

Méthode d'éléments finis discontinus hybridisée pour la résolution itérative accélérée de problèmes d'ondes en fréquence

A. Modave^{1,*}, R. Greffe², C. Geuzaine², A. Rappaport¹, A. Chabib¹

¹ POEMS, CNRS, Inria, ENSTA, Institut Polytechnique de Paris, 91120 Palaiseau, France

² Institut Montefiore, Université de Liège, Belgique

* Correspondance : axel.modave@ensta.fr

Résumé — La discrétisation de problèmes de propagation d'ondes en régime harmonique par éléments finis conduit à des systèmes linéaires coûteux à résoudre. Nous considérons une méthode d'éléments finis discontinus hybridisée en utilisant des variables de transmission aux interfaces entre les mailles. Cette approche permet d'accélérer la convergence des schémas itératifs et possède une structure algorithmique adaptée au calcul parallèle, notamment sur cartes graphiques. Nous présentons et étudions des implémentations de cette méthode au moyen de résultats 3D obtenus avec un code C++ dédié.

Mots clés — Propagation des ondes ; Éléments finis discontinus ; Méthode d'ordre élevé ; HPC ; GPU.

1 Introduction

Les méthodes d'éléments finis sont largement utilisées pour étudier des phénomènes de propagation d'ondes. Cependant, lorsqu'on s'intéresse aux problèmes réalistes en fréquence, ces méthodes conduisent à des systèmes linéaires creux de très grande taille dont la résolution est difficile. D'une part, les solveurs directs sont coûteux et complexes à mettre en œuvre sur des architectures de calcul parallèles. D'autre part, les solveurs itératifs classiques, qui sont simples à paralléliser, convergent lentement en raison de propriétés intrinsèques des problèmes de propagation d'ondes en régime harmonique, e.g. [5].

Nous nous intéressons aux approches permettant d'accélérer la résolution itérative de tels problèmes en tirant parti des architectures parallèles modernes composées de processeurs multicœurs (CPU) et de cartes graphiques (GPU). Les recherches sur ce sujet portent principalement sur des techniques de préconditionnement et de décomposition de domaine. Malheureusement, les techniques généralement développées pour les problèmes coercifs ne fonctionnent pas pour les problèmes en fréquence, et des approches *ad hoc* sont nécessaires, e.g. [2, 7, 10]. Parallèlement, des méthodes d'éléments finis alternatives, utilisant notamment des fonctions de base discontinues, sont développées. Ce domaine de recherche est très actif.

Dans le cadre du projet ANR WavesDG, nous travaillons sur une méthode de type Galerkin discontinu hybridisée (HDG) avec des flux numériques de type *upwind*. L'hybridisation consiste à introduire une variable supplémentaire aux interfaces entre les éléments afin de découpler les inconnues physiques correspondants à des éléments différents. Ces inconnues sont ensuite éliminées du système dans un processus de condensation statique, menant à un système hybridisé aux propriétés différentes. Avec l'approche HDG standard, e.g. [1, 4, 6], la variable supplémentaire est une trace numérique. Avec notre approche, appelée CHDG [11], il s'agit d'une variable de transmission.

L'approche CHDG augmente la taille du système hybridisé qu'il faut résoudre, mais les expériences numériques indiquent que les solveurs itératifs convergent plus rapidement. De plus, la structure algorithmique des schémas de résolution, qui est proche de celle des solveurs DG développés pour les problèmes en temps [9, 12, 13], est adaptée au calcul parallèle sur CPU et GPU. Nous pensons qu'il s'agit d'une approche prometteuse pour des simulations rapides utilisant pleinement les ressources des architectures de calcul modernes.

Dans cet article, nous présentons et discutons les principaux ingrédients de la méthode CHDG (Section 2), des stratégies de mise en œuvre (Section 3) et quelques résultats préliminaires obtenus avec un code de calcul parallèle C++ dédié (Section 4).

2 Méthode CHDG

Problème considéré. Pour décrire la méthode, nous considérons un problème d'ondes scalaires en régime harmonique sur un domaine borné $\Omega \subset \mathbb{R}^3$,

$$\begin{cases} -ikp + \nabla \cdot \mathbf{u} = 0, & \text{dans } \Omega, \\ -ik\mathbf{u} + \nabla p = 0, & \text{dans } \Omega, \\ p - \mathbf{n} \cdot \mathbf{u} = s, & \text{sur } \partial\Omega, \end{cases} \quad (1)$$

avec la pression $p(\mathbf{x})$, la vitesse particulière $\mathbf{u}(\mathbf{x})$, le nombre d'ondes $k = \omega/c$, la pulsation ω , la vitesse de phase c et un terme source $s(\mathbf{x})$. Nous supposons une dépendance temporelle $e^{i\omega t}$. La méthode peut s'étendre à d'autres conditions limites [11], aux milieux hétérogènes [16] et à d'autres types d'ondes [15, 17].

