIGURE 6.25 – Illustration de la simulation 3D de la résolution de l'équation d'élasticité

Paramètres utilisés : discréétisation P1G1, maillage $m2=6\text{mm}$, pas de temps $\Delta t = 10^{-5}$, durée totale $T = 0.001 \text{ s}$. Les vidéos affichent le déplacement du solide \mathbf{u}_s et la pression du fluide p_f projetés sur l'interface fluide–structure Γ_{fs} . On voit que le fluide reçoit le déplacement initial au bon endroit (zone d'impact) dès les premiers instants. L'onde de pression naît à cette même zone et se propage ensuite dans le fluide. Les variations de p_f concordent avec les déplacements normaux u_n de la paroi : pics lors des phases de compression, baisse lors du recul. Cette cohérence valide qualitativement la transmission des efforts à l'interface et le bon fonctionnement du couplage.

6.4 Benchmarking

L'évaluation des performances a été réalisée à l'aide des outils de **benchmarking** fournis par la bibliothèque **Feel++** [14]. Trois stratégies principales sont disponibles pour analyser les résultats :

- **Performance** : cette métrique fournit le temps de calcul brut, sans normalisation particulière. Elle constitue la base de l'analyse.
- **Relative performance** : cette stratégie exprime la part relative de chaque phase de calcul (par exemple, communication versus calcul) par rapport au temps total. Les proportions obtenues s'additionnent toujours à 100 %.
- **Speedup** : cette métrique calcule l'accélération obtenue en fonction du nombre de tâches. Mathématiquement, le speedup $S(p)$ pour p processus est défini par :

$$S(p) = \frac{T(1)}{T(p)}$$

où $T(1)$ est le temps d'exécution sur un seul processus et $T(p)$ est le temps d'exécution sur p processus. Le *speedup optimal* (linéaire) correspond à $S_{\text{opt}}(p) = p$, représentant une accélération parfaite. Le *half-optimal* est défini comme $S_{\text{half}}(p) = \frac{p}{2}$, servant de référence minimale acceptable pour une scalabilité raisonnable.

Scalabilité forte La scalabilité forte mesure la capacité d'un algorithme parallèle à maintenir son efficacité lorsque la taille du problème est fixe et que le nombre de processus augmente. L'efficacité forte $E_{\text{forte}}(p)$ est donnée par :

$$E_{\text{forte}}(p) = \frac{S(p)}{p} = \frac{T(1)}{p \cdot T(p)}$$

Une efficacité proche de 1 indique une scalabilité forte idéale, où le temps de calcul diminue proportionnellement à l'augmentation des ressources. Une efficacité qui décroît rapidement avec p révèle des limitations dans la parallélisation.

Ces métriques permettent de quantifier à la fois la *scalabilité forte* et l'efficacité relative des différentes phases du calcul (initialisation, assemblage, résolution, post-traitement). L'analyse présentée ci-dessous s'appuie sur ces définitions pour comparer différentes configurations de maillages, d'ordres de discréétisation et de solveurs.

6.4.1 Benchmarking des simulations avec la Toolbox mécanique des solides

Effets de la parallélisation

Les figures 6.27 à 6.29 présentent l'évolution du *speedup* pour trois niveaux de maillage ($m1, m2, m3$), tandis que la figure 6.26 détaille la répartition des temps d'exécution entre les principales phases de calcul. Le post-traitement ici est limité à l'extraction des mesures à un point et est très faible comparé aux autres phases (de l'ordre de la micro-seconde).

Il est aussi pertinent de noter que l'augmentation de l'ordre géométrique cause aussi une augmentation de la complexité du système et ainsi du temps de calcul.

Analyse des temps d'exécution : La figure 6.26 montre que le temps de résolution domine largement le coût total de la simulation, surtout pour les maillages fins ($m1$). À mesure que le nombre de processus augmente, la part relative de la phase d'initialisation (`Constructor_init`) reste faible : le problème est essentiellement gouverné par le solveur, qui se parallélise mieux que les autres étapes.

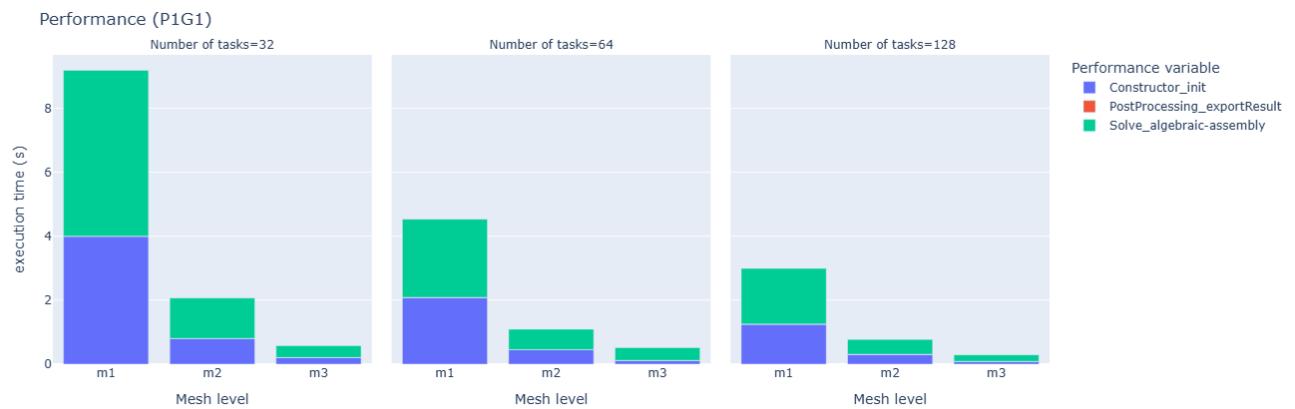


FIGURE 6.26 – Décomposition des temps d'exécution pour $m1, m2$ et $m3$ (toolbox mécanique des solides).

Analyse du speedup : Pour les maillages $m1$ et $m2$, le *speedup* croît régulièrement avec le nombre de tâches et se rapproche de la borne « half-optimal ». L'étape de résolution (`Solve_algebraic-assembly`) bénéficie le plus du parallélisme, avec une accélération significative jusqu'à 128 processus. En revanche, pour le maillage le plus grossier ($m3$), les gains restent limités : le coût de communication prend le dessus et la scalabilité sature rapidement.

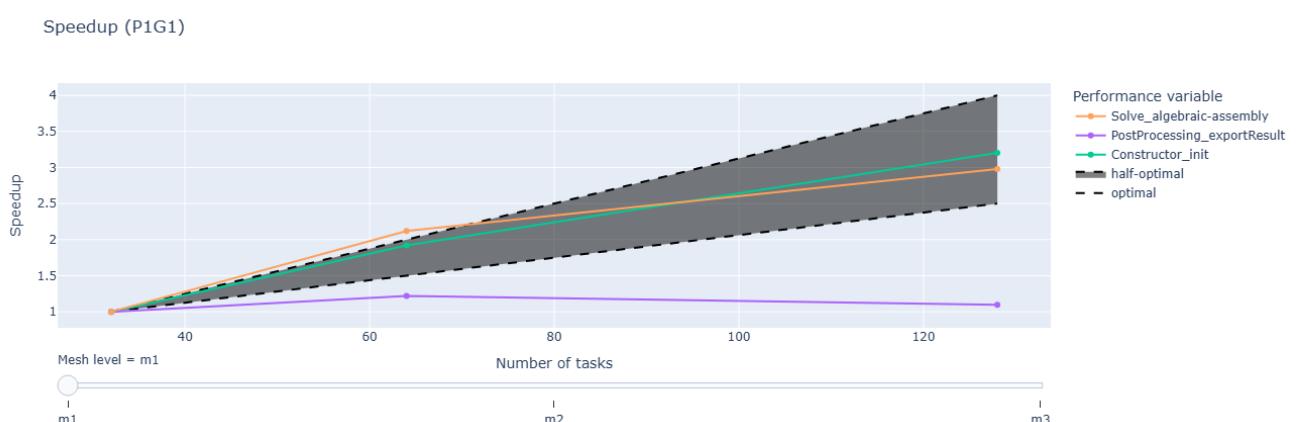


FIGURE 6.27 – Speedup obtenu pour $m1$ (toolbox mécanique des solides).

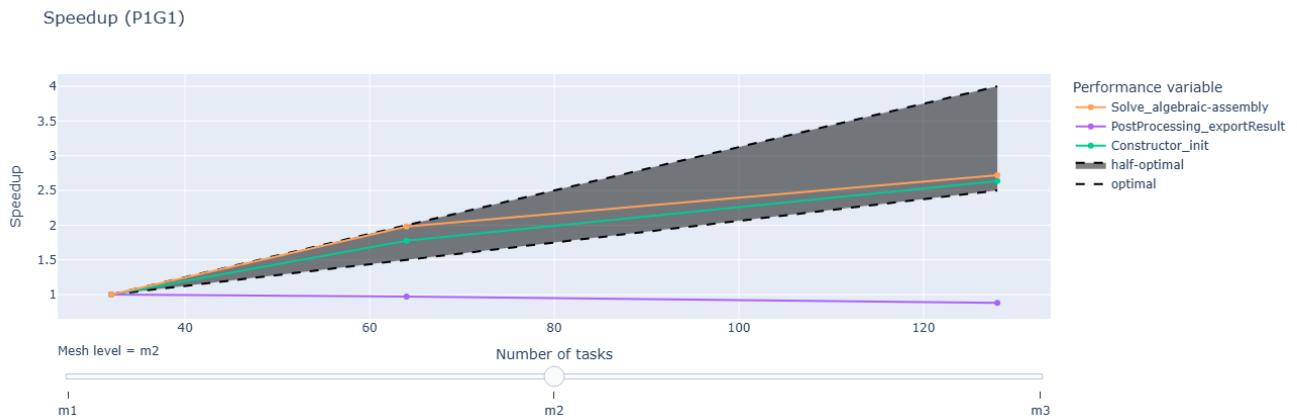


FIGURE 6.28 – Speedup obtenu pour $m2$ (toolbox mécanique des solides).

