High order discontinuous Galerkin methods for time-harmonic elastodynamics

Marie Bonnasse-Gahot

UNIVERSITÉ NICE - SOPHIA ANTIPOLIS - UFR SCIENCES
École Doctorale des Sciences Fondamentales et Appliquées - ED 364

THÈSE

Pour l'obtention du titre de
DOCTEUR EN SCIENCES
de L'UNIVERSITÉ Nice Sophia Antipolis
Spécialité : Mathématiques Appliquées

Présentée et soutenue par
Marie BONNASSE-GAHOT

SIMULATION DE LA PROPAGATION D'ONDES ELASTIQUES EN DOMAINE FREQUENTIEL PAR DES METHODES GALERKINE DISCONTINUES

Thèse dirigée par : **M. Henri CALANDRA**
M. Julien DIAZ
M. Stéphane LANTERI

préparée à l'INRIA Bordeaux-Sud-Ouest
et à l'INRIA Méditerranée-Sophia Antipolis

Soutenue le 15 décembre 2015

Devant le jury :

Mme Hélène BARUCQ	Directrice de Recherche, INRIA	Examinatrice
M. Henri CALANDRA	Ingénieur de Recherche, TOTAL E&P USA	Directeur de thèse
M. Gilles CARBOU	Professeur, Université de Pau et des Pays de l'Adour	Président du jury
M. Julien DIAZ	Chargé de recherche, INRIA	Directeur de thèse
M. Christophe GEUZAINE	Professeur, University of Liege, Belgique	Rapporteur
M. Russell HEWETT	Ingénieur de Recherche, TOTAL E&P USA	Examinateur
M. Stéphane LANTERI	Directrice de Recherche, INRIA	Directeur de thèse
M. Jean VIRIEUX	Professeur, Université Joseph Fourier, Grenoble	Rapporteur

C'est en s'éloignant des choses simples qu'on se rend compte à quel point elles sont magnifiques.

Remerciements

Mes premiers remerciements s'adressent à Hélène Barucq, Henri Calandra, Julien Diaz et Stéphane Lantéri qui m'ont permis de réaliser cette thèse. Je les remercie énormément et avec insistance pour la confiance qu'ils m'ont accordé pour cette thèse; confiance qu'ils me renouvellent pour un nouveau projet. Je les remercie également de m'avoir donné l'opportunité de participer à de nombreuses conférences nationales et internationales.
Je remercie particulièrement mes deux directeurs de thèse, Julien et Stéphane, pour toute leur disponibilité, leurs nombreux conseils et explications et toute la patience dont ils ont fait preuve durant ces trois ans. Merci aussi pour les efforts pour comprendre mes explications parfois (souvent) compliquées! Un très grand merci à Julien pour tout le temps qu'il m'a consacré et la patience dont il a fait preuve pour m'aider dans mon travail et, surtout dans mes débugs.

Je remercie ensuite tous les membres de mon jury d'avoir été présents le jour de ma soutenance. Merci à Monsieur C. Guezaine et Monsieur J. Virieux d'avoir accepté d'être rapporteur, merci à Monsieur G. Carbou d'avoir accepté d'être le président du jury et merci à Monsieur R. Hewitt d'avoir accepté d'être membre et d'avoir fait ce long déplacement.

Je remercie tous les membres de l'équipe Magique 3D (anciens et nouveaux :)) de m'avoir intégré dans l'équipe et, surtout, pour leur bonne humeur quotidienne! Un grand merci à ceux avec qui j'ai partagé le bureau et qui m'ont supporté râler pendant mes longues heures de débogage :) : les premiers d'abord Elodie, Lionel et Florent. Et puis ceux qui sont arrivés après: Vincent et Florian (sans oublier nos grands débâts: le premier sur une marque à la pomme dont nous ne citerons pas le nom ;) (mais au final je crois qu'on a gagné avec Vincent :p) et le deuxième sur ma conversion au tikz: mes figures ne sont pas trop mal finalement, non? :p. *We study wave equations* cf. Florian!), Izar (y grazias por las discusiones y todos los secretos ;)) et Ha. Un grand merci à Ha et Izar pour les moments de réconfort dans mes derniers mois de thèse!
Merci aussi à Josy, notre ancienne assistante pour sa convivialité (et l'organisation des petits déjeuners lors de ses visites paloises :)) et son aide, et à Nicolas, notre assistant actuel pour son aide et sa disponibilité.
Merci aux "anciens" Mila, Jéjé et Angel avec qui j'ai partagé de nombreuses pauses, bons moments et grandes discussions (philosophiques ou non ;)) :) ! Merci aussi aux deux Rouennais Théo et Wilfredo pour leurs nombreux conseils (mathématiques ou non)!
Merci à Vanessa, collègue et amie, avec qui j'ai partagé de très nombreuses pauses et de très bons moments! Merci pour tes très nombreux conseils! Pour ta grande écoute et ton grand soutien!

Un grand merci aussi à mes copines "toulousaines": Noémie, Perrine, Lolo, Marion, Clém (merci particulier pour la super visite de Boston :)) et Chacha, pour leur grand soutien, leur écoute et leur compréhension!!

Un grand merci aussi à tous mes copains et mon équipe de hand (sans oublier "les vieilles")! Tous les bons moments passés avec vous m'ont permis de m'"évader" de ma thèse et de prendre du recul sur mon travail. Un grand merci à mes amies: Marjo, Lolo, Marie, Chacha, Blondie, Dadou, Mina et

Emilie qui ont toujours été là, dans les bons et les mauvais moments, qui m'ont écouté et conseillé, qui ont toujours cru en moi et qui me poussent à donner le meilleur de moi-même!

Enfin je remercie énormément ma famille, toujours présente quand il faut! Je suis très fière et heureuse de vous avoir ! Mais surtout très chanceuse d'avoir une famille comme la mienne :) !

Mes derniers mots reviennent à mes parents, mon frère et ma soeur. Le meilleur et le plus grand soutien que j'ai pu avoir! Un énorme merci pour tout!! Merci de me supporter depuis tout ce temps :), de toujours croire en moi, de me pousser à donner le meilleur de moi, à ne pas lâcher et surtout merci de me soutenir dans tout ce que je peux entreprendre!! Une pensée (très émue) pour mon papa, parti trop tôt, qui n'a pu voir cette thèse aboutie mais qui aurait été très heureux et fier.

Contents

List of Figures

List of Tables

General Introduction

The numerical solution of wave equations (acoustics, elastodynamics or electromagnetics) is increasingly used in many areas such as civil engineering, mechanical engineering, aerospace engineering, geophysics, medicine, biology or telecommunications. Applications are then various: noise reduction, radar and antenna design, satellites and waveguides, detection of hidden targets, medical imaging, seismic imaging, earthquakes dynamics, etc.

The scientific context of this thesis is seismic imaging, which aims at recovering the structure of the earth. A seismic imaging method using a wave equation model is called a *seismic-reflection technique*. This kind of method is very popular in the petroleum industry because it is the one that yields the most accurate images of the subsurface in view of analyzing them to find hydrocarbons. We refer to [1] for a detailed description of seismic imaging methods.

The principle of a seismic acquisition, as we can see on fig. 3, is simple: sources, generally placed on the top of the subsurface, emit waves inside the earth; then receivers, which are usually placed at the top or in the depth, record the reflected waves (arrival time and amplitude). The amplitudes of reflected fields allow to recover the material characteristics constituting the ground, and the arrival times allow to establish the position of the reflectors, which are the interfaces between two different media.

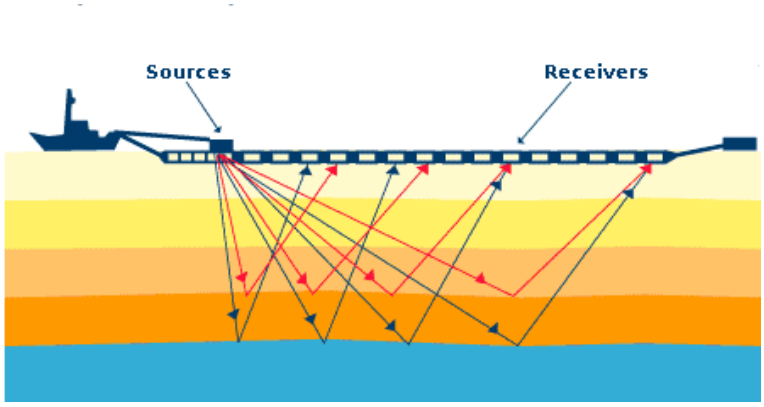


Fig. 1: Principle of a seismic acquisition.

A seismic acquisition procedure consists in two main steps: the *stacking* and the *migration*. The stacking is the sum of the different signals recorded at the receivers and allows to perform seismic tomogrography i.e. to recover wave velocities. The stacking contains the position of the reflectors in time: it corresponds to the arrival time of the reflected waves. The migration consists in turning this information in time into an information in space in order to locate the reflectors in the subsurface. These two main steps are done one after the other, the order in which they are performed depends on the type of seismic image which is looked for. For simple velocities and structures (post-stack time migration (TM)) or for complex velocities and simple structures (post-stack depth migration (DM)), the stacking is done before the migration. For simple velocities and complex structures (pre-stack TM), or for complex velocities and complex structures (pre-stack DM), it is the migration which is

done before the stacking.

Seismic imaging campaigns can be done in the sea (fig. 4a) or on the ground (fig. 4b) and with its results, one can construct a map of the variations of the velocity in the medium. The latter is referred to as the velocity model. Its quality depends on the number of sources used. It is the reason why in seismic imaging the number of sources is usually large (about 10000). Consequently, the efficiency of the whole procedure is directly related to the efficiency of the numerical method used to solve wave equations.

(a)

(b)

Fig. 2: Seismic imaging campaigns: in the sea 4a and on the ground 4b.

There exist several seismic imaging methods: the Reverse Time Migration (RTM) and the Full Wave Inversion (FWI) are the most widely used. Both methods are based on the resolution of wave equations.

The RTM, which is a depth migration technique, is based on the reversibility of wave equations. From the sources, one propagates waves to obtain the propagated field. Then one makes use of the recorded data of the reflected waves as initial data, and propagates them in order to obtain the back-propagated field. Finally, the image of the subsurface is obtained by correlating the propagated fields and back-propagated fields: at each point where there is a correlation, a reflector is deduced. For more details on the RTM technique, we refer to [2] or [3].

The FWI, which is reviewed in [4, 5], aims at solving the tomography problem and migration problem simultaneously. It is an iterative procedure solving $2N$ harmonic wave equations at each iteration of the algorithm if N sources are used. It defines an inversion process (see [6] for more details about inversion process), and thus it is composed of two major steps: first, one solves the forward problem (i.e. the propagation of the wave) which models the phenomenon for each source; then, one computes residuals due to a comparison between the recorded data and the numerical data using, for instance, a least-squares method. These residuals are used to update the velocity and the process is repeated until there is accordance between recorded data and numerical data.

Seismic imaging can be performed in the time-domain, as it is presented in [2] for the RTM, or in the frequency-domain; we refer to [7] or [8] for examples of RTM in frequency-domain and to [9] for the FWI. Compared to frequency-domain case, time-domain approaches do not require an important computational cost, but the implementation of the imaging condition is more involved. Furthermore, in frequency-domain approaches, it is not necessary to store the solution at each time step of the forward simulation. This is interesting because seismic imaging involves very large problems with a lot of data and a particular attention should be taken to memory consumption. In the frequency-domain case, the main drawback then lies in the need to solve a large linear system of equations which represents a challenging task when considering realistic 3D elastic media, despite the recent advances on high

performance numerical linear algebra solvers [10]. The use of domain decomposition techniques or Schwarz methods, such as the ones proposed in [11, 12, 13, 14, 15] for Helmholtz and Maxwell's equations can greatly reduce the computational costs and the memory consumption especially when it is coupled to an appropriate preconditionner [16, 17]. This method is very performant for solving a single problem associated to one source. However, for seismic problem where one generally has to invert a linear system with more than 1000 right hand sides (each of them being associated to a source), achieving highly efficient solution strategy can be quite challenging with iterative or hybrid iterative direct methods.

Today, the main challenges in seismic imaging are the speed up of seismic modelling; the 3D data handling, the size of realistic 3D geophysical problems being close to terabytes; and the construction of velocity models.

In this thesis, we are interested in the modeling part, i.e. the resolution of the forward problem of the FWI, assuming a time-harmonic regime, leading to the so-called Helmholtz equation.

During the past decades, the geophysical community focused on the acoustic wave equation since FWI or RTM for elastodynamics was out of reach of the supercomputers of that time. Now, with the progress of high performance computing, it is possible to consider 2D elastodynamics but it is still complicated to deal with realistic 3D elastic models. The main objective of our work is to propose and develop a new finite element (FE) type solver characterized by a reduced-size discrete operator (as compared to existing FE solvers) without hampering the accuracy of the numerical solution.

A wide variety of approximation methods of wave equations is currently available. The FWI is performed with methods belonging to the family of finite difference (FD) methods or finite element (FE) methods. FD methods use regular grids, which make them easy to implement, and allow to obtain easier systems to solve (as compared to FE methods). We refer to [18] for examples of FD methods in the geophysics frameworks and to [19] for 3D applications. Their main disadvantage is their lack of sufficient accuracy when dealing with highly heterogeneous media or when the underlying mesh is too coarse. Moreover with FD methods, one does not take accurately into account the irregular topography of the subsurface because the structured mesh cannot correctly approach interfaces [20, 21]. Indeed, with Cartesian meshes, it is difficult to handle a steep subsurface.

By contrast, with FE methods [22, 23], one can use unstructured meshes to discretize accurately complex domains. One inherits from a greater flexibility in the construction of the mesh. However, FE methods require more memory space than FD methods. Two kinds of FE method seem to be adapted to wave propagation simulation: spectral elements (SE) methods (see [24] or [25] for example of SE methods for geophysical applications) and discontinuous Galerkin (DG) methods (see [26] for a comparison between DG methods and FD methods applied to wave equations for seismic applications).

SE methods use high order functions and need less memory space than classical FE methods without hampering the numerical convergence order. However, SE methods are formulated on quadrangular (2D case)/hexahedral (3D case) meshes which may not be ideally adapted to the discretization of complex geometries in the 3D case. DG methods and FE methods mainly differ on basis functions: DG basis functions are only piecewise continuous. Moreover, in addition to the fact that they are formulated on unstructured triangular meshes, they are more suitable than classical continuous FE methods to deal with *hp*-adaptivity (interpolation degree p or mesh step h can change from element to another) [27], providing a greater flexibility in the mesh construction and the discretization of the different regions of the subsurface. For time-domain problems, DG methods provide explicit representation of the solution because the mass matrix is at worse block diagonal. In addition, they are nicely adapted to high performance computing.

Their main drawback is that they have an higher computational cost (CPU time and memory) as compared to classical FE methods because they are expensive in terms of the number of unknowns, especially when arbitrarily high order interpolation of the field components is used. This is due to the fact that the degrees of freedom belong to only one element (because basis functions are discontinuous at the interfaces of the elements) and so, the degrees of freedom placed at the interfaces have to be duplicated. In this case DG methods lead to larger sparse linear systems with a higher number of globally coupled degrees of freedom as compared to classical FE methods on a given mesh.

To overcome this drawback, we consider in this thesis a new DG method, the hybridizable DG method (HDG) introduced by Cockburn, Peraire and Lazarov in 2009 for second order elliptic problem [28]. The principle of this HDG method consists in introducing a Lagrange multiplier representing the trace of the numerical solution on each face of the mesh cells. This new variable exists only on the faces of the mesh and the unknowns of the problem depend on it. This allows to reduce the number of degrees of freedom for the discretization and thus, the number of unknowns of the global linear system. Now the size of the matrix to be inverted only depends on the number of degrees of freedom of each face and the number of faces of the mesh. It is worth noting that for a nodal DG method, this matrix size depends on the number of degrees of freedom of each element and on the number of elements of the mesh. Finally the solution of the initial problem is recovered thanks to a simple linear independent elementwise calculation.
Moreover, as it is a DG method, the HDG method benefits of the advantages of them i.e. *hp*-adaptivity and an easily parallelizable implementation.

The HDG method has been adapted to solve Maxwell's equations in frequency-domain [29, 30] and fluid mechanics models [31, 32, 33, 34]. In [35], the method has been applied to time-implicit elastodynamics, but no work has ever addressed the extension of the HDG method to elastodynamics in harmonic domain.

The main contributions of this thesis are: the development of an HDG method for elastodynamics in harmonic domain, for isotropic and anisotropic media in two and three dimensions; the performance analysis of this new method; and the comparisons of its performances with the ones of classical DG methods. This thesis is divided into five parts: in the first part, we remind how the elastic wave equations are obtained and we review the different numerical methods which allow to solve them. The second chapter is dedicated to two nodal DG methods: the centered fluxes DG method and the upwind fluxes DG method. We explain these two methods and compare them on a simple test case in order to determine a reference method for the comparison with the HDG method.
We describe the proposed 2D HDG formulation in chapter three and we detail the implementation of the algorithm. We analyze the numerical convergence of the method and the condition number of the resulting matrix to be inverted. We also study the influence of the penalization parameter on the accuracy of the solution and we discuss the *p*-adaptivity property of the HDG formulation.
In chapter four, we compare the performance of the HDG method to two classical DG methods: the upwind fluxes DG method that we present in chapter two and the interior penalty discontinuous Galerkin method. We test the three methods on simple test cases: the propagation of a plane wave and the scattering by disk-shaped obstacles. Then, we compare the performances of HDG and IPDG methods on a more realistic test case which is a classical benchmark in geophysics: the Marmousi test case. Finally, we apply the HDG method to anisotropic media.
In the last chapter, we present the 3D HDG formulation for the general anisotropic case and we detail the implementation of the algorithm. We present preliminary results in order to validate the developed numerical code.

Introduction Générale

La résolution numérique de l'équation des ondes (acoustiques, élasto-dynamiques ou électromagnétiques) est de plus en plus utilisée dans de nombreux domaines tels que le génie civil, le génie mécanique, l'aérospatiale, la géophysique, la médecine, la biologie ou encore les télécommunications. Ses applications sont alors très variées: réduction du bruit, application aux radars et antennes, satellites et guides d'ondes, détection de cibles cachées, imagerie médical, imagerie sismique, séismes, etc.

Cette thèse s'inscrit dans le cadre de l'imagerie sismique, dont le but consiste à reconstituer la structure de la Terre. Une méthode d'imagerie sismique utilisant le modèle des équations d'ondes est la technique de sismique-réflexion. Ce type de méthode est très populaire dans l'industrie pétrolière parce qu'elle permet d'obtenir des images très précises du sous-sol qui seront analyser afin de trouver des hydrocarbures. Nous faisons référence à [1] pour une description détaillée des méthodes d'imagerie sismique.

Le principe d'acquisition sismique est simple, comme on peut le voir sur la fig. 3: des sources, généralement placées à la surface du sous-sol, émettent des ondes à l'intérieur de la Terre; des récepteurs, placés ou en surface ou en plus en profondeur, enregistrent ensuite les ondes réfléchies (temps d'arrivée et amplitude). Ce sont les amplitudes des champs réfléchis qui permettent de retrouver les caractéristiques des matériaux qui constituent le sous-sol, et les temps d'arrivée permettent d'établir la position des réflecteurs, qui sont les interfaces entre deux milieux différents.

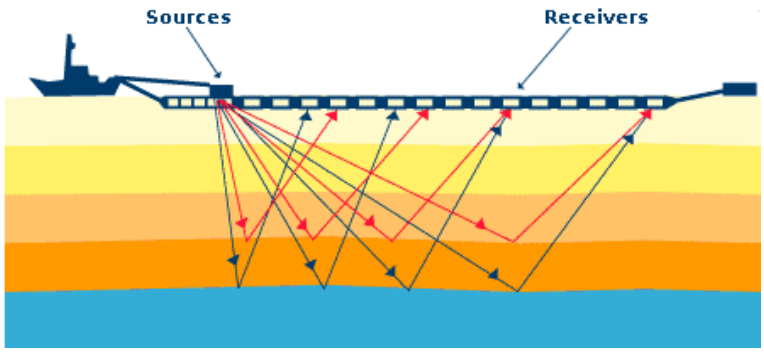


Fig. 3: Principle of a seismic acquisition.

Une procédure d'acquisition sismique est constituée de deux étapes principales: le *stacking* et la *migration*. Le stacking est la somme des différents signaux enregistrés par les récepteurs et permet d'effectuer une tomographie sismique, c'est-à-dire de retrouver les vitesses des ondes. Le stacking contient également les positions des récepteurs en temps: cela correspond au temps d'arrivée des ondes réfléchies. La migration quant à elle consiste à transformer cette information en temps en une information spatiale dans le but de localiser la position des réflecteurs dans le sous-sol. Ces deux principales étapes sont faites l'une après l'autre, et l'ordre dans lequel elles sont effectuées dépend du type d'image sismique que l'on souhaite avoir. Pour des vitesses et des structures simples (post-stack time migration (TM)) ou pour des vitesses complexes et des structures simples (post-stack

depth migration (DM)), le stacking est effectué avant la migration. Pour des vitesses simples et des structures complexes (pre-stack TM), ou pour des vitesses et des structures complexes (pre-stack DM), la migration est faite avant le stacking.

Les campagnes d'imagerie sismique peuvent être faites dans la mer (fig. 4a) ou dans le sous-sol (fig. 4b). Avec leurs résultats, on est capable de construire une carte représentant les variations de vitesse dans le milieu. Cette dernière est appelée le modèle de vitesses et sa qualité dépend du nombre de sources utilisées. C'est la raison pour laquelle en imagerie sismique le nombre de sources est généralement très grand (autour de 1000). Par conséquent, l'efficacité de toute la procédure d'imagerie est directement reliée à l'efficacité de la méthode numérique utilisée pour résoudre l'équations des ondes.

(a)

(b)

Fig. 4: Seismic imaging campaigns: in the sea 4a and on the ground 4b.

Il existe plusieurs méthodes d'imagerie sismique: la Reverse Time Migration (RTM) et la Full Wave Inversion (FWI) sont les deux méthodes les plus largement utilisées. Elles sont toutes les deux basées sur la résolution d'équations d'ondes.

La RTM, qui est une technique de migration profondeur, est basée sur la réversibilité des équations d'ondes. A partir des sources, on émet des ondes afin d'obtenir un champ propagé. Ensuite, on utilise les données enregistrées des ondes réfléchies comme données initiales et on propage dans le but d'obtenir un champ rétro-propagé. Finalement, l'image du sous-sol est obtenue en corrélant les champs propagé et rétro-propagé: à chaque point où il y a corrélation, on en déduit qu'il y a un réflecteur. Pour plus de détails sur la RTM, nous faisons référence à [2] ou [3].

La FWI, qui est expliquée dans [4, 5], a pour but de résoudre en même temps un problème de tomographie et un problème de migration. C'est une procédure itérative qui requiert de résoudre $2N$ équations d'ondes harmoniques à chaque itération de l'algorithme si N sources sont utilisées. La FWI définit un problème inverse (voir [6] pour plus de détails sur les problèmes inverses), et est alors composée de deux étapes principales: d'abord, la résolution du problème direct (c'est-à-dire la résolution de la propagation d'ondes) qui modélise le phénomène pour chaque source; puis, le calcul de résidus en comparant les données enregistrées et les données calculées numériquement. On peut par exemple utiliser la méthode des moindres carrés pour effectuer cette comparaison. Une fois les résidus calculés, ils sont ensuite utilisés pour mettre à jour le modèle de vitesses et le processus est ainsi répété jusqu'à ce qu'il y ait concordance entre données numériques et données observées.

L'imagerie sismique peut être effectuée en domaine temporel, comme présenté dans [2] pour la RTM, ou en domaine fréquentiel; nous faisons référence à [7] ou [8] pour des exemples de RTM en domaine fréquentiel et à [9] pour la FWI. Les approches en domaine temporel ne nécessitent pas d'importants coûts de calcul, mais l'implémentation de la condition d'imagerie est plus compliquée

qu'en domaine fréquentiel. D'autre part, en domaine fréquentiel, il n'est pas nécessaire de stocker la solution à chaque pas de temps de la simulation directe. Ceci est un point intéressant car les problèmes d'imagerie sismique sont de grands problèmes impliquant un grand nombre de données et une attention particulière doit être apportée à la consommation mémoire. En domaine fréquentiel, le principal inconvénient est la résolution de grands systèmes linéaires. Cela représente un véritable challenge si l'on veut considérer des cas réels 3D dans des milieux élastiques, et ce malgré les récents progrès sur les solveurs numériques linéaires haute performance [10]. L'utilisation des techniques de décomposition de domaines ou des méthodes de Schwarz, telles que celles proposées dans [11, 12, 13, 14, 15] pour les équations d'Helmholtz et de Maxwell peuvent grandement réduire les coûts de calcul et la consommation mémoire, particulièrement si elles sont couplées avec un préconditionneur approprié [16, 17]. Ces méthodes sont très performantes pour résoudre un problème associé à une seule source. Cependant, pour des problèmes sismiques où on doit généralement inverser un système linéaire pour plus de 1000 second membres (chacun d'eux étant associé à une source), obtenir une méthode de résolution assez efficace peut être assez difficile avec des méthodes itératives ou méthodes hybrides (combinant des méthodes itératives et directes).

Aujourd'hui, les principaux challenges en imagerie sismique sont l'amélioration de la modélisation sismique; le traitement des données 3D, la taille des problèmes réalistes géophysiques 3D avoisinant les térabytes; et la construction des modèles de vitesse.

Dans cette thèse, nous nous intéressons à la partie modélisation, c'est-à-dire à la résolution du problème direct de la FWI, en supposant un régime harmonique; ce qui nous amène à résoudre l'équation d'Helmholtz.

Au cours des dernières décennies, la communité géophysique s'est focalisée sur l'équation des ondes acoustiques étant donnée qu'effectuer une RTM ou une FWI pour un problème élastodynamique était hors de portée des moyens informatiques de l'époque. Aujourd'hui, grâce aux progrès en calcul haute performance, il est possible de considérer les problèmes élastodynamiques 2D mais considérer des modèles élastiques réalistes 3D reste toujours compliqué. L'objectif principal de ce travail est de proposer et de développer un nouveau type de solveur élément fini (FE) caracterisé par un opérateur discret de taille réduite (comparé aux solveurs FE déjà existants) sans modifier la précision de la solution numérique.

Une large variété de méthodes d'approximation des équations d'ondes sont aujourd'hui utilisées. La FWI est utilisée par exemple avec des méthodes appartenant aux familles des méthodes de différences finies (FD) ou d'éléments finis (FE). Les méthodes FD s'utilisent avec des grilles régulières, ce qui les rend simples à implémenter. Elles permettent d'obtenir des systèmes faciles à résoudre (comparé aux méthodes FE). Nous pouvons citer [18] comme exemple de méthodes FD dans un contexte géophysique et [19] comme exemple d'applications 3D. Leur principal inconvénient est qu'elles manquent de précision lorsque l'on souhaite traiter des problèmes hautement héterogènes ou quand le maillage utilisé est trop grossier. D'autre part, elles ne prennent pas bien en compte la topographie irrégulière du sous-sol terrestre, étant donné que les maillages structurés ne permettent pas d'approcher correctement les interfaces [20, 21]. En effet, avec un maillage cartésien, il est difficile de traiter un terrain escarpé.

A l'inverse, avec les méthodes FE [22, 23], on peut utiliser des maillages non-structurés pour discrétiser plus précisement les domaines complexes. On hérite alors d'une grande flexibilité en matière de construction du maillage. Cependant, les méthodes FE requièrent plus d'espace mémoire que les méthodes FD. Deux classes de méthodes FE semblent être les mieux adaptées à la simulation de la propagation d'ondes: les méthodes d'éléments spectraux (SE) (voir [24] ou [25] pour des exemples d'applications géophysiques des méthodes SE) et les méthodes de Galerkine discontinues (voir [26]

pour une comparaison entre les méthodes DG et les méthodes FD appliquées aux équations d'ondes dans un cadre sismique).

Les méthodes SE peuvent s'utiliser avec des fonctions d'ordre élevé et nécessitent moins d'espace mémoire que les méthodes FE classiques tout en gardant le même ordre de convergence numérique. Cependant, les méthodes SE sont formulées sur maillages composés de quadrangles en 2D ou d'hexaèdres en 3D ce qui n'est pas idéalement adapté à la discrétisation de géométrie complexe, surtout dans le cas 3D. Les méthodes DG diffèrent principalement des méthodes FE par leurs fonctions de base: les fonctions de base DG sont seulement continues par morceaux. Ajouté au fait qu'elles sont formulées sur des maillages triangulaires non-structurés, elles sont alors plus adaptées que les méthodes FE continues classiques pour gérer l'*hp*-adaptivité (degré d'interpolation p ou pas de maillage h pouvant changer d'un élément à l'autre) [27], fournissant un important degré de flexibilité dans la construction du maillage et dans la discrétisation des différentes régions du sous-sol. Pour les problèmes temporels, les méthodes DG fournissent une représentation explicite de la solution, car la matrice de masse est au pire diagonale par blocs. D'autre part, ces méthodes s'adaptent très bien au calcul haute performance.

Leur principal inconvénient est l'important coût de calcul (temps CPU et mémoire) qu'elles génèrent comparé aux méthodes FE classiques. Ceci est dû à un grand nombre d'inconnues, particulièrement quand on utilise un ordre d'interpolation arbitrairement élevé pour les composantes du champs. Cela s'explique par le fait que comme les degrés de liberté appartiennent à un seul élément (à cause des fonctions de bases discontinues aux interfaces des éléments), ceux placés aux interfaces doivent être dupliqués. Par conséquent, les méthodes DG conduisent à des systèmes linéaires creux très grands avec un grand nombre de degrés de liberté couplés comparé aux méthodes FE classiques sur un même maillage.

Pour pallier cet inconvénient, on considère dans cette thèse une nouvelle méthode DG, la méthode DG hybride (HDG) introduite par Cockburn, Peraire et N'Guyen en 2009 pour les problèmes elliptiques du second ordre [28]. Le principe de la méthode HDG consiste à introduire un multiplicateur de Lagrange représentant la trace de la solution numérique sur chaque face des cellules du maillage. Cette nouvelle variable existe alors seulement sur les faces du maillage et les inconnues du problème dépendront d'elle. Ceci permet de réduire le nombre de degrés de liberté pour la discrétisation et donc de réduire le nombre d'inconnues du système linéaire global. La taille de la matrice à inverser dépend alors seulement du nombre de degrés de liberté sur chaque face et du nombre de faces du maillage. Il est intéressant de noter que dans le cas d'une méthode DG nodale, la taille de cette matrice dépend du nombre de degrés de liberté de chaque élément et du nombre d'éléments du maillage. La solution au problème initial est finalement obtenue par de simples calculs linéaires indépendants.
D'autre part, comme c'est une méthode DG, la méthode HDG bénéficie des mêmes avantages: adaptivité *hp* et une implémentation simple pour le calcul parallèle.

La méthode HDG a été adaptée à la résolution des équations de Maxwell en domaine fréquentiel [29, 30] et aux modèles de mécanique des fluides [31, 32, 33, 34]. Dans [35], elle a été appliquée à l'élastodynamique en temps implicite, mais aucune étude n'a été effectuée sur l'extension de la méthode HDG à l'élastodynamique en domaine harmonique.

Les principales contributions de cette thèse sont: le développement d'une méthode HDG pour l'élastodynamique en domaine fréquentiel, pour des milieux isotropes et anisotropes en deux et trois dimensions; l'analyse de performance de cette méthode; et des comparaisons de ces performances à celles obtenues avec des méthodes DG classiques. Cette thèse est divisée en cinq parties: dans la première partie, nous rappelons comment sont obtenues les équations des ondes élastiques et nous revoyons les différents types de méthodes numériques qui permettent de les résoudre. Le second chapitre se consacre à deux méthodes DG nodales: la méthode DG à flux centrés et la méthode DG à flux décentrés. Nous expliquons ces deux méthodes et nous les comparons grâce à un cas test simple

dans le but d'obtenir une méthode de référence pour la comparaison avec la méthode HDG.
Nous décrivons la formulation HDG 2D que nous proposons dans le chapitre 3 et nous détaillons l'implémentation de son algorithme. Nous analysons sa convergence numérique et le conditionnement de la matrice à inverser. Nous étudions aussi l'influence du paramètre de pénalisation sur la précision de la solution et nous discutons sur la propriété de p-adaptivité de la formulation HDG.
Dans le chapitre quatre, nous comparons les performances de la méthode HDG à deux méthodes DG classiques: la méthode DG à flux décentrés que nous avons présentée dans le chapitre 2 et la méthode DG avec pénalité intérieure. Nous testons ces trois méthodes sur des cas tests simples: la propagation d'une onde plane dans le vide et la diffraction par des obstacles de forme circulaire. Ensuite, nous comparons les performances des méthodes HDG et IPDG sur un cas plus réaliste qui est un exemple classique géophysique: le modèle Marmousi. Enfin, nous appliquons la méthode HDG à des milieux anisotropes.
Dans le dernier chapitre, nous présentons la formulation HDG 3D dans le cas général anisotrope et nous détaillons la mise en oeuvre de son algorithme. Nous présentons des résultats préliminaires dans le but de valider le code développé.

Cette thèse a donné lieu aux communications suivantes:

Publications

- **Hybridizable Discontinuous Galerkin method for solving the 2D Helmholtz equations for geophysical applications.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri. *en préparation*

Conférences

- **Discontinuous Galerkin methods for solving Helmholtz isotropic waves equations for seismic applications.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri. *Workshop HOSCAR, Bordeaux (33), sept. 2013*
- **Discontinuous Galerkin methods for solving Helmholtz elastic waves equations for seismic imaging.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri. *WCCM XI-ECCM V-ECFD VI, Barcelone, Espagne, Juil. 2014*
- **Hybridizable Discontinuous Galerkin method for solving Helmholtz elastic wave equations.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri. *Workshop EAGE Chania, Crête, sept. 2014*
- **Performance analysis of DG and HDG methods for the simulation of seismic wave propagation in harmonic domain.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri. *Second French-Russian Workshop, Novossibirsk, Russie, sept. 2014*
- **Numerical schemes for the simulation of seismic wave propagation in frequency domain.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri. *Réunion des Sciences de la Terre, Pau, oct. 2014*
- **Performance comparison between hybridizable DG and classical DG methods for elastic waves simulation in harmonic domain.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri. *Workshop, Rice, Etats-Unis, mars 2015*
- **Modelling of seismic waves propagation in harmonic domain by hybridizable discontinuous Galerkin method.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri. *Workshop, Pau, mai 2015*

- **Modelling of elastic Helmholtz equations by hybridizable discontinuous Galerkin method for geophysical applications.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri. *Workshop DIP, Pau, Juin 2015*

- **Modeling of elastic Helmholtz equations by hybridizable discontinuous Galerkin method (HDG) for geophysical applications.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri. *Workshop HOSCAR, Nice, Septembre 2015*

- **Modeling of elastic Helmholtz equations by hybridizable discontinuous Galerkin method (HDG) for geophysical applications** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri. *Journées MATHIAS, Paris, Octobre 2015*

Posters

- **Discontinuous Galerkin methods for solving Helmholtz isotropic waves equations.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri. *Journées Ondes du Sud-Ouest, Toulouse, fev. 2014*

- **Performance comparison of HDG and classical DG method for the simulation of seismic wave propagation in harmonic domain.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri. *Journées MATHIAS, Paris, oct. 2014*

Rapport de recherches

- **Discontinuous Galerkin methods for the simulation of the propagation of the elastic waves equation in the frequency domain.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri, 2015

- **Hybridizable Discontinuous Galerkin method for the simulation of the propagation of the elastic wave equations in the frequency domain.** M. Bonnasse-Gahot, H. Calandra, J.Diaz, S.Lanteri, 2015

Abstract

Cette thèse s'inscrit dans le cadre de l'action stratégique **DIP** (Depth Imaging Partnership), partenariat entre l'entreprise Total et l'institut de recherches Inria. Le projet DIP s'intéresse à l'imagerie sismique, qui consiste à reconstituer la structure du sous-sol terrestre. Pour cela, des sources placées au dessus des zones que l'on souhaite explorer émettent des ondes qui se propagent à l'intérieur du sous-sol. Des récepteurs enregistrent ensuite les ondes réfléchies (temps d'arrivée et amplitude). C'est à partir des données enregistrées aux récepteurs que l'on peut reconstruire une image du sous-sol.
L'imagerie sismique peut s'effectuer en domaine temporel ou en domaine fréquentiel. L'implémentation de la condition d'imagerie en domaine temporel est plus complexe qu'en domaine fréquentiel, mais effectuer des simulations réalistes en domaine fréquentiel s'avère presque impossible, car les coûts de calcul sont beaucoup trop importants. En effet, d'une part, les problèmes d'imagerie sismique sont généralement des problèmes avec un grand nombre de données (le nombre de sources utilisées et de récepteurs étant très grand). D'autre part, en domaine fréquentiel, on doit résoudre de grands systèmes linéaires d'équations ce qui représente un véritable défi si l'on veut considérer des cas réels 3D dans des milieux élastiques, et ce malgré les progrès sur les solveurs linéaires haute performance.

Dans cette thèse, nous nous intéressons à la résolution de l'équation des ondes en domaine fréquentiel, le but étant d'essayer de réduire la taille du système linéaire à résoudre.

Les équations des ondes élastiques en domaine fréquentiel, encore appelées équations d'Helmholtz, s'écrivent:

$$\text{Pour } \mathbf{x} = (x,\ y,\ z)^T \in \Omega \subset \mathbb{R}^3, \qquad \left\{ \begin{array}{lcll} i\omega\rho(\mathbf{x})\mathbf{v}(\mathbf{x}) & = & \nabla \cdot \underline{\sigma}(\mathbf{x}) + \mathbf{F}_{\mathbf{v}}(\mathbf{x}) & \text{dans } \Omega \\ i\omega\underline{\sigma}(\mathbf{x}) & = & \underline{\underline{C}}(\mathbf{x}) : \underline{\epsilon}(\mathbf{v}(\mathbf{x})) & \text{dans } \Omega \end{array} \right. \tag{0.0.1}$$

où :

$\rho(\mathbf{x})$ densité ;

$\mathbf{v}(\mathbf{x}) = (\ v_x(\mathbf{x}),\ \ v_y(\mathbf{x}),\ \ v_z(\mathbf{x})\ )^T$, est le vecteur vitesse ;

$\underline{\underline{\epsilon}}$ est le tenseur des déformations avec $\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial j} + \frac{\partial v_j}{\partial i}\right), \qquad i,j = x,z;$

$\underline{\underline{\sigma}}$ est le tenseur des contraintes avec, dans le cas isotrope, $\sigma_{ij} = \lambda\delta_{ij}tr(\underline{\underline{\epsilon}}) + 2\mu\epsilon_{ij}, \qquad i,j = x,z;$

$\underline{\underline{C}}$ est le tenseur d'ordre 4 symétrique des coefficients élastiques. On se ramène à une matrice 6×6 en utilisant la notation de Voigt.

Dans le cas anisotrope, on a $\underline{\underline{C}}(\mathbf{x}) = \begin{pmatrix} C_{11}(\mathbf{x}) & C_{12}(\mathbf{x}) & C_{13}(\mathbf{x}) & C_{14}(\mathbf{x}) & C_{15}(\mathbf{x}) & C_{16}(\mathbf{x}) \\ C_{12}(\mathbf{x}) & C_{22}(\mathbf{x}) & C_{23}(\mathbf{x}) & C_{24}(\mathbf{x}) & C_{25}(\mathbf{x}) & C_{26}(\mathbf{x}) \\ C_{13}(\mathbf{x}) & C_{23}(\mathbf{x}) & C_{33}(\mathbf{x}) & C_{34}(\mathbf{x}) & C_{35}(\mathbf{x}) & C_{36}(\mathbf{x}) \\ C_{14}(\mathbf{x}) & C_{24}(\mathbf{x}) & C_{34}(\mathbf{x}) & C_{44}(\mathbf{x}) & C_{45}(\mathbf{x}) & C_{46}(\mathbf{x}) \\ C_{15}(\mathbf{x}) & C_{25}(\mathbf{x}) & C_{35}(\mathbf{x}) & C_{45}(\mathbf{x}) & C_{55}(\mathbf{x}) & C_{56}(\mathbf{x}) \\ C_{16}(\mathbf{x}) & C_{26}(\mathbf{x}) & C_{36}(\mathbf{x}) & C_{46}(\mathbf{x}) & C_{56}(\mathbf{x}) & C_{66}(\mathbf{x}) \end{pmatrix}.$

Dans le cas isotrope, $\underline{\underline{C}}(\mathbf{x}) = \begin{pmatrix} \lambda(\mathbf{x})+2\mu(\mathbf{x}) & \lambda(\mathbf{x}) & \lambda(\mathbf{x}) & 0 & 0 & 0 \\ \lambda(\mathbf{x}) & \lambda(\mathbf{x})+2\mu(\mathbf{x}) & \lambda(\mathbf{x}) & 0 & 0 & 0 \\ \lambda(\mathbf{x}) & \lambda(\mathbf{x}) & \lambda(\mathbf{x})+2\mu(\mathbf{x}) & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu(\mathbf{x}) & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu(\mathbf{x}) & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu(\mathbf{x}) \end{pmatrix}.$

λ et μ sont les coefficients de Lamé du milieu;

$\mathbf{F_v}(\mathbf{x})$ représente les forces de volume.

Dans la suite, on n'écrira plus les dépendances en espace des paramètres physiques ρ, λ et μ, des tenseurs $\underline{\underline{C}}, \underline{\underline{\sigma}}$ et $\underline{\underline{\epsilon}}$ et du vecteur $\mathbf{v}$.

Les conditions aux limites utilisées pour assurer l'existence et l'unicité du schéma sont:

- La condition de Dirichlet pour exprimer la surface libre

$$\underline{\underline{\sigma}} \cdot \mathbf{n} = 0 \ \text{ sur } \Gamma_l, \tag{0.0.2}$$

- Une condition de Sommerfeld pour représenter une condition absorbante

$$\underline{\underline{\sigma}} \cdot \mathbf{n} + PA(\theta')P^T\mathbf{v} = 0 \ \text{ sur } \Gamma_a, \tag{0.0.3}$$

avec Γ_l la surface libre et Γ_a la surface absorbante, et on a $\Gamma_l \cup \Gamma_a = \partial\Omega$ et $\Gamma_l \cap \Gamma_a = \emptyset$.

Plusieurs méthodes permettent de discrétiser ces équations:

- Les méthodes de différences finies: elles s'utilisent avec des grilles régulières ce qui facilite leur implémentation et nécessitent peu de coûts de calcul. En revanche, elles souffrent d'un manque de précision et ne s'adaptent pas bien à la topographie du sous-sol, notamment lorsqu'on se trouve en présence de milieux escarpés.

- Les méthodes d'éléments finis: leur avantage est qu'elles sont formulées sur des maillages non-structurés. Elles s'adaptent donc bien à des topographies irréguliers. D'autre part, la solution est représentée à partir d'approximations polynomiales et la méthode converge numériquement

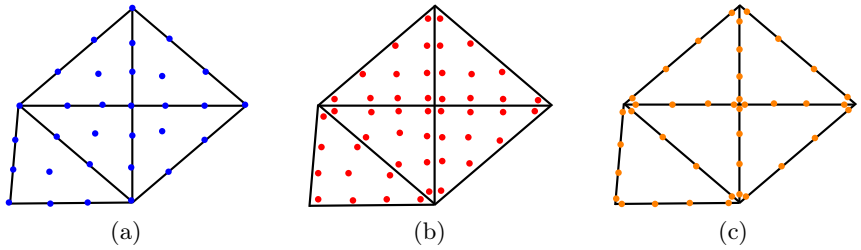
avec un ordre optimal (i.e. la méthode converge à l'ordre $p+1$ si le degré d'interpolation est p). Cependant, les méthodes d'éléments finis font intervenir une matrice de masse pleine qu'il faut inverser, ce qui rend ces méthodes plus coûteuses que les méthodes de différences finies. Par ailleurs, en domaine fréquentiel, elles souffrent de pollution numérique.

- Les méthodes d'éléments spectraux: ce sont une classe d'éléments finis et elles ont donc la même précision que ces dernières. Leur principal avantage est la diagonalisation de la matrice de masse par l'utilisation de points d'interpolation de Lagrange et de quadratures de Gauss-Lobatto-Legendre. Elles peuvent aussi s'utiliser en formulation d'ordre élevé. Malheureusement, elles sont principalement formulées sur des quadrangles en 2D et des hexaèdres en 3D. Leur formulation sur des maillages triangulaires est assez complexe. En domaine fréquentiel, elles souffrent aussi de problèmes de pollution.

- Les méthodes de Galerkine discontinues: ce sont aussi une classe d'éléments finis. En réalité, c'est une combination entre méthode d'éléments finis car la solution est approchée par des polynômes, et méthode des volumes finis car les interactions aux interfaces sont calculées à partir de flux numériques. Bénéficiant d'une matrice de masse diagonale par blocs, elles peuvent être vue comme une méthode concurrente des méthodes d'éléments spectraux. Par ailleurs, elles ont l'avantage d'être formulées sur des maillages triangulaires non-structurés. Le fait que les fonctions de base des méthodes de Galerkine discontinues soient continues sur l'élément mais discontinues à ses interfaces permet d'adapter l'ordre d'interpolation à chaque élément (p-adaptivité) rend ces méthodes bien adaptées au calcul parallèle. Cependant, ces discontinuités impliquent que les degrés de liberté appartiennent à un seul élément, et ceux se trouvant aux interfaces doivent alors être dupliqués pour n'appartenir qu'à un seul élément. Cela implique alors un très grand nombre de degrés de liberté et donc un important coût de calcul, comparé au coût de calcul des méthodes éléments finis classiques.

- La méthode de Galerkine discontinue hybride: c'est une classe de méthodes de Galerkine discontinues. Elle bénéficie donc des mêmes avantages: formulée sur des maillages triangulaires non-structurés, approximation polynomiale, hp-adaptivité, bien adaptée au calcul parallèle, fonctions de base discontinues. Leur différence avec les méthodes de Galerkine discontinues classiques vient du fait qu'on va exprimer les inconnues du problème initial en fonction d'une nouvelle variable, un multiplicateur de Lagrange Λ, vivant uniquement aux interfaces et représentant la trace de la solution numérique sur chaque face du maillage. Le multiplicateur Λ vivant uniquement sur les faces du maillage, les fonctions de base de la méthode sont alors continues sur les faces mais discontinues aux frontières. Par exemple, en 2D, les fonctions de base sont continues sur les arêtes mais discontinues aux sommets. En 3D, elles sont continues sur les faces mais discontinues sur les arêtes. De cette manière, seuls les degrés de liberté situés sur les frontières des interfaces doivent être dupliqués. Etant donné que le système linéaire à résoudre est fonction de Λ, le nombre global d'inconnues est alors moins important que pour des méthodes de Galerkine discontinues classiques. La solution au problème initial est ensuite retrouvée à partir de simples calculs indépendants.

L'objectif de cette thèse est de développer la méthode de Galerkine discontinue hybride (HDG) pour les équations élastodynamiques en domaine fréquentiel.
La première étape a été de déterminer une méthode de référence pour nous permettre d'évaluer l'efficacité de la méthode HDG. Nous nous sommes d'abord concentrés sur deux méthodes de Galerkine discontinues (DG) nodales: la méthode DG à flux centrés et la méthode DG à flux décentrés. Une étude de ces deux méthodes sur un cas test analytique, le cas test de l'onde plane se propageant dans le vide, nous a permis de montrer que la méthode DG à flux centrés est sous-optimale, i.e. elle ne



(a) (b) (c)

Fig. 0.0.5: Distribution des degrés de liberté globaux pour les méthodes éléments finis 1.2.4a, de Galerkine discontinues 1.2.4b et de Galerkine discontinue hybride 1.2.4c avec un degré d'interpolation de 3.

converge numériquement qu'à l'ordre p pour un degré d'interpolation p, et qu'elle n'est donc pas adaptée pour la comparaison avec la méthode HDG. La méthode DG à flux décentrés, bien que plus coûteuse, converge bien optimalement et c'est cette méthode que nous avons retenue comme méthode de référence.
Nous avons aussi comparé la méthode HDG à la méthode DG avec pénalité intérieure (IPDG).
Dans ce résumé, nous ne présenterons pas les méthodes DG à flux décentrés et IPDG, nous nous concentrons sur la formulation HDG pour l'équation des ondes élastiques. Nous ferons ensuiteun résumé des principaux résultats obtenus.

Méthode DG hybride

Nous allons expliquer brièvement comment obtenir la formulation HDG des équations d'Helmholtz élastiques. Avant cela, nous introduisons les notations et définitions que nous allons utiliser dans la formulation.

Notations

Soit $\mathcal{T}_h$ une triangulation du domaine de calcul Ω constitué d'éléments K (triangles en 2D, tétraèdres en 3D). Nous définisons

- $\mathcal{F}(K)$ l'ensemble des faces de l'élément K,
- F une face quelconque de l'élément K,
- $\mathcal{F}_b$ l'ensemble des faces F_b, situées sur le bord Γ du domaine de calcul Ω, c'est-à-dire $F_b = \partial K \cap \Gamma$,
- $\mathcal{F}_i$ l'ensemble des faces internes F_i c.à.d $F_i = \partial K \cap \partial K'$ où K et K' sont deux éléments voisins,
- $\mathcal{F}_h$ l'ensemble des faces du maillage $\mathcal{T}_h$, c.à.d $\mathcal{F}_h = \mathcal{F}_i \cup \mathcal{F}_b$,
- $\mathbf{n}$ est la normale sortante de l'élément K.

Soit $P_p(D)$ l'ensemble des polynômes de degré au plus p sur le domaine D; pour chaque élément $K \in \mathcal{T}_h, V^p(K)$ est l'espace $P_p(K)$, $\mathbf{V}^p(K)$ l'espace $(P_p(K))^2$ et $\mathbf{\Sigma}^p(K)$ l'espace $(P_p(K))^4$.

$$V_h^p = \{v \in L^2(\Omega) : \quad v|_K \in V^p(K), \forall K \in \mathcal{T}_h\},$$
$$\mathbf{V}_h^p = \{\mathbf{v} \in \left(L^2(\Omega)\right)^2 : \quad \mathbf{v}|_K \in \mathbf{V}^p(K), \forall K \in \mathcal{T}_h\},$$

$$\boldsymbol{\Sigma}_h^p = \{\underline{\underline{\sigma}} \in \left(L^2(\Omega)\right)^3 : \quad \underline{\underline{\sigma}}|_K \in \boldsymbol{\Sigma}^p(K), \forall K \in \mathcal{T}_h\},$$
$$\mathbf{M}_h = \{\eta \in \left(L^2(\mathcal{F}_h)\right)^2 : \quad \eta|_F \in \left(P_p(F)\right)^2, \forall F \in \mathcal{F}_h\}.$$

Finalement, nous définisons le saut $[\![\cdot]\!]$ d'un vecteur $\mathbf{v}$ tel que

$$[\![\mathbf{v}]\!] = \mathbf{v}^+ \cdot \mathbf{n}^+ + \mathbf{v}^- \cdot \mathbf{n}^-,$$

et d'un tenseur $\underline{\underline{\sigma}}$ tel que

$$[\![\underline{\underline{\sigma}}]\!] = \underline{\underline{\sigma}}^+ \cdot \mathbf{n}^+ + \underline{\underline{\sigma}}^- \cdot \mathbf{n}^-.$$

Formulation HDG

Pour établir la formulation HDG, on considère les équations d'Helmholtz élastiques sur un élément K de $\mathcal{T}_h$, en supposant les paramètres physiques ρ, λ et μ constants par élément.
On multiplie ces équations par une fonction test $(\mathbf{w}, \underline{\underline{\xi}}) \in \mathbf{V}^p(K) \times \boldsymbol{\Sigma}^p(K)$ et on intègre par parties.
On trouve alors l'approximation $(\mathbf{v}_h, \underline{\underline{\sigma}}_h) \in \mathbf{V}_h^p \times \boldsymbol{\Sigma}_h^p$ telle que

$$\left\{ \begin{array}{l} \displaystyle\int_K i\omega\rho_K \mathbf{v}_h \cdot \mathbf{w} + \int_K \underline{\underline{\sigma}}_h : \nabla \mathbf{w} - \int_{\partial K} \widehat{\underline{\underline{\sigma}}}_h \mathbf{n} \cdot \mathbf{w} = \int_K \mathbf{f} \cdot \mathbf{w}, \\ \displaystyle\int_K i\omega \underline{\underline{\sigma}}_h : \underline{\underline{\xi}} + \int_K \mathbf{v}_h \cdot \nabla \cdot \left(\underline{\underline{C}}_K \underline{\underline{\xi}}\right) - \int_{\partial K} \widehat{\mathbf{v}}_h \cdot \underline{\underline{C}}_K \underline{\underline{\xi}} \cdot \mathbf{n} = 0. \end{array} \right. \tag{0.0.4}$$

On note par $\underline{\underline{a}} : \underline{\underline{b}}$ le produit scalaire de deux tenseurs $\underline{\underline{a}}$ et $\underline{\underline{b}}$.

Les traces numériques $\widehat{\underline{\underline{\sigma}}}_h$ et $\widehat{\mathbf{v}}_h$ sont les approximations de $\underline{\underline{\sigma}}$ et $\mathbf{v}$ sur la frontière ∂K.
On pose

$$\lambda_h = \widehat{\mathbf{v}}_h, \forall F \in \mathcal{F}_h, \qquad \lambda_h \in \mathbf{M}_h. \tag{0.0.5}$$

et on définit $\widehat{\underline{\underline{\sigma}}}_h$ en fonction des autres inconnues:

$$\widehat{\underline{\underline{\sigma}}}_h = \underline{\underline{\sigma}}_h - \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right) \otimes \mathbf{n} \qquad \text{sur } \partial K. \tag{0.0.6}$$

$\mathbf{S}$ est un matrice de stabilisation locale qui a un effet important sur la précision et la stabilité du schéma et nous la définissons telle que

$$\mathbf{S} = \tau \mathbf{I}.$$

En sommant les contributions de (0.0.4) sur tous les éléments et en imposant la continuité de la composante normale de $\widehat{\underline{\underline{\sigma}}}_h$, le problème peut être reformulé de la manière suivante:
trouver $(\mathbf{v}_h, \underline{\underline{\sigma}}_h, \lambda_h) \in \mathbf{V}_h^p \times \boldsymbol{\Sigma}_h^p \times \mathbf{M}_h^0$ tel que: $\forall (\mathbf{w}, \underline{\underline{\xi}}, \eta) \in \mathbf{V}^p(K) \times \boldsymbol{\Sigma}^p(K) \times \mathbf{M}_h^0$,

$$\left\{ \begin{array}{l} \displaystyle\sum_{K\in\mathcal{T}_h} \int_K i\omega\rho_K \mathbf{v}_h \cdot \mathbf{w} + \sum_{K\in\mathcal{T}_h} \int_K \underline{\underline{\sigma}}_h : \nabla \mathbf{w} - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n} \cdot \mathbf{w} = \sum_{K\in\mathcal{T}_h} \int_K \mathbf{f} \cdot \mathbf{w}, \\ \displaystyle\sum_{K\in\mathcal{T}_h} \int_K i\omega \underline{\underline{\sigma}}_h : \underline{\underline{\xi}} + \sum_{K\in\mathcal{T}_h} \int_K \mathbf{v}_h \cdot \nabla \cdot \left(\underline{\underline{C}}_K \underline{\underline{\xi}}\right) - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \lambda_h \cdot \underline{\underline{C}}_K \underline{\underline{\xi}} \mathbf{n} = 0, \\ \displaystyle\sum_{F\in\mathcal{F}_h} \int_F [\![\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n}]\!] \cdot \eta = 0. \end{array} \right. \tag{0.0.7}$$

La dernière équation de ce système est appelée *condition de conservativité* et exprime faiblement la continuité de la composante normale de $\widehat{\underline{\underline{\sigma}}}_h$.
D'après (0.0.6), on remarque que sur ∂K

$$\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n} = \underline{\underline{\sigma}}_h \cdot \mathbf{n} - \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right), \tag{0.0.8}$$

et que

$$\sum_{F\in\mathcal{F}_h}\int_F [\![\widehat{\underline{\underline{\sigma}}}_h\cdot\mathbf{n}]\!]\cdot\eta=\sum_{K\in\mathcal{T}_h}\int_{\partial K}\left(\underline{\underline{\sigma}}_h\cdot\mathbf{n}\right)\cdot\eta-\sum_{K\in\mathcal{T}_h}\int_{\partial K}\mathbf{S}\left(\mathbf{v}_h-\lambda_h\right)\cdot\eta. \tag{0.0.9}$$

En introduisant les équations (0.0.8) et (0.0.9) dans, respectivement, la première et la dernière équation du système (0.0.7), on obtient alors le système global suivant

$$\begin{cases}\displaystyle\sum_{K\in\mathcal{T}_h}\int_K i\omega\rho_K\mathbf{v}_h\cdot\mathbf{w}-\sum_{K\in\mathcal{T}_h}\int_K\left(\nabla\cdot\underline{\underline{\sigma}}_h\right)\cdot\mathbf{w}+\sum_{K\in\mathcal{T}_h}\int_{\partial K}\mathbf{S}\left(\mathbf{v}_h-\lambda_h\right)\cdot\mathbf{w}=\sum_{K\in\mathcal{T}_h}\int_K\mathbf{f}\cdot\mathbf{w},\\ \displaystyle\sum_{K\in\mathcal{T}_h}\int_K i\omega\underline{\underline{\sigma}}_h:\underline{\underline{\xi}}+\sum_{K\in\mathcal{T}_h}\int_K\mathbf{v}_h\cdot\nabla\cdot\left(\underline{\underline{C}}_K\underline{\underline{\xi}}\right)-\sum_{K\in\mathcal{T}_h}\int_{\partial K}\lambda_h\cdot\underline{\underline{C}}_K\underline{\underline{\xi}}\cdot\mathbf{n}=0,\\ \displaystyle\sum_{K\in\mathcal{T}_h}\int_{\partial K}\underline{\underline{\sigma}}_h\mathbf{n}\cdot\eta-\sum_{K\in\mathcal{T}_h}\int_{\partial K}\mathbf{S}\left(\mathbf{v}_h-\lambda_h\right)\cdot\eta=0.\end{cases} \tag{0.0.10}$$

Le problème local sur un élément K s'écrit

$$\begin{cases}\displaystyle\int_K i\omega\rho_K\mathbf{v}^K\cdot\mathbf{w}-\int_K\left(\nabla\cdot\underline{\underline{\sigma}}^K\right)\cdot\mathbf{w}+\int_{\partial K}\mathbf{S}\left(\mathbf{v}^K-\lambda_h\right)\cdot\mathbf{w}=\int_K\mathbf{f}^K\cdot\mathbf{w},\\ \displaystyle\int_K i\omega\underline{\underline{\sigma}}^K:\underline{\underline{\xi}}+\int_K\mathbf{v}^K\cdot\nabla\cdot\left(\underline{\underline{C}}_K\underline{\underline{\xi}}\right)-\int_{\partial K}\lambda_h\cdot\underline{\underline{C}}_K\underline{\underline{\xi}}\cdot\mathbf{n}=0.\end{cases} \tag{0.0.11}$$

Principaux résultats numériques

Nous avons étudié la méthode HDG sur plusieurs cas tests analytiques 2D:

- la propagation de l'onde plane dans le vide, qui nous a permis de confirmer que la méthode HDG convergeait optimalement, de montrer l'importance du paramètre de stabilisation τ sur la précision des résultats;
- la diffraction de l'onde plane par un obstacle cylindrique et par un solide élastique.

Ces trois cas tests nous ont permis de démontrer que, pour un même maillage et un même ordre d'interpolation, la méthode HDG était moins coûteuse que les méthodes DG classiques.
Nous avons également évalué les performances de la méthode HDG sur un cas test anisotrope. Ce cas test nous a montré que la valeur optimale du paramètre de stabilisation semble être v_p^2. D'autre part, nous avons pu constater qu'exécuter l'algorithme HDG anisotrope n'impliquait pas de coût de calcul supplémentaire par rapport au cas isotrope.
Finalement, nous avons testé la méthode HDG sur un cas test géophysique très courant: le modèle Marmousi. Avec ce dernier cas test, nous avons testé l'implémentation parallèle de l'algorithme HDG. Nous avons ensuite étendu l'algorithme 2D au cas 3D. Les résultats préliminaires de comparaison HDG/DG classiques confirment qu'à maillage donné et degré d'interpolation donné, la méthode HDG est plus performante.

Chapter 1

General presentation of the modeling of the elastic waves propagation

In this chapter, we remind the expression of the elastic wave equations obtained by linearization of the equations of continuum mechanics. Then, we review the different numerical methods that have been developed to solve them.

1.1 Elastic waves equations

The propagation of a seismic wave inside the Earth is represented by the elastic waves differential equations: a seismic wave moving inside a part $\mathbf{V}$ of the subsurface is characterized by short vibrations which have a small amplitude as compared to the size of the volume $\mathbf{V}$ (the deformation after the passage of the wave is very small: around 1 mm per km). So the subsurface is supposed to have an elastic behavior; we remind that the elasticity defines in particular the capacity of a solid material to return to its original shape after having been deformed.

1.1.1 Linear elasticity

Seismic waves travelling inside the subsurface induce small deformations of the subsurface which can be described by a linear elasticity model. Linear elastic equations are established thanks to the equation of motion, obtained from equations representing the linear deformations and the conservation of linear momentum law, and taking into account the equation representing the material behaviour.

- Equation of motion
  - Linear deformations
    Considering a fixed reference frame in space and time, a small volume element $\mathbf{V}$ whose gravity center has a position $\mathbf{x_0} = (x_0, y_0, z_0)^T$ at time t_0. After the deformation, its position at time t is $\mathbf{x} = (x, y, z)^T$. Its displacement $\mathbf{u}(\mathbf{x}) = (u_x, u_y, u_z)^T$ is then defined as $\mathbf{u}(\mathbf{x}) = \mathbf{x} - \mathbf{x_0}$. Its velocity is the time derivative of its displacement, $\mathbf{v}(\mathbf{x}) = \dfrac{\partial \mathbf{u}(\mathbf{x})}{\partial t}$.
    The measure of the deformation of the volume is expressed thanks to the strain tensor

$$\underline{\underline{\epsilon}} = \frac{\mathbf{V} - \mathbf{V_{def}}}{\mathbf{V}},$$

    where $\mathbf{V_{def}}$ is the volume of the element after the deformation.
    Considering that we are in the case of infinitesimal strain theory (or small deformations theory) i.e.

$$||\mathbf{u}(\mathbf{x})|| << 1 \text{ and } ||\nabla \mathbf{u}(\mathbf{x})|| << 1,$$

    the strain tensor $\underline{\underline{\epsilon}}$ can be linearized, i.e. the second-order terms of the finite strain tensor are neglected. Thus, we write the relation between strain tensor and displacement (linearity

between displacement and strain)

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \quad i,j = x,y,z.$$

Remark. *In the case of the infinitesimal theory, the tensor* $\underline{\underline{\epsilon}}$ *is a symmetric tensor, according to its definition.*

– Conservation of linear momentum law
Newton's second law of motion (or conservation of linear momentum law) tells us that

$$\mathbf{F} = \frac{\partial M\mathbf{v}}{\partial t},$$

where $\mathbf{F}$ is the forces applied to a body, M is its mass and $\mathbf{v}$ its speed.
As we consider an elastic volume $\mathbf{V}$ whose mass is M, we apply the mass conservation law

$$\frac{\partial M}{\partial t} = 0.$$

Knowing that $M = \iiint_V \rho(\mathbf{x})$ with ρ the mass density, we apply Newton's second law and the mass conservation law to the elastic volume $\mathbf{V}$. We obtain

$$\iiint_V \mathbf{F_v}(\mathbf{x}) + \iint_S \mathbf{F_s}(\mathbf{x}) = \iiint_V \rho(\mathbf{x})\frac{\partial^2 \mathbf{u}(\mathbf{x})}{\partial t^2},$$

where $\mathbf{F_v}$ is the vector gathering the volume forces and $\mathbf{F_s}$ the vector gathering the surface forces. Cauchy's theory allows to linearize $\mathbf{F_s}$ as a function of the normal outward vector $\mathbf{n}$

$$\mathbf{F_s}(\mathbf{x}) = \underline{\underline{\sigma}}(\mathbf{x})\mathbf{n},$$

where $\underline{\underline{\sigma}}$ is the stress tensor. We apply the divergence theorem in order to transform the integral over the surface into an integral over the volume

$$\iint_S \underline{\underline{\sigma}}(\mathbf{x})\mathbf{n} = \iiint_V \nabla \cdot \underline{\underline{\sigma}}(\mathbf{x}).$$

Finally, we write the equation of motion

$$\rho(\mathbf{x})\frac{\partial^2 \mathbf{u}(\mathbf{x})}{\partial t^2} = \nabla \cdot \underline{\underline{\sigma}}(\mathbf{x}) + \mathbf{F_v}(\mathbf{x}),$$

that we can also express with the velocity $\mathbf{v}$

$$\rho(\mathbf{x})\frac{\partial \mathbf{v}(\mathbf{x})}{\partial t} = \nabla \cdot \underline{\underline{\sigma}}(\mathbf{x}) + \mathbf{F_v}(\mathbf{x}). \tag{1.1.1}$$

- Material behaviour
For an elastic material, Hooke's law links the strain tensor and the stress tensor, where $\underline{\underline{C}}$ is the stiffness tensor or elasticity tensor (in Pa)

$$\underline{\underline{\sigma}}(\mathbf{x}) = \underline{\underline{C}}(\mathbf{x})\underline{\underline{\epsilon}}(\mathbf{u}(\mathbf{x})),$$

this means that

$$\sigma_{ij} = \sum_{k,l} C_{ijkl}\epsilon_{kl}.$$

The expression of the fourth-order tensor $\underline{\underline{C}}$ depends on the physical properties of the medium. The most general elastic case is the anisotropic case, where physical properties of the material are depending on the space directions. In that way, an anisotropic material could present different properties depending on its orientation. In order to better understand what is an anisotropic material, we can give the example of polarized eyeglasses: they do not allow the light to pass through as a function of the direction that they face.
Moreover, mathematically, the tensor $\underline{\underline{C}}$ follows symmetric properties

$$C_{ijkl} = C_{jikl} = C_{ijlk} = C_{klij}.$$

Thanks to these properties, we can reduce it, in 3D, as a 6×6 matrix $\mathcal{C}$, using the following Voigt's notation for numerical indexes

$$(1, 2, 3, 4, 5, 6)^T = (xx, yy, zz, yz, xz, xy)^T . \tag{1.1.2}$$

In the general anisotropic elastic case, we write

$$\mathcal{C}(\mathbf{x}) = \begin{pmatrix} C_{11}(\mathbf{x}) & C_{12}(\mathbf{x}) & C_{13}(\mathbf{x}) & C_{14}(\mathbf{x}) & C_{15}(\mathbf{x}) & C_{16}(\mathbf{x}) \\ C_{21}(\mathbf{x}) & C_{22}(\mathbf{x}) & C_{23}(\mathbf{x}) & C_{24}(\mathbf{x}) & C_{25}(\mathbf{x}) & C_{26}(\mathbf{x}) \\ C_{31}(\mathbf{x}) & C_{32}(\mathbf{x}) & C_{33}(\mathbf{x}) & C_{34}(\mathbf{x}) & C_{35}(\mathbf{x}) & C_{36}(\mathbf{x}) \\ C_{41}(\mathbf{x}) & C_{42}(\mathbf{x}) & C_{43}(\mathbf{x}) & C_{44}(\mathbf{x}) & C_{45}(\mathbf{x}) & C_{46}(\mathbf{x}) \\ C_{51}(\mathbf{x}) & C_{52}(\mathbf{x}) & C_{53}(\mathbf{x}) & C_{54}(\mathbf{x}) & C_{55}(\mathbf{x}) & C_{56}(\mathbf{x}) \\ C_{61}(\mathbf{x}) & C_{62}(\mathbf{x}) & C_{63}(\mathbf{x}) & C_{64}(\mathbf{x}) & C_{65}(\mathbf{x}) & C_{66}(\mathbf{x}) \end{pmatrix}.$$

Finally, the tensor $\underline{\underline{C}}$ satisfies

$$\underline{\underline{C\epsilon}} : \underline{\underline{\epsilon}} > 0,$$

where $\underline{\underline{a}} : \underline{\underline{b}}$ defines the scalar product between two tensors $\underline{\underline{a}}$ and $\underline{\underline{b}}$

$$\underline{\underline{a}} : \underline{\underline{b}} = \sum a_{ij} b_{ij}.$$

According to this property, $\underline{\underline{C}}$ is invertible in the space of symmetric tensors. This means that for each symmetric $\underline{\underline{\sigma}}$, it exists one and only one symmetric $\underline{\underline{\epsilon}}$ such that

$$\underline{\underline{\sigma}} = \underline{\underline{C\epsilon}}, \quad \text{and } \underline{\underline{\epsilon}} = \underline{\underline{D\sigma}},$$

where $\underline{\underline{D}}$ satisfies the same properties as $\underline{\underline{C}}$. We denote $\underline{\underline{C}}^{-1} = \underline{\underline{D}}$.