Éléments finis discontinus. Le problème est discrétisé sur un maillage composé de tétraèdres. Les mailles et leurs faces sont notées K et F , respectivement. L'ensemble \mathcal{K} contient toutes les mailles du maillage. L'ensemble \mathcal{F}_K contient toutes les faces de la maille K . Pour définir la formulation variationnelle du problème, nous utiliserons la notation

$$(u, v)_K := \int_K u \bar{v} d\mathbf{x} \quad \text{et} \quad \langle u, v \rangle_F := \int_F u \bar{v} d\mathbf{x}, \quad (2)$$

pour tout champs scalaires u et v , et une notation similaire pour les champs vectoriels.

Sur chaque maille $K \in \mathcal{K}$, les champs de pression et de vitesse particulière sont représentés par des fonctions polynomiales complexes de degré maximum p . Ils vérifient le *problème local* suivant :

$$\left\{ \begin{array}{l} \text{Trouver } p_K \in \mathbb{P}_p(K) \text{ et } \mathbf{u}_K \in [\mathbb{P}_p(K)]^3 \text{ tel que} \\ \begin{cases} -ik(p_K, q_K)_K - (\mathbf{u}_K, \nabla q_K)_K + \sum_{F \in \mathcal{F}_K} \langle \mathbf{n}_{K,F} \cdot \hat{\mathbf{u}}_F, q_K \rangle_F = 0, \\ -ik(\mathbf{u}_K, \mathbf{v}_K)_K - (p_K, \nabla \cdot \mathbf{v}_K)_K + \sum_{F \in \mathcal{F}_K} \langle \hat{p}_F, \mathbf{n}_{K,F} \cdot \mathbf{v}_K \rangle_F = 0, \end{cases} \\ \text{pour tout } q_K \in \mathbb{P}_p(K) \text{ et } \mathbf{v}_K \in [\mathbb{P}_p(K)]^3, \end{array} \right. \quad (3)$$

où $\mathbf{n}_{K,F} \cdot \hat{\mathbf{u}}_F$ et \hat{p}_F sont les *flux numériques* (à définir). Le *problème global* est composé de tous les problèmes locaux (3).

Flux upwind et variables de transmission. Les flux numériques permettent d'imposer faiblement les conditions de continuité aux interfaces entre les éléments et les conditions aux limites à la frontière du domaine. Ils sont choisis de manière à assurer la consistance et la stabilité du schéma.

Nous considérons des flux numériques de type *upwind* défini par

$$\hat{p}_F := \begin{cases} \frac{1}{2}(p_K + p_{K'}) + \frac{1}{2} \mathbf{n}_{K,F} \cdot (\mathbf{u}_K - \mathbf{u}_{K'}) & \text{si } F \not\subset \partial\Omega \text{ avec } F = K \cap K', \\ (p_K + \mathbf{n}_{K,F} \cdot \mathbf{u}_K + s)/2 & \text{si } F \subset \partial\Omega, \end{cases} \quad (4a)$$

$$\mathbf{n}_{K,F} \cdot \hat{\mathbf{u}}_F := \begin{cases} \frac{1}{2} \mathbf{n}_{K,F} \cdot (\mathbf{u}_K + \mathbf{u}_{K'}) + \frac{1}{2} (p_K - p_{K'}) & \text{si } F \not\subset \partial\Omega \text{ avec } F = K \cap K', \\ (p_K + \mathbf{n}_{K,F} \cdot \mathbf{u}_K - s)/2 & \text{si } F \subset \partial\Omega, \end{cases} \quad (4b)$$

où $\mathbf{n}_{K,F}$ est la normale unitaire extérieure à la maille K sur la face F . Si F est une face intérieure (i.e. $F \not\subset \partial\Omega$), alors K' est l'élément voisin qui partage cette face avec K .

Ces flux s'obtiennent en utilisant la technique du *flux splitting*, qui permet de traiter séparément les ondes sortantes et entrantes aux interfaces. Ils peuvent se réécrire

$$\hat{p}_F = \frac{1}{2} (g_{K,F}^{\oplus} + g_{K,F}^{\ominus}), \quad (5a)$$

$$\mathbf{n}_{K,F} \cdot \hat{\mathbf{u}}_F = \frac{1}{2} (g_{K,F}^{\oplus} - g_{K,F}^{\ominus}), \quad (5b)$$

avec les *variables de transmission sortante et entrante*, respectivement notées $g_{K,F}^{\oplus}$ et $g_{K,F}^{\ominus}$, définies par

$$g_{K,F}^{\oplus} := p_K + \mathbf{n}_{K,F} \cdot \mathbf{u}_K, \quad (6a)$$

$$g_{K,F}^{\ominus} := \begin{cases} g_{K',F}^{\oplus} & \text{si } F \not\subset \partial\Omega \text{ avec } F = K \cap K', \\ s & \text{si } F \subset \partial\Omega. \end{cases} \quad (6b)$$

Ces variables sont définies pour chaque face F de chaque élément K . La variable sortante est définie par les champs physiques de la maille locale, alors que la variable entrante est définie par ceux de la maille voisine ou par la conditions aux limites. Ces variables peuvent être interprétées comme des données entrantes et sortantes qui traversent chaque face.