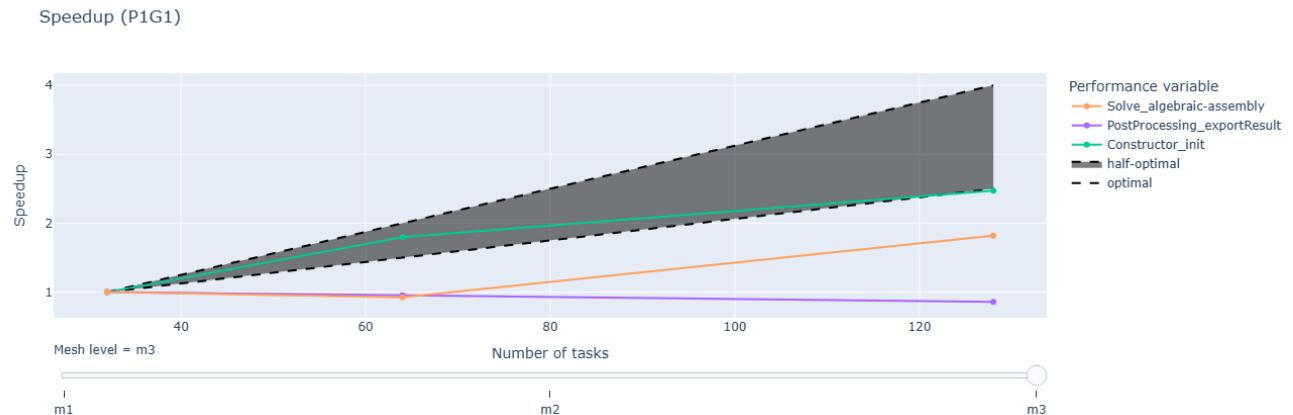


FIGURE 6.29 – Speedup obtenu pour $m3$ (toolbox mécanique des solides).

Bilan. Ces résultats démontrent que la formulation du problème à l'aide de la toolbox mécanique des solides exploite correctement le parallélisme MPI sur des maillages suffisamment fins. Le gain est notable pour $m1$ et $m2$, tandis que l'intérêt d'utiliser un grand nombre de coeurs sur $m3$ reste limité.

Effets de l'augmentation des ordres de discréétisation

La figure 6.30 présente les temps d'exécution mesurés pour différentes configurations de discréétisation (P1 à P5, ordres géométriques et tailles de maillage associées), en utilisant 128 coeurs de calcul. On observe une croissance marquée du temps de calcul lorsque l'ordre de discréétisation augmente : la résolution pour P4 et P5 entraîne des coûts supérieurs de plusieurs ordres de grandeur par rapport aux cas P1 ou P2.

Performance (All cases @128 tasks)

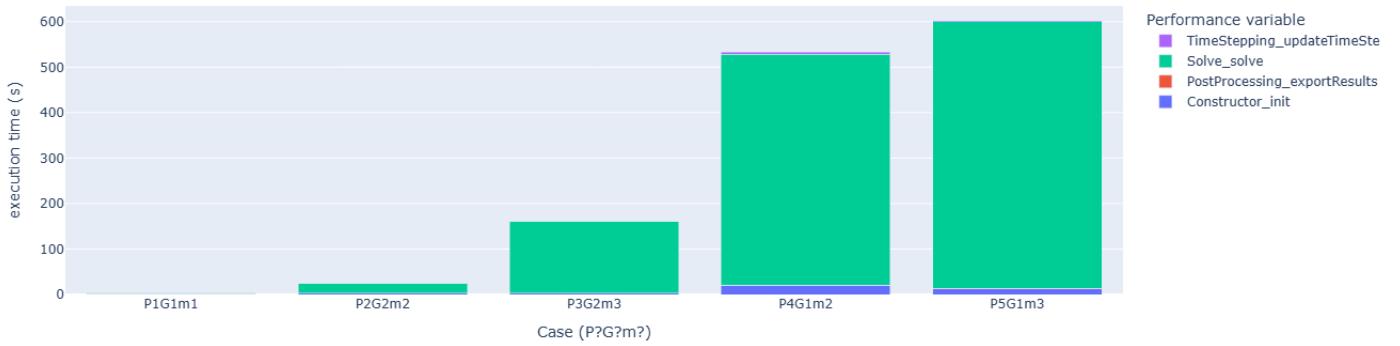


FIGURE 6.30 – Performance des simulations selon l’ordre de discréétisation (128 tâches).

La figure 6.30 montre que le *Solve* domine largement le coût. Le tableau 6.3 croise le nombre de degrés de liberté (DOF) et le temps *Solve*. Le coût n’est pas proportionnel aux DOF : **P4G1m2** (36.46 M DOF) se résout en 508 s alors que **P5G1m3** (17.87 M DOF) prend 587 s. Le *coût par million de DOF* croît fortement avec l’ordre polynomial, signe d’un conditionnement plus difficile à haut P .

TABLE 6.3 – DOF et temps *Solve* à 128 tâches, avec coût normalisé (s par million de DOF).

Cas	DOF	Temps Solve [s]	s / MDOF
P1G1m1	3 567 740	2.27	0.636
P2G2m2	5 453 808	21.21	3.889
P3G2m3	4 390 983	157	35.755
P4G1m2	36 459 141	508	13.933
P5G1m3	17 869 257	587	32.850

Constats

- Le coût par DOF explose au-delà de $P2$ ($P3$: 35.8 s/MDOF ; $P5$: 32.9 s/MDOF).
- $P4$ a beaucoup plus de DOF que $P5$, mais un coût total plus faible : l’ordre P pénalise plus que le seul volume de DOF.
- À DOF comparable, $P-/G-$ raffinement décrit mieux les fronts d’ondes que le h -raffinement, mais le *Solve* implicite devient un goulet pour $P \geq 3$.

Les simulations ont été achevées avec différentes durée au long du stage. Voici l’occupation mémoire de ces simulations en normalisant par le nombre de pas de temps de chacune :

TABLE 6.4 – Taille totale (GiB) et taille par pas (MiB) — valeurs sources en KiB.

Cas	Taille brute [KiB]	Pas de temps	Total [GiB]	Par pas [MiB]
P1G1m1	136 620 468	41 160	130.29	3.24
P2G2m2	4 742 132	940	4.52	4.93
P3G2m3	1 709 072	407	1.63	4.10
P4G1m3	42 210 612	190	40.26	216.95
P5G1m3	65 339 156	150	62.31	425.39

6.4.2 Benchmarking des simulations de couplage vibro-acoustique - élasticité

Ordres de grandeur (onde et couplage, 3D, P1, 128 tâches). Par manque de temps, le benchmarking détaillé a porté sur l’élasticité dans le solide. Pour référence, les temps ci-dessous donnent une idée des coûts pour l’équation des ondes et pour le couplage avec la même configuration.

TABLE 6.5 – Temps rapportés (secondes). Couplage : temps *par pas* selon les logs.

Cas	Solve Wave [s]	Coupling [s/pas]
m1	0.482	17.088
m2	0.147	5.316
m3	0.065	3.273

Analyse comparative des discrétisations P1G1, P2G1 et P3G1

Les figures 6.31, 6.32 et 6.33 présentent la décomposition des temps d'exécution pour les trois niveaux de maillage (m_1 , m_2 , m_3), où m_1 est le plus fin (3 mm), m_2 (6mm) et m_3 (12mm) le plus grossier. Les figures de *speedup* pour tous les maillages et discrétisations (Fig. 6.34–6.36) révèlent des tendances de performance parallèle similaires. Dans ces simulations on exporte des fichiers de simulations au post-traitement.

Analyse des temps d'exécution totaux

Pour les trois schémas (P1G1, P2G1, P3G1), la répartition des coûts computationnels suit un profil similaire :

- **Constructor** : La différence de temps d'initialisation reste marginal pour tous les schémas mais augmente avec le nombre de processus.
- **Solve** : Phase la plus coûteuse en calcul séquentiel, bénéficiant modérément de la parallélisation.
- **PostProcessing** : Sa part relative devient prépondérante à mesure que le nombre de tâches augmente. Cette tendance est observée sur tous les maillages et pour tous les ordres polynomiaux.

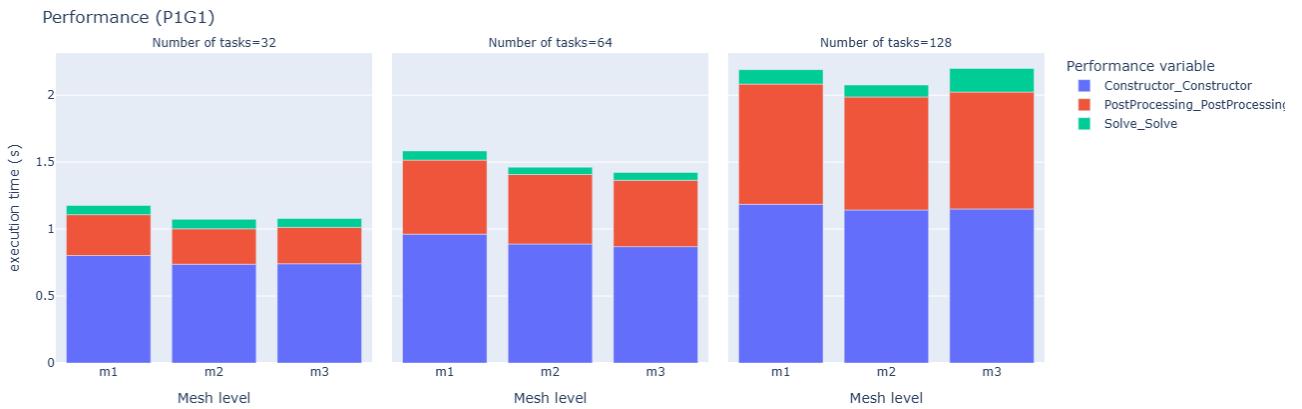


FIGURE 6.31 – Décomposition des temps d'exécution pour les maillages m_1 (3mm), m_2 (6mm) et m_3 (12mm) - Discrétisation P1G1