Remark. *When we use the matrix form to express $\underline{\underline{C}}$, we also have: $\mathcal{C}^{-1} = \mathcal{D}$.*

The simplest case of elasticity is the isotropic one. An isotropic medium is a medium (or material) whose physical properties are independent of the space directions. In that case, the coefficients of $\underline{\underline{\sigma}}$ simply depend on $\underline{\underline{\epsilon}}$ and on two variables μ and λ called Lamé's coefficients:

$$\sigma_{ij}(\mathbf{x}) = \lambda(\mathbf{x}) \delta_{ij} \mathrm{Tr}(\underline{\underline{\epsilon}}(\mathbf{x})) + 2\mu(\mathbf{x}) \epsilon_{ij}(\mathbf{x}), \qquad i, j = x, y, z,$$

where δ represents the Kronecker delta.
Consequently, the tensor $\underline{\underline{C}}$ only depends on the two variables μ and λ

$$\mathcal{C}(\mathbf{x}) = \begin{pmatrix} \lambda(\mathbf{x}) + 2\mu(\mathbf{x}) & \lambda(\mathbf{x}) & \lambda(\mathbf{x}) & 0 & 0 & 0 \\ \lambda(\mathbf{x}) & \lambda(\mathbf{x}) + 2\mu(\mathbf{x}) & \lambda(\mathbf{x}) & 0 & 0 & 0 \\ \lambda(\mathbf{x}) & \lambda(\mathbf{x}) & \lambda(\mathbf{x}) + 2\mu(\mathbf{x}) & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu(\mathbf{x}) & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu(\mathbf{x}) & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu(\mathbf{x}) \end{pmatrix}.$$

Remark. *Thereafter, to simplify, we omit the* $\mathbf{x}$*-dependency in the physical parameters* ρ, λ *and* μ*, the tensors* $\underline{\underline{C}}, \underline{\underline{\sigma}}$ *and* $\underline{\underline{\epsilon}}$ *and the vectors* $\mathbf{u}$ *and* $\mathbf{v}$*.*

The parameter $\mu = G$ is the shear modulus and λ does not have physical meaning ($\lambda = K - \frac{2}{3}G$, with K the bulk modulus). The parameters λ and μ can also be replaced by Young's modulus E and Poisson's coefficient ν defined by

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu}, \qquad \nu = \frac{\lambda}{2(\lambda + \mu)}.$$

Most of geophysical media are weakly anisotropic, i.e. they are anisotropic only along one direction and isotropic in the orthogonal plane. We distinguish two kinds of transverse isotropy: the vertical transverse isotropy (VTI) where the symmetric axis is vertical, and the tilted transverse isotropy (TTI) where the symmetric axis is tilted.
In the VTI or TTI case, the elastic coefficients C_{ij} are depending on Lamé's coefficients λ and μ, and on Thomsen's coefficients ε_T, γ and δ (see [36] for more details); and in the TTI case, they also depend on θ the tilt angle and ϕ the azimuth (in the horizontal plane, angle between a reference direction and tilt angle's direction, as it is shown on fig. 1.1.1).

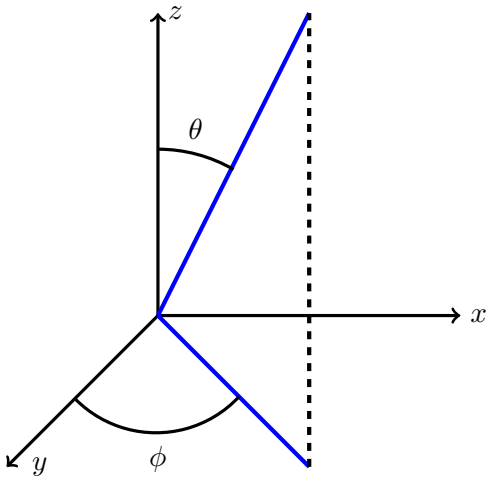


Fig. 1.1.1: θ and ϕ angles' representation

The expression of the tensor $\underline{\underline{C}}$ for the VTI case is

$$\underline{\underline{C}}^{VTI} = \begin{pmatrix} C_{11} & (C_{11} - 2C_{66}) & C_{13} & 0 & 0 & 0 \\ (C_{11} - 2C_{66}) & C_{11} & C_{13} & 0 & 0 & 0 \\ C_{13} & C_{13} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{pmatrix},$$

where

$$
\begin{aligned}
&C_{11} = (\lambda + 2\mu)\,(1 + 2\varepsilon_T)\,,\\
&C_{33} = (\lambda + 2\mu)\,,\\
&C_{44} = \mu,\\
&C_{66} = \mu\,(1 + 2\delta)\,,\\
&(C_{13} + C_{44})^2 = (\lambda + \mu)^2 + 2\delta\,(\lambda + 2\mu)\,(\lambda + \mu)\,.
\end{aligned}
$$

The TTI coefficients of tensor $\underline{\underline{C}}$ are expressed as functions of $\underline{\underline{C}}^{VTI}$ and a rotation matrix R^{TTI}

$$
C_{ijkl} = \sum_{p=1}^{3}\sum_{q=1}^{3}\sum_{r=1}^{3}\sum_{s=1}^{3} R^{TTI}_{pi} R^{TTI}_{qj} R^{TTI}_{rk} R^{TTI}_{sl} C^{VTI}_{pqrs}\,,
$$

where

$$
R^{TTI} = \begin{pmatrix} \cos\theta\cos\phi & \cos\theta\sin\phi & -\sin\theta \\ -\sin\phi & \cos\phi & 0 \\ \sin\theta\cos\phi & \sin\theta\cos\phi & \cos\theta \end{pmatrix}.
$$

Remark. *In fluids, the propagation of waves is modeled by the acoustic equations. In that case $\mu = 0$ (or $\nu = 0.5$) which greatly simplifies the expression of the elastic tensor $\underline{\underline{C}}$. In the acoustic case, we have $\sigma_{xx} = \sigma_{yy} = \sigma_{zz}$ and $\sigma_{xy} = \sigma_{xz} = \sigma_{yz} = 0$. The stress tensor $\underline{\underline{\sigma}}$ is then reduced to only one component which is called the pressure p . The pressure can be expressed in terms of velocity vector or displacement. The number of unknowns, in the 3D acoustic case, is reduced to four: the three components of the velocity or displacement vector and the pressure.*
Hence, solving the acoustic wave equations is much less computationally intensive than solving the elastodynamics equations. For this reason, the acoustic wave equations have been used for a long time in the geophysical community to model wave propagation inside the subsurface. Thanks to the increase of the computational power, it is now possible to consider elastic wave equations in order to be more accurate and to explore more and more complex regions.

Finally, gathering the equation of motion as a function of the velocity and Hooke's law, we obtain the first order formulation of the elastodynamics equations in time domain

$$
\begin{cases}
\rho\dfrac{\partial \mathbf{v}}{\partial t} = \nabla\cdot\underline{\underline{\sigma}} + \mathbf{F}_{\mathbf{v}},\\
\dfrac{\partial \underline{\underline{\sigma}}}{\partial t} = \underline{\underline{C}} : \underline{\underline{\epsilon}}(\mathbf{v}).
\end{cases}
\tag{1.1.3}
$$

Thereafter, we prefer to consider the following formulation

$$
\begin{cases}
\rho\dfrac{\partial \mathbf{v}}{\partial t} = \nabla\cdot\underline{\underline{\sigma}},\\
\dfrac{\partial \underline{\underline{\sigma}}}{\partial t} = \underline{\underline{C}} : \underline{\underline{\epsilon}}(\mathbf{v}) + \mathbf{F}_{\sigma}.
\end{cases}
\tag{1.1.4}
$$

The two formulations (1.1.3) and (1.1.4) are equivalent if and only if

$$
\frac{\partial \mathbf{F}_{\sigma}}{\partial t} = div\mathbf{F}_{\mathbf{v}}.
$$

1.1.2 Elastic waves equations in harmonic domain

System (1.1.4) can be expressed in harmonic domain by using a Fourier transform which transforms time variable t in frequency variable ξ

$$f(t) = \int_{-\infty}^{+\infty} \widehat{f}(\xi) e^{2\pi i \xi t} d\xi, \quad \forall t \in \mathbb{R},$$

and conversely

$$\widehat{f}(\xi) = \int_{-\infty}^{+\infty} f(t) e^{-2\pi i \xi t} dt, \quad \forall \xi \in \mathbb{R}.$$

The first order formulation of the elastodynamics equations in harmonic domain is thus

$$\begin{cases} i\omega\rho\widehat{\mathbf{v}} = \nabla \cdot \widehat{\underline{\underline{\sigma}}}, \\ i\omega\widehat{\underline{\underline{\sigma}}} = \underline{\underline{\underline{\underline{C}}}} : \underline{\underline{\epsilon}}(\widehat{\mathbf{v}}) + \widehat{\mathbf{F}_\sigma}, \end{cases} \tag{1.1.5}$$

where i is the imaginary unit and $\omega = 2\pi f$ is the angular frequency, f is the frequency.

In this thesis, since we are interested in the elastodynamics equations in harmonic domain only, we will omit the notation $\widehat{\cdot}$ for the sake of simplicity.

We develop system (1.1.5). For $(x, y, z) \in \Omega \subset \mathbb{R}^3$, in the general case, we get

$$\begin{cases} i\omega v_x &= \frac{1}{\rho}\left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z}\right), \\ i\omega v_y &= \frac{1}{\rho}\left(\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z}\right), \\ i\omega v_z &= \frac{1}{\rho}\left(\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z}\right), \\ i\omega \sigma_{xx} &= C_{11}\frac{\partial v_x}{\partial x} + C_{12}\frac{\partial v_y}{\partial y} + C_{13}\frac{\partial v_z}{\partial z} \\ &\quad + C_{14}\left(\frac{\partial v_z}{\partial y} + \frac{\partial v_y}{\partial z}\right) + C_{15}\left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x}\right) + C_{16}\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right) + f_{xx}, \\ i\omega \sigma_{yy} &= C_{12}\frac{\partial v_x}{\partial x} + C_{22}\frac{\partial v_y}{\partial y} + C_{23}\frac{\partial v_z}{\partial z} \\ &\quad + C_{24}\left(\frac{\partial v_z}{\partial y} + \frac{\partial v_y}{\partial z}\right) + C_{25}\left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x}\right) + C_{26}\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right) + f_{yy}, \\ i\omega \sigma_{zz} &= C_{13}\frac{\partial v_x}{\partial x} + C_{23}\frac{\partial v_y}{\partial y} + C_{33}\frac{\partial v_z}{\partial z} \\ &\quad + C_{34}\left(\frac{\partial v_z}{\partial y} + \frac{\partial v_y}{\partial z}\right) + C_{35}\left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x}\right) + C_{36}\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right) + f_{zz}, \\ i\omega \sigma_{xy} &= C_{16}\frac{\partial v_x}{\partial x} + C_{26}\frac{\partial v_y}{\partial y} + C_{36}\frac{\partial v_z}{\partial z} \\ &\quad + C_{46}\left(\frac{\partial v_z}{\partial y} + \frac{\partial v_y}{\partial z}\right) + C_{56}\left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x}\right) + C_{66}\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right) + f_{xy}, \\ i\omega \sigma_{xz} &= C_{15}\frac{\partial v_x}{\partial x} + C_{25}\frac{\partial v_y}{\partial y} + C_{35}\frac{\partial v_z}{\partial z} \\ &\quad + C_{45}\left(\frac{\partial v_z}{\partial y} + \frac{\partial v_y}{\partial z}\right) + C_{55}\left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x}\right) + C_{56}\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right) + f_{xz}, \\ i\omega \sigma_{yz} &= C_{14}\frac{\partial v_x}{\partial x} + C_{24}\frac{\partial v_y}{\partial y} + C_{34}\frac{\partial v_z}{\partial z} \\ &\quad + C_{44}\left(\frac{\partial v_z}{\partial y} + \frac{\partial v_y}{\partial z}\right) + C_{45}\left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x}\right) + C_{46}\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right) + f_{yz}. \end{cases} \tag{1.1.6}$$

Under a vectorial form, this system is written

$$i\omega\mathbf{W} + \mathbf{A}_x\frac{\partial \mathbf{W}}{\partial x} + \mathbf{A}_y\frac{\partial \mathbf{W}}{\partial y} + \mathbf{A}_z\frac{\partial \mathbf{W}}{\partial z} = 0, \tag{1.1.7}$$

with

$$\mathbf{W} = (V_x, V_y, V_z, \sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{xz})^T ,$$

$$\mathbf{A}_x = - \begin{pmatrix} 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} \\ C_{11} & C_{16} & C_{15} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{12} & C_{26} & C_{25} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{13} & C_{36} & C_{35} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{16} & C_{66} & C_{65} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{15} & C_{56} & C_{55} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{14} & C_{46} & C_{45} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{A}_y = - \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 \\ C_{16} & C_{12} & C_{14} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{26} & C_{22} & C_{24} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{36} & C_{23} & C_{34} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{66} & C_{26} & C_{46} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{56} & C_{25} & C_{45} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{46} & C_{24} & C_{44} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{A}_z = - \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 & 0 \\ C_{15} & C_{14} & C_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{25} & C_{24} & C_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{35} & C_{34} & C_{33} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{65} & C_{46} & C_{36} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{55} & C_{45} & C_{35} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{45} & C_{44} & C_{34} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Remark. *In the isotropic case, the matrices* $\mathbf{A}_x$, $\mathbf{A}_y$ *and* $\mathbf{A}_z$ *read as*

$$\mathbf{A}_x = - \begin{pmatrix} 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} \\ \lambda + 2\mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{A}_y = - \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda + 2\mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{A}_z = -\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda+2\mu & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The analysis of the eigenvalues of matrices $\mathbf{A}_x, \mathbf{A}_y$ and $\mathbf{A}_z$ in the isotropic case (which is given in section 2.2.3, chapter 2) shows that the perturbation of an isotropic elastic medium generates two kinds of waves:

- P-waves (*pressure* waves or primary waves). They are compressional waves and longitudinal (i.e. they move alongside to the direction of the waves propagation, as it is illustrated by fig. 1.1.2). They travel faster than other waves through the Earth and arrive first at seismograph stations (hence the name of "Primary waves"). Their velocity v_p is equal to $v_p = \sqrt{\frac{\lambda+2\mu}{\rho}}$. Moreover, P-waves can travel through any type of material, including fluids.

- S-waves (*shear* waves or secondary waves). They are transverse (i.e. they move perpendicularly to the direction of the waves propagation, as illustrated by fig.1.1.3). They propagate slower than P-waves: their velocity v_s is equal to $v_s = \sqrt{\frac{\mu}{\rho}}$, and typically $v_s < \frac{v_p}{\sqrt{2}}$. Moreover, contrarily to P-waves, S-waves can travel only through solids, as fluids (liquids and gases) do not support shear stresses.

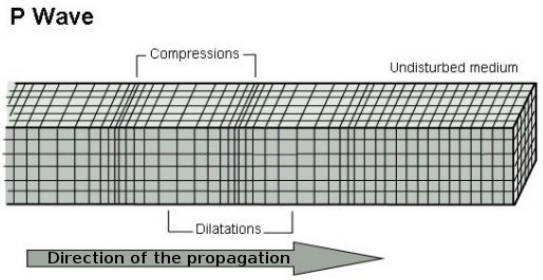


Fig. 1.1.2: P-waves propagation

Remark. *In system* (1.1.6)*, we can generate P-waves only by taking* $f_{xy} = f_{xz} = f_{yz} = 0$ *and for only S-waves,* $f_{xx} = f_{yy} = f_{zz} = 0$*.*

1.1.3 Elastic waves 2D approximation

After some calculations, we can approximate the 3D system (1.1.6) in the isotropic case by a 2D system by assuming an infinite invariant direction. In geophysics, it is generally the y direction, and it

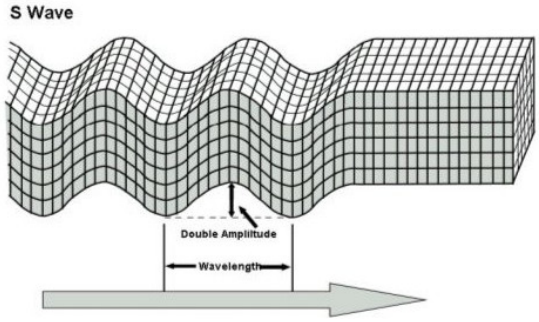


Fig. 1.1.3: S-waves propagation

consists in removing all the spatial y derivatives in the 3D system. The 2D general system read as

$$\left\{\begin{array}{rcl}
i\omega v_x &=& \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xx}}{\partial x}+\dfrac{\partial \sigma_{xz}}{\partial z}\right),\\
i\omega v_y &=& \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xy}}{\partial x}+\dfrac{\partial \sigma_{yz}}{\partial z}\right),\\
i\omega v_z &=& \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xz}}{\partial x}+\dfrac{\partial \sigma_{zz}}{\partial z}\right),\\
i\omega \sigma_{xx} &=& C_{11}\dfrac{\partial v_x}{\partial x}+C_{13}\dfrac{\partial v_z}{\partial z}+C_{14}\dfrac{\partial v_y}{\partial z}+C_{15}\left(\dfrac{\partial v_x}{\partial z}+\dfrac{\partial v_z}{\partial x}\right)+C_{16}\dfrac{\partial v_y}{\partial x}+f_{xx},\\
i\omega \sigma_{yy} &=& C_{12}\dfrac{\partial v_x}{\partial x}+C_{23}\dfrac{\partial v_z}{\partial z}+C_{24}\dfrac{\partial v_y}{\partial z}+C_{25}\left(\dfrac{\partial v_x}{\partial z}+\dfrac{\partial v_z}{\partial x}\right)+C_{26}\dfrac{\partial v_y}{\partial x}+f_{yy},\\
i\omega \sigma_{zz} &=& C_{13}\dfrac{\partial v_x}{\partial x}+C_{33}\dfrac{\partial v_z}{\partial z}+C_{34}\dfrac{\partial v_y}{\partial z}+C_{35}\left(\dfrac{\partial v_x}{\partial z}+\dfrac{\partial v_z}{\partial x}\right)+C_{36}\dfrac{\partial v_y}{\partial x}+f_{zz},\\
i\omega \sigma_{xy} &=& C_{16}\dfrac{\partial v_x}{\partial x}+C_{36}\dfrac{\partial v_z}{\partial z}+C_{46}\dfrac{\partial v_y}{\partial z}+C_{56}\left(\dfrac{\partial v_x}{\partial z}+\dfrac{\partial v_z}{\partial x}\right)+C_{66}\dfrac{\partial v_y}{\partial x}+f_{xy},\\
i\omega \sigma_{xz} &=& C_{15}\dfrac{\partial v_x}{\partial x}+C_{35}\dfrac{\partial v_z}{\partial z}+C_{45}\dfrac{\partial v_y}{\partial z}+C_{55}\left(\dfrac{\partial v_x}{\partial z}+\dfrac{\partial v_z}{\partial x}\right)+C_{56}\dfrac{\partial v_y}{\partial x}+f_{xz},\\
i\omega \sigma_{yz} &=& C_{14}\dfrac{\partial v_x}{\partial x}+C_{34}\dfrac{\partial v_z}{\partial z}+C_{44}\dfrac{\partial v_y}{\partial z}+C_{45}\left(\dfrac{\partial v_x}{\partial z}+\dfrac{\partial v_z}{\partial x}\right)+C_{46}\dfrac{\partial v_y}{\partial x}+f_{yz}.
\end{array}\right. \tag{1.1.8}$$

In the isotropic case, we can distinguish two 2D systems:

- System (1.1.9) where P and S waves are only polarized in (x,z) plane. We have P-SV (V for vertical) waves. We remark that the interactions between P and SV waves at the surface generate surface waves which are called Rayleigh waves (see fig. 1.1.4 for Rayleigh waves propagation).

$$\left\{\begin{array}{rcl}
i\omega v_x &=& \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xx}}{\partial x}+\dfrac{\partial \sigma_{xz}}{\partial z}\right),\\
i\omega v_z &=& \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xz}}{\partial x}+\dfrac{\partial \sigma_{zz}}{\partial z}\right),\\
i\omega \sigma_{xx} &=& C_{11}\dfrac{\partial v_x}{\partial x}+C_{13}\dfrac{\partial v_z}{\partial z}+C_{15}\left(\dfrac{\partial v_x}{\partial z}+\dfrac{\partial v_z}{\partial x}\right)+f_{xx},\\
i\omega \sigma_{zz} &=& C_{13}\dfrac{\partial v_x}{\partial x}+C_{33}\dfrac{\partial v_z}{\partial z}+C_{35}\left(\dfrac{\partial v_x}{\partial z}+\dfrac{\partial v_z}{\partial x}\right)+f_{zz},\\
i\omega \sigma_{xz} &=& C_{15}\dfrac{\partial v_x}{\partial x}+C_{35}\dfrac{\partial v_z}{\partial z}+C_{55}\left(\dfrac{\partial v_x}{\partial z}+\dfrac{\partial v_z}{\partial x}\right).
\end{array}\right. \tag{1.1.9}$$

- SH (H for horizontal) waves (or Love waves) are polarized in planes (x,y) and (y,z), they are

independant of P and SV waves and their propagation (see fig.1.1.5) is given by system :

$$\begin{cases} i\omega v_y &= \frac{1}{\rho}\left(\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial z}\right), \\ i\omega \sigma_{yy} &= C_{24}\frac{\partial v_y}{\partial z} + C_{26}\frac{\partial v_y}{\partial x} + f_{yy}, \\ i\omega \sigma_{xy} &= C_{46}\frac{\partial v_y}{\partial z} + C_{66}\frac{\partial v_y}{\partial x} + f_{xy}, \\ i\omega \sigma_{yz} &= C_{44}\frac{\partial v_y}{\partial z} + C_{46}\frac{\partial v_y}{\partial x} + f_{yz}. \end{cases} \tag{1.1.10}$$

We do not describe in more details the characteristics and role of Love and Rayleigh waves.

Remark. *In this thesis when we focus on the 2D case, we consider the system 1.1.9.*

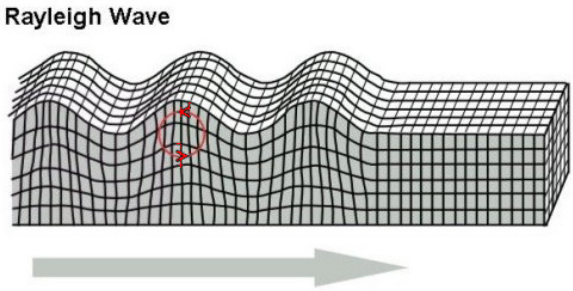


Fig. 1.1.4: Rayleigh-waves propagation

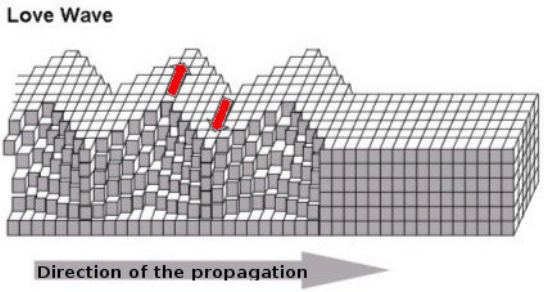


Fig. 1.1.5: Love-waves propagation

1.1.4 Source term $\mathbf{f_v}$

So far, we have described how we obtain the elastic waves equations and how the presence of the different media is expressed. We have not described yet the source term $\mathbf{f_v} = (f_x,\ f_y,\ f_z)$ that generates the waves.
In geophysics, the source term $\mathbf{f_v}$ generally represents the pulse emitted by the physical explosive source. This is expressed by a Dirac distribution

$$< \varphi(\mathbf{x})\delta(\mathbf{x}) >= \varphi(0), \quad \forall \varphi \in C_0(\Omega).$$

But, as this can be difficult to model numerically, we can express it by a Gaussian function

$$f(x,y,z) = Ae^{-(\frac{x-x_0}{2\varsigma_x^2}+\frac{y-y_0}{2\varsigma_y^2}+\frac{z-z_0}{2\varsigma_z^2})},$$

where A is the amplitude, (x_0, y_0, z_0) are the coordinates of the source and ς_x, ς_y, and ς_z the spreads of the blob as a function of x, y and z respectively.
A second solution, specific to Finite Element methods, consists in approximating δ by a function δ_h that satisfies

$$\int_\Omega \delta_h \varphi_i = \varphi_i(0)$$

for all the basis functions φ_i of the finite element space. This is the solution that we have chosen in our codes.

1.1.5 Boundary conditions

Finally, we need to discuss how to handle the physical boundary where the domain is bounded and how to model transparent boundary where the domain is infinite or very large compared to the wavelength. From a mathematical point of view, these spatial boundary conditions are crucial to insure the consistency of the numerical scheme and the uniqueness of the solution.
We will distinguish two kinds of physical boundary conditions:

- For a free surface condition simulating a physical interface between the domain and the air on the boundary Γ_l, we impose

$$\underline{\underline{\sigma}}\mathbf{n} = 0, \tag{1.1.11}$$

  (no normal tractions), and no particular condition on the velocity components.

- For a wall boundary condition on the boundary Γ_N

$$V_i = 0, \qquad i = x, y, z, \tag{1.1.12}$$

For the simulation, the theoretically unbounded propagation domain is artificially truncated and we have to define an appropriate numerical treatment if we want to reduce as much as possible the reflections on the boundary Γ_a.More precisely, we would like to impose some conditions on the corresponding boundary faces which allow the absorption of the waves when they reach the artificial boundary. There exist several possible strategies including Perfectly Matched Layers (PML, C-PML, etc.) or absorbing conditions which are approximations of an exact transparent condition. The latter option has been adopted in this thesis and we use two kinds of absorbing conditions. Both are first order absorbing boundaries conditions (ABC). They are less accurate than higher order ABC but much easier to implement.
We consider a Sommerfeld condition which is expressed, according to [37], in the 3D TTI case on Γ_a such as

$$\underline{\underline{\sigma}} \cdot \mathbf{n} + PA(\theta', \phi')P^T\mathbf{v} = 0, \tag{1.1.13}$$

where,

- P is defined such as :

  - if $|n_z| \neq 1$, $P = \begin{pmatrix} n_x & -\frac{n_y}{\sqrt{n_x^2+n_y^2}} & -\frac{n_x n_z}{\sqrt{n_x^2+n_y^2}} \\ n_y & \frac{n_x}{\sqrt{n_x^2+n_y^2}} & -\frac{n_y n_z}{\sqrt{n_x^2+n_y^2}} \\ n_z & 0 & \sqrt{n_x^2+n_y^2} \end{pmatrix}$,

– if $|n_z| = 1$, $P = \begin{pmatrix} \cos(n_z\frac{\pi}{2}) = 0 & 0 & -\sin(n_z\frac{\pi}{2}) \\ 0 & 1 & 0 \\ \sin(n_z\frac{\pi}{2}) & 0 & \cos(n_z\frac{\pi}{2}) = 0 \end{pmatrix}$,

- $A(\theta,\phi) = A_p(\theta,\phi) + A_s(\theta,\phi) = \begin{pmatrix} A_{p1}(\theta,\phi) \\ A_{p2}(\theta,\phi) \\ A_{p3}(\theta,\phi) \end{pmatrix} + \begin{pmatrix} A_{s1}(\theta,\phi) \\ A_{s2}(\theta,\phi) \\ A_{s3}(\theta,\phi) \end{pmatrix}$, with

$$
\begin{aligned}
A_{p1}(\theta,\phi) &= \varsigma(\theta,\phi)\left(\kappa\sin^2\phi + \cos^2\phi(\kappa\cos^2\theta + \sin^2\theta)\right), \\
A_{p2}(\theta,\phi) &= -\varsigma(\theta,\phi)\left((\kappa-1)\cos\phi\sin\phi\sin^2\theta\right), \\
A_{p3}(\theta,\phi) &= -\varsigma(\theta,\phi)\left((\kappa-1)\cos\phi\cos\theta\sin\theta\right), \\
A_{s1}(\theta,\phi) &= 0, \\
A_{s2}(\theta,\phi) &= -\rho v_s \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}^T, \\
A_{s3}(\theta,\phi) &= -\rho v_s \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}^T,
\end{aligned}
$$

where $\kappa = \sqrt{1-2\varepsilon_T}$ and

$$
\varsigma(\theta,\phi) = \frac{-\rho v_p}{\sqrt{\kappa^2\cos^2\theta\cos^2\phi + \kappa^2\sin^2\theta + \sin^2\theta\cos^2\phi}} \begin{pmatrix} \kappa\sin^2\phi + \cos^2\phi(\kappa\cos^2\theta + \sin^2\theta) \\ -(\kappa-1)\cos\phi\sin\phi\sin^2\theta \\ -(\kappa-1)\cos\theta\sin\theta\cos\phi \end{pmatrix}^T.
$$

- θ' is defined by

$$
\begin{cases}
\cos\theta' = \left(\dfrac{-n_x n_z}{\sqrt{n_x^2+n_y^2}}\cos\phi - \dfrac{n_y n_z}{\sqrt{n_x^2+n_y^2}}\sin\phi\right)\sin\theta + \sqrt{n_x^2+n_y^2}\cos\theta & \text{if } |n_z| \neq 1, \\
\cos\theta' = -n_z\cos\phi\sin\theta & \text{if } |n_z| = 1,
\end{cases}
$$

and $\sin\theta' = \sqrt{1-\cos^2\theta}$.

- ϕ' is defined such as

– if $\sin\theta' \neq 0$,

$$
\text{if } |n_z| \neq 1, \quad \begin{cases} \cos\phi' = (n_x\cos\phi + n_y\sin\phi)\dfrac{\sin\theta}{\sin\theta'} + n_z\dfrac{\cos\theta}{\sin\theta'}, \\ \sin\phi' = \left(\dfrac{-n_y\cos\phi}{\sqrt{n_x^2+n_y^2}} - \dfrac{n_x\sin\phi}{\sqrt{n_x^2+n_y^2}}\right)\dfrac{\sin\theta}{\sin\theta'}. \end{cases}
$$

$$
\text{if } |n_z| = 1, \quad \begin{cases} \cos\phi' = n_z\dfrac{\cos\theta}{\sin\theta'}, \\ \sin\phi' = \dfrac{\sin\theta\sin\phi}{\sin\theta'}. \end{cases}
$$

– if $\sin\theta' = 0$,

$$\cos\phi' = 1 \text{ and } \sin\phi' = 0.$$

Remark. *In the isotropic case,* $\kappa = 1$, $\theta = 0$ *and* $\phi = 0$ *involving* $\cos\phi' = 1$ *and* $\sin\phi' = 0$ *and* $\cos\theta' = 1$ *and* $\sin\theta' = 0$. *Matrix* $A(\theta, \phi)$ *becomes*

$$A(\theta,\phi) = \begin{pmatrix} -\rho v_p & 0 & 0 \\ 0 & -\rho v_s & 0 \\ 0 & 0 & -\rho v_s \end{pmatrix}.$$

The matrix $PA(\theta', \phi')P^T$ *expresses as*

$$\begin{pmatrix} -\rho v_p n_x^2 - \rho v_s (n_x^2 n_z^2 + n_y^2)\dfrac{1}{n_x^2+n_y^2} & -\rho v_p n_x n_y + \rho v_s \frac{n_x n_y}{n_x^2+n_y^2}(1-n_z^2) & -\rho v_p n_x n_z + \rho v_s n_x n_z \\ -\rho v_p n_x n_y + \rho v_s \dfrac{n_x n_y}{n_x^2+n_y^2}(1-n_z^2) & -\rho v_p n_y^2 + \rho v_s (n_x^2 + n_z^2 n_y^2)\frac{1}{n_x^2+n_y^2} & -\rho v_p n_y n_z + \rho v_s n_y n_z \\ -\rho v_p n_x n_z + \rho v_s n_x n_z & -\rho v_p n_y n_z + \rho v_s n_y n_z & -\rho v_p n_z^2 - \rho v_s (n_x^2 + n_y^2) \end{pmatrix}.$$

Another option for the absorbing condition is to use the plus-minus decomposition of the flux matrix $\mathbf{D_n}$; this will be detailed in the next chapter. Here, we just give the main idea of this decomposition: when a wave crosses over an interface between two media M^+ and M^-, we have

$$\underline{\underline{\sigma}}^+ \mathbf{n}^+ + \underline{\underline{\sigma}}^- \mathbf{n}^- = 0, \quad \text{and } \mathbf{v}^+ + \mathbf{v}^- = 0.$$

So, we can define a simple treatment which consists in considering that the inflow flux $(\mathbf{D_n})^-$ is zero for each face on the artificial boundary, i.e.

$$(\mathbf{D_n})^- \left(\mathbf{v}, \underline{\underline{\sigma}}\right)^T = 0 \text{ on } \Gamma_a. \tag{1.1.14}$$

1.2 Numerical scheme: modeling of the elastic system

In the following, we review the most popular methods for discretizing system (1.1.5).

1.2.1 Finite difference method (FDM)

The finite difference method is the most simple and intuitive discretization method. A lot of numerical FD schemes for geophysical simulation in time domain are based on the seminal work of [18]. We refer to [19] for an example in 3D domain.
FDM is formulated on uniform grids and based on the strong formulation of the equations. Thanks to the fact that the partial derivatives $\dfrac{\partial}{\partial i}, i = x, y, z$, are approximated by Taylor expansions

$$\frac{\partial \mathbf{u}(\mathbf{x})}{\partial i} = \frac{\mathbf{u}(\mathbf{x} + \Delta i) - \mathbf{u}(\mathbf{x})}{\Delta i}, \quad i = x, y, z,$$

where Δi is a constant called the step. Generally, $\Delta x = \Delta y = \Delta z = h$. This implies simple systems to solve and an implementation easy to do. It presents the best compromise between accuracy and computational efficiency, but it lacks flexibility: as it is not formulated on unstructured meshes, the mesh cannot be adapted to local wavelength or to the topography of the subsurface. Moreover, if we want accurate solutions, we have to use very fine discretization and as a consequence computational costs increases drastically. When the subsurface has a strong topography, it requires a specific treatment as proposed for instance in [20] or [21], which strongly complicates the implementation of the method. However, today it is still the most popular numerical method used by oil industries in their production algorithms.

1.2.2 Finite element method (FEM)

The finite element method is based on a weak formulation. The numerical solution is a polynomial approximation and one obtains an optimal convergence rate $p+1$ if p is the used interpolation order. Its principal advantage is that it is formulated on unstructured meshes, so it can handle the complex topography of the subsurface and its heterogeneities. Examples of extension of FEM to the elastic wave equations can be found in [22, 23]. However, the use of unstructured meshes implies a computational cost more important than the one needed for the FDM. Moreover, with the construction of the linear system, a mass matrix and a stiffness matrix appear. The resolution of the system leads, in time domain, to an implicit representation of the solution which indicates that we have to invert the mass matrix which is generally a full matrix whose inversion can be expensive. To overcome this problem, one can use a mass condensation technique in time domain as proposed in [38, 39].
In frequency domain, the scheme is subject to pollution effects [40, 41, 42].
Beside the mass condensation technique, another particular FEM has been developed: the spectral element method (SEM). This method combines the use of Lagrange interpolants and of Gauss-Lobatto-Legendre quadratures that makes the mass matrix diagonal, and this without loss of accuracy. In other words, we can inverse easily the mass matrix and we keep the same convergence order than with classical FEM, namely $p+1$ if we use interpolation order p. Another advantage of the SEM is it can be used with high order formulation. We refer to the works of Komatitsch et al. ([24, 43, 44]), and to De Basabe and Sen's work [25] for example of applications of SEM in geophysical context.
The main drawback of SEM is that it is formulated on meshes made of quadrangles in 2D and on hexahedra in 3D. This strongly limits the geometrical flexibility. Here are some works in 2D where the SEM is applied on triangular meshes ([45, 46, 47]) but the implementation is complicated, particularly when we consider high-order FE.
Furthermore, the SEM is particularly used for time domain waves propagation. Indeed in frequency domain, as classical FEM, they suffer from pollution effects.

1.2.3 Discontinuous Galerkin (DG) method

The discontinuous Galerkin method is a particular FEM such that the mass matrix is block diagonal and so easily inversible. Hence, it is a rival method to the SEM.
It is a combination between FEM, because the solution is represented by a polynomial approximation, and finite volume method (FVM), because the interactions between the interfaces are computed with numerical fluxes.
The first DG approximation has been proposed by Reed and Hill in 1973 for the resolution of the hyperbolic problem of neutrons transport equation [48]. Since then DG methods have been developed to solve various partial derivatives equations. In [49], B. Cockburn describes DG methods and applies them to various linear and non-linear problems.
A first advantage of this method is that it can be formulated on unstructured triangular meshes (triangles in 2D and tetrahedrons in 3D) and can be applied on hexahedra elements too. Thus, we benefit from a high level of flexibility in the mesh construction, we can adapt the size of the element (h adaptivity, see fig. 1.2.2) to the local wavelength or to a heckled topography.
DG methods use basis functions which are, as their name indicate, discontinuous functions: they are polynomial functions on the element but are discontinuous through its boundaries. It is important to note that the exchange of informations between two elements is performed across the interfaces only by means of numerical fluxes. The distinction between all the different DG methods arises from the expression of their numerical fluxes. We refer to [50] for a review of the various DG methods which can be applied to the elliptic problems for example.
The discontinuity of the basis functions implies that the degrees of freedom (DOF) belong to only one element. Thus, the DOF that are located at the interfaces (edges in 2D and faces in 3D) have to be

duplicated as we can see on fig. 1.2.4b. Due to the fact that one element has its own degrees of freedom and that the information is only exchanged at the interfaces, the elements are independent from each other and therefore, each element can have its own interpolation order (p adaptivity, see fig.1.2.3) for the local approximation of the field elements. We remark that as we use polynomial functions, we benefit from a large choice of interpolation orders, and prove that if the fluxes are appropriately defined, we obtain the same convergence order as classical FEM.
Moreover, with the use of discontinuous functions, any kind of boundary condition can be implemented. One can even treat transmission conditions between elastic and acoustic media [51] as often encountered in geophysical applications.
Finally, the use of discontinuous basis functions makes DG method a local method, and, as a consequence an easily parallelizable method.

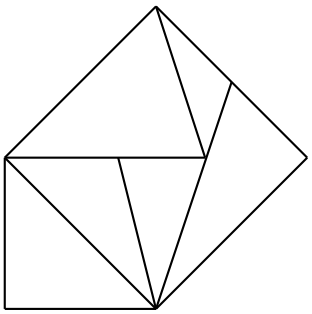


Fig. 1.2.1: Example of non-conforming mesh.

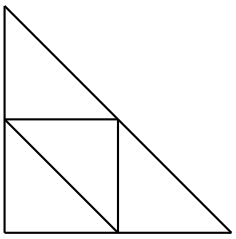


Fig. 1.2.2: Example of h-adaptivity.

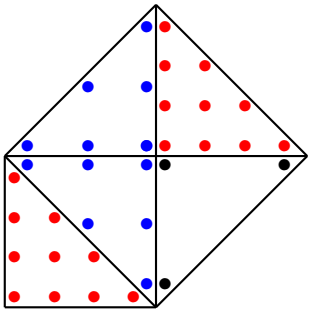


Fig. 1.2.3: Example of p-adaptivity (P_1, P_2, P_3).

For these reasons, DG methods have became very popular during the last two decades. A lot of works have been devoted to their applications to wave equations. We can cite, for example, in time domain

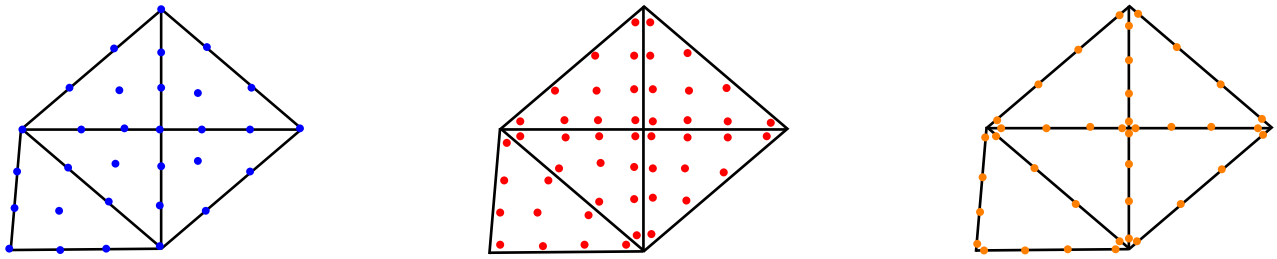
the works of Dumbser and Käser. In [52, 53], they treat the isotropic elastic case considering fluxes based on the solution of Riemann problems and computed with upwind scheme. In [54] they extend their method to anisotropic media considering Godunov and Rusanov fluxes. In [55] Delcourte *et al.* are interested in a DG method using centered flux for the first order formulation of the elastic wave equations. In [56], Grote *et al.* use centered flux DG formulation with interior penalty, also known as interior penalty DG method (IPDG), to solve the acoustic wave equation. The stability of this method is demonstrated in [57]. In [27], they use the *hp*-property of DG method to solve the wave equations. The application of DG methods to seismic imaging in frequency domain is more recent. It has been proposed by Brossier in his PhD. thesis [58]. We also refer the reader to the paper of Brossier, Etienne, Operto and Virieux [59].

1.2.4 Hybridizable Discontinuous Galerkin (HDG) method

The main drawback of DG methods is the large number of DOF as compared to classical FEM or SEM. As we have explained, in DG methods the degrees of freedom belong to only one element, and consequently, those located at the faces have to be duplicated as it is shown figs. 1.2.4a and 1.2.4b. This involves important computation costs.
To reduce the computational costs, Cockburn, Gopalakrishnan and Lazarov introduced the hybridizable discontinuous Galerkin method in 2009 for second order elliptic problems, see [28]. It has been applied to various problems: in particular to the Maxwell's equations [30, 29] in time harmonic domain and to the acoustics and elastodynamics, considering displacement gradient-velocity-pressure formulation, in time implicit domain [35]. We refer the reader to chapter 3 for a more detailed bibliography .
The HDG method is a class of DG method and we recover the same advantages of the latter: it is based on unstructured meshes and on discontinuous basis functions, it is an easily parallelizable method and it is well suited to $hp-$adaptivity. The main difference is that the HDG formulation is based on the introduction of a new variable Λ which is defined only on the interfaces of the mesh. The variable Λ is a Lagrange multiplier and represents the trace of the numerical solution on each interface (edge in 2D, face in 3D) of the mesh.
As Λ is defined only on the interfaces, HDG basis functions are discontinuous at the boundaries of the edges/faces as it is shown on fig. 1.2.4c. In other words the basis functions are continuous over an edge in 2D but discontinuous at its vertices. In 3D they are continuous over a face but discontinuous at the edges. So, only the vertices in 2D or the degrees of freedom located at the edges in 3D have to be duplicated. Hence, it contains less unknowns than $\left(\mathbf{v}, \underline{\sigma}\right)^T$, and the linear system involving only Λ is easier to solve. Once this is done, we recover the solution of the initial problem $\left(\mathbf{v}, \underline{\sigma}\right)^T$ thanks to independent elementwise calculations.
This procedure allows us to reduce the number of unknowns of the linear global system, i.e. the global number of degrees of freedom. On the one hand because Λ, as we will see later, has three components in 3D (two in 2D) whereas in classical DG method the vector of unknowns of the linear system has nine components in 3D (five in 2D). On the other hand because the size of the matrix to be inverted and the number of global degrees of freedom only depend on the number of edges/faces of the mesh and on the number of degrees of freedom of each edge/face, whereas for classical DG method they depend on the number of cells of the mesh and on the number of degrees of freedom of each cell.

Conclusion

In this chapter, we have recalled how to obtain the elastodynamics equations in time domain and in frequency domain. In this thesis, we are interested in solving the elastodynamics problem in



(a) (b) (c)

Fig. 1.2.4: Distribution of the global degrees of freedom for the FEM 1.2.4a, the DG method 1.2.4b and the HDG method 1.2.4c with an interpolation order of 3

frequency domain (1.1.5)

$$\begin{cases} i\omega\rho\widehat{v} = \nabla\cdot\widehat{\underline{\underline{\sigma}}} + \widehat{\mathbf{f}_{\mathbf{v}}}, \\ i\omega\widehat{\underline{\underline{\sigma}}} = \underline{\underline{C}} : \underline{\underline{\epsilon}}(\widehat{v}). \end{cases}$$

Our objective is to solve the 3D formulation with the HDG method because as the 3D formulation is very costly in memory terms specifically, we want to try to reduce this memory cost by using the HDG method.
Before doing this, we have to test and validate the HDGM for the 2D formulation. To do that, we first develop classical DG methods, the centered flux DG method and the upwind flux DG method, which are presented in the next chapter, for the resolution of equations (1.1.5). These nodal DG methods will serve as reference methods, specifically the upwind flux DG method, for the comparison with the HGD method.
We also consider another DG formulation for the comparison, the interior penalty DG method (IPDG) based on the second order formulation of the wave equations. We are interested in the comparison between HDG formulation and this method because, as we will see later, the unknowns vector of the linear global system has the "same" number of components (i.e. (u_x, u_y, u_z) for the IPDG formulation and $(\lambda_x, \lambda_y, \lambda_z)$ for the HGD method), only the number of degrees of freedom (i.e. the global size of the unknowns vector) is changing.

Chapter 2

Classical DG methods: centered flux DG and upwind flux DG methods

In the previous chapter, we have presented various numerical methods for solving the wave equations and we have explained that discontinuous Galerkin (DG) methods, which are a combination between FEM (with the use of polynomial functions) and finite volume method (the neighboring elements are linked thanks to numerical fluxes) present many advantages. They are local; they can be formulated on unstructured meshes; they are well-adapted to high performance computing; they allow *hp*-adaptivity. In this chapter, we describe and we analyze the performance of two classical DG schemes for the first order formulation of the elastic wave formulation: a centered flux DG method, as it is developed in [55], and an upwind flux DG scheme based on the same principle than the one developed for Maxwell's equations in [60]. In the first section, we introduce the specific notations to DG methods and we describe the discontinuous basis functions that will be used to discretize the formulations. In the second section, we give the general expression of a DG schemes and emphasize once again that the various DG schemes differ from the expression of the numerical fluxes. Then, we express the numerical fluxes leading to a centered fluxes DG formulation and to an upwind fluxes DG formulation. In the last part of this chapter, we compare the performances of the two DG methods on a simple test case, in order to determine the most appropriate formulation to serve as a benchmark for the HDG method that we develop in chapter 3.

2.1 Problem statement and notations

In this section, we recall the formulation of the isotropic elastic system and we introduce all the notations and definitions that we need for the expressions of the centered flux and the upwind flux DG formulations. Finally, we describe the Lagrange basis functions for DG methods.

2.1.1 Isotropic elastodynamics system

We remind that we want to solve the first order formulation of the elastodynamics system in harmonic domain. In the 3D case, for $\mathbf{x} = (x, y, z) \in \Omega \subset \mathbb{R}^3$

$$\left\{ \begin{array}{rcll} i\omega\rho\mathbf{v} & = & \nabla \cdot \underline{\underline{\sigma}} & \text{in } \Omega, \\ i\omega\underline{\underline{\sigma}} & = & \underline{\underline{C}} : \underline{\underline{\epsilon}}(\mathbf{v}) + \mathbf{f}_\sigma & \text{in } \Omega, \end{array} \right. \tag{2.1.1}$$

The boundary conditions are given by

$$\left\{ \begin{array}{rl} \underline{\underline{\sigma}}\mathbf{n} & = 0 \text{ on } \Gamma_a, \\ (\mathbf{D_n})^- \mathbf{W} & = 0 \text{ on } \Gamma_b, \end{array} \right. \tag{2.1.2}$$

with $\Gamma_a \cup \Gamma_b = \partial\Omega$ and $\Gamma_a \cap \Gamma_b = \emptyset$. We remind that the first relation of (2.1.2) describes a free surface condition on Γ_a while the second relation represents an absorbing boundary condition on Γ_b.

For ρ and $\underline{\underline{C}}$ regular enough, system (2.1.1) with boundary conditions (2.1.2) admits an unique solution $\left(\mathbf{v}, \underline{\underline{\sigma}}\right)$ in $\overline{H}^1(\Omega) \times H^{div}(\Omega)$.

The developed equations in the 3D isotropic case write

$$\left\{\begin{array}{rcl} i\omega v_x &=& \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xx}}{\partial x}+\dfrac{\partial \sigma_{xy}}{\partial y}+\dfrac{\partial \sigma_{xz}}{\partial z}\right), \\ i\omega v_y &=& \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xy}}{\partial x}+\dfrac{\partial \sigma_{yy}}{\partial y}+\dfrac{\partial \sigma_{yz}}{\partial z}\right), \\ i\omega v_z &=& \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xz}}{\partial x}+\dfrac{\partial \sigma_{yz}}{\partial y}+\dfrac{\partial \sigma_{zz}}{\partial z}\right), \\ i\omega \sigma_{xx} &=& (\lambda+2\mu)\dfrac{\partial v_x}{\partial x}+\lambda\left(\dfrac{\partial v_y}{\partial y}+\dfrac{\partial v_z}{\partial z}\right), \\ i\omega \sigma_{yy} &=& (\lambda+2\mu)\dfrac{\partial v_y}{\partial y}+\lambda\left(\dfrac{\partial v_x}{\partial x}+\dfrac{\partial v_z}{\partial z}\right), \\ i\omega \sigma_{zz} &=& (\lambda+2\mu)\dfrac{\partial v_z}{\partial z}+\lambda\left(\dfrac{\partial v_y}{\partial y}+\dfrac{\partial v_x}{\partial x}\right), \\ i\omega \sigma_{xy} &=& \mu\left(\dfrac{\partial v_x}{\partial y}+\dfrac{\partial v_y}{\partial x}\right), \\ i\omega \sigma_{yz} &=& \mu\left(\dfrac{\partial v_z}{\partial y}+\dfrac{\partial v_y}{\partial z}\right), \\ i\omega \sigma_{xz} &=& \mu\left(\dfrac{\partial v_x}{\partial z}+\dfrac{\partial v_z}{\partial x}\right). \end{array}\right. \tag{2.1.3}$$

We prefer to use the vectorial form (1.1.7) that we have given in the previous chapter

$$i\omega \mathbf{W} + \mathbf{A}_x \frac{\partial \mathbf{W}}{\partial x} + \mathbf{A}_y \frac{\partial \mathbf{W}}{\partial y} + \mathbf{A}_z \frac{\partial \mathbf{W}}{\partial z} = 0, \tag{2.1.4}$$

with

$$\mathbf{W} = \left(v_x, v_y, v_z, \sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{xz}\right)^T,$$

$$\mathbf{A}_x = -\begin{pmatrix} 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} \\ \lambda+2\mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{A}_y = -\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda+2\mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{A}_z = -\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda+2\mu & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

In 2D, we consider the system (3.1.1) where the y components vanish and where we are still in presence of P and S waves (contrary to the system (1.1.10) where only SH waves propagate). The obtained system is then composed of 5 equations with 5 unknowns

$$\begin{cases} i\omega v_x &= \frac{1}{\rho}\left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xz}}{\partial z}\right), \\ i\omega v_z &= \frac{1}{\rho}\left(\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{zz}}{\partial z}\right), \\ i\omega \sigma_{xx} &= (\lambda+2\mu)\frac{\partial v_x}{\partial x} + \lambda\frac{\partial v_z}{\partial z}, \\ i\omega \sigma_{zz} &= (\lambda+2\mu)\frac{\partial v_z}{\partial z} + \lambda\frac{\partial v_x}{\partial x}, \\ i\omega \sigma_{xz} &= \mu\left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x}\right). \end{cases} \tag{2.1.5}$$

The 2D vectorial form is then

$$i\omega \mathbf{W} + \mathbf{A}_x\frac{\partial \mathbf{W}}{\partial x} + \mathbf{A}_z\frac{\partial \mathbf{W}}{\partial z} = 0, \tag{2.1.6}$$

with

$$\mathbf{W} = (v_x, v_z, \sigma_{xx}, \sigma_{zz}, \sigma_{xz})^T,$$

$$\mathbf{A}_x = -\begin{pmatrix} 0 & 0 & \frac{1}{\rho} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\rho} \\ \lambda+2\mu & 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{A}_z = -\begin{pmatrix} 0 & 0 & 0 & 0 & \frac{1}{\rho} \\ 0 & 0 & 0 & \frac{1}{\rho} & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & \lambda+2\mu & 0 & 0 & 0 \\ \mu & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Thereafter, we consider the 3D vectorial form as the 2D formulation can be deduced from it by removing y components.

2.1.2 Notations and definitions

We discretize the computational domain Ω by a collection of disjoint elements (tetrahedra in the 3D case, triangles in the 2D case) $\mathcal{T}_h$. We denote by

- F : a face of an element K (see fig. 2.1.1),
- $\mathcal{F}(K)$: the set of faces (edges in the 2D case, anyhow, we call it face) of an element K,

- $\mathcal{F}_b$: the union of all the boundary faces F_b, i.e. $F_b = \partial K \cap \partial\Omega$,
- $\mathcal{F}_i$: the union of all interior faces F_i i.e. $F_i = \partial K \cap \partial K'$ where K and K' are neighbours,
- $\mathcal{F}_h$: the set of all faces $\mathcal{T}_h$, i.e. $\mathcal{F}_h = \mathcal{F}_i \cup \mathcal{F}_b$,
- $\mathbf{n}$: the outward unit normal vector to K, $\mathbf{n}'$ the outward unit normal vector to K' (see fig. 2.1.2).

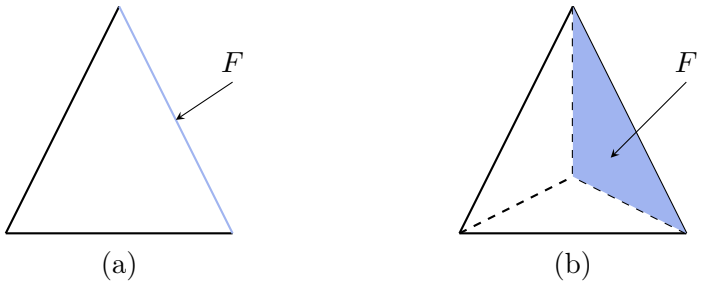


Fig. 2.1.1: Element K with a face F in 2D (a) and in 3D (b).

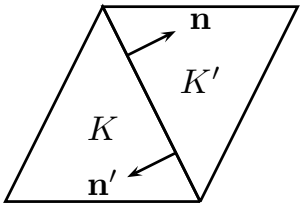


Fig. 2.1.2: Definition of the normal outward vectors $\mathbf{n}$ and $\mathbf{n}'$.

In 3D, the tetrahedron defined by vertices $\widehat{S}_1$ (0,0,0), $\widehat{S}_2$ (1,0,0), $\widehat{S}_3$ (0,1,0) and $\widehat{S}_4$ (0,0,1) (see fig.2.1.3) is called the reference element $\widehat{K}$, the associated reference face $\widehat{F}$ is the triangle $\widehat{S}_1\widehat{S}_2\widehat{S}_3$. In 2D, the reference element $\widehat{K}$ is the triangle defined by vertices $\widehat{S}_1$ (0,0), $\widehat{S}_2$ (1,0) and $\widehat{S}_3$ (0,1) (see fig . 2.1.3) and the reference edge $\widehat{F}$ is defined by the segment $\left[\widehat{S}_1\widehat{S}_2\right]$.

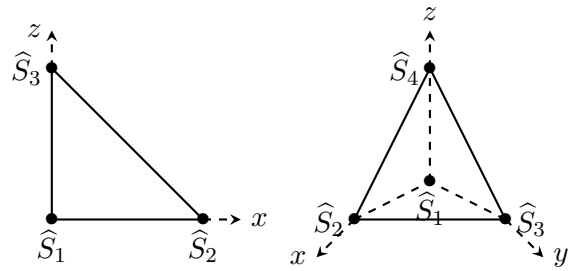


Fig. 2.1.3: Reference element $\widehat{K}$ in 2D and in 3D.

For each element K of the mesh $\mathcal{T}_h$, there exists a linear function F_K which transforms a point $\widehat{\mathbf{x}}\,(\widehat{x},\widehat{y},\widehat{z})$ in the element $\widehat{K}$ into a point $\mathbf{x}$ in the element K

$$F_K\,(\widehat{x},\widehat{y},\widehat{z}) = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} + \begin{pmatrix} x_2-x_1 & x_3-x_1 & x_4-x_1 \\ y_2-y_1 & y_3-y_1 & y_4-y_1 \\ z_2-z_1 & z_3-z_1 & z_4-z_1 \end{pmatrix} \begin{pmatrix} \widehat{x} \\ \widehat{y} \\ \widehat{z} \end{pmatrix},$$

where (x_i, y_i, z_i) are the coordinates of the vertex S_i of K. An illustration of the F_K transformation is given in fig. 2.1.4.
F_K can also be written under the vectorial form as

$$F_K\,(\widehat{x}) = A_K\widehat{x} + b_K,$$

where the jacobian J_K satisfies

$$J_K = A_K \qquad \text{and} \qquad |J_K| = |\det A_K|.$$

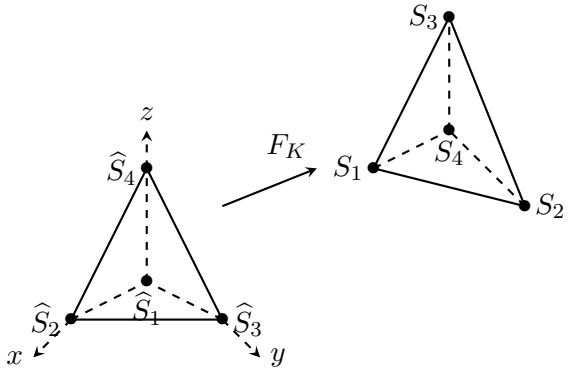


Fig. 2.1.4: Linear function F_K in 3D.

Let $\mathbb{P}_p(D)$ denote the space of polynomial functions of degree at most p on domain D.
For any element K of $\mathcal{T}_h$, we denote by $\Phi^K = \left(\varphi^K\right)_{i=1,n_K}$, a basis of $\mathbb{P}_p(K)$. The variable n_K denotes the number of degrees of freedom (ndof) and $n_K = \dfrac{(p+s)!}{p!s!}$ with s the space dimension. For example in tab. 2.1.1, we give the number of degrees of freedom in 2D and in 3D for polynomial functions until order 4.

Dimension	2D				3D			
Interpolation order	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4
Ndof by Triangle/Tetrahedron	3	6	10	15	4	10	20	35
Ndof by Edge/Face	2	3	4	5	3	6	10	15

Tab. 2.1.1: Number of degrees of freedom per element as a function of the interpolation degree.

We consider here Lagrange polynomial functions φ_i^K which are defined by:

$$\varphi_i^K \in \mathbb{P}_p, \quad \begin{cases} \varphi_i^K(a_j^K) = \delta_{ij} = \begin{cases} 1 \text{ if } i=j, & 1 \le i,j \le n_K, \\ 0 \text{ otherwise} \end{cases} \\ \varphi_i^K(a_j^{K'}) = 0 \qquad \forall K' \neq K. \end{cases} \tag{2.1.7}$$

where δ_{ij} is the Kronecker symbol, a_j^K the coordinates of the j^{th} degree of freedom (dof) of K and $a_j^{K'}$ the coordinates of the j^{th} degree of freedom of K'.

Remark. *We denote $\widehat{\varphi}_i = \varphi_i^{\widehat{K}}$ the Lagrange polynomial functions on the reference element $\widehat{K}$.*

To illustrate the definition of functions φ_i^K, we have represented them in 1D on fig. 2.1.5 for three interpolation orders.

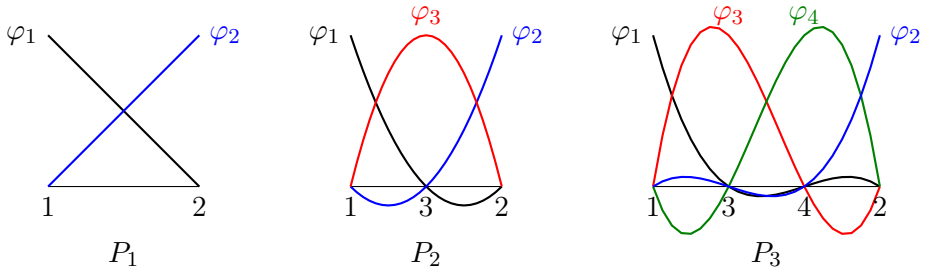


Fig. 2.1.5: 1D Lagrange polynomial functions φ_i^K.

Thanks to the fact that the support of functions φ_i^K is only contained in the element K, we deduce that the only non-zero basis functions on the element K are the one associated to the degrees of freedom of the element K. Then, using the application F_K, we express basis functions φ_i^K in terms of the basis functions $\widehat{\varphi}_i$ defined on the reference element $\widehat{K}$:

$$\varphi_i^K \in \mathbb{P}_p, \widehat{\varphi}_i \in \mathbb{P}_p, \quad \varphi_i^K = \widehat{\varphi}_i \circ F_K^{-1}, \quad 1 \leq i \leq n_K.$$

The degrees of freedom $\widehat{a}$ of the reference element $\widehat{K}$ are located at the points of coordinates:

$$(\widehat{x}_i, \widehat{y}_j, \widehat{z}_l) = \left(\frac{i-1}{p}, \frac{j-1}{p}, \frac{l-1}{p} \right), \tag{2.1.8}$$

with $i, j, l = 1, ..., p+1$ and $i \leq p+1,\ i+j \leq p+2,\ i+j+l \leq 3$.
We represent these points in 2D for four interpolation orders on fig. 2.1.6 and fig. 2.1.7 in 3D for three interpolation orders.
The location of the degrees of freedom a of the element K can be obtained by the relation:

$$a = F_K(\widehat{a}).$$

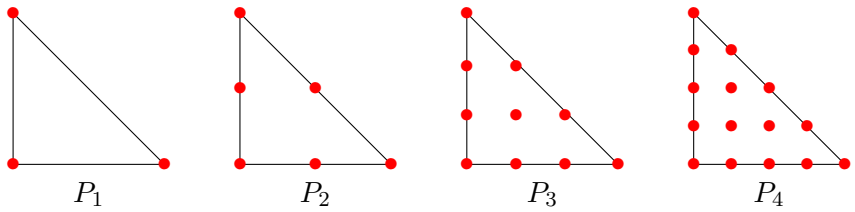


Fig. 2.1.6: Location of the degrees of freedom in 2D for four interpolation orders.

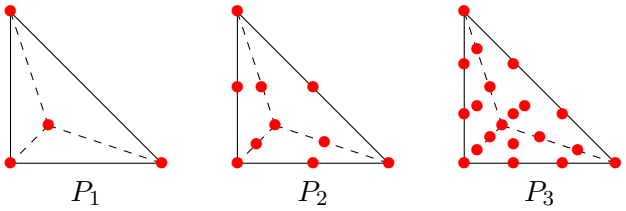


Fig. 2.1.7: Location of the degrees of freedom in 3D for three interpolation orders.

Thanks to the definition of the location of the degrees of freedom of $\widehat{K}$ (2.1.8) and the definition of the Lagrange polynomial functions (2.1.7), we deduce the expression of basis functions $\widehat{\varphi}_i$. For example, we give the expression of P_1 basis functions and P_2 basis functions:

- P_1 basis functions:

$$\begin{aligned}
\widehat{\varphi}_1(\widehat{x},\widehat{y},\widehat{z}) &= 1-\widehat{x}-\widehat{y}-\widehat{z},\\
\widehat{\varphi}_2(\widehat{x},\widehat{y},\widehat{z}) &= \widehat{x},\\
\widehat{\varphi}_3(\widehat{x},\widehat{y},\widehat{z}) &= \widehat{y},\\
\widehat{\varphi}_4(\widehat{x},\widehat{y},\widehat{z}) &= \widehat{z}.
\end{aligned}$$

- P_2 basis functions:

$$\begin{aligned}
\widehat{\varphi}_1(\widehat{x},\widehat{y},\widehat{z}) &= 2(1-\widehat{x}-\widehat{y}-\widehat{z})\left(\frac{1}{2}-\widehat{x}-\widehat{y}-\widehat{z}\right),\\
\widehat{\varphi}_2(\widehat{x},\widehat{y},\widehat{z}) &= 2\widehat{x}\left(\widehat{x}-\frac{1}{2}\right),\\
\widehat{\varphi}_3(\widehat{x},\widehat{y},\widehat{z}) &= 2\widehat{y}\left(\widehat{y}-\frac{1}{2}\right),\\
\widehat{\varphi}_4(\widehat{x},\widehat{y},\widehat{z}) &= 2\widehat{z}\left(\widehat{z}-\frac{1}{2}\right),\\
\widehat{\varphi}_5(\widehat{x},\widehat{y},\widehat{z}) &= 4\widehat{x}(1-\widehat{x}-\widehat{y}-\widehat{z}),\\
\widehat{\varphi}_6(\widehat{x},\widehat{y},\widehat{z}) &= 4\widehat{x}\widehat{y},\\
\widehat{\varphi}_7(\widehat{x},\widehat{y},\widehat{z}) &= 4\widehat{y}(1-\widehat{x}-\widehat{y}-\widehat{z}),\\
\widehat{\varphi}_8(\widehat{x},\widehat{y},\widehat{z}) &= 4\widehat{z}(1-\widehat{x}-\widehat{y}-\widehat{z}),\\
\widehat{\varphi}_9(\widehat{x},\widehat{y},\widehat{z}) &= 4\widehat{x}\widehat{z},\\
\widehat{\varphi}_{10}(\widehat{x},\widehat{y},\widehat{z}) &= 4\widehat{y}\widehat{z}.
\end{aligned}$$

We now introduce the following approximation spaces:

$$V_h = \{v_h \in L^2(\Omega) | \forall K \in \mathcal{T}_h, v_h|_K \in \mathbb{P}_p(K)\},$$

$$\mathbf{V}_h = \{\mathbf{v}_h \in \left(L^2(\Omega)\right)^3 | \forall K \in \mathcal{T}_h, \mathbf{v}_h|_K \in (\mathbb{P}_p(K))^3\},$$

$$\underline{\underline{\mathbf{\Sigma}}}_h = \{\underline{\underline{\sigma}}_h \in \left(L^2(\Omega)\right)^6 | \forall K \in \mathcal{T}_h, \underline{\underline{\sigma}}_h|_K \in (\mathbb{P}_p(K))^6\},$$

where $L^2(\Omega)$ is the space of square integrable functions on the domain Ω.
The dimension of V_h is the number of degrees of freedom associated to the mesh $\mathcal{T}_h$ i.e. $nb_{cells} \times n_K$,

the dimension of $\mathbf{V}_h$ is $3 \times nb_{cells} \times n_K$, and the dimension of $\mathbf{\Sigma}_h$ is $6 \times nb_{cells} \times n_K$. For all functions $v \in V_h$:

$$v|_K(\mathbf{x}) = \sum_{i=1}^{n_K} v(a_i)\varphi_i^K(\mathbf{x}).$$

For all vectors $\mathbf{v} \in \mathbf{V}_h$:

$$\mathbf{v}|_K(\mathbf{x}) = \sum_{i=1}^{n_K} v_x(a_i)\varphi_i^K(\mathbf{x})\mathbf{e_x} + \sum_{i=1}^{n_K} v_y(a_i)\varphi_i^K(\mathbf{x})\mathbf{e_y} + \sum_{i=1}^{n_K} v_z(a_i)\varphi_i^K(\mathbf{x})\mathbf{e_z}.$$

The space $\mathbf{\Sigma}_h$ is constructed using the same principle.

2.2 DG formulations

We first detail the general principle of the construction of DG methods. Then, we develop the expression of the numerical fluxes associated to the centered and the upwind fluxes. The last part of this section is dedicated to the processing of the boundary conditions.

2.2.1 Principles

In order to establish the DG formulations, we multiply the vectorial equation (2.1.4) of the elastodynamics system by a test-function $\Phi \in \mathbf{V}_h \times \mathbf{\Sigma}_h$ and integrate over an element K of $\mathcal{T}_h$.
The DG method seeks an approximate solution $\mathbf{W}_h = \left(\mathbf{v}_h, \underline{\underline{\sigma}}_h\right)^T$ in the space $\mathbf{V}_h \times \mathbf{\Sigma}_h$ satisfying for all K in $\mathcal{T}_h$

$$\int_K i\omega \mathbf{W}_h \cdot \Phi d\mathbf{x} + \int_K \mathbf{A}_x \frac{\partial \mathbf{W}_h}{\partial x} \cdot \Phi d\mathbf{x} + \int_K \mathbf{A}_y \frac{\partial \mathbf{W}_h}{\partial y} \cdot \Phi d\mathbf{x} + \int_K \mathbf{A}_z \frac{\partial \mathbf{W}_h}{\partial z} \cdot \Phi d\mathbf{x} = 0. \tag{2.2.1}$$

In the following, we omit the infinitesimal integral element $d\mathbf{x}$ in order to simplify the presentation. Moreover, we choose to make a classic approximation of coefficients ρ, λ and μ consideringthem as piecewise constant, i.e. constant over an element. There is no theoritical difficulty to consider polynomial variations inside one element. However, for seismic application is difficult to determine the polynomial variations of the velocity. In other words, it is more difficult to obtain the polynomial velocity model than to integrate it inside the variational formulation.
Matrices $\mathbf{A}_x, \mathbf{A}_y$ and $\mathbf{A}_z$, which depend only on these coefficients, are also piecewise-constant.
Integration by parts of (2.2.1) yields

$$\begin{aligned} \int_K i\omega \mathbf{W}_h \cdot \Phi - \int_K \mathbf{A}_x \mathbf{W}_h \cdot \frac{\partial \Phi}{\partial x} + \int_{\partial K} \mathbf{A}_x \mathbf{W}_h \cdot \Phi n_x - \int_K \mathbf{A}_y \mathbf{W}_h \cdot \frac{\partial \Phi}{\partial y} + \int_{\partial K} \mathbf{A}_y \mathbf{W}_h \cdot \Phi n_y \\ - \int_K \mathbf{A}_z \mathbf{W}_h \cdot \frac{\partial \Phi}{\partial z} + \int_{\partial K} \mathbf{A}_z \mathbf{W}_h \cdot \Phi n_z = 0. \end{aligned} \tag{2.2.2}$$

We introduce the matrix $\mathbf{D}$ such that

$$\mathbf{D}\mathbf{W}_h = \left(\mathbf{A}_x \mathbf{W}_h, \mathbf{A}_y \mathbf{W}_h, \mathbf{A}_z \mathbf{W}_h\right)^T,$$

in order to obtain

$$\begin{aligned} \mathbf{D}\mathbf{W}_h \cdot \mathbf{n} &= \mathbf{A}_x \mathbf{W}_h n_x + \mathbf{A}_y \mathbf{W}_h n_y + \mathbf{A}_z \mathbf{W}_h n_z \\ &= \mathbf{D_n} \mathbf{W}_h, \end{aligned}$$

where $\mathbf{D_n}$ is defined such that

$$\mathbf{D_n} = (\mathbf{A}_x n_x + \mathbf{A}_y n_y + \mathbf{A}_z n_z)\,.$$

We rewrite (2.2.2) as

$$\int_K i\omega \mathbf{W}_h \cdot \Phi - \int_K \mathbf{A}_x \mathbf{W}_h \cdot \frac{\partial \Phi}{\partial x} - \int_K \mathbf{A}_y \mathbf{W}_h \cdot \frac{\partial \Phi}{\partial y} - \int_K \mathbf{A}_z \mathbf{W}_h \cdot \frac{\partial \Phi}{\partial z} + \int_{\partial K} (\mathbf{D_n} \mathbf{W}_h) \cdot \Phi = 0. \quad (2.2.3)$$

By summing equations (2.2.3) over all elements, we obtain the following global equation

$$\begin{aligned} \sum_{K\in\mathcal{T}_h} \int_K i\omega \mathbf{W}_h \cdot \Phi - \sum_{K\in\mathcal{T}_h} \int_K \mathbf{A}_x \mathbf{W}_h \cdot \frac{\partial \Phi}{\partial x} - \sum_{K\in\mathcal{T}_h} \int_K \mathbf{A}_y \mathbf{W}_h \cdot \frac{\partial \Phi}{\partial y} - \sum_{K\in\mathcal{T}_h} \int_K \mathbf{A}_z \mathbf{W}_h \cdot \frac{\partial \Phi}{\partial z} \\ + \sum_{K\in\mathcal{T}_h} \int_{\partial K} (\mathbf{D_n} \mathbf{W}_h) \cdot \Phi = 0. \end{aligned} \quad (2.2.4)$$

We have to pay attention to the numerical treatment of the term $(\mathbf{D_n}\mathbf{W}_h)$ since $\mathbf{W}_h$ is discontinuous across an element boundary. As we have illustrated in 1D on fig.2.2.1, at the boundary ∂K the physical values associated to the "right" and the "left" elements are different. We thus define the *jumps*.