Hybridisation. La résolution directe du problème global nécessite de résoudre un système linéaire dont les inconnues sont associées aux champs physiques. Au lieu de faire cela, nous introduisons une variable supplémentaire sur les faces du maillage de manière à découpler les champs physiques des maillages voisins. Cela permet d'éliminer les inconnues associées aux champs physiques en résolvant des problèmes locaux découplés, à la manière d'une *condensation statique* ou d'un *complément de Schur*. Il en résulte un *problème hybridisé* dont les inconnues sont associées à la variable supplémentaire.

Avec la méthode HDG (*Hybridized Discontinuous Galerkin*) classique, la variable supplémentaire correspond au flux numérique \widehat{p}_F . L'approche est généralement motivée par la réduction du nombre d'inconnues du système hybridisé résultant. Pour le problème d'ondes scalaires, ces inconnues sont associées à un unique champs scalaire inconnu défini sur chaque face du maillage.

Dans ce travail, nous considérons une stratégie alternative, appelée CHDG¹, qui consiste à utiliser la variable de transmission entrante $g_{K,F}^\ominus$ comme variable d'hybridisation. Comparé à la méthode HDG classique, le nombre d'inconnues augmente : pour chaque face intérieure du maillage, il y a deux variables de transmissions entrantes, chacune étant associée à l'un des côtés de la face. Cependant, le système hybridisé obtenu avec l'approche CHDG possède de meilleures propriétés pour une résolution itérative. Nous observons que les schémas itératifs convergent plus rapidement.

Problème hybridisé. Le problème hybridisé est défini sur le squelette du maillage. Avec l'approche CHDG, l'inconnue, notée

$$g_h \in \mathcal{G}_h := \bigcup_{K \in \mathcal{X}} \bigcup_{F \in \mathcal{F}_K} \mathbb{P}_p(F), \quad (7)$$

correspond à la variable de transmission entrante $g_h|_{K,F} := g_{K,F}^\ominus$ définie par (6b) sur chaque face de chaque maille. En introduisant cette variable dans le problème local (3), celui-ci se réécrit

$$\left\{ \begin{array}{l} \text{Trouver } p_K \in \mathbb{P}_p(K) \text{ et } \mathbf{u}_K \in [\mathbb{P}_p(K)]^3 \text{ tel que} \\ \left\{ \begin{array}{l} -ik(p_K, q_K)_K - (\mathbf{u}_K, \nabla q_K)_K + \sum_{F \in \mathcal{F}_K} \frac{1}{2} \langle p_K + \mathbf{n}_{K,F} \cdot \mathbf{u}_K, q_K \rangle_F \\ \quad = \sum_{F \in \mathcal{F}_K} \frac{1}{2} \langle g_{K,F}^\ominus, q_K \rangle_F, \\ -ik(\mathbf{u}_K, \mathbf{v}_K)_K - (p_K, \nabla \cdot \mathbf{v}_K)_K + \sum_{F \in \mathcal{F}_K} \frac{1}{2} \langle (p_K + \mathbf{n}_{K,F} \cdot \mathbf{u}_K), \mathbf{n}_{K,F} \cdot \mathbf{v}_K \rangle_F \\ \quad = -\sum_{F \in \mathcal{F}_K} \frac{1}{2} \langle g_{K,F}^\ominus, \mathbf{n}_{K,F} \cdot \mathbf{v}_K \rangle_F, \end{array} \right. \\ \text{pour tout } q_K \in \mathbb{P}_p(K) \text{ et } \mathbf{v}_K \in [\mathbb{P}_p(K)]^3. \end{array} \right. \quad (8)$$

Les problèmes locaux sont donc découplés. En utilisant la définition (6b), nous obtenons une expression du problème hybridisé :

$$\left\{ \begin{array}{l} \text{Trouver } g_h \in \mathcal{G}_h \text{ tel que} \\ \sum_{K \in \mathcal{X}} \sum_{F \in \mathcal{F}_K} \langle (\mathbf{I} - \mathbf{S})g_h, \xi_h \rangle_F = \sum_{K \in \mathcal{X}} \sum_{F \in \mathcal{F}_K} \langle b, \xi_h \rangle_F, \\ \text{pour tout } \xi_h \in \mathcal{G}_h, \end{array} \right. \quad (9)$$

où les opérateurs \mathbf{I} , \mathbf{S} et $\mathbf{\Pi}$ et le terme b sont définis par

$$(\mathbf{I}g_h)|_{K,F} := g_{K,F}, \quad (\text{opérateur identité}) \quad (10a)$$

$$(\mathbf{S}g_h)|_{K,F} := p_K(g_h) + \mathbf{n}_{K,F} \cdot \mathbf{u}_K(g_h), \quad (\text{opérateur de diffraction}) \quad (10b)$$

$$(\mathbf{\Pi}g_h)|_{K,F} := \begin{cases} g_{K',F} & \text{si } F \not\subset \partial\Omega \text{ avec } F = K \cap K', \\ 0 & \text{si } F \subset \partial\Omega, \end{cases} \quad (\text{opérateur d'échange}) \quad (10c)$$

$$b|_{K,F} := \begin{cases} 0 & \text{si } F \not\subset \partial\Omega, \\ s & \text{si } F \subset \partial\Omega, \end{cases} \quad (\text{source globale}) \quad (10d)$$

1. Le 'C' provient du fait que les variables de transmission sont reliées aux variables caractéristiques des solveurs de Riemann, qui sont utilisés pour définir les flux *upwind*.

où $(p_K(g_h), \mathbf{u}_K(g_h))$ est la solution du problème local (8) pour la donnée g_h .