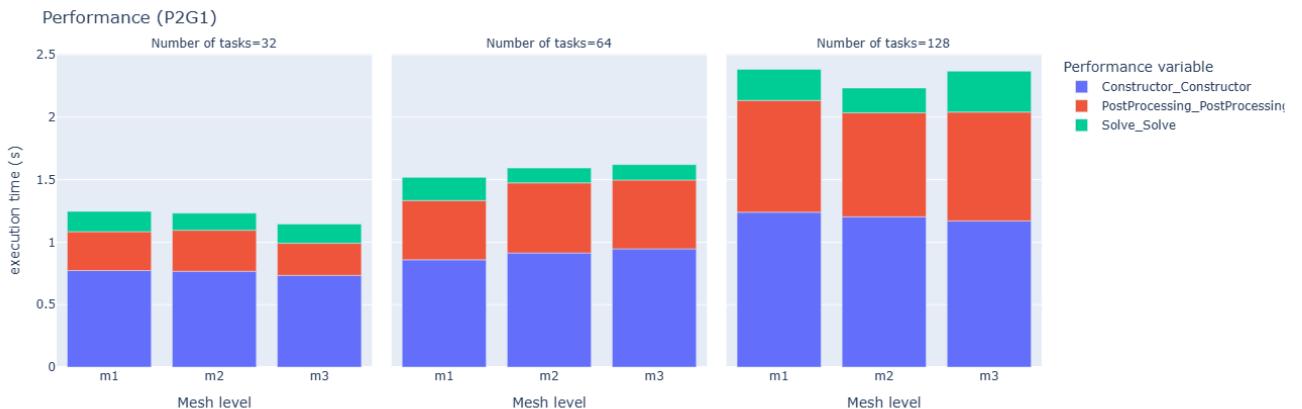


FIGURE 6.32 – Décomposition des temps d'exécution pour les maillages m_1 (3mm), m_2 (6mm) et m_3 (12mm) - Discrétisation P2G1

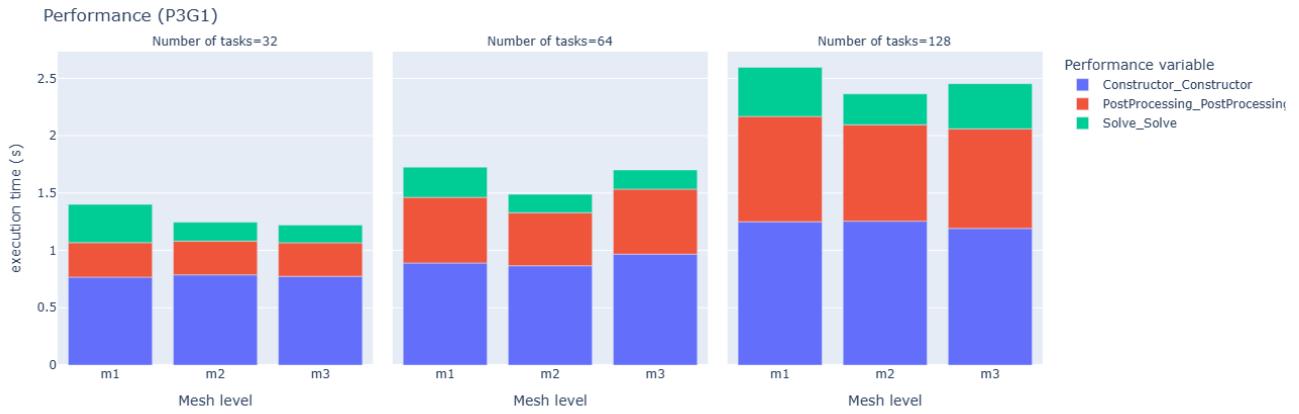


FIGURE 6.33 – Décomposition des temps d'exécution pour les maillages m1 (3mm), m2 (6mm) et m3 (12mm) - Discrétisation P3G1

Analyse du speedup : Le *speedup* obtenu reste très éloigné du comportement idéal pour les trois discrétisations et tous les niveaux de maillage. Au-delà de 32-64 tâches, les courbes s'aplatissent systématiquement, indiquant une saturation. L'analyse par phase montre que :

- La phase **Solve** est la seule à maintenir une accélération modeste mais constante.
- La phase **PostProcessing** a la scalabilité la plus faible.

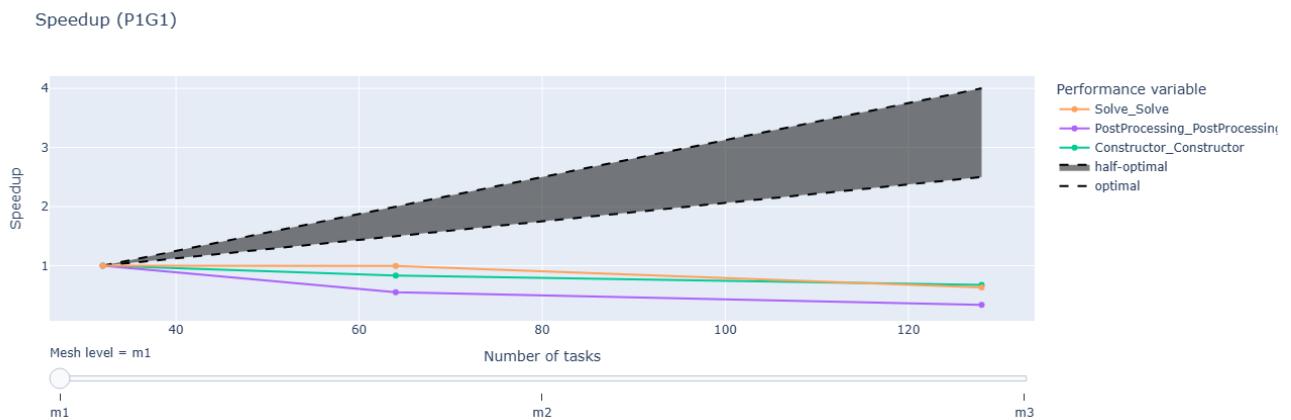


FIGURE 6.34 – Speedup et efficacité parallèle - Discrétisation P1G1 avec maillage fin m1 (3mm)

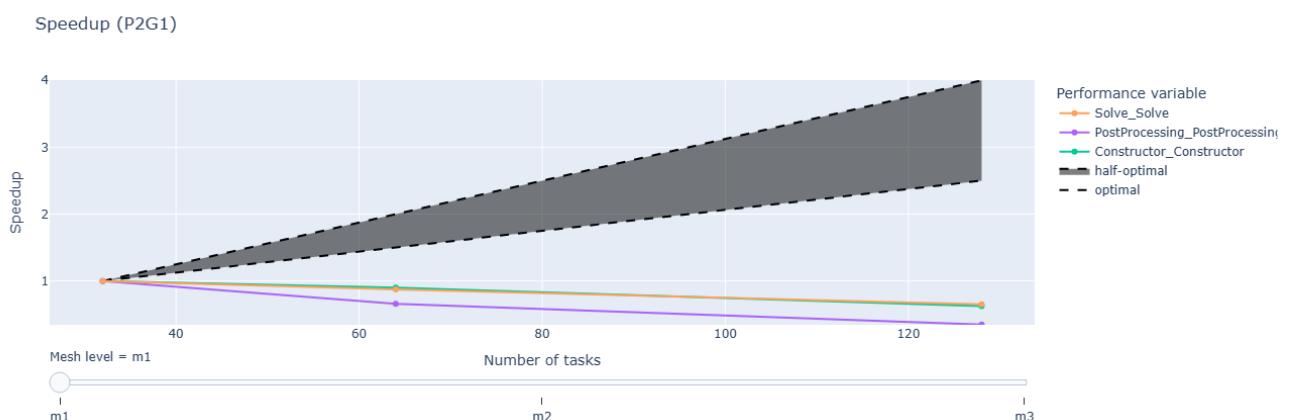


FIGURE 6.35 – Speedup et efficacité parallèle - Discrétisation P2G1 avec maillage fin m1 (3mm)

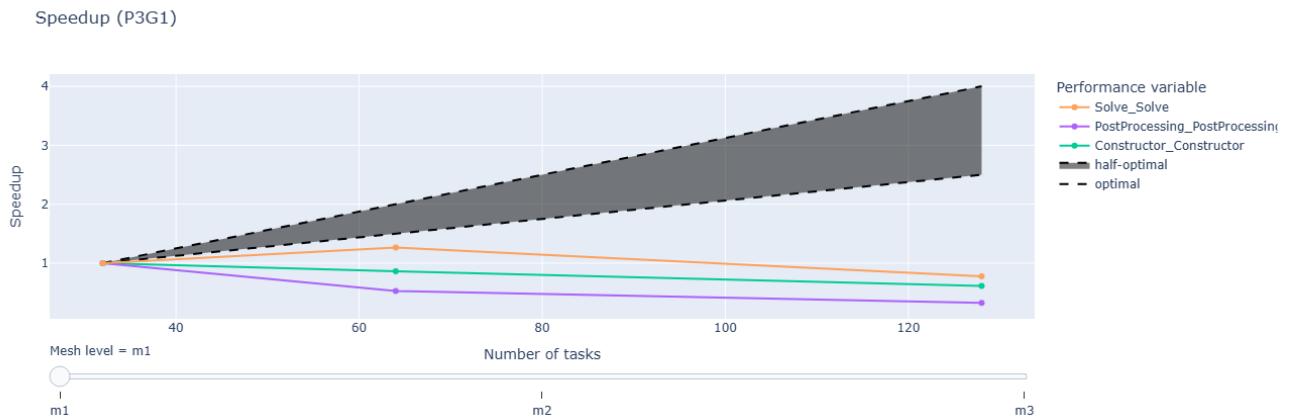


FIGURE 6.36 – Speedup et efficacité parallèle - Discréétisation P3G1 avec maillage fin m1 (3mm)

Bilan global

L'efficacité parallèle est fortement limitée, et l'utilisation d'un nombre élevé de processus MPI ne se traduit pas par une accélération. La qualité du maillage ou l'ordre plus élevé ne suffisent pas à améliorer la scalabilité, le principal frein restant le coût du post-traitement. Une optimisation du code pourrait être indispensable pour tirer profit d'une exécution sur un grand nombre de coeurs. Il se peut aussi que le système 2D ne soit pas bien parallélisable dû au surcoût de communication.

Simulations 3D (P1G1 et P2G1)

Les figures 6.39 à 6.41 présentent les résultats de *speedup* obtenus pour P1G1 en 3D, tandis que la figure 6.38 décompose les temps d'exécution. De même, les figures 6.42 à 6.44 illustrent les performances de P2G1, complétées par la figure 6.37. Pour réduire le temps d'exécution on se contente d'extraire les mesures sur la surface d'excitation au post-traitement. Pour cette même raison, on définit aussi une taille de maillage intermédiaire de 9mm ainsi on a hsize_m2 = 6mm hsize_m3 = 9mm et hsize_m4 = 12mm.