Definition. *We define the jump* $[\![\cdot]\!]$ *of a vector* $\mathbf{u}$*:*

- *On a face* $F \in \mathcal{F}_i$ *as*

$$[\![\mathbf{u}]\!] = \mathbf{u}^{K^+} \cdot \mathbf{n}^+ + \mathbf{u}^{K^-} \cdot \mathbf{n}^- = \mathbf{u}^{K^+} \cdot \mathbf{n}^+ - \mathbf{u}^{K^-} \cdot \mathbf{n}^+, \text{ with } \mathbf{n}^+ = -\mathbf{n}^-,$$

- *On a face* $F \in \mathcal{F}_b$ *as*

$$[\![\mathbf{u}]\!] = \mathbf{u}^K \cdot \mathbf{n},$$

The jump of a tensor $\underline{\underline{\sigma}}$ *is:*

- *On a face* $F \in \mathcal{F}_i$

$$[\![\underline{\underline{\sigma}}]\!] = \underline{\underline{\sigma}}^{K^+} \mathbf{n}^+ + \underline{\underline{\sigma}}^{K^-} \mathbf{n}^- = \underline{\underline{\sigma}}^{K^+} \mathbf{n}^+ - \underline{\underline{\sigma}}^{K^-} \mathbf{n}^+.$$

- *On a face* $F \in \mathcal{F}_b$

$$[\![\underline{\underline{\sigma}}]\!] = \underline{\underline{\sigma}}^K \mathbf{n}.$$

From this definition, we denote

$$[\![\mathbf{W}]\!] = \begin{bmatrix} [\![\mathbf{v}]\!] \\ [\![\underline{\underline{\sigma}}]\!] \end{bmatrix}.$$

‘

Remark. *To ensure the continuity of the solution, if* $\left(\mathbf{v}, \underline{\underline{\sigma}}\right)$ *is solution of* (2.1.1)*, then*

$$[\![\mathbf{v}]\!] = 0 \text{ and } [\![\underline{\underline{\sigma}}]\!] = 0.$$

So,

$$[\![\mathbf{W}]\!] = 0.$$

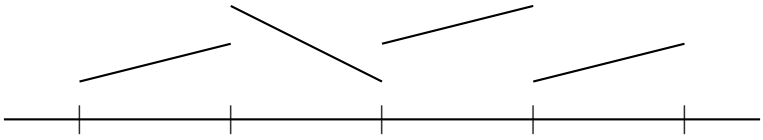


Fig. 2.2.1: Discontinuous function in 1D.

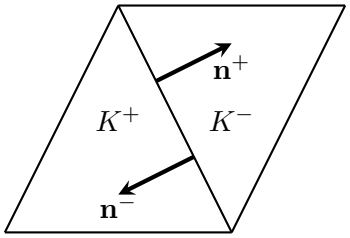


Fig. 2.2.2: Definition of the jump

Taking this into account, we get:

$$\sum_{K\in\mathcal{T}_h}\int_{\partial K}(\mathbf{D_n}\mathbf{W}_h)\cdot\Phi=\sum_{F\in\mathcal{F}_h}\int_F[\![(\mathbf{D_n}\mathbf{W}_h)\cdot\Phi]\!]=\sum_{F\in\mathcal{F}_b}\int_F(\mathbf{D_n}\mathbf{W}_h)\cdot\Phi+\sum_{F\in\mathcal{F}_i}\int_F\left(\left(\mathbf{D}^+_{\mathbf{n}^+}\mathbf{W}^+_h\right)\cdot\Phi^+ + \left(\mathbf{D}^-_{\mathbf{n}^-}\mathbf{W}^-_h\right)\cdot\Phi^-\right.$$

and using the definition of the jump, we write:

$$\sum_{K\in\mathcal{T}_h}\int_{\partial K}(\mathbf{D_n}\mathbf{W}_h)\cdot\Phi=\sum_{F\in\mathcal{F}_h}\int_F[\![(\mathbf{D}\mathbf{W}_h)\cdot\Phi]\!].$$

The global DG formulation is finally given by

$$\begin{aligned}\sum_{K\in\mathcal{T}_h}\int_K i\omega\mathbf{W}_h\cdot\Phi-\sum_{K\in\mathcal{T}_h}\int_K\mathbf{A}_x\mathbf{W}_h\cdot\frac{\partial\Phi}{\partial x}-\sum_{K\in\mathcal{T}_h}\int_K\mathbf{A}_y\mathbf{W}_h\cdot\frac{\partial\Phi}{\partial y}-\sum_{K\in\mathcal{T}_h}\int_K\mathbf{A}_z\mathbf{W}_h\cdot\frac{\partial\Phi}{\partial z}\\+\sum_{F\in\mathcal{F}_h}\int_F[\![(\mathbf{D}\mathbf{W}_h)\cdot\Phi]\!]=0.\end{aligned}\tag{2.2.5}$$

As we use basis functions Φ whose support is in the element K, equation (2.2.5) can be reduced to one equation per element K. The surface integral over K's boundary is written as the sum over all K faces

$$\int_{\partial K}(\mathbf{D_n}\mathbf{W}_h)\cdot\Phi=\sum_{F\in\mathcal{F}(K)}(\mathbf{D_n}\mathbf{W}_h)_F\cdot\Phi.$$

Moreover, as matrices $\mathbf{A}_x,\mathbf{A}_y$ and $\mathbf{A}_z$ are piecewise constant, we obtain the following local equation

$$\begin{aligned}\int_K i\omega\mathbf{W}^K_h\cdot\Phi^K-\int_K\mathbf{A}^K_x\mathbf{W}^K_h\cdot\frac{\partial\Phi^K}{\partial x}-\int_K\mathbf{A}^K_y\mathbf{W}^K_h\cdot\frac{\partial\Phi^K}{\partial y}-\int_K\mathbf{A}^K_z\mathbf{W}^K_h\cdot\frac{\partial\Phi^K}{\partial z}\\+\sum_{F\in\mathcal{F}(K)}\int_F(\mathbf{D_n}\mathbf{W}_h)\,|_F\cdot\Phi^K=0.\end{aligned}\tag{2.2.6}$$

This is the generic formulation of a DG method and, as we said previously, the difference between the methods lie in the expression of the fluxes. To compute the numerical trace $(\mathbf{D_n}\mathbf{W}_h)\,|_F$ in equation (2.2.6), we will consider two options: a centered scheme and an upwind scheme.

2.2.2 Centered flux DG scheme

To establish the centered flux DG formulation, we approximate $(\mathbf{D_n W}_h)\,|_F$ on a face F by its average.

Definition. *We define the mean (average) $\{\cdot\}$ of a variable u such as:*

- *On a face $F \in \mathcal{F}_i$*
$$\{u\} = \frac{u^K + u^{K'}}{2},$$
- *On a face $F \in \mathcal{F}_b$*
$$\{u\} = u^K.$$

The average of a vector $\mathbf{u}$ is

- *On a face $F \in \mathcal{F}_i$*
$$\{\mathbf{u}\} = \frac{\mathbf{u}^K + \mathbf{u}^{K'}}{2}.$$
- *On a face $F \in \mathcal{F}_b$*
$$\{\mathbf{u}\} = \mathbf{u}^K.$$

Using the definition of the average of a vector for $(\mathbf{D_n W}_h)\,|_F$, the local equation (2.2.6) becomes

$$\begin{aligned}\int_K i\omega \mathbf{W}_h^K \cdot \Phi^K - \int_K \mathbf{A}_x^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial x} - \int_K \mathbf{A}_y^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial y} - \int_K \mathbf{A}_z^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial z} \\ + \sum_{F \in \partial K \cap \partial K'} \int_F \frac{1}{2}\left(\mathbf{D}_{\mathbf{n}}^K \mathbf{W}_h^K + \mathbf{D}_{\mathbf{n}'}^{K'} \mathbf{W}_h^{K'}\right) \cdot \Phi^K \\ + \sum_{F \in \partial K \cap \partial \Omega} \int_F \left(\mathbf{D_n}^K \mathbf{W}_h^K\right) \cdot \Phi^K = 0.\end{aligned} \tag{2.2.7}$$

We easily verify that summing the centered flux local equation over all the elements yields the expected global DG equation

$$\begin{aligned}\sum_{K \in \mathcal{T}_h} \int_K i\omega \mathbf{W}_h^K \cdot \Phi^K - \sum_{K \in \mathcal{T}_h} \int_K \mathbf{A}_x^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial x} - \sum_{K \in \mathcal{T}_h} \int_K \mathbf{A}_y^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial y} - \sum_{K \in \mathcal{T}_h} \int_K \mathbf{A}_z^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial z} \\ + \sum_{F \in \mathcal{F}_h} \int_F \{\mathbf{D_n W}_h\} [\![\Phi]\!] = 0.\end{aligned}$$

Property (Relation between the average and the jump)**.** *Let $\mathbf{u}$ and $\mathbf{v}$ be two vectors, we have:*

$$[\![\mathbf{u} \cdot \mathbf{v}]\!] = \{\mathbf{u}\}[\![\mathbf{v}]\!] + [\![\mathbf{u}]\!]\{\mathbf{v}\}.$$

Proof.

$$\begin{aligned}\{\mathbf{u}\}[\![\mathbf{v}]\!] + [\![\mathbf{u}]\!]\{\mathbf{v}\} &= \frac{1}{2}\left(\mathbf{u} + \mathbf{u}'\right) \cdot \left(\mathbf{v} \cdot \mathbf{n} + \mathbf{v}' \cdot \mathbf{n}'\right) + \frac{1}{2}\left(\mathbf{u} \cdot \mathbf{n} + \mathbf{u}' \cdot \mathbf{n}'\right) \cdot \left(\mathbf{v} + \mathbf{v}'\right) \\ &= \frac{1}{2}\left(\mathbf{u} + \mathbf{u}'\right) \cdot \left(\mathbf{v} - \mathbf{v}'\right) \cdot \mathbf{n} + \frac{1}{2}\left(\mathbf{u} - \mathbf{u}'\right) \cdot \left(\mathbf{v} + \mathbf{v}'\right) \cdot \mathbf{n} \\ &= \left(\mathbf{u} \cdot \mathbf{v} - \mathbf{u}' \cdot \mathbf{v}'\right) \cdot \mathbf{n} \\ &= [\![\mathbf{u} \cdot \mathbf{v}]\!].\end{aligned}$$

□

The fact that $\mathbf{W}_h$ is solution of the elastodynamics equations (2.1.1) implies:

$$[\![\mathbf{W}_h]\!] = 0,$$

where $[\![\mathbf{W}_h]\!]$ gathers $[\![\mathbf{v}_h]\!]$ and $[\![\underline{\underline{\sigma}}_h]\!]$. Thus, if we multiply $\mathbf{W}_h$ by a constant matrix $\mathbf{D}$

$$[\![\mathbf{D}\mathbf{W}_h]\!] = 0,$$

and we can deduce that

$$\int_{\mathcal{F}_h} [\![(\mathbf{D}\mathbf{W}_h)\cdot\Phi]\!] = \sum_{F\in\mathcal{F}_h}\int_F \{\mathbf{D_n}\mathbf{W}_h\}[\![\Phi]\!] + \sum_{F\in\mathcal{F}_h}\int_F [\![\mathbf{D}\mathbf{W}_h]\!]\{\Phi\} = \int_{\mathcal{F}_h}\{\mathbf{D_n}\mathbf{W}_h\}[\![\Phi]\!].$$

we remind that $\mathbf{n}$ is included in the definition of the jump $[\![\cdot]\!]$.
Finally, replacing the surface integral, we recover the global equation (2.2.5).

Discretization

In equation (2.2.7), as $\mathbf{W}_h^K \in \mathbf{V}_h \times \mathbf{\Sigma}_h$, each component of $\mathbf{W}_h^K$ can be decomposed in the basis $\Phi^K = \left(\varphi^K\right)_i$:

$$\mathbf{W}_h^K = \sum_{j=1}^{n_K} \mathbf{W}_j^K \varphi_j^K, \tag{2.2.8}$$

where

$$\mathbf{W}_j^K = \left(v_{x_j}, v_{y_j}, v_{z_j}, \sigma_{xx_j}, \sigma_{xx_j}, \sigma_{yy_j}, \sigma_{zz_j}, \sigma_{xy_j}, \sigma_{yz_j}, \sigma_{xz_j}\right)^T.$$

Choosing the test function as $\varphi_l^K \in \Phi^K$, $l = 1, n_K$, we obtain the following local discretization

$$\begin{aligned} i\omega \sum_{j=1}^{n_K} \mathbf{W}_j^K \int_K \varphi_j^K \varphi_l^K &- \mathbf{A}_x^K \sum_{j=1}^{n_K} \mathbf{W}_j^K \int_K \varphi_j^K \frac{\partial \varphi_l^K}{\partial x} \\ &- \mathbf{A}_y^K \sum_{j=1}^{n_K} \mathbf{W}_j^K \int_K \varphi_j^K \frac{\partial \varphi_l^K}{\partial y} - \mathbf{A}_z^K \sum_{j=1}^{n_K} \mathbf{W}_j^K \int_K \varphi_j^K \frac{\partial \varphi_l^K}{\partial z} \\ &+ \sum_{F\in\partial K\cap\partial K'} \frac{1}{2}\left(\mathbf{D}_{\mathbf{n}}^K \sum_{j=1}^{n_K} \mathbf{W}_j^K \int_F \varphi_j^K \varphi_l^K + \mathbf{D}_{\mathbf{n}}^{K'} \sum_{j=1}^{n_K} \mathbf{W}_j^{K'} \int_F \varphi_j^{K'} \varphi_l^K\right) \\ &+ \sum_{F\in\partial K\cap\partial\Omega} \int_F \left(\mathbf{D}_{\mathbf{n}}^K \mathbf{W}_h^K\right)\cdot\Phi^K = 0. \end{aligned} \tag{2.2.9}$$

Let us introduce the following additional notations

$$\begin{aligned} \underline{\mathbf{W}}^K &= \left(\mathbf{W}_1^K, ..., \mathbf{W}_{d_i}^K\right)^T, \\ \mathcal{M}^K(i,j) &= \int_K \varphi_j^K \varphi_i^K, \\ \mathcal{D}_u^K(i,j) &= \int_K \varphi_j^K \frac{\partial \varphi_i^K}{\partial u}, \text{ with } u = x, y, z, \\ \mathcal{E}_1^K(i,j) &= \int_F \varphi_j^K \varphi_i^K, \\ \mathcal{E}_2^{KK'}(i,j) &= \int_F \varphi_j^{K'} \varphi_i^K. \end{aligned} \tag{2.2.10}$$

We have seen in section 3.1.1 that φ_i^K can be expressed as a function of $\widehat{\varphi}_i$:

$$\varphi_i^K(\mathbf{x}) = \widehat{\varphi}_i(F_K^{-1}(\mathbf{x})).$$

Using the change of variables: $\mathbf{x} = F_K(\widehat{\mathbf{x}})$, we get:

$$\begin{aligned}
&\mathcal{M}^K(i,j) = |detJ_{F_K}| \int_{\widehat{K}} \widehat{\varphi}_j \widehat{\varphi}_i, \\
&\mathcal{D}_u^K(i,j) = |detJ_{F_K}| \left(J_{F_K}^{-T} \int_{\widehat{K}} \widehat{\varphi}_j \nabla \widehat{\varphi}_i \right) \cdot \mathbf{e_u}, \text{ with } u = x, y, z, \text{ and } (\mathbf{e_u})_{m=1,3} = \delta_{um}, \\
&\mathcal{E}_1^K(i,j) = |detJ_{F_F}| \int_{\widehat{F}} \widehat{\varphi}_j \widehat{\varphi}_i, \\
&\mathcal{E}_2^{KK'}(i,j) = |detJ_{F_F}| \int_{\widehat{F}} \widehat{\varphi}'_j \widehat{\varphi}_i.
\end{aligned} \tag{2.2.11}$$

Thus, to compute the above matrices, we only have to compute the integrals over the reference element, the Jacobian J_{F_K} and the coefficients $|detJ_{F_K}|$ and $|detJ_{F_F}|$ which greatly simplifies the computation. Finally, we write the local discretization into the matrix form:

$$\begin{aligned}
i\omega \mathcal{M}^K \underline{\mathbf{W}}^K &- \mathbf{A}_x^K \mathcal{D}_x^K \underline{\mathbf{W}}^K - \mathbf{A}_y^K \mathcal{D}_y^K \underline{\mathbf{W}}^K - \mathbf{A}_z^K \mathcal{D}_z^K \underline{\mathbf{W}}^K \\
&+ \sum_{F \in \partial K \cap \partial K'} \frac{1}{2} \left(\mathbf{D}_{\mathbf{n}}^K \mathcal{E}_1^K \underline{\mathbf{W}}^K + \mathbf{D}_{\mathbf{n}}^{K'} \mathcal{E}_2^{KK'} \underline{\mathbf{W}}^{K'} \right) \\
&+ \sum_{F \in \partial K \cap \partial \Omega} \int_F \left(\mathbf{D}_{\mathbf{n}}^K \mathbf{W}_h^K \right) \cdot \Phi^K = 0,
\end{aligned} \tag{2.2.12}$$

Remark. *We have not developed the boundary term* $\sum_{F \in \partial K \cap \partial \Omega} \int_F \left(\mathbf{D}_{\mathbf{n}}^K \mathbf{W}_h^K \right) \cdot \Phi^K$. *We will detail the treatment of this term in section 2.2.4.*

By gathering all the local equations (2.2.12) of all elements, the global equation (2.2.5) can be written as

$$(\mathcal{M} + \mathcal{R})\, \underline{\mathcal{W}} + \sum_{K \in \mathcal{T}_h} \sum_{F \in \partial K \cap \partial \Omega} \int_F \left(\mathbf{D}_{\mathbf{n}}^K \mathbf{W}_h^K \right) \cdot \Phi^K = 0, \tag{2.2.13}$$

where $\underline{\mathcal{W}} = \sum_{K \in \mathcal{T}_h} \underline{\mathbf{W}}^K$, and $\mathcal{M}$ and $\mathcal{R}$ are defined such as

$$\mathcal{M}^{KL} = \begin{cases} \mathcal{M}^K, & \text{if } K = L \\ 0, & \text{otherwise,} \end{cases}$$

$$\mathcal{R}^{KL} = \begin{cases} -\mathbf{A}_x^K \mathcal{D}_x^K - \mathbf{A}_y^K \mathcal{D}_y^K - \mathbf{A}_z^K \mathcal{D}_z^K + \sum_{F \in \partial K \cap \partial K'} \frac{1}{2} \left(\mathbf{D}_{\mathbf{n}}^K \mathcal{E}_1^K \right), & \text{if } K = L \\ \sum_{F \in \partial K \cap \partial K'} \frac{1}{2} \left(\mathbf{D}_{\mathbf{n}}^{K'} \mathcal{E}_2^{KK'} \right), & \text{otherwise.} \end{cases}$$

Remark. *The global matrix* $\mathcal{M}$ *is block diagonal.*

2.2.3 Upwind flux DG scheme

We recall the local DG equation (2.2.6):

$$\int_K i\omega \mathbf{W}_h^K \cdot \Phi^K - \int_K \mathbf{A}_x^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial x} - \int_K \mathbf{A}_y^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial y} - \int_K \mathbf{A}_z^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial z} + \sum_{F \in \mathcal{F}(K)} \int_F \left(\mathbf{D_n} \mathbf{W}_h\right)|_F \cdot \Phi^K = 0.$$

As we said before, the two DG formulations considered in this study differ in the expression of the flux $\left(\mathbf{D_n} \mathbf{W}_h\right)|_F$ across the interface F.
We first study the properties of matrix $(\mathbf{D_n})$. It is a square matrix which can be diagonalized. The corresponding eigenvalues are

$$\xi_1 = -v_p,\ \xi_2 = \xi_3 = -v_s,\ \xi_4 = \xi_5 = \xi_6 = 0,\ \xi_7 = \xi_8 = v_s,\ \xi_9 = v_p,$$

where $v_p = \sqrt{\dfrac{\lambda + 2\mu}{\rho}}$ and $v_s = \sqrt{\dfrac{\mu}{\rho}}$. The associated eigenvectors are gathered in the 9×9 matrix $\mathbf{R_n}$, where kth column is the eigenvector associated to the eigenvalue ξ_k

$$\mathbf{R_n} = \begin{pmatrix} n_x\, v_p & n_z\, v_s & 0 & 0 & 0 & 0 & \cdots \\ n_y\, v_p & 0 & n_z\, v_s & 0 & 0 & 0 & \cdots \\ n_z\, v_p & -n_x\, v_s & -n_y\, v_s & 0 & 0 & 0 & \cdots \\ \lambda + 2\,\mu\, n_x^2 & 2\,\mu\, n_x\, n_z & 0 & 2\, n_y\, n_z & 0 & 0 & \cdots \\ \lambda + 2\,\mu\, n_y^2 & 0 & 2\,\mu\, n_y\, n_z & 0 & 2\, n_x\, n_z & 0 & \cdots \\ 2\,\mu\, n_z^2 + \lambda & -2\,\mu\, n_x\, n_z & -2\,\mu\, n_y\, n_z & 0 & 0 & 2\, n_x\, n_y & \cdots \\ 2\,\mu\, n_y\, n_x & \mu\, n_y\, n_z & \mu\, n_x\, n_z & -n_x\, n_z & -n_y\, n_z & n_z^2 & \cdots \\ 2\,\mu\, n_y\, n_z & -\mu\, n_x\, n_y & \mu\, \left(n_z^2 - n_y^2\right) & n_x^2 & -n_y\, n_x & -n_z\, n_x & \cdots \\ 2\,\mu\, n_z\, n_x & \mu\, \left(n_z^2 - n_x^2\right) & -\mu\, n_x\, n_y & -n_x\, n_y & {n_y}^2 & -n_z\, n_y & \cdots \end{pmatrix},$$

$$\begin{pmatrix} \cdots & 0 & -n_z\, v_s & -n_x\, v_p \\ \cdots & -n_z\, v_s & 0 & -n_y\, v_p \\ \cdots & n_y\, v_s & n_x\, v_s & -n_z\, v_p \\ \cdots & 0 & 2\,\mu\, n_x\, n_z & \lambda + 2\,\mu\, n_x^2 \\ \cdots & 2\,\mu\, n_y\, n_z & 0 & \lambda + 2\,\mu\, n_y^2 \\ \cdots & -2\,\mu\, n_y\, n_z & -2\,\mu\, n_x\, n_z & 2\,\mu\, n_z^2 + \lambda \\ \cdots & \mu\, n_x\, n_z & \mu\, n_y\, n_z & 2\,\mu\, n_y\, n_x \\ \cdots & \mu\, \left(n_z^2 - n_y^2\right) & -\mu\, n_x\, n_y & 2\,\mu\, n_y\, n_z \\ \cdots & -\mu\, n_x\, n_y & \mu\, \left(n_z^2 - n_x^2\right) & 2\,\mu\, n_z\, n_x \end{pmatrix}.$$

Remark. *The eigenvalues and eigenvectors can be computed analytically because we consider isotropic media. Their expression is much more complicated for anisotropic media and we refer to [54] for more details on the anisotropic formulation.*

Definition. *Let $\mathbf{D_n}$ be a diagonalizable matrix whose eigenvectors are gathered in matrix $\mathbf{R_n}$, positive eigenvalues in diagonal matrix $\mathbf{\Lambda}^+$ and negative eigenvalues in $\mathbf{\Lambda}^-$.*
We define $\mathbf{D_n^+}$ such that

$$\mathbf{D_n^+} = \mathbf{R_n} \mathbf{\Lambda}^+ \left(\mathbf{R_n}\right)^{-1},$$

and $\mathbf{D_n^-}$ such that

$$\mathbf{D_n^-} = \mathbf{R_n} \mathbf{\Lambda}^- \left(\mathbf{R_n}\right)^{-1}.$$

Property. *Using the above definitions, we have*

$$(\mathbf{D_n}) = \mathbf{D}_\mathbf{n}^+ + \mathbf{D}_\mathbf{n}^- \quad \text{and} \quad |(\mathbf{D_n})| = \mathbf{D}_\mathbf{n}^+ - \mathbf{D}_\mathbf{n}^-. \tag{2.2.14}$$

For the upwind flux DG formulation, we define the numerical trace $((\mathbf{D_n})\,\mathbf{W}_h)\,|_F$ on a face F as

$$((\mathbf{D_n})\,\mathbf{W}_h)\,|_F = \left(\mathbf{D}_\mathbf{n}^K\right)^+ \mathbf{W}_h^K + \left(\mathbf{D}_\mathbf{n}^{K'}\right)^- \mathbf{W}_h^{K'}. \tag{2.2.15}$$

In equation (2.2.6) we replace the surface term by its approximation for an interior face, we obtain

$$\begin{aligned}\int_K i\omega \mathbf{W}_h^K \cdot \Phi^K - \int_K \mathbf{A}_x^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial x} - \int_K \mathbf{A}_y^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial y} - \int_K \mathbf{A}_z^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial z} \\ + \sum_{F\in\partial K\cap\partial K'} \int_F \left[\left(\mathbf{D}_\mathbf{n}^K\right)^+ \mathbf{W}_h^K + \left(\mathbf{D}_\mathbf{n}^{K'}\right)^- \mathbf{W}_h^{K'}\right] \cdot \Phi^K \\ + \sum_{F\in\partial K\cap\partial\Omega} \int_F \left(\mathbf{D}_\mathbf{n}^K \mathbf{W}_h^K\right) \cdot \Phi^K = 0.\end{aligned}$$

Summing over all the elements, we get

$$\begin{aligned}\sum_{K\in\mathcal{T}_h} \int_K i\omega \mathbf{W}_h^K \cdot \Phi^K - \sum_{K\in\mathcal{T}_h} \int_K \mathbf{A}_x^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial x} - \sum_{K\in\mathcal{T}_h} \int_K \mathbf{A}_y^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial y} - \sum_{K\in\mathcal{T}_h} \int_K \mathbf{A}_z^K \mathbf{W}_h^K \cdot \frac{\partial \Phi^K}{\partial z} \\ + \sum_{F\in\mathcal{F}_i} \int_F \left(\mathbf{D}_\mathbf{n}^K\right)^+ \mathbf{W}_h^K \cdot \Phi^K + \sum_{F\in\mathcal{F}_i} \int_F \left(\mathbf{D}_\mathbf{n}^K\right)^- \mathbf{W}_h^K \cdot \Phi^K \\ + \sum_{F\in\mathcal{F}_b} \int_F \left(\mathbf{D}_\mathbf{n}^K \mathbf{W}_h^K\right) \cdot \Phi^K = 0.\end{aligned}$$

and we recover equation (2.2.5). Indeed the plus-minus decomposition (2.2.15) of $(\mathbf{D_n W}_h)$ can be seen as the decomposition into the inflow flux $\left(\mathbf{D}_\mathbf{n}^K\right)^- \mathbf{W}_h^K$ and the outflow flux $\left(\mathbf{D}_\mathbf{n}^K\right)^+ \mathbf{W}_h^K$, and when we sum these two terms we recover the jump of $(\mathbf{D_n W}_h)$ on the face F.

Discretization

Using the decomposition (2.2.8) of $\mathbf{W}_h^K$ in the basis Φ^K and taking, as test-function, $\varphi_l^K, l = 1, n_K$, we obtain the local equation for the upwind DG scheme

$$\begin{aligned}i\omega \sum_{j=1}^{n_K} \mathbf{W}_j^K \int_K \varphi_j^K \varphi_l^K - \mathbf{A}_x^K \sum_{j=1}^{n_K} \mathbf{W}_j^K \int_K \varphi_j^K \frac{\partial \varphi_l^K}{\partial x} - \mathbf{A}_y^K \sum_{j=1}^{n_K} \mathbf{W}_j^K \int_K \varphi_j^K \frac{\partial \varphi_l^K}{\partial y} - \mathbf{A}_z^K \sum_{j=1}^{n_K} \mathbf{W}_j^K \int_K \varphi_j^K \frac{\partial \varphi_l^K}{\partial z} \\ + \sum_{F\in\partial K\cap\partial K'} \left(\left(\mathbf{D}_\mathbf{n}^K\right)^+ \sum_{j=1}^{n_K} \mathbf{W}_j^K \int_F \varphi_j^K \varphi_l^K + \left(\mathbf{D}_\mathbf{n}^{K'}\right)^- \sum_{j=1}^{n_K} \mathbf{W}_j^{K'} \int_F \varphi_j^{K'} \varphi_l^K \right) \\ + \sum_{F\in\partial K\cap\partial\Omega} \int_F \left(\mathbf{D}_\mathbf{n}^K \mathbf{W}_h^K\right) \cdot \Phi^K = 0.\end{aligned} \tag{2.2.16}$$

Similarly to the centered scheme, we write the above equation in a matrix form

$$\begin{aligned}i\omega \mathcal{M}^K \underline{\mathbf{W}}^K - \mathbf{A}_x^K \mathcal{D}_x^K \underline{\mathbf{W}}^K - \mathbf{A}_y^K \mathcal{D}_y^K \underline{\mathbf{W}}^K - \mathbf{A}_z^K \mathcal{D}_z^K \underline{\mathbf{W}}^K \\ + \sum_{F\in\partial K\cap\partial K'} \left(\left(\mathbf{D}_\mathbf{n}^K\right)^+ \mathcal{E}_1^K \underline{\mathbf{W}}^K + \left(\mathbf{D}_\mathbf{n}^{K'}\right)^- \mathcal{E}_2^{KK'} \underline{\mathbf{W}}^{K'} \right) \\ + \sum_{F\in\partial K\cap\partial\Omega} \int_F \left(\mathbf{D}_\mathbf{n}^K \mathbf{W}_h^K\right) \cdot \Phi^K = 0,\end{aligned} \tag{2.2.17}$$

where matrices $\mathcal{M}^K, \mathcal{D}_u^K, u = x, y, z, \mathcal{E}_1^K$ and $\mathcal{E}_2^{KK'}$ are defined by (2.2.11).
Finally, the global equation for the upwind DG scheme in a matrix form is

$$(\mathcal{M} + \mathcal{R})\,\underline{\mathcal{W}} + \sum_{K \in \mathcal{T}_h} \sum_{F \in \partial K \cap \partial\Omega} \int_F \left(\mathbf{D}_{\mathbf{n}}^K \mathbf{W}_h^K\right) \cdot \Phi^K = 0, \tag{2.2.18}$$

where $\underline{\mathcal{W}} = \sum_{K \in \mathcal{T}_h} \underline{\mathbf{W}}^K$, and $\mathcal{M}$ and $\mathcal{R}$ are defined in this case such as

$$\mathcal{M}^{KL} = \begin{cases} \mathcal{M}^K, & \text{if } K = L \\ 0, & \text{otherwise,} \end{cases}$$

$$\mathcal{R}^{KL} = \begin{cases} -\mathbf{A}_x^K \mathcal{D}_x^K - \mathbf{A}_y^K \mathcal{D}_y^K - \mathbf{A}_z^K \mathcal{D}_z^K + \displaystyle\sum_{F \in \partial K \cap \partial K'} \left(\mathbf{D}_{\mathbf{n}}^K\right)^+ \mathcal{E}_1^K, & \text{if } K = L \\ \displaystyle\sum_{F \in \partial K \cap \partial K'} \left(\mathbf{D}_{\mathbf{n}}^{K'}\right)^- \mathcal{E}_2^{KK'}, & \text{otherwise.} \end{cases}$$

2.2.4 Boundary conditions

We focus here on the numerical treatment of the integral term over the boundary of the computational domain,

$$\sum_{F \in \partial K \cap \partial\Omega} \int_F \left(\mathbf{D}_{\mathbf{n}}^K \mathbf{W}_h^K\right) \cdot \Phi^K.$$

Two boundary conditions have been introduced in the problem statement in section 2.1.1:

- A free surface condition to simulate a physical interface between the domain and the air;
- An absorbing boundary condition to simulate an infinite domain.

Free surface condition

On a free surface boundary we impose

$$\underline{\underline{\sigma}}\mathbf{n} = 0, \tag{2.2.19}$$

and no particular condition is applied to the velocity components. We then have

$$\int_{F_b} \left(\mathbf{D}_{\mathbf{n}}^K \mathbf{W}_h^K\right) \cdot \Phi^K = \int_{F_b} \mathbf{L}^K \mathbf{W}_h^K \cdot \Phi^K,$$

where $\mathbf{L}^K = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ (\lambda+2\mu)\,n_x & \lambda n_y & \lambda n_z & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda n_x & (\lambda+2\mu)\,n_y & \lambda n_z & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda n_x & \lambda n_y & (\lambda+2\mu)\,n_z & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu n_y & \mu n_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mu n_z & \mu n_y & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu n_z & 0 & \mu n_x & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$.

In the matrix form, the free surface boundary condition is written

$$\mathbf{L}^K \underline{\mathbf{W}}^K \mathcal{B}^K.$$

Absorbing boundary condition

In section 1.1.5, we have introduced two kinds of absorbing boundary condition. For the two DG schemes studied in this chapter, we consider the plus-minus decomposition of the flux matrix (2.2.14). We define the simple treatment which consists in considering that the inflow flux is zero for each face on the artificial boundary, i.e.

$$\left(\mathbf{D}_{\mathbf{n}}\right)^{-}\mathbf{W}=0 \text{ on } \mathcal{F}_b. \tag{2.2.20}$$

Then

$$\begin{aligned}\int_{F_b}\left(\mathbf{D}_{\mathbf{n}}^K\mathbf{W}_h^K\right)\cdot\Phi^K &\simeq \int_{F_b}\left(\mathbf{D}_{\mathbf{n}}^K\right)^{+}\mathbf{W}_h^K\cdot\Phi^K\\ &\simeq \left(\mathbf{D}_{\mathbf{n}}^K\right)^{+}\sum_{l,j=1}^{n_K}\mathbf{W}_j^K\int_{F_b}\varphi_j^K\varphi_l^K=\left(\mathbf{D}_{\mathbf{n}}^K\right)^{+}\underline{\mathbf{W}}^K\mathcal{B}^K.\end{aligned}$$

where $\mathcal{B}^K(l,j)=\displaystyle\int_{F_b}\varphi_j^K\varphi_l^K=|detF_{F_b}|\int_{\widehat{F}}\widehat{\varphi}_j\widehat{\varphi}_l$.

Finally, the global discrete system reads as

$$\begin{aligned}i\omega\sum_{K\in\mathcal{T}_h}\mathcal{M}^K\underline{\mathbf{W}}^K&-\sum_{K\in\mathcal{T}_h}\mathbf{A}_x^K\mathcal{D}_x^K\underline{\mathbf{W}}^K-\sum_{K\in\mathcal{T}_h}\mathbf{A}_y^K\mathcal{D}_y^K\underline{\mathbf{W}}^K-\sum_{K\in\mathcal{T}_h}\mathbf{A}_z^K\mathcal{D}_z^K\underline{\mathbf{W}}^K\\&+\sum_{K\in\mathcal{T}_h}\sum_{F\in\partial K\cap\partial K'}\left(\mathbf{H}_{\mathbf{n}}^K\mathcal{E}_1^K\underline{\mathbf{W}}^K+\mathbf{H}_{\mathbf{n}}^{K'}\mathcal{E}_2^{KK'}\underline{\mathbf{W}}^{K'}\right)\\&+\sum_{K\in\mathcal{T}_h}\sum_{F\in\partial K\cap\partial\Omega}\mathbf{G}_{\mathbf{n}}\mathcal{B}^K\underline{\mathbf{W}}^K=0,\end{aligned}\tag{2.2.21}$$

where $\mathbf{H}_{\mathbf{n}}^K=\left(\mathbf{D}_{\mathbf{n}}^K\right)^{+}$ and $\mathbf{H}_{\mathbf{n}}^{K'}=\left(\mathbf{D}_{\mathbf{n}}^{K'}\right)^{-}$ in the case of the upwind numerical flux, $\mathbf{H}_{\mathbf{n}}^K=\frac{1}{2}\mathbf{D}_{\mathbf{n}}^K$ and $\mathbf{H}_{\mathbf{n}}^{K'}=\frac{1}{2}\mathbf{D}_{\mathbf{n}}^{K'}$ for the centered numerical scheme, and $\mathbf{G}_{\mathbf{n}}^K=\mathbf{L}^K$ or $\mathbf{G}_{\mathbf{n}}^K=\left(\mathbf{D}_{\mathbf{n}}^K\right)^{+}$ depending on the type of boundary conditions.

2.3 Numerical results: Plane wave propagation in an homogeneous medium

In this section, we provide some numerical results in 2D to assess the performances (accuracy and efficiency) of the centered and upwind flux DG schemes introduced in the previous section. These schemes have been implemented in a Fortran 90 software. We use the MUMPS sparse direct solver for the resolution of the linear system (see [61] for more details).
The simulations are performed on a computer system whose characteristics are:

- 2 Quad-core Nehalem Intel® Xeon® X5550
- Frequency : 2,66 GHz
- Cache L3 : 8 Mb
- RAM : 24Gb (DDR3 1333MHz)
- Infiniband QDR : 40Gb/s
- Ethernet : 1Gb/s

All the simulations are done in sequential.
First, we consider the simple test problem of the propagation of a plane wave in an homogeneous medium. The computational domain Ω is a 10000 m $\times$10000 m square. The physical properties of the medium are $\rho = 1000$ kg.m^{-3} and the Lamé's coefficients λ and μ are set to 8 MPa and 4 MPa respectively. These values imply a velocity v_p of P-waves equal to 4000 m.s^{-1} and a velocity v_s of S-waves equal to 2000 m.s^{-1}. On the boundaries we impose an absorbing condition such that the exact solution is a plane wave incident field

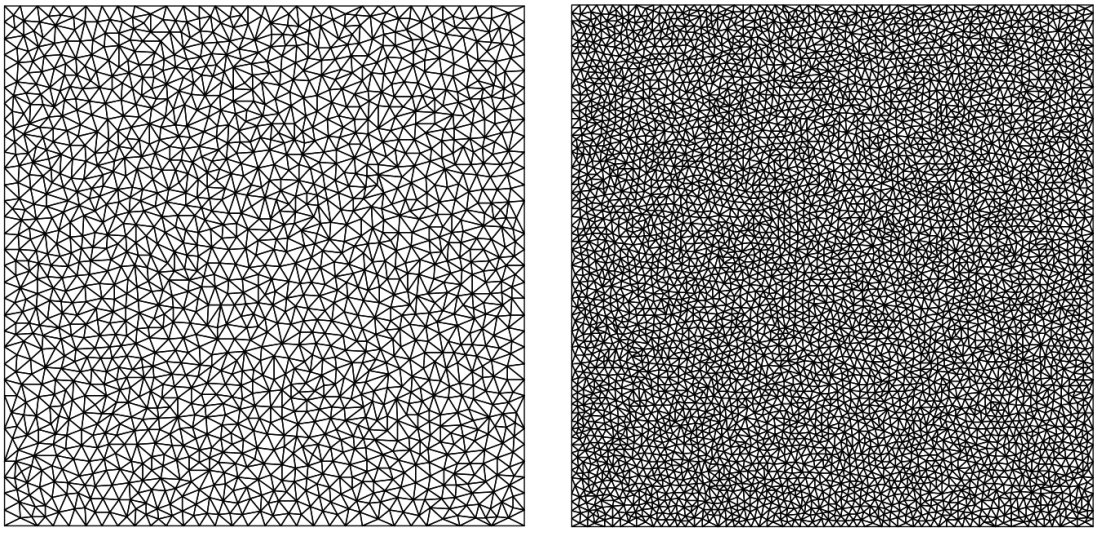
$$W_e = \nabla e^{i(k_x \cos\theta x + k_z \sin\theta z)} = \begin{pmatrix} V_{x0} \\ V_{z0} \\ \sigma_{xx0} \\ \sigma_{zz0} \\ \sigma_{xz0} \end{pmatrix} e^{i(k_x \cos\theta x + k_z \sin\theta z)},$$

where $k = \sqrt{k_x^2 + k_z^2} = \dfrac{\omega}{v_p}$ is the wavenumber, $k_x = \dfrac{\omega}{v_p}\cos\theta$ and $k_z = \dfrac{\omega}{v_p}\sin\theta$, and θ is the incidence angle. ω is the angular frequency, $\omega = 2\pi f$ where f is the frequency.
If we choose arbitrarily V_{x0}, we can express the other components such as:

$$\begin{cases} V_{z0} = \dfrac{k_x k_z (\lambda + \mu)}{\rho\omega^2 - k_x^2\mu - k_z^2\lambda + 2\mu)} V_{x0}, \\ \sigma_{xx0} = \dfrac{-1}{\omega}\left(k_x (\lambda + 2\mu) V_{x0} + \lambda k_z V_{z0}\right), \\ \sigma_{zz0} = \dfrac{-1}{\omega}\left(\lambda k_x V_{x0} + (\lambda + 2\mu) k_z V_{z0}\right), \\ \sigma_{xz0} = \dfrac{-\mu}{\omega}\left(k_z V_{x0} + k_x V_{z0}\right). \end{cases}$$

Remark. *For more details on the analytical expression of the plane wave, we refer to annex A.*

In all the experiments, we choose θ equal to 0° and $f = 2$ Hz, so that $\omega = 4\pi \simeq 12.56$ Hz. We discretize the computational domain Ω with three unstructured meshes with respectively 3000, 10 000 and 45 000 elements. Two of these meshes are shown on figs. 2.3.1a and 2.3.1b and their characteristics are gathered in tab. 2.3.1.



(a) Mesh M1, 3000 elements. (b) Mesh M2, 10000 elements.

Fig. 2.3.1: Discretization of Ω.

Mesh	# Mesh elements	# Mesh vertices	h_{min} (m)	h_{max} (m)	h_{max}/h_{min}
M1	3 100	1 620	193.6	625.0	3.23
M2	10 300	5 200	107.5	312.5	2.91
M3	45 000	22 500	45.4	156.3	3.44

Tab. 2.3.1: Characteristics of the three meshes

We compare the obtained numerical solutions by focusing on the V_x component. When using the coarsest mesh with 3 100 triangles and the DG method with uniform interpolation of order 1 (noted DGFD-$\mathbb{P}_1$ method), we obtain the numerical solutions shown on figs. 2.3.3a and 2.3.3b for the centered DG scheme and upwind DG scheme respectively. We can compare these two solutions to the exact one represented on fig. 2.3.2. Clearly, for this relatively coarse mesh, the DGFD-$\mathbb{P}_1$ formulation based on the centered scheme solution is notably less accurate than the solution obtained with the upwind scheme. Increasing the interpolation degree (figs. 2.3.4a and 2.3.4b) or the resolution of the mesh (figs. 2.3.5a and 2.3.5b) leads to numerical solutions that are closer to the exact one. These results are confirmed by the 1D x-wise plots of the V_x component for $y = 5000$ m on figs. 2.3.6 to 2.3.8. On these plots, the solution is recorded every 10 m on the $x-$axis.

Remark. *We use **Paraview** for the visualisation, see [62] and [63] for more details.*

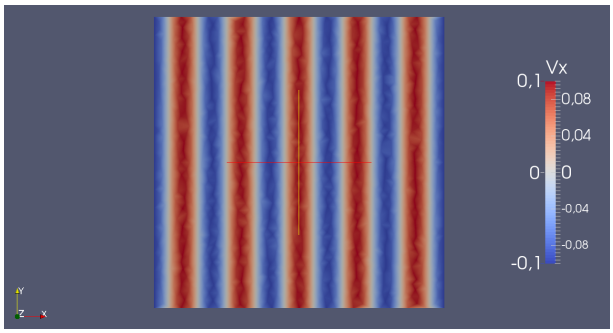
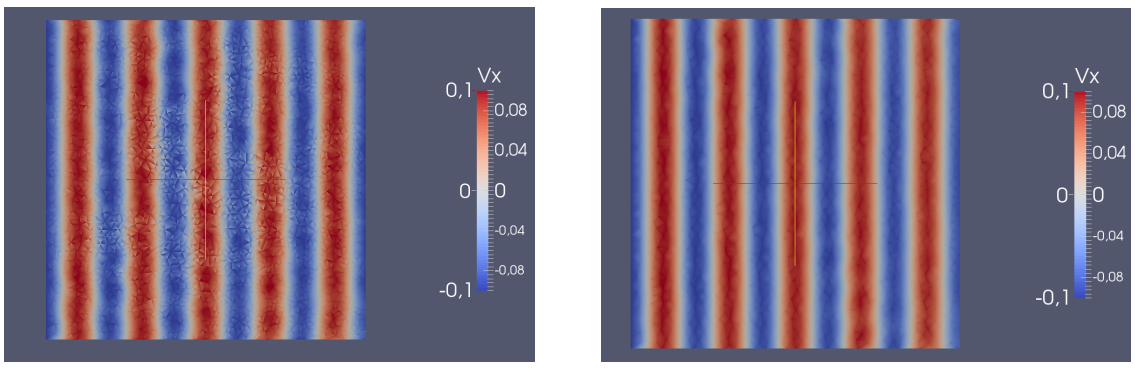
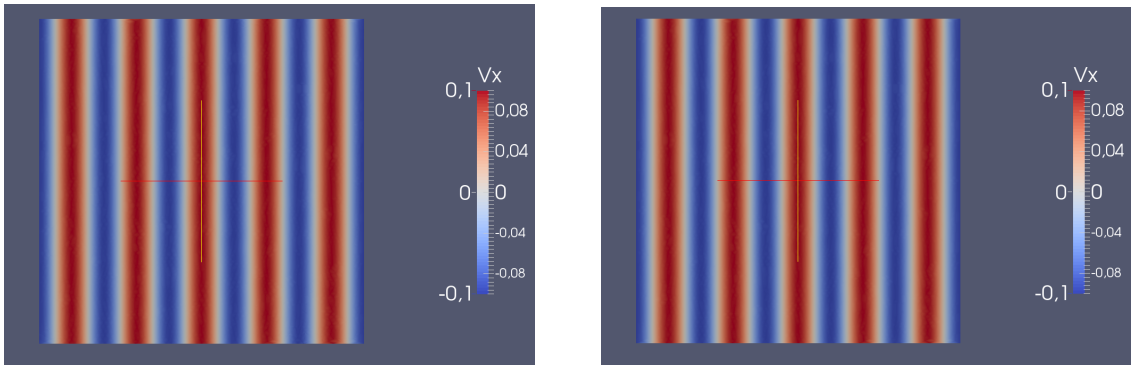


Fig. 2.3.2: Exact solution, V_x component.



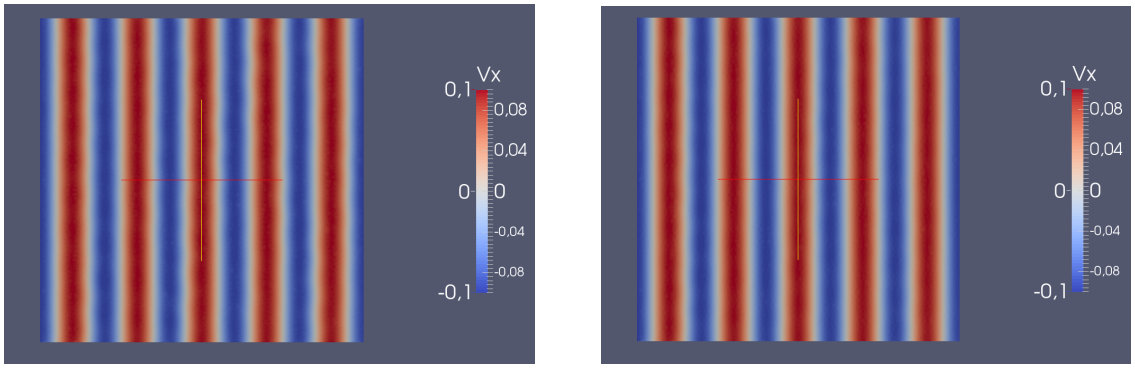
(a) Centered flux DGFD-$\mathbb{P}_1$ scheme. (b) Upwind flux DGFD-$\mathbb{P}_1$ scheme.

Fig. 2.3.3: Numerical solution, mesh M1, V_x component.



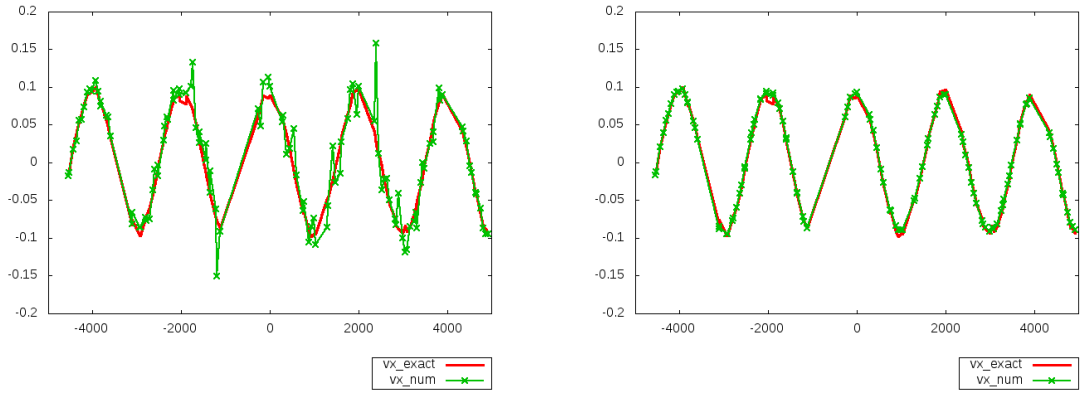
(a) Centered flux DGFD-$\mathbb{P}_2$ scheme. (b) Upwind flux DGFD-$\mathbb{P}_2$ scheme.

Fig. 2.3.4: Numerical solution, mesh M1, V_x component.



(a) Centered flux DGFD-$\mathbb{P}_1$ scheme. (b) Upwind flux DGFD-$\mathbb{P}_1$ scheme.

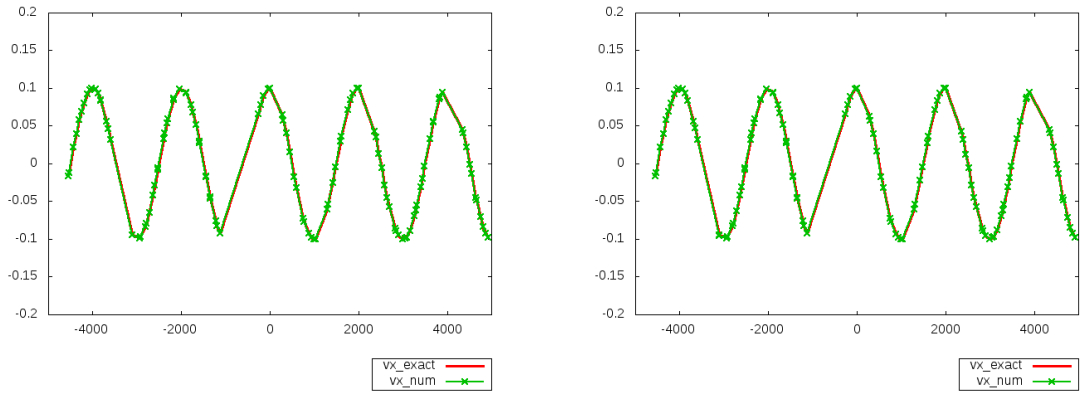
Fig. 2.3.5: Numerical solution, mesh M2, V_x component.



(a) Centered flux DGFD-$\mathbb{P}_1$ scheme. (b) Upwind flux DGFD-$\mathbb{P}_1$ scheme.

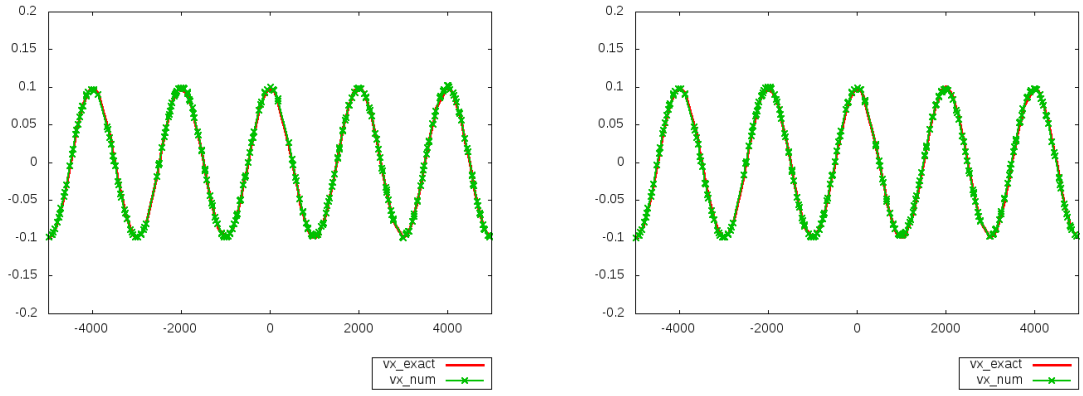
Fig. 2.3.6: x-wise distribution of V_x, mesh M1 at range y=5000 : in red line the exact solution, in green cross the centered (left) and upwind (right) DGFD-$\mathbb{P}_1$ solution.

Fig. 2.3.10 shows the numerical convergence of the centered and upwind DGFD methods. These two graphs confirm that the centered DGFD scheme is less accurate than the upwind DGFD scheme



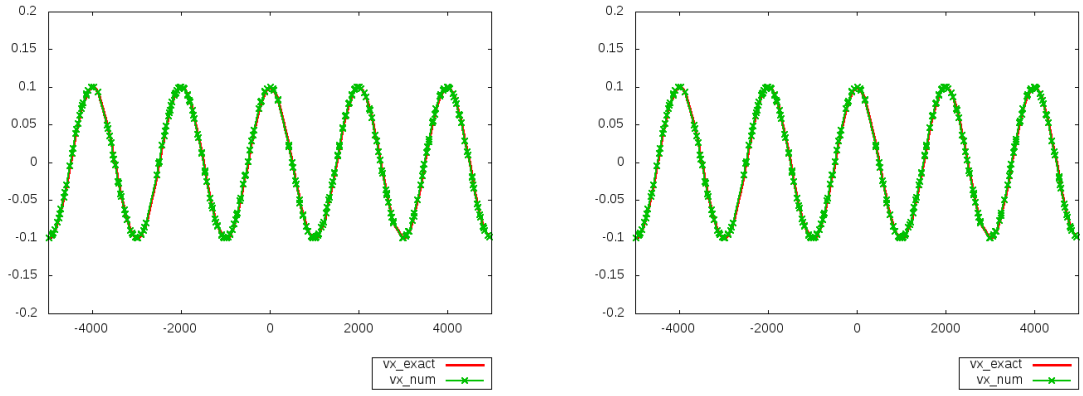
(a) Centered flux DGFD-$\mathbb{P}_2$ scheme. (b) Upwind flux DGFD-$\mathbb{P}_2$ scheme.

Fig. 2.3.7: x-wise distribution of V_x, mesh M1 at range y=5000 : in red line the exact solution, in green cross the centered (left) and upwind (right) DGFD-$\mathbb{P}_2$ solution.



(a) Centered flux DGFD-$\mathbb{P}_1$ scheme. (b) Upwind flux DGFD-$\mathbb{P}_1$ scheme.

Fig. 2.3.8: x-wise distribution of V_x, mesh M2 at range y=5000: in red line the exact solution, in green cross the centered (left) and upwind (right) DGFD-$\mathbb{P}_1$ solution.



(a) Centered flux DGFD-$\mathbb{P}_2$ scheme. (b) Upwind flux DGFD-$\mathbb{P}_2$ scheme.

Fig. 2.3.9: x-wise distribution of V_x, mesh M2 at range y=5000: in red line the exact solution, in green cross the centered (left) and upwind (right) DGFD-$\mathbb{P}_1$ solution.

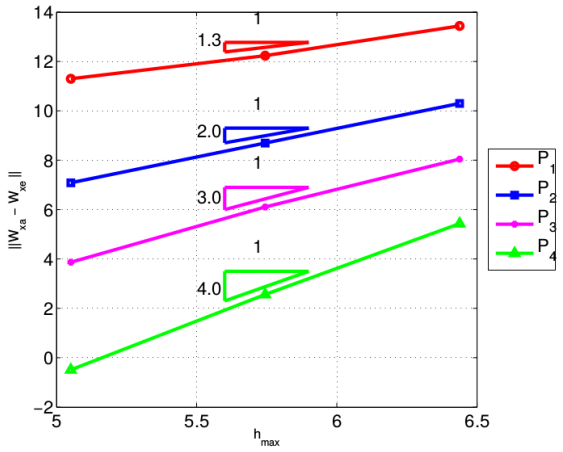
and we observe that the centered DGFD scheme converges with order p whereas the upwind DGFD scheme converges with order $p+1$, i.e. with optimal rate. In tabs. 2.3.4-2.3.5 we present the mean and relative errors in the V_x and σ_{xx} components. The mean error is computed, for example for the V_x component, as follow

$$err_{mean} = \frac{||V_{x_{exa}} - V_{x_{num}}||_{L^2}}{\text{number of elements}}. \tag{2.3.1}$$

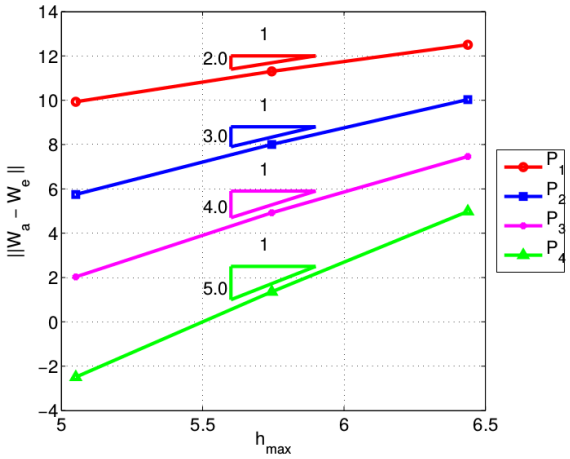
The relative error is computed as

$$err_{relative} = \frac{||V_{x_{exa}} - V_{x_{num}}||_{L^2}}{||V_{x_{exa}}||_{L^2}}. \tag{2.3.2}$$

The centered fluxes DG method is in general less accurate than the upwind fluxes DG method especially if the mesh is too coarse and for low interpolation degree.



(a) Centered flux DGFD method.



(b) Upwind flux DGFD method.

Fig. 2.3.10: Convergence order of the centered 2.3.10a and upwind 2.3.10b flux DGFD methods for plane wave propagation in a homogeneous medium.

The computational performances of both methods for all the performed simulations for this test problem are summarized in tab. 2.3.2 for the number of non-zero terms in the global matrix and the

memory used (in MB), tab. 2.3.3 for the time required for the construction of the global matrix (in seconds) and for the resolution (in seconds). The upwind DG scheme requires more memory space than the centered DG scheme. This is due to the fact that there is more non-zero terms in the global matrix in the former case (at least 1.5 times more non-zero terms). Albeit this increase in the number of non-zero terms in the global matrix, the contruction of the L and U factors for the upwind DG discrete operator does not require more CPU than that of the corresponding factors for the centered DG discrete operator. However, a difference is clearly observed in the solution time.

# Mesh elements	Interpolation degree	Non-zeros terms		Memory (MB)	
		Cent. scheme	Upw. scheme	Cent. scheme	Upw. scheme
3 100	1	7.5e+05	1.5e+06	204	288
10 300		2.5e+06	5.1e+06	877	1076
45 000		1.1e+07	2.2e+07	4489	5492
3 100	2	2.2e+06	4.3e+06	527	804
10 300		7.4e+06	1.4e+07	2036	3097
45 000		3.2e+07	6.2e+07	10687	15965
3 100	3	5.8e+06	9.4e+06	1246	1656
10 300		1.9e+07	3.1e+07	5020	6600
45 000		8.3e+07	1.3e+08	27228	34597
3 100	4	1.2e+07	1.8e+07	1980	2749
10 300		4.1e+07	5.9e+07	7372	10098
45 000		1.8e+08	2.6e+08	37018	50297

Tab. 2.3.2: Number of non-zero terms in the global matrix and memory used.

# Mesh elements	Interpolation degree	Construction time (s)		Solution time (s)	
		Cent. scheme	Upw. scheme	Cent. scheme	Upw. scheme
3 100	1	2.8e-02	4.0e-02	1.3	1.5
10 300		9.9e-02	0.1	6.7	7.2
45 000		0.4	0.7	48.4	68.0
3 100	2	8.8e-02	0.1	4.1	6.0
10 300		0.3	0.3	19.0	28.8
45 000		1.3	1.5	155.0	224.5
3 100	3	0.2	0.2	10.0	14.4
10 300		0.7	0.8	50.5	78.2
45 000		3.4	3.4	438.9	643.2
3 100	4	0.5	0.5	18.0	28.1
10 300		1.6	1.8	86.1	135.2
45 000		11.7	7.6	915.3	1077.4

Tab. 2.3.3: Time required for the global matrix construction and for the system resolution.

h (m)	Interpolation degree	Mean Error V_x		Relative Error V_x (%)	
		Cent. scheme	Upw. scheme	Cent. scheme	Upw. scheme
625.0		4.8e-02	1.8e-02	15.9	6.0
312.5	1	3.2e-03	1.7e-03	3.3	1.8
156.3		1.7e-04	9.9e-05	0.8	0.4
625.0		1.7e-03	1.6e-03	0.5	0.5
312.5	2	5.9e-05	5.7e-05	6.0e-02	5.8e-02
156.3		1.4e-06	1.4e-06	6.1e-03	6.1e-03
625.0		1.4e-04	1.3e-04	4.2e-02	4.1e-02
312.5	3	3.2e-06	3.1e-06	3.3e-03	3.2e-03
156.3		3.8e-08	3.7e-08	1.7e-04	1.7e-04
625.0		1.1e-05	1.0e-05	3.4e-03	3.2e-03
312.5	4	7.6e-08	6.8e-08	7.8e-05	7.0e-05
156.3		3.5e-10	3.1e-10	1.6e-06	1.4e-06

Tab. 2.3.4: Mean and relative errors on V_x.

h (m)	Interpolation degree	Mean Error σ_{xx}		Relative Error σ_{xx} (%)		Convergence order	
		Cent. scheme	Upw. scheme	Cent. scheme	Upw. scheme	Cent. scheme	Upw. scheme
625.0		165.5	72.9	13.6	6.0	-	-
312.5	1	17.3	6.6	4.5	1.7	1.7	1.7
156.3		1.6	0.4	1.8	0.4	1.3	2.0
625.0		7.6	5.9	0.6	0.5	-	-
312.5	2	0.5	0.3	0.1	6.5e-02	2.3	2.9
156.3		2.2e-02	6.2e-03	2.5e-02	6.9e-03	2.3	3.2
625.0		0.8	0.4	6.4e-02	3.1e-02	-	-
312.5	3	3.8e-02	1.0e-02	9.8e-03	2.7e-03	2.8	3.7
156.3		9.5e-04	1.4e-04	1.1e-03	1.5e-04	3.2	4.2
625.0		5.9e-02	3.4e-02	4.6e-03	2.7e-03	-	-
312.5	4	1.1e-03	3.1e-04	2.8e-04	7.9e-05	4.1	5.2
156.3		1.2e-05	1.5e-06	1.3e-05	1.7e-06	4.4	5.6

Tab. 2.3.5: Mean and relative errors on σ_{xx} and convergence order.

Conclusion

In this chapter, we studied two nodal DG methods, the centered flux DG method and the upwind flux DG method, for the solution of the elastic Helmholtz equations.
With the test problem of the 2D propagation of a plane wave, we have seen that the centered formulation is suboptimal: we loose one numerical convergence order than classical finite elements methods or than upwind formulation.
However the upwind scheme requires more memory space and computational time. This is due to the fact that, in the upwind flux case, the number of non-zero terms is around twice as high as the one of the centered method.
Another drawback of the upwind DG method that we do not study in this work but that we should not neglect is the dissipative character of this scheme as it is studied in [64].
Despite of this disadvantage, in the following, we prefer to consider the upwind DG method as a reference method for the comparison with the hybridizable DG (HDG) method rather than the centered DG method because this latter is not enough accurate and suboptimal.

Chapter 3

Hybridizable Discontinuous Galerkin (HDG) method for 2D elastic Helmholtz equations

In the previous chapter, we have introduced two DG formulations respectively based on centered fluxes and upwind fluxes. We have emphasized that, despite their advantages (use of unstructured meshes, adaptivity in particular), DG methods present the important drawback to be very expensive methods because of the huge number of globally coupled degrees of freedom. To overcome this drawback, a new class of DG method has been introduced recently: the hybridizable discontinuous Galerkin (HDG) method.

The HDG method drew inspiration from Fraejis De Veubeke's work on the hybridization of finite element method for the linear elasticity model [65] in 1965 where the hybridization is presented as an implementation technique. This technique is based on the introduction of an auxiliary unknown λ_h and is close to a static condensation technique. In 1985, Arnold and Brezzi showed that this unknown λ_h could be assimilated to a Lagrange multiplier, associated to a continuity condition on the approximate flux.
In 2004, Cockburn and Gopalakrishnan defined λ_h as the solution of a weak formulation and they developed the hybridization of Raviart-Thomas (RT) [66] and (BDM) [67] mixed methods of arbitrary degree [68]. They extended the method to local DG method for Stokes problem in [69] in 2006. A summary of their work on this new hybridization technique is done in [70]. Finally, Cockburn, Gopalakrishnan and Lazarov gave an unified framework for a second order elliptic model problem [28] in 2009 and proposed the hybridizable DG (HDG) method.
Then, the HDG method has been applied to various problems: the Maxwell's equations [29, 30] in time-harmonic domain; the acoustics and elastodynamics, considering displacement gradient-velocity-pressure formulation, in time implicit domain [35]; the Stokes problem [31] or for fluid problems in [71] where Giorgiani exploits the p-adaptativity feature of the HDG method; and more general partial differential equations in continuum mechanics [32] ; linear [34] and non-linear convection-diffusion problems [33].

In order to develop efficient and accurate new solvers for geophysical applications, we propose to apply the HDG method to the elastic Helmholtz equations.
This chapter is divided into four parts: first, we remind the 2D first order formulation of the elastic wave equations and we introduce the notations and definitions that we use in the HDG formulation (section 3.1). Then, we apply the HDG method to the Helmholtz equations and we discuss the well-posedness and its relation with classical DG methods (section 3.2). In section 3.3 we describe its implementation and give a summary of the HDG algorithm at the end of the section. We finish this chapter by the numerical part: the test case of the propagation of a plane wave in an homogeneous medium is used to study the numerical convergence of the method and to take a look at the condition number of the associated discrete operator. We also study the influence of the hybrid parameter on the accuracy of the solution. Finally, we test the p-adaptivity property of the HDG formulation and

compare performances with the ones obtained with the HDG method using an uniform interpolation degree. Additional numerical results, including comparisons between the upwind flux DG method, the interior penalty DG (IPDG) method and the proposed HDG method, are presented in the next chapter.

3.1 Problem statement and notations

In this section, we remind the 2D elastic waves equations and introduce some additional notations.