Les opérateurs agissent sur \mathcal{G}_h . L’opérateur S peut s’interpréter comme un opérateur “*variable entrante vers variable sortante*”, qui demande la résolution des problèmes locaux. L’application de l’opérateur Π revient à transférer des données entre les éléments voisins. Nous pouvons prouver que l’opérateur ΠS est strictement contractant pour une norme L^2 définie sur les faces, voir [11] pour plus de détails. Cette propriété implique que le problème est bien posé, et qu’il peut être résolu avec un point fixe.

3 Mise en œuvre

Système hybridisé. Une base polynomiale est choisie pour représenter l’inconnue g_h et pour définir les fonctions de test du problème hybridisé (9). Cela conduit au *système linéaire*

$$\mathbf{M}[\mathbf{I} - \Pi S] \mathbf{g} = \mathbf{M} \mathbf{b}, \quad (11)$$

où \mathbf{g} contient les coefficients de g_h sur la base considérée, et \mathbf{b} contient ceux de la projection de b , \mathbf{M} contient les matrices de masse associées aux faces, et Π et S sont les versions matricielles de Π et S . Les matrices \mathbf{M} et \mathbf{S} sont diagonales par blocs, la taille de chaque bloc dépendant du nombre d’inconnues discrètes par face. L’application de la matrice \mathbf{S} demande la résolution de la version discrète du problème local (8) pour tous les éléments du maillage. La matrice Π est binaire et creuse.

Dans ce travail, nous utilisons des *fonctions de base nodales* avec lesquelles les inconnues discrètes correspondent aux valeurs des champs sur des nœuds définis sur les éléments. Ces fonctions sont utilisées pour représenter les champs physiques (sur les éléments) et g_h (sur les faces). Ce choix permet d’appliquer des stratégies d’implémentation utilisées dans des codes de calcul DG écrits pour les problèmes en temps. En particulier, ces stratégies simplifient la gestion des orientations des éléments, l’évaluation de la trace des champs physiques aux faces, les échanges des données aux interfaces, et le calcul des matrices élémentaires, voir [8].

Schémas itératifs. Dans ce travail, différents schémas itératifs sont comparés pour résoudre le système (11). D’abord, nous considérons le *point fixe*,

$$\mathbf{g}^{\ell+1} = \Pi S \mathbf{g}^\ell + \mathbf{b}, \quad \ell = 0, 1, \dots \quad (12)$$

pour une solution initiale \mathbf{g}^0 donnée. Puisque ΠS est une contraction stricte [11], ce schéma converge toujours. En pratique, les méthodes de Krylov sont largement utilisées pour résoudre les problèmes de propagation d’ondes en régime harmonique. Nous considérons le *schéma CGNR*, qui consiste à appliquer un gradient conjugué à l’équation normale $(\mathbf{A}^* \mathbf{A}) \mathbf{g} = \mathbf{A}^* \mathbf{M} \mathbf{b}$ avec $\mathbf{A} := \mathbf{M}[\mathbf{I} - \Pi S]$, et le *schéma GMRES*. Dans les deux cas, nous utilisons un préconditionnement symétrique avec \mathbf{M} , qui peut s’interpréter comme une réécriture du système (11) dans une base polynomiale orthonormale. L’empreinte mémoire du CGNR est plus petite que celle du GMRES, car ce dernier requiert le stockage des vecteurs d’une base de Krylov, mais elle nécessite l’application de la matrice adjointe.

Dans [11], il est observé que CGNR est presque aussi efficace que GMRES en terme de nombre d’itérations pour la résolution du système hybridisé sur un ensemble de benchmarks 2D. En pratique, pour des problèmes réalistes, la conclusion dépendra aussi des temps de calcul et des ressources disponibles.

Programmation parallèle. La construction du système hybridisé et les schémas itératifs ont été implémentés dans un code dédié écrit en C++. Deux versions sont actuellement en cours de développement. Une version CPU qui utilise `MPI` et `OpenMP` pour la programmation parallèle et la bibliothèque `mk1` pour la résolution des systèmes locaux. Une version GPU qui utilise `CUDA` pour la programmation parallèle et `MAGMA` pour la résolution des systèmes locaux.

Dans les deux versions du code, les applications des matrices globales sur tout vecteur sont implémentées dans des fonctions dédiées. Des stratégies similaires sont utilisées :

- L’application de \mathbf{M} se fait en reconstruisant/appliquant les matrices de masse locales à chaque appel. Le passage à l’élément de référence permet de ne stocker que la matrice de référence.
- L’application de Π est un mouvement de données aux interface entre les éléments. D’un point de vue algorithmique, ce mouvement est similaire à celui de solveurs DG en temps, voir e.g. [12].