Analyse des temps d'exécution : Les figures de décomposition confirment que le coût global augmente avec la taille du maillage, mais la part relative de la résolution devient prédominante en 3D. Pour P1, le post-traitement reste visible mais secondaire par rapport au solveur. Pour P2, la résolution domine nettement le temps total, ce qui explique la meilleure efficacité parallèle observée. Les phases de construction restent modestes mais varient légèrement avec le nombre de tâches.

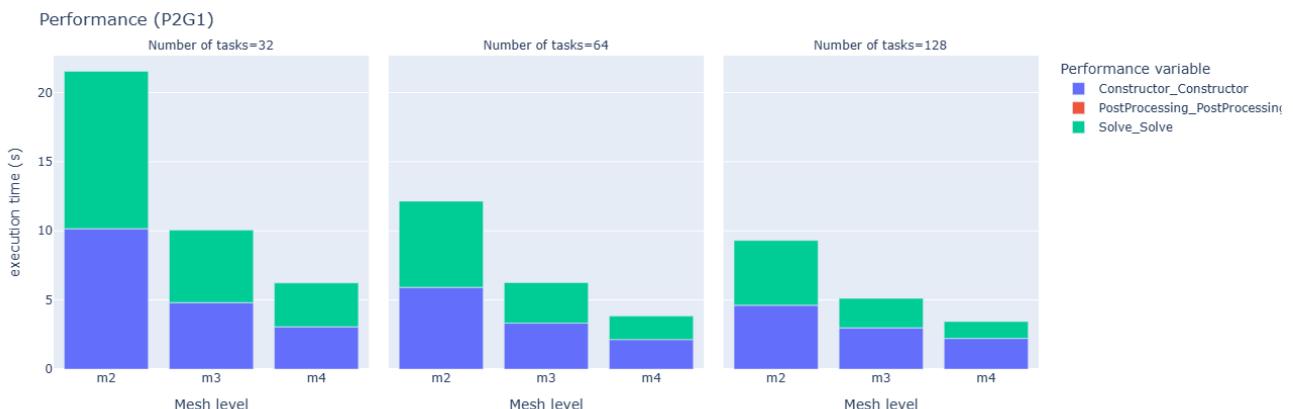


FIGURE 6.37 – Décomposition des temps d'exécution (P2G1, 3D).

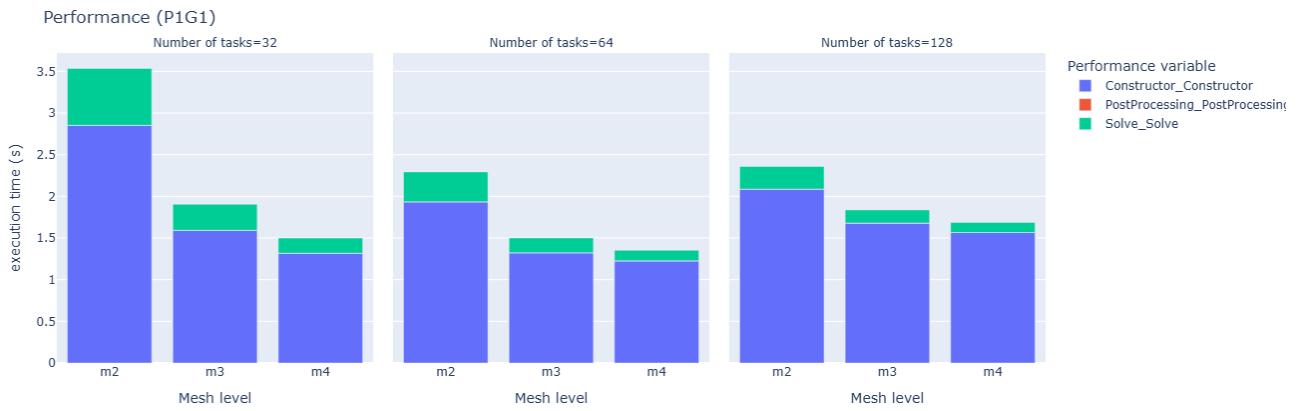


FIGURE 6.38 – Décomposition des temps d'exécution (P1G1, 3D).

Analyse du speedup (P1G1 3D) : Pour le degré P1 en 3D, on observe une meilleure tendance que dans les simulations 2D : le *speedup* progresse avec le nombre de tâches et se rapproche de la zone « half-optimal ». L'étape de résolution (*Solve*) est celle qui bénéficie le plus du parallélisme. Toutefois, l'écart reste notable avec la droite optimale, en particulier au-delà de 64 tâches, où les gains se stabilisent.

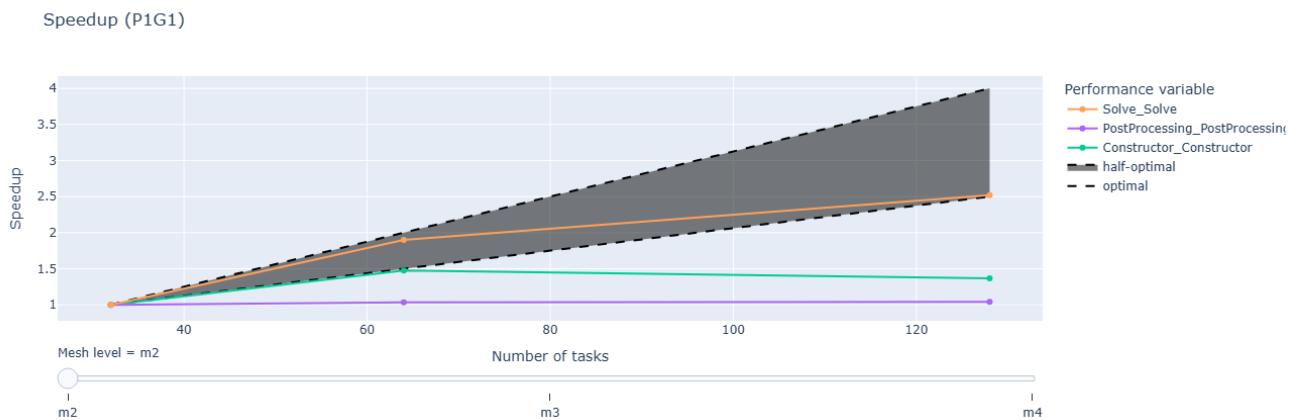


FIGURE 6.39 – Speedup obtenu pour m2 (P1G1, 3D).

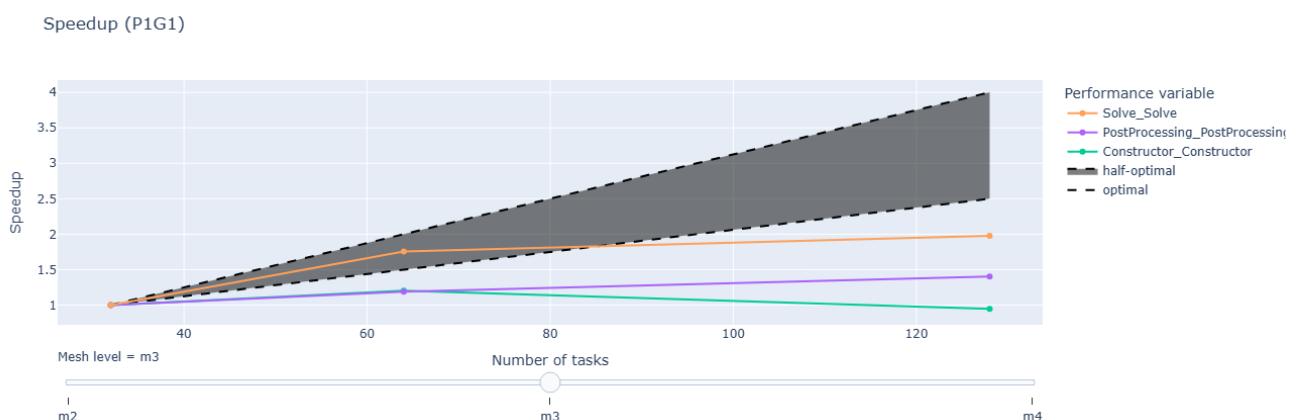


FIGURE 6.40 – Speedup obtenu pour m3 (P1G1, 3D).

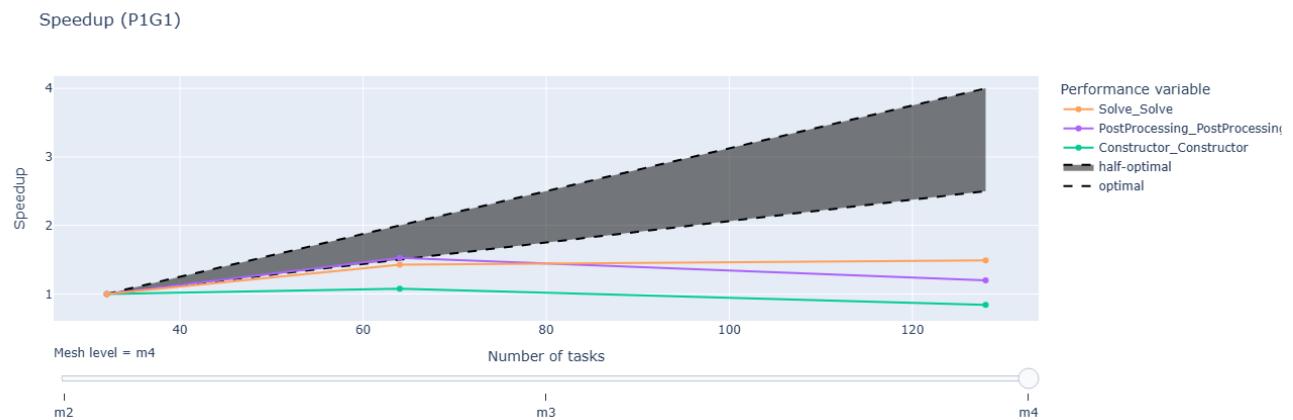


FIGURE 6.41 – Speedup obtenu pour $m4$ (P1G1, 3D).

Analyse du speedup (P2G1 3D) : Le passage à un ordre supérieur (P2G1) améliore encore la scalabilité. Les courbes de **Solve** et **Constructor** se rapprochent de la borne idéale, montrant que l'augmentation de la charge de calcul par élément profite au parallélisme.