3.1.1 2D elastic wave equations in harmonic domain

The 2D elastodynamics system is obtained from the 3D system by considering an infinite invariant direction. This means to remove the corresponding spatial component, i.e. in geophysical applications the y component. By this way, we get, for $\mathbf{x} = (x, z) \in \Omega \subset \mathbb{R}^2$

$$\left\{ \begin{array}{rcll} i\omega\rho\mathbf{v} & = & \nabla \cdot \underline{\underline{\sigma}} & \text{in } \Omega, \\ i\omega\underline{\underline{\sigma}} & = & \underline{\underline{C}}\, \underline{\underline{\epsilon}}(\mathbf{v}) + \mathbf{f} & \text{in } \Omega. \end{array} \right. \tag{3.1.1}$$

We recall that we have omitted the $\mathbf{x}$-dependency in the mass density ρ, the velocity vector $\mathbf{v} = (\ v_x,\ \ v_z\ )^T$, the stress tensor $\underline{\underline{\sigma}}$, the source term f, and the symmetric elastic tensor $\underline{\underline{C}}$.
In 2D and using Voigt's notation, this tensor $\underline{\underline{C}}$ is reduced to a 3×3 matrix which writes in the general 2D anisotropic case as

$$\underline{\underline{C}} = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{12} & C_{22} & C_{23} \\ C_{13} & C_{23} & C_{33} \end{pmatrix}.$$

In the 2D isotropic case, it reduces to

$$\underline{\underline{C}} = \begin{pmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & \mu \end{pmatrix}.$$

Remark. *To express the elastic tensor $\underline{\underline{C}}$ into a matrix form, we use the Voigt's notation. In section 1.1 we have given it in 3D. In 2D we have*

$$(1,\ 2,\ 3)^T = (xx,\ zz,\ xz)^T.$$

Remark. *In the geophysics framework, the source term f represents the pulse emitted by the explosive source and is considered as a point source S or a Gaussian function.*

As in the previous chapter, we assume that the physical parameters (ρ, λ and μ for the isotropic case and ρ and the C_{ij} coefficients in the general case) are piecewise constant.

We can develop the equations of system (3.1.1). For $(x, z) \in \Omega \subset \mathbb{R}^2$, in the anisotropic case

$$\left\{ \begin{array}{rcl} i\omega v_x & = & \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xx}}{\partial x} + \dfrac{\partial \sigma_{xz}}{\partial z}\right), \\ i\omega v_z & = & \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xz}}{\partial x} + \dfrac{\partial \sigma_{zz}}{\partial z}\right), \\ i\omega \sigma_{xx} & = & C_{11}\dfrac{\partial v_x}{\partial x} + C_{12}\dfrac{\partial v_z}{\partial z} + C_{13}\left(\dfrac{\partial v_x}{\partial z} + \dfrac{\partial v_z}{\partial x}\right) + f_x, \\ i\omega \sigma_{zz} & = & C_{12}\dfrac{\partial v_x}{\partial x} + C_{22}\dfrac{\partial v_z}{\partial z} + C_{23}\left(\dfrac{\partial v_x}{\partial z} + \dfrac{\partial v_z}{\partial x}\right) + f_z, \\ i\omega \sigma_{xz} & = & C_{13}\dfrac{\partial v_x}{\partial x} + C_{23}\dfrac{\partial v_z}{\partial z} + C_{33}\left(\dfrac{\partial v_x}{\partial z} + \dfrac{\partial v_z}{\partial x}\right), \end{array} \right. \tag{3.1.2}$$

and for the isotropic case

$$\begin{cases} i\omega v_x &= \frac{1}{\rho}\left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xz}}{\partial z}\right), \\ i\omega v_z &= \frac{1}{\rho}\left(\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{zz}}{\partial z}\right), \\ i\omega \sigma_{xx} &= (\lambda+2\mu)\frac{\partial v_x}{\partial x} + \lambda\frac{\partial v_z}{\partial z} + f_x, \\ i\omega \sigma_{zz} &= \lambda\frac{\partial v_x}{\partial x} + (\lambda+2\mu)\frac{\partial v_z}{\partial z} + f_z, \\ i\omega \sigma_{xz} &= \mu\left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x}\right). \end{cases} \tag{3.1.3}$$

We consider two types of boundary conditions:

- The Dirichlet condition to express the free surface condition

$$\underline{\underline{\sigma}} \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_l, \tag{3.1.4}$$

- The Sommerfeld condition for the absorbing boundary condition

$$\underline{\underline{\sigma}} \cdot \mathbf{n} + PA(\theta')P^T\mathbf{v} = 0 \quad \text{on } \Gamma_a, \tag{3.1.5}$$

with Γ_l the free boundary and Γ_a the absorbing boundary, and $\Gamma_l \cup \Gamma_a = \partial\Omega$ and $\Gamma_l \cap \Gamma_a = \emptyset$.

Remark. *The expressions of the matrices P, $A(\theta)$ and θ' are detailed for the 3D case in section 1.1.5.*

Remark. *The boundary condition* (3.1.5) *is valid for TTI media. In the particular case of an isotropic medium, it reduces to*

$$\underline{\underline{\sigma}} \cdot \mathbf{n} - \rho v_p(\mathbf{v}\cdot\mathbf{n})\mathbf{n} - \rho v_s(\mathbf{v}\cdot\mathbf{t})\mathbf{t} = 0 \quad \text{on } \Gamma_a, \tag{3.1.6}$$

with $v_p = \sqrt{\dfrac{\lambda+2\mu}{\rho}}$ *the P-wave velocity,* $v_s = \sqrt{\dfrac{\mu}{\rho}}$ *the S-wave velocity,* $\mathbf{n}$ *the outward unit normal vector and* $\mathbf{t}$ *the unit tangent.*

3.1.2 Notations

As we have done in the previous chapter, we consider a triangulation $\mathcal{T}_h$ of the computational domain Ω, composed of triangles K. We use the same notations and definitions as in chapter 2, i.e:

- F : a face of the element K (see fig. 2.1.1),
- $\mathcal{F}(K)$: the set of the faces of an element K of $\mathcal{T}_h$,
- $\mathcal{F}_b$: the set of the boundary faces F_b, i.e. $F_b = \partial K \cap \Gamma$, where $\Gamma = \partial\Omega$,
- $\mathcal{F}_i$: the set of the interior faces F_i i.e. $F_i = \partial K \cap \partial K'$ where K and K' are neighbours,
- $\mathcal{F}_h$: the set of all the faces of the mesh, i.e. $\mathcal{F}_h = \mathcal{F}_i \cup \mathcal{F}_b$,
- $\mathbf{n}$: the outward unit normal vector to K, $\mathbf{t}$ its tangent.

In 2D, the reference element $\widehat{K}$ is the triangle defined by the vertices $\widehat{S}_1$ (0,0), $\widehat{S}_2$ (1,0) and $\widehat{S}_3$ (0,1) (see fig. 2.1.3) and the reference edge $\widehat{F}$ is defined by the segment $\left[\widehat{S}_1\widehat{S}_2\right]$.

The linear function F_K which allows to switch from the reference point $\widehat{\mathbf{x}}\,(\widehat{x},\ \widehat{z})$ to the point $\mathbf{x}\,(x,\ z)$ is defined in the 2D case by

$$F_K\,(\widehat{x},\widehat{z}) = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ z_2 - z_1 & z_3 - z_1 \end{pmatrix} \begin{pmatrix} \widehat{x} \\ \widehat{z} \end{pmatrix} + \begin{pmatrix} x_1 \\ z_1 \end{pmatrix}.$$

The set $\mathbb{P}_p(D)$ is the set of polynomials of degree at most p on the domain D.
Let Φ^K a basis of $\mathbb{P}_p(K)$ composed of Lagrange polynomial functions φ_i^K defined by the relation (2.1.7) in chapter 2.
For each element $K \in \mathcal{T}_h$, we define $V^p(K)$ as the space $\mathbb{P}_p(K)$, $\mathbf{V}^p(K)$ as the space $(\mathbb{P}_p(K))^2$ and $\mathbf{\Sigma}^p(K)$ as the space $(\mathbb{P}_p(K))^3$. The discontinuous finite element spaces that we need for the HDG formulation are given by

$$\begin{aligned} V_h^p &= \{v \in L^2(\Omega) : \quad v|_K \in V^p(K), \forall K \in \mathcal{T}_h\}, \\ \mathbf{V}_h^p &= \{\mathbf{v} \in \left(L^2(\Omega)\right)^2 : \quad \mathbf{v}|_K \in \mathbf{V}^p(K), \forall K \in \mathcal{T}_h\}, \\ \mathbf{\Sigma}_h^p &= \{\underline{\underline{\sigma}} \in \left(L^2(\Omega)\right)^3 : \quad \underline{\underline{\sigma}}|_K \in \mathbf{\Sigma}^p(K), \forall K \in \mathcal{T}_h\}, \end{aligned}$$

where $L^2(\Omega)$ is the space of square-integrable functions on the domain Ω.
We also need to introduce the traced finite element space

$$\mathbf{M}_h = \{\eta \in \left(L^2(\mathcal{F}_h)\right)^2 : \quad \eta|_F \in (P_p(F))^2, \forall F \in \mathcal{F}_h\}.$$

The space $\mathbf{M}_h$ represents the space of functions that are continuous on an edge but discontinuous at its ends.
In order to better understand the difference between the spaces $\mathbf{M}_h$ and $\mathbf{V}_h^p$, we show on fig. 3.1.1 the location of the degrees of freedom of these two spaces.

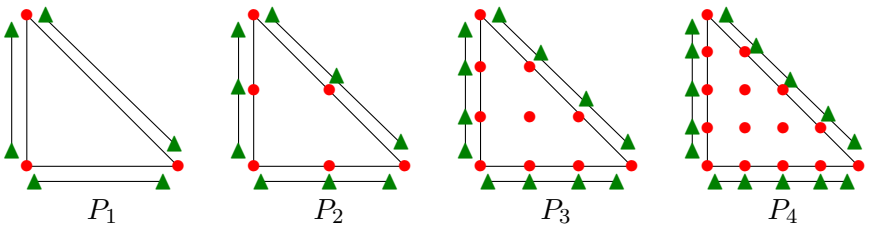


Fig. 3.1.1: Representation of the locations of the degrees of freedom of the spaces $\mathbf{V}_h^p$ (●) and $\mathbf{M}_h$ (▲).

Finally, we recall the definition of the jump $[\![\cdot]\!]$ of a vector $\mathbf{v}$ for an interior interface $F = \partial K^+ \cap \partial K^- \in \mathcal{F}_i$

$$[\![\mathbf{v} \cdot \mathbf{n}]\!] = \mathbf{v}^+ \cdot \mathbf{n}^+ + \mathbf{v}^- \cdot \mathbf{n}^-,$$

and for a boundary face $F = \partial K^+ \cap \Gamma \in \mathcal{F}_b$

$$[\![\mathbf{v} \cdot \mathbf{n}]\!] = \mathbf{v}^+ \cdot \mathbf{n}^+.$$

The jump of a tensor $\underline{\underline{\sigma}}$ is defined for an interior face F_i by

$$[\![\underline{\underline{\sigma}} \cdot \mathbf{n}]\!] = \underline{\underline{\sigma}}^+ \cdot \mathbf{n}^+ + \underline{\underline{\sigma}}^- \cdot \mathbf{n}^-,$$

and on an external face F_b by

$$[\![\underline{\underline{\sigma}} \cdot \mathbf{n}]\!] = \underline{\underline{\sigma}}^+ \cdot \mathbf{n}^+.$$

3.2 HDG formulation

Now that we have introduced the problem statement and the notations and spaces that we need, we present the weak form of the HDG formulation of the problem. We also verify the well-posedness of the local problems of the formulation and we establish a link between the HDG formulation and the upwind fluxes DG formulation.

3.2.1 Principles

To establish the HDG formulation, we consider equations (3.1.1) on an element K of $\mathcal{T}_h$. We multiply them by test functions $(\mathbf{w}, \underline{\underline{\xi}}) \in \mathbf{V}^p(K) \times \mathbf{\Sigma}^p(K)$ and we apply an integration by parts. We look for an approximate solution $(\mathbf{v}_h, \underline{\underline{\sigma}}_h) \in \mathbf{V}_h^p \times \mathbf{\Sigma}_h^p$ such as

$$\left\{ \begin{array}{rcl} \displaystyle\int_K i\omega\rho_K \mathbf{v}_h \cdot \mathbf{w} - \int_K \left(\nabla \cdot \underline{\underline{\sigma}}_h\right) \cdot \mathbf{w} & = & 0, \\ \displaystyle\int_K i\omega \underline{\underline{\sigma}}_h : \underline{\underline{\xi}} - \int_K \left(\underline{\underline{C}}_K \underline{\underline{\epsilon}}(\mathbf{v}_h)\right) : \underline{\underline{\xi}} & = & \displaystyle\int_K \mathbf{f} : \underline{\underline{\xi}}. \end{array} \right. \tag{3.2.1}$$

Remark. *We denote by $\underline{\underline{a}} : \underline{\underline{b}}$ the scalar product between two tensors $\underline{\underline{a}}$ and $\underline{\underline{b}}$.*

By integrating by parts we obtain

$$\left\{ \begin{array}{rcl} \displaystyle\int_K i\omega\rho_K \mathbf{v}_h \cdot \mathbf{w} + \int_K \underline{\underline{\sigma}}_h : \nabla \mathbf{w} - \int_{\partial K} \widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n} \cdot \mathbf{w} & = & 0, \\ \displaystyle\int_K i\omega \underline{\underline{\sigma}}_h : \underline{\underline{\xi}} + \int_K \mathbf{v}_h \cdot \nabla \cdot \left(\underline{\underline{C}}_K \underline{\underline{\xi}}\right) - \int_{\partial K} \widehat{\mathbf{v}}_h \cdot \underline{\underline{C}}_K \underline{\underline{\xi}} \cdot \mathbf{n} & = & \displaystyle\int_K \mathbf{f} : \underline{\underline{\xi}}. \end{array} \right. \tag{3.2.2}$$

On the boundary ∂K the numerical traces $\widehat{\underline{\underline{\sigma}}}_h$ and $\widehat{\mathbf{v}}_h$ represent the values of $\underline{\underline{\sigma}}_h$ and $\mathbf{v}_h$ respectively.

The principle of the HDG formulation consists in expressing $(\mathbf{v}_h, \underline{\underline{\sigma}}_h)$ in terms of a hybrid unknown λ_h only. This unknown $\lambda_h \in \mathbf{M}_h$ is a Lagrange multiplier and is mainly introduced in order to replace the numerical trace $\widehat{\mathbf{v}}_h$. We thus define

$$\lambda_h = \widehat{\mathbf{v}}_h, \forall F \in \mathcal{F}_h, \qquad \lambda_h \in \mathbf{M}_h. \tag{3.2.3}$$

The numerical trace $\widehat{\underline{\underline{\sigma}}}_h$ is then expressed in terms of the other unknowns $\underline{\underline{\sigma}}_h$, $\mathbf{v}_h$ and, λ_h by the following relation

$$\widehat{\underline{\underline{\sigma}}}_h = \underline{\underline{\sigma}}_h - \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right) \otimes \mathbf{n} \qquad \text{on } \partial K. \tag{3.2.4}$$

where $\mathbf{S}$ is a local stabilization matrix which has an important effect on both accuracy and stability of the numerical scheme as we will see in the sequel.

Remark. *In [35], Cockburn, Peraire and Nguyen express the time-implicit isotropic elastodynamics equations using the displacement gradient-velocity-pressure formulation, i.e. they choose to express the isotropic elastic waves equations as a function of the velocity $\mathbf{v}$, the displacement gradient tensor*

$\mathbf{H} = \nabla\mathbf{u}$, *and the hydrostatic pressure* $p = (\mu + \lambda)(\nabla \cdot \mathbf{u})$. *In their HDG formulation, they express (see equation (33)) the numerical trace* $\left(\mu\widehat{\mathbf{H}}_h^n + \widehat{p}_h^n\mathbf{I}\right)$ *as a function of* $\mathbf{v}_h$, $\lambda_h = \widehat{\mathbf{v}}_h$ *the Lagrange multiplier,* $\mathbf{H}_h$ *and* p_h *as*

$$\mu\widehat{\mathbf{H}}_h^n + \widehat{p}_h^n\mathbf{I} = \mu\mathbf{H}_h^n + p_h^n\mathbf{I} - \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right) \otimes \mathbf{n}. \tag{3.2.5}$$

We easily check that

$$\nabla\left(\mu\mathbf{H}_h^n + p_h^n\mathbf{I}\right) = \nabla\underline{\underline{\sigma}}_h. \tag{3.2.6}$$

Consequently, assuming that

$$\nabla\left(\mu\widehat{\mathbf{H}}_h^n + \widehat{p}_h^n\mathbf{I}\right) = \nabla\widehat{\underline{\underline{\sigma}}}_h, \tag{3.2.7}$$

and using the fact that the operator ∇ *is a linear operator, we replace* $\left(\mu\widehat{\mathbf{H}}_h^n + \widehat{p}_h^n\mathbf{I}\right)$ *in this equation by its definition* (3.2.5)*, and we found that*

$$\nabla\widehat{\underline{\underline{\sigma}}}_h = \nabla\left(\underline{\underline{\sigma}}_h - \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right) \otimes \mathbf{n}\right). \tag{3.2.8}$$

We deduce the relation (3.2.4) *and establish an equivalence between the two isotropic elastic formulations.*

Remark. *In the next section, we will prove from an energy identity that* $\mathbf{S}$ *has to be definite positive in order to ensure the well-posedness of the local HDG formulation.*
Moreover, in [35], it is proved from the energy identity that $\mathbf{S}$ *can have the form* $\tau\mathbf{I}$ *(eq. 37 of [35]) where* $\tau > 0$ *is a local stabilization parameter and* $\mathbf{I}$ *the identity matrix. It is this expression of* $\mathbf{S}$ *that we choose in this thesis. We study the impact of this parameter on the accuracy of the solution in the numerical section.*

If we replace $\widehat{\mathbf{v}_h}$ by λ_h in (5.3.1)

$$\left\{\begin{array}{rcl} \displaystyle\int_K i\omega\rho_K\mathbf{v}_h \cdot \mathbf{w} + \int_K \underline{\underline{\sigma}}_h : \nabla\mathbf{w} - \int_{\partial K} \widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n} \cdot \mathbf{w} & = & 0, \\ \displaystyle\int_K i\omega\underline{\underline{\sigma}}_h : \underline{\underline{\xi}} + \int_K \mathbf{v}_h \cdot \nabla \cdot \left(\underline{\underline{C}}_K\underline{\underline{\xi}}\right) - \int_{\partial K} \lambda_h \cdot \underline{\underline{C}}_K\underline{\underline{\xi}} \cdot \mathbf{n} & = & \displaystyle\int_K \mathbf{f} : \underline{\underline{\xi}}. \end{array}\right.$$

we observe that we have now three unknowns but only two equations. Thus, we have to introduce a new equation in order to determine λ_h. To be compliant with the underlying physics, we enforce the continuity of the normal component of $\widehat{\underline{\underline{\sigma}}}_h$.
By summing contributions of equations (5.3.1) over all elements and enforcing the continuity of the normal component of $\widehat{\underline{\underline{\sigma}}}_h$, the problem can be rewritten in the following way:
find $(\mathbf{v}_h, \underline{\underline{\sigma}}_h, \lambda_h) \in \mathbf{V}_h^p \times \mathbf{\Sigma}_h^p \times \mathbf{M}_h$ such that: $\forall(\mathbf{w}, \underline{\underline{\xi}}, \eta) \in \mathbf{V}^p(K) \times \mathbf{\Sigma}^p(K) \times \mathbf{M}_h$,

$$\left\{\begin{array}{rcl} \displaystyle\sum_{K\in\mathcal{T}_h}\int_K i\omega\rho_K\mathbf{v}_h \cdot \mathbf{w} + \sum_{K\in\mathcal{T}_h}\int_K \underline{\underline{\sigma}}_h : \nabla\mathbf{w} - \sum_{K\in\mathcal{T}_h}\int_{\partial K} \widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n} \cdot \mathbf{w} & = & 0, \\ \displaystyle\sum_{K\in\mathcal{T}_h}\int_K i\omega\underline{\underline{\sigma}}_h : \underline{\underline{\xi}} + \sum_{K\in\mathcal{T}_h}\int_K \mathbf{v}_h \cdot \nabla \cdot \left(\underline{\underline{C}}_K\underline{\underline{\xi}}\right) - \sum_{K\in\mathcal{T}_h}\int_{\partial K} \lambda_h \cdot \underline{\underline{C}}_K\underline{\underline{\xi}} \cdot \mathbf{n} & = & \displaystyle\sum_{K\in\mathcal{T}_h}\int_K \mathbf{f} : \underline{\underline{\xi}}, \\ \displaystyle\sum_{F\in\mathcal{F}_h}\int_F [\![\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n}]\!] \cdot \eta & = & 0. \end{array}\right. \tag{3.2.9}$$

Remark. *The last equation of* (3.2.9)*, which is called the conservativity condition, expresses the weak continuity of the normal component of* $\widehat{\underline{\underline{\sigma}}}_h$.

Property. *According to* (3.2.4),

$$\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n} = \underline{\underline{\sigma}}_h \cdot \mathbf{n} - \mathbf{S}\left(\mathbf{v}_h - \widehat{\mathbf{v}}_h\right). \tag{3.2.10}$$

Moreover, we have:

$$\sum_{F\in\mathcal{F}_h} \int_F [\![\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n}]\!] \cdot \eta = \sum_{K\in\mathcal{T}_h} \int_{\partial K} \left(\underline{\underline{\sigma}}_h \cdot \mathbf{n}\right) \cdot \eta - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \mathbf{S}\left(\mathbf{v}_h - \widehat{\mathbf{v}}_h\right) \cdot \eta. \tag{3.2.11}$$

Proof. For the equation (3.2.10), we have

$$\begin{aligned}
\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n} &= \left(\underline{\underline{\sigma}}_h - \mathbf{S}\left(\mathbf{v}_h - \widehat{\mathbf{v}}_h\right) \otimes \mathbf{n}\right) \cdot \mathbf{n} \\
&= \underline{\underline{\sigma}}_h \cdot \mathbf{n} - \mathbf{S}\left(\mathbf{v}_h - \widehat{\mathbf{v}}_h\right) \otimes \mathbf{n} \cdot \mathbf{n} \\
&= \underline{\underline{\sigma}}_h \cdot \mathbf{n} - \mathbf{S}\left(\mathbf{v}_h - \widehat{\mathbf{v}}_h\right).
\end{aligned}$$

For equation (3.2.11), on a face $F = \partial K^+ \cap \partial K^-$, we have

$$\begin{aligned}
\int_F [\![\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n}]\!] \cdot \eta &= \int_F [\![\underline{\underline{\sigma}}_h \cdot \mathbf{n} - \mathbf{S}\left(\mathbf{v}_h - \widehat{\mathbf{v}}_h\right)]\!] \cdot \eta \\
&= \int_F \left(\underline{\underline{\sigma}}_h^{K^+} \cdot \mathbf{n}^{K^+} - \mathbf{S}^{K^+}\left(\mathbf{v}_h^{K^+} - \widehat{\mathbf{v}}_h\right)\right) \cdot \eta + \left(\underline{\underline{\sigma}}_h^{K^-} \cdot \mathbf{n}^{K^-} - \mathbf{S}^{K^-}\left(\mathbf{v}_h^{K^-} - \widehat{\mathbf{v}}_h\right)\right) \cdot \eta.
\end{aligned}$$

Then, using the definition of the jump, we obtain equation (3.2.11).

□

Introducing equations (3.2.10) and (3.2.11) in the first and third equations of system (3.2.9) respectively, and using an integration by parts of the first equation, we obtain the following global HDG formulation

$$\left\{\begin{array}{rcl}
\displaystyle\sum_{K\in\mathcal{T}_h} \int_K i\omega\rho_K \mathbf{v}_h \cdot \mathbf{w} - \sum_{K\in\mathcal{T}_h} \int_K \left(\nabla \cdot \underline{\underline{\sigma}}_h\right) \cdot \mathbf{w} + \sum_{K\in\mathcal{T}_h} \int_{\partial K} \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right) \cdot \mathbf{w} &=& 0, \\
\displaystyle\sum_{K\in\mathcal{T}_h} \int_K i\omega \underline{\underline{\sigma}}_h : \underline{\underline{\xi}} + \sum_{K\in\mathcal{T}_h} \int_K \mathbf{v}_h \cdot \nabla \cdot \left(\underline{\underline{C}}_K \underline{\underline{\xi}}\right) - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \lambda_h \cdot \underline{\underline{C}}_K \underline{\underline{\xi}} \cdot \mathbf{n} &=& \displaystyle\sum_{K\in\mathcal{T}_h} \int_K \mathbf{f} : \underline{\underline{\xi}}, \\
\displaystyle\sum_{K\in\mathcal{T}_h} \int_{\partial K} \left(\underline{\underline{\sigma}}_h \cdot \mathbf{n}\right) \cdot \eta - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right) \cdot \eta &=& 0.
\end{array}\right. \tag{3.2.12}$$

The local problem on an element K reads as

$$\left\{\begin{array}{rcl}
\displaystyle\int_K i\omega\rho_K \mathbf{v}_h^K \cdot \mathbf{w} - \int_K \left(\nabla \cdot \underline{\underline{\sigma}}_h^K\right) \cdot \mathbf{w} + \int_{\partial K} \mathbf{S}\left(\mathbf{v}_h^K - \lambda_h\right) \cdot \mathbf{w} &=& 0, \\
\displaystyle\int_K i\omega \underline{\underline{\sigma}}_h^K : \underline{\underline{\xi}} + \int_K \mathbf{v}_h^K \cdot \nabla \cdot \left(\underline{\underline{C}}_K \underline{\underline{\xi}}\right) - \int_{\partial K} \lambda_h \cdot \underline{\underline{C}}_K \underline{\underline{\xi}} \cdot \mathbf{n} &=& \displaystyle\int_K \mathbf{f} : \underline{\underline{\xi}}.
\end{array}\right. \tag{3.2.13}$$

Remark. *The source term is taken into account through the local problem, else the HDG method is able to handle with volumic, surfacic or point source as well as classical DG methods.*

3.2.2 Well-posedness of the local problem

In order to analyze the well-posedness of the local problem, we consider the local equations (3.2.13) without source term

$$\begin{cases} \displaystyle\int_K i\omega\rho_K \mathbf{v}_h^K \cdot \mathbf{w} - \int_K \left(\nabla \cdot \underline{\underline{\sigma}}_h^K\right) \cdot \mathbf{w} + \int_{\partial K} \mathbf{S}\left(\mathbf{v}_h^K - \lambda_h\right) \cdot \mathbf{w} = 0, \qquad \forall \mathbf{w} \in \mathbf{V}^p(K) \\ \displaystyle\int_K i\omega \underline{\underline{\sigma}}_h^K : \underline{\underline{\xi}} + \int_K \mathbf{v}_h^K \cdot \nabla \cdot \left(\underline{\underline{C}}_K \underline{\underline{\xi}}\right) - \int_{\partial K} \lambda_h \cdot \underline{\underline{C}}_K \underline{\underline{\xi}} \cdot \mathbf{n} = 0, \qquad \forall \xi \in \mathbf{\Sigma}^p(K) \end{cases} \tag{3.2.14}$$

We choose $\mathbf{w} = \overline{\mathbf{v}}_h^K$ and $\xi = \left(\underline{\underline{\mathbf{C}}}^K\right)^{-1} \overline{\underline{\underline{\sigma}}}_h^K$ as test functions and we add the two above equations. We obtain

$$\int_K i\omega\rho_K \mathbf{v}_h^K \cdot \overline{\mathbf{v}}_h^K - \int_K \left(\nabla \cdot \underline{\underline{\sigma}}_h^K\right) \cdot \overline{\mathbf{v}}_h^K + \int_{\partial K} \mathbf{S}\left(\mathbf{v}_h^K - \lambda_h\right) \cdot \overline{\mathbf{v}}_h^K +$$

$$\int_K i\omega \underline{\underline{\sigma}}_h^K : \underline{\underline{\mathbf{C}}}^{-1} \overline{\underline{\underline{\sigma}}}_h^K + \int_K \mathbf{v}_h^K \cdot \nabla \cdot \left(\overline{\underline{\underline{\sigma}}}_h^K\right) - \int_{\partial K} \lambda_h \cdot \overline{\underline{\underline{\sigma}}}_h^K \cdot \mathbf{n} = 0.$$

Taking $\lambda_h = 0$, we get

$$\int_K i\omega\rho_K \mathbf{v}_h^K \cdot \overline{\mathbf{v}}_h^K + \int_K i\omega \underline{\underline{\mathbf{C}}}^{-1} \underline{\underline{\sigma}}_h^K : \overline{\underline{\underline{\sigma}}}_h^K + \int_K 2i\Im\left(\left(\nabla \cdot \underline{\underline{\sigma}}_h^K\right) \cdot \overline{\mathbf{v}}_h^K\right) + \int_{\partial K} \mathbf{S}\mathbf{v}_h^K \cdot \overline{\mathbf{v}}_h^K = 0. \tag{3.2.15}$$

We remind that ω which is the angular frequency and ρ which is the mass density are two strictly positive reals variables; the tensor $\underline{\underline{\mathbf{C}}}^{-1}$, whose properties are detailed in section 1.1.1 in chapter 1, is a positive definite tensor in the space of symmetric tensors (i.e. it verifies $\underline{\underline{\mathbf{C}}}^{-1} \underline{\underline{\xi}} : \underline{\underline{\xi}} > 0, \forall \xi$ symmetric tensor).

So if we consider the real part of the equation (3.2.15), we have $\displaystyle\int_{\partial K} \mathbf{S}\mathbf{v}_h^K \cdot \overline{\mathbf{v}}_h^K = 0$ which implies that $\mathbf{v}_h^K = 0$ on ∂K if $\mathbf{S}$ is a definite positive matrix. This shows us that we can consider any form of $\mathbf{S}$ as long as it is a definite positive matrix. We have chosen to consider $\mathbf{S} = \tau\mathbf{I}$ because this form leads to sparser local matrices and because it is the form used in [35]. Moreover, we have tested numerically others forms of $\mathbf{S}$ in 2D and we do not have seen large differences on the results.

For the HDG-$\mathbb{P}_1$ and HDG-$\mathbb{P}_2$ formulations, all the degrees of freedom are on ∂K, so $\mathbf{v}_h^K = 0$ on the entire element K. This implies that $\underline{\underline{\sigma}}^K = 0$ on the element K. Thus we can ensure the local well-posedness of the local problem for the HDG-$\mathbb{P}_1$ and HDG-$\mathbb{P}_2$ methods.

Remark. *In 3D, the HDG-$\mathbb{P}_3$ method also has its degrees of freedom on ∂K. Thus, the local well-posedness of the HDG-$\mathbb{P}_3$ method is ensured in 3D.*

3.2.3 Relationship between HDG and upwind flux DG

A link between the HDG formulation and the upwind flux DG method can be shown by considering the global HDG formulation

$$\begin{cases} \displaystyle\sum_{K\in\mathcal{T}_h} \int_K i\omega\rho_K \mathbf{v}_h \cdot \mathbf{w} - \sum_{K\in\mathcal{T}_h} \int_K \left(\nabla \cdot \underline{\underline{\sigma}}_h\right) \cdot \mathbf{w} + \sum_{K\in\mathcal{T}_h} \int_{\partial K} \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right) \cdot \mathbf{w} &= 0, \\ \displaystyle\sum_{K\in\mathcal{T}_h} \int_K i\omega \underline{\underline{\sigma}}_h : \underline{\underline{\xi}} + \sum_{K\in\mathcal{T}_h} \int_K \mathbf{v}_h \cdot \nabla \cdot \left(\underline{\underline{C}}_K \underline{\underline{\xi}}\right) - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \lambda_h \cdot \underline{\underline{C}}_K \underline{\underline{\xi}} \cdot \mathbf{n} &= \sum_{K\in\mathcal{T}_h} \int_K \mathbf{f} : \underline{\underline{\xi}}, \\ \displaystyle\sum_{K\in\mathcal{T}_h} \int_{\partial K} \left(\underline{\underline{\sigma}}_h \cdot \mathbf{n}\right) \cdot \eta - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right) \cdot \eta &= 0. \end{cases}$$

We write it as

$$\sum_{K\in\mathcal{T}_h}\int_K i\omega\rho_K\mathbf{v}_h\cdot\mathbf{w}+\sum_{K\in\mathcal{T}_h}\int_K \underline{\underline{\sigma}}_h:(\nabla\cdot\mathbf{w})-\sum_{K\in\mathcal{T}_h}\int_{\partial K}\underline{\underline{\sigma}}_h\cdot\mathbf{w}\mathbf{n}+\sum_{K\in\mathcal{T}_h}\int_{\partial K}\mathbf{S}\left(\mathbf{v}_h-\lambda_h\right)\cdot\mathbf{w}=0, \quad (3.2.16)$$

$$\sum_{K\in\mathcal{T}_h}\int_K i\omega\underline{\underline{\sigma}}_h:\underline{\underline{C}}_K^{-1}\underline{\underline{\xi}}'+\sum_{K\in\mathcal{T}_h}\int_K\mathbf{v}_h\cdot\nabla\cdot\underline{\underline{\xi}}'-\sum_{K\in\mathcal{T}_h}\int_{\partial K}\lambda_h\cdot\underline{\underline{\xi}}'\cdot\mathbf{n}=\sum_{K\in\mathcal{T}_h}\int_K\mathbf{f}:\underline{\underline{C}}_K^{-1}\underline{\underline{\xi}}', \quad (3.2.17)$$

$$\sum_{F\in\mathcal{F}_h}\int_F[\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!]\cdot\eta-\sum_{F\in\mathcal{F}_h}\int_F\left(\mathbf{S}^+\mathbf{v}_h^++\mathbf{S}^-\mathbf{v}_h^-\right)\cdot\eta+\sum_{F\in\mathcal{F}_h}\int_F\left(\mathbf{S}^++\mathbf{S}^-\right)\lambda_h\cdot\eta=0. \quad (3.2.18)$$

From equation (3.2.18), we deduce

$$\sum_{F\in\mathcal{F}_h}\int_F\left(\mathbf{S}^++\mathbf{S}^-\right)\lambda_h\cdot\eta=\sum_{F\in\mathcal{F}_h}\left(\int_F\left(\mathbf{S}^+\mathbf{v}_h^++\mathbf{S}^-\mathbf{v}_h^-\right)\cdot\eta-\int_F[\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!]\cdot\eta\right),$$

and, by setting $\eta'=\left(\mathbf{S}^++\mathbf{S}^-\right)\eta$ on each internal face

$$\sum_{F\in\mathcal{F}_h}\int_F\lambda_h\cdot\eta'=\sum_{F\in\mathcal{F}_h}\left(\int_F\frac{\left(\mathbf{S}^+\mathbf{v}_h^++\mathbf{S}^-\mathbf{v}_h^-\right)}{\left(\mathbf{S}^++\mathbf{S}^-\right)}\cdot\eta'-\int_F\frac{1}{\left(\mathbf{S}^++\mathbf{S}^-\right)}[\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!]\cdot\eta'\right).$$

In equation (3.2.17) we can write

$$\sum_{K\in\mathcal{T}_h}\int_{\partial K}\lambda_h\cdot\underline{\underline{\xi}}'\cdot\mathbf{n}=\sum_{F\in\mathcal{F}_h}\int_F\lambda_h\cdot[\![\underline{\underline{\xi}}'\cdot\mathbf{n}]\!].$$

If the interpolation degree of $\mathbf{v}_h$ and $\underline{\underline{\sigma}}_h$ is lower or equal to the one of λ_h then there exists $\varphi\in\mathbf{M}_h$ such that $\varphi|_F=[\![\underline{\underline{\xi}}'\cdot\mathbf{n}]\!]$ on each face F. Then, we deduce from the two above expressions, by choosing $\eta'=\varphi$

$$\sum_{K\in\mathcal{T}_h}\int_{\partial K}\lambda_h\cdot\underline{\underline{\xi}}'\cdot\mathbf{n}=\sum_{F\in\mathcal{F}_h}\left(\int_F\frac{\left(\mathbf{S}^+\mathbf{v}_h^++\mathbf{S}^-\mathbf{v}_h^-\right)}{\left(\mathbf{S}^++\mathbf{S}^-\right)}[\![\underline{\underline{\xi}}'\cdot\mathbf{n}]\!]-\int_F\frac{1}{\left(\mathbf{S}^++\mathbf{S}^-\right)}[\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!]\cdot[\![\underline{\underline{\xi}}'\cdot\mathbf{n}]\!]\right).$$

We develop the expression of $\dfrac{\left(\mathbf{S}^+\mathbf{v}_h^++\mathbf{S}^-\mathbf{v}_h^-\right)}{\left(\mathbf{S}^++\mathbf{S}^-\right)}$

$$\begin{aligned}\frac{\mathbf{S}^+\mathbf{v}_h^++\mathbf{S}^-\mathbf{v}_h^-}{\mathbf{S}^++\mathbf{S}^-}&=\frac{\mathbf{S}^+\mathbf{v}_h^++\mathbf{S}^-\mathbf{v}_h^-+\mathbf{S}^-\mathbf{v}_h^++\mathbf{S}^+\mathbf{v}_h^-}{2\left(\mathbf{S}^++\mathbf{S}^-\right)}+\frac{\mathbf{S}^+\mathbf{v}_h^++\mathbf{S}^-\mathbf{v}_h^-}{2\left(\mathbf{S}^++\mathbf{S}^-\right)}-\frac{\mathbf{S}^-\mathbf{v}_h^++\mathbf{S}^+\mathbf{v}_h^-}{2\left(\mathbf{S}^++\mathbf{S}^-\right)}\\&=\frac{\mathbf{v}_h^++\mathbf{v}_h^-}{2}+\frac{\mathbf{S}^+}{2\left(\mathbf{S}^++\mathbf{S}^-\right)}\left(\mathbf{v}_h^+-\mathbf{v}_h^-\right)-\frac{\mathbf{S}^-}{2\left(\mathbf{S}^++\mathbf{S}^-\right)}\left(\mathbf{v}_h^+-\mathbf{v}_h^-\right)\\&=\frac{\mathbf{v}_h^++\mathbf{v}_h^-}{2}+\frac{\left(\mathbf{S}^+-\mathbf{S}^-\right)\left(\mathbf{v}_h^+-\mathbf{v}_h^-\right)}{2\left(\mathbf{S}^++\mathbf{S}^-\right)},\end{aligned}$$

in order to write equation (3.2.17) as

$$\begin{aligned}&\sum_{K\in\mathcal{T}_h}\int_K i\omega\underline{\underline{\sigma}}_h:\underline{\underline{\xi}}+\sum_{K\in\mathcal{T}_h}\int_K\mathbf{v}_h\cdot\nabla\cdot\underline{\underline{C}}_K\underline{\underline{\xi}}-\sum_{F\in\mathcal{F}_h}\int_F\{\mathbf{v}_h\}[\![\underline{\underline{C}}_K\underline{\underline{\xi}}\cdot\mathbf{n}]\!]\\&+\sum_{F\in\mathcal{F}_h}\int_F\frac{1}{\mathbf{S}^++\mathbf{S}^-}[\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!][\![\underline{\underline{C}}_K\underline{\underline{\xi}}\cdot\mathbf{n}]\!]-\sum_{F\in\mathcal{F}_h}\int_F\frac{\left(\mathbf{S}^+-\mathbf{S}^-\right)\left(\mathbf{v}_h^+-\mathbf{v}_h^-\right)}{2\left(\mathbf{S}^++\mathbf{S}^-\right)}[\![\underline{\underline{C}}_K\underline{\underline{\xi}}\cdot\mathbf{n}]\!]=\sum_{K\in\mathcal{T}_h}\int_K\mathbf{f}:\underline{\underline{\xi}}.\end{aligned} \quad (3.2.19)$$

Considering now equation (3.2.16), we develop the surface terms as follow

$$\begin{aligned}\sum_{K\in\mathcal{T}_h}\int_{\partial K}\underline{\underline{\sigma}}_h\cdot\mathbf{n}\cdot\mathbf{w}-\int_{\partial K}\mathbf{S}\left(\mathbf{v}_h-\lambda_h\right)\cdot\mathbf{w} &= \sum_{F\in\mathcal{F}_h}\int_F[\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}\cdot\mathbf{w}]\!]-\sum_{F\in\mathcal{F}_h}\int_F\left(\mathbf{S}^+\mathbf{v}_h^+\cdot\mathbf{w}^++\mathbf{S}^-\mathbf{v}_h^-\cdot\mathbf{w}^-\right)\\ &\quad+\sum_{F\in\mathcal{F}_h}\int_F\lambda_h\cdot\left(\mathbf{S}^+\mathbf{w}^++\mathbf{S}^-\mathbf{w}^-\right)\\ &= \sum_{F\in\mathcal{F}_h}\int_F\Big([\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!]\cdot\{\mathbf{w}\}+\{\underline{\underline{\sigma}}_h\}\cdot[\![\mathbf{w}\cdot\mathbf{n}]\!]\Big)\\ &\quad-\int_F\left(\mathbf{S}^+\mathbf{v}_h^+\cdot\mathbf{w}^++\mathbf{S}^-\mathbf{v}_h^-\cdot\mathbf{w}^-\right)+\int_F\lambda_h\cdot\left(\mathbf{S}^+\mathbf{w}^++\mathbf{S}^-\mathbf{w}^-\right).\end{aligned}$$

If the interpolation degree of λ_h is higher than the one of $\mathbf{w}$, then there exists $\varphi\in\mathbf{M}_h$ such that $\varphi|_F=\left(\mathbf{S}^+\mathbf{w}^++\mathbf{S}^-\mathbf{w}^-\right)$ on each internal face F. Thus, we write

$$\sum_{K\in\mathcal{T}_h}\int_{\partial K}\underline{\underline{\sigma}}_h\cdot\mathbf{n}\cdot\mathbf{w}-\int_{\partial K}\mathbf{S}\left(\mathbf{v}_h-\lambda_h\right)\cdot\mathbf{w}=A+B+C+D,$$

where

$$\begin{aligned}A &= \sum_{F\in\mathcal{F}_h}\int_F\Big([\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!]\cdot\{\mathbf{w}\}+\{\underline{\underline{\sigma}}_h\}\cdot[\![\mathbf{w}\cdot\mathbf{n}]\!]\Big),\\ B &= -\sum_{F\in\mathcal{F}_h}\int_F\left(\mathbf{S}^+\mathbf{v}_h^+\cdot\mathbf{w}^++\mathbf{S}^-\mathbf{v}_h^-\cdot\mathbf{w}^-\right),\\ C &= -\sum_{F\in\mathcal{F}_h}\int_F\frac{1}{\mathbf{S}^++\mathbf{S}^-}[\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!]\cdot\left(\mathbf{S}^+\mathbf{w}^++\mathbf{S}^-\mathbf{w}^-\right),\\ D &= \sum_{F\in\mathcal{F}_h}\int_F\frac{\mathbf{S}^+\mathbf{v}_h^++\mathbf{S}^-\mathbf{v}_h^-}{\mathbf{S}^++\mathbf{S}^-}\cdot\left(\mathbf{S}^+\mathbf{w}^++\mathbf{S}^-\mathbf{w}^-\right).\end{aligned}$$

Knowing that

$$\mathbf{S}^+\mathbf{w}^++\mathbf{S}^-\mathbf{w}^-=\frac{\mathbf{S}^++\mathbf{S}^-}{2}\left(\mathbf{w}^++\mathbf{w}^-\right)+\frac{\mathbf{S}^+-\mathbf{S}^-}{2}\left(\mathbf{w}^+-\mathbf{w}^-\right),$$

we express C as

$$\begin{aligned}C &= -\sum_{F\in\mathcal{F}_h}\int_F\frac{1}{\mathbf{S}^++\mathbf{S}^-}[\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!]\cdot\left(\mathbf{S}^+\mathbf{w}^++\mathbf{S}^-\mathbf{w}^-\right)\\ &= -\sum_{F\in\mathcal{F}_h}\left(\int_F[\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!]\cdot\{\mathbf{w}\}+\int_F\frac{\mathbf{S}^+-\mathbf{S}^-}{2\left(\mathbf{S}^++\mathbf{S}^-\right)}\left(\mathbf{w}^+-\mathbf{w}^-\right)\cdot[\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!]\right).\end{aligned}$$

Similarly, for B, we have

$$\begin{aligned}B &= -\sum_{F\in\mathcal{F}_h}\int_F\left(\mathbf{S}^+\mathbf{v}_h^+\cdot\mathbf{w}^++\mathbf{S}^-\mathbf{v}_h^-\cdot\mathbf{w}^-\right)\\ &= -\sum_{F\in\mathcal{F}_h}\left(\int_F\frac{\mathbf{v}_h^++\mathbf{v}_h^-}{2}\cdot\left(\mathbf{S}^+\mathbf{w}^++\mathbf{S}^-\mathbf{w}^-\right)+\int_F\frac{\mathbf{v}_h^+-\mathbf{v}_h^-}{2}\cdot\left(\mathbf{S}^+\mathbf{w}^+-\mathbf{S}^-\mathbf{w}^-\right)\right).\end{aligned}$$

Thus, the sum $B + D$ becomes

$$
\begin{aligned}
B + D &= -\sum_{F\in\mathcal{F}_h}\int_F \left(\mathbf{S}^+\mathbf{v}_h^+\cdot\mathbf{w}^+ + \mathbf{S}^-\mathbf{v}_h^-\cdot\mathbf{w}^-\right) + \sum_{F\in\mathcal{F}_h}\int_F \frac{\mathbf{S}^+\mathbf{v}_h^+ + \mathbf{S}^-\mathbf{v}_h^-}{\mathbf{S}^+ + \mathbf{S}^-}\cdot\left(\mathbf{S}^+\mathbf{w}^+ + \mathbf{S}^-\mathbf{w}^-\right)\\
&= -\sum_{F\in\mathcal{F}_h}\int_F \frac{\mathbf{v}_h^+ + \mathbf{v}_h^-}{2}\cdot\left(\mathbf{S}^+\mathbf{w}^+ + \mathbf{S}^-\mathbf{w}^-\right) - \sum_{F\in\mathcal{F}_h}\int_F \frac{\mathbf{v}_h^+ - \mathbf{v}_h^-}{2}\cdot\left(\mathbf{S}^+\mathbf{w}^+ - \mathbf{S}^-\mathbf{w}^-\right)\\
&\quad + \sum_{F\in\mathcal{F}_h}\int_F \frac{\mathbf{v}_h^+ + \mathbf{v}_h^-}{2}\cdot\left(\mathbf{S}^+\mathbf{w}^+ + \mathbf{S}^-\mathbf{w}^-\right) + \sum_{F\in\mathcal{F}_h}\int_F \frac{\left(\mathbf{S}^+ - \mathbf{S}^-\right)\left(\mathbf{v}_h^+ - \mathbf{v}_h^-\right)}{2\left(\mathbf{S}^+ + \mathbf{S}^-\right)}\cdot\left(\mathbf{S}^+\mathbf{w}^+ + \mathbf{S}^-\mathbf{w}^-\right)\\
&= -\sum_{F\in\mathcal{F}_h}\int_F \frac{\left(\mathbf{S}^+ - \mathbf{S}^-\right)\left(\mathbf{v}_h^+ - \mathbf{v}_h^-\right)}{4}\left(\mathbf{w}^+ + \mathbf{w}^-\right) - \sum_{F\in\mathcal{F}_h}\int_F \frac{\left(\mathbf{v}_h^+ - \mathbf{v}_h^-\right)\left(\mathbf{S}^+ + \mathbf{S}^-\right)}{4}\cdot\left(\mathbf{w}^+ - \mathbf{w}^-\right)\\
&\quad + \sum_{F\in\mathcal{F}_h}\int_F \frac{\left(\mathbf{S}^+ - \mathbf{S}^-\right)\left(\mathbf{v}_h^+ - \mathbf{v}_h^-\right)}{2\left(\mathbf{S}^+ + \mathbf{S}^-\right)}\cdot\frac{\mathbf{S}^+ + \mathbf{S}^-}{2}\left(\mathbf{w}^+ + \mathbf{w}^-\right)\\
&\quad + \sum_{F\in\mathcal{F}_h}\int_F \frac{\left(\mathbf{S}^+ - \mathbf{S}^-\right)\left(\mathbf{v}_h^+ - \mathbf{v}_h^-\right)}{4}\cdot\left(\mathbf{w}^+ - \mathbf{w}^-\right)\\
&= -\sum_{F\in\mathcal{F}_h}\int_F \{\mathbf{S}\}\frac{\left(\mathbf{v}_h^+ - \mathbf{v}_h^-\right)}{2}\cdot\left(\mathbf{w}^+ - \mathbf{w}^-\right) + \sum_{F\in\mathcal{F}_h}\int_F \frac{\left(\mathbf{S}^+ - \mathbf{S}^-\right)^2\left(\mathbf{v}_h^+ - \mathbf{v}_h^-\right)}{4\left(\mathbf{S}^+ + \mathbf{S}^-\right)}\cdot\left(\mathbf{w}^+ - \mathbf{w}^-\right).
\end{aligned}
$$

Finally,

$$
\begin{aligned}
A + B + C + D = &\sum_{F\in\mathcal{F}_h}\int_F \{\underline{\underline{\sigma}}_h\}\cdot[\![\mathbf{w}\cdot\mathbf{n}]\!] - \int_F \frac{\mathbf{S}^+ - \mathbf{S}^-}{2\left(\mathbf{S}^+ + \mathbf{S}^-\right)}\left(\mathbf{w}^+ - \mathbf{w}^-\right)\cdot[\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!]\\
&- \int_F \{\mathbf{S}\}\frac{\left(\mathbf{v}_h^+ - \mathbf{v}_h^-\right)}{2}\cdot\left(\mathbf{w}^+ - \mathbf{w}^-\right) + \int_F \frac{\left(\mathbf{S}^+ - \mathbf{S}^-\right)^2\left(\mathbf{v}_h^+ - \mathbf{v}_h^-\right)}{4\left(\mathbf{S}^+ + \mathbf{S}^-\right)}\cdot\left(\mathbf{w}^+ - \mathbf{w}^-\right),
\end{aligned}
$$

and equation (3.2.16) becomes

$$
\begin{aligned}
&\sum_{K\in\mathcal{T}_h}\int_K i\omega\rho_K\mathbf{v}_h\cdot\mathbf{w} + \sum_{K\in\mathcal{T}_h}\int_K \underline{\underline{\sigma}}_h : (\nabla\cdot\mathbf{w}) + \sum_{F\in\mathcal{F}_h}\int_F \{\underline{\underline{\sigma}}_h\}\cdot[\![\mathbf{w}\cdot\mathbf{n}]\!]\\
&- \sum_{F\in\mathcal{F}_h}\int_F \frac{\mathbf{S}^+ - \mathbf{S}^-}{2\left(\mathbf{S}^+ + \mathbf{S}^-\right)}\left(\mathbf{w}^+ - \mathbf{w}^-\right)\cdot[\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!] - \sum_{F\in\mathcal{F}_h}\int_F \{\mathbf{S}\}\frac{\left(\mathbf{v}_h^+ - \mathbf{v}_h^-\right)}{2}\cdot\left(\mathbf{w}^+ - \mathbf{w}^-\right) \qquad (3.2.20)\\
&+ \sum_{F\in\mathcal{F}_h}\int_F \frac{\left(\mathbf{S}^+ - \mathbf{S}^-\right)^2\left(\mathbf{v}_h^+ - \mathbf{v}_h^-\right)}{4\left(\mathbf{S}^+ + \mathbf{S}^-\right)}\cdot\left(\mathbf{w}^+ - \mathbf{w}^-\right) = 0
\end{aligned}
$$

Hence, $\mathbf{v}_h$ and $\underline{\underline{\sigma}}_h$ are solutions of a classical upwind scheme. The HDG system has properties similar to the upwind DG scheme and thus, it can be studied following the method proposed in Ainsworth, Monk and Muniz [64].

Finally, let us remark that, in equations (3.2.19) and (3.2.20), if $\mathbf{S}^+ = \mathbf{S}^- = \mathbf{S}$, we obtain

$$\left\{\begin{array}{ll} \displaystyle\sum_{K\in\mathcal{T}_h}\int_K i\omega\rho_K\mathbf{v}_h\cdot\mathbf{w} & + \displaystyle\sum_{K\in\mathcal{T}_h}\int_K \underline{\underline{\sigma}}_h : (\nabla\cdot\mathbf{w}) + \sum_{F\in\mathcal{F}_h}\int_F \{\underline{\underline{\sigma}}_h\}\cdot[\![\mathbf{w}\cdot\mathbf{n}]\!] \\ & \displaystyle - \sum_{F\in\mathcal{F}_h}\int_F \{\mathbf{S}\}\frac{\left(\mathbf{v}_h^+ - \mathbf{v}_h^-\right)}{2}\cdot\left(\mathbf{w}^+ - \mathbf{w}^-\right) = 0 \\ \displaystyle\sum_{K\in\mathcal{T}_h}\int_K i\omega\underline{\underline{\sigma}}_h : \underline{\underline{\xi}} & + \displaystyle\sum_{K\in\mathcal{T}_h}\int_K \mathbf{v}_h\cdot\nabla\cdot\underline{\underline{C}}_K\underline{\underline{\xi}} - \sum_{F\in\mathcal{F}_h}\int_F \{\mathbf{v}_h\}[\![\underline{\underline{C}}_K\underline{\underline{\xi}}\cdot\mathbf{n}]\!] \\ & \displaystyle + \sum_{F\in\mathcal{F}_h}\int_F \frac{1}{2\mathbf{S}}[\![\underline{\underline{\sigma}}_h\cdot\mathbf{n}]\!][\![\underline{\underline{C}}_K\underline{\underline{\xi}}\cdot\mathbf{n}]\!] = \sum_{K\in\mathcal{T}_h}\int_K \mathbf{f}:\underline{\underline{\xi}}. \end{array}\right. \tag{3.2.21}$$

In that case, the stabilization parameter introduces numerical dissipation in the scheme.

3.3 Implementation

To solve the HDG system (3.2.12), we have to implement the local system (3.2.13). To do that we need to discretize this local system. We first consider the case of an isotropic medium and then, the case of an anisotropic medium. The discretization of the boundary conditions requires a special attention. In the last part of this section, we give the main lines of the HDG implementation.

3.3.1 Discretization for the isotropic case

For the discretization of the local equations (3.2.13), we first consider the simple isotropic case and assume the right-hand side $\mathbf{f}$ is equal to zero.
Taking as test function the basis function φ_i^K, we develop the local equations (3.2.13) and write the local solution $(\mathbf{v}^K, \underline{\underline{\sigma}}^K)$ as a function of λ (simplified notation for λ_h). In order to avoid confusion with λ the Lamé's coefficient and λ the Lagrange multiplier, we denote by λ_L the Lamé's coefficient.

$$\left\{\begin{array}{rcl} \displaystyle\int_K i\omega\rho_K v_x^K\varphi_i^K - \int_K \frac{\partial\sigma_{xx}^K}{\partial x}\varphi_i^K - \int_K \frac{\partial\sigma_{xz}^K}{\partial z}\varphi_i^K + \int_{\partial K}\tau^K v_x^K\varphi_i^K - \int_{\partial K}\tau^K\lambda_x\varphi_i^K &=& 0, \\ \displaystyle\int_K i\omega\rho_K v_z^K\varphi_i^K - \int_K \frac{\partial\sigma_{xz}^K}{\partial x}\varphi_i^K - \int_K \frac{\partial\sigma_{zz}^K}{\partial z}\varphi_i^K + \int_{\partial K}\tau^K v_z^K\varphi_i^K - \int_{\partial K}\tau^K\lambda_z\varphi_i^K &=& 0, \\ \displaystyle\int_K i\omega\sigma_{xx}^K\varphi_i^K + \int_K (\lambda_L+2\mu)\, v_x^K\frac{\partial\varphi_i^K}{\partial x} + \int_K \lambda_L v_z^K\frac{\partial\varphi_i^K}{\partial z} & & \\ \qquad\qquad - \displaystyle\int_{\partial K}(\lambda_L+2\mu)\,\lambda_x\varphi_i^K n_x - \int_{\partial K}\lambda_L\lambda_z\varphi_i^K n_z &=& 0, \\ \displaystyle\int_K i\omega\sigma_{zz}^K\varphi_i^K + \int_K \lambda_L v_x^K\frac{\partial\varphi_i^K}{\partial x} + \int_K (\lambda_L+2\mu)\, v_z^K\frac{\partial\varphi_i^K}{\partial z} & & \\ \qquad\qquad - \displaystyle\int_{\partial K}\lambda_L\lambda_x\varphi_i^K n_x - \int_{\partial K}(\lambda_L+2\mu)\,\lambda_z\varphi_i^K n_z &=& 0, \\ \displaystyle\int_K i\omega\sigma_{xz}^K\varphi_i^K + \int_K \mu v_x^K\frac{\partial\varphi_i^K}{\partial z} + \int_K \mu v_z^K\frac{\partial\varphi_i^K}{\partial x} - \int_{\partial K}\mu\lambda_x\varphi_i^K n_z - \int_{\partial K}\mu\lambda_z\varphi_i^K n_x &=& 0. \end{array}\right. \tag{3.3.1}$$

For an element K, we decompose the local solutions $(\mathbf{v}^K, \underline{\underline{\sigma}}^K)$ as follow

$$\begin{aligned} v_l^K &= \sum_{j=1}^{d_i^K}\underline{v}_{l,j}^K\varphi_j^K, \qquad l = x, z \\ \sigma_{kl}^K &= \sum_{j=1}^{d_i^K}\underline{\sigma}_{kl,j}^K\varphi_j^K, \qquad k, l = x, z \end{aligned} \tag{3.3.2}$$

where φ_j^K are the basis functions of $\mathbb{P}_p(K)$ and d_i^K the associated number of degrees of freedom. For a face F, λ is represented by

$$\lambda_l^F = \sum_{j=1}^{d_i^F} \underline{\lambda}_{l,j}^F \psi_j^F, \qquad l = x, z \tag{3.3.3}$$

where ψ_j^F are the basis functions of $\mathbb{P}_p(F)$ and d_i^F the associated number of degrees of freedom. We denote by $\beta(K,l)$ the global index of the l-th face of the element K $(l = 1, 2, 3)$. For example, if the l-th face of K is the j-th face F_j then $\beta(K,l) = j$ (see fig. 3.3.1).
Conversely, if F_j is the common face between K and K', we define $\zeta(j,+) = K$ and $\zeta(j,-) = K'$.

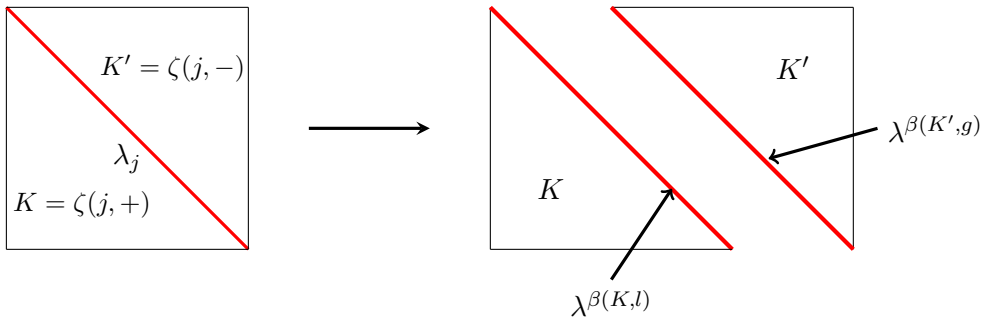


Fig. 3.3.1: Definitions of $\beta(K,l)$ and $\zeta(j,+)$.

After discretization, the local linear system resulting from (3.3.1) is

$$\begin{cases}
i\omega\rho\mathbb{M}^K \underline{v}_x^K - \mathbb{D}_x^{K^T} \underline{\sigma}_{xx}^K - \mathbb{D}_z^{K^T} \underline{\sigma}_{xz}^K + \sum_{l=1}^{3} \tau^{(K,l)} \mathbb{E}_l^K \underline{v}_x^K - \sum_{l=1}^{3} \tau^{(K,l)} \mathbb{F}_l^K \underline{\lambda}_x^{\beta(K,l)} = 0, \\
i\omega\rho\mathbb{M}^K \underline{v}_z^K - \mathbb{D}_x^{K^T} \underline{\sigma}_{xz}^K - \mathbb{D}_z^{K^T} \underline{\sigma}_{zz}^K + \sum_{l=1}^{3} \tau^{(K,l)} \mathbb{E}_l^K \underline{v}_z^K - \sum_{l=1}^{3} \tau^{(K,l)} \mathbb{F}_l^K \underline{\lambda}_z^{\beta(K,l)} = 0, \\
i\omega\mathbb{M}^K \underline{\sigma}_{xx}^K + (\lambda_L + 2\mu)\mathbb{D}_x^K \underline{v}_x^K + \lambda_L \mathbb{D}_z^K \underline{v}_z^K - \sum_{l=1}^{3} (\lambda_L + 2\mu)\underline{\lambda}_x^{\beta(K,l)} \mathbb{Q}_{xl}^K - \sum_{l=1}^{3} \lambda_L \underline{\lambda}_z^{\beta(K,l)} \mathbb{Q}_{zl}^K = 0, \\
i\omega\mathbb{M}^K \underline{\sigma}_{zz}^K + (\lambda_L + 2\mu)\mathbb{D}_z^K \underline{v}_z^K + \lambda_L \mathbb{D}_x^K \underline{v}_x^K - \sum_{l=1}^{3} (\lambda_L + 2\mu)\underline{\lambda}_z^{\beta(K,l)} \mathbb{Q}_{zl}^K - \sum_{l=1}^{3} \lambda_L \underline{\lambda}_x^{\beta(K,l)} \mathbb{Q}_{xl}^K = 0, \\
i\omega\mathbb{M}^K \underline{\sigma}_{xz}^K + \mu \left(\mathbb{D}_x^K \underline{v}_z^K + \mathbb{D}_z^K \underline{v}_x^K \right) - \sum_{l=1}^{3} \mu \left(\underline{\lambda}_x^{\beta(K,l)} \mathbb{Q}_{zl}^K + \underline{\lambda}_z^{\beta(K,l)} \mathbb{Q}_{xl}^K \right) = 0.
\end{cases} \tag{3.3.4}$$

The entries of the local matrices are defined by

$$\begin{cases} \mathbb{M}^K_{ij} = \displaystyle\int_K \varphi^K_i \varphi^K_j \, d\mathbf{x}; \\ \mathbb{D}^K_{u,ij} = \displaystyle\int_K \varphi^K_i \partial_u \varphi^K_j \, d\mathbf{x}, \quad \text{with } u = x, z; \\ \mathbb{E}^K_{l,ij} = \displaystyle\int_{\partial K^l} \varphi^K_i \varphi^K_j \, ds; \\ \mathbb{F}^K_{l,ij} = \displaystyle\int_{\partial K^l} \psi^{\beta(K,l)}_i \varphi^K_j \, ds; \\ \mathbb{Q}^K_{ul,ij} = \displaystyle\int_{\partial K^l} n^K_u \psi^{\beta(K,l)}_i \varphi^K_j, \quad \text{with } u = x, z. \end{cases}$$

where ∂K^l denotes the face of index l of the element K.

From (3.3.4) we define the unknowns vector $\underline{W}^K = \left(\underline{v}^K_x, \ \underline{v}^K_z, \ \underline{\sigma}^K_{xx}, \ \underline{\sigma}^K_{zz}, \ \underline{\sigma}^K_{xz}\right)^T$ and the vector $\underline{\Lambda}^K = \left(\underline{\lambda}^{\beta(K,1)}_x, \ \underline{\lambda}^{\beta(K,2)}_x, \ \underline{\lambda}^{\beta(K,3)}_x, \ \underline{\lambda}^{\beta(K,1)}_z, \ \underline{\lambda}^{\beta(K,2)}_z, \ \underline{\lambda}^{\beta(K,3)}_z\right)^T$.
The local linear system on the element K can be written as

$$\mathbb{A}^K \underline{W}^K + \mathbb{C}^K \underline{\Lambda}^K = 0, \tag{3.3.5}$$

with

$$\mathbb{A}^K = \begin{bmatrix} i\omega\rho\mathbb{M}^K + \displaystyle\sum_{l=1}^{3} \tau^{(K,l)} \mathbb{E}^K_l & 0 & -\mathbb{D}^{K^T}_x & 0 & -\mathbb{D}^{K^T}_z \\ 0 & i\omega\rho\mathbb{M}^K + \displaystyle\sum_{l=1}^{3} \tau^{(K,l)} \mathbb{E}^K_l & 0 & -\mathbb{D}^{K^T}_z & -\mathbb{D}^{K^T}_x \\ (\lambda_L + 2\mu)\mathbb{D}^K_x & \lambda_L \mathbb{D}^K_z & i\omega\mathbb{M}^K & 0 & 0 \\ \lambda_L \mathbb{D}^K_x & (\lambda_L + 2\mu)\mathbb{D}^K_z & 0 & i\omega\mathbb{M}^K & 0 \\ \mu\mathbb{D}^K_z & \mu\mathbb{D}^K_x & 0 & 0 & i\omega\mathbb{M}^K \end{bmatrix},$$

$$\mathbb{C}^K = -\begin{bmatrix} \tau^{(K,1)}\mathbb{F}^K_1 & \tau^{(K,2)}\mathbb{F}^K_2 & \tau^{(K,3)}\mathbb{F}^K_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tau^{(K,1)}\mathbb{F}^K_1 & \tau^{(K,2)}\mathbb{F}^K_2 & \tau^{(K,3)}\mathbb{F}^K_3 \\ (\lambda_L + 2\mu)\mathbb{Q}^K_{x1} & (\lambda_L + 2\mu)\mathbb{Q}^K_{x2} & (\lambda_L + 2\mu)\mathbb{Q}^K_{x3} & \lambda_L \mathbb{Q}^K_{z1} & \lambda_L \mathbb{Q}^K_{z2} & \lambda_L \mathbb{Q}^K_{z3} \\ \lambda_L \mathbb{Q}^K_{x1} & \lambda_L \mathbb{Q}^K_{x2} & \lambda_L \mathbb{Q}^K_{x3} & (\lambda_L + 2\mu)\mathbb{Q}^K_{z1} & (\lambda_L + 2\mu)\mathbb{Q}^K_{z2} & (\lambda_L + 2\mu)\mathbb{Q}^K_{z3} \\ \mu\mathbb{Q}^K_{z1} & \mu\mathbb{Q}^K_{z2} & \mu\mathbb{Q}^K_{z3} & \mu\mathbb{Q}^K_{x1} & \mu\mathbb{Q}^K_{x2} & \mu\mathbb{Q}^K_{x3} \end{bmatrix}.$$

We consider now the discretization of the transmission condition (the last equation of (3.2.12)). We remind that the transmission condition on a face $F_j = \partial K \cap \partial K'$ such as $j = \beta(K,l) = \beta(K',g)$ is written as, $\forall \eta \in \mathbf{M}_h$,

$$\int_F \left(\underline{\underline{\sigma}}_h^K \cdot \mathbf{n}^K \cdot \eta + \underline{\underline{\sigma}}_h^{K'} \cdot \mathbf{n}^{K'} \cdot \eta - \mathbf{S}^K \left(\mathbf{v}_h^K - \lambda_h \right) \cdot \eta - \mathbf{S}^{K'} \left(\mathbf{v}_h^{K'} - \lambda_h \right) \cdot \eta \right) = 0.$$

The resulting discretization is given by the two following systems

$$\begin{aligned} &\mathbb{Q}_{xl}^{K^T} \underline{\sigma}_{xx}^K + \mathbb{Q}_{zl}^{K^T} \underline{\sigma}_{xz}^K - \tau^{(K,l)} \mathbb{F}_l^{K^T} \underline{v}_x^K + \tau^{(K,l)} \mathbb{G}^j \underline{\lambda}_x^{\beta(K,l)} \\ &+ \mathbb{Q}_{xl}^{K'^T} \underline{\sigma}_{xx}^{K'} + \mathbb{Q}_{zl}^{K'^T} \underline{\sigma}_{xz}^{K'} - \tau^{(K',l)} \mathbb{F}_l^{K'^T} \underline{v}_x^{K'} + \tau^{(K',l)} \mathbb{G}^j \underline{\lambda}_x^{\beta(K^-,l)} = 0, \end{aligned} \tag{3.3.6}$$

and

$$\begin{aligned} &\mathbb{Q}_{xl}^{K^T} \underline{\sigma}_{xz}^K + \mathbb{Q}_{zl}^{K^T} \underline{\sigma}_{zz}^K - \tau^{(K,l)} \mathbb{F}_l^{K^T} \underline{v}_z^K + \tau^{(K,l)} \mathbb{G}^j \underline{\lambda}_z^{\beta(K,l)} \\ &+ \mathbb{Q}_{xl}^{K'^T} \underline{\sigma}_{xz}^{K'} + \mathbb{Q}_{zl}^{K'^T} \underline{\sigma}_{xz}^{K'} - \tau^{(K',l)} \mathbb{F}_l^{K'^T} \underline{v}_z^{K'} + \tau^{(K',l)} \mathbb{G}^j \underline{\lambda}_z^{\beta(K',l)} = 0, \end{aligned} \tag{3.3.7}$$

where

$$\mathbb{G}_{im}^j = \int_{F_j} \psi_i \psi_m ds.$$

From (3.3.6) and (3.3.7), we write a local system for λ:

$$\mathbb{B}^K \underline{W}^K + \mathbb{L}^K \underline{\Lambda}^K + \mathcal{R}^K = 0, \tag{3.3.8}$$

where

$$\mathbb{B}^K = \begin{bmatrix} -\tau^{(K,1)} \mathbb{F}_1^{K^T} & 0 & \mathbb{Q}_{x1}^{K^T} & 0 & \mathbb{Q}_{z1}^{K^T} \\ -\tau^{(K,2)} \mathbb{F}_2^K & 0 & \mathbb{Q}_{x2}^{K^T} & 0 & \mathbb{Q}_{z2}^{K^T} \\ -\tau^{(K,3)} \mathbb{F}_3^{K^T} & 0 & \mathbb{Q}_{x3}^{K^T} & 0 & \mathbb{Q}_{z3}^{K^T} \\ 0 & -\tau^{(K,1)} \mathbb{F}_1^{K^T} & 0 & \mathbb{Q}_{z1}^{K^T} & \mathbb{Q}_{x1}^{K^T} \\ 0 & -\tau^{(K,2)} \mathbb{F}_2^{K^T} & 0 & \mathbb{Q}_{z2}^{K^T} & \mathbb{Q}_{x2}^{K^T} \\ 0 & -\tau^{(K,3)} \mathbb{F}_3^{K^T} & 0 & \mathbb{Q}_{z3}^{K^T} & \mathbb{Q}_{x3}^{K^T} \end{bmatrix},$$

$$\mathbb{L}^K = \begin{bmatrix} \tau^{(K,1)} \mathbb{G}^{\beta(K,1)} & 0 & 0 & 0 & 0 & 0 \\ 0 & \tau^{(K,2)} \mathbb{G}^{\beta(K,2)} & 0 & 0 & 0 & 0 \\ 0 & 0 & \tau^{(K,3)} \mathbb{G}^{\beta(K,3)} & 0 & 0 & 0 \\ 0 & 0 & 0 & \tau^{(K,1)} \mathbb{G}^{\beta(K,1)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \tau^{(K,2)} \mathbb{G}^{\beta(K,2)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \tau^{(K,3)} \mathbb{G}^{\beta(K,3)} \end{bmatrix},$$

and $\mathcal{R}^K$ gathers the contributions from the neighboring elements.
If the neighboring elements of K are K_1, K_2 and K_3 with the common edges $l = 1, 2$, and 3 respectively, we define

$$\mathcal{R}^K = \begin{bmatrix} \mathcal{R}_x^{K_1} \\ \mathcal{R}_x^{K_2} \\ \mathcal{R}_x^{K_3} \\ \mathcal{R}_z^{K_1} \\ \mathcal{R}_z^{K_2} \\ \mathcal{R}_z^{K_3} \end{bmatrix},$$

where

$$\mathcal{R}_x^{K_i} = \begin{bmatrix} -\tau^{(K_i,i)}\mathbb{F}_i^{K_i^T} & 0 & \mathbb{Q}_{xi}^{K_i^T} & 0 & \mathbb{Q}_{zi}^{K_i^T} \end{bmatrix} \underline{W}^{K_i} + \tau^{(K_i,i)}\mathbb{G}^j \underline{\lambda}_x^{\beta(K_i,i)},$$

and

$$\mathcal{R}_z^{K_i} = \begin{bmatrix} 0 & -\tau^{(K_i,i)}\mathbb{F}_i^{K_i^T} & 0 & \mathbb{Q}_{zi}^{K_i^T} & \mathbb{Q}_{xi}^{K_i^T} \end{bmatrix} \underline{W}^{K_i} + \tau^{(K_i,i)}\mathbb{G}^j \underline{\lambda}_z^{\beta(K_i,i)}.$$

Definition. *We denote by $\underline{\Lambda}$ the vector which gathers all the degrees of freedom of $\underline{\lambda}$ for each face*

$$\underline{\Lambda} = \left(\underline{\lambda}^1, ..., \underline{\lambda}^{N_f}\right)^T,$$

where N_f is the number of faces of the mesh. We write $\overline{N}_\lambda$ the size of $\underline{\Lambda}$.
We define the local trace space spreading operator $\mathcal{A}_{HDG}^K$ as a matrix which allows to transform the unique global trace space values $\underline{\Lambda}$ onto the vector $\underline{\Lambda}^K$ of the degrees of freedom localized on the faces of K, i.e:

$$\mathcal{A}_{HDG}^K \underline{\Lambda} = \underline{\Lambda}^K.$$

If $N_\lambda^{(K,l)}$ is the number of degrees of freedom of the l-th face of K and $\overline{N}_\lambda^K$ the total number of degrees of freedom of Λ^K, then

$$\overline{N}_\lambda^K = \sum_{l=1}^{3} N_\lambda^{(K,l)}.$$

The size of $\mathcal{A}_{HDG}^K$ is thus $\overline{N}_\lambda^K \times \overline{N}_\lambda$. It only contains 0 and 1 and, for each element K, it extracts the degrees of freedom Λ located on the faces of K
Finally, we define the trace space spreading operator $\mathcal{A}_{HDG}$ such as

$$\mathcal{A}_{HDG} = \begin{pmatrix} \mathcal{A}_{HDG}^1 \\ \vdots \\ \mathcal{A}_{HDG}^{N_e} \end{pmatrix}.$$

$\mathcal{A}_{HDG}$ can be seen as a connectivity table linking the local degrees of freedom on the element K to the global degrees of freedom of the Lagrange multiplier Λ.