— Pour l’application de \mathbf{S} , deux alternatives sont considérées :

- (1) Les systèmes locaux sont assemblés et résolus en parallèle à chaque appel de la fonction. Cette approche “*On The Fly*” (OTF) minimise la quantité de données à conserver tout au long du calcul, mais elle implique de répéter des opérations identiques à chaque itération.
- (2) Les blocs diagonaux de \mathbf{S} sont précalculés lors d’une phase d’initialisation au cours de laquelle les matrices des systèmes locaux sont assemblées et inversées. Ces blocs sont conservés tout au long du calcul et appliqués à chaque appel de la fonction. Cette approche “*Input To Output*” (I2O) augmente l’empreinte mémoire, mais réduit la quantité de calcul par itération.

Les matrices locales sont calculées à l’aide de stratégies similaires à celles décrites dans [8].

Plus d’informations sur l’implémentation GPU se trouvent dans [3] et dans article en préparation.

4 Quelques résultats préliminaires

Validation et comparaison des schémas itératifs. Nous considérons deux problèmes de référence définis sur le domaine $[0, 1]^3$: la propagation d’une onde en espace libre (conditions de Robin non-homogène et $k = 10\pi$) et une cavité (conditions de Dirichlet homogène, source volumique unitaire et $k = 5,5\sqrt{2}\pi$), voir figure 1. Dans les deux cas, la taille caractéristique d’un élément est $h = 1/8$ et le degré polynomial est $p = 6$. Le maillage est composé de 2716 tétraèdres, le système hybridisé possède

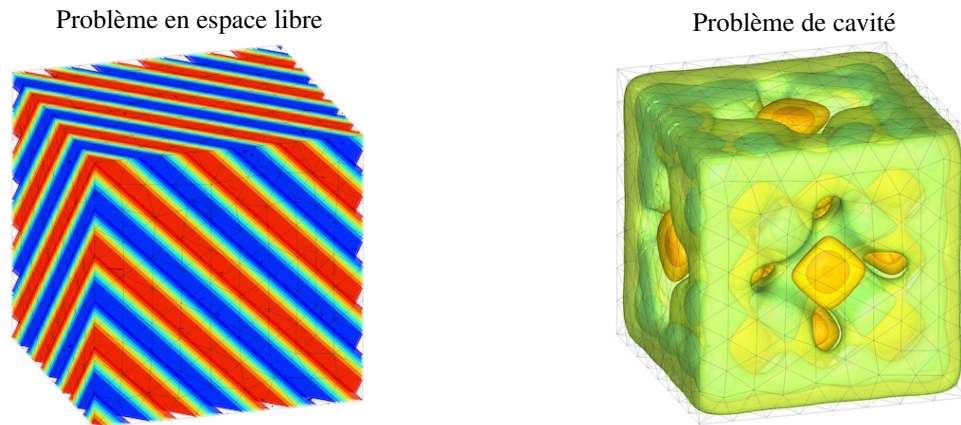


FIGURE 1 – Validation : Partie réelle des solutions des problèmes de référence. Les surfaces d’isovaleurs sont représentées à l’intérieur du maillage de la frontière.

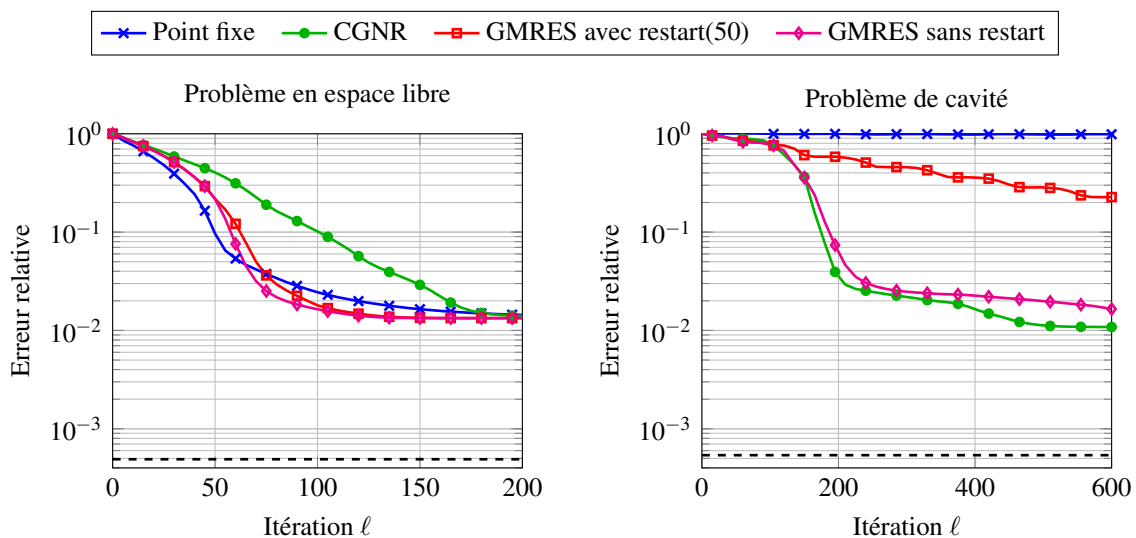


FIGURE 2 – Validation : Erreur sur $u(x)$ en norme L^2 en fonction de l’itération avec différents schémas itératifs. La solution numérique est comparée à la solution analytique de référence. Les lignes pointillées correspondent à l’erreur de la projection de la solution analytique sur l’espace d’approximation.