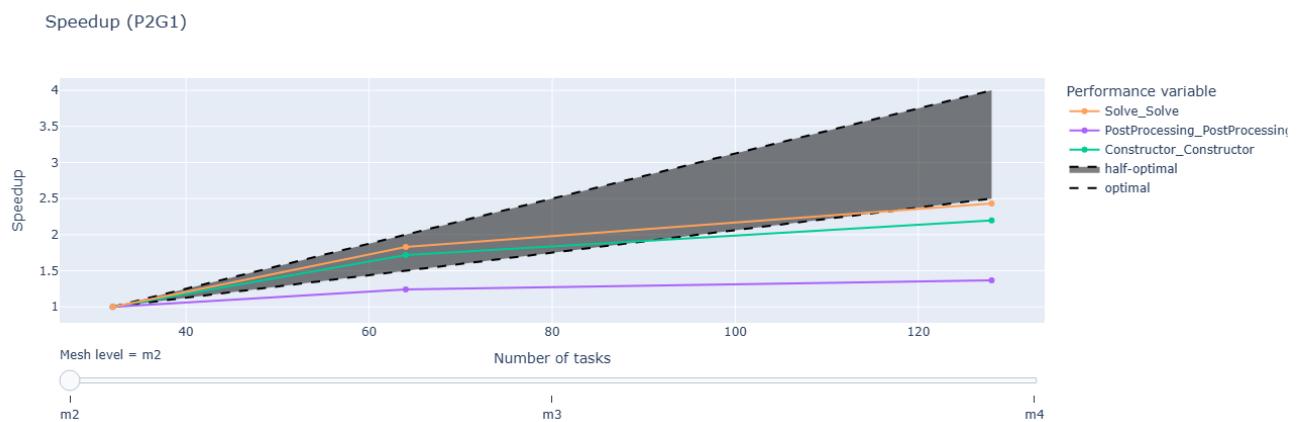


FIGURE 6.42 – Speedup obtenu pour $m2$ (P2G1, 3D).

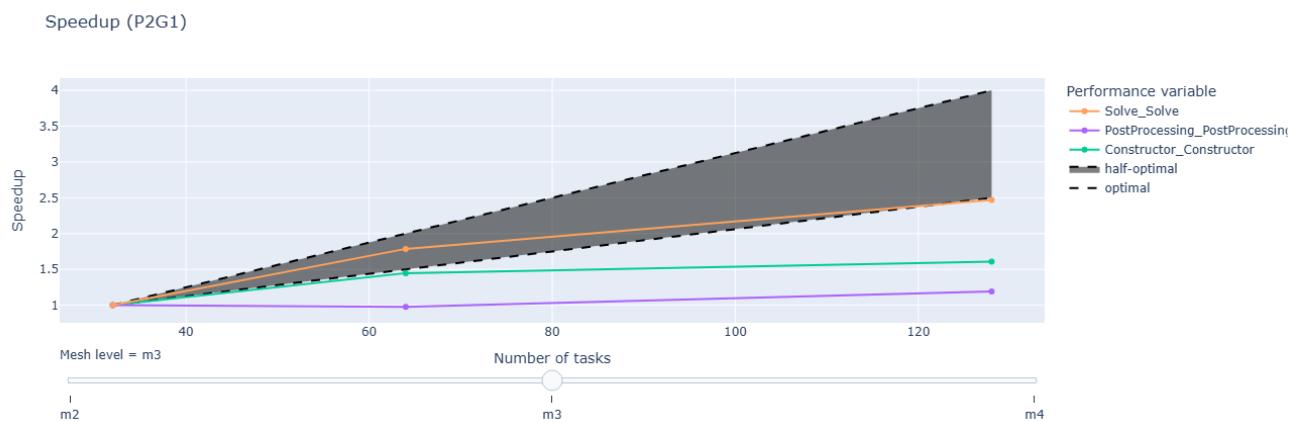


FIGURE 6.43 – Speedup obtenu pour $m3$ (P2G1, 3D).

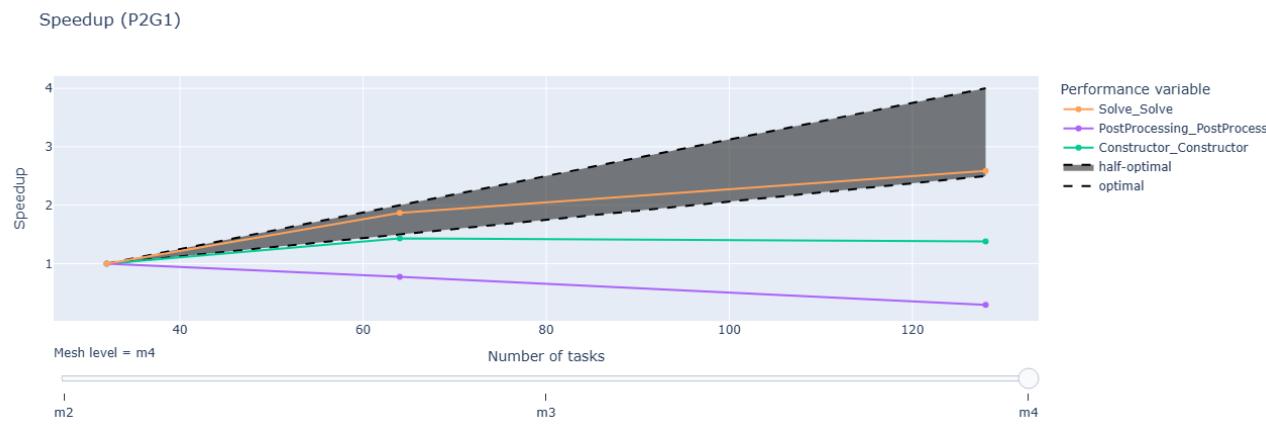


FIGURE 6.44 – Speedup obtenu pour $m4$ (P2G1, 3D).

Bilan : Ces résultats montrent qu'en 3D, la scalabilité est plus favorable qu'en 2D, surtout pour les ordres d'approximation supérieurs. La complexité accrue du problème (maillage plus dense, matrices plus grandes) permet d'amortir les surcoûts de communication MPI. Ainsi, P2G1 en 3D représente le meilleur compromis, avec un parallélisme mieux exploité.

Chapitre 7

Conclusions et perspectives

7.1 Perspectives

L'implémentation actuelle reste implicite pour l'élasticité, car le schéma explicite devient instable en 3D malgré de petits pas Δt . De plus, la matrice de masse n'étant pas diagonale, une explicitation efficace du pas de temps est hors de portée. Côté fluide, la carte ALE n'est pas encore intégrée. La géométrie et les conditions aux limites ne reflètent pas fidèlement l'expérience (cuve percée à la colllerette et tenue par trois poutres). Enfin, les données expérimentales restent peu exploitées : les signaux de coda, avec eau et points d'excitation multiples, pourront servir à la validation et à la calibration.

Travail à mener.

1. **Passage à une méthode spectrale explicite (SEM).**
 - Quadratures de Gauss–Lobatto–Legendre (collocation) et éléments d'ordre polynomiaux et géométriques $P, G > 5$.
 - *Masse diagonale* : construction avec GLL pour obtenir M diagonale ; application de M^{-1} par produit point-à-point, hors boucle temporelle.
 - *Amortissement* : activer un amortissement (Rayleigh α, β extraits des mesures expérimentales) durant la résolution.
2. **Compléter le couplage.**
 - Implémenter la transformation ALE côté fluide.
 - Implémenter le benchmarking et les fichiers de scalabilité pour l'équation des ondes et le couplage.
3. **Réaliser la géométrie/CL réelles.**
 - Modéliser la colllerette percée et les trois poutres de support ; mettre à jour les CL (encastrements/liasons) et masses ajoutées des capteurs et transducteurs.
 - injecter directement l'impact du marteau mesuré expérimentalement.
4. **Retournement temporel numérique.**
 - Une fois la SEM explicite stabilisée : enregistrer, rétropropager, focaliser ; évaluer la qualité de focalisation et la sensibilité au bruit.
 - Cette implémentation pourra être comparée aux mesures expérimentales avec ou sans fluide de SONO-RHC.

7.2 Conclusions

Les résultats montrent que la propagation d'ondes est mieux capturée en augmentant l'ordre polynomial et l'ordre géométrique qu'en diminuant la taille des éléments du maillage. Ce constat oriente vers une implémentation complète de la méthode des éléments spectraux (SEM) pour ce cas vibro-acoustique.

Le stage s'est déroulé en deux volets. D'abord, un benchmarking de l'implémentation avec la toolbox *Solid*, avec des scripts Python pour le post-traitement et la gestion des différentes configurations (jusqu'à une centaine de simulations aux configurations variées). Ensuite, une implémentation C++ orientée objet d'un schéma explicite, menée en binôme en collaboration avec une autre stagiaire.

Ce cadre a imposé un rythme régulier (réunions hebdomadaires) et un travail coordonné, avec un bon équilibre entre autonomie et échanges.

Il y a eu des difficultés et des points à améliorer : gestion du temps (estimations trop optimistes, dispersion en multitâche, pression des échéances) ; recherche et documentation à mieux structurer ; communication à initier plus tôt et à mieux cadrer ; priorité à la qualité et à la reproductibilité plutôt qu'au volume.

7.3 Annexe

7.3.1 Dérivation de l'équation d'onde acoustique dans le fluide

Pour expliquer comment les équations gouvernantes de l'acoustique dans un fluide sont obtenues, on part des lois fondamentales de la dynamique des fluides. L'équation d'onde acoustique décrit comment de petites perturbations de pression, de vitesse et de densité se propagent sous forme d'ondes sonores dans un milieu compressible.

Hypothèses sous-jacentes à l'équation d'onde acoustique

La dérivation de l'équation d'onde acoustique linéaire repose sur les hypothèses suivantes :

Petites perturbations : Les champs de pression, densité et vitesse sont linéarisés autour d'un état de référence au repos :

$$\rho = \rho_0 + \delta\rho, \quad p = p_0 + \delta p, \quad \mathbf{v} = \delta\mathbf{v}$$

Les termes non linéaires (par exemple, les produits des perturbations) sont négligés.