Remark. *We will compare the total number of degrees of freedom of the HDG method and the one of the upwind fluxes DG method in the next chapter.*

We rewrite (3.3.16) as

$$\mathbb{A}^K \underline{W}^K + \mathbb{C}^K \mathcal{A}_{HDG}^K \underline{\Lambda} = 0, \tag{3.3.9}$$

and consequently we can express $\underline{W}^K$ in terms of $\underline{\Lambda}$

$$\underline{W}^K = -(\mathbb{A}^K)^{-1}\mathbb{C}^K \mathcal{A}_{HDG}^K \underline{\Lambda}. \tag{3.3.10}$$

By summing all the equations of the transmission condition on all the faces of each element, element by element, we obtain

$$\sum_{K\in\mathcal{T}_h} (\mathcal{A}^K_{HDG})^T \left[\mathbb{B}^K \underline{W}^K + \mathbb{L}^K \mathcal{A}^K_{HDG}\underline{\Lambda}\right] = 0, \tag{3.3.11}$$

where the sum over all the elements along with the left application of the transpose of $\mathcal{A}^K_{HDG}$ allows to gather the element-wise contributions corresponding to faces.

By replacing $\underline{W}^K$ by (3.3.10) in (3.3.11), we obtain a global system in $\underline{\Lambda}$

$$\sum_{K\in\mathcal{T}_h} (\mathcal{A}^K_{HDG})^T \left[-\mathbb{B}^K(\mathbb{A}^K)^{-1}\mathbb{C}^K + \mathbb{L}^K\right] \mathcal{A}^K_{HDG}\underline{\Lambda} = 0. \tag{3.3.12}$$

Now, if we consider a point source on the element K_s, then the local equation (3.2.13) reads as

$$\left\{\begin{array}{rcl} \displaystyle\int_{K_s} i\omega\rho_{K_s}\mathbf{v}_h^{K_s}\cdot\mathbf{w} - \int_{K_s}\left(\nabla\cdot\underline{\underline{\sigma}}_h^{K_s}\right)\cdot\mathbf{w} + \int_{\partial K_s}\mathbf{S}\left(\mathbf{v}_h^{K_s}-\lambda_h\right)\cdot\mathbf{w} &=& 0,\\ \displaystyle\int_{K_s} i\omega\underline{\underline{\sigma}}_h^{K_s} : \underline{\underline{\xi}} + \int_{K_s}\mathbf{v}_h^{K_s}\cdot\nabla\cdot\left(\underline{\underline{C}}_{K_s}\underline{\underline{\xi}}\right) - \int_{\partial K_s}\lambda_h\cdot\underline{\underline{C}}_{K_s}\underline{\underline{\xi}}\cdot\mathbf{n} &=& \displaystyle\int_{K_s}\mathbf{f}:\underline{\underline{\xi}}, \end{array}\right.$$

and the corresponding discretization (3.3.9) becomes

$$\mathbb{A}^{K_s}\underline{W}^{K_s} + \mathbb{C}^{K_s}\mathcal{A}^{K_s}_{HDG}\underline{\Lambda} = \mathbb{S}^{K_s}.$$

That leads to

$$\underline{W}^{K_s} = \left(\mathbb{A}^{K_s}\right)^{-1}\mathbb{S}^{K_s} - \left(\mathbb{A}^{K_s}\right)^{-1}\mathbb{C}^{K_s}\mathcal{A}^{K_s}_{HDG}\underline{\Lambda}.$$

Finally, we obtain the following global system

$$\sum_{K\in\mathcal{T}_h} (\mathcal{A}^K_{HDG})^T \left[-\mathbb{B}^K(\mathbb{A}^K)^{-1}\mathbb{C}^K + \mathbb{L}^K\right] \mathcal{A}^K_{HDG}\underline{\Lambda} = \sum_{K\in\mathcal{T}_h} -(\mathcal{A}^K_{HDG})^T\mathbb{B}^K(\mathbb{A}^K)^{-1}\mathbb{S}^K. \tag{3.3.13}$$

3.3.2 Discretization for the anisotropic case

The discretization of the local problem in the anistropic case proceeds similarly. We take the basis function φ_i^K for the test function; we develop the local equations (3.2.13), where the coefficients of the tensor $\underline{\underline{C}}$ are the anisotropic coefficients; and we express the local solution $(\mathbf{v}^K, \underline{\underline{\sigma}}^K)$ as a function of λ.

The first step yields

$$
\begin{cases}
\displaystyle\int_K i\omega\rho_K v_x^K\varphi_i^K - \int_K \frac{\partial\sigma_{xx}^K}{\partial x}\varphi_i^K - \int_K \frac{\partial\sigma_{xz}^K}{\partial z}\varphi_i^K + \int_{\partial K}\tau^K v_x^K\varphi_i^K - \int_{\partial K}\tau^K\lambda_x\varphi_i^K = 0,\\
\displaystyle\int_K i\omega\rho_K v_z^K\varphi_i^K - \int_K \frac{\partial\sigma_{xz}^K}{\partial x}\varphi_i^K - \int_K \frac{\partial\sigma_{zz}^K}{\partial z}\varphi_i^K + \int_{\partial K}\tau^K v_z^K\varphi_i^K - \int_{\partial K}\tau^K\lambda_z\varphi_i^K = 0,\\
\displaystyle\int_K i\omega\sigma_{xx}^K\varphi_i^K + \int_K C_{11}v_x^K\frac{\partial\varphi_i^K}{\partial x} + \int_K C_{12}v_z^K\frac{\partial\varphi_i^K}{\partial z} + \int_K C_{13}\left(v_x^K\frac{\partial\varphi_i^K}{\partial z} + v_z^K\frac{\partial\varphi_i^K}{\partial x}\right)\\
\qquad\displaystyle - \int_{\partial K} C_{11}\lambda_x\varphi_i^K n_x - \int_{\partial K} C_{12}\lambda_z\varphi_i^K n_z - \int_{\partial K} C_{13}\left(\lambda_x\varphi_i^K n_z + \lambda_z\varphi_i^K n_x\right) = 0,\\
\displaystyle\int_K i\omega\sigma_{zz}^K\varphi_i^K + \int_K C_{12}v_x^K\frac{\partial\varphi_i^K}{\partial x} + \int_K C_{22}v_z^K\frac{\partial\varphi_i^K}{\partial z} + \int_K C_{23}\left(v_x^K\frac{\partial\varphi_i^K}{\partial z} + v_z^K\frac{\partial\varphi_i^K}{\partial x}\right)\\
\qquad\displaystyle - \int_{\partial K} C_{12}\lambda_x\varphi_i^K n_x - \int_{\partial K} C_{22}\lambda_z\varphi_i^K n_z - \int_{\partial K} C_{23}\left(\lambda_x\varphi_i^K n_z + \lambda_z\varphi_i^K n_x\right) = 0,\\
\displaystyle\int_K i\omega\sigma_{xz}^K\varphi_i^K + \int_K C_{13}v_x^K\frac{\partial\varphi_i^K}{\partial x} + \int_K C_{23}v_z^K\frac{\partial\varphi_i^K}{\partial z} + \int_K C_{33}\left(v_x^K\frac{\partial\varphi_i^K}{\partial z} + v_z^K\frac{\partial\varphi_i^K}{\partial x}\right)\\
\qquad\displaystyle - \int_{\partial K} C_{13}\lambda_x\varphi_i^K n_x - \int_{\partial K} C_{23}\lambda_z\varphi_i^K n_z - \int_{\partial K} C_{33}\left(\lambda_x\varphi_i^K n_z - \lambda_z\varphi_i^K n_x\right) = 0.
\end{cases}
\tag{3.3.14}
$$

Using the discretization given in section 3.3.1, the local linear system resulting from (3.3.14) is

$$
\begin{cases}
i\omega\rho\mathbb{M}^K\underline{v}_x^K - \mathbb{D}_x^{K^T}\underline{\sigma}_{xx}^K - \mathbb{D}_z^{K^T}\underline{\sigma}_{xz}^K + \displaystyle\sum_{l=1}^{3}\tau^{(K,l)}\mathbb{E}_l^K\underline{v}_x^K - \sum_{l=1}^{3}\tau^{(K,l)}\mathbb{F}_l^K\underline{\lambda}_x^{\beta(K,l)} = 0,\\
i\omega\rho\mathbb{M}^K\underline{v}_z^K - \mathbb{D}_x^{K^T}\underline{\sigma}_{xz}^K - \mathbb{D}_z^{K^T}\underline{\sigma}_{zz}^K + \displaystyle\sum_{l=1}^{3}\tau^{(K,l)}\mathbb{E}_l^K\underline{v}_z^K - \sum_{l=1}^{3}\tau^{(K,l)}\mathbb{F}_l^K\underline{\lambda}_z^{\beta(K,l)} = 0,\\
i\omega\mathbb{M}^K\underline{\sigma}_{xx}^K + C_{11}^K\mathbb{D}_x^K\underline{v}_x^K + C_{12}^K\mathbb{D}_z^K\underline{v}_z^K + C_{13}^K\left(\mathbb{D}_x^K\underline{v}_z^K + \mathbb{D}_z^K\underline{v}_x^K\right)\\
\qquad\displaystyle - \sum_{l=1}^{3}C_{11}^K\underline{\lambda}_x^{\beta(K,l)}\mathbb{Q}_{xl}^K - \sum_{l=1}^{3}C_{12}^K\underline{\lambda}_z^{\beta(K,l)}\mathbb{Q}_{zl}^K - \sum_{l=1}^{3}C_{13}^K\left(\underline{\lambda}_x^{\beta(K,l)}\mathbb{Q}_{zl}^K + \underline{\lambda}_z^{\beta(K,l)}\mathbb{Q}_{xl}^K\right) = 0,\\
i\omega\mathbb{M}^K\underline{\sigma}_{zz}^K + C_{12}^K\mathbb{D}_x^K\underline{v}_x^K + C_{22}^K\mathbb{D}_z^K\underline{v}_z^K + C_{23}^K\left(\mathbb{D}_x^K\underline{v}_z^K + \mathbb{D}_z^K\underline{v}_x^K\right)\\
\qquad\displaystyle - \sum_{l=1}^{3}C_{12}^K\underline{\lambda}_x^{\beta(K,l)}\mathbb{Q}_{xl}^K - \sum_{l=1}^{3}C_{22}^K\underline{\lambda}_z^{\beta(K,l)}\mathbb{Q}_{zl}^K - \sum_{l=1}^{3}C_{23}^K\left(\underline{\lambda}_x^{\beta(K,l)}\mathbb{Q}_{zl}^K + \underline{\lambda}_z^{\beta(K,l)}\mathbb{Q}_{xl}^K\right) = 0,\\
i\omega\mathbb{M}^K\underline{\sigma}_{xz}^K + C_{13}^K\mathbb{D}_x^K\underline{v}_x^K + C_{23}^K\mathbb{D}_z^K\underline{v}_z^K + C_{33}^K\left(\mathbb{D}_x^K\underline{v}_z^K + \mathbb{D}_z^K\underline{v}_x^K\right)\\
\qquad\displaystyle - \sum_{l=1}^{3}C_{13}^K\underline{\lambda}_x^{\beta(K,l)}\mathbb{Q}_{xl}^K - \sum_{l=1}^{3}C_{23}^K\underline{\lambda}_z^{\beta(K,l)}\mathbb{Q}_{zl}^K - \sum_{l=1}^{3}C_{33}^K\left(\underline{\lambda}_x^{\beta(K,l)}\mathbb{Q}_{zl}^K + \underline{\lambda}_z^{\beta(K,l)}\mathbb{Q}_{xl}^K\right) = 0.
\end{cases}
\tag{3.3.15}
$$

with matrices $\mathbb{M}^K, \mathbb{D}^K, \mathbb{E}^K, \mathbb{F}^K$ and $\mathbb{Q}^K$ defined in section 3.3.1. The local linear system on one element K is written in the same way as in the isotropic case

$$
\mathbb{A}^K\underline{W}^K + \mathbb{C}^K\underline{\Lambda}^K = 0. \tag{3.3.16}
$$

$\underline{W}^K$ and $\underline{\Lambda}^K$ are the same vectors than in section 3.3.1; only matrices $\mathbb{A}^K$ and $\mathbb{C}^K$ change

$$\mathbb{A}^K = \begin{bmatrix} i\omega\rho\mathbb{M}^K + \sum_{l=1}^{3}\tau^{(K,l)}\mathbb{E}_l^K & 0 & -\mathbb{D}_x^{K^T} & 0 & -\mathbb{D}_z^{K^T} \\ 0 & i\omega\rho\mathbb{M}^K + \sum_{l=1}^{3}\tau^{(K,l)}\mathbb{E}_l^K & 0 & -\mathbb{D}_z^{K^T} & -\mathbb{D}_x^{K^T} \\ C_{11}^K\mathbb{D}_x^K + C_{13}^K\mathbb{D}_z^K & C_{12}^K\mathbb{D}_z^K + C_{13}^K\mathbb{D}_x^K & i\omega\mathbb{M}^K & 0 & 0 \\ C_{12}^K\mathbb{D}_x^K + C_{23}^K\mathbb{D}_z^K & C_{22}^K\mathbb{D}_z^K + C_{23}^K\mathbb{D}_x^K & 0 & i\omega\mathbb{M}^K & 0 \\ C_{13}^K\mathbb{D}_x^K + C_{33}^K\mathbb{D}_z^K & C_{23}^K\mathbb{D}_z^K + C_{33}^K\mathbb{D}_x^K & 0 & 0 & i\omega\mathbb{M}^K \end{bmatrix},$$

$$\mathbb{C}^K = -\begin{bmatrix} \begin{bmatrix} \tau^{(K,1)}\mathbb{F}_1^K & \tau^{(K,2)}\mathbb{F}_2^K & \tau^{(K,3)}\mathbb{F}_3^K & ... \\ 0 & 0 & 0 & ... \\ C_{11}^K\mathbb{Q}_{x1}^K + C_{13}^K\mathbb{Q}_{z1}^K & C_{11}^K\mathbb{Q}_{x2}^K + C_{13}^K\mathbb{Q}_{z2}^K & C_{11}^K\mathbb{Q}_{x3}^K + C_{13}^K\mathbb{Q}_{z3}^K & ... \\ C_{12}^K\mathbb{Q}_{x1}^K + C_{23}^K\mathbb{Q}_{z1}^K & C_{12}^K\mathbb{Q}_{x2}^K + C_{23}^K\mathbb{Q}_{z2}^K & C_{12}^K\mathbb{Q}_{x3}^K + C_{23}^K\mathbb{Q}_{z3}^K & ... \\ C_{13}^K\mathbb{Q}_{x1}^K + C_{33}^K\mathbb{Q}_{z1}^K & C_{13}^K\mathbb{Q}_{x2}^K + C_{33}^K\mathbb{Q}_{z2}^K & C_{13}^K\mathbb{Q}_{x3}^K + C_{33}^K\mathbb{Q}_{z3}^K & ... \end{bmatrix} \\ \begin{bmatrix} ... & 0 & 0 & 0 \\ ... & \tau^{(K,1)}\mathbb{F}_1^K & \tau^{(K,2)}\mathbb{F}_2^K & \tau^{(K,3)}\mathbb{F}_3^K \\ ... & C_{12}^K\mathbb{Q}_{z1}^K + C_{13}^K\mathbb{Q}_{x1}^K & C_{12}^K\mathbb{Q}_{z2}^K + C_{13}^K\mathbb{Q}_{x2}^K & C_{12}^K\mathbb{Q}_{z3}^K + C_{13}^K\mathbb{Q}_{x3}^K \\ ... & C_{22}^K\mathbb{Q}_{z1}^K + C_{23}^K\mathbb{Q}_{x1}^K & C_{22}^K\mathbb{Q}_{z2}^K + C_{23}^K\mathbb{Q}_{x2}^K & C_{22}^K\mathbb{Q}_{z3}^K + C_{23}^K\mathbb{Q}_{x3}^K \\ ... & C_{23}^K\mathbb{Q}_{z1}^K + C_{33}^K\mathbb{Q}_{x1}^K & C_{23}^K\mathbb{Q}_{z2}^K + C_{33}^K\mathbb{Q}_{x2}^K & C_{23}^K\mathbb{Q}_{z3}^K + C_{33}^K\mathbb{Q}_{x3}^K \end{bmatrix} \end{bmatrix}.$$

The transmission condition on a face $F_j = \partial K \cap \partial K'$ is unchanged from the isotropic case and its local discretization stil reads as

$$\mathbb{B}^K\underline{W}^K + \mathbb{L}^K\underline{\Lambda}^K + \mathcal{R}^K = 0.$$

Matrices $\mathbb{B}^K$ and $\mathbb{L}^K$ and "operator" $\mathcal{R}^K$ are the same as in the isotropic case.

Finally, considering a source term non-equal to zero and using the trace space spreading operator $\mathcal{A}_{HDG}$, we obtain a similar global system as (3.3.13), where only the entries of matrices $\mathbb{A}^K$ and $\mathbb{C}^K$ change

$$\sum_{K\in\mathcal{T}_h}(\mathcal{A}_{HDG}^K)^T\left[-\mathbb{B}^K(\mathbb{A}^K)^{-1}\mathbb{C}^K + \mathbb{L}^K\right]\mathcal{A}_{HDG}^K\underline{\Lambda} = \sum_{K\in\mathcal{T}_h} -(\mathcal{A}_{HDG}^K)^T\mathbb{B}^K(\mathbb{A}^K)^{-1}\mathbb{S}^K.$$

3.3.3 Boundary conditions

To complete the HDG formulation, we have to describe the discretization of the transmission condition on a boundary face. We recall the boundary conditions (3.1.4) and (3.1.5) that we have chosen:

- The free surface condition on Γ_l : $\underline{\underline{\sigma}} \cdot \mathbf{n} = 0$
- The absorbing boundary condition on Γ_a : $\underline{\underline{\sigma}} \cdot \mathbf{n} + PA(\theta')P^T\mathbf{v} = 0$
  In 2D, the matrix P is reduced to

$$P = \begin{pmatrix} n_x & -n_z \\ n_z & n_x \end{pmatrix},$$

and the matrice $A(\theta)$ to

$$A(\theta) = \begin{pmatrix} -\rho v_p \dfrac{\kappa\cos^2\theta + \sin^2\theta}{\sqrt{\kappa^2\cos^2\theta + \sin^2\theta}} \left(\kappa\cos^2\theta + \sin^2\theta\right) & -\rho v_p \dfrac{\kappa\cos^2\theta + \sin^2\theta}{\sqrt{\kappa^2\cos^2\theta + \sin^2\theta}} \left(-(\kappa-1)\cos\theta\sin\theta\right) \\ -\rho v_p \dfrac{(\kappa-1)\cos\theta\sin\theta}{\sqrt{\kappa^2\cos^2\theta + \sin^2\theta}} \left(-(\kappa\cos^2\theta + \sin^2\theta)\right) & -\rho v_p \dfrac{(\kappa-1)\cos\theta\sin\theta}{\sqrt{\kappa^2\cos^2\theta + \sin^2\theta}} \left((\kappa-1)\cos\theta\sin\theta\right) - \rho v_s \end{pmatrix}.$$

The parameter θ is the tilted angle of the wave in the TTI medium, $\kappa = \sqrt{1-2\varepsilon_T}$, with ε_T one of Thomsen's constants defining the TTI nature of the medium and expressed by $\varepsilon_T = \dfrac{C_{11} - C_{13}}{2C_{33}}$.
Finally, in 2D, the angle θ' is defined such as

$$\begin{cases} \cos\theta' = -n_z\sin\theta + n_x\cos\theta, \\ \sin\theta' = \sqrt{1-\cos^2\theta'}. \end{cases}$$

Remark. *As we have said in section 3.1.1, the absorbing boundary condition in the isotropic case is written:*

$$\underline{\underline{\sigma}} \cdot \mathbf{n} - \rho\left(v_p(\mathbf{v}\cdot\mathbf{n})\mathbf{n} + v_s(\mathbf{v}\cdot\mathbf{t})\mathbf{t}\right) = 0.$$

Indeed, in the isotropic case, $\kappa = 1$ and $\theta = 0$ which implies that

$$A(\theta) = \begin{pmatrix} -\rho v_p & 0 \\ 0 & -\rho v_s \end{pmatrix}.$$

The boundary conditions are taken into account in the last equation of the global formulation (3.2.12) and they thus modify the expression of the transmission condition. Using the notation Bc for the matrix $PA(\theta')P^T$, we write a general global formulation for both the isotropic and anisotropic cases

$$\begin{aligned} \sum_{F\in\mathcal{F}_h\setminus\Gamma_l\cup\Gamma_a} \int_F \left(\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n}\right)\cdot\eta + \sum_{F\in\Gamma_l}\int_F \left(\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n}\right)\cdot\eta + \sum_{F\in\Gamma_a}\int_F \left(\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n} + (Bc\lambda_h)\right)\cdot\eta \\ = \sum_{F\in\Gamma_l}\int_F g_l\cdot\eta + \sum_{F\in\Gamma_a}\int_F g_a\cdot\eta, \end{aligned} \tag{3.3.17}$$

where in our case, $g_l = 0$ and $g_a = 0$.
Replacing $\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n}$ by the expression (3.2.10), we obtain

$$\sum_{F\in\mathcal{F}_h}\int_F \left(\underline{\underline{\sigma}}_h \cdot \mathbf{n}\right)\cdot\eta - \sum_{F\in\mathcal{F}_h}\int_F \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right)\cdot\eta + \sum_{F\in\Gamma_a}\int_F (Bc\lambda_h)\cdot\eta = 0. \tag{3.3.18}$$

Taking now into account the boundary conditions in the transmission condition, the HDG formulation reads as

find $(\mathbf{v}_h, \underline{\underline{\sigma}}_h, \lambda_h) \in \mathbf{V}_h^p \times \mathbf{\Sigma}_h^p \times \mathbf{M}_h^0$ such that $\forall (\mathbf{w}, \underline{\underline{\xi}}, \eta) \in \mathbf{V}^p(K) \times \mathbf{\Sigma}^p(K) \times \mathbf{M}_h^0$,

$$\begin{cases} \sum_{K\in\mathcal{T}_h} \int_K i\omega\rho_K \mathbf{v}_h \cdot \mathbf{w} - \sum_{K\in\mathcal{T}_h} \int_K \left(\nabla \cdot \underline{\underline{\sigma}}_h\right) \cdot \mathbf{w} + \sum_{K\in\mathcal{T}_h} \int_{\partial K} \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right) \cdot \mathbf{w} = 0, \\ \sum_{K\in\mathcal{T}_h} \int_K i\omega \underline{\underline{\sigma}}_h : \underline{\underline{\xi}} + \sum_{K\in\mathcal{T}_h} \int_K \mathbf{v}_h \cdot \nabla \cdot \left(\underline{\underline{C}}_K \underline{\underline{\xi}}\right) - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \lambda_h \cdot \underline{\underline{C}}_K \underline{\underline{\xi}} \cdot \mathbf{n} = \sum_{K\in\mathcal{T}_h} \int_K \mathbf{f} : \underline{\underline{\xi}}, \\ \sum_{F\in\mathcal{F}_h} \int_F \left(\underline{\underline{\sigma}}_h \cdot \mathbf{n}\right) \cdot \eta - \sum_{F\in\mathcal{F}_h} \int_F \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right) \cdot \eta + \sum_{F\in\Gamma_a} \int_F \left(Bc\lambda_h\right) \cdot \eta = 0. \end{cases} \tag{3.3.19}$$

The discretization of the last equation of (3.3.19) on the boundary Γ_a leads to the two systems:

$$\mathbb{Q}_{xl}^K \underline{\sigma}_{xx}^K + \mathbb{Q}_{zl}^K \underline{\sigma}_{xz}^K - \tau^{(K,l)} \mathbb{F}_l^K \underline{v}_x^K + \tau^{(K,l)} \mathbb{G}^j \underline{\lambda}_x^{\beta(K,l)} + Bc_{11} \mathbb{G}^F \underline{\lambda}_x^{\beta(k,l)} + Bc_{12} \mathbb{G}^F \underline{\lambda}_z^{\beta(k,l)} = 0 \tag{3.3.20}$$

and

$$\mathbb{Q}_{xl}^K \underline{\sigma}_{xz}^K + \mathbb{Q}_{zl}^K \underline{\sigma}_{zz}^K - \tau^{(K,l)} \mathbb{F}_l^K \underline{v}_z^K + \tau^{(K,l)} \mathbb{G}^j \underline{\lambda}_z^{\beta(K,l)} + Bc_{22} \mathbb{G}^F \underline{\lambda}_z^{\beta(k,l)} + Bc_{21} \mathbb{G}^F \underline{\lambda}_x^{\beta(k,l)} = 0 \tag{3.3.21}$$

Remark. *The expressions of Bc coefficients in the isotropic case are*

$$Bc_{11} = -\rho \left(v_p^K n_x^{K^2} + v_s^K n_z^{K^2}\right),$$

$$Bc_{12} = Bc_{21} = -\rho n_x^K n_z^K \left(v_p^K - v_s^K\right),$$

$$Bc_{22} = -\rho \left(v_p^K n_z^{K^2} + v_s^K n_x^{K^2}\right).$$

The Bc coefficients in the anisotropic case have a more complicated expression that we do not detail here.

We remind that the local system in Λ^K corresponding to the local transmission condition is written

$$\mathbb{B}^K \underline{W}^K + \mathbb{L}^K \underline{\Lambda}^K + \mathcal{R}^K = 0.$$

Then, for an element having an absorbing face, from (3.3.6), (3.3.7), (3.3.20) and, (3.3.20), we deduce that the inclusion of the absorbing condition does not modify the expression of the matrix $\mathbb{B}^K$ but affects the values of some coefficients in the matrix $\mathbb{L}^K$. For example if the edge $l = 1$ of K is an absorbing edge, $\mathbb{L}^K$ becomes

$$\mathbb{L}^K = \begin{bmatrix} \left(\tau^{(K,1)} + Bc_{11}\right) \mathbb{G}^{\beta(K,1)} & 0 & 0 & Bc_{12} \mathbb{G}^{\beta(K,1)} & 0 & 0 \\ 0 & \tau^{(K,2)} \mathbb{G}^{\beta(K,)} & 0 & 0 & 0 & 0 \\ 0 & 0 & \tau^{(K,3)} \mathbb{G}^{\beta(K,3)} & 0 & 0 & 0 \\ Bc_{21} \mathbb{G}^{\beta(K,1)} & 0 & 0 & \left(\tau^{(K,1)} + Bc_{22}\right) \mathbb{G}^{\beta(K,1)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \tau^{(K,2)} \mathbb{G}^{\beta(K,2)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \tau^{(K,3)} \mathbb{G}^{\beta(K,3)} \end{bmatrix}.$$

3.3.4 Algorithm

Besides the expression of the variational formulation, the HDG method also differs from classical DG methods concerning its computer implementation. In classical DG method, we only have to consider the volumic degrees of freedom which are local, while in HDG method we also have to deal with the Lagrange multiplier, which is global. In that sense, the implementation of the HDG method is a mix between the DG method (local volumic degrees of freedom) and the FE method (global degrees of freedom). This distinction is even more important if we want to integrate p-adaptivity in our HDG algorithm.
In this section, we describe the main blocks of the HDG algorithm. The most important steps are the construction of the linear system (algorithm 1) and of the right-hand side (algorithm 2).
The resolution part (algorithm 3) is divided into two tasks: the resolution of the system defined for the hybrid variable Λ with the MUMPS solver (see [61] for more details) and from the Λ solution, the elementwise resolution of the local system (3.3.10) in order to obtain the solution of the initial problem $\left(\mathbf{v}, \underline{\underline{\sigma}}\right)$.

Algorithm 1 Construction of the linear system

1: **for** $K = 1$ to Nb_{tri} **do**
2: Compute the local mass matrix $\mathbb{M}^K$ and the stiffness matrices $\mathcal{D}_u^K$, with $u = x, z$, considering the interpolation degree p_K of the element
3: **for** $l = 1$ to 3 **do**
4: Compute matrices $\mathbb{E}_l^K$, $\mathbb{F}_l^K$, $\mathbb{Q}_{ul}$, with $u = x, z$, and $\mathbb{G}^{j=\beta(K,l)}$, considering the interpolation degree $p_l = \max\left(p_K, p_{K'}\right)$ of the face
5: **end for**
6: Compute matrices $\mathbb{A}^K, (\mathbb{A}^K)^{-1}$
7: Compute matrix $\mathbb{C}^K$
8: Compute $\mathbb{R}^K = (\mathbb{A}^K)^{-1}\mathbb{C}^K$
9: Compute matrix $\mathbb{B}^K$
10: Compute matrix $\mathbb{L}^K$, taking into account the boundary conditions if it is necessary in the construction of $\mathbb{L}^K$
11: Compute $\mathbb{K}^K = \mathbb{B}^K(\mathbb{A}^K)^{-1}\mathbb{C}^K + \mathbb{L}^K$
12: Use of operator $\mathcal{A}_{HDG}^K \Leftrightarrow$ Connection between local indexes of the $(2 \times (ndof(p_1) \times ndof(p_2) \times ndof(p_3)))$ degrees of freedom and their global indexes
13: Construction of the corresponding section into the global matrix $\mathbb{K}$
14: **end for**

Algorithm 2 Construction of the right hand side

1: Localisation of the point source S
2: **for** $K = 1$ to Nb_{tri} **do**
3: Construction of the local matrices $\mathbb{S}^K$, of size $((5 \times n_K) \times 1)$
4: Multiplication $\mathbb{B}^K\mathbb{S}^K$
5: Use of operator $\mathcal{A}_{HDG}^K \Leftrightarrow$ use of the relation between local indexes and their global number
6: Construction of the corresponding section into the global right-hand side $\mathbb{S}$
7: **end for**

The sizes of the different matrices involved in the construction of the HDG discrete operator are presented in tab. 3.3.1. In this table, n_K is the number of degrees of freedom associated to the element and $ndof(p_f)$ the number of degrees of freedom associated to the face. The size of the matrix $\mathcal{K}$ which is

Algorithm 3 Resolution

1: Resolution of the linear system $\mathbb{K}\Lambda = \mathbb{S}$ with the use of the solver MUMPS
2: Computation of the solutions of the initial problem $\underline{W}^K$:
1: **for** $K = 1$ to Nb_{tri} **do**
2: Use of operator $\mathcal{A}_{HDG}^K \Leftrightarrow$ use of the relation between local indexes and their global number
3: Compute $\underline{W}^K = -(\mathbb{A}^K)^{-1}\mathbb{C}^K\mathcal{A}_{HDG}^K\underline{\Lambda} = -(\mathbb{A}^K)^{-1}\mathbb{C}^K\underline{\Lambda}^K$
4: **end for**

the matrix to be inverted for obtaining the value of Λ is $\left(2 \times \sum_{f=1}^{n_{faces}} ndof(p_f)\right) \times \left(2 \times \sum_{f=1}^{n_{faces}} ndof(p_f)\right)$.

Matrix	Size	
	m	n
$\mathbb{A}^K, (\mathbb{A}^K)^{-1}$	$(5 \times n_K)$	$(5 \times n_K)$
$\mathbb{C}^K$	$(5 \times n_K)$	$(2 \times (ndof(p_1) + ndof(p_2) + ndof(p_3)))$
$\mathbb{R}^K$	$(5 \times n_K)$	$(2 \times (ndof(p_1) + ndof(p_2) + ndof(p_3)))$
$\mathbb{B}^K$	$(2 \times (ndof(p_1) + ndof(p_2) + ndof(p_3)))$	$(5 \times n_K)$
$\mathbb{L}^K$	$(2 \times (ndof(p_1) + ndof(p_2) + ndof(p_3)))$	$(2 \times (ndof(p_1) + ndof(p_2) + ndof(p_3)))$
$\mathbb{K}^K$	$(2 \times (ndof(p_1) + ndof(p_2) + ndof(p_3)))$	$(2 \times (ndof(p_1) + ndof(p_2) + ndof(p_3)))$
$\mathbb{K}$	$\left(2 \times \sum_{f=1}^{n_{faces}} ndof(p_f)\right)$	$\left(2 \times \sum_{f=1}^{n_{faces}} ndof(p_f)\right)$
$\mathbb{S}^K$	$(5 \times n_K)$	1
$\mathbb{B}^K\mathbb{S}^K$	$(2 \times (ndof(p_1) + ndof(p_2) + ndof(p_3)))$	1
$\mathbb{S}$	$\left(2 \times \sum_{f=1}^{n_{faces}} ndof(p_f)\right)$	1

Tab. 3.3.1: Size of the matrices.

3.4 Numerical results

In this section, we provide numerical results in order to validate and evaluate the efficiency of the HDG method. As for the upwind flux and the centered flux DG methods of chapter 2, the HDG method is implemented in a Fortran 90 software and we use the MUMPS sparse direct solver for the resolution of the linear system defined for the hybrid variable Λ.
The computations are done on a computer system whose features are given in section 2.3 of chapter 2. For this preliminary numerical study of the HDG method, we only consider here the simple problem of the propagation of a plane wave in an homogeneous medium. More complicated and more realistic problems will be considered in the next chapter.

3.4.1 Plane wave propagation in an homogeneous medium

In order to be coherent and to make comparisons, we take the same physical properties of the medium than in section 2.3 of chapter 2, i.e. a mass density ρ equal to 1000 kg.m^{-3} and Lamé's

coefficients λ and μ equal to 8 MPa and 4 MPa respectively which imply a velocity v_p of P-waves equal to 4000 m.s^{-1} and a velocity v_s of S-waves equal to 2000 m.s^{-1}.
The same boundary condition that we have imposed in order to obtain a plane wave is applied:

$$u = \nabla e^{i(k\cos\theta x + k\sin\theta y)}$$

where $k = \dfrac{\omega}{v_p}$ is the wave number, θ the incidence angle (we have set $\theta = 0$), and ω the angular frequency, $\omega = 2\pi f$ where f is the frequency (we have taken $f = 2$Hz, so $\omega = 4\pi \simeq 12.56$ Hz).
We also use the same discretization of the computational domain Ω i.e. Ω is discretized into three unstructured meshes, two of them are represented on figs. 2.3.1a and 2.3.1b.
The features of the three meshes (number of elements, vertices, maximum length of an edge h_{max}, minimum length h_{min} and the ratio h_{min}/h_{max}) are summarized in tab. 2.3.1.

As for the classical DG methods, we focus on the numerical solution of the V_x component. When using the coarsest mesh with 3 000 triangles and the HDG-$\mathbb{P}_1$ formulation, we obtain the numerical solution shown on fig. 3.4.2a. On fig. 3.4.2b, we look at the absolute error between this numerical solution and the exact one represented on fig. 3.4.1. Clearly, for this relatively coarse mesh, the HDG-$\mathbb{P}_1$ scheme is not accurate enough. Increasing the interpolation degree (fig. 3.4.3a) leads to a numerical solution which is closer to the exact one. If we increase the resolution of the mesh (fig. 3.4.4a), the numerical solution is closer too to the exact one, but we clearly see that the second mesh is not again well adapted for the HDG-$\mathbb{P}_1$ scheme with our choices of parameters for this test case.

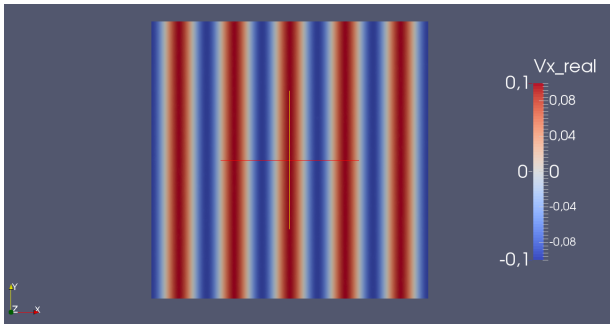
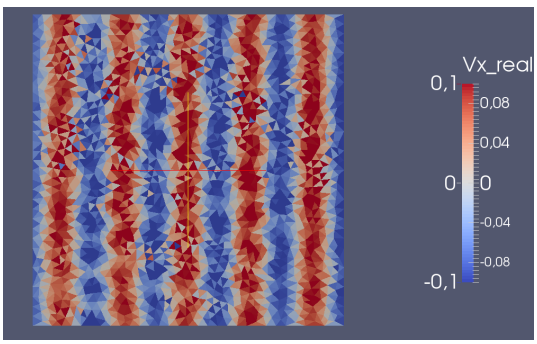
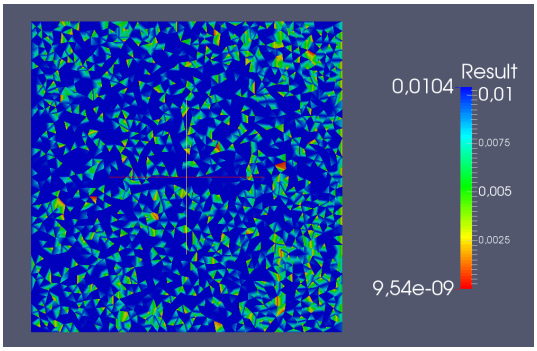


Fig. 3.4.1: Exact solution, V_x component.

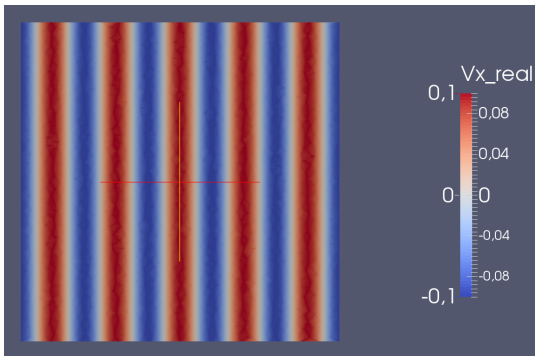


(a) Numerical solution.

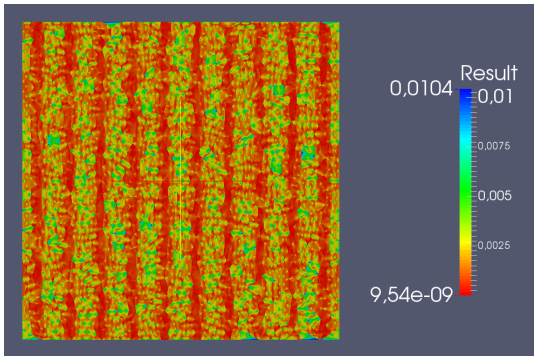


(b) Absolute error between the exact solution and the numerical solution .

Fig. 3.4.2: Mesh M1, HDG-$\mathbb{P}_1$ scheme, V_x component.

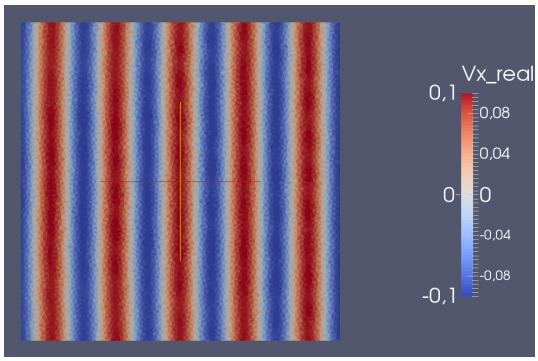


(a) Numerical solution.

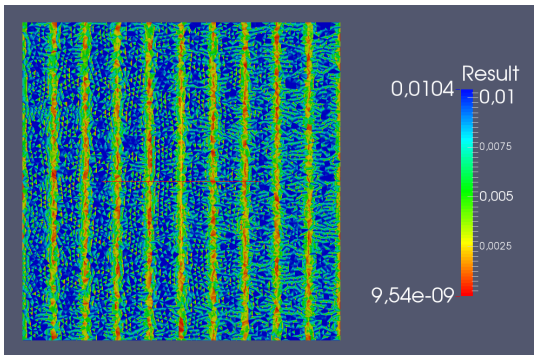


(b) Absolute error between the exact solution and the numerical solution.

Fig. 3.4.3: Mesh M1, HDG-$\mathbb{P}_2$ scheme, V_x component.



(a) Numerical solution.



(b) Absolute error between the exact solution and the numerical solution.

Fig. 3.4.4: Mesh M2, HDG-$\mathbb{P}_1$ scheme, V_x component.

Fig. 3.4.5 shows the numerical convergence of the HDG method. Similarly to classical finite elements methods and to the upwind flux DG method, the HDG method proposed here converges with order $p+1$, i.e with optimal rate.

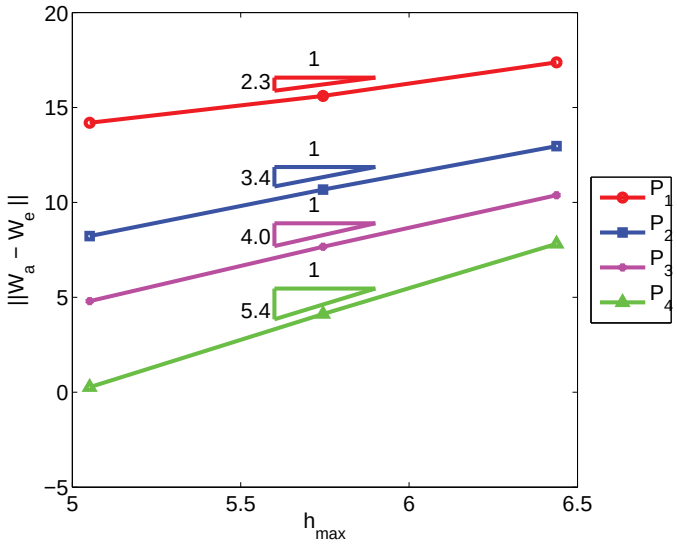
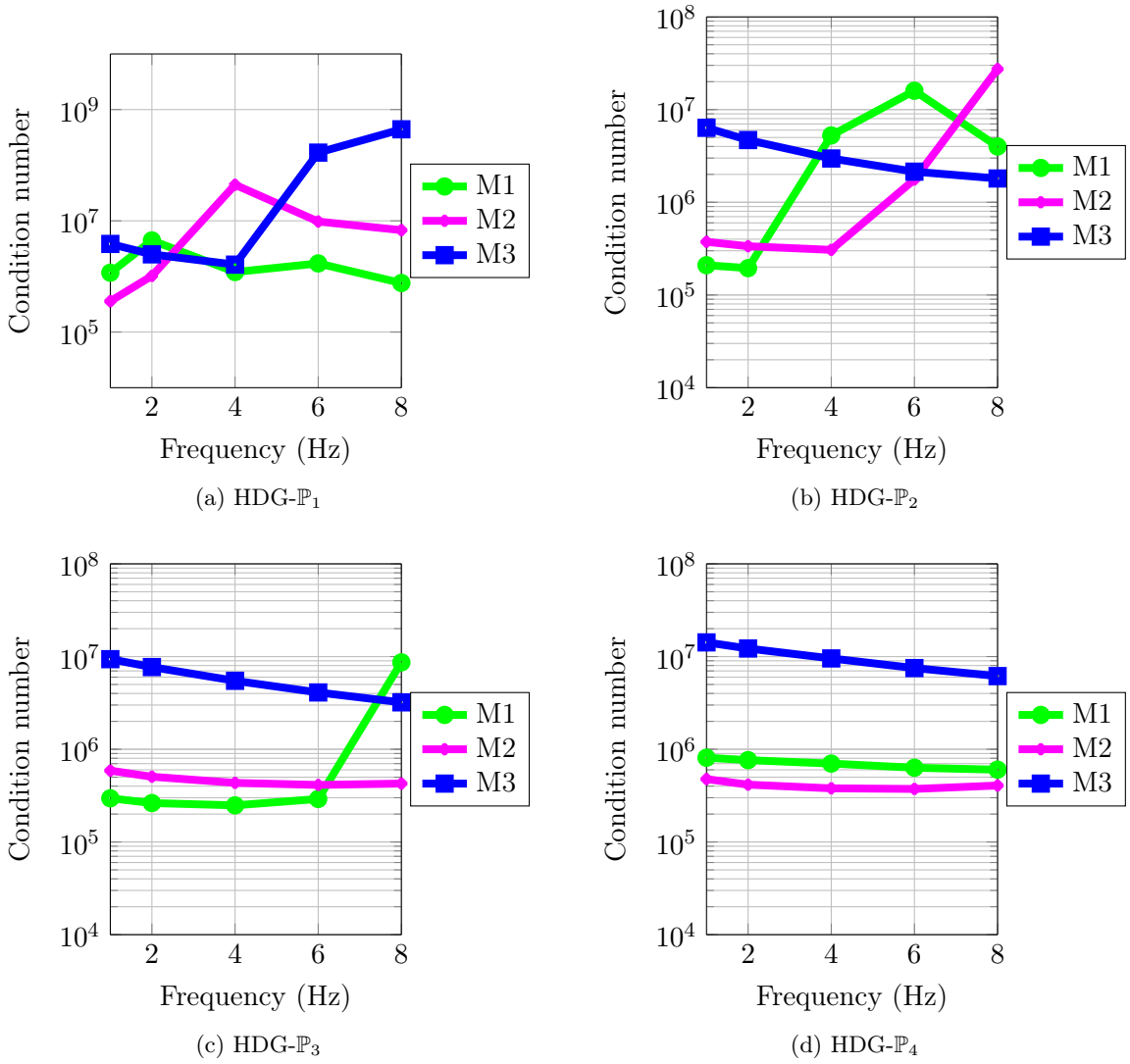


Fig. 3.4.5: Convergence order of the HDG method

For various frequencies (from 1 Hz to 8 Hz), the condition number of the global matrix is represented on figs. 3.4.6a to 3.4.6d for each interpolation degree.
The variations of the condition number of the HDG-$\mathbb{P}_1$ global matrix proves that the HDG-$\mathbb{P}_1$ interpolation is unstable. Indeed, this condition number varies strongly with both the frequency and the size of the mesh, which impacts the accuracy of the solution. Hence, to obtain an accurate solution with the HDG-$\mathbb{P}_1$ scheme we have to use very fine meshes. The HDG-$\mathbb{P}_2$ interpolation is less sensitive and can be used with fine meshes while the HDG-$\mathbb{P}_3$ interpolation can be use correctly with medium meshes. Coarse meshes can be used with the HDG-$\mathbb{P}_4$ interpolation.



(a) HDG-$\mathbb{P}_1$ (b) HDG-$\mathbb{P}_2$

(c) HDG-$\mathbb{P}_3$ (d) HDG-$\mathbb{P}_4$

Fig. 3.4.6: Condition number of the global matrix as a function of the frequency, with $\tau = 1$.

On tab. 3.4.1, we present the mean and the relative errors of the V_x component computed with the HDG method and the UDG method. More comparisons are given in section 4.2 in the next chapter. For a same mesh and using the same interpolation order, the HDG method has a larger error and seems to be less accurate than the UDG scheme.

On tabs. 3.4.2 and 3.4.3, we present the characteristics of the mesh for a given error level with different interpolation degrees. We also specify the number of degrees of freedom per wavelength $\lambda_w = \dfrac{v_p}{f}$. The global number of degrees of freedom of the HDG method is presented in the next chapter and is compared to the one of the UDG method.

When we established the HDG formulation section 3.2, we introduced a stabilization matrix $\mathbf{S}$ which plays a central role on the accuracy of the method as verified here. In our formulation, we have taken $\mathbf{S}$ equal to $\tau\mathbf{I}$ where $\mathbf{I}$ is the identity matrix and τ is the stabilization parameter. Thus, it is the value of the parameter τ which influences the accuracy of the HDG method and in the following, we want to study this for the simple test case considered here. To do that, for several values of interpolation degree, we compute the numerical solution of the propagation of the plane wave on the second mesh (composed of 10300 elements) while the parameter τ varied from 10^{-4} to 10^{4}. The relative

h (m)	Interpolation degree	Mean Error V_x		Relative Error V_x (%)	
		HDG scheme	UDG scheme	HDG scheme	UDG scheme
625.0		0.2	1.8e-02	52.8	6.0
312.5	1	1.3e-02	1.7e-03	13.7	1.8
56.25		1.2e-03	9.9e-05	5.5	0.4
625.0		6.7e-03	1.6e-03	2.1	0.5
312.5	2	5.9e-04	5.7e-05	0.6	5.8e-02
56.25		3.1e-05	1.4e-06	0.1	6.1e-03
625.0		5.4e-04	1.3e-04	0.2	4.1e-02
312.5	3	2.6e-05	3.1e-06	2.7e-02	3.2e-03
56.25		6.5e-07	3.7e-08	2.9e-03	1.7e-04
625.0		3.9e-05	1.0e-05	1.2e-02	3.2e-03
312.5	4	9.5e-07	6.8e-08	9.7e-04	7.0e-05
56.25		1.1e-08	3.1e-10	4.9e-05	1.4e-06

Tab. 3.4.1: Mean and relative errors on V_x, $\tau = 1$.

Interpolation degree	# elements	# vertices	h_{max} (m)	h_{min} (m)	# dof per wavelength
1	32 000	16 320	156.25	41.7	97
2	3 100	1 620	625.0	193.6	14
3	640	350	1 250	435.2	5
4	230	130	2 500	742.3	2

Tab. 3.4.2: Characteristics of the meshes in order to reach an accuracy level of 1% on V_x for $\tau = 1$.

Interpolation degree	# elements	# vertices	h_{max} (m)	h_{min} (m)	# dof per wavelength
1	78 000	39 000	99.1	24.6	233
2	45 000	22 500	156.3	45.4	201
3	3 100	1 620	625.0	193.6	19
4	780	420	1 250	394.74	6

Tab. 3.4.3: Characteristics of the meshes in order to reach an accuracy level of 1% on V_x for $\tau = 1$.

error depending on the value of τ is presented on figs. 3.4.7a to 3.4.7d. The thick line represents the UDG relative error for the same mesh using the same interpolation order. We remark that for this test case, when the value of the parameter τ is of the order of the v_p value, we obtain the same relative error than that the UDG scheme. The value of τ has no influence on the memory requirement and computational time. So for example, for the HDG-$\mathbb{P}_2$ scheme on mesh M2, we need 355 MB and 6 seconds to obtain the numerical solution. On figs. 3.4.8a to 3.4.8d we plot the variation of the relative error on the different meshes when we take $\tau = 1000$. Except for the HDG-$\mathbb{P}_1$ scheme, the behavior of the relative error when $\tau = 1000$ is the same as the behavior of the relativor error of the UDG scheme.

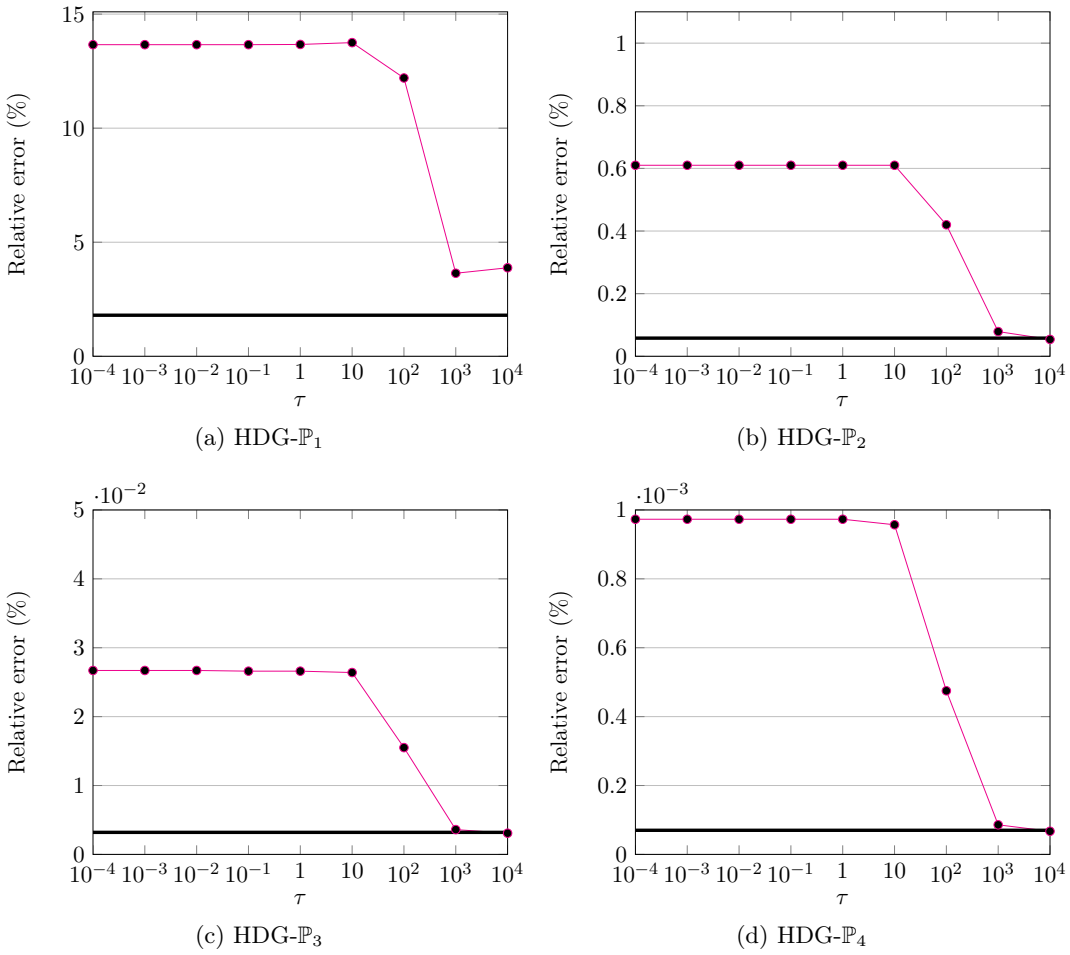
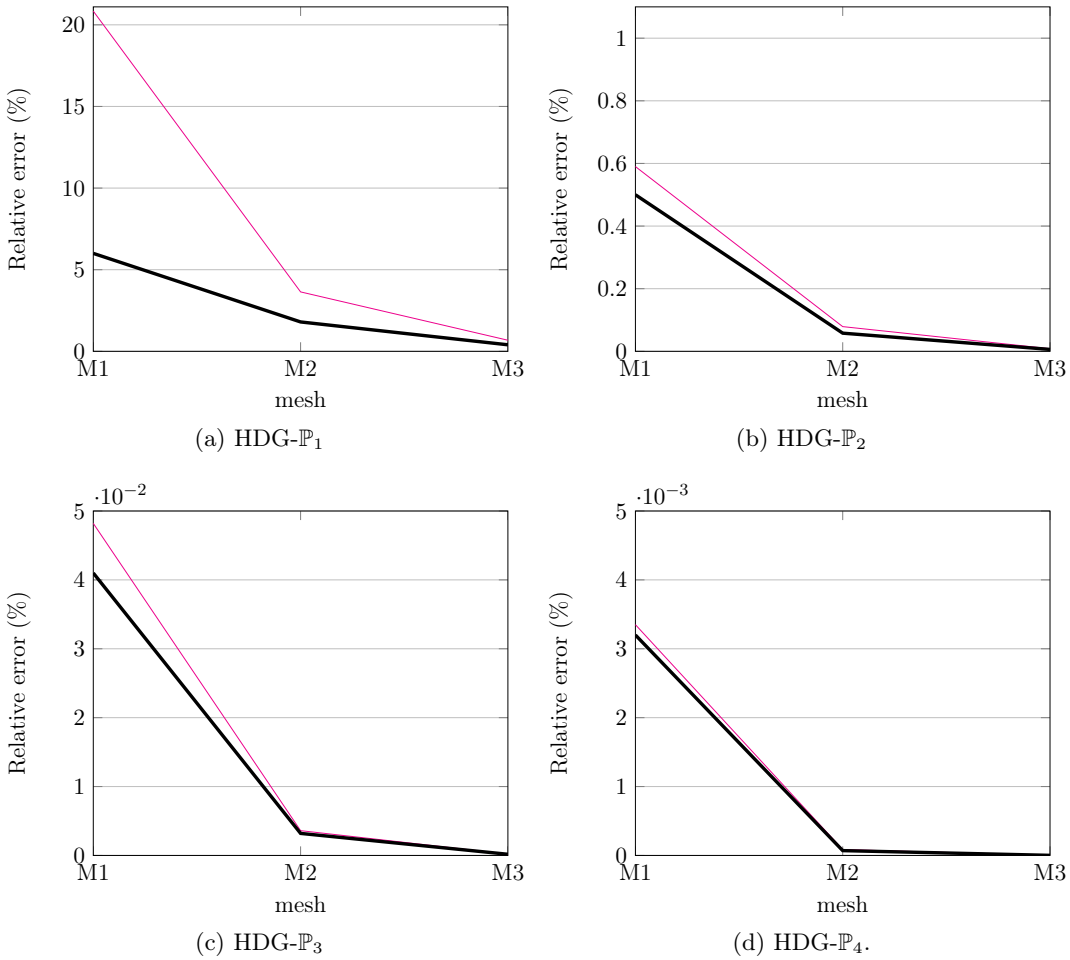


Fig. 3.4.7: Influence of the parameter τ on the relative error using the same mesh and the same physical parameters.

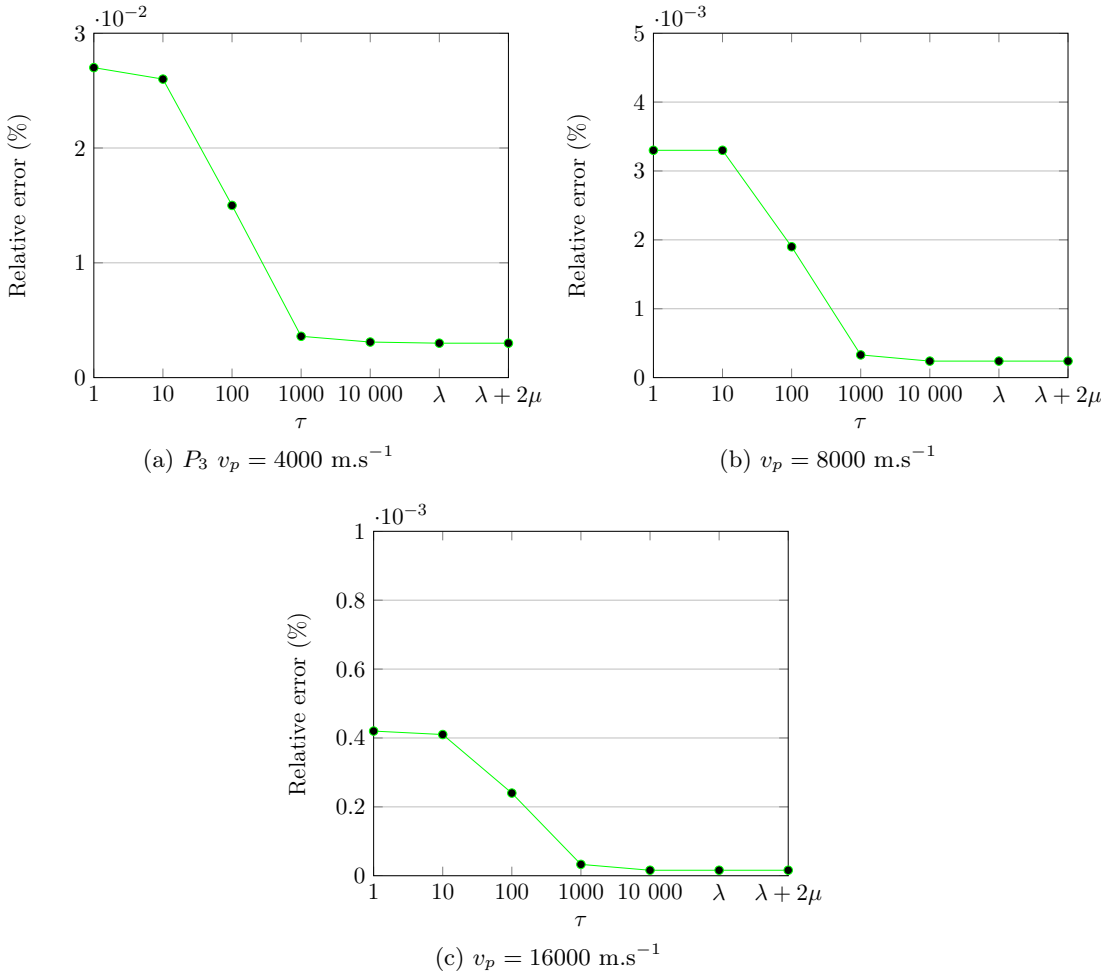


(a) HDG-$\mathbb{P}_1$ (b) HDG-$\mathbb{P}_2$

(c) HDG-$\mathbb{P}_3$ (d) HDG-$\mathbb{P}_4$.

Fig. 3.4.8: Variation of the relative error as a function of the mesh for the UDG scheme (−) and for the HDG scheme (–) when $\tau = 1000$.

We have also plotted the influence of the parameter τ on the condition number for different frequencies on figs. 3.4.13a to 3.4.13d. We remark that we have a jump of the value of the condition number when the value of the parameter τ is equal to $v_p^2 = \lambda + 2\mu$.

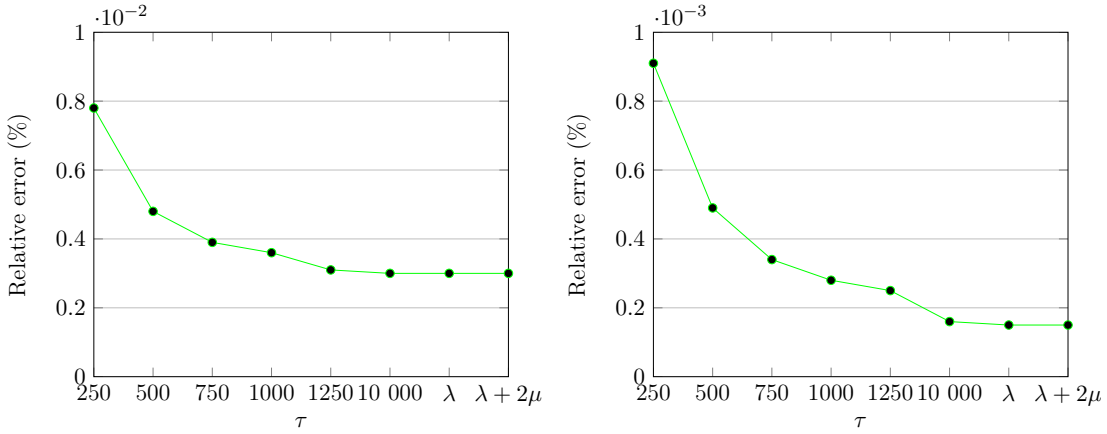


(a) P_3 $v_p = 4000$ m.s^{-1}

(b) $v_p = 8000$ m.s^{-1}

(c) $v_p = 16000$ m.s^{-1}

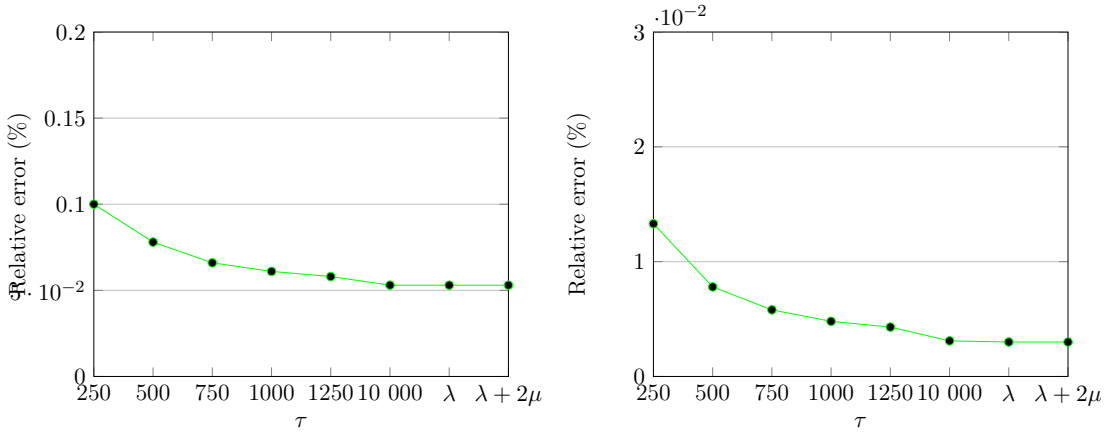
Fig. 3.4.9: Variation of the relative error as a function of the parameter τ for only different values of v_p, with HDG-$\mathbb{P}_3$, mesh M2.



(a) $v_p = 4000$ m.s^{-1}, $v_s = 2000$ m.s^{-1}, $f = 2$ Hz, $\rho = 1$ kg.m^{-3}.

(b) $v_p = 8000$ m.s^{-1}, $v_s = 4000$ m.s^{-1}, $f = 2$ Hz, $\rho = 1$ kg.m^{-3}.

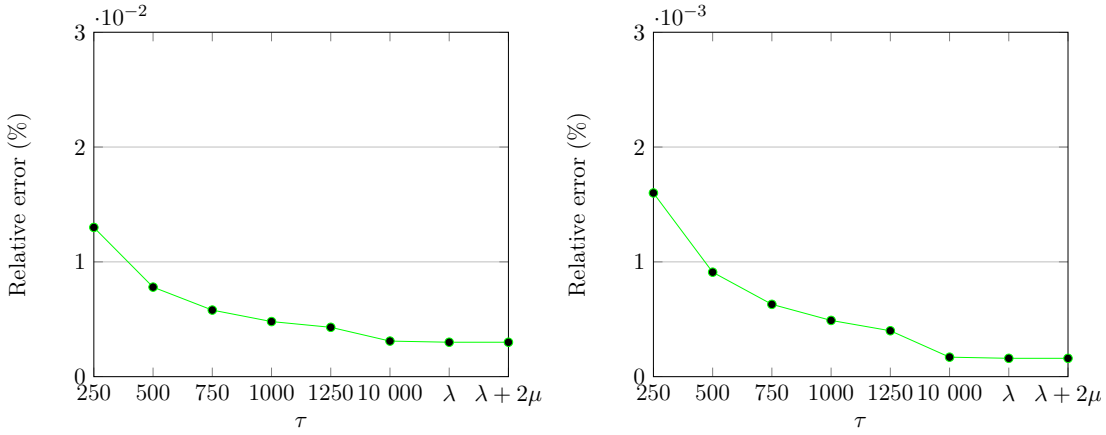
Fig. 3.4.10: Variations of the relative error as a function of the parameter τ when v_p and v_s change, HDG-$\mathbb{P}_3$, mesh M2.