304 192 inconnues, et le problème physique de départ possède 912 576 inconnues (336 par élément).

La décroissance de l'erreur en fonction du nombre d'itérations est présentée sur la figure 2 pour les différents schémas itératifs : point fixe, CGNR et GMRES avec ou sans redémarrage. L'erreur est calculée en comparant la solution obtenue à chaque itération à la solution analytique de référence, elle ne peut donc pas descendre en dessous du niveau qui serait obtenu avec un solveur direct. L'objectif est d'atteindre ce niveau le plus rapidement possible. L'erreur correspondant à la meilleure approximation sur l'espace d'approximation est donnée à titre indicatif.

On observe que le point fixe converge pour les deux problèmes, mais il est très lent pour la cavité. La convergence de GMRES est légèrement plus rapide que celle de CGNR pour le problème en espace libre, mais elle est significativement plus lente pour la cavité si un restart de 50 est utilisé. Cette comparaison préliminaire ne mène pas à une conclusion claire. D'autres configurations, la comparaison des temps de calcul et la combinaison avec des méthodes de décomposition de domaine seront étudiées prochainement.

Performance de calcul sur GPU. Nous avons mesuré les temps d'initialisation et de calcul par itération d'un point fixe pour les deux alternatives d'implémentation du code GPU : l'une sans pré-calcul de la matrice de diffraction \mathbf{S} (stratégie OTF) et l'autre avec pré-calcul (stratégie I2O). Les temps sont présentés en fonction de la taille du maillage pour différents degrés polynomiaux (figure 3).

Dans tous les cas, la phase d'initialisation est négligeable avec la stratégie OTF, mais le temps de calcul par itération est environ deux ordres de grandeur plus élevé qu'avec la stratégie I2O. La stratégie I2O est donc clairement plus intéressante lorsque de nombreuses itérations sont nécessaires. À titre d'exemple, la résolution du problème en espace libre avec 200 itérations du point fixe prend environ 17 secondes avec OTF (le temps est principalement associé aux itérations) et environ une seconde avec I2O (le temps est principalement associé à la phase d'initialisation).

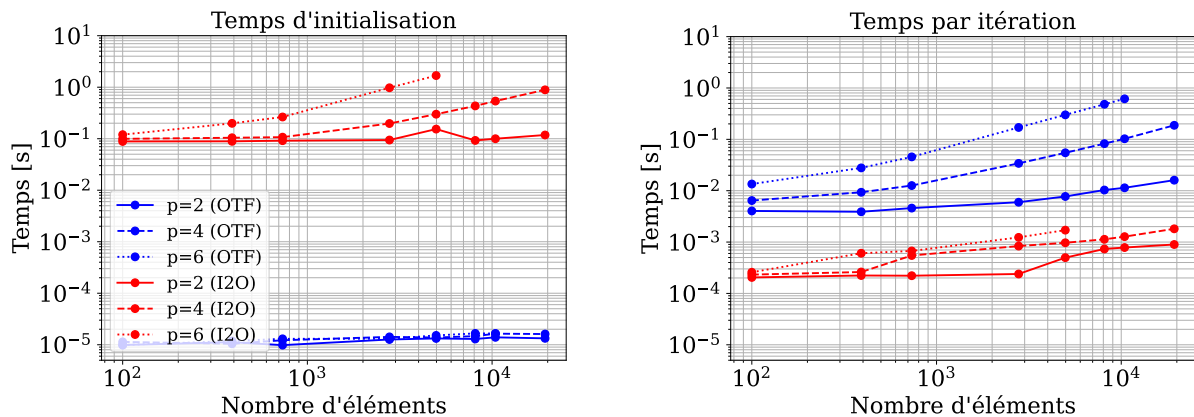


FIGURE 3 – Performance GPU : Temps d'initialisation et temps de calcul par itération du point fixe avec les deux stratégies d'implémentation, en fonction de la taille du maillage, pour des degrés polynomiaux différents. Ces résultats sont obtenus en utilisant un GPU NVIDIA A100 sur un nœud du cluster Lucia.

Illustration d'application. À titre d'illustration, nous considérons la propagation du bruit généré par un turboréacteur. Pour ce résultat préliminaire, la simulation est réalisée sans écoulement, en utilisant la géométrie d'un réacteur [14, 10] placée dans un domaine de calcul cylindrique. Le bruit est généré par une condition de Dirichlet à l'intérieur du réacteur. Une condition absorbante est définie sur l'extérieur du modèle. Le maillage est composé de 497 163 tétraèdres, le degré polynomial est $p = 4$, le système hybridisé possède 29 829 780 inconnues, et le problème physique de départ en possède 69 602 820 (140 par élément). Le schéma itératif est CGNR. La convergence vers la solution est illustrée sur la figure 4.