Fluide non visqueux : Les effets visqueux sont ignorés. Le mouvement du fluide est gouverné par les équations d'Euler sans viscosité, et non par les équations complètes de Navier-Stokes.

Processus isentropique : Le fluide est supposé adiabatique et réversible, de sorte que la pression ne dépend que de la densité :

$$\delta p = c^2 \delta\rho$$

où c est la vitesse du son.

Pas d'écoulement moyen : La vitesse de fond est nulle ($\mathbf{v}_0 = 0$), ce qui simplifie les équations et élimine les termes convectifs.

Écoulement irrotationnel : Les perturbations acoustiques sont supposées irrotationnelles, et toute vorticité générée par des effets non linéaires est négligée.

État de référence uniforme : La pression p_0 et la densité ρ_0 de fond sont constantes dans l'espace et dans le temps.

Remarque : Effet d'une température non uniforme La dérivation ci-dessus suppose un fluide isentropique avec une vitesse du son constante c . Si la température du fluide n'est pas uniforme, cette hypothèse n'est plus valide car la vitesse du son dépend de la température :

$$c(\mathbf{x}) = \sqrt{\gamma \frac{RT(\mathbf{x})}{M}}$$

où :

- γ : rapport des chaleurs spécifiques,
- R : constante universelle des gaz parfaits,
- M : masse molaire du fluide,
- $T(\mathbf{x})$: température variant dans l'espace.

Dans ce cas, l'équation d'onde devient :

$$\frac{\partial^2 p}{\partial t^2} - c(\mathbf{x})^2 \Delta p = 0$$

Équations gouvernantes (fluide compressible non visqueux)

Soient ρ la densité, \mathbf{v} la vitesse particulière, et p la pression. Les équations sont :

— **Conservation de la masse (continuité) :**

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

— **Conservation de la quantité de mouvement (équation d'Euler) :**

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) = -\nabla p$$

— **Équation d'état (fluide isentropique) :**

$$p = p(\rho) \Rightarrow \delta p = c^2 \delta\rho$$

Linéarisation autour de l'équilibre

On suppose de petites perturbations autour d'un état statique :

$$\rho = \rho_0 + \delta\rho, \quad p = p_0 + \delta p, \quad \mathbf{v} = \delta\mathbf{v}$$

avec ρ_0, p_0 constants.

En négligeant les termes non linéaires, on obtient le système linéarisé :

— **Équation de continuité linéarisée :**

$$\frac{\partial \delta\rho}{\partial t} + \rho_0 \nabla \cdot \delta\mathbf{v} = 0$$

— **Équation de quantité de mouvement linéarisée :**

$$\rho_0 \frac{\partial \delta\mathbf{v}}{\partial t} = -\nabla \delta p$$

— **Équation d'état linéarisée :**

$$\delta p = c^2 \delta\rho$$

Dérivation de l'équation d'onde

1. On dérive en temps l'équation de continuité linéarisée :

$$\frac{\partial^2 \delta\rho}{\partial t^2} + \rho_0 \nabla \cdot \frac{\partial \delta\mathbf{v}}{\partial t} = 0$$

2. On insère l'équation de quantité de mouvement linéarisée :

$$\frac{\partial \delta\mathbf{v}}{\partial t} = -\frac{1}{\rho_0} \nabla \delta p \quad \Rightarrow \quad \frac{\partial^2 \delta\rho}{\partial t^2} - \nabla \cdot \nabla \delta p = 0$$

3. On utilise l'équation d'état :

$$\delta\rho = \frac{1}{c^2} \delta p \quad \Rightarrow \quad \frac{1}{c^2} \frac{\partial^2 \delta p}{\partial t^2} - \Delta \delta p = 0$$

On obtient ainsi l'équation d'onde acoustique classique :

$$\boxed{\frac{\partial^2 p}{\partial t^2} - c^2 \Delta p = 0 \quad \text{dans } \Omega_f}$$

Dérivation de la forme ALE de l'équation d'onde acoustique

L'équation classique en formulation eulérienne est :

$$\frac{\partial^2 p}{\partial t^2} - c^2 \Delta p = 0 \quad \text{dans un domaine fixe}$$

Dans le cadre ALE, le domaine fluide $\Omega_f(t)$ est mobile. Soit :

- $\hat{\mathbf{x}} \in \hat{\Omega}_f$ la coordonnée de référence,
- $\mathbf{x}(\hat{\mathbf{x}}, t) = \phi(\hat{\mathbf{x}}, t) = \hat{\mathbf{x}} + \mathbf{d}_f(\hat{\mathbf{x}}, t)$ l'application ALE,
- \mathbf{d}_f le déplacement du maillage,
- $\mathbf{w} = \frac{\partial \mathbf{d}_f}{\partial t}$ la vitesse du maillage.

La dérivée temporelle ALE d'un scalaire $p(\mathbf{x}, t)$ est :

$$\left. \frac{dp}{dt} \right|_{ALE} = \frac{\partial p}{\partial t} - \mathbf{w} \cdot \nabla p$$

En dérivant une seconde fois et en négligeant les termes d'ordre supérieur :

$$\frac{d^2 p}{dt^2} \approx \frac{\partial^2 p}{\partial t^2} - 2\mathbf{w} \cdot \nabla \left(\frac{\partial p}{\partial t} \right) - (\nabla \cdot \mathbf{w}) \frac{\partial p}{\partial t}$$

En substituant dans l'équation d'onde classique, on obtient :

$$\boxed{\frac{\partial^2 p}{\partial t^2} + 2\mathbf{w} \cdot \nabla \left(\frac{\partial p}{\partial t} \right) + (\nabla \cdot \mathbf{w}) \frac{\partial p}{\partial t} - c^2 \Delta p = 0 \quad \text{dans } \Omega_f(t)}$$

Cette forme tient compte du déplacement du maillage fluide et assure la cohérence avec le mouvement de la structure sur l'interface Γ_{fsi} .

7.3.2 Estimation de l'ordre de grandeur du déplacement et de l'accélération dans la cuve sous une excitation maximale de 1N

Si l'on considère la paroi de la cuve comme une plaque mince en acier, on peut utiliser l'approximation d'une plaque infinie soumise à une force harmonique.

La fréquence d'anneau (f_0) est une fréquence caractéristique des plaques, c'est une fréquence critique à laquelle une plaque mince commence à vibrer de manière significative sous une excitation extérieure. Elle correspond au seuil à partir duquel la propagation d'ondes de flexion devient importante.

- À la fréquence d'anneau, la réponse en déplacement est amplifiée car la structure résonne avec l'excitation.
- Elle conditionne l'impédance mécanique et donc la relation entre force appliquée et déplacement.

La fréquence d'anneau f_0 pour une plaque mince peut être estimée par :

$$f_0 = \frac{c}{2\pi R}$$

où : - $c = 5415 \text{ m/s}$ est la célérité des ondes longitudinales dans l'acier.

- E est le module de Young de l'acier ($E \approx 2.1 \times 10^{11} \text{ pa}$).

- ρ est la masse volumique de l'acier ($\rho \approx 7870 \text{ kg/m}^3$).

- R est le rayon de la cuve, donné par $R = \frac{0.944}{2} = 0.472 \text{ m}$.

- À basse fréquence, la plaque se comporte comme un solide rigide → réponse quasi-statique.

- À haute fréquence, la plaque présente des ondes de flexion qui se propagent, et la réponse devient dispersive.

- La fréquence d'anneau (f_0) marque la transition entre ces deux régimes.

Calcul de la fréquence d'anneau

$$f_0 = \frac{5.17 \times 10^3}{2\pi \times 0.472}$$

$$f_0 \approx \frac{5.415 \times 10^3}{2\pi \times 0.472} = \frac{5.415 \times 10^3}{2.964} \approx 1825 \text{ Hz}$$

Interprétation :

- Si l'excitation est à $f_c = 5000 \text{ Hz}$, alors $f_c > f_0$, ce qui signifie que : - L'impédance mécanique commence à diminuer fortement. - Le déplacement peut être amplifié dans certaines régions à cause des modes vibratoires. - Il peut y avoir de la dispersion d'ondes

Expression de l'impédance Z_{plate}

$$Z_{\text{plate}} = 8h^2 \sqrt{\frac{E\rho}{12(1-\nu^2)}}$$

ou approximé sous la forme :

$$Z_{\text{plate}} = 2.3c_L\rho h^2$$

où :

- $c_L = \sqrt{\frac{E}{\rho(1-\nu^2)}}$ est la vitesse des ondes longitudinales dans l'acier, - $\rho = 7870 \text{ kg/m}^3$ (masse volumique de l'acier), - $h = 3 \text{ mm} = 0.003 \text{ m}$ (épaisseur de la cuve), - $E = 2.1 \times 10^{11} \text{ pa}$ (module de Young de l'acier), - $\nu = 0.3$ (coefficients de poisson).

— Nous appliquons l'ondelette $-\sin(2\pi f_c t) \times \exp(-5 \times (f_c t - 2))$ qui représente un effort qui diminue rapidement après application, ce qui met la structure en vibration forcée puis libre, la vitesse qui résulte de cette force est représentée par la relation :

$$V = \frac{F}{Z}, Z = \frac{F}{V}$$

avec $V = j\omega u$, donc :

$$u = \frac{F}{Z\omega}$$

En utilisant l'expression de l'impédance :

$$u = \frac{F}{(2.3c_L\rho h^2)\omega}$$

Approximation numérique pour l'acier

On prend :

- $F = 1 \text{ N}$, - $c_L \approx 5415 \text{ m/s}$, - $\rho = 7870 \text{ kg/m}^3$, - $h = 3 \times 10^{-3} \text{ m}$, - $\omega = 2\pi f_c = 2\pi \times 5000$.

$$Z_{\text{plate}} \approx 2.3 \times 5415 \times 7870 \times (3 \times 10^{-3})^2$$

$$Z_{\text{plate}} \approx 2.3 \times 5415 \times 7870 \times 9 \times 10^{-6}$$

$$Z_{\text{plate}} \approx 882.15 \text{ N.s/m}$$

Maintenant, calculons u :

$$u = \frac{1}{(882.15)(2\pi \times 5000)}$$

$$u \approx \frac{1}{2.77 \times 10^6}$$

$$u \approx 3.61 \times 10^{-8} \text{ m} = 36.08 \text{ nm}$$

- Accélération :

$$a(t) = \frac{dv}{dt} = -\omega^2 u$$

Amplitude de l'accélération :

$$a_0 = \omega^2 u_0$$

Application à notre cas :

- $u_0 = 3.61 \times 10^{-8} \text{ m}$

- $\omega = 2\pi f_c = 2\pi \times 5000 = 31415.93 \text{ rad/s}$

Donc :

$$a_0 = \omega^2 u_0 = (31415.93)^2 \times 3.61 \times 10^{-8}$$

$$a_0 \approx 9.87 \times 10^8 \times 3.61 \times 10^{-8}$$

$$a_0 \approx 35.63 \text{ m/s}^2$$

7.3.3 Trouver un pas de temps idéal

Détermination d'un pas de temps idéal pour la simulation de propagation d'ondes sur une cuve nucléaire de 3 mm d'épaisseur avec une ondelette d'excitation de la forme :

$$F(t) = -\sin(2\pi f_c t) \cdot \exp(-5(f_c t - 2)^2)$$

tq

$$f_c = 5000 \text{ Hz.}$$

Le pas de temps doit être choisi en fonction :

1. De la fréquence maximale de l'onde pour respecter le critère de Nyquist.
2. De la vitesse de propagation dans l'acier.
3. Du critère de Courant-Friedrichs-Lowy (CFL) pour la stabilité de la simulation.