(a) $f = 4$ Hz, $v_p = 4000$ m.s^{-1}, $v_s = 2000$ m.s^{-1}, $\rho = 1$ kg.m^{-3}.

(b) $f = 4$ Hz, $v_p = 8000$ m.s^{-1}, $v_s = 4000$ m.s^{-1}, $\rho = 1$ kg.m^{-3}.

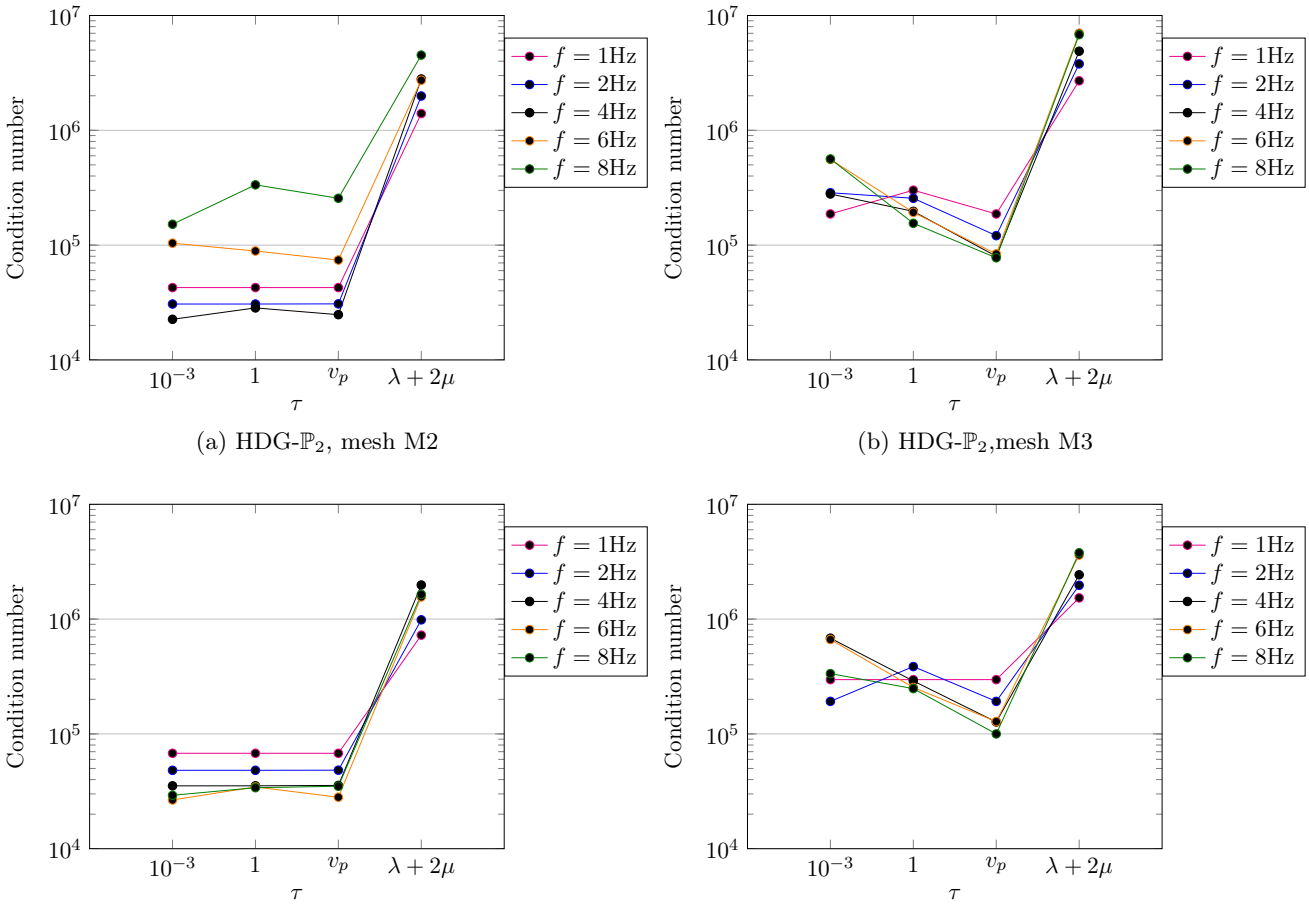
Fig. 3.4.11: Variations of the relative error as a function of the parameter τ when f only changes, HDG-$\mathbb{P}_3$, mesh M2.



(a) $f = 2$ Hz, $v_p = 4000$ m.s^{-1}, $v_s = 2000$ m.s^{-1}, $\rho = 2$ kg.m^{-3}.

(b) $f = 2$ Hz, $v_p = 8000$ m.s^{-1}, $v_s = 4000$ m.s^{-1}, $\rho = 2$ kg.m−3.

Fig. 3.4.12: Variations of the relative error as a function of the parameter τ when only ρ changes, HDG-$\mathbb{P}_3$, mesh M2.



(a) HDG-$\mathbb{P}_2$, mesh M2

(b) HDG-$\mathbb{P}_2$,mesh M3

(c) HDG-$\mathbb{P}_3$,mesh m2

(d) HDG-$\mathbb{P}_3$,mesh m3

Fig. 3.4.13: Influence of the parameter τ on the condition number using the same mesh and the same physical parameters.

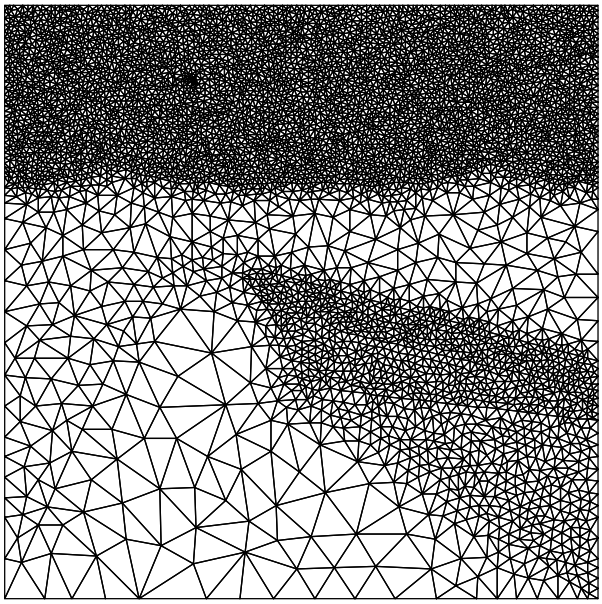


Fig. 3.4.14: Mesh used for testing p-adaptivity.

3.4.2 P-ADAPTIVITY

We test the p-adaptivity property of the HDG method. To do this, we still consider the case of the propagation of a plane wave in an homogeneous medium. The features of the considered medium are a mass density $\rho = 1$ kg.m^{-3}, a P-waves velocity v_p=8000 m.s^{-1}, and a S-waves velocity v_s= 4000 m.s^{-1}. The unstructured mesh used for this test is presented on fig. 3.4.14 and its characteristics are summarized in tab. 3.4.4.

# Mesh elements	# Mesh vertices	h_{min} (m)	h_{max} (m)	h_{max}/h_{min}	a_{min}	a_{max}
11 500	5 900	136.5	3 404	24.9	8 226	3 000 000

Tab. 3.4.4: Characteristics of the selcted unstructured triangular mesh for testing p-adaptivity.

The interpolation degree p is distributed as follows:

1. We determine approximately the height h_t of each triangle by considering $h_t = 2 \times \sqrt{a_t}$, where a_t is the area of each triangle;
2. We compute $n_t = \lambda_S / h_t$, where λ_S is the S-wavelength. The value n_t represents the number of points per wavelength;
3. Considering tab. 3.4.1, we determine for each interpolation degree p, the value n_t which allows to obtain a relative error on the order of 1%. This level of error is a classical level of error required by seismic imaging application. Thus, we assign the interpolation degree as follows
   if $(12.0 < n_t)$ then p=1,
   if ($4.0 < n_t < 12.0$) then p=2,
   if ($1.5 < n_t < 4.0$) then p=3,
   if ($0.5 < n_t < 1.5$) then p=4.

These values are determinated thanks to tab. 3.4.2.

We test the p-adaptivity with two frequencies: f=1 Hz, and f= 2 Hz; and the distribution of triangles for each experiment is given in tab. 3.4.5.

p	# triangles (f=1 Hz)	# triangles (f=2 Hz)
1	3 155	0
2	7 817	9 608
3	485	1 465
4	77	461

Tab. 3.4.5: Distribution of triangles for the p-adaptivity.

We now compare the computational cost of the HDG method with p-adaptivity and the HDG method using a uniform distribution of the interpolation degree, in terms of the relative error on the V_x component and the computational coss (memory and construction and resolution times). Results are presented in tabs. 3.4.6 and 3.4.7.

p	V_x Relative error (%)	Memory (MB)	Construction time (s)	Solution time (s)
1	108.7	192	0.8	1.7
2	2.6	408	3.1	4.4
3	0.2	716	9.4	9.4
4	2.8e-02	1082	24.9	17.0
p-adaptivity	0.2	387	4.2	4.3

Tab. 3.4.6: Comparison of the HDG-$\mathbb{P}_p$ performances using uniform degree and p-adaptivity for f=1 Hz.

p	V_x Relative error (%)	Memory (MB)	Construction time (s)	Solution time (s)
1	1303.8	192	0.8	1.7
2	282.0	408	3.2	4.4
3	10.7	716	9.3	9.4
4	1.1	1082	25.1	17.1
p-adaptivity	1.0	485	6.3	6.1

Tab. 3.4.7: Comparison of the HDG-$\mathbb{P}_p$ performances using uniform degree and p-adaptivity for f=2 Hz.

These results demonstrate the potential advantages of using a non-uniform distribution of the interpolation degree which allows to improve the error on the solution without increasing the memory and the computational time. Indeed, in the first experiment, the mesh is composed of 68% of triangles with degree 2, 27% with degree 1, and only 4% with degree 3 and 1% with degree 4. This configuration allows to obtain a result as accurate as the one obtained using an uniform degree equal to 3, but with memory and the computational time costs divided by 2 in comparison to the HDG-$\mathbb{P}_3$ costs. These costs are similar to the ones required for an uniform degree equal to 2. In the second experiment, the mesh is composed of 83% of triangles with degree 2, 13% with degree 3 and only 4% with degree 4 and in that case we obtain an error similar to the one obtained with an uniform degree equal to 4,

but the required memory and the computational time are closed to the one needed for a HDG-$\mathbb{P}_2$ computation.

Conclusion

In this chapter, we have applied the HDG method to the 2D elastic Helmholtz equations and we have studied its well posedness. We have shown, thanks to numerical experiments that the convergence order of the HDG method is optimal as with classical finite elements.
The value of the stabilization parameter τ impacts on the accuracy of the HDG method. Numerically, its optimal value seems to be close to the value of v_p. In any case, the HDG-$\mathbb{P}_1$ scheme seems to be numerically unstable when we look at the condition number of the global matrix, and thus higher order schemes should be prefered.
We have shown that, thanks to p-adaptivity on a given mesh, we could reach the accuracy of a global method of degree p for the cost of a method of degree $p-1$ or $p-2$.

These results should be confirmed theoretically, following the strategy proposed by Cockburn, Gopalakrishnan and Sayas in [72], which was extended by Griesmaier and Monk to the acoustic Helmholtz equation in [73]. It has been shown in [74] that, on classical second order elliptic problem, the Lagrange multiplier λ contained extra informations so that the accuracy of the solution could be enhanced by applying an appropriate local post processing.

In the next chapter, we will compare the efficiency of the HDG method to the one of classical DG methods. We will show that, as it was observed in [75] by Kirby, Sherwin and Cockburn, when comparing HDG and continuous Galerkin (CG) methods (see [38]), HDG outperforms classical DG methods as soon as the discretization order is high enough.

Chapter 4

2D numerical results - Methods comparison

We have presented a first performance analysis of the HDG method in the previous chapter. We now want to compare its performances to the ones of classical DG methods. We focus on the upwind flux DG method since, as we have seen in chapter 2, the centered flux DG method is suboptimal whereas the upwind flux DG method converges with order $p+1$ as FE methods and as the HDG method.
We also compare the HDG performances with the ones of an interior penalty DG (IPDG) method. Indeed, even if the IPDG method is based on the second order formulation of the wave equations, this comparison is interesting because in the IPDG method, the unknown is vectorial (one solves the problem in terms of $\mathbf{v}$). Actually, the HDG method can be seen locally as a first order formulation of the wave equations involving the vectorial unknowns $\mathbf{v}$ and the tensorial unknowns $\underline{\underline{\sigma}}$, and globally as a second order formulation involving the vectorial unknown λ. It is thus interesting to compare the construction of the global matrix and the resolution of the global system for the two methods.
In the first section of this chapter, we give a brief presentation of the IPDG method. Then, we compare the results of the HDG, upwind flux DG and IPDG methods for the propagation of a plane wave in an homogeneous medium using the same configuration as the one presented in sections 2.3 of chapter 2 and 3.4.1 of chapter 3. We complicate the geometry of the problem by considering the disk shaped scattering with a free surface condition in section 4.3. In the following section, we extend our study to the elastic solid scattering problem before considering a classical 2D test case that is used as a benchmark in geophysics where we compare the parallel performances of the HDG and IPDG methods. Finally, in the last section of this chapter, we test the HDG method with anisotropic media.

4.1 Interior Penalty Discontinuous Galerkin (IPDG) method

For our study, we apply to the elastic Helmholtz equations the interior penalty discontinuous Galerkin (IPDG) method used in [56] or [64] to discretize the wave equations in the time-domain. It is worth noting that the IPDG scheme has been designed for the second order formulation

$$-\omega^2\rho\mathbf{v} - \nabla\cdot(\underline{\underline{C}}\nabla\mathbf{v}) = \mathbf{f}_\mathbf{v} \quad \text{in } \Omega. \tag{4.1.1}$$

We use the same notations and definitions of the jump and the mean than the ones introduced in chapter 2 section 2.1.2. We also use the approximation spaces $\mathbf{V}_h$ defined in that section. As we have done for the nodal DG methods applied to the first order formulation of the wave equations in chapter 2, we multiply equation (4.1.1) by a test function $\varphi \in \mathbf{V}_h$ on an element K

$$-\int_K \omega^2\rho\mathbf{u}\cdot\varphi - \int_K \nabla\cdot(\underline{\underline{C}}\nabla\mathbf{u})\cdot\varphi = \int_K \mathbf{f}_\mathbf{v}\cdot\varphi.$$

The use of a Green formula leads to

$$-\int_K \omega^2\rho\mathbf{u}\cdot\varphi + \int_K \underline{\underline{C}}\nabla\mathbf{u}\cdot\nabla\varphi - \int_{\partial K}\underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}\varphi = \int_K \mathbf{f}_\mathbf{v}\cdot\varphi.$$

Let $\mathcal{T}_h$ a triangulation of Ω. We assume that ρ and $\underline{\underline{C}}$ are regular enough so that, on each internal edge F of $\mathcal{T}_h$

$$[\![\mathbf{u}]\!] = 0, \quad \text{and} \quad [\![\underline{\underline{C}}\nabla\mathbf{u}]\!] = 0.$$

Summing over all the elements of $\mathcal{T}_h$, we obtain

$$\sum_{K\in\mathcal{T}_h} -\int_K \omega^2\rho\mathbf{u}\cdot\mathbf{v} + \sum_{K\in\mathcal{T}_h}\int_K \underline{\underline{C}}\nabla\mathbf{u}\cdot\nabla\mathbf{v} - \sum_{K\in\mathcal{T}_h}\int_{\partial K}\underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}\varphi = \sum_{K\in\mathcal{T}_h}\int_K \mathbf{f}\cdot\mathbf{v}.$$

As explained previously for the other DG methods, it is the expression of the numerical flux $\int_{\partial K}\underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}\varphi$ on the boundary which differentiates all the DG methods. We briefly explain how it is defined in the IPDG formulation.
As we use discontinuous functions and following what we have done in 2.2.1, we write

$$\begin{aligned}\sum_{K\in\mathcal{T}_h}\int_{\partial K}\underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}\varphi &= \sum_{F\in\mathcal{F}_b}\int_F \underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}\varphi + \sum_{F\in\mathcal{F}_i}\int_F \left(\underline{\underline{C}}^+\nabla\mathbf{u}^+\cdot\mathbf{n}^+\varphi^+ + \underline{\underline{C}}^-\nabla\mathbf{u}^-\cdot\mathbf{n}^-\varphi^-\right)\\ &= \sum_{F\in\mathcal{F}_h}\int_F [\![\underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}\varphi]\!].\end{aligned}$$

Using the property linking the mean and the jump, we obtain

$$\sum_{F\in\mathcal{F}_h}\int_F [\![\underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}\varphi]\!] = \sum_{F\in\mathcal{F}_h}\int_F \{\underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}\}\cdot[\![\varphi]\!] + \sum_{F\in\mathcal{F}_h}\int_F [\![\underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}]\!]\{\varphi\},$$

As $\mathbf{u}$ is solution of the elastic wave equations, we have

$$[\![\mathbf{u}]\!] = 0, \quad \text{and} \quad [\![\underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}]\!] = 0.$$

Thus, the boundary term reads

$$\sum_{F\in\mathcal{F}_h}\int_F [\![\underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}\varphi]\!] = \{\underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}\}\cdot[\![\varphi]\!].$$

In order to have a symmetric formulation, we can write

$$\sum_{F\in\mathcal{F}_h}\int_F [\![\underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}\varphi]\!] = \{\underline{\underline{C}}\nabla\mathbf{u}\cdot\mathbf{n}\}\cdot[\![\varphi]\!] + [\![\mathbf{u}]\!]\{\underline{\underline{C}}\nabla\varphi\cdot\mathbf{n}\}.$$

Finally, to ensure the coercivity of the bilinear form of the IPDG formulation we add a positive term defined by

$$\sum_{F\in\mathcal{F}_h}\int_F \gamma[\![\mathbf{u}]\!]\cdot[\![\mathbf{v}]\!],$$

where γ is the parameter which penalizes the jump of the numerical solution on the face F. This parameter is defined by $\gamma = \dfrac{\alpha}{h_F}$, where α only depends on the interpolation order p and h_F is the smallest inscribed circle of the two elements sharing F.
The IPDG formulation is finally given by, find $\mathbf{u}_h \in \mathbf{V}_h$ such that, $\forall\varphi\in\mathbf{V}_h$

$$\begin{aligned}&\sum_{K\in\mathcal{T}_h} -\int_K \omega^2\rho\mathbf{u}_h\cdot\varphi + \sum_{K\in\mathcal{T}_h}\int_K \underline{\underline{C}}\nabla\mathbf{u}_h\cdot\nabla\varphi\\ &\quad - \sum_{F\in\mathcal{F}_h}\int_F \left(\{\underline{\underline{C}}\nabla\mathbf{u}_h\}\cdot[\![\varphi]\!] + [\![\mathbf{u}_h]\!]\{\underline{\underline{C}}\nabla\varphi\}\right) + \sum_{F\in\mathcal{F}_h}\int_F \gamma[\![\mathbf{u}_h]\!]\cdot[\![\mathbf{v}]\!] = \sum_{K\in\mathcal{T}_h}\int_K \mathbf{f}_\mathbf{v}\cdot\varphi.\end{aligned} \tag{4.1.2}$$

This associated linear system writes

$$\mathcal{M}\underline{U}_h + \mathcal{K}\underline{U}_h + \mathcal{B}\underline{U}_h = \mathcal{F}, \tag{4.1.3}$$

where $\underline{U}_h$ is the vector gathering all the components of the vector $\mathbf{u}_h$, $\mathcal{M}$ is the mass matrix which is block diagonal, $\mathcal{K}$ is the stiffness matrix and $\mathcal{B}$ is the matrix gathering the terms coming from the discretization of the absorbing condition (which is not detailed here) and $\mathcal{F}$ is the source term.

4.2 Plane wave propagation in an homogeneous medium

The configuration of this problem is the one we have presented in sections 2.3 of chapter 2 and 3.4.1 of chapter 3 and we do not detail it further. Here, we compare the results obtained with the HDG method (HDG), the upwind DG method (UDG) and the IPDG method (IPDG).

Tabs. 4.2.1 and 4.2.2 compare the mean and the relative errors obtained using the HDG and the UDG formulations for V_x and σ_{xx} components respectively. For the HDG formulation, we use two values of τ, $\tau = 1$ and $\tau = v_p$. As the IPDG errors are similar to the UDG errors, we do not compare in these two tables the results obtained with the IPDG method.
We focus on the V_x component (tab. 4.2.1) because the error level on σ_{xx} component is similar, except for $p = 1$. We test the HDG formulation with two values of the parameter τ, 1 and v_p, because we have seen in section 3.4.1 that this parameter has an influence on the accuracy of the solution. We observe that the mean and relative errors obtained with the UDG scheme and the HDG scheme with $\tau = v_p$ are close, except for $p = 1$ when using the coarsest mesh. When we compare the methods on the same mesh and with the same interpolation order, the largest errors (mean and relative) are obtained with the HDG scheme using $\tau = 1$.
Tabs. 4.2.4 and 4.2.6 compare, respectively, the number of non-zero terms in the global matrix and the memory that the solver needs for the inversion of the global matrix and resolution process for both methods and for a solution computed on the same mesh and with the same interpolation order. We also add the results obtained with the IPDG formulation in these tables. These experiments show that to obtain a solution computed on the same mesh and with the same interpolation order, the HDG method requires less memory than classical DG methods, except for $p = 1$. This is due to the fact that the HDG method reduces the global number of unknowns, and thus, the size of the global matrix that we have to invert. This gain increases with the order p.
The global number of unknowns corresponds to the number of degrees of freedom and is presented in tab. 4.2.5. For each method, it can be easily computed. If we consider uniform interpolation degree, for the 2D elastic Helmholtz equations, in the classical DG framework it is

$$N_{DG} = 5 \times nde \times N_e,$$

where nde is the number of degrees of freedom on each element and N_e is the number of elements. The global number of unknowns for the HDG method is

$$N_{HDG} = 2 \times ndf \times N_f,$$

where ndf is the number of degrees of freedom on each face (edge in 2D) and N_f is the number of faces. Finally for the IPDG scheme, it is

$$N_{IPDG} = 2 \times nde \times N_e.$$

In 2D, $ndf = p + 1$, $nde = \dfrac{(p+1)(p+2)}{2}$, and $N_f \simeq \dfrac{3}{2}N_e$. Thus, replacing ndf, nde and N_f in order

to compare N_{DG}, N_{IPDG} and N_{HDG}, we get

$$\begin{aligned} N_{HDG} &= 3\,(p_{HDG}+1) \times N_e, \\ N_{DG} &= 5\frac{(p_{DG}+1)(p_{DG}+2)}{2}N_e, \\ \text{and } N_{IPDG} &= (p_{IPDG}+1)(p_{IPDG}+2)N_e. \end{aligned} \tag{4.2.1}$$

So, if $p_{DG} = p_{IPDG} = p_{HDG} = p$ and if the number of elements N_e is the same for the three methods

$$N_{HDG} = \frac{6}{5(p+2)}N_{DG} = \frac{3}{(p+2)}N_{IPDG}.$$

For $p \geq 1$, the ratio $\dfrac{6}{5(p+2)}$ will always be smaller than 1, and thus the number of unknowns of the HDG method will always be smaller than the one of a classical DG method. This explains why when $p_{HGD} = p_{DG}$, the HGD method needs less memory space.
When $p = 1$ the global number of unknowns of the IPDG formulation is equal to the one of the HDG method, as observed in tab. 4.2.4. Then when $p > 1$, N_{HDG} becomes smaller than N_{IPDG}.
Tabs. 4.2.1 and 4.2.2 show that if τ is set to an appropriate value ($\tau = v_p$), the accuracy of the HDG method is equivalent to the one of the UDG method (and of the IPDG method). However, it is not always possible to determine the optimal value of τ. Tabs. 4.2.8 and 4.2.9 also show that when τ is far from this optimal value ($\tau = 1$), then one can recover the same accuracy as the UDG method by increasing the interpolation order by 1. Even with this increase of the interpolation degree, the numerical costs of the HDG method remains below than the ones of the UDG method. Indeed this numerical costs strongly depends on N_{HDG}, N_{DG} and N_{IPDG} which represent the size of the system to be inverted. From (4.2.1), we deduce that

$$N_{HDG} \leq N_{DG} \text{ and } N_{HDG} \leq N_{IPDG},$$

when

$$p_{HDG} \leq \frac{5}{6}(p_{DG}+1)(p_{DG}+2) - 1 \quad \text{or} \quad p_{HDG} \leq \frac{1}{3}(p_{IPDG}+1)(p_{IPDG}+2) - 1.$$

For example, if $p_{DG} = 1$, the HDG scheme will require less memory space than the UDG formulation if $p_{HDG} \leq 4$. When we compare the HDG and the IPDG memory space, if $p_{IPDG} = 1$, p_{HDG} has to be smaller than 3.

In tab. 4.2.7, the computational times of each method is presented. The construction of the HDG global matrix requires more computational time than the construction of the UDG matrix. This additional time is largely compensated by the time required for the resolution. Finally, the total simulation time is also smaller when we use the HDG scheme.

In tabs. 4.2.8 and 4.2.9, we compare the computational performances of the UDG method and the HDG method for a target accuracy level. We note that in this comparison we have taken a value of the parameter $\tau = 1$ which is not optimal. We want to emphasize once again that for a given accuracy, the HDG method is still more competitive than the UDG method. When the computations are done on the same mesh (tab. 4.2.8), to obtain the same accuracy level, the HDG method requires one interpolation order more than the UDG method. Despite this, it is less expensive in memory terms. In tab. 4.2.9 we use the same interpolation order for the two schemes. In this case, the HDG method needs a more refined mesh, but, except for $p = 1$, is still less expensive in terms of the memory space.

We present the pattern of the global matrix in fig. 4.2.1 for the HDG method, fig. 4.2.2 for the UDG method and fig. 4.2.3 for the IPDG method. For mesh M2 and $p = 2$, the three matrices are

h (m)	Interpolation degree	Mean Error V_x			Relative Error V_x (%)		
		HDG $\tau = 1$	HDG $\tau = v_p$	UDG	HDG $\tau = 1$	HDG $\tau = v_p$	UDG
625.0		0.2	7.8e-02	1.8e-02	52.8	25.5	6.0
312.5	1	1.3e-02	5.3e-04	1.7e-03	13.7	0.5	1.8
56.25		1.2e-03	1.3e-04	9.9e-05	5.5	0.6	0.4
625.0		6.7e-03	1.5e-03	1.6e-03	2.1	0.5	0.5
312.5	2	5.9e-04	7.1e-06	5.7e-05	0.6	7.3e-03	5.8e-02
56.25		3.1e-05	1.3e-06	1.4e-06	0.1	5.8e-03	6.1e-03
625.0		5.4e-04	1.4e-04	1.3e-04	0.2	4.4e-02	4.1e-02
312.5	3	2.6e-05	1.7e-07	3.1e-06	2.7e-02	1.7e-04	3.2e-03
56.25		6.5e-07	3.7e-08	3.7e-08	2.9e-03	1.7e-04	1.7e-04
625.0		3.9e-05	1.0e-05	1.0e-05	1.2e-02	3.2e-03	3.2e-03
312.5	4	9.5e-07	1.8e-09	6.8e-08	9.7e-04	1.8e-06	7.0e-05
56.25		1.1e-08	3.1e-10	3.1e-10	4.9e-05	1.4e-06	1.4e-06

Tab. 4.2.1: Plane wave propagation in an homogeneous medium: mean and relative errors on V_x.

h (m)	Interpolation degree	Mean Error σ_{xx}			Relative Error σ_{xx} (%)		
		HDG $\tau = 1$	HDG $\tau = v_p$	UDG	HDG $\tau = 1$	HDG $\tau = v_p$	UDG
625.0		472.1	197.3	72.9	38.8	16.2	6.0
312.5	1	33.3	8.0	6.6	8.7	0.5	1.7
56.25		2.0	0.5	0.4	2.2	0.6	0.4
625.0		6.4	6.2	5.9	0.5	0.5	0.5
312.5	2	0.2	0.1	0.3	5.0e-2	7.7e-03	6.5e-02
56.25		3.6e-03	6.7e-03	6.2e-03	4.0e-3	7.4e-03	6.9e-03
625.0		0.4	0.4	0.4	3.2e-02	3.8e-02	3.1e-02
312.5	3	8.7e-03	2.5e-03	1.0e-02	2.2e-03	1.6e-04	2.7e-03
56.25		1.2e-04	1.5e-04	1.4e-04	1.4e-04	1.6e-04	1.5e-04
625.0		3.5e-02	3.7e-02	3.4e-02	2.7e-03	2.9e-03	2.7e-03
312.5	4	2.8e-04	3.4e-05	3.1e-05	7.1e-05	2.2e-05	7.9e-06
56.25		1.4e-06	1.7e-06	1.5e-06	1.5e-06	1.9e-06	1.7e-06

Tab. 4.2.2: Plane wave propagation in an homogeneous medium: mean and relative errors on σ_{xx}.

h (m)	Interpolation degree	Convergence order HDG	UDG
625.0		-	-
312.5	1	2.5	1.7
56.25		2.0	2.0
625.0		-	-
312.5	2	3.3	2.9
56.25		3.5	3.2
625.0		-	-
312.5	3	3.9	3.7
56.25		4.1	4.2
625.0		-	-
312.5	4	5.3	5.2
56.25		5.6	5.6

Tab. 4.2.3: Plane wave propagation in an homogeneous medium: convergence order of the HDG and UDG methods.

# Mesh elements	Interpolation degree	Non-zeros terms		
		HDG scheme	UDG scheme	IPDG scheme
3100		4.5e+05	1.5e+06	4.5e+05
10300	1	1.5e+06	5.1e+06	1.5e+06
45000		6.4e+06	2.2e+07	6.4e+06
3100		1.0e+06	4.3e+06	1.8e+06
10300	2	3.3e+06	1.4e+07	5.9e+06
45000		1.4e+07	6.2e+07	2.6e+07
3100		1.8e+06	9.4e+06	4.9e+06
10300	3	5.9e+06	3.1e+07	1.6e+07
45000		2.6e+07	1.3e+08	7.1e+07
3100		2.8e+06	1.8e+07	1.1e+07
10300	4	9.2e+06	5.9e+07	3.7e+07
45000		4.0e+07	2.6e+08	1.6e+08

Tab. 4.2.4: Plane wave propagation in an homogeneous medium: number of non-zero terms in the global matrix.

# Mesh elements	Interpolation degree	# dof			#dof/wavelength		
		HDG	UDG	IPDG	HDG	UDG	IPDG
3100	1	1.9e+04	4.7e+04	1.9e+04	9	23	9
10300		6.2e+04	1.5e+05	6.2e+04	31	77	31
45000		2.7e+05	6.7e+05	2.6e+05	134	334	134
3100	2	2.8e+04	9.4e+04	3.7e+04	14	46	19
10300		9.3e+04	3.1e+05	1.2e+05	46	154	61
45000		4.0e+05	1.3e+06	5.3e+05	201	668	267
3100	3	3.8e+04	1.6e+05	6.2e+04	19	78	31
10300		1.2e+05	5.1e+05	2.1e+05	62	256	102
45000		5.4e+05	2.2e+06	8.9e+05	268	1114	445
3100	4	4.7e+04	2.3e+05	9.4e+04	23	117	47
10300		1.5e+05	7.7e+05	3.1e+05	77	384	154
45000		6.7e+05	3.3e+06	1.3e+06	335	1671	668

Tab. 4.2.5: Plane wave propagation in an homogeneous medium: total number degrees of freedom (ndof) and ndof by wavelength (λ_w) for both methods

# Mesh elements	Interpolation degree	Memory (MB)			Memory ratio		
		HDG	UDG	IPDG	HDG	UDG	IPDG.
3100	1	44	288	58	1	6.55	1.32
10300		161	1076	221	1	6.68	1.37
45000		797	5492	1156	1	6.89	1.45
3100	2	97	804	215	1	8.29	2.22
10300		355	3097	852	1	8.72	2.4
45000		1746	15965	4454	1	9.14	2.55
3100	3	170	1656	598	1	9.74	3.52
10300		624	6600	2394	1	10.58	3.84
45000		3080	34597	12362	1	11.23	4.01
3100	4	254	2749	1324	1	10.82	5.21
10300		947	10098	5251	1	10.66	5.54
45000		4653	50297	27314	1	10.81	5.87

Tab. 4.2.6: Plane wave propagation in an homogeneous medium: memory consumption.

# Mesh elements	Interpolation degree	Construction time (s)			Solution time (s)		
		HDG	UDG	IPDG	HDG	UDG	IPDG
3100	1	0.2	4.0e-02	1.0	0.2	1.5	0.3
10300		0.8	0.1	3.4	1.0	7.2	1.6
45000		3.3	0.7	14.4	6.1	68.0	11.9
3100	2	1.3	0.1	3.9	0.6	6.0	1.6
10300		2.8	0.3	13.0	2.3	28.8	9.2
45000		12.2	1.5	55.8	15.4	224.5	79.8
3100	3	4.8	0.2	10.9	1.0	14.4	6.2
10300		8.1	0.8	35.4	4.4	78.2	38.1
45000		35.8	3.4	154.3	32.2	643.2	337.8
3100	4	5.8	0.5	24.0	1.6	28.1	18.7
10300		21.1	1.8	79.4	8.7	135.2	118.2
45000		106.1	7.6	341.4	70.7	1077.4	1068.0

Tab. 4.2.7: Plane wave propagation in an homogeneous medium: time required for the global matrix construction and for the system resolution.

Level of Error	# Mesh elements	Interpolation degree		Memory (MB)		Construction time (s)		Solution time (s)	
		HDG	UDG	HDG	UDG	HDG	UDG	HDG	UDG
1e-02	3100	2	1	97	288	1.3	4.0e-02	0.6	1.5
1e-03	3100	3	2	170	804	4.8	0.1	1.0	6.0
1e-03	10300	2	1	355	1076	2.8	0.1	2.3	7.2
1e-04	3100	4	3	254	1656	5.8	0.2	1.6	14.4

Tab. 4.2.8: Plane wave propagation in an homogeneous medium: for the same level of error, comparison between computational time and memory required with a same mesh for both methods, with $\tau = 1$ in the HDG method.

Level of Error	Interpolation degree	# Mesh elements		Memory (MB)		Construction time (s)		Solution time (s)	
		HDG	UDG	HDG	UDG	HDG	UDG	HDG	UDG
1e-02	1	45000	3100	797	288	3.3	4.0e-02	6.1	1.5
1e-03	2	10300	3100	355	804	2.8	0.1	2.3	6.0
1e-04	3	10300	3100	624	1656	8.1	0.2	4.4	14.4

Tab. 4.2.9: Plane wave propagation in an homogeneous medium: for the same level of error, comparison between computational time and memory required with a same interpolation order for both methods, with $\tau = 1$ in the HDG method.

sparse matrices as expected. The fill rate of the HDG matrix is 0.038%, the one of the UDG matrix is 0.015%, and the one of the IPDG matrix is 0.041%. The advantage of the UDG matrix is that it is sparser than the HDG one, but its size is larger (around 3 times more).
The structure of the HDG matrix is close to the one of the IPDG matrix. We distinguish four blocks confirming that the unknowns are gathered in a vector and that in this unknown vector we have two distinct parts (corresponding to λ_x and λ_z in the HDG case and V_x and V_z in the IPDG case). However, the pattern of each of the four blocks of the HDG matrix is similar to the pattern of the global UDG matrix.

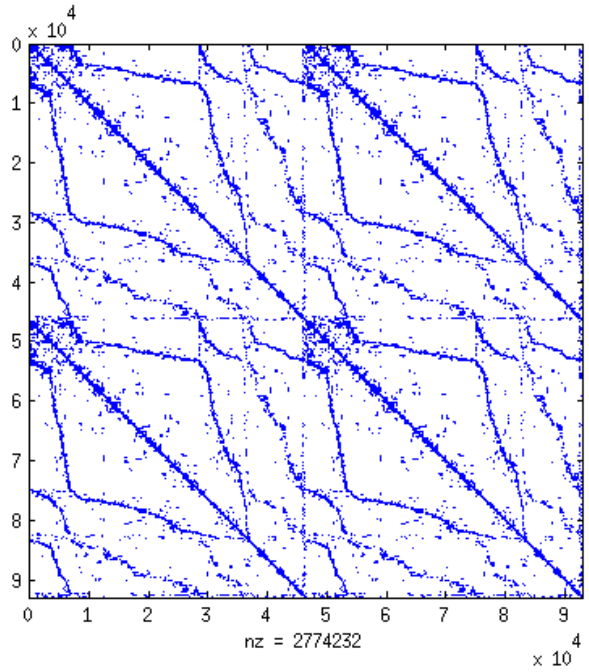


Fig. 4.2.1: Plane wave propagation in an homogeneous medium: pattern of the HDG matrix for the mesh M2 and an interpolation degree of 2

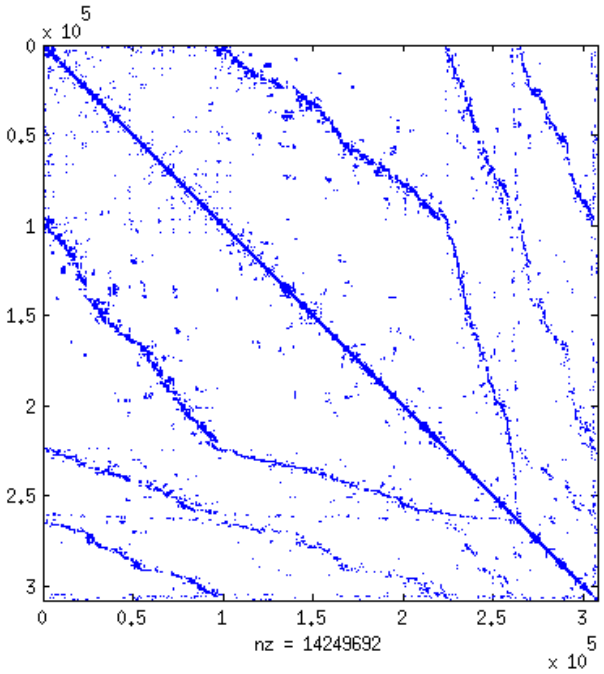


Fig. 4.2.2: Plane wave propagation in an homogeneous medium: pattern of the UDG matrix for the mesh M2 and an interpolation degree of 2

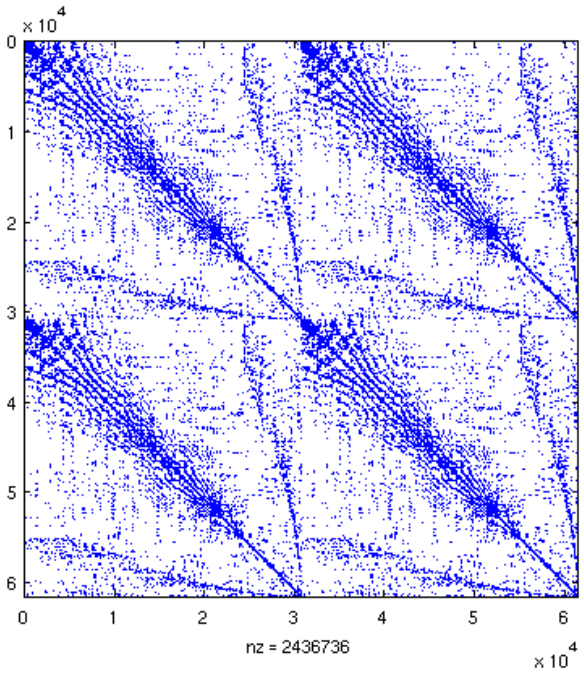


Fig. 4.2.3: Plane wave propagation in an homogeneous medium: pattern of the IPDG matrix for the mesh M2 and an interpolation degree of 2

Looking at the computational costs required for the resolution of the system in tabs. 4.2.6 and 4.2.7, it is clear that the sparsity of the upwind DG matrix does not compensate the higher number of degrees of freedom.

4.3 Disk-shaped scatterer

We now complicate the geometry of the computational domain. We consider the scattering of an incident plane wave $\mathbf{v}^{inc}$ by a circle of radius r on which we impose a free surface condition. The solution of this problem can be computed analytically and we give the expression of the exact solution in appendix B.
We remind that when a wave encounters an interface between two media, it generates two kinds of waves (see fig. 4.3.1) : the reflected wave and the transmitted wave.

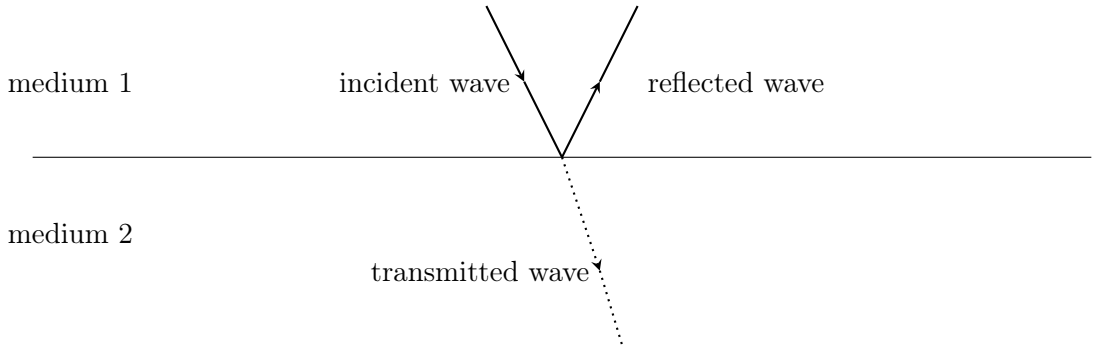


Fig. 4.3.1: Scattering of a wave.

In this example, the computational domain Ω (see fig. 4.3.2) is the ring included between the circle of radius $a = 2000$ m whose boundary Γ_l corresponds to the free boundary, and the concentric circle of radius $b = 8000$ m whose boundary Γ_a is assumed to be an artificial boundary. If we denote

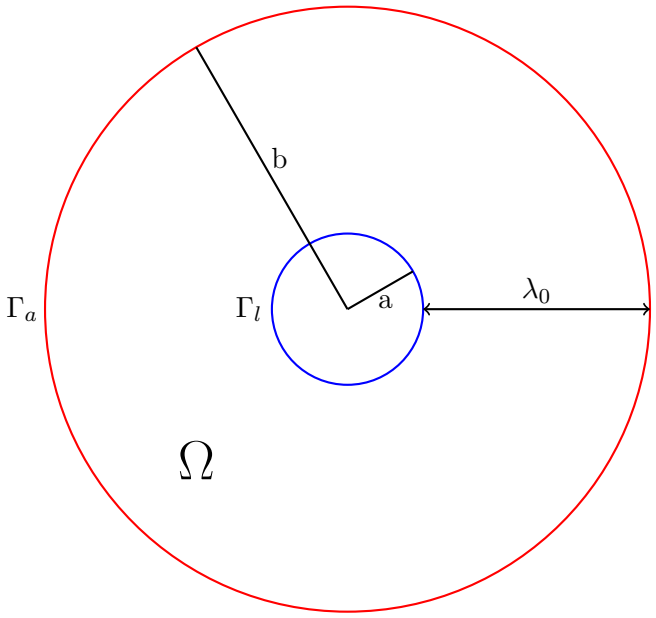


Fig. 4.3.2: Disk-shaped scatterer: configuration of the computational domainl Ω

by $\left(\mathbf{v}^s, \underline{\underline{\sigma}}^s\right)$ the scattered field and $\left(\mathbf{v}^t, \underline{\underline{\sigma}}^t\right)$ the total field, in the domain Ω, the total field satisfies

$$\mathbf{v}^t = \mathbf{v}^s + \mathbf{v}^{inc} \text{ and } \underline{\underline{\sigma}}^t = \underline{\underline{\sigma}}^s + \underline{\underline{\sigma}}^{inc}.$$

Moreover, we remind that the fields $\left(\mathbf{v}^t, \underline{\underline{\sigma}}^t\right)$, $\left(\mathbf{v}^s, \underline{\underline{\sigma}}^s\right)$ and $\left(\mathbf{v}^{inc}, \underline{\underline{\sigma}}^{inc}\right)$ satisfy the wave equations (4.3.1) without source term

$$\begin{cases} i\omega\rho\mathbf{v} &= \nabla\cdot\underline{\underline{\sigma}} & \text{in } \Omega, \\ i\omega\underline{\underline{\sigma}} &= \underline{\underline{C}}\,\underline{\underline{\epsilon}}(\mathbf{v}) & \text{in } \Omega. \end{cases} \tag{4.3.1}$$

In this problem, the total field verifies the free surface condition (1.1.11) on Γ_l

$$\underline{\underline{\sigma}}^t\mathbf{n} = 0.$$

On Γ_a, the absorbing condition (1.1.13) for an isotropic medium is only applied on the scattered field

$$\underline{\underline{\sigma}}^s\cdot\mathbf{n} - \rho v_p\left(\mathbf{v}^s\cdot\mathbf{n}\right)\mathbf{n} - \rho v_s\left(\mathbf{v}^s\cdot\mathbf{t}\right)\mathbf{t} = 0.$$

In the following, to simplify, we formulate this condition as $B(\mathbf{v}^s, \underline{\underline{\sigma}}^s) = 0$.
We thus have two choices for computing the numerical solution of the scattering problem:

1. We compute the total field $\left(\mathbf{v}^t, \underline{\underline{\sigma}}^t\right)$, and then we deduce the scattered field $\left(\mathbf{v}^s, \underline{\underline{\sigma}}^s\right)$, i.e. we solve the wave equations for $\left(\mathbf{v}^t, \underline{\underline{\sigma}}^t\right)$ taking the following boundary conditions

$$\begin{cases} \underline{\underline{\sigma}}^t\mathbf{n} = 0 \quad \text{on } \Gamma_l, \\ B(\mathbf{v}^s, \underline{\underline{\sigma}}^s) = 0 \Leftrightarrow B(\mathbf{v}^s, \underline{\underline{\sigma}}^s) + B(\mathbf{v}^{inc}, \underline{\underline{\sigma}}^{inc}) = B(\mathbf{v}^{inc}, \underline{\underline{\sigma}}^{inc}) \Leftrightarrow B(\mathbf{v}^t, \underline{\underline{\sigma}}^t) = B(\mathbf{v}^{inc}, \underline{\underline{\sigma}}^{inc}) \quad \text{on } \Gamma_a, \end{cases}$$

2. We directly compute the scattered field $\left(\mathbf{v}^s, \underline{\underline{\sigma}}^s\right)$ solving the wave equations for $\left(\mathbf{v}^s, \underline{\underline{\sigma}}^s\right)$ with

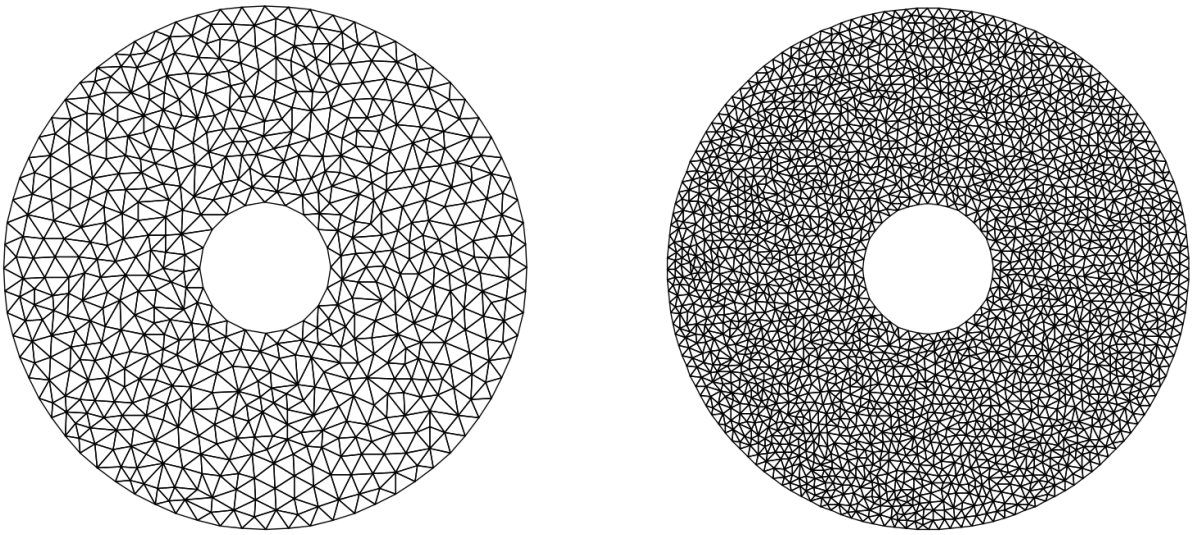
$$\begin{cases} \underline{\underline{\sigma}}^t\mathbf{n} = 0 \Leftrightarrow \underline{\underline{\sigma}}^s = -\underline{\underline{\sigma}}^{inc} \quad \text{on } \Gamma_l, \\ B(\mathbf{v}^s, \underline{\underline{\sigma}}^s) = B(\mathbf{v}^{inc}, \underline{\underline{\sigma}}^{inc}) \quad \text{on } \Gamma_a, \end{cases}$$

The HDG and the UDG solutions are computed with the first approach, whereas the IPDG solution is computed with the second option.

The characteristics of the homogeneous medium chosen for the experiment are a mass density $\rho = 1000$ kg.m^{-3} and Lamé's coefficients $\lambda = 8$ MPa and $\mu = 4$ MPa, which imply a P-waves velocity v_p equal to 4000 m.s^{-1} and a $S-$waves velocity v_s equal to 2000 m.s^{-1}.
In Ω, the distance between circles of radius a and b corresponds to 6 times the wavelength λ_o. We remind that $\lambda_o = \dfrac{v_p}{f}$, where f is the frequency. f is taken equal to 4 Hz. In these conditions, the radius a is equal to 2 times the wavelength λ_o and the radius b to 8 times.

We discretize Ω using three unstructured meshes: one composed of 1200 elements (fig. 4.3.3a), one of 5100 elements (fig. 4.3.3b) and one of 21000 elements, which is not represented because it is too fine. The characteristics of these meshes are summarized in table 4.3.1.

We plot the exact solution of the scattering problem for V_x component on fig. 4.3.4. The numerical solution for V_x computed on the second mesh and using the HDG-$\mathbb{P}_2$ scheme is presented on fig. 4.3.5a; and using the UDG-$\mathbb{P}_2$ scheme on fig. 4.3.6a. Whereas the numerical solution computed with the UDG method fits well with the exact solution, the numerical solution computed with the HDG method is not accurate enough. Comparisons with the exact solution are given in fig. 4.3.5b for the HDG method and in fig. 4.3.6b for the UDG method. To obtain a better numerical solution with the HDG method, we can increase the interpolation order, keeping the same mesh (fig. 4.3.7a) or we can refine



(a) Mesh M1, 1200 elements.

(b) Mesh M2, 5100 elements.

Fig. 4.3.3: Disk-shaped scatterer: discretization of Ω.

Mesh	# Mesh elements	# Mesh vertices	h_{min}	h_{max}	h_{max}/h_{min}
M1	1200	640	440	1016	2.31
M2	5100	2630	212	490	2.31
M3	21000	11000	105	245	2.32

Tab. 4.3.1: Disk-shaped scatterer: characteristics of the three meshes

the mesh (fig. 4.3.8a), keeping the same interpolation order.

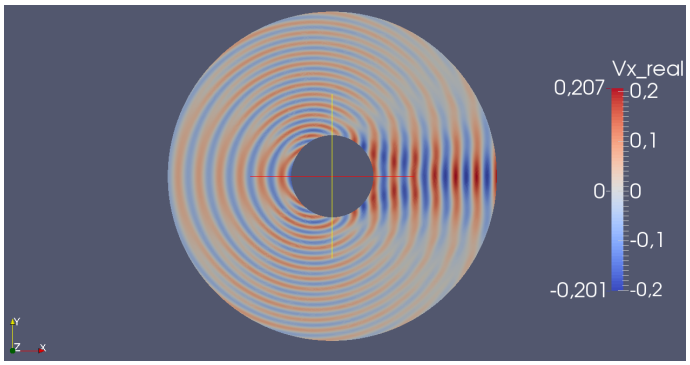
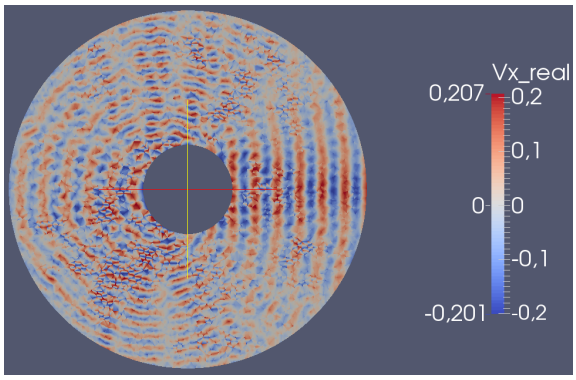
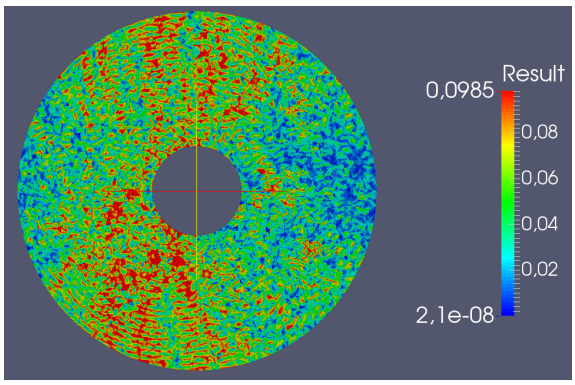


Fig. 4.3.4: Disk-shaped scatterer: exact solution, V_x component.

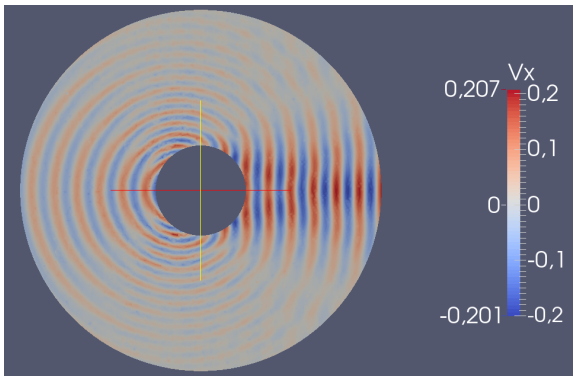


(a) Numerical solution

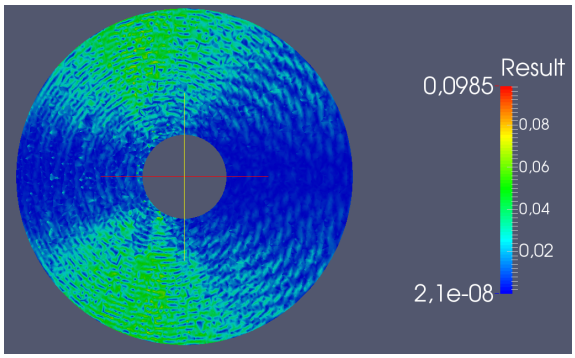


(b) Absolute error between the exact solution and the numerical solution

Fig. 4.3.5: Disk-shaped scatterer: numerical solution, mesh M2, HDG-$\mathbb{P}_2$ scheme, V_x component.

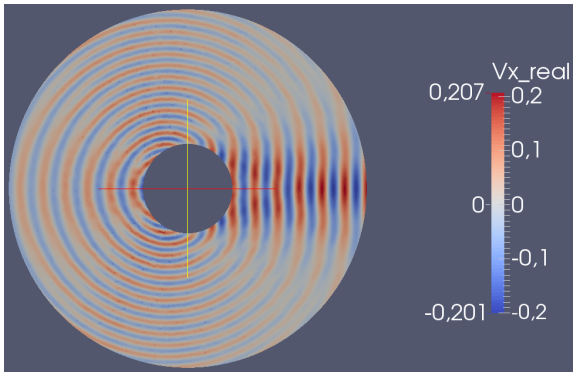


(a) Numerical solution

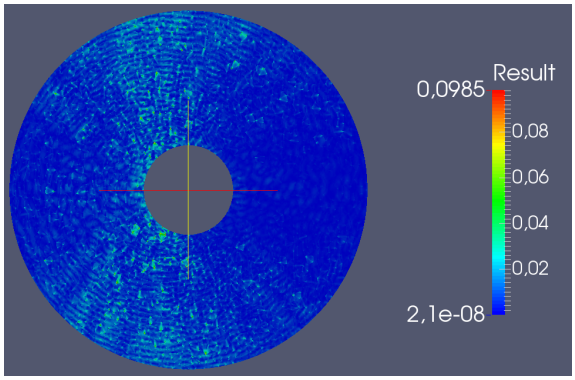


(b) Absolute error between the exact solution and the numerical solution

Fig. 4.3.6: Disk-shaped scatterer: numerical solution, mesh M2, UDG-$\mathbb{P}_2$ scheme, V_x component.

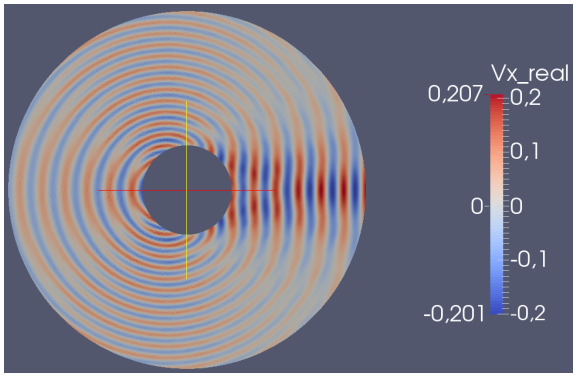


(a) Numerical solution

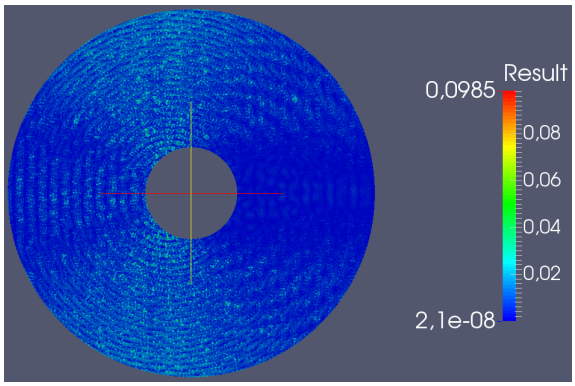


(b) Absolute error between the exact solution and the numerical solution

Fig. 4.3.7: Disk-shaped scatterer: numerical solution, mesh M2, HDG-$\mathbb{P}_3$ scheme, V_x component.



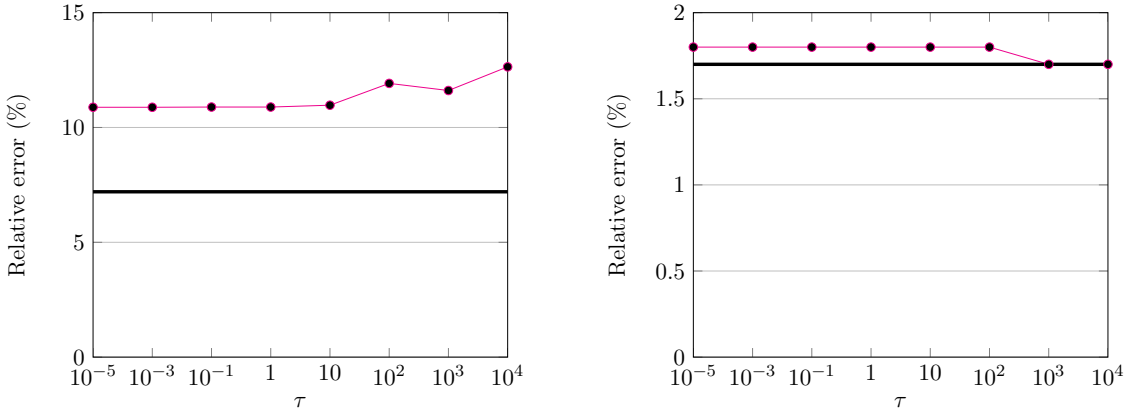
(a) Numerical solution



(b) Absolute error between the exact solution and the numerical solution

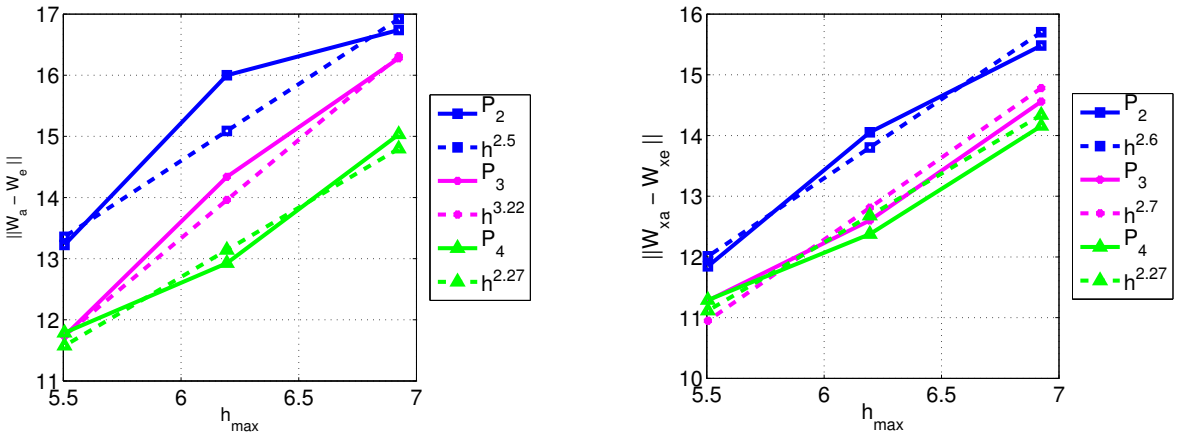
Fig. 4.3.8: Disk-shaped scatterer: numerical solution, mesh M3, HDG-$\mathbb{P}_2$ scheme, V_x component.

As we have done in section 3.4.1 for the propagation of a plane wave, we study the influence of the value of the parameter τ. With a circular computational domain, we see on figs. 4.3.9a and 4.3.9b that the parameter τ seems to have no influence on the accuracy of the solution. This is due to the fact that the geometric error dominates since the circle is not discretized by curved elements. Hence, in this test case, the geometric error is greater than the numerical error resulting from the inappropriate value of τ. The fact that the HDG solution is computed on the boundaries of the element makes it more sensitive to the geometric error and explains why the influence of the parameter τ is hidden if the geometric error is large.
This is confirmed by the convergence curves fig. 4.3.10a for the HDG method and fig. 4.3.10b for the upwind DG method. The two numerical convergence curves have the same behavior and do not achieve the optimal rate. The main reason is that the curved boundaries are discretized by affine elements and this is a limitation for obtaining higher convergence orders.



(a) HDG-$\mathbb{P}_3$, mesh M2 (b) HDG-$\mathbb{P}_3$, mesh M3

Fig. 4.3.9: Disk-shaped scatterer: variation of the relative error as a function of the parameter τ.



(a) HDG method. (b) Upwind DG method.

Fig. 4.3.10: Disk-shaped scatterer: convergence order of the HDG and UDG methods.

In tab. 4.3.4 we have gathered the memory space and computational time required to obtain the same error level with the two methods on a same mesh. We observe that the modification of the geometry of the computational domain has no influence on the computational performances of the HDG method. It is still less expensive than the UDG method, even when we use it with one interpolation order more than the UDG method on a same mesh. To obtain a same error level on this example, we need 3 times less memory and 2 times less computational time with the HDG method. When we compare the mean and relative errors for V_x and σ_{xx} components (tabs. 4.3.2 and 4.3.3), we observe that, except for the coarsest mesh (mesh M1) and for $p = 2$, we have the same error level for both methods. As said when we have looked at the influence of the parameter τ for the numerical convergence, this is due to the geometric error which dominates over the numerical error due to the method used.
On the other hand, the geometric error has no effect on the memory requirement and the computational time, see them tabs. 4.3.6 and 4.3.8. As we have seen in the previous test case, the HDG scheme requires less memory than the UDG method (around 10 times less for this test case). The construction of the HDG global matrix needs more time than the UDG method but this additional time is widely

compensated by the resolution time. Finally, as for the previous test case, the whole HDG computation needs less time than the upwind DG simulation.
In tabs. 4.3.6, 4.3.5, 4.3.8, and 4.3.7, as we have done previously, we also compare the computational performances of the HDG method to the ones of the IPDG scheme and we deduce the same conclusions.

h (m)	Interpolation degree	Mean Error V_x		Relative Error V_x (%)	
		HDG scheme	UDG scheme	HDG scheme	UDG scheme
1016.9	2	2.2	1.0	273	123
490.1		0.3	8.1e-02	122	36
245.9		6.3e-03	1.8e-03	11.1	3.2
1016.9	3	1.5	0.5	154	56
490.1		2.5e-02	1.6e-02	10.9	7.2
245.9		1.0e-03	9.6e-04	1.8	1.7
1016.9	4	0.5	0.4	48.5	37.3
490.1		1.2e-02	1.3e-02	5.4	5.5
245.9		9.9e-04	9.9e-04	1.7	1.7

Tab. 4.3.2: Disk-shaped scatterer: mean and relative errors on V_x.

h (m)	Interpolation degree	Mean Error σ_{xx}		Relative Error σ_{xx} (%)		Convergence order	
		HDG scheme	UDG scheme	HDG scheme	UDG scheme	HDG scheme	UDG scheme
625.0	2	7101	3802	242	128	-	-
312.5		635	159	83	20	1.0	2.0
56.25		9.9	4.8	5.1	2.5	4.0	3.2
625.0	3	3687	1195	112	36	-	-
312.5		43	38	5.6	4.9	4.0	2.7
56.25		2.9	3.0	1.5	1.5	2.3	1.9
625.0	4	1033	743	31.0	22.3	-	-
312.5		30.2	31.0	3.9	3.4	2.9	2.4
56.25		2.9	3.0	1.5	1.5	1.6	1.6

Tab. 4.3.3: Disk-shaped scatterer: mean and relative errors on σ_{xx} and convergence order.

Level of Error	# Mesh elements	Interpolation degree		Memory (MB)		Construction time (s)		Solution time (s)	
		HDGm	UDGm	HDGm	UDGm	HDGm	UDGm	HDGm	UDGm
1e-02	5100	3	2	61	152	4.3	0.3	3.4	14.7
1e-02	5100	4	3	462	2921	10.9	0.7	6.2	38.2
1e-03	21000	3	2	246	611	16.5	1.2	26.4	93.7

Tab. 4.3.4: Disk-shaped scatterer: for the same level of error, comparison between computational time and memory required with the same mesh for both methods.

# Mesh elements	Interpolation degree	Non-zeros terms		
		HDG scheme	UDG scheme	IPDG scheme
1200	2	3.9e+05	1.6e+06	6.7e+05
5100		1.6e+06	7.0e+06	2.9e+06
21000		6.6e+06	2.7e+07	1.2e+07
1200	3	6.9e+05	3.6e+06	1.9e+06
5100		2.9e+06	1.5e+07	8.0e+06
21000		1.2e+07	6.2e+07	3.2e+07
1200	4	1.1e+06	6.8e+06	4.2e+06
5100		4.6e+06	2.9e+07	1.8e+07
21000		1.8e+07	1.2e+08	7.3e+07

Tab. 4.3.5: Disk-shaped scatterer: number of non-zero terms in the global matrix

# Mesh elements	Interpolation degree	Memory (MB)			Memory ratio		
		HDGm	UDGm	IPDGm	HDGm	UDGm	IPDGm
1200	2	44	269	69	1	6.1	1.6
5100		179	1360	375	1	7.6	2.1
21000		783	6578	1776	1	8.4	2.3
1200	3	70	525	187	1	7.5	2.7
5100		309	2921	1006	1	9.5	3.3
21000		1384	14131	4952	1	10.2	3.6
1200	4	100	895	418	1	9.0	4.2
5100		462	4537	2251	1	9.8	4.9
21000		2101	21186	11074	1	10.1	5.3

Tab. 4.3.6: Disk-shaped scatterer: number of non-zero terms in the global matrix and memory used.

# Mesh elements	Interpolation degree	# dof			#dof/wavelength		
		HDGm	UDGm	IPDGm	HDGm	UDGm	IPDGm
1200	2	1.1e+04	3.6e+04	1.4e+04	11	35	14
5100		4.6e+04	1.5e+05	6.1e+04	46	152	61
21000		1.8e+05	6.1e+05	2.4e+05	184	611	244
1200	3	1.5e+04	6.0e+04	2.4e+04	14	59	24
5100		6.1e+04	2.5e+05	1.0e+05	61	253	101
21000		2.5e+05	1.0e+06	4.1e+05	246	1018	407
1200	4	1.8e+04	8.9e+04	3.6e+04	18	89	36
5100		7.7e+04	3.8e+05	1.5e+05	77	380	152
21000		3.1e+05	1.5e+06	6.1e+05	307	1527	611

Tab. 4.3.7: Disk-shaped scatterer: total number degrees of freedom (ndof) and ndof by wavelength (λ_w) for both methods

# Mesh elements	Interpolation degree	Construction time(s)			Solution time (s)		
		HDGm	UDGm	IPDGm	HDGm	UDGm	IPDGm
1200	2	0.4	7.1e-2	1.8	0.3	2.6	0.5
5100		1.5	0.3	7.9	1.7	14.7	3.9
21000		5.5	1.2	31.9	12.0	93.7	26.1
1200	3	1.1	0.2	5.1	0.6	5.3	1.7
5100		4.3	0.7	21.9	3.4	38.2	14.0
21000		16.5	2.7	90.6	26.4	249.3	107.2
1200	4	2.8	0.3	10.7	0.9	10.2	5.0
5100		10.9	1.4	45.7	6.2	65.6	42.3
21000		43.7	5.5	184.5	49.5	447.3	336.2

Tab. 4.3.8: Disk-shaped scatterer: time required for the global matrix construction and for the system resolution.