La simulation a été réalisée sur un nœud du cluster Cholesky du mésocentre IDCS, composé de deux processeurs Intel Xeon Gold 6230 à 20 cœurs. Nous avons utilisé la version CPU du code, une parallélisation avec 40 processus MPI, sans pré-calcul de \mathbf{S} . La phase d'initialisation a duré environ une seconde, et chaque itération demande 9.94 secondes. La solution finale est obtenue après environ 1 000 itérations et 2h45 de calcul. Ce temps pourra être considérablement réduit en effectuant un pré-calcul de \mathbf{S} et en utilisant la version GPU du code sur un cluster composé de plusieurs GPU.

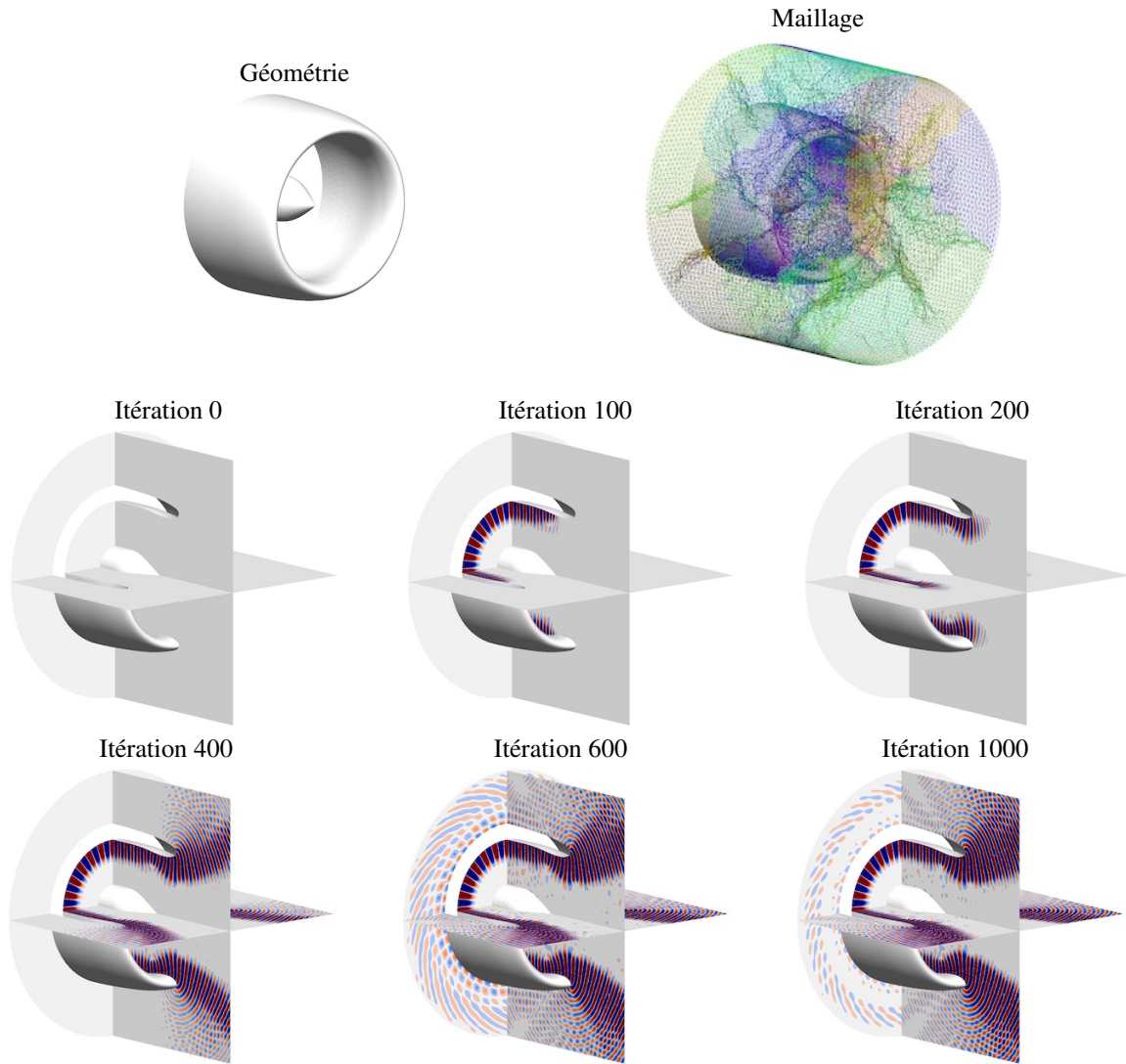


FIGURE 4 – Illustration d’application : géométrie, maillage, et partie réelle de la solution du problème au fur et à mesure des itérations.