1. avec une fréquence de 5000 Hz pour éviter l'aliasing temporel, il faut respecter le théorème de Nyquist :

$$\Delta t \leq \frac{1}{2f_{\max}}$$

tq

f_{\max} est la fréquence maximale présente dans le signal.

Si on suppose que l'ondelette possède des harmoniques jusqu'à $3f_c$ (soit 15 kHz), alors :

$$\Delta t_{\text{Nyquist}} \leq \frac{1}{2 \times 15000} = 3.3 \times 10^{-5} \text{ s} \quad (33 \mu\text{s})$$

Cependant, ceci est une limite maximale, un pas plus petit peut être nécessaire.

2. Critère de stabilité CFL pour éviter des instabilités numériques, on utilise la condition CFL :

$$\frac{c\Delta t}{\Delta x} \leq C_{\text{CFL}}$$

tq :

- c est la vitesse des ondes dans l'acier (5415 m/s pour les ondes longitudinales).
 - Δx est le maillage spatial.
 - C_{CFL} est une constante (typiquement entre 0.1 et 1).
- on a $\Delta x = 3, 6, 12 \text{ mm}$ (10^{-3} m), alors :

$$\Delta t_{\text{CFL}} \leq \frac{C_{\text{CFL}} \cdot \Delta x}{c} = \frac{0.1 \times 3 \cdot 10^{-3}}{5415}$$

selon la finesse du maillage :

$$\Delta t_{\text{CFL}} \approx 10^{-7} \text{ à } 1 \cdot 10^{-6} \text{ s}$$

3. En combinant Nyquist et CFL, on choisit :

$\text{ts} \approx 5 \cdot 10^{-7} \text{ s} \rightarrow$ Utilisé pour les simulations explicites.

$5 \cdot 10^{-6}, 10^{-5} \text{ s} \rightarrow$ Bon compromis entre précision et rapidité pour les simulations avec schémas implicites.

$\text{ts} > 3.3 \cdot 10^{-5} \text{ s} \rightarrow$ Risque d'aliasing temporel, 10^{-4} s trop grand, perte de précision.

7.3.4 Estimation du temps final pour observer la propagation des ondes dans la cuve

Pour déterminer ce temps, il faut considérer :

1. La vitesse des ondes dans l'acier (matériau de la cuve).

2. Les dimensions géométriques de la cuve afin d'estimer le temps nécessaire aux ondes pour traverser la structure.

1. Détermination de la vitesse des ondes dans l'acier

L'acier est un matériau élastique dans lequel plusieurs types d'ondes peuvent se propager :

- Ondes longitudinales (ondes de compression) - Ondes transversales (ondes de cisaillement)

Il existe également des ondes de surface, notamment :

Ondes de Rayleigh : Ondes de surface avec un mouvement elliptique, se propageant à la surface de la cuve.

Ondes de Love : Ondes de surface horizontales se propageant dans des matériaux stratifiés, limitées à la surface.

Ondes de Lamb : Ondes de surface ondes de surface qui se propagent à l'intérieur des structures élancées. Elles se propagent à travers la plaque en se déformant selon des modes de flexion.

On s'intéresse principalement aux ondes longitudinales et transversales. Les vitesses de ces ondes sont données par :

$$v_L = \sqrt{\frac{E}{\rho(1-\nu^2)}}$$

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

Avec les valeurs de l'acier :

- Module d'élasticité : $E = 2.1 \times 10^{11} \text{ pa}$

- Densité : $\rho = 7870 \text{ kg/m}^3$

- Coefficient de poisson : $\nu = 0.3$

En remplaçant :

- Vitesse des ondes longitudinales : $v_L \approx 5415 \text{ m/s}$

- Vitesse des ondes transversales : $v_T \approx 3204 \text{ m/s}$

2. Temps de propagation dans la cuve

La cuve a :

- Rayon : $R = \frac{0.944}{2} = 0.472 \text{ m}$ - Hauteur : $H = 0.965 \text{ m}$

On considère que les ondes peuvent se propager dans trois directions principales :

2.1. Autour de la cuve (périmètre circulaire) Distance parcourue :

$$d \approx 2\pi R = 2\pi \times 0.472 \approx 2.96 \text{ m}$$

Temps de propagation :

$$t_{\text{cercle}} = \frac{d}{v_T} = \frac{2.96}{3204} \approx 0.93 \text{ ms}$$

2.2. Du haut vers le bas de la cuve Distance parcourue : $H = 0.965 \text{ m}$ Temps de propagation :

$$t_{\text{hauteur}} = \frac{H}{v_L} = \frac{0.965}{5415} \approx 0.18 \text{ ms}$$

2.3. En diagonale (onde combinant propagation verticale et autour de la cuve) Distance approximée :

$$d_{\text{diag}} = \sqrt{H^2 + d^2} = \sqrt{(0.965)^2 + (2.96)^2} \approx 3.1 \text{ m}$$

Temps de propagation :

$$t_{\text{diag}} = \frac{d_{\text{diag}}}{v_T} = \frac{3.1}{3204} \approx 0.97 \text{ ms}$$

3. Détermination du temps final

- Temps final t_{final} Le but est de pouvoir récupérer 0.5s de signal par la simulation. pour étudier la sensibilité aux paramètres on prendra d'abord 10% de ce temps comme référence.

Comme $t_{\text{diag}} \approx 0.97 \text{ ms}$ est le temps nécessaire pour qu'une onde traverse la structure en diagonale :

$$t_{\text{final}} = 2.5 \text{ ms} = 2.5 \cdot 10^{-3} \text{ s}$$

Est assez pour observer quelques rebonds des ondes dans la cuve.

7.3.5 Algorithme explicite pour la résolution du problème de vibro-acoustique transitoire

Équation d'élastodynamique et discréétisation temporelle

La forme variationnelle au temps t^{n+1} de l'équation d'élasticité est donnée par :

$$\frac{1}{\beta \Delta t^2} \int_{\Omega_s} \rho_s \mathbf{u}_s^{n+1} \cdot \mathbf{v}_s dx + \int_{\Omega_s} \boldsymbol{\sigma}(\mathbf{u}_s^{n+1}) : \nabla \mathbf{v}_s dx = \int_{\Omega_s} \rho_s \left[\frac{1}{\beta \Delta t^2} \mathbf{u}_s^n + \frac{1}{\beta \Delta t} \dot{\mathbf{u}}_s^n + \frac{1 - 2\beta}{2\beta} \ddot{\mathbf{u}}_s^n \right] \cdot \mathbf{v}_s dx + \int_{\Omega_s} f^n \cdot \mathbf{v}_s dx$$

Pour une forme explicite, on prend $\sigma(u_s^{n+1}) = \sigma(u_s^n)$, d'où :

$$\frac{1}{\beta \Delta t^2} \int_{\Omega_s} \rho_s \mathbf{u}_s^{n+1} \cdot \mathbf{v}_s dx = \int_{\Omega_s} \rho_s \left[\frac{1}{\beta \Delta t^2} \mathbf{u}_s^n + \frac{1}{\beta \Delta t} \dot{\mathbf{u}}_s^n + \frac{1 - 2\beta}{2\beta} \ddot{\mathbf{u}}_s^n \right] \cdot \mathbf{v}_s dx + \int_{\Omega_s} f^n \cdot \mathbf{v}_s dx - \int_{\Omega_s} \boldsymbol{\sigma}(\mathbf{u}_s^n) : \nabla \mathbf{v}_s dx$$

Discréétisation spatiale (solide)

Soit $\{\phi_j\}$ une base de $[H^1(\Omega_s)]^3$. On écrit :

$$\mathbf{u}_s(x, t) \approx \sum_j U_j(t) \phi_j(x)$$

En prenant $\mathbf{v}_s = \phi_i$, on obtient :

$$\frac{1}{\beta \Delta t^2} \mathbf{M} \mathbf{u}_s^{n+1} = \mathbf{M} \left[\frac{1}{\beta \Delta t^2} \mathbf{u}_s^n + \frac{1}{\beta \Delta t} \dot{\mathbf{u}}_s^n + \frac{1 - 2\beta}{2\beta} \ddot{\mathbf{u}}_s^n \right] + \mathbf{f}^n - \mathbf{K} \mathbf{u}_s^n$$

avec :

- $\mathbf{M} = \int_{\Omega_s} \rho_s \phi_i \cdot \phi_j dx$: matrice de masse
- $\mathbf{K} = \int_{\Omega_s} \boldsymbol{\sigma}(\phi_j) : \nabla \phi_i dx$: matrice de raideur
- $\mathbf{f}^n = \int_{\Omega_s} f^n \phi_i dx$: terme source

Équation des ondes et discréétisation temporelle

En ignorant les termes ALE :

$$\int_{\Omega_f} \frac{\partial^2 p}{\partial t^2} q dx + c^2 \int_{\Omega_f} \nabla p \cdot \nabla q dx = \rho_f \int_{\Gamma_{fsi}} \frac{\partial \mathbf{u}_s}{\partial t} \cdot \mathbf{n}_s q ds$$