4.4 Scattering by an elastic circle

The next test problem considered is the scattering of a plane wave by an elastic disk-solid, which corresponds to an heterogeneous wave propagation problem.

The computational domain Ω, fig. 4.4.1, is composed of the circle Ω_a and the ring Ω_b, so $\Omega = \Omega_a \cup \Omega_b$.

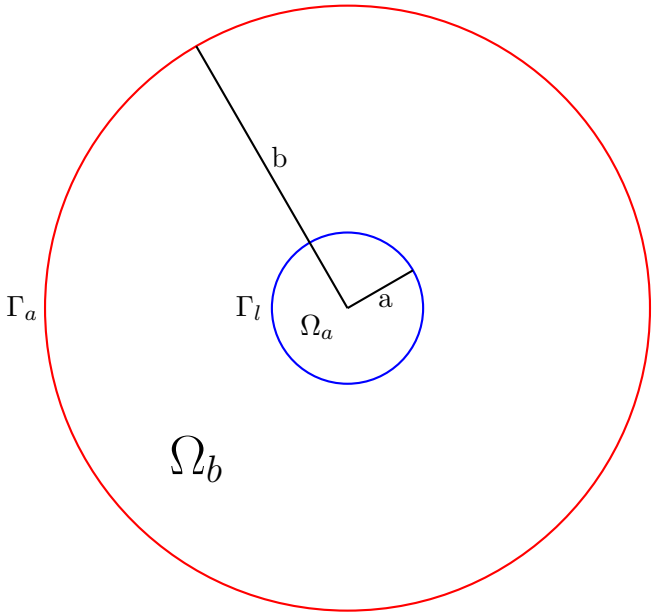


Fig. 4.4.1: Elastic solid scatterer problem: configuration of the computational domain Ω.

As for the previous case test, we note $\left(\mathbf{v}^s, \underline{\underline{\sigma}}^s\right)$ the scattered field, $\left(\mathbf{v}^t, \underline{\underline{\sigma}}^t\right)$ the total field and $\left(\mathbf{v}^r, \underline{\underline{\sigma}}^r\right)$ the refracted field. According to fig. 4.3.1, in the configuration of the problem, the total field satisfies

$$\left\{ \begin{array}{lllllllll} \mathbf{v}^t & = & \mathbf{v}^{t_a} & = & \mathbf{v}^r & \text{and} & \underline{\underline{\sigma}}^t & = & \underline{\underline{\sigma}}^{t_a} = \underline{\underline{\sigma}}^r & \text{in } \Omega_a, \\ \mathbf{v}^t & = & \mathbf{v}^{t_b} & = & \mathbf{v}^s + \mathbf{v}^{inc} & \text{and} & \underline{\underline{\sigma}}^t & = & \underline{\underline{\sigma}}^{t_b} = \underline{\underline{\sigma}}^s + \underline{\underline{\sigma}}^{inc} & \text{in } \Omega_b. \end{array} \right.$$

with $\left(\mathbf{v}^s, \underline{\underline{\sigma}}^s\right)$, $\left(\mathbf{v}^t, \underline{\underline{\sigma}}^t\right)$, $\left(\mathbf{v}^{inc}, \underline{\underline{\sigma}}^{inc}\right)$ and $\left(\mathbf{v}^r, \underline{\underline{\sigma}}^r\right)$ satisfying the wave equations (4.3.1) without source term. The variables $\left(\mathbf{v}^{t_a}, \underline{\underline{\sigma}}^{t_a}\right)$ denote the total field in Ω_a and $\left(\mathbf{v}^{t_b}, \underline{\underline{\sigma}}^{t_b}\right)$ the total field in Ω_b.
On the boundary Γ_a, we impose the absorbing condition (1.1.13) to simulate the infinite circle that we denote $B(u, \underline{\underline{\sigma}})$ for sake of simplicity. The boundary Γ_l is the interface between the two domains Ω_a and Ω_b, and transmission conditions have to be imposed on it in order to ensure the propagation of the wave in the second medium. The classical transmission conditions are

$$\underline{\underline{\sigma}}^{t_a}\mathbf{n} = \underline{\underline{\sigma}}^{t_b}\mathbf{n} \quad \text{and} \quad \mathbf{v}^{t_a} = \mathbf{v}^{t_b} \quad \text{on } \Gamma_l.$$

The solution of this problem can be computed analytically and we give the expression of the exact solution in appendix C.
To compute the numerical solution, we have to compute the solution in Ω_a and in Ω_b. In Ω_a, we have simply to solve the wave equations for $\left(\mathbf{v}^{t_a}, \underline{\underline{\sigma}}^{t_a}\right)$. In Ω_b, as for the scattering of a plane wave by an empty circle, we have two choices for computing the numerical solution:

1. We compute the total field $\left(\mathbf{v}^{t_b}, \underline{\underline{\sigma}}^{t_b}\right)$, and then we deduce the scattered field $\left(\mathbf{v}^s, \underline{\underline{\sigma}}^s\right)$ by substracting $\left(\mathbf{v}^{inc}, \underline{\underline{\sigma}}^{inc}\right)$, i.e. we solve the wave equations for $\left(\mathbf{v}^{t_b}, \underline{\underline{\sigma}}^{t_b}\right)$ taking the following boundary conditions

$$\left\{ \begin{array}{l} \underline{\underline{\sigma}}^{t_b}\mathbf{n} = \underline{\underline{\sigma}}^{t_a}\mathbf{n} \quad \text{and} \quad \mathbf{v}^{t_b} = \mathbf{v}^{t_a} \quad \text{on } \Gamma_l, \\ B(\mathbf{v}^s, \underline{\underline{\sigma}}^s) + B(\mathbf{v}^{inc}, \underline{\underline{\sigma}}^{inc}) = B(\mathbf{v}^{inc}, \underline{\underline{\sigma}}^{inc}) \Leftrightarrow B(\mathbf{v}^{t_b}, \underline{\underline{\sigma}}^{t_b}) = B(\mathbf{v}^{inc}, \underline{\underline{\sigma}}^{inc}) \quad \text{on } \Gamma_a, \end{array} \right.$$

2. We directly compute the scattered field $\left(\mathbf{v}^s, \underline{\underline{\sigma}}^s\right)$ solving the wave equations for $\left(\mathbf{v}^s, \underline{\underline{\sigma}}^s\right)$ with

$$\begin{cases} \underline{\underline{\sigma}}^{t_b}\mathbf{n} = \underline{\underline{\sigma}}^{t_a}\mathbf{n} \quad \text{and} \quad \mathbf{v}^{t_b} = \mathbf{v}^{t_a} \quad \text{on } \Gamma_l, \\ B(\mathbf{v}^s, \underline{\underline{\sigma}}^s) = B(\mathbf{v}^{inc}, \underline{\underline{\sigma}}^{inc}) \quad \text{on } \Gamma_a. \end{cases}$$

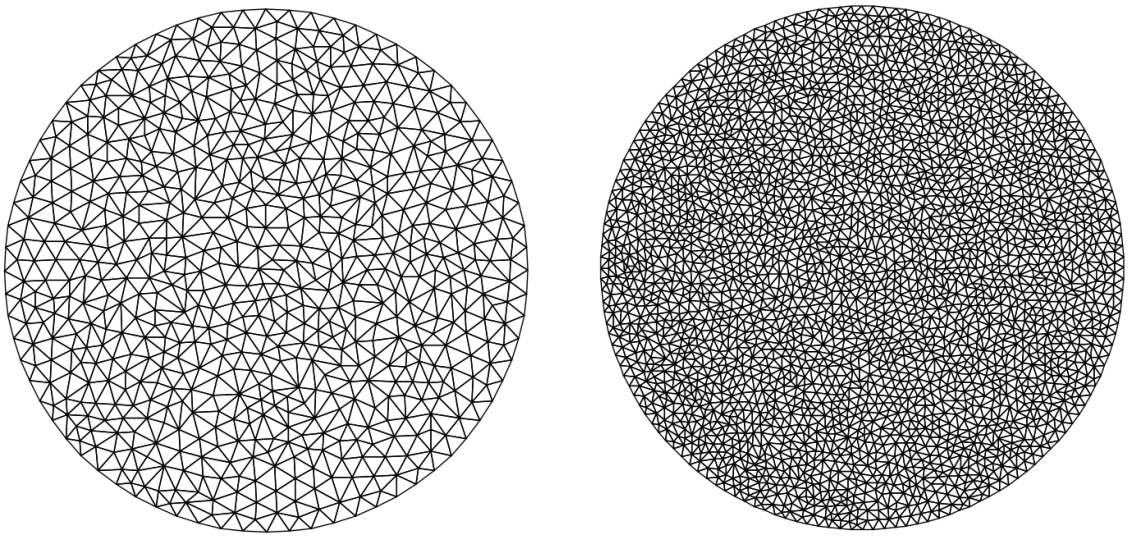
The HDG and the UDG solutions are computed with the first method, whereas the IPDG solution is computed with the second option.

The physical characteristics of the two media are gathered in tab. 4.4.2. The frequency taken for the simulation is equal to 4 Hz, and thus $\omega \simeq 25.13$.

Medium	ρ (kg.m^{-3})	λ (MPa)	μ (MPa)	v_p (m.s^{-1})	v_s (m.s^{-1})
Ω_a	2,0	6,4e+01	3,2e+01	8,0e+03	4,0e+03
Ω_b	1,0	8,0	4,0	4,0e+03	2,0e+03

Tab. 4.4.1: Elastic solid scatterer problem: characteristics of the two media.

The computational domain Ω is discretized by three unstructured meshes, as we have done in the two previous test cases. One is composed of 1300 elements (fig. 4.4.2a), one of 5400 elements (fig. 4.4.2b) and one of 22000 elements (which is not represented because it is too fine). Their features are presented on tab. 4.4.2.



(a) Mesh M1, 1300 elements

(b) Mesh M2, 5400 elements

Fig. 4.4.2: Elastic solid scatterer problem: discretization of Ω.

The exact solution of V_x component is represented on fig. 4.4.3. The comparison between this solution and the numerical solution computed with the HDG-$\mathbb{P}_2$ scheme on the mesh M2 (fig. 4.4.4a) is presented on fig. 4.4.4b. The comparison wih the numerical solution computed with the UDG-$\mathbb{P}_2$ scheme on the same mesh (fig. 4.4.5a) is given on fig. 4.4.5b. The observations are the same as the ones made for the disk-shaped scattering problem: the HDG solution is not enough accurate and suitable

Mesh	# Mesh elements	# Mesh vertices	h_{min}	h_{max}	h_{max}/h_{min}
M1	1300	700	440.52	1016.91	2.31
M2	5400	2800	211.56	490.08	2.32
M3	22000	11000	105.86	245.87	2.32

Tab. 4.4.2: Elastic solid scatterer problem: characteristics of the three meshes

for low interpolation order on the same coarse mesh. As previously, we increase the interpolation order of the HDG scheme (fig. 4.4.6a) and on fig. 4.4.7a, we refine the mesh, keeping the HDG-$\mathbb{P}_2$ scheme.

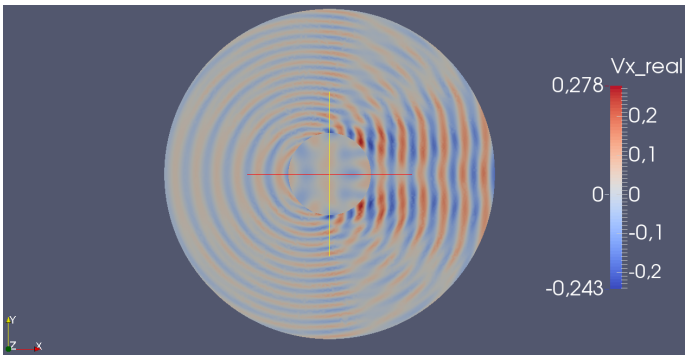
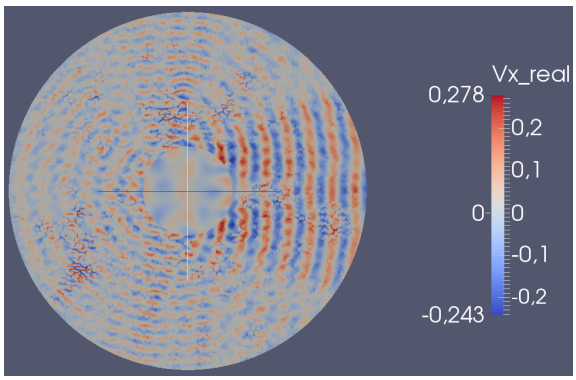
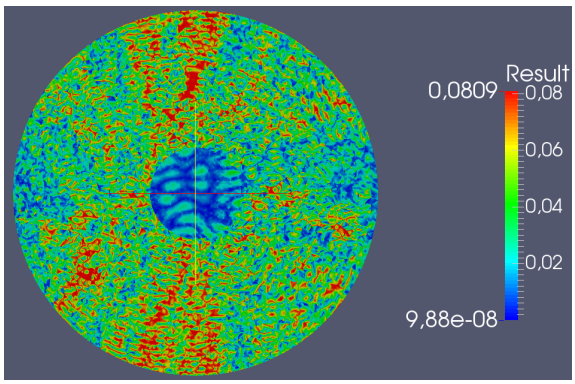


Fig. 4.4.3: Elastic solid scatterer problem: exact solution, V_x component.

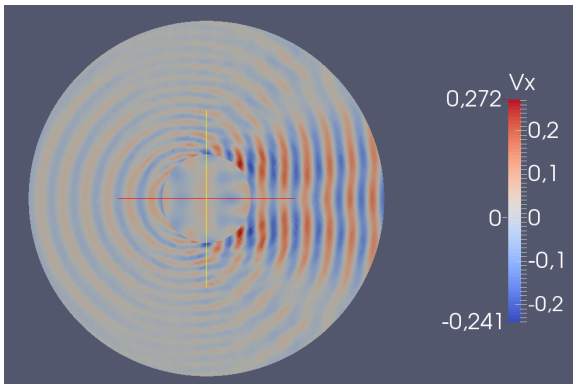


(a) Numerical solution

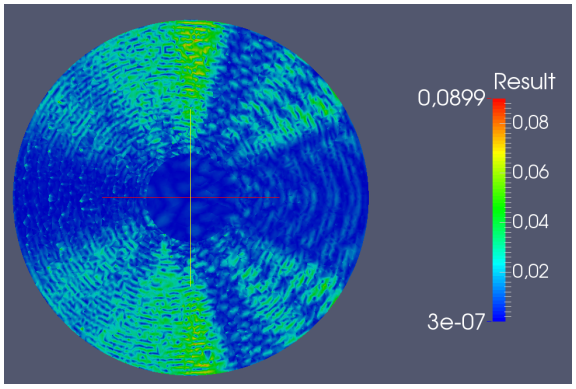


(b) Absolute error between the exact solution and the numerical solution

Fig. 4.4.4: Elastic solid scatterer problem: numerical solution, mesh M2, HDG-$\mathbb{P}_2$ scheme, V_x component.

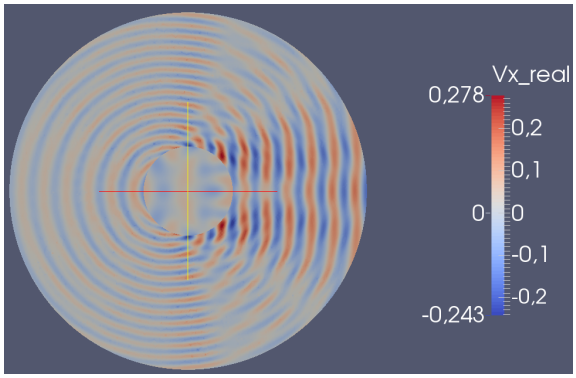


(a) Numerical solution

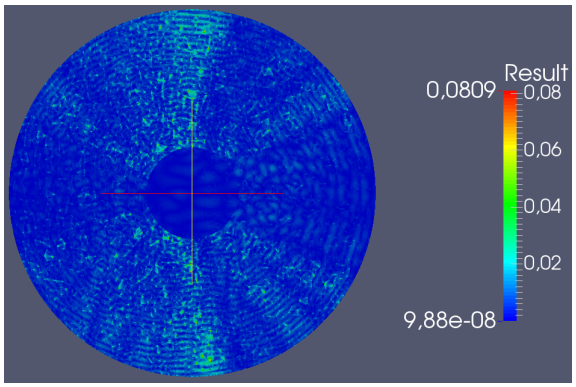


(b) Absolute error between the exact solution and the numerical solution

Fig. 4.4.5: Elastic solid scatterer problem: numerical solution, mesh M2, UDG-$\mathbb{P}_2$ scheme, V_x component.

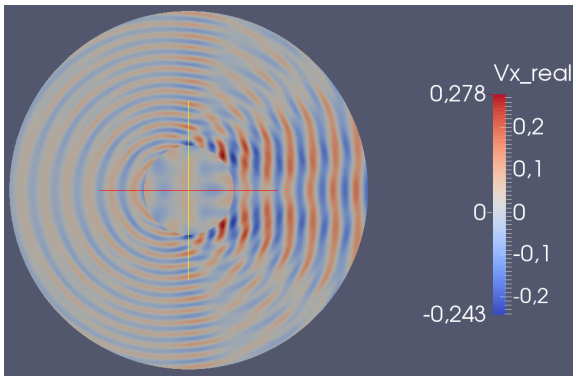


(a) Numerical solution

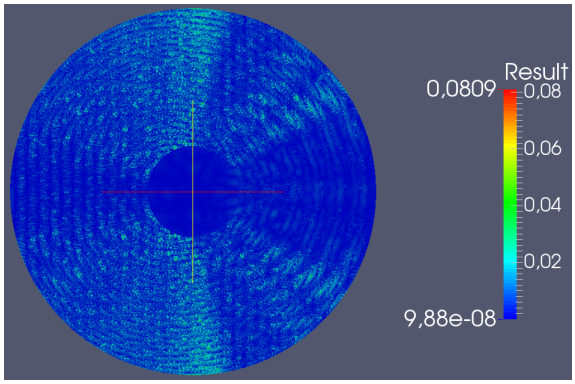


(b) Absolute error between the exact solution and the numerical solution

Fig. 4.4.6: Elastic solid scatterer problem: numerical solution, mesh M2, HDG-$\mathbb{P}_3$ scheme, V_x component.



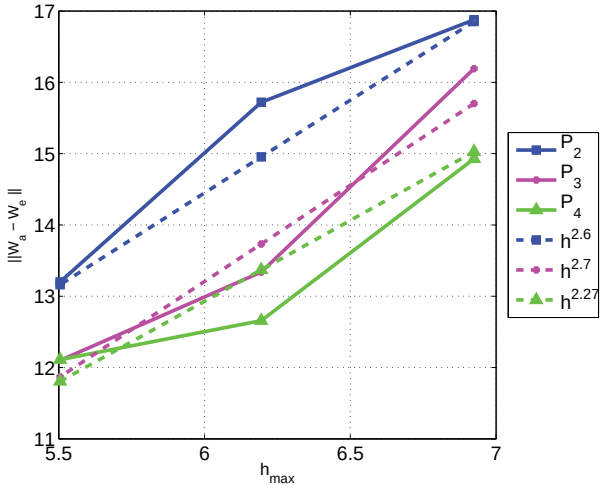
(a) Numerical solution



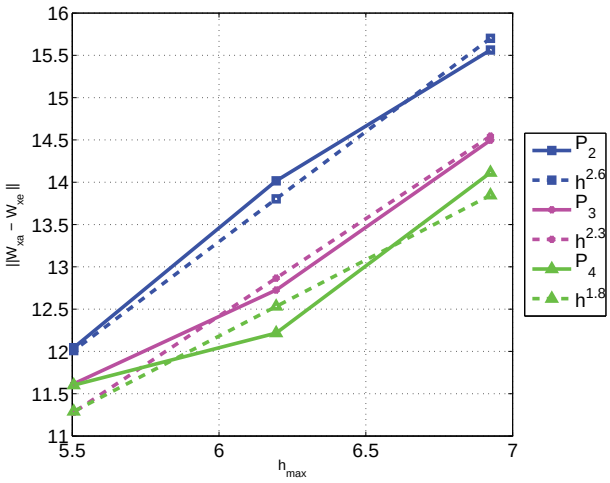
(b) Absolute error between the exact solution and the numerical solution

Fig. 4.4.7: Elastic solid scatterer problem: numerical solution, mesh M3, HDG-$\mathbb{P}_2$ scheme, V_x component.

The numerical convergence of the HDG method is shown on fig. 4.4.8a and the one of the UDG method on fig. 4.4.8b. As for the previous test case, the two numerical convergences have the same behavior. The optimal rate is not obtained. As we have remarked in the disk-shaped scattering problem, the main reason is because the geometric error dominates. The mean and relative errors on V_x and σ_{xx} components are presented on tabs. 4.4.3-4.4.4.



(a) HDG method.



(b) UDG method.

Fig. 4.4.8: Elastic solid scatterer problem: numerical convergences of the HDG and UDG methods.

The memory consumption is given on tab. 4.4.5. The computational time of the simulations are presented on tab. 4.4.6, the number of non-zero terms in the global matrix in tab. 4.4.7 and the number of degrees of freedom in tab. 4.4.8. Tab. 4.4.9 presents the comparison of both methods on a same mesh for a given error level.

We draw the same conclusions than in the previous test case: the HDG method requires less memory and computational time (see tabs. 4.4.5, 4.4.6 and 4.4.9). Moreover, as the geometric error

h (m)	Interpolation degree	Mean Error V_x		Relative Error V_x (%)	
		HDG	UDG	HDG	UDG
1016.9		2.1	0.9	273.5	120.1
490.1	2	0.2	6.4e-02	91.5	31.0
245.9		5.0e-03	1.7e-03	9.6	3.3
1016.9		1.2	0.4	134.7	46.4
490.1	3	2.0e-02	1.4e-02	9.5	6.5
245.9		1.0e-03	9.9e-04	2.0	1.9
1016.9		0.4	0.3	42.0	29.9
490.1	4	7.3e-03	7.7e-03	3.4	3.6
245.9		9.9e-04	9.9e-04	1.9	1.9

Tab. 4.4.3: Elastic solid scatterer problem: mean and relative errors on V_x.

h (m)	Interpolation degree	Mean Error σ_{xx}		Relative Error σ_{xx} (%)		Convergence order	
		HDG	UDG	HDG	UDG	HDG	UDG
1016.9		7351.8	3742.9	213.2	108.6	-	-
490.1	2	457.8	143.2	52.3	16.4	1.6	2.1
245.9		9.1	5.5	4.1	2.5	3.7	2.9
1016.9		3141.9	1054.3	83.2	27.9	-	-
490.1	3	41.7	41.6	4.7	4.7	3.9	2.4
245.9		3.8	3.9	1.7	1.7	1.8	1.6
1016.9		838.6	667.3	22.1	17.6	-	-
490.1	4	25.5	26.9	2.9	3.0	3.1	2.6
245.9		3.8	3.8	1.7	1.7	0.8	0.9

Tab. 4.4.4: Elastic solid scatterer problem: mean and relative errors on σ_{xx} and convergence order.

# Mesh elements	Interpolation degree	Memory (MB)			Memory ratio		
		HDG	UDG	IPDG	HDG	UDG	IPDG
1300		48	307	92	1	6.4	1.9
5400	2	202	1543	457	1	7.6	2.3
22000		879	7318	2211	1	8.3	2.5
1300		77	604	236	1	7.8	3.1
5400	3	351	3233	1246	1	9.2	3.5
22000		1553	15741	6104	1	10.1	3.9
1300		111	988	529	1	8.9	4.8
5400	4	526	5160	2789	1	9.8	5.3
22000		2362	23982	13362	1	10.2	5.7

Tab. 4.4.5: Elastic solid scatterer problem: memory used.

# Mesh elements	Interpolation degree	Construction time(s)			Resolution time (s)		
		HDG	UDG	IPDG	HDG	UDG	IPDG
1300	2	0.2	4.7e-02	1.6	0.2	2.3	0.6
5400		0.9	0.2	6.5	1.1	14.4	4.5
22000		3.5	0.8	25.0	6.6	98.2	34.1
1300	3	0.6	0.1	4.2	0.3	5.1	2.1
5400		2.7	0.5	17.6	2.0	36.7	18.4
22000		11.1	1.9	69.3	13.7	348.1	143.4
1300	4	1.7	0.2	9.2	0.5	9.2	6.3
5400		7.2	1.0	38.6	3.4	69.6	58.1
22000		30.1	3.8	153.3	23.8	525.4	473.2

Tab. 4.4.6: Elastic solid scatterer problem: time required for the global matrix construction and for the system resolution.

# Mesh elements	Interpolation degree	Non-zeros terms		
		HDG	UDG	IPDG
1300	2	4.1e+05	1.7e+06	7.2e+05
5400		1.8e+06	7.5e+06	3.1e+06
22000		7.0e+06	3.0e+07	1.2e+07
1300	3	7.3e+05	3.8e+06	2.0e+06
5400		3.1e+06	1.6e+07	8.6e+06
22000		1.2e+07	6.6e+07	3.5e+07
1300	4	1.1e+06	7.3e+06	4.5e+06
5400		4.9e+06	3.1e+07	1.9e+07
22000		1.9e+07	1.2e+08	7.8e+07

Tab. 4.4.7: Elastic solid scatterer problem: number of non-zero terms in the global matrix .

# Mesh elements	Interpolation degree	# dof			#dof/wavelength		
		HDG	UDG	IPDG	HDG	UDG	IPDG
1300	2	1.1e+04	3.8e+04	1.5e+04	6	19	8
5400		4.9e+04	1.6e+05	6.5e+04	24	81	32
22000		2.0e+05	6.5e+05	2.6e+05	98	325	130
1300	3	1.6e+04	6.3e+04	2.5e+04	8	31	13
5400		6.5e+04	2.7e+05	1.1e+05	32	135	54
22000		2.6e+05	1.1e+06	4.3e+05	130	541	216
1300	4	1.9e+04	9.5e+04	3.8e+04	10	47	19
5400		8.2e+04	4.1e+05	1.6e+05	41	202	81
22000		3.3e+05	1.6e+06	6.5e+05	163	812	325

Tab. 4.4.8: Elastic solid scatterer problem: total number degrees of freedom (ndof) and ndof by wavelength (λ_w) for the three methods.

Level of Error	# Mesh elements	Interpolation degree		Memory (MB)		Construction time (s)		Solution time (s)	
		HDG	UDG	HDG	UDG	HDG	UDG	HDG	UDG
1e-02	5400	3	2	351	1543	2.7	0.2	2.0	14.4
1e-03	22000	3	2	1553	7318	11.1	0.8	13.7	98.2

Tab. 4.4.9: Elastic solid scatterer problem: for a same level of error, comparison between computational time and memory required with a same mesh for both methods.

dominates due to the fact that we approximate a circular domain by triangles, the study of the numerical error and of the accuracy of the methods is difficult to realize. It is the reason why we do not study the influence of the parameter τ in this test case.

4.5 Geophysic test case: the Marmousi model

Finally, for testing the 2D isotropic HDG method in a high performance computing (HPC) framework, we consider a geophysic benchmark: the Marmousi model. As the UDG method is the most expensive method between the HDG, IPDG and UDG methods, we only present the HDG and IPDG performances. The v_p velocity model is represented on fig. 4.5.1. The mesh of the domain, which is composed of 235 000 elements, is shown on fig. 4.5.2.

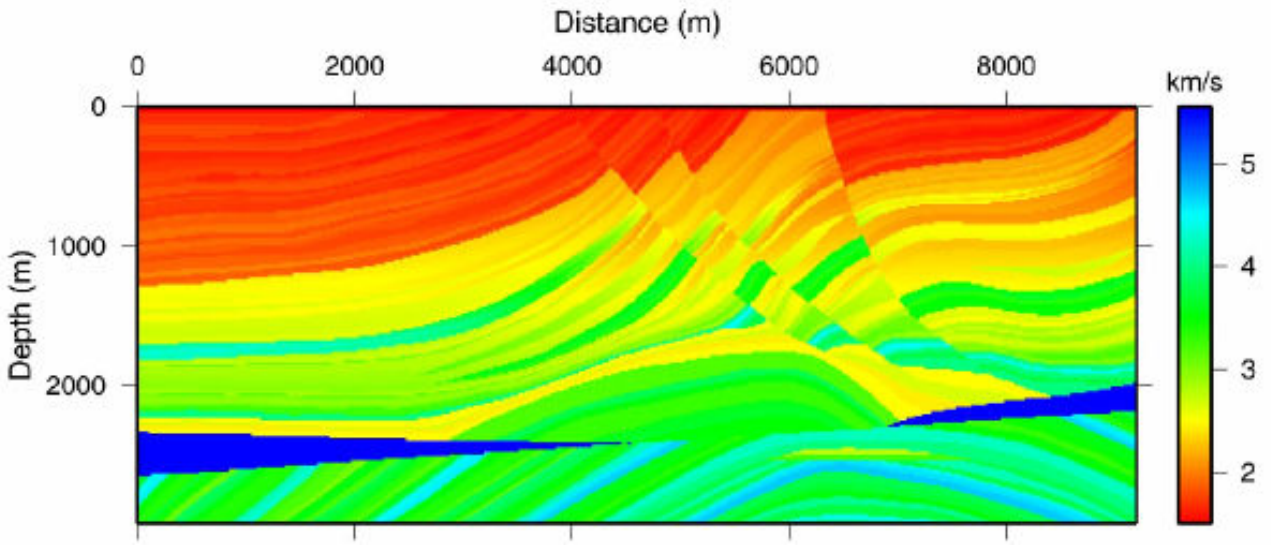


Fig. 4.5.1: Marmousi velocity card and topography.

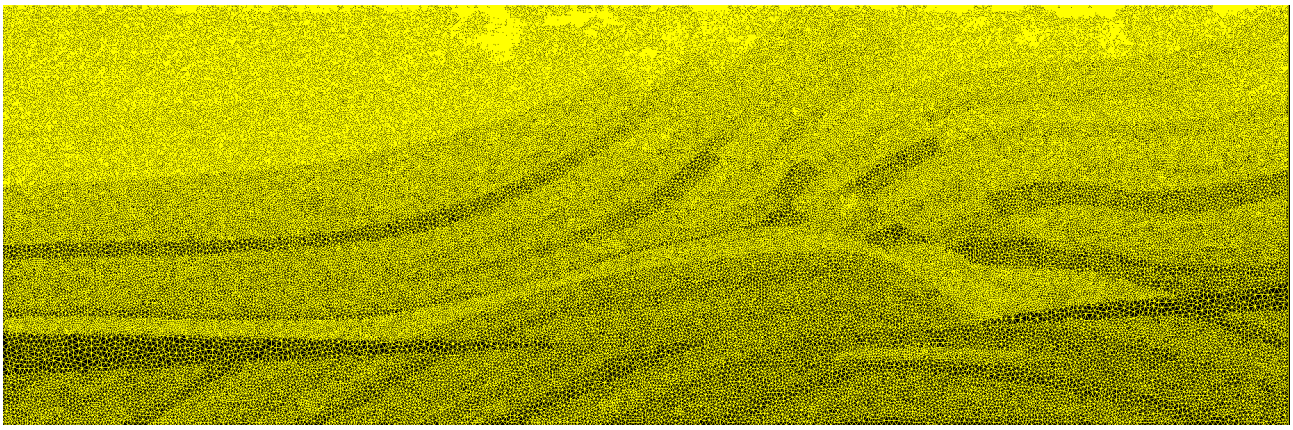
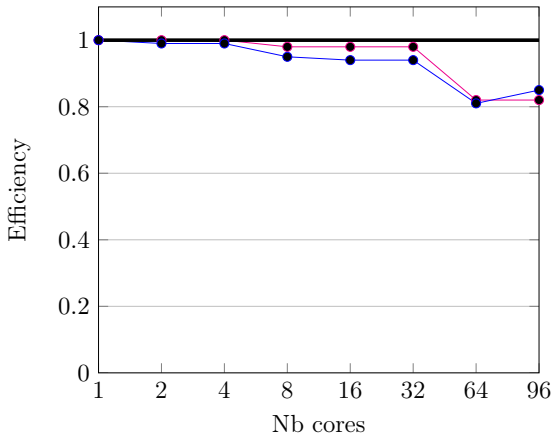


Fig. 4.5.2: Marmousi discretization.

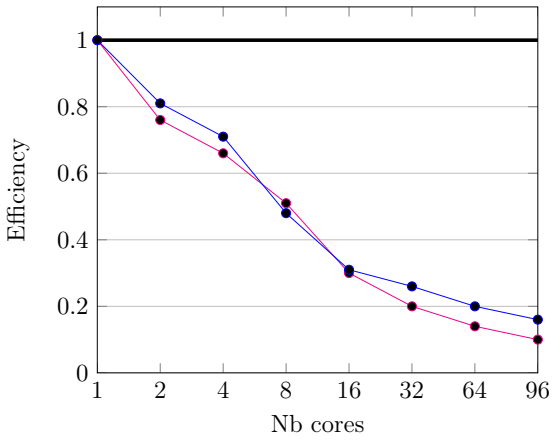
The characteristics of the computer system that we used for the simulations are:

- 2 Hexa-core Westmere Intel® Xeon® X5670
- Frequency : 2,93 GHz
- Cache L3 : 12 Mb
- RAM : 96 Gb
- Infiniband DDR : 20Gb/s
- Ethernet : 1Gb/s

We compare the efficiency of both methods by dividing the sequential computational time by the parallel computational time. First, we compare the efficiency of the construction of the global system on fig. 4.5.3a. We observe that the efficiency of the construction of the HDG global matrix is close to the one of the IPDG matrix. Then, when we compare the efficiency of the resolution part of the algorithm on fig. 4.5.3b, we observe that the efficiency of both methods is also close and collapse. This collapse is essentially due to the use of a direct solver which exhibits a poor scalability.



(a) construction of the global matrix and right-hand side.



(b) Resolution.

Fig. 4.5.3: Efficiency of the parallelism of the HDG method (red) and of the IPDG method (blue).

Figure 4.5.4 presents the memory required by the direct solver for the inversion and the resolution process. For both methods, the memory increases with the number of cores, this is intrinsic to the LU solver. We note that using the HDG method divides the memory consumption by a factor 3 to 4. Hence, solving the Marmousi problem with the IPDG method requires around 80 GB while the HDG method only requires 20 GB.

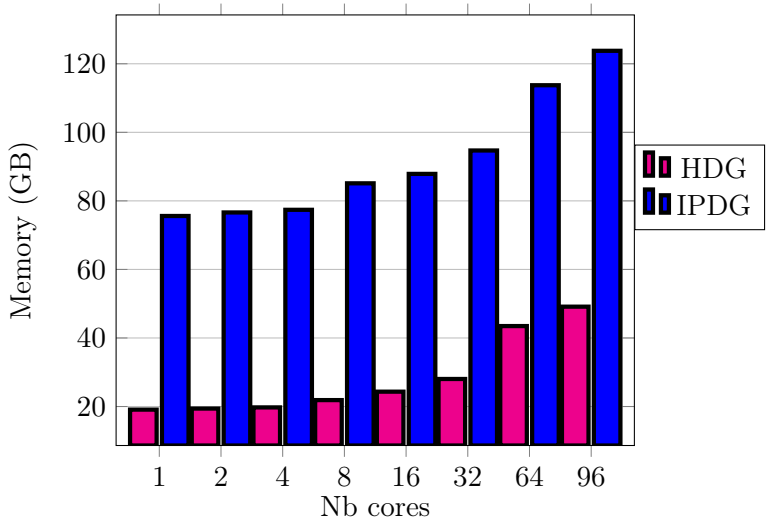
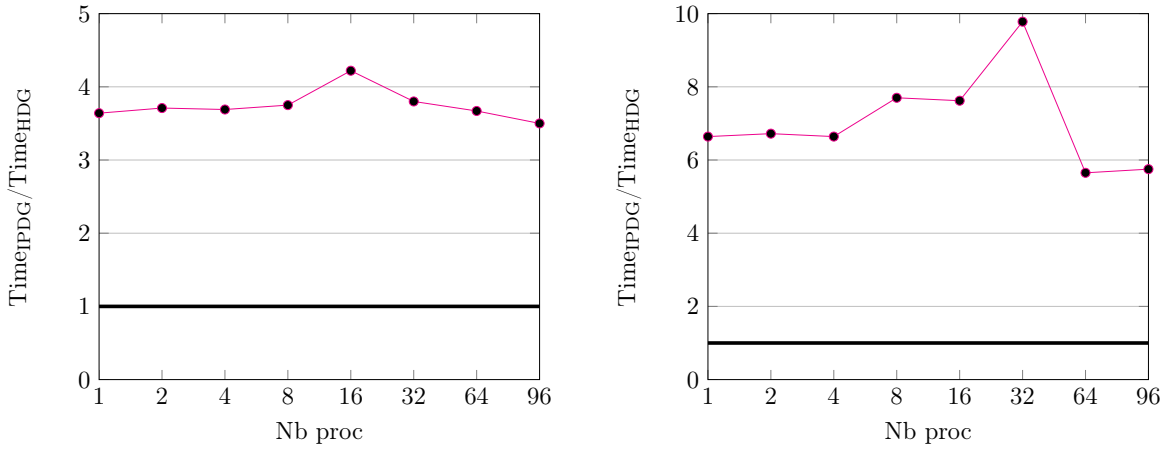


Fig. 4.5.4: Memory consumption (GB) for the resolution.

The speed up is presented on figs. 4.5.5a and 4.5.5b. In fig. 4.5.5a we compare the time that we need for the construction of the global matrix. It is relatively stable when we compare it to the number of cores used. The construction of the HDG global matrix is performed about four times faster in terms of CPU time than the one of the IPDG matrix. For the full simulation (fig. 4.5.5b), the HDG algorithm is performed around seven times faster.



(a) Global matrix construction.

(b) Total simulation.

Fig. 4.5.5: Speed up.

4.6 Anisotropic test-case

Finally, we present results that we have obtained with the HDG method when considering an anisotropic problem. We consider the domain shown on fig. 4.6.1 composed of six layers: a zone of water, a layer of water sand, a zone of shale, a layer of sandstone, a layer of salt and another layer of water sand. The physical characteristics of the six media are summarized in tab. 4.6.1. The computational domain is reduced to a 3000 m $\times$ 3000 m square. On the upper boundary, we put the free surface condition (3.1.4); on the other boundaries, we impose the absorbing condition (3.1.5).
The computational domain is discretized into three unstructured meshes with respectively 600, 3000 and 28000 elements. Their characteristics are given in tab. 4.6.2.
For this test case, as we do not have access to an analytical solution, therefore we compare the HDG results with the one obtained with the interior penalty DG (IPDG) method (see fig. 4.6.3). The HDG solution was computed with different values of τ, shown on figs. 4.6.4a to 4.6.4f. This anisotropic example allows to demonstrate the importance of the value of the parameter τ. Indeed the HDG solution closer to the IPDG solution is the one computed with $\tau = \lambda + 2\mu$ (fig. 4.6.4f). With the other solutions (figs. 4.6.4a to 4.6.4e), we distinguish 4 different regions, but the wave propagation is not correct when we compare to the IPDG propagation. The solution on figs. 4.6.4a to 4.6.4e are not accurate enough.
It is interesting to take a look on the computational performances of the two algorithms (tab. 4.6.4). The HDG formulation is more competitive in terms of memory and computational time than the IPDG method: we need 3 times less memory and between 3 and 9 times less computational time. This depends on the mesh refinement.
Another remark on this test case is the presence of pollution in the region where the source wave is localized. This is due to the fact that the first layer, which is a layer of water, so an acoustic medium, is approximated with a pseudo-acoustic medium: we have taken an elastic medium with $v_s = 0$.
Finally we also use this test case to compare the computational performances in tab. 4.6.3 of the program when it performs in an isotropic framework and anisotropic context. We clearly see that taking into account the anisotropy does not increase the computational cost. This is due to the fact that, in the HDG formulation, the anisotropic property affects only two local matrices and more specifically affects only the value of some coefficients which are not equal zero in the isotropic case.

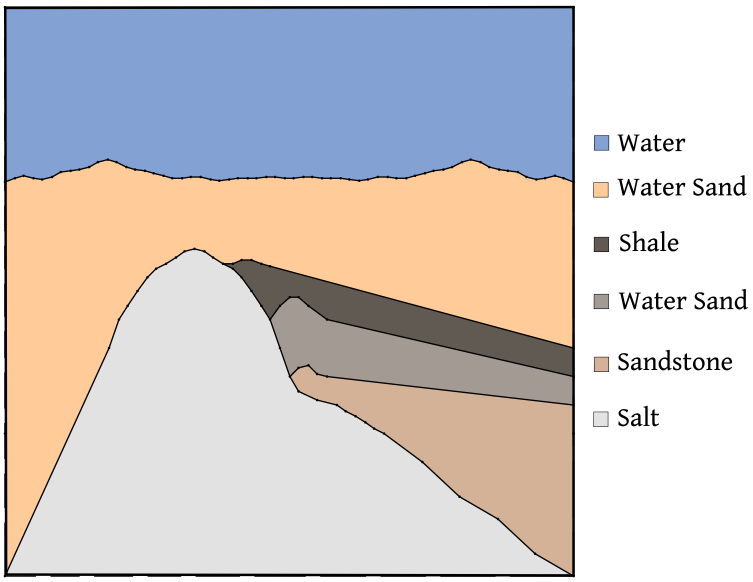


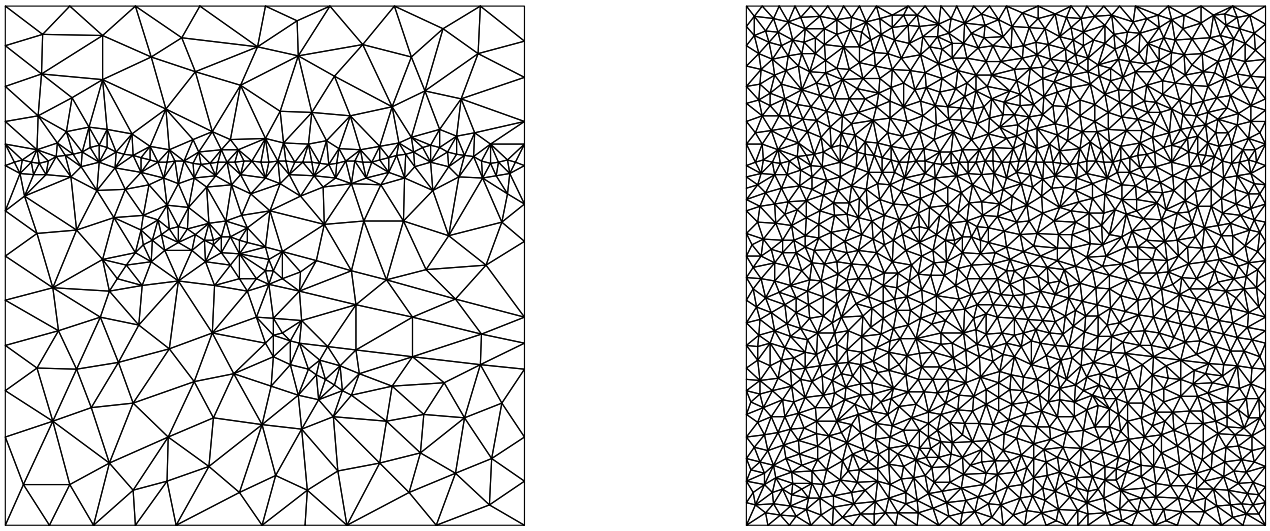
Fig. 4.6.1: Anisotropic test case.

	Water	Water Sand 1	Shale	Water Sand 2	Sanstone	Salt
v_p	1500	1409	4359	1609	4633	5334
v_s	0	480	3048	780	3231	3353
ρ	1000	2030	2810	2030	2710	2710
ϵ	0	0.022	0.172	0.022	-0.026	0.369
δ	0	0.018	0	0.018	-0.033	0.579
θ	0	0	10	15	25	20

Tab. 4.6.1: Characteristics of the anisotropic media.

Mesh	# Mesh elements	# Mesh vertices	h_{min} (m)	h_{max} (m)	h_{max}/h_{min}
M1	600	330	30	512	17.07
M2	3 000	1 500	30	187.5	6.25
M3	28 000	14 180	13.6	64.9	4.77

Tab. 4.6.2: Characteristics of the three meshes.



(a) Mesh M1, 600 elements

(b) Mesh M2, 3 000 elements

Fig. 4.6.2: Triangulations of the domain.

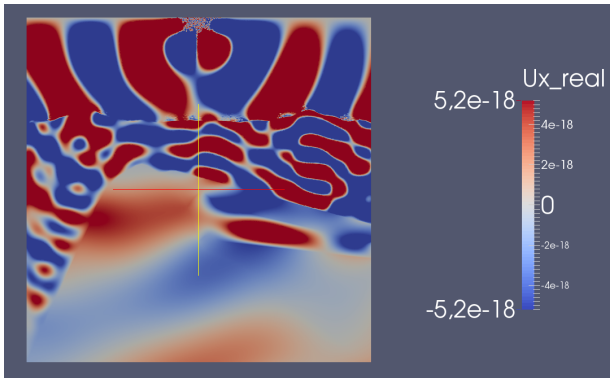
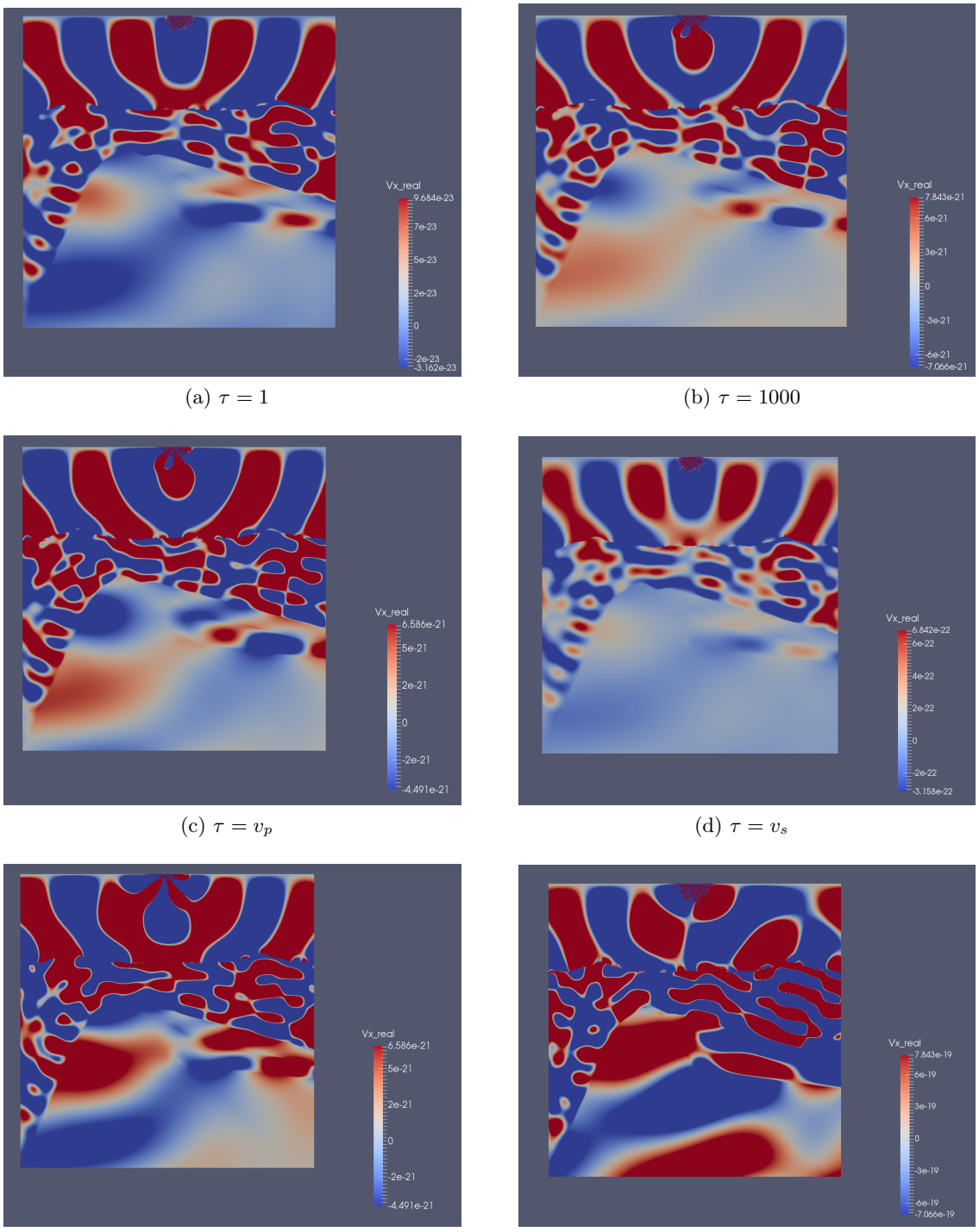


Fig. 4.6.3: IPDG-$\mathbb{P}_3$ numerical solution, computed on the mesh M3.



(a) $\tau = 1$ (b) $\tau = 1000$

(c) $\tau = v_p$ (d) $\tau = v_s$

(e) $\tau = \mu$ (f) $\tau = \lambda + 2\mu$

Fig. 4.6.4: HDG-$\mathbb{P}_3$ numerical solutions for different values of τ, mesh M3.

Conclusion

Thanks to this numerical study, we have seen that for a same computation (same mesh and same interpolation order), the HDG method is more competitive in memory and computational time terms

Mesh	Memory (MB)		Construction time (s)		Solution time (s)	
	Isotropic	Anisotropic	Isotropic	Anisotropic	Isotropic	Anisotropic
M1	38	38	0.5	0.5	0.2	0.2
M2	168	168	2.5	2.5	1.6	1.6
M3	1971	1971	22.3	22.2	41.6	40.2

Tab. 4.6.3: Comparison between isotropic and anisotropic HDG-$\mathbb{P}_3$ performances.

Mesh	Memory (MB)		Construction time (s)		Solution time (s)	
	HDG	IPDG	HDG	IPDG	HDG	IPDG
M1	38	98	0.5	2.8	0.2	1.0
M2	168	558	2.5	12.6	1.6	12.1
M3	1971	7496	22.3	128.0	40.2	440.0

Tab. 4.6.4: Comparison between HDG-$\mathbb{P}_3$ and IPDG-$\mathbb{P}_3$ performances.

than classical DG methods. Moreover performing the HDG scheme in an anisotropic formulation does not involve significant additional computational cost as compared to the isotropic formulation.
Following the value of the stabilization parameter τ, the solution can be more or less accurate. We thus have three choices to obtain a better solution with the HDG formulation. First, we can change the value of the parameter τ, the optimal value seems to be close to v_p^2 as demonstrated in the anisotropic problem. This solution does not increase the required memory. The second option is to increase the interpolation order. We have seen that, even if we use the HDG scheme with one or two interpolation order more than classical DG methods, it still needs less memory. The last option for a better accuracy is to refine the mesh, and we have seen too that even with finer meshes the HDG method is still competitive in memory terms.
Finally, the parallel implementation of the HDG algorithm performs well.

All these conclusions have to be confirmed for the 3D elastic Helmholtz equations.

Chapter 5

Hybridizable Discontinuous Galerkin (HDG) method for 3D elastic Helmholtz equations

In this last chapter, we extend the development of the HDG method to the 3D elastic wave equations. We recall the 3D anisotropic elastodynamics in the first section and we introduce notations specific to DG methods in the next section. Then, in the third part, we derive the HDG formulation that we discretize in section 4. Finally in the last section, we analyze the performances of the 3D HDG method and we compare it to the performances of the IPDG method on a simple test case, the propagation of a plane wave in an homogeneous medium, and on a more realistic benchmark, the Epati test case.

5.1 3D elastic wave equations in harmonic domain: problem statement

We recall the first order formulation of the 3D elastodynamics problem in harmonic domain: for $\mathbf{x} = (x, y, z) \in \Omega \subset \mathbb{R}^3$

$$\left\{ \begin{array}{rcll} i\omega\rho\mathbf{v} & = & \nabla \cdot \underline{\underline{\sigma}}, & \text{in } \Omega, \\ i\omega\underline{\underline{\sigma}} & = & \underline{\underline{C}}\ \underline{\underline{\epsilon}}(\mathbf{v}) + \mathbf{f}, & \text{in } \Omega. \end{array} \right. \tag{5.1.1}$$

As we have in previous chapters, we omit the space dependency in the mass density ρ, the velocity vector $\mathbf{v} = (v_x,\ v_y,\ v_z)^T$, the strain tensor $\underline{\underline{\epsilon}}$, the stress tensor $\underline{\underline{\sigma}}$, the elasticity tensor $\underline{\underline{C}}$, and the source term $\mathbf{f}$.
In 3D, the elasticity tensor $\underline{\underline{C}}$ is a fourth order $3 \times 3 \times 3 \times 3$ symetric tensor, that we express using Voigt's notations, as a 6×6 matrix. In the general case, we have

$$\underline{\underline{C}} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46} \\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56} \\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} \end{pmatrix}.$$

Remark. *We remind that the 3D Voigt's notation writes*

$$(xx,\ yy,\ zz,\ yz,\ xz,\ xy)^T = (1,\ 2,\ 3,\ 4,\ 5,\ 6)^T.$$

The developed anistropic equations of system (5.1.1) reads as: for $(x, y, z) \in \Omega \subset \mathbb{R}^3$

$$
\begin{cases}
i\omega v_x = \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xx}}{\partial x} + \dfrac{\partial \sigma_{xy}}{\partial y} + \dfrac{\partial \sigma_{xz}}{\partial z}\right), \\
i\omega v_y = \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xy}}{\partial x} + \dfrac{\partial \sigma_{yy}}{\partial y} + \dfrac{\partial \sigma_{yz}}{\partial z}\right), \\
i\omega v_z = \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xz}}{\partial x} + \dfrac{\partial \sigma_{yz}}{\partial y} + \dfrac{\partial \sigma_{zz}}{\partial z}\right), \\
i\omega \sigma_{xx} = C_{11}\dfrac{\partial v_x}{\partial x} + C_{12}\dfrac{\partial v_y}{\partial y} + C_{13}\dfrac{\partial v_z}{\partial z} \\
\qquad + C_{14}\left(\dfrac{\partial v_z}{\partial y} + \dfrac{\partial v_y}{\partial z}\right) + C_{15}\left(\dfrac{\partial v_x}{\partial z} + \dfrac{\partial v_z}{\partial x}\right) + C_{16}\left(\dfrac{\partial v_x}{\partial y} + \dfrac{\partial v_y}{\partial x}\right) + f_x, \\
i\omega \sigma_{yy} = C_{12}\dfrac{\partial v_x}{\partial x} + C_{22}\dfrac{\partial v_y}{\partial y} + C_{23}\dfrac{\partial v_z}{\partial z} \\
\qquad + C_{24}\left(\dfrac{\partial v_z}{\partial y} + \dfrac{\partial v_y}{\partial z}\right) + C_{25}\left(\dfrac{\partial v_x}{\partial z} + \dfrac{\partial v_z}{\partial x}\right) + C_{26}\left(\dfrac{\partial v_x}{\partial y} + \dfrac{\partial v_y}{\partial x}\right) + f_y, \\
i\omega \sigma_{zz} = C_{13}\dfrac{\partial v_x}{\partial x} + C_{23}\dfrac{\partial v_y}{\partial y} + C_{33}\dfrac{\partial v_z}{\partial z} \\
\qquad + C_{34}\left(\dfrac{\partial v_z}{\partial y} + \dfrac{\partial v_y}{\partial z}\right) + C_{35}\left(\dfrac{\partial v_x}{\partial z} + \dfrac{\partial v_z}{\partial x}\right) + C_{36}\left(\dfrac{\partial v_x}{\partial y} + \dfrac{\partial v_y}{\partial x}\right) + f_z, \\
i\omega \sigma_{xy} = C_{16}\dfrac{\partial v_x}{\partial x} + C_{26}\dfrac{\partial v_y}{\partial y} + C_{36}\dfrac{\partial v_z}{\partial z} \\
\qquad + C_{46}\left(\dfrac{\partial v_z}{\partial y} + \dfrac{\partial v_y}{\partial z}\right) + C_{56}\left(\dfrac{\partial v_x}{\partial z} + \dfrac{\partial v_z}{\partial x}\right) + C_{66}\left(\dfrac{\partial v_x}{\partial y} + \dfrac{\partial v_y}{\partial x}\right), \\
i\omega \sigma_{xz} = C_{15}\dfrac{\partial v_x}{\partial x} + C_{25}\dfrac{\partial v_y}{\partial y} + C_{35}\dfrac{\partial v_z}{\partial z} \\
\qquad + C_{45}\left(\dfrac{\partial v_z}{\partial y} + \dfrac{\partial v_y}{\partial z}\right) + C_{55}\left(\dfrac{\partial v_x}{\partial z} + \dfrac{\partial v_z}{\partial x}\right) + C_{56}\left(\dfrac{\partial v_x}{\partial y} + \dfrac{\partial v_y}{\partial x}\right), \\
i\omega \sigma_{yz} = C_{14}\dfrac{\partial v_x}{\partial x} + C_{24}\dfrac{\partial v_y}{\partial y} + C_{34}\dfrac{\partial v_z}{\partial z} \\
\qquad + C_{44}\left(\dfrac{\partial v_z}{\partial y} + \dfrac{\partial v_y}{\partial z}\right) + C_{45}\left(\dfrac{\partial v_x}{\partial z} + \dfrac{\partial v_z}{\partial x}\right) + C_{46}\left(\dfrac{\partial v_x}{\partial y} + \dfrac{\partial v_y}{\partial x}\right).
\end{cases}
\tag{5.1.2}
$$

Remark. *In the isotropic case, the 3D tensor* $\underline{\underline{C}}$ *is expressed as*

$$
\underline{\underline{C}} = \begin{pmatrix}
\lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\
\lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\
\lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\
0 & 0 & 0 & \mu & 0 & 0 \\
0 & 0 & 0 & 0 & \mu & 0 \\
0 & 0 & 0 & 0 & 0 & \mu
\end{pmatrix}.
$$

And thus the developed equations of the isotropic system are

$$\begin{cases} i\omega v_x = \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xx}}{\partial x} + \dfrac{\partial \sigma_{xy}}{\partial y} + \dfrac{\partial \sigma_{xz}}{\partial z}\right), \\ i\omega v_y = \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xy}}{\partial x} + \dfrac{\partial \sigma_{yy}}{\partial y} + \dfrac{\partial \sigma_{yz}}{\partial z}\right), \\ i\omega v_z = \dfrac{1}{\rho}\left(\dfrac{\partial \sigma_{xz}}{\partial x} + \dfrac{\partial \sigma_{yz}}{\partial y} + \dfrac{\partial \sigma_{zz}}{\partial z}\right), \\ i\omega \sigma_{xx} = (\lambda + 2\mu)\dfrac{\partial v_x}{\partial x} + \lambda\left(\dfrac{\partial v_y}{\partial y} + \dfrac{\partial v_z}{\partial z}\right) + f_x, \\ i\omega \sigma_{yy} = (\lambda + 2\mu)\dfrac{\partial v_y}{\partial y} + \lambda\left(\dfrac{\partial v_x}{\partial x} + \dfrac{\partial v_z}{\partial z}\right) + f_y, \\ i\omega \sigma_{zz} = \lambda\left(\dfrac{\partial v_x}{\partial x}\partial x + \dfrac{\partial v_y}{\partial y}\right) + (\lambda + 2\mu)\dfrac{\partial v_z}{\partial z} + f_z, \\ i\omega \sigma_{xy} = \mu\left(\dfrac{\partial v_x}{\partial y} + \dfrac{\partial v_y}{\partial x}\right), \\ i\omega \sigma_{xz} = \mu\left(\dfrac{\partial v_x}{\partial z} + \dfrac{\partial v_z}{\partial x}\right), \\ i\omega \sigma_{yz} = \mu\left(\dfrac{\partial v_y}{\partial z} + \dfrac{\partial v_z}{\partial y}\right). \end{cases} \tag{5.1.3}$$

To complete the description of the numerical scheme, the boundary conditions are given by

- The free surface condition

$$\underline{\underline{\sigma}} \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_l, \tag{5.1.4}$$

- The absorbing boundary condition

$$\underline{\underline{\sigma}} \cdot \mathbf{n} - PA(\theta', \phi')P^T\mathbf{v} = 0 \quad \text{on } \Gamma_a, \tag{5.1.5}$$

where $\Gamma_l \cup \Gamma_a = \Gamma = \partial\Omega$ and $\mathbf{n}$ is the outward unit norm vector. We remind the expressions of P, A, θ', and ϕ' which are described in section 1.1.5

- P is defined such as

  - if $|n_z| \neq 1$, $P = \begin{pmatrix} n_x & -\frac{n_y}{\sqrt{n_x^2+n_y^2}} & -\frac{n_x n_z}{\sqrt{n_x^2+n_y^2}} \\ n_y & \frac{n_x}{\sqrt{n_x^2+n_y^2}} & -\frac{n_y n_z}{\sqrt{n_x^2+n_y^2}} \\ n_z & 0 & \sqrt{n_x^2+n_y^2} \end{pmatrix}$,
  - if $|n_z| = 1$, $P = \begin{pmatrix} \cos(n_z\frac{\pi}{2}) = 0 & 0 & -\sin(n_z\frac{\pi}{2}) \\ 0 & 1 & 0 \\ \sin(n_z\frac{\pi}{2}) & 0 & \cos(n_z\frac{\pi}{2}) = 0 \end{pmatrix}$.

- $A(\theta, \phi) = \begin{pmatrix} A_1(\theta,\phi) \\ A_2(\theta,\phi) \\ A_3(\theta,\phi) \end{pmatrix}$, with

$$A_1(\theta,\phi) = \varsigma(\theta,\phi)\left(\kappa\sin^2\phi + \cos^2\phi(\kappa\cos^2\theta + \sin^2\theta)\right)\begin{pmatrix} \kappa\sin^2\phi + \cos^2\phi(\kappa\cos^2\theta + \sin^2\theta) \\ -(\kappa-1)\cos\phi\sin\phi\sin^2\theta \\ -(\kappa-1)\cos\theta\sin\theta\cos\phi \end{pmatrix}^T,$$

$$A_2(\theta,\phi) = \varsigma(\theta,\phi)\left((\kappa-1)\cos\phi\sin\phi\sin^2\theta\right)\begin{pmatrix}-\kappa\sin^2\phi+\cos^2\phi(\kappa\cos^2\theta+\sin^2\theta)\\(\kappa-1)\cos\phi\sin\phi\sin^2\theta\\(\kappa-1)\cos\theta\sin\theta\cos\phi\end{pmatrix}^T - \rho v_s\begin{pmatrix}0\\1\\0\end{pmatrix}^T,$$

$$A_3(\theta,\phi) = \varsigma(\theta,\phi)\left((\kappa-1)\cos\phi\cos\theta\sin\theta\right)\begin{pmatrix}-\kappa\sin^2\phi+\cos^2\phi(\kappa\cos^2\theta+\sin^2\theta)\\(\kappa-1)\cos\phi\sin\phi\sin^2\theta\\(\kappa-1)\cos\theta\sin\theta\cos\phi\end{pmatrix}^T - \rho v_s\begin{pmatrix}0\\0\\1\end{pmatrix}^T,$$

where $\kappa = \sqrt{1-2\varepsilon_T}$ and $\varsigma(\theta,\phi) = \dfrac{-\rho v_p}{\sqrt{\kappa^2\cos^2\theta\cos^2\phi+\kappa^2\sin^2\theta+\sin^2\theta\cos^2\phi}}$.

- θ' is defined by

$$\begin{cases}\cos\theta' = \left(\dfrac{-n_x n_z}{\sqrt{n_x^2+n_y^2}}\cos\phi - \dfrac{n_y n_z}{\sqrt{n_x^2+n_y^2}}\sin\phi\right)\sin\theta + \sqrt{n_x^2+n_y^2}\cos\theta & \text{if } |n_z|\neq 1,\\ \cos\theta' = -n_z\cos\phi\sin\theta & \text{if } |n_z| = 1,\end{cases}$$

  and $\sin\theta' = \sqrt{1-\cos^2\theta}$.

- ϕ' is defined such as

  – if $\sin\theta' \neq 0$,

$$\text{if } |n_z|\neq 1,\quad \begin{cases}\cos\phi' = (n_x\cos\phi + n_y\sin\phi)\dfrac{\sin\theta}{\sin\theta'} + n_z\dfrac{\cos\theta}{\sin\theta'},\\ \sin\phi' = \left(\dfrac{-n_y\cos\phi}{\sqrt{n_x^2+n_y^2}} - \dfrac{n_x\sin\phi}{\sqrt{n_x^2+n_y^2}}\right)\dfrac{\sin\theta}{\sin\theta'}.\end{cases}$$

$$\text{if } |n_z| = 1,\quad \begin{cases}\cos\phi' = n_z\dfrac{\cos\theta}{\sin\theta'},\\ \sin\phi' = \dfrac{\sin\theta\sin\phi}{\sin\theta'}.\end{cases}$$

  – if $\sin\theta' = 0$,

$$\cos\phi' = 1 \text{ and } \sin\phi' = 0.$$

Remark. *In the isotropic case, $\kappa = 1$, $\theta = 0$ and $\phi = 0$ so that $\cos\phi' = 1$, $\sin\phi' = 0$, $\cos\theta' = 1$ and $\sin\theta' = 0$. Matrix $A(\theta,\phi)$ becomes*

$$A(\theta,\phi) = \begin{pmatrix}-\rho v_p & 0 & 0\\ 0 & -\rho v_s & 0\\ 0 & 0 & -\rho v_s\end{pmatrix}.$$

The matrix $PA(\theta',\phi')P^T$ is given by

$$\begin{pmatrix}-\rho v_p n_x^2 - \rho v_s(n_x^2 n_z^2 + n_y^2)\dfrac{1}{n_x^2+n_y^2} & -\rho v_p n_x n_y + \rho v_s\frac{n_x n_y}{n_x^2+n_y^2}(1-n_z^2) & -\rho v_p n_x n_z + \rho v_s n_x n_z\\ -\rho v_p n_x n_y + \rho v_s\dfrac{n_x n_y}{n_x^2+n_y^2}(1-n_z^2) & -\rho v_p n_y^2 + \rho v_s(n_x^2+n_z^2 n_y^2)\frac{1}{n_x^2+n_y^2} & -\rho v_p n_y n_z + \rho v_s n_y n_z\\ -\rho v_p n_x n_z + \rho v_s n_x n_z & -\rho v_p n_y n_z + \rho v_s n_y n_z & -\rho v_p n_z^2 - \rho v_s(n_x^2+n_y^2)\end{pmatrix}.$$

5.2 Notations

In 3D, the mesh $\mathcal{T}_h$ of the computational domain Ω is composed of tetrahedra K. We use similar notations to the 2D case, i.e

- F is a face of K,
- $\mathcal{F}(K)$ is the set of faces of an element K,
- $\mathcal{F}_b$ is the set of the boundary faces F_b, i.e. $F_b = \partial K \cap \Gamma$, where $\Gamma = \partial\Omega$,
- $\mathcal{F}_i$ is the set of interior faces F_i i.e. $F_i = \partial K \cap \partial K'$ where K and K' are neighbours,
- $\mathcal{F}_h$ is the set of all the faces of $\mathcal{T}_h$, i.e. $\mathcal{F}_h = \mathcal{F}_i \cup \mathcal{F}_b$,
- $\mathbf{n}$ is the outward normal vector to K, $\mathbf{t}$ an arbitrary tangent vector to K.

$P_p(D)$ denotes the set of polynomials of degree at most p on the domain D. For each element $K \in \mathcal{T}_h$, $V^p(K)$ is the space $P_p(K)$, $\mathbf{V}^p(K)$ is the space $(P_p(K))^3$ and $\mathbf{\Sigma}^p(K)$ is the space $(P_p(K))^6$. The approximation spaces in 3D are defined by

$$
\begin{aligned}
V_h^p &= \{v \in L^2(\Omega) : \quad v|_K \in V^p(K), \forall K \in \mathcal{T}_h\},\\
\mathbf{V}_h^p &= \{\mathbf{v} \in \left(L^2(\Omega)\right)^3 : \quad \mathbf{v}|_K \in \mathbf{V}^p(K), \forall K \in \mathcal{T}_h\},\\
\mathbf{\Sigma}_h^p &= \{\underline{\underline{\sigma}} \in \left(L^2(\Omega)\right)^6 : \quad \underline{\underline{\sigma}}|_K \in \mathbf{\Sigma}^p(K), \forall K \in \mathcal{T}_h\},\\
\mathbf{M}_h &= \{\eta \in \left(L^2(\mathcal{F}_h)\right)^3 : \quad \eta|_F \in (P_p(F))^3, \forall F \in \mathcal{F}_h\}.
\end{aligned}
$$

The definition of the jump is the same as in 2D, namely, for an interior interface $F = \partial K^+ \cap \partial K^- \in \mathcal{F}_i$, the jump $[\![\cdot]\!]$ of a vector $\mathbf{v}$ is

$$[\![\mathbf{v} \cdot \mathbf{n}]\!] = \mathbf{v}^+ \cdot \mathbf{n}^+ + \mathbf{v}^- \cdot \mathbf{n}^-,$$

and of a tensor $\underline{\underline{\sigma}}$

$$[\![\underline{\underline{\sigma}} \cdot \mathbf{n}]\!] = \underline{\underline{\sigma}}^+ \cdot \mathbf{n}^+ + \underline{\underline{\sigma}}^- \cdot \mathbf{n}^-.$$

For a boundary face $F = \partial K^+ \cap \Gamma \in \mathcal{F}_b$, the jump $[\![\cdot]\!]$ of a vector $\mathbf{v}$ is

$$[\![\mathbf{v} \cdot \mathbf{n}]\!] = \mathbf{v}^+ \cdot \mathbf{n}^+,$$

and of a tensor $\underline{\underline{\sigma}}$

$$[\![\underline{\underline{\sigma}} \cdot \mathbf{n}]\!] = \underline{\underline{\sigma}}^+ \cdot \mathbf{n}^+.$$

5.3 Principle of the 3D HDG formulation

The principle of the 3D HDG formulation is similar to the 2D case. We start by considering equations (5.1.1) on an element K of $\mathcal{T}_h$, still assuming physical piecewise constant parameters (ρ, λ and μ for the isotropic case and ρ and the C_{ij} coefficients in the general case).
We multiply (5.1.1) by test functions $(\mathbf{w}, \underline{\underline{\xi}}) \in \mathbf{V}_h^p(K) \times \mathbf{\Sigma}_h^p(K)$, integrate by parts and find the approximation $(\mathbf{v}_h, \underline{\underline{\sigma}}_h) \in \mathbf{V}_h^p \times \mathbf{\Sigma}_h^p$ such as

$$
\left\{
\begin{array}{rcl}
\displaystyle\int_K i\omega\rho_K \mathbf{v}_h \cdot \mathbf{w} + \int_K \underline{\underline{\sigma}}_h : \nabla\mathbf{w} - \int_{\partial K} \widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n} \cdot \mathbf{w} &=& 0,\\
\displaystyle\int_K i\omega \underline{\underline{\sigma}}_h : \underline{\underline{\xi}} + \int_K \mathbf{v}_h \cdot \nabla \cdot \left(\underline{\underline{C}}_K \underline{\underline{\xi}}\right) - \int_{\partial K} \widehat{\mathbf{v}}_h \cdot \underline{\underline{C}}_K \underline{\underline{\xi}} \cdot \mathbf{n} &=& \displaystyle\int_K \mathbf{f} : \underline{\underline{\xi}}.
\end{array}
\right. \tag{5.3.1}
$$

As in 2D, the numerical traces $\widehat{\underline{\underline{\sigma}}}_h$ and $\widehat{\mathbf{v}}_h$ represent the approximations of $\underline{\underline{\sigma}}$ and $\mathbf{v}$ on ∂K; and we define them as

$$\widehat{\mathbf{v}}_h = \lambda_h, \forall F \in \mathcal{F}_h, \qquad \lambda_h \in \mathbf{M}_h, \tag{5.3.2}$$

and

$$\widehat{\underline{\underline{\sigma}}}_h = \underline{\underline{\sigma}}_h - \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right) \otimes \mathbf{n} \qquad \text{on } \partial K. \tag{5.3.3}$$

We choose $\mathbf{S}$, the local stabilization matrix, to be a diagonal matrix, such that $\mathbf{S} = \tau \mathbf{I}$ where $\tau > 0$ is the local stabilization parameter and $\mathbf{I}$ the identity matrix.