Remerciements

Ce travail a été soutenu par le projet ANR JCJC WavesDG (ANR-21-CE46-0010). Dans le cadre de France 2030, ce travail a bénéficié d’une aide de l’Etat gérée par l’Agence Nationale de la Recherche et attribuée au projet Exa-MA (qui fait partie du programme prioritaire de recherche NumPEX) sous la référence ANR-22-EXNU-0002. Ce travail a bénéficié de l’accès aux ressources de calcul de l’unité de service IDCS de l’Ecole polytechnique.

Les moyens de calcul ont été fournis par le Consortium des Équipements de Calcul Intensif (CÉCI), financé par le Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) sous la convention n° 2.5020.11 et par la Région Wallonne. Les présents travaux ont bénéficié de moyens de calcul mis à disposition sur Lucia, le supercalculateur Tier-1 de la Région Wallonne, infrastructure financée par la Région wallonne sous la convention n°1910247.

Références

- [1] H. Barucq, N. Rouxelin, S. Tordeux. *Construction and analysis of a HDG solution for the total-flux formulation of the convected Helmholtz equation*. Mathematics of Computation, 92(343), 2097-2131, 2023. [hal:04006555](https://hal.archives-ouvertes.fr/hal-04006555)
- [2] Y. Boubendir, X. Antoine, C. Geuzaine. *A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation*. Journal of Computational Physics, 231(2), 262-280, 2012.
- [3] A. Chabib, T. Greffe, C. Geuzaine, A. Modave. *Portage GPU d’un solveur éléments finis discontinus hybridisé pour les problèmes d’ondes en fréquence*. Actes du 26e Congrès Français de Mécanique (CFM2025), Août

2025, Metz, France. [\(hal:05235205\)](#)

- [4] B. Cockburn. *Static condensation, hybridization, and the devising of the HDG methods*. in *Building bridges : connections and challenges in modern approaches to numerical partial differential equations*. Springer International Publishing, 129-177, 2016.
- [5] O. G. Ernst, M. J. Gander. *Why it is difficult to solve Helmholtz problems with classical iterative methods*. Numerical analysis of multiscale problems, 325-363, 2011. [\(doi:10.1007/978-3-642-22061-6_10\)](#)
- [6] P. Fernandez, A. Christophe, S. Terrana, N. C. Nguyen, J. Peraire. *Hybridized discontinuous Galerkin methods for wave propagation*. Journal of Scientific Computing, 77(3), 1566-1604, 2018.
- [7] M. J. Gander, Z. Hui. *A class of iterative solvers for the Helmholtz equation : Factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods*. SIAM Review, 61, 3-76, 2019. [\(arXiv:1610.02270\)](#)
- [8] J. S. Hesthaven, T. Warburton. *Nodal discontinuous Galerkin methods : algorithms, analysis, and applications*. Springer New York, 2008. [\(doi:10.1007/978-0-387-72067-8\)](#)
- [9] A. Klöckner, T. Warburton, J. Bridge, J. S. Hesthaven. *Nodal discontinuous Galerkin methods on graphics processors*. Journal of Computational Physics, 228(21), 7863-7882, 2009. [\(arXiv:0901.1024\)](#)
- [10] P. Marchner, H. Bériot, S. Le Bras, X. Antoine, C. Geuzaine. *A domain decomposition solver for large scale time-harmonic flow acoustics problems*. SIAM Journal on Scientific Computing, 47(2), B333-B359, 2025.
- [11] A. Modave, T. Chaumont-Frelet. *A hybridizable discontinuous Galerkin method with characteristic variables for Helmholtz problems*. Journal of Computational Physics, 493, 112459, 2023. [\(hal:03909368\)](#)
- [12] A. Modave, A. St-Cyr, W. A. Mulder, T. Warburton. *A nodal discontinuous Galerkin method for reverse-time migration on GPU clusters*. Geophysical Journal International, 203(2), 1419-1435, 2015. [\(arXiv:1506.00907\)](#)
- [13] A. Modave, A. St-Cyr, T. Warburton. *GPU performance analysis of a nodal discontinuous Galerkin method for acoustic and elastic models*. Computers & Geosciences, 91, 64-76, 2016. [\(arXiv:1602.07997\)](#)
- [14] P. Mustafi. *Improved turbofan intake liner design and optimization*. PhD thesis, University of Southampton, 2013.
- [15] S. Pescuma. *Discontinuous Galerkin finite element methods with transmission variables for time-harmonic wave propagation*. Thèse de l'Institut Polytechnique de Paris. 2025.
- [16] S. Pescuma, G. Gabard, T. Chaumont-Frelet, A. Modave. *A hybridizable discontinuous Galerkin method with transmission variables for time-harmonic acoustic problems in heterogeneous media*. Journal of Computational Physics, 534, 114009, 2025. [\(hal:04821539\)](#)
- [17] A. E. Rappaport, T. Chaumont-Frelet, A. Modave. *A hybridizable discontinuous Galerkin method with transmission variables for time-harmonic electromagnetic problems*. 2025. [\(hal:05016261\)](#)