Avec l'approximation :

$$\frac{\partial^2 p}{\partial t^2} \approx \frac{p^{n+1} - 2p^n + p^{n-1}}{\Delta t^2}$$

on obtient :

$$\int_{\Omega_f} p^{n+1} q dx = \int_{\Omega_f} (2p^n - p^{n-1}) q dx - \Delta t^2 c^2 \int_{\Omega_f} \nabla p^n \cdot \nabla q dx + \Delta t^2 \rho_f \int_{\Gamma_{fsi}} \dot{\mathbf{u}}_s^n \cdot \mathbf{n}_f q ds$$

Discrétisation spatiale (fluide)

$$\mathbf{M}\mathbf{p}^{n+1} = \mathbf{M}[2\mathbf{p}^n - \mathbf{p}^{n-1}] + \Delta t^2[\mathbf{V}_{bc}^n - c^2\mathbf{K}\mathbf{p}^n]$$

où :

- $\mathbf{M}_{ij} = \int_{\Omega_f} \psi_i \psi_j dx$: matrice de masse
- $\mathbf{K}_{ij} = \int_{\Omega_f} \nabla \psi_j \cdot \nabla \psi_i dx$: matrice de raideur
- $\mathbf{V}_{bc}^n = \int_{\Gamma_{fsi}} \dot{u}^n \mathbf{n}_s \cdot \psi_i ds$: terme de couplage fluide-structure

Carte ALE

La méthode ALE (Arbitrary Lagrangian-Eulerian) permet de combiner une description lagrangienne au bord du domaine et une description eulérienne à l'intérieur [15]. Elle établit un lien entre le domaine fluide de référence Ω_f^* et le domaine fluide courant Ω_f^t à chaque instant t .

On note \mathcal{A}_t l'application ALE, qui associe à tout point $x^* \in \Omega_f^*$ sa position courante $x \in \Omega_f^t$:

$$\mathcal{A}_t : \Omega_f^* \longrightarrow \Omega_f^t, \quad x = \mathcal{A}_t(x^*) = x^* + \xi(x^*, t),$$

où ξ désigne le champ de déplacement du maillage. L'application \mathcal{A}_t est un homéomorphisme de Ω_f^* vers Ω_f^t .

Extension harmonique. Le champ de déplacement ξ est calculé comme une extension harmonique des déplacements imposés au bord du domaine. Le problème associé s'écrit :

$$\begin{cases} -(1+\tau)\Delta\xi(h, 1) = 0, & \text{dans } \Omega_\delta^*, \\ \xi(h, 1) = g_\xi, & \text{sur } \Gamma_M^*, \\ \xi(h, 1) = 0, & \text{sur } \Gamma_F^*, \\ \nabla\xi(h, 1) \cdot n^* = 0, & \text{sur } \Gamma_N^*. \end{cases}$$

L'application ALE de premier ordre est alors donnée par :

$$\mathcal{A}(h, 1)(x^*) = x^* + \xi(h, 1)(x^*).$$

Transformation ALE d'ordre supérieur. Afin d'obtenir une approximation plus régulière sur le bord, une correction d'ordre supérieur peut être appliquée. On définit pour chaque élément $K^* \in T_{(h,1)}^{*,b}$ (adjacent au bord Γ_M^*) une correction $\mathcal{A}(h_K^*, N)$ solution de :

$$\begin{cases} \int_{K^*} \nabla \mathcal{A}(h_K^*, N) : \nabla z dx = 0, & \forall z \in [\mathbf{P}^N(K^*)]^d, \\ \mathcal{A}(h_K^*, N)(x^*) = x^* + g_\xi - \mathcal{A}(h, 1)(x^*), & \forall x^* \in \partial K^* \cap \Gamma_M^*, \\ \mathcal{A}(h_K^*, N) = 0, & \text{sur le reste de } \partial K^*. \end{cases}$$

La transformation ALE d'ordre N s'écrit alors :

$$\mathcal{A}(h, N)(x^*) = \mathcal{A}(h, 1)(x^*) + \sum_{K^* \in T_{(h,1)}^{*,b}} \mathcal{A}(h_K^*, N)(x^*).$$

Mouvement du maillage ALE

Pour permettre la mise à jour du maillage fluide déformé, on résout :

$$-\nabla \cdot (k \nabla \mathbf{d}_f) = \mathbf{0} \quad \text{dans } \Omega_f,$$

avec la condition de couplage sur l'interface :

$$\mathbf{d}_f = \mathbf{u}_s \quad \text{sur } \Gamma_{fsi}.$$

La vitesse du maillage s'exprime par :

$$\mathbf{w} = \frac{\partial \mathbf{d}_f}{\partial t}.$$

Déplacement ALE :

$$\mathbf{d}_f(\mathbf{x}, 0) = \mathbf{0}.$$

Fluide : Onde acoustique (forme faible ALE) $\mathbf{v}_f \in [H^1(\Omega_f)]^3$: fonction test pour le déplacement ALE.
Trouver $p \in H^1(\Omega_f)$ tel que :

$$\int_{\Omega_f} \frac{\partial^2 p}{\partial t^2} q \, dx + 2 \int_{\Omega_f} \mathbf{w} \cdot \nabla \left(\frac{\partial p}{\partial t} \right) q \, dx + \int_{\Omega_f} (\nabla \cdot \mathbf{w}) \frac{\partial p}{\partial t} q \, dx + c^2 \int_{\Omega_f} \nabla p \cdot \nabla q \, dx = \rho_f \int_{\Gamma_{fsi}} \frac{\partial^2 \mathbf{u}_s}{\partial t^2} \cdot \mathbf{n}_s q \, ds$$

Mouvement ALE (Lissage de Laplace) Trouver $\mathbf{d}_f \in [H^1(\Omega_f)]^3$ tel que :

$$\int_{\Omega_f} k \nabla \mathbf{d}_f : \nabla \mathbf{v}_f \, dx = 0, \quad \text{avec } \mathbf{d}_f = \mathbf{u}_s \text{ sur } \Gamma_{fsi}$$

7.3.6 Amortissement (solide)

Méthode des pics

Extraire les fréquences propres f_n et les largeurs de bande Δf_n des pics de résonance. L'amortissement modal est donné par :

$$\zeta = \frac{\Delta f_n}{2f_n}$$

Décroissance logarithmique

Repérer les oscillations libres, mesurer l'enveloppe décroissante du signal, puis calculer l'amortissement modal par la méthode du logarithme décrémental :

$$\zeta_n = \frac{1}{2\pi n} \ln \left(\frac{u(t)}{u(t+nT)} \right)$$

Relation avec le modèle de Rayleigh

L'amortissement modal est relié au modèle de Rayleigh par :

$$\zeta = \frac{1}{2}(\omega_n \alpha + \beta \omega_n)$$

avec $\omega_n = 2\pi f_n$.

On résout le système suivant pour obtenir α et β :

$$\begin{bmatrix} \frac{1}{2\omega_1} & \omega_1 \\ \frac{1}{2\omega_2} & \omega_2 \\ \vdots & \vdots \\ \frac{1}{2\omega_N} & \omega_N \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_N \end{bmatrix}$$

Ajout d'un amortissement Rayleigh au signal expérimental

L'ajout d'amortissement dans le signal est réalisé en plusieurs étapes :

1. Estimation des coefficients d'amortissement Rayleigh

Les coefficients α et β du modèle de Rayleigh sont estimés à partir des données expérimentales. Pour cela, on extrait les fréquences propres f_n et les taux d'amortissement ζ_n (par FFT ou décrément logarithmique), puis on résout le système linéaire :

$$\zeta_n = \frac{\alpha}{2\omega_n} + \frac{\beta\omega_n}{2}$$

où $\omega_n = 2\pi f_n$.

2. Calcul de la fréquence dominante du signal

La fréquence principale du signal est déterminée par la position du pic maximal dans le spectre de Fourier.

3. Calcul du taux d'amortissement modal

À partir des coefficients estimés et de la fréquence dominante, le taux d'amortissement modal est calculé :

$$\zeta = \frac{\alpha}{2\omega_n} + \frac{\beta\omega_n}{2}$$

4. Application de l'amortissement au signal

Le signal est multiplié par une enveloppe exponentielle décroissante pour simuler la dissipation d'énergie :

$$u_{\text{amorti}}(t) = u(t) \cdot \exp(-\zeta\omega_n t)$$

Cette opération permet d'obtenir un signal amorti, conforme au modèle de Rayleigh.

5. Visualisation

Le code permet de visualiser le signal original et le signal amorti afin de comparer l'effet de l'amortissement.

Cette approche permet de simuler numériquement l'effet de l'amortissement modal observé expérimentalement, en ajustant les paramètres du modèle de Rayleigh à partir des mesures.

Bibliographie

- [1] Semblat, J. F., & Dangla, P. (2005). Modélisation de la propagation d'ondes et de l'interaction solstructure : approches par éléments finis et éléments de frontière. Bulletin des laboratoires des ponts et chaussées, (256-257), 163-178.
- [2] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods : Evolution to Complex Geometries and Applications to Fluid Dynamics*. Springer, 2007.
- [3] Smith, W. D. (1975). The application of finite element analysis to body wave propagation problems. *Geophysical Journal International*, 42(2), 747-768.
- [4] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method : Its Basis and Fundamentals*. Butterworth-Heinemann, 7th edition, 2013.
- [5] T.J.R. Hughes. *The Finite Element Method : Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2012.
- [6] G. Karniadakis and S. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, 2nd edition, 2005.
- [7] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2nd edition, 2003.
- [8] J.S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods : Algorithms, Analysis, and Applications*. Springer, 2007.
- [9] J.P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Publications, 2nd edition, 2001.
- [10] N.M. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*, 85(3) :67–94, 1959.
- [11] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II : Stiff and Differential-Algebraic Problems*. Springer, 2nd edition, 1996.
- [12] C.A. Felippa, K.C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25) :3247–3270, 2001.
- [13] S. Krenk. Energy conservation in newmark based time integration algorithms. HAL Science, 2005.
- [14] Feel++ Benchmarking Documentation. <https://bench.feelpp.org/benchmarking>. Consulté en août 2025.
- [15] Vincent Chabannes. Vers la simulation des écoulements sanguins. Médecine humaine et pathologie. Université de Grenoble, 2013. Français. (NNT : 2013GRENM061). (tel-00923731v2)