We sum contributions of (5.3.1) over all elements and enforce the continuity of the normal component of $\widehat{\underline{\underline{\sigma}}}_h$ in order to rewrite the problem:
find $(\mathbf{v}_h, \underline{\underline{\sigma}}_h, \lambda_h) \in \mathbf{V}_h^p \times \mathbf{\Sigma}_h^p \times \mathbf{M}_h$ such as: $\forall (\mathbf{w}, \underline{\underline{\xi}}, \eta) \in \mathbf{V}^p(K) \times \mathbf{\Sigma}^p(K) \times \mathbf{M}_h$,

$$\left\{\begin{array}{rcl} \displaystyle\sum_{K\in\mathcal{T}_h} \int_K i\omega\rho_K \mathbf{v}_h \cdot \mathbf{w} + \sum_{K\in\mathcal{T}_h} \int_K \underline{\underline{\sigma}}_h : \nabla \mathbf{w} - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n} \cdot \mathbf{w} &=& 0, \\ \displaystyle\sum_{K\in\mathcal{T}_h} \int_K i\omega\underline{\underline{\sigma}}_h : \underline{\underline{\xi}} + \sum_{K\in\mathcal{T}_h} \int_K \mathbf{v}_h \cdot \nabla \cdot \left(\underline{\underline{C}}_K \underline{\underline{\xi}}\right) - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \lambda_h \cdot \underline{\underline{C}}_K \underline{\underline{\xi}} \cdot \mathbf{n} &=& \displaystyle\sum_{K\in\mathcal{T}_h} \int_K \mathbf{f} : \underline{\underline{\xi}}, \\ \displaystyle\sum_{F\in\mathcal{F}_h} \int_F [\![\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n}]\!] \cdot \eta &=& 0. \end{array}\right. \tag{5.3.4}$$

According to the definition (5.3.3), the expression of $\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n}$ on ∂K reads as

$$\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n} = \underline{\underline{\sigma}}_h \cdot \mathbf{n} - \mathbf{S}\left(\mathbf{v}_h - \widehat{\mathbf{v}}_h\right). \tag{5.3.5}$$

Hence,

$$\sum_{F\in\mathcal{F}_h} \int_F [\![\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n}]\!] \cdot \eta = \sum_{K\in\mathcal{T}_h} \int_{\partial K} \left(\underline{\underline{\sigma}}_h \cdot \mathbf{n}\right) \cdot \eta - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \mathbf{S}\left(\mathbf{v}_h - \widehat{\mathbf{v}}_h\right) \cdot \eta,$$

and we can rewrite system (5.3.4) as

$$\left\{\begin{array}{rcl} \displaystyle\sum_{K\in\mathcal{T}_h} \int_K i\omega\rho_K \mathbf{v}_h \cdot \mathbf{w} - \sum_{K\in\mathcal{T}_h} \int_K \left(\nabla \cdot \underline{\underline{\sigma}}_h\right) \cdot \mathbf{w} + \sum_{K\in\mathcal{T}_h} \int_{\partial K} \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right) \cdot \mathbf{w} &=& 0, \\ \displaystyle\sum_{K\in\mathcal{T}_h} \int_K i\omega\underline{\underline{\sigma}}_h : \underline{\underline{\xi}} + \sum_{K\in\mathcal{T}_h} \int_K \mathbf{v}_h \cdot \nabla \cdot \left(\underline{\underline{C}}_K \underline{\underline{\xi}}\right) - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \lambda_h \cdot \underline{\underline{C}}_K \underline{\underline{\xi}} \cdot \mathbf{n} &=& \displaystyle\sum_{K\in\mathcal{T}_h} \int_K \mathbf{f} : \underline{\underline{\xi}}, \\ \displaystyle\sum_{K\in\mathcal{T}_h} \int_{\partial K} \left(\underline{\underline{\sigma}}_h \cdot \mathbf{n}\right) \cdot \eta - \sum_{K\in\mathcal{T}_h} \int_{\partial K} \mathbf{S}\left(\mathbf{v}_h - \lambda_h\right) \cdot \eta &=& 0. \end{array}\right. \tag{5.3.6}$$

The local problem on an element K is written as in 2D

$$\left\{\begin{array}{rcl} \displaystyle\int_K i\omega\rho_K \mathbf{v}_h^K \cdot \mathbf{w} - \int_K \left(\nabla \cdot \underline{\underline{\sigma}}_h^K\right) \cdot \mathbf{w} + \int_{\partial K} \mathbf{S}\left(\mathbf{v}_h^K - \lambda_h\right) \cdot \mathbf{w} &=& 0, \\ \displaystyle\int_K i\omega\underline{\underline{\sigma}}_h^K : \underline{\underline{\xi}} + \int_K \mathbf{v}_h^K \cdot \nabla \cdot \left(\underline{\underline{C}}_K \underline{\underline{\xi}}\right) - \int_{\partial K} \lambda_h \cdot \underline{\underline{C}}_K \underline{\underline{\xi}} \cdot \mathbf{n} &=& \displaystyle\int_K \mathbf{f} : \underline{\underline{\xi}}. \end{array}\right. \tag{5.3.7}$$

From now, we omit the index h after the unknowns $\mathbf{v}^K$, $\underline{\underline{\sigma}}^K$ and λ in order to simplify the presentation.

5.4 Discretization of the 3D HDG formulation

The discretization of the 3D HDG formulation is similar to the 2D case, the differences mainly concern the number of unknowns and the size of the matrices. We first consider the discretization of the isotropic case, before discretizing the anisotropic equations. In both cases, as in 2D, the right-hand side $\mathbf{f}$ is first assumed equal to zero.

5.4.1 Discretization for the isotropic case

Taking as test-function the basis function φ^K, we develop the local equations (5.3.7) and write the local solution $(\mathbf{v}^K, \underline{\underline{\sigma}}^K)$ as a function of λ. We obtain the following equations

$$
\left\{
\begin{array}{rcl}
\displaystyle\int_K i\omega\rho_K v_x^K\varphi^K - \int_K \frac{\partial\sigma_{xx}^K}{\partial x}\varphi^K - \int_K \frac{\partial\sigma_{xy}^K}{\partial y}\varphi^K - \int_K \frac{\partial\sigma_{xz}^K}{\partial z}\varphi^K & & \\
\qquad + \displaystyle\int_{\partial K} \tau^K v_x^K\varphi^K - \int_{\partial K}\tau^K\lambda_x\varphi^K & = & 0, \\
\displaystyle\int_K i\omega\rho_K v_y^K\varphi^K - \int_K \frac{\partial\sigma_{xy}^K}{\partial x}\varphi^K - \int_K \frac{\partial\sigma_{yy}^K}{\partial y}\varphi^K - \int_K \frac{\partial\sigma_{yz}^K}{\partial z}\varphi^K & & \\
\qquad + \displaystyle\int_{\partial K} \tau^K v_y^K\varphi^K - \int_{\partial K}\tau^K\lambda_y\varphi^K & = & 0, \\
\displaystyle\int_K i\omega\rho_K v_z^K\varphi^K - \int_K \frac{\partial\sigma_{xz}^K}{\partial x}\varphi^K - \int_K \frac{\partial\sigma_{yz}^K}{\partial y}\varphi^K - \int_K \frac{\partial\sigma_{zz}^K}{\partial z}\varphi^K & & \\
\qquad + \displaystyle\int_{\partial K} \tau^K v_z^K\varphi^K - \int_{\partial K}\tau^K\lambda_z\varphi^K & = & 0, \\
\displaystyle\int_K i\omega\sigma_{xx}^K\varphi^K + \int_K (\lambda_L+2\mu)\, v_x^K\frac{\partial\varphi^K}{\partial x} + \int_K \lambda_L v_y^K\frac{\partial\varphi^K}{\partial y} + \int_K \lambda_L v_z^K\frac{\partial\varphi^K}{\partial z} & & \\
\qquad - \displaystyle\int_{\partial K}(\lambda_L+2\mu)\,\lambda_x\varphi^K n_x - \int_{\partial K}\lambda_L\lambda_y\varphi^K n_y - \int_{\partial K}\lambda_L\lambda_z\varphi^K n_z & = & 0, \\
\displaystyle\int_K i\omega\sigma_{yy}^K\varphi^K + \int_K \lambda_L v_x^K\frac{\partial\varphi^K}{\partial x} + \int_K (\lambda_L+2\mu)\, v_y^K\frac{\partial\varphi^K}{\partial y} + \int_K \lambda_L v_z^K\frac{\partial\varphi^K}{\partial z} & & \\
\qquad - \displaystyle\int_{\partial K}\lambda_L\lambda_x\varphi^K n_x - \int_{\partial K}(\lambda_L+2\mu)\,\lambda_y\varphi^K n_y - \int_{\partial K}\lambda_L\lambda_z\varphi^K n_z & = & 0, \\
\displaystyle\int_K i\omega\sigma_{zz}^K\varphi^K + \int_K \lambda_L v_x^K\frac{\partial\varphi^K}{\partial x} + \int_K \lambda_L v_y^K\frac{\partial\varphi^K}{\partial y} + \int_K (\lambda_L+2\mu)\, v_z^K\frac{\partial\varphi^K}{\partial z} & & \\
\qquad - \displaystyle\int_{\partial K}\lambda_L\lambda_x\varphi^K n_x - \int_{\partial K}\lambda_L\lambda_y\varphi^K n_y - \int_{\partial K}(\lambda_L+2\mu)\,\lambda_z\varphi^K n_z & = & 0, \\
\displaystyle\int_K i\omega\sigma_{xy}^K\varphi^K + \int_K \mu v_x^K\frac{\partial\varphi^K}{\partial y} + \int_K \mu v_y^K\frac{\partial\varphi^K}{\partial x} & & \\
\qquad - \displaystyle\int_{\partial K}\mu\lambda_x\varphi^K n_y - \int_{\partial K}\mu\lambda_y\varphi^K n_x & = & 0, \\
\displaystyle\int_K i\omega\sigma_{xz}^K\varphi^K + \int_K \mu v_x^K\frac{\partial\varphi^K}{\partial z} + \int_K \mu v_z^K\frac{\partial\varphi^K}{\partial x} & & \\
\qquad - \displaystyle\int_{\partial K}\mu\lambda_x\varphi^K n_z - \int_{\partial K}\mu\lambda_z\varphi^K n_x & = & 0, \\
\displaystyle\int_K i\omega\sigma_{yz}^K\varphi^K + \int_K \mu v_y^K\frac{\partial\varphi^K}{\partial z} + \int_K \mu v_z^K\frac{\partial\varphi^K}{\partial y} & & \\
\qquad - \displaystyle\int_{\partial K}\mu\lambda_y\varphi^K n_z - \int_{\partial K}\mu\lambda_z\varphi^K n_y & = & 0.
\end{array}
\right. \tag{5.4.1}
$$

Remark. *We now denote Lamé's coefficient by λ_L in order to avoid confusion with the Lagrange multiplier λ.*

We decompose the local solutions $(\mathbf{v}^K, \underline{\underline{\sigma}}^K)$ as

$$\begin{aligned} &\text{- For an element } K,\ v_l^K = \sum_{j=1}^{d_i^K} \underline{v}_{l,j}^K \varphi_j^K, \qquad l = x, y, z, \\ &\qquad\qquad \sigma_{kl}^K = \sum_{j=1}^{d_i^K} \underline{\sigma}_{kl,j}^K \varphi_j^K, \qquad k, l = x, y, z, \end{aligned} \tag{5.4.2}$$

where φ_j^K are the basis functions of $P_p(K)$ and d_i^K the associated degrees of freedom.

$$\text{- For a face } F,\ \lambda_l^F = \sum_{j=1}^{d_i^F} \underline{\lambda}_{l,j}^F \psi_j^F, \qquad l = x, y, z, \tag{5.4.3}$$

where ψ_j^F are the basis functions of $P_p(F)$ and d_i^F the associated degrees of freedom.
We write by $\beta(K,l)$ the global index of the *l-th* face of the element K ($l = 1, 2, 3, 4$); for example, if the *l-th* face of K is the *j-th* face F_j then $\beta(K,l) = j$.
Conversely if F_l is the common face between K^+ and K^-, we define $\eta(l,+) = K^+$ and $\eta(l,-) = K^-$.
After discretization, the local linear system resulting from (5.4.1) is

$$\left\{\begin{array}{rcl}
i\omega\rho\mathbb{M}^K \underline{v}_x^K - \mathbb{D}_x^{K^T} \underline{\sigma}_{xx}^K - \mathbb{D}_y^{K^T} \underline{\sigma}_{xy}^K - \mathbb{D}_z^{K^T} \underline{\sigma}_{xz}^K & & \\
\qquad + \sum_{l=1}^4 \tau^{(K,l)} \mathbb{E}_l^K \underline{v}_x^K - \sum_{l=1}^4 \tau^{(K,l)} \mathbb{F}_l^K \underline{\lambda}_x^{\beta(K,l)} & = & 0, \\
i\omega\rho\mathbb{M}^K \underline{v}_y^K - \mathbb{D}_x^{K^T} \underline{\sigma}_{xy}^K - \mathbb{D}_y^{K^T} \underline{\sigma}_{yy}^K - \mathbb{D}_z^{K^T} \underline{\sigma}_{yz}^K & & \\
\qquad + \sum_{l=1}^4 \tau^{(K,l)} \mathbb{E}_l^K \underline{v}_y^K - \sum_{l=1}^4 \tau^{(K,l)} \mathbb{F}_l^K \underline{\lambda}_y^{\beta(K,l)} & = & 0, \\
i\omega\rho\mathbb{M}^K \underline{v}_z^K - \mathbb{D}_x^{K^T} \underline{\sigma}_{xz}^K - \mathbb{D}_y^{K^T} \underline{\sigma}_{yz}^K - \mathbb{D}_z^{K^T} \underline{\sigma}_{zz}^K & & \\
\qquad + \sum_{l=1}^4 \tau^{(K,l)} \mathbb{E}_l^K \underline{v}_z^K - \sum_{l=1}^4 \tau^{(K,l)} \mathbb{F}_l^K \underline{\lambda}_z^{\beta(K,l)} & = & 0, \\
i\omega\mathbb{M}^K \underline{\sigma}_{xx}^K + (\lambda_L + 2\mu)\mathbb{D}_x^K \underline{v}_x^K + \lambda_L \mathbb{D}_y^K \underline{v}_y^K + \lambda_L \mathbb{D}_z^K \underline{v}_z^K & & \\
\qquad - \sum_{l=1}^4 (\lambda_L + 2\mu)\underline{\lambda}_x^{\beta(K,l)} \mathbb{Q}_{xl}^K - \sum_{l=1}^4 \lambda_L \underline{\lambda}_y^{\beta(K,l)} \mathbb{Q}_{yl}^K - \sum_{l=1}^4 \lambda_L \underline{\lambda}_z^{\beta(K,l)} \mathbb{Q}_{zl}^K & = & 0, \\
i\omega\mathbb{M}^K \underline{\sigma}_{yy}^K + \lambda_L \mathbb{D}_x^K \underline{v}_x^K + (\lambda_L + 2\mu)\mathbb{D}_y^K \underline{v}_y^K + \lambda_L \mathbb{D}_z^K \underline{v}_z^K & & \\
\qquad - \sum_{l=1}^4 \lambda_L \underline{\lambda}_x^{\beta(K,l)} \mathbb{Q}_{xl}^K - \sum_{l=1}^4 (\lambda_L + 2\mu)\underline{\lambda}_y^{\beta(K,l)} \mathbb{Q}_{yl}^K - \sum_{l=1}^4 \lambda_L \underline{\lambda}_z^{\beta(K,l)} \mathbb{Q}_{zl}^K & = & 0, \\
i\omega\mathbb{M}^K \underline{\sigma}_{zz}^K + \lambda_L \mathbb{D}_x^K \underline{v}_x^K + \lambda_L \mathbb{D}_y^K \underline{v}_y^K + (\lambda_L + 2\mu)\mathbb{D}_z^K \underline{v}_z^K & & \\
\qquad - \sum_{l=1}^4 \lambda_L \underline{\lambda}_x^{\beta(K,l)} \mathbb{Q}_{xl}^K - \sum_{l=1}^4 \lambda_L \underline{\lambda}_y^{\beta(K,l)} \mathbb{Q}_{yl}^K - \sum_{l=1}^4 (\lambda_L + 2\mu)\underline{\lambda}_z^{\beta(K,l)} \mathbb{Q}_{zl}^K & = & 0, \\
i\omega\mathbb{M}^K \underline{\sigma}_{xy}^K + \mu\left(\mathbb{D}_x^K \underline{v}_y^K + \mathbb{D}_y^K \underline{v}_x^K\right) - \sum_{l=1}^4 \mu\left(\underline{\lambda}_x^{\beta(K,l)} \mathbb{Q}_{yl}^K + \underline{\lambda}_y^{\beta(K,l)} \mathbb{Q}_{xl}^K\right) & = & 0, \\
i\omega\mathbb{M}^K \underline{\sigma}_{xz}^K + \mu\left(\mathbb{D}_x^K \underline{v}_z^K + \mathbb{D}_z^K \underline{v}_x^K\right) - \sum_{l=1}^4 \mu\left(\underline{\lambda}_x^{\beta(K,l)} \mathbb{Q}_{zl}^K + \underline{\lambda}_z^{\beta(K,l)} \mathbb{Q}_{xl}^K\right) & = & 0, \\
i\omega\mathbb{M}^K \underline{\sigma}_{yz}^K + \mu\left(\mathbb{D}_y^K \underline{v}_z^K + \mathbb{D}_z^K \underline{v}_y^K\right) - \sum_{l=1}^4 \mu\left(\underline{\lambda}_y^{\beta(K,l)} \mathbb{Q}_{zl}^K + \underline{\lambda}_z^{\beta(K,l)} \mathbb{Q}_{yl}^K\right) & = & 0.
\end{array}\right. \tag{5.4.4}$$

with the following entries:

$$\mathbb{M}^K_{ij} = \int_K \varphi^K_i \varphi^K_j d\mathbf{x},$$
$$\mathbb{D}^K_{u,ij} = \int_K \varphi^K_i \partial_u \varphi^K_j d\mathbf{x}, \quad \text{with } u = x, y, z,$$
$$\mathbb{E}^K_{l,ij} = \int_{\partial K^l} \varphi^K_i \varphi^K_j ds,$$
$$\mathbb{F}^K_{l,ij} = \int_{\partial K^l} \psi^{\beta(K,l)}_i \varphi^K_j ds,$$
$$\mathbb{Q}^K_{ul,ij} = \int_{\partial K^l} n^K_u \psi^{\beta(K,l)}_i \varphi^K_j, \quad \text{with } u = x, y, z.$$

The linear local system on an element K reads as

$$\mathbb{A}^K \underline{W}^K + \mathbb{C}^K \underline{\Lambda}^K = 0, \tag{5.4.5}$$

where

$$\underline{W}^K = \left[\underline{v}^K_x, \ \underline{v}^K_y, \ \underline{v}^K_z, \ \underline{\sigma}^K_{xx}, \ \underline{\sigma}^K_{yy}, \ \underline{\sigma}^K_{zz}, \ \underline{\sigma}^K_{xy}, \ \underline{\sigma}^K_{xz}, \ \underline{\sigma}^K_{yz}\right]^T,$$

$$\underline{\Lambda}^K = \left[\underline{\lambda}_x, \ \underline{\lambda}_y, \ \underline{\lambda}_z\right]^T, \ \text{with } \underline{\lambda}_x = \left[\underline{\lambda}^{\beta(K,1)}_x, \ \underline{\lambda}^{\beta(K,2)}_x, \ \underline{\lambda}^{\beta(K,3)}_x, \ \underline{\lambda}^{\beta(K,4)}_x\right],$$
$$\underline{\lambda}_y = \left[\underline{\lambda}^{\beta(K,1)}_y, \ \underline{\lambda}^{\beta(K,2)}_y, \ \underline{\lambda}^{\beta(K,3)}_y, \ \underline{\lambda}^{\beta(K,4)}_y\right],$$
$$\underline{\lambda}_z = \left[\underline{\lambda}^{\beta(K,1)}_z, \ \underline{\lambda}^{\beta(K,2)}_z, \ \underline{\lambda}^{\beta(K,3)}_z, \ \underline{\lambda}^{\beta(K,4)}_z\right],$$

$$\mathbb{A}^K = \left[\mathbb{A}^K_1 \ \ \mathbb{A}^K_2 \ \ \mathbb{A}^K_3 \ \ \mathbb{A}^K_4 \ \ \mathbb{A}^K_5 \ \ \mathbb{A}^K_6 \ \ \mathbb{A}^K_7 \ \ \mathbb{A}^K_8 \ \ \mathbb{A}^K_9\right]^T,$$

$$\text{with } \mathbb{A}^K_1 = \left[aux \ \ , \ 0, \ 0, \ -\mathbb{D}^{K^T}_x, \ 0, \ 0, \ -\mathbb{D}^{K^T}_y, \ -\mathbb{D}^{K^T}_z, \ 0\right]$$
$$\mathbb{A}^K_2 = \left[\ 0 \ \ , \ aux, \ 0, \ 0, \ -\mathbb{D}^{K^T}_y, \ 0, \ -\mathbb{D}^{K^T}_x, \ 0, \ -\mathbb{D}^{K^T}_z\right]$$
$$\mathbb{A}^K_3 = \left[\ 0 \ \ , \ 0, \ aux, \ 0, \ 0, \ -\mathbb{D}^{K^T}_z, \ 0, \ -\mathbb{D}^{K^T}_x, \ -\mathbb{D}^{K^T}_y\right]$$
$$\mathbb{A}^K_4 = \left[(\lambda+2\mu)\mathbb{D}^K_x, \ \lambda\mathbb{D}^K_y, \ \lambda\mathbb{D}^K_z, \ i\omega\rho\mathbb{M}^K \ \ 0, \ 0 \ \ 0, \ 0 \ \ 0\right]$$
$$\mathbb{A}^K_5 = \left[\ \lambda\mathbb{D}^K_x, \ (\lambda+2\mu)\mathbb{D}^K_y, \ \lambda\mathbb{D}^K_z, \ 0 \ \ i\omega\rho\mathbb{M}^K, \ 0 \ \ 0, \ 0 \ \ 0\right]$$
$$\mathbb{A}^K_6 = \left[\ \lambda\mathbb{D}^K_x, \ \lambda\mathbb{D}^K_y, \ (\lambda+2\mu)\mathbb{D}^K_z, \ 0 \ \ 0, \ i\omega\rho\mathbb{M}^K \ \ 0, \ 0 \ \ 0\right]$$
$$\mathbb{A}^K_7 = \left[\ \mu\mathbb{D}^K_y, \ \mu\mathbb{D}^K_x, \ 0, \ 0, \ 0, \ 0 \ \ i\omega\rho\mathbb{M}^K \ \ 0 \ \ 0\right]$$
$$\mathbb{A}^K_8 = \left[\ \mu\mathbb{D}^K_z, \ 0 \ \ \mu\mathbb{D}^K_x \ \ 0, \ 0, \ 0, \ 0, \ i\omega\rho\mathbb{M}^K \ \ 0\right]$$
$$\mathbb{A}^K_9 = \left[\ \ 0 \ \ , \ \mu\mathbb{D}^K_z, \ \mu\mathbb{D}^K_y, \ 0, \ 0, \ 0, \ 0, \ 0, \ i\omega\rho\mathbb{M}^K\right]$$

where $aux = i\omega\rho\mathbb{M}^K + \sum_{l=1}^{4} \tau^{(K,l)}\mathbb{E}_l^K$.

$$\mathbb{C}^K = -\begin{bmatrix}\mathbb{C}_x^K & \mathbb{C}_y^K & \mathbb{C}_z^K\end{bmatrix},$$

with $\mathbb{C}_x^K = -\begin{bmatrix}
\tau^{(K,1)}\mathbb{F}_1^K & \tau^{(K,2)}\mathbb{F}_2^K & \tau^{(K,3)}\mathbb{F}_3^K & \tau^{(K,4)}\mathbb{F}_4^K \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
(\lambda_L+2\mu)\mathbb{Q}_{x1}^K & (\lambda_L+2\mu)\mathbb{Q}_{x2}^K & (\lambda_L+2\mu)\mathbb{Q}_{x3}^K & (\lambda_L+2\mu)\mathbb{Q}_{x4}^K \\
\lambda_L\mathbb{Q}_{x1}^K & \lambda_L\mathbb{Q}_{x2}^K & \lambda_L\mathbb{Q}_{x3}^K & \lambda_L\mathbb{Q}_{x4}^K \\
\lambda_L\mathbb{Q}_{x1}^K & \lambda_L\mathbb{Q}_{x2}^K & \lambda_L\mathbb{Q}_{x3}^K & \lambda_L\mathbb{Q}_{x4}^K \\
\mu\mathbb{Q}_{y1}^K & \mu\mathbb{Q}_{y2}^K & \mu\mathbb{Q}_{y3}^K & \mu\mathbb{Q}_{y4}^K \\
\mu\mathbb{Q}_{z1}^K & \mu\mathbb{Q}_{z2}^K & \mu\mathbb{Q}_{z3}^K & \mu\mathbb{Q}_{z4}^K \\
0 & 0 & 0 & 0
\end{bmatrix}$,

$\mathbb{C}_y^K = -\begin{bmatrix}
0 & 0 & 0 & 0 \\
\tau^{(K,1)}\mathbb{F}_1^K & \tau^{(K,2)}\mathbb{F}_2^K & \tau^{(K,3)}\mathbb{F}_3^K & \tau^{(K,4)}\mathbb{F}_4^K \\
0 & 0 & 0 & 0 \\
\lambda_L\mathbb{Q}_{y1}^K & \lambda_L\mathbb{Q}_{y2}^K & \lambda_L\mathbb{Q}_{y3}^K & \lambda_L\mathbb{Q}_{y4}^K \\
(\lambda_L+2\mu)\mathbb{Q}_{y1}^K & (\lambda_L+2\mu)\mathbb{Q}_{y2}^K & (\lambda_L+2\mu)\mathbb{Q}_{y3}^K & (\lambda_L+2\mu)\mathbb{Q}_{y4}^K \\
\lambda_L\mathbb{Q}_{y1}^K & \lambda_L\mathbb{Q}_{y2}^K & \lambda_L\mathbb{Q}_{y3}^K & \lambda_L\mathbb{Q}_{y4}^K \\
\mu\mathbb{Q}_{x1}^K & \mu\mathbb{Q}_{x2}^K & \mu\mathbb{Q}_{x3}^K & \mu\mathbb{Q}_{x4}^K \\
0 & 0 & 0 & 0 \\
\mu\mathbb{Q}_{z1}^K & \mu\mathbb{Q}_{z2}^K & \mu\mathbb{Q}_{z3}^K & \mu\mathbb{Q}_{z4}^K
\end{bmatrix}$,

$\mathbb{C}_z^K = -\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\tau^{(K,1)}\mathbb{F}_1^K & \tau^{(K,2)}\mathbb{F}_2^K & \tau^{(K,3)}\mathbb{F}_3^K & \tau^{(K,4)}\mathbb{F}_4^K \\
\lambda_L\mathbb{Q}_{z1}^K & \lambda_L\mathbb{Q}_{z2}^K & \lambda_L\mathbb{Q}_{z3}^K & \lambda_L\mathbb{Q}_{z4}^K \\
\lambda_L\mathbb{Q}_{z1}^K & \lambda_L\mathbb{Q}_{z2}^K & \lambda_L\mathbb{Q}_{z3}^K & \lambda_L\mathbb{Q}_{z4}^K \\
(\lambda_L+2\mu)\mathbb{Q}_{z1}^K & (\lambda_L+2\mu)\mathbb{Q}_{z2}^K & (\lambda_L+2\mu)\mathbb{Q}_{z3}^K & (\lambda_L+2\mu)\mathbb{Q}_{z4}^K \\
0 & 0 & 0 & 0 \\
\mu\mathbb{Q}_{x1}^K & \mu\mathbb{Q}_{x2}^K & \mu\mathbb{Q}_{x3}^K & \mu\mathbb{Q}_{x4}^K \\
\mu\mathbb{Q}_{y1}^K & \mu\mathbb{Q}_{y2}^K & \mu\mathbb{Q}_{y3}^K & \mu\mathbb{Q}_{y4}^K
\end{bmatrix}$.

The transmission condition on a face $F_j = \partial K^+ \cap \partial K^-$ such as $j = \beta(K^+, l) = \beta(K^-, g)$ reads as $\forall \eta \in \mathbf{M}_h$

$$\int_{F_j} \left(\underline{\underline{\sigma}}_h^{K^+} \cdot \mathbf{n}^{K^+} \cdot \eta + \underline{\underline{\sigma}}_h^{K^-} \cdot \mathbf{n}^{K^-} \cdot \eta - \mathbf{S}^{K^+}\left(\mathbf{v}_h^{K^+} - \lambda_h\right) \cdot \eta - \mathbf{S}^{K^-}\left(\mathbf{v}_h^{K^-} - \lambda_h\right) \cdot \eta \right) = 0.$$

So, we have the corresponding discretization

$$\begin{cases} \mathbb{Q}_{xl}^{K^+{}^T}\underline{\sigma}_{xx}^{K^+} + \mathbb{Q}_{yl}^{K^+{}^T}\underline{\sigma}_{xy}^{K^+} + \mathbb{Q}_{zl}^{K^+{}^T}\underline{\sigma}_{xz}^{K^+} \\ \qquad - \tau^{(K^+,l)}\mathbb{F}_l^{K^+{}^T}\underline{v}_x^{K^+} + \tau^{(K^+,l)}\mathbb{G}^j\underline{\lambda}_x^{\beta(K^+,l)} \\ \qquad + \mathbb{Q}_{xl}^{K^-{}^T}\underline{\sigma}_{xx}^{K^-} + \mathbb{Q}_{yl}^{K^-{}^T}\underline{\sigma}_{xy}^{K^-} + \mathbb{Q}_{zl}^{K^-{}^T}\underline{\sigma}_{xz}^{K^-} \\ \qquad - \tau^{(K^-,l)}\mathbb{F}_l^{K^-{}^T}\underline{v}_x^{K^-} + \tau^{(K^-,l)}\mathbb{G}^j\underline{\lambda}_x^{\beta(K^-,l)} = 0, \\ \mathbb{Q}_{xl}^{K^+{}^T}\underline{\sigma}_{xy}^{K^+} + \mathbb{Q}_{yl}^{K^+{}^T}\underline{\sigma}_{yy}^{K^+} + \mathbb{Q}_{zl}^{K^+{}^T}\underline{\sigma}_{yz}^{K^+} \\ \qquad - \tau^{(K^+,l)}\mathbb{F}_l^{K^+{}^T}\underline{v}_y^{K^+} + \tau^{(K^+,l)}\mathbb{G}^j\underline{\lambda}_y^{\beta(K^+,l)} \\ \qquad + \mathbb{Q}_{xl}^{K^-{}^T}\underline{\sigma}_{xy}^{K^-} + \mathbb{Q}_{yl}^{K^-{}^T}\underline{\sigma}_{yy}^{K^-} + \mathbb{Q}_{zl}^{K^-{}^T}\underline{\sigma}_{yz}^{K^-} \\ \qquad - \tau^{(K^-,l)}\mathbb{F}_l^{K^-{}^T}\underline{v}_y^{K^-} + \tau^{(K^-,l)}\mathbb{G}^j\underline{\lambda}_y^{\beta(K^-,l)} = 0, \\ \mathbb{Q}_{xl}^{K^+{}^T}\underline{\sigma}_{xz}^{K^+} + \mathbb{Q}_{yl}^{K^+{}^T}\underline{\sigma}_{yz}^{K^+} + \mathbb{Q}_{zl}^{K^+{}^T}\underline{\sigma}_{zz}^{K^+} \\ \qquad - \tau^{(K^+,l)}\mathbb{F}_l^{K^+{}^T}\underline{v}_z^{K^+} + \tau^{(K^+,l)}\mathbb{G}^j\underline{\lambda}_z^{\beta(K^+,l)} \\ \qquad + \mathbb{Q}_{xl}^{K^-{}^T}\underline{\sigma}_{xz}^{K^-} + \mathbb{Q}_{yl}^{K^-{}^T}\underline{\sigma}_{yz}^{K^-} + \mathbb{Q}_{zl}^{K^-{}^T}\underline{\sigma}_{xz}^{K^-} \\ \qquad - \tau^{(K^-,l)}\mathbb{F}_l^{K^-{}^T}\underline{v}_z^{K^-} + \tau^{(K^-,l)}\mathbb{G}^j\underline{\lambda}_z^{\beta(K^-,l)} = 0. \end{cases} \tag{5.4.6}$$

where

$$\mathbb{G}_{im}^j = \int_{F_j} \psi_i \psi_m ds.$$

From system (5.4.6) we deduce a local system for Λ^K

$$\mathbb{B}^K \underline{W}^K + \mathbb{L}^K \underline{\Lambda}^K + \mathcal{R}^K = 0 \tag{5.4.7}$$

where $\mathcal{R}^K$ gathers the contributions from the neighboring elements,

$$\mathbb{B}^K = \begin{bmatrix} -\tau^{(K,1)}\mathbb{F}_1^{K^T} & 0 & 0 & \mathbb{Q}_{x1}^{K^T} & 0 & 0 & \mathbb{Q}_{y1}^{K^T} & \mathbb{Q}_{z1}^{K^T} & 0 \\ -\tau^{(K,2)}\mathbb{F}_2^{K^T} & 0 & 0 & \mathbb{Q}_{x2}^{K^T} & 0 & 0 & \mathbb{Q}_{y2}^{K^T} & \mathbb{Q}_{z2}^{K^T} & 0 \\ -\tau^{(K,3)}\mathbb{F}_3^{K^T} & 0 & 0 & \mathbb{Q}_{x3}^{K^T} & 0 & 0 & \mathbb{Q}_{y3}^{K^T} & \mathbb{Q}_{z3}^{K^T} & 0 \\ -\tau^{(K,4)}\mathbb{F}_4^{K^T} & 0 & 0 & \mathbb{Q}_{x4}^{K^T} & 0 & 0 & \mathbb{Q}_{y4}^{K^T} & \mathbb{Q}_{z4}^{K^T} & 0 \\ 0 & -\tau^{(K,1)}\mathbb{F}_1^{K^T} & 0 & 0 & \mathbb{Q}_{y1}^{K^T} & 0 & \mathbb{Q}_{x1}^{K^T} & 0 & \mathbb{Q}_{z1}^{K^T} \\ 0 & -\tau^{(K,2)}\mathbb{F}_2^{K^T} & 0 & 0 & \mathbb{Q}_{y2}^{K^T} & 0 & \mathbb{Q}_{x2}^{K^T} & 0 & \mathbb{Q}_{z2}^{K^T} \\ 0 & -\tau^{(K,3)}\mathbb{F}_3^{K^T} & 0 & 0 & \mathbb{Q}_{y3}^{K^T} & 0 & \mathbb{Q}_{x3}^{K^T} & 0 & \mathbb{Q}_{z3}^{K^T} \\ 0 & -\tau^{(K,4)}\mathbb{F}_4^{K^T} & 0 & 0 & \mathbb{Q}_{y4}^{K^T} & 0 & \mathbb{Q}_{x4}^{K^T} & 0 & \mathbb{Q}_{z4}^{K^T} \\ 0 & 0 & -\tau^{(K,1)}\mathbb{F}_1^{K^T} & 0 & 0 & \mathbb{Q}_{z1}^{K^T} & 0 & \mathbb{Q}_{x1}^{K^T} & \mathbb{Q}_{y1}^{K^T} \\ 0 & 0 & -\tau^{(K,2)}\mathbb{F}_2^{K^T} & 0 & 0 & \mathbb{Q}_{z2}^{K^T} & 0 & \mathbb{Q}_{x2}^{K^T} & \mathbb{Q}_{y2}^{K^T} \\ 0 & 0 & -\tau^{(K,3)}\mathbb{F}_3^{K^T} & 0 & 0 & \mathbb{Q}_{z3}^{K^T} & 0 & \mathbb{Q}_{x3}^{K^T} & \mathbb{Q}_{y3}^{K^T} \\ 0 & 0 & -\tau^{(K,4)}\mathbb{F}_4^{K^T} & 0 & 0 & \mathbb{Q}_{z4}^{K^T} & 0 & \mathbb{Q}_{x4}^{K^T} & \mathbb{Q}_{y4}^{K^T} \end{bmatrix},$$

and $\mathbb{L}^K = \begin{bmatrix} \mathbb{L}_x^K & \mathbb{L}_y^K & \mathbb{L}_z^K \end{bmatrix}$, with

$$\mathbb{L}_x^K = \begin{bmatrix} \tau^{(K,1)}\mathbb{G}^{\beta(K,1)} & 0 & 0 & 0 \\ 0 & \tau^{(K,2)}\mathbb{G}^{\beta(K,2)} & 0 & 0 \\ 0 & 0 & \tau^{(K,3)}\mathbb{G}^{\beta(K,3)} & 0 \\ 0 & 0 & 0 & \tau^{(K,4)}\mathbb{G}^{\beta(K,4)} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbb{L}_y^K = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \tau^{(K,1)}\mathbb{G}^{\beta(K,1)} & 0 & 0 & 0 \\ 0 & \tau^{(K,2)}\mathbb{G}^{\beta(K,2)} & 0 & 0 \\ 0 & 0 & \tau^{(K,3)}\mathbb{G}^{\beta(K,3)} & 0 \\ 0 & 0 & 0 & \tau^{(K,4)}\mathbb{G}^{\beta(K,4)} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbb{L}_z^K = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \tau^{(K,1)}\mathbb{G}^{\beta(K,1)} & 0 & 0 & 0 \\ 0 & \tau^{(K,2)}\mathbb{G}^{\beta(K,2)} & 0 & 0 \\ 0 & 0 & \tau^{(K,3)}\mathbb{G}^{\beta(K,3)} & 0 \\ 0 & 0 & 0 & \tau^{(K,4)}\mathbb{G}^{\beta(K,4)} \end{bmatrix}.$$

As in 2D, we denote by $\underline{\Lambda}$ the vector of size $\overline{N}_\lambda$ gathering all the degrees of freedom of $\underline{\lambda}$ for each face

$$\underline{\Lambda} = \left(\underline{\lambda}^1, ..., \underline{\lambda}^{N_f}\right)^T,$$

with N_f the number of faces of the mesh. The number of degrees of freedom of the l-th face of the element K is denoted by $N_\lambda^{(K,l)}$ and the total number of degrees of freedom of Λ^K by $\overline{N}_\lambda^K$. Then, we have

$$\overline{N}_\lambda^K = \sum_{l=1}^{4} N_\lambda^{(K,l)}.$$

Thanks to the local trace space spreading operator $\mathcal{A}_{HDG}^K$ of size $\overline{N}_\lambda^K \times N_\lambda$ that we have defined in section 3.3.1 which allows to extract the local trace vector Λ^K from the global trace vector Λ, we write

$$\mathcal{A}_{HDG}^K \underline{\Lambda} = \underline{\Lambda}^K,$$

and we rewrite equation (5.4.5) as

$$\mathbb{A}^K \underline{W}^K + \mathbb{C}^K \mathcal{A}_{HDG}^K \underline{\Lambda} = 0. \tag{5.4.8}$$

We now express $\underline{W}^K$ in terms of $\underline{\Lambda}$

$$\underline{W}^K = -(\mathbb{A}^K)^{-1} \mathbb{C}^K \mathcal{A}_{HDG}^K \underline{\Lambda}. \tag{5.4.9}$$

Summing all the equations of the transmission condition on all the faces of each element, element by element, we obtain

$$\sum_{K \in \mathcal{T}_h} (\mathcal{A}_{HDG}^K)^T \left[\mathbb{B}^K \underline{W}^K + \mathbb{L}^K \mathcal{A}_{HDG}^K \underline{\Lambda} \right] = 0, \tag{5.4.10}$$

where the sum over all the elements along with the left application of the transpose of $\mathcal{A}_{HDG}^K$ allow to gather the elementwise contributions corresponding to faces.
Finally, by remplacing $\underline{W}^K$ in (5.4.10), we obtain a global system in $\underline{\Lambda}$:

$$\sum_{K \in \mathcal{T}_h} (\mathcal{A}_{HDG}^K)^T \left[-\mathbb{B}^K (\mathbb{A}^K)^{-1} \mathbb{C}^K + \mathbb{L}^K \right] \mathcal{A}_{HDG}^K \underline{\Lambda} = 0. \tag{5.4.11}$$

Remark. *The 3D HDG global system has the same expression than the 2D HDG global system* (3.3.12)*, only the sizes of the involved matrices change.*

Considering now a point source on element K_s, the local equations (5.3.7) reads as

$$\left\{ \begin{array}{rcl} \displaystyle\int_{K_s} i\omega\rho_{K_s} \mathbf{v}_h^{K_s} \cdot \mathbf{w} - \int_{K_s} \left(\nabla \cdot \underline{\underline{\sigma}}_h^{K_s} \right) \cdot \mathbf{w} + \int_{\partial K_s} \mathbf{S} \left(\mathbf{v}_h^{K_s} - \lambda_h \right) \cdot \mathbf{w} & = & 0, \\ \displaystyle\int_{K_s} i\omega \underline{\underline{\sigma}}_h^{K_s} : \underline{\underline{\xi}} + \int_{K_s} \mathbf{v}_h^{K_s} \cdot \nabla \cdot \left(\underline{\underline{C}}_{K_s} \underline{\underline{\xi}} \right) - \int_{\partial K_s} \lambda_h \cdot \underline{\underline{C}}_{K_s} \underline{\underline{\xi}} \cdot \mathbf{n} & = & \displaystyle\int_{K_s} \mathbf{f} : \underline{\underline{\xi}}. \end{array} \right.$$

and the corresponding discretization (5.4.8) becomes

$$\mathbb{A}^{K_s} \underline{W}^{K_s} + \mathbb{C}^{K_s} \mathcal{A}_{HDG}^{K_s} \underline{\Lambda} = \mathbb{S}^{K_s},$$

which leads to

$$\underline{W}^{K_s} = \left(\mathbb{A}^{K_s} \right)^{-1} \mathbb{S}^{K_s} - \left(\mathbb{A}^{K_s} \right)^{-1} \mathbb{C}^{K_s} \mathcal{A}_{HDG}^{K_s} \underline{\Lambda}.$$

Finally, we obtain the following 3D global system

$$\sum_{K \in \mathcal{T}_h} (\mathcal{A}_{HDG}^K)^T \left[-\mathbb{B}^K (\mathbb{A}^K)^{-1} \mathbb{C}^K + \mathbb{L}^K \right] \mathcal{A}_{HDG}^K \underline{\Lambda} = \sum_{K \in \mathcal{T}_h} -(\mathcal{A}_{HDG}^K)^T \mathbb{B}^K (\mathbb{A}^K)^{-1} \mathbb{S}^K. \tag{5.4.12}$$

5.4.2 Discretization for the anisotropic case

We now consider the local equation (5.3.7) for the anisotropic case. As we have done for all the discretizations in this study, we take the basis function φ^K as test function. We obtain

$$
\left\{
\begin{array}{rcl}
\displaystyle\int_K i\omega\rho_K v_x^K\varphi^K - \int_K \frac{\partial\sigma_{xx}^K}{\partial x}\varphi^K - \int_K \frac{\partial\sigma_{xy}^K}{\partial y}\varphi^K - \int_K \frac{\partial\sigma_{xz}^K}{\partial z}\varphi^K & & \\
\qquad + \displaystyle\int_{\partial K} \tau^K v_x^K\varphi^K - \int_{\partial K}\tau^K\lambda_x\varphi^K & = & 0 \\
\displaystyle\int_K i\omega\rho_K v_y^K\varphi^K - \int_K \frac{\partial\sigma_{xy}^K}{\partial x}\varphi^K - \int_K \frac{\partial\sigma_{yy}^K}{\partial y}\varphi^K - \int_K \frac{\partial\sigma_{yz}^K}{\partial z}\varphi^K & & \\
\qquad + \displaystyle\int_{\partial K} \tau^K v_y^K\varphi^K - \int_{\partial K}\tau^K\lambda_y\varphi^K & = & 0 \\
\displaystyle\int_K i\omega\rho_K v_z^K\varphi^K - \int_K \frac{\partial\sigma_{xz}^K}{\partial x}\varphi^K - \int_K \frac{\partial\sigma_{yz}^K}{\partial y}\varphi^K - \int_K \frac{\partial\sigma_{zz}^K}{\partial z}\varphi^K & & \\
\qquad + \displaystyle\int_{\partial K} \tau^K v_z^K\varphi^K - \int_{\partial K}\tau^K\lambda_z\varphi^K & = & 0 \\
\displaystyle\int_K i\omega\sigma_{xx}^K\varphi^K + \int_K v_x^K\left(C_{11}\frac{\partial\varphi^K}{\partial x} + C_{15}\frac{\partial\varphi^K}{\partial z} + C_{16}\frac{\partial\varphi^K}{\partial y}\right) + \int_K v_y^K\left(C_{12}\frac{\partial\varphi^K}{\partial y} + C_{14}\frac{\partial\varphi^K}{\partial z} + C_{16}\frac{\partial\varphi^K}{\partial x}\right) & & \\
+ \displaystyle\int_K v_z^K\left(C_{13}\frac{\partial\varphi^K}{\partial z} + C_{14}\frac{\partial\varphi^K}{\partial y} + C_{15}\frac{\partial\varphi^K}{\partial x}\right) - \int_{\partial K}\lambda_x\left(C_{11}\varphi^K n_x + C_{15}\varphi^K n_z + C_{16}\varphi^K n_y\right) & = & 0 \\
- \displaystyle\int_{\partial K}\lambda_y\left(C_{12}\varphi^K n_y + C_{14}\varphi^K n_z + C_{16}\varphi^K n_x\right) - \int_{\partial K}\lambda_z\left(C_{13}\varphi^K n_z + C_{14}\varphi^K n_y + C_{15}\varphi^K n_x\right) & & \\
\displaystyle\int_K i\omega\sigma_{yy}^K\varphi^K + \int_K v_x^K\left(C_{12}\frac{\partial\varphi^K}{\partial x} + C_{25}\frac{\partial\varphi^K}{\partial z} + C_{26}\frac{\partial\varphi^K}{\partial y}\right) + \int_K v_y^K\left(C_{22}\frac{\partial\varphi^K}{\partial y} + C_{24}\frac{\partial\varphi^K}{\partial z} + C_{26}\frac{\partial\varphi^K}{\partial x}\right) & & \\
+ \displaystyle\int_K v_z^K\left(C_{23}\frac{\partial\varphi^K}{\partial z} + C_{24}\frac{\partial\varphi^K}{\partial y} + C_{25}\frac{\partial\varphi^K}{\partial x}\right) - \int_{\partial K}\lambda_x\left(C_{12}\varphi^K n_x + C_{25}\varphi^K n_z + C_{26}\varphi^K n_y\right) & = & 0 \\
- \displaystyle\int_{\partial K}\lambda_y\left(C_{22}\varphi^K n_y + C_{24}\varphi^K n_z + C_{26}\varphi^K n_x\right) - \int_{\partial K}\lambda_z\left(C_{23}\varphi^K n_z + C_{24}\varphi^K n_y + C_{25}\varphi^K n_x\right) & & \\
\displaystyle\int_K i\omega\sigma_{zz}^K\varphi^K + \int_K v_x^K\left(C_{13}\frac{\partial\varphi^K}{\partial x} + C_{35}\frac{\partial\varphi^K}{\partial z} + C_{36}\frac{\partial\varphi^K}{\partial y}\right) + \int_K v_y^K\left(C_{23}\frac{\partial\varphi^K}{\partial y} + C_{34}\frac{\partial\varphi^K}{\partial z} + C_{36}\frac{\partial\varphi^K}{\partial x}\right) & & \\
+ \displaystyle\int_K v_z^K\left(C_{33}\frac{\partial\varphi^K}{\partial z} + C_{34}\frac{\partial\varphi^K}{\partial y} + C_{35}\frac{\partial\varphi^K}{\partial x}\right) - \int_{\partial K}\lambda_x\left(C_{13}\varphi^K n_x + C_{35}\varphi^K n_z + C_{36}\varphi^K n_y\right) & = & 0 \\
- \displaystyle\int_{\partial K}\lambda_y\left(C_{23}\varphi^K n_y + C_{34}\varphi^K n_z + C_{36}\varphi^K n_x\right) - \int_{\partial K}\lambda_z\left(C_{33}\varphi^K n_z + C_{34}\varphi^K n_y + C_{35}\varphi^K n_x\right) & & \\
\displaystyle\int_K i\omega\sigma_{xy}^K\varphi^K + \int_K v_x^K\left(C_{16}\frac{\partial\varphi^K}{\partial x} + C_{56}\frac{\partial\varphi^K}{\partial z} + C_{66}\frac{\partial\varphi^K}{\partial y}\right) + \int_K v_y^K\left(C_{26}\frac{\partial\varphi^K}{\partial y} + C_{46}\frac{\partial\varphi^K}{\partial z} + C_{66}\frac{\partial\varphi^K}{\partial x}\right) & & \\
+ \displaystyle\int_K v_z^K\left(C_{36}\frac{\partial\varphi^K}{\partial z} + C_{46}\frac{\partial\varphi^K}{\partial y} + C_{56}\frac{\partial\varphi^K}{\partial x}\right) - \int_{\partial K}\lambda_x\left(C_{16}\varphi^K n_x + C_{56}\varphi^K n_z + C_{66}\varphi^K n_y\right) & = & 0 \\
- \displaystyle\int_{\partial K}\lambda_y\left(C_{26}\varphi^K n_y + C_{46}\varphi^K n_z + C_{66}\varphi^K n_x\right) - \int_{\partial K}\lambda_z\left(C_{36}\varphi^K n_z + C_{46}\varphi^K n_y + C_{56}\varphi^K n_x\right) & & \\
\displaystyle\int_K i\omega\sigma_{xz}^K\varphi^K + \int_K v_x^K\left(C_{15}\frac{\partial\varphi^K}{\partial x} + C_{55}\frac{\partial\varphi^K}{\partial z} + C_{56}\frac{\partial\varphi^K}{\partial y}\right) + \int_K v_y^K\left(C_{25}\frac{\partial\varphi^K}{\partial y} + C_{45}\frac{\partial\varphi^K}{\partial z} + C_{56}\frac{\partial\varphi^K}{\partial x}\right) & & \\
+ \displaystyle\int_K v_z^K\left(C_{35}\frac{\partial\varphi^K}{\partial z} + C_{45}\frac{\partial\varphi^K}{\partial y} + C_{55}\frac{\partial\varphi^K}{\partial x}\right) - \int_{\partial K}\lambda_x\left(C_{15}\varphi^K n_x + C_{55}\varphi^K n_z + C_{56}\varphi^K n_y\right) & = & 0 \\
- \displaystyle\int_{\partial K}\lambda_y\left(C_{25}\varphi^K n_y + C_{45}\varphi^K n_z + C_{56}\varphi^K n_x\right) - \int_{\partial K}\lambda_z\left(C_{35}\varphi^K n_z + C_{45}\varphi^K n_y + C_{55}\varphi^K n_x\right) & & \\
\displaystyle\int_K i\omega\sigma_{yz}^K\varphi^K + \int_K v_x^K\left(C_{14}\frac{\partial\varphi^K}{\partial x} + C_{45}\frac{\partial\varphi^K}{\partial z} + C_{46}\frac{\partial\varphi^K}{\partial y}\right) + \int_K v_y^K\left(C_{24}\frac{\partial\varphi^K}{\partial y} + C_{44}\frac{\partial\varphi^K}{\partial z} + C_{46}\frac{\partial\varphi^K}{\partial x}\right) & & \\
+ \displaystyle\int_K v_z^K\left(C_{34}\frac{\partial\varphi^K}{\partial z} + C_{44}\frac{\partial\varphi^K}{\partial y} + C_{45}\frac{\partial\varphi^K}{\partial x}\right) - \int_{\partial K}\lambda_x\left(C_{14}\varphi^K n_x + C_{45}\varphi^K n_z + C_{46}\varphi^K n_y\right) & = & 0 \\
- \displaystyle\int_{\partial K}\lambda_y\left(C_{24}\varphi^K n_y + C_{44}\varphi^K n_z + C_{46}\varphi^K n_x\right) - \int_{\partial K}\lambda_z\left(C_{34}\varphi^K n_z + C_{44}\varphi^K n_y + C_{45}\varphi^K n_x\right) & &
\end{array}
\right. \tag{5.4.13}
$$

Using the discretization given in the previous section, the local linear system resulting from (5.4.13) is:

(5.4.14)

$$
\begin{cases}
i\omega\rho\mathbb{M}^K \underline{v}_x^K - \mathbb{D}_x^{K^T}\underline{\sigma}_{xx}^K - \mathbb{D}_y^{K^T}\underline{\sigma}_{xy}^K - \mathbb{D}_z^{K^T}\underline{\sigma}_{xz}^K + \sum_{l=1}^{4}\tau^{(K,l)}\mathbb{E}_l^K\underline{v}_x^K - \sum_{l=1}^{4}\tau^{(K,l)}\mathbb{F}_l^K\underline{\lambda}_x^{\beta(K,l)} = 0\\
i\omega\rho\mathbb{M}^K \underline{v}_y^K - \mathbb{D}_x^{K^T}\underline{\sigma}_{xy}^K - \mathbb{D}_y^{K^T}\underline{\sigma}_{yy}^K - \mathbb{D}_z^{K^T}\underline{\sigma}_{yz}^K + \sum_{l=1}^{4}\tau^{(K,l)}\mathbb{E}_l^K\underline{v}_y^K - \sum_{l=1}^{4}\tau^{(K,l)}\mathbb{F}_l^K\underline{\lambda}_y^{\beta(K,l)} = 0\\
i\omega\rho\mathbb{M}^K \underline{v}_z^K - \mathbb{D}_x^{K^T}\underline{\sigma}_{xz}^K - \mathbb{D}_y^{K^T}\underline{\sigma}_{yz}^K - \mathbb{D}_z^{K^T}\underline{\sigma}_{zz}^K + \sum_{l=1}^{4}\tau^{(K,l)}\mathbb{E}_l^K\underline{v}_z^K - \sum_{l=1}^{4}\tau^{(K,l)}\mathbb{F}_l^K\underline{\lambda}_z^{\beta(K,l)} = 0\\
i\omega\mathbb{M}^K\underline{\sigma}_{xx}^K + \left(C_{11}\mathbb{D}_x^K + C_{15}\mathbb{D}_z^K + C_{16}\mathbb{D}_y^K\right)\underline{v}_x^K + \left(C_{12}\mathbb{D}_y^K + C_{14}\mathbb{D}_z^K + C_{16}\mathbb{D}_x^K\right)\underline{v}_y^K\\
\qquad + \left(C_{13}\mathbb{D}_z^K + C_{14}\mathbb{D}_y^K + C_{15}\mathbb{D}_x^K\right)\underline{v}_z^K - \sum_{l=1}^{4}\underline{\lambda}_x^{\beta(K,l)}\left(C_{11}\mathbb{Q}_{xl}^K + C_{15}\mathbb{Q}_{zl}^K + C_{16}\mathbb{Q}_{yl}^K\right)\\
\qquad - \sum_{l=1}^{4}\underline{\lambda}_y^{\beta(K,l)}\left(C_{12}\mathbb{Q}_{yl}^K + C_{14}\mathbb{Q}_{zl}^K + C_{16}\mathbb{Q}_{xl}^K\right) - \sum_{l=1}^{4}\underline{\lambda}_z^{\beta(K,l)}\left(C_{13}\mathbb{Q}_{zl}^K + C_{14}\mathbb{Q}_{yl}^K + C_{15}\mathbb{Q}_{xl}^K\right) = 0\\
i\omega\mathbb{M}^K\underline{\sigma}_{yy}^K + \left(C_{12}\mathbb{D}_x^K + C_{25}\mathbb{D}_z^K + C_{26}\mathbb{D}_y^K\right)\underline{v}_x^K + \left(C_{22}\mathbb{D}_y^K + C_{24}\mathbb{D}_z^K + C_{26}\mathbb{D}_x^K\right)\underline{v}_y^K\\
\qquad + \left(C_{23}\mathbb{D}_z^K + C_{24}\mathbb{D}_y^K + C_{25}\mathbb{D}_x^K\right)\underline{v}_z^K - \sum_{l=1}^{4}\underline{\lambda}_x^{\beta(K,l)}\left(C_{12}\mathbb{Q}_{xl}^K + C_{25}\mathbb{Q}_{zl}^K + C_{26}\mathbb{Q}_{yl}^K\right)\\
\qquad - \sum_{l=1}^{4}\underline{\lambda}_y^{\beta(K,l)}\left(C_{22}\mathbb{Q}_{yl}^K + C_{24}\mathbb{Q}_{zl}^K + C_{26}\mathbb{Q}_{xl}^K\right) - \sum_{l=1}^{4}\underline{\lambda}_z^{\beta(K,l)}\left(C_{23}\mathbb{Q}_{zl}^K + C_{24}\mathbb{Q}_{yl}^K + C_{25}\mathbb{Q}_{xl}^K\right) = 0\\
i\omega\mathbb{M}^K\underline{\sigma}_{zz}^K + \left(C_{13}\mathbb{D}_x^K + C_{35}\mathbb{D}_z^K + C_{36}\mathbb{D}_y^K\right)\underline{v}_x^K + \left(C_{23}\mathbb{D}_y^K + C_{34}\mathbb{D}_z^K + C_{36}\mathbb{D}_x^K\right)\underline{v}_y^K\\
\qquad + \left(C_{33}\mathbb{D}_z^K + C_{34}\mathbb{D}_y^K + C_{35}\mathbb{D}_x^K\right)\underline{v}_z^K - \sum_{l=1}^{4}\underline{\lambda}_x^{\beta(K,l)}\left(C_{13}\mathbb{Q}_{xl}^K + C_{35}\mathbb{Q}_{zl}^K + C_{36}\mathbb{Q}_{yl}^K\right)\\
\qquad - \sum_{l=1}^{4}\underline{\lambda}_y^{\beta(K,l)}\left(C_{23}\mathbb{Q}_{yl}^K + C_{34}\mathbb{Q}_{zl}^K + C_{36}\mathbb{Q}_{xl}^K\right) - \sum_{l=1}^{4}\underline{\lambda}_z^{\beta(K,l)}\left(C_{33}\mathbb{Q}_{zl}^K + C_{34}\mathbb{Q}_{yl}^K + C_{35}\mathbb{Q}_{xl}^K\right) = 0\\
i\omega\mathbb{M}^K\underline{\sigma}_{xy}^K + \left(C_{16}\mathbb{D}_x^K + C_{56}\mathbb{D}_z^K + C_{66}\mathbb{D}_y^K\right)\underline{v}_x^K + \left(C_{26}\mathbb{D}_y^K + C_{46}\mathbb{D}_z^K + C_{66}\mathbb{D}_x^K\right)\underline{v}_y^K\\
\qquad + \left(C_{36}\mathbb{D}_z^K + C_{46}\mathbb{D}_y^K + C_{56}\mathbb{D}_x^K\right)\underline{v}_z^K - \sum_{l=1}^{4}\underline{\lambda}_x^{\beta(K,l)}\left(C_{16}\mathbb{Q}_{xl}^K + C_{56}\mathbb{Q}_{zl}^K + C_{66}\mathbb{Q}_{yl}^K\right)\\
\qquad - \sum_{l=1}^{4}\underline{\lambda}_y^{\beta(K,l)}\left(C_{26}\mathbb{Q}_{yl}^K + C_{46}\mathbb{Q}_{zl}^K + C_{66}\mathbb{Q}_{xl}^K\right) - \sum_{l=1}^{4}\underline{\lambda}_z^{\beta(K,l)}\left(C_{36}\mathbb{Q}_{zl}^K + C_{46}\mathbb{Q}_{yl}^K + C_{56}\mathbb{Q}_{xl}^K\right) = 0\\
i\omega\mathbb{M}^K\underline{\sigma}_{xz}^K + \left(C_{15}\mathbb{D}_x^K + C_{55}\mathbb{D}_z^K + C_{56}\mathbb{D}_y^K\right)\underline{v}_x^K + \left(C_{25}\mathbb{D}_y^K + C_{45}\mathbb{D}_z^K + C_{56}\mathbb{D}_x^K\right)\underline{v}_y^K\\
\qquad + \left(C_{35}\mathbb{D}_z^K + C_{45}\mathbb{D}_y^K + C_{55}\mathbb{D}_x^K\right)\underline{v}_z^K - \sum_{l=1}^{4}\underline{\lambda}_x^{\beta(K,l)}\left(C_{15}\mathbb{Q}_{xl}^K + C_{55}\mathbb{Q}_{zl}^K + C_{56}\mathbb{Q}_{yl}^K\right)\\
\qquad - \sum_{l=1}^{4}\underline{\lambda}_y^{\beta(K,l)}\left(C_{25}\mathbb{Q}_{yl}^K + C_{45}\mathbb{Q}_{zl}^K + C_{56}\mathbb{Q}_{xl}^K\right) - \sum_{l=1}^{4}\underline{\lambda}_z^{\beta(K,l)}\left(C_{35}\mathbb{Q}_{zl}^K + C_{45}\mathbb{Q}_{yl}^K + C_{55}\mathbb{Q}_{xl}^K\right) = 0\\
i\omega\mathbb{M}^K\underline{\sigma}_{yz}^K + \left(C_{14}\mathbb{D}_x^K + C_{45}\mathbb{D}_z^K + C_{46}\mathbb{D}_y^K\right)\underline{v}_x^K + \left(C_{24}\mathbb{D}_y^K + C_{44}\mathbb{D}_z^K + C_{46}\mathbb{D}_x^K\right)\underline{v}_y^K\\
\qquad + \left(C_{34}\mathbb{D}_z^K + C_{44}\mathbb{D}_y^K + C_{45}\mathbb{D}_x^K\right)\underline{v}_z^K - \sum_{l=1}^{4}\underline{\lambda}_x^{\beta(K,l)}\left(C_{14}\mathbb{Q}_{xl}^K + C_{45}\mathbb{Q}_{zl}^K + C_{46}\mathbb{Q}_{yl}^K\right)\\
\qquad - \sum_{l=1}^{4}\underline{\lambda}_y^{\beta(K,l)}\left(C_{24}\mathbb{Q}_{yl}^K + C_{44}\mathbb{Q}_{zl}^K + C_{46}\mathbb{Q}_{xl}^K\right) - \sum_{l=1}^{4}\underline{\lambda}_z^{\beta(K,l)}\left(C_{34}\mathbb{Q}_{zl}^K + C_{44}\mathbb{Q}_{yl}^K + C_{45}\mathbb{Q}_{xl}^K\right) = 0
\end{cases}
$$

with matrices $\mathbb{M}^K, \mathbb{D}^K, \mathbb{E}^K, \mathbb{F}^K$ and $\mathbb{Q}^K$ defined previously.

The linear local system on an element K is written similarly to the isotropic case

$$\mathbb{A}^K \underline{W}^K + \mathbb{C}^K \underline{\Lambda}^K = 0, \tag{5.4.15}$$

with

$$\underline{W}^K = \left[\underline{v}_x^K, \ \underline{v}_y^K, \ \underline{v}_z^K, \ \underline{\sigma}_{xx}^K, \ \underline{\sigma}_{yy}^K, \ \underline{\sigma}_{zz}^K, \ \underline{\sigma}_{xy}^K, \ \underline{\sigma}_{xz}^K, \ \underline{\sigma}_{yz}^K\right]^T,$$

$$\underline{\Lambda}^K = \left[\underline{\lambda}_x, \ \underline{\lambda}_y, \ \underline{\lambda}_z\right]^T, \ \text{with } \underline{\lambda}_x = \left[\underline{\lambda}_x^{\beta(K,1)}, \ \underline{\lambda}_x^{\beta(K,2)}, \ \underline{\lambda}_x^{\beta(K,3)}, \ \underline{\lambda}_x^{\beta(K,4)}\right],$$
$$\underline{\lambda}_y = \left[\underline{\lambda}_y^{\beta(K,1)}, \ \underline{\lambda}_y^{\beta(K,2)}, \ \underline{\lambda}_y^{\beta(K,3)}, \ \underline{\lambda}_y^{\beta(K,4)}\right],$$
$$\underline{\lambda}_z = \left[\underline{\lambda}_z^{\beta(K,1)}, \ \underline{\lambda}_z^{\beta(K,2)}, \ \underline{\lambda}_z^{\beta(K,3)}, \ \underline{\lambda}_z^{\beta(K,4)}\right],$$

Matrices $\mathbb{A}^K$ and $\mathbb{C}^K$ change to

$$\mathbb{A}^K = \left[\mathbb{A}_1^K \quad \mathbb{A}_2^K \quad \mathbb{A}_3^K \quad \mathbb{A}_4^K \quad \mathbb{A}_5^K \quad \mathbb{A}_6^K \quad \mathbb{A}_7^K \quad \mathbb{A}_8^K \quad \mathbb{A}_9^K\right],$$

where

$$\begin{aligned}
\mathbb{A}_1^K = \Bigg[& i\omega\rho\mathbb{M}^K + \sum_{l=1}^{4} \tau^{(K,l)} \mathbb{E}_l^K, \quad 0, \quad 0, \quad ... \\
& ..., \quad \left(C_{11}\mathbb{D}_x^K + C_{15}\mathbb{D}_z^K + C_{16}\mathbb{D}_y^K\right), \quad \left(C_{12}\mathbb{D}_x^K + C_{25}\mathbb{D}_z^K + C_{26}\mathbb{D}_y^K\right), \quad ... \\
& ..., \quad \left(C_{13}\mathbb{D}_x^K + C_{35}\mathbb{D}_z^K + C_{36}\mathbb{D}_y^K\right), \quad \left(C_{16}\mathbb{D}_x^K + C_{56}\mathbb{D}_z^K + C_{66}\mathbb{D}_y^K\right), \quad ... \\
& ..., \quad \left(C_{15}\mathbb{D}_x^K + C_{55}\mathbb{D}_z^K + C_{56}\mathbb{D}_y^K\right), \quad \left(C_{14}\mathbb{D}_x^K + C_{45}\mathbb{D}_z^K + C_{46}\mathbb{D}_y^K\right)\Bigg]^T,
\end{aligned}$$

$$\begin{aligned}
\mathbb{A}_2^K = \Bigg[& 0, \quad i\omega\rho\mathbb{M}^K + \sum_{l=1}^{4} \tau^{(K,l)} \mathbb{E}_l^K, \quad 0, \quad ... \\
& ..., \quad \left(C_{12}\mathbb{D}_y^K + C_{14}\mathbb{D}_z^K + C_{16}\mathbb{D}_x^K\right), \quad \left(C_{22}\mathbb{D}_y^K + C_{24}\mathbb{D}_z^K + C_{26}\mathbb{D}_x^K\right), \quad ... \\
& ..., \quad \left(C_{23}\mathbb{D}_y^K + C_{34}\mathbb{D}_z^K + C_{36}\mathbb{D}_x^K\right), \quad \left(C_{26}\mathbb{D}_y^K + C_{46}\mathbb{D}_z^K + C_{66}\mathbb{D}_x^K\right), \quad ... \\
& ..., \quad \left(C_{25}\mathbb{D}_y^K + C_{45}\mathbb{D}_z^K + C_{56}\mathbb{D}_x^K\right), \quad \left(C_{24}\mathbb{D}_y^K + C_{44}\mathbb{D}_z^K + C_{46}\mathbb{D}_x^K\right)\Bigg]^T,
\end{aligned}$$

$$\begin{aligned}
\mathbb{A}_3^K = \Bigg[& 0, \quad 0, \quad i\omega\rho\mathbb{M}^K + \sum_{l=1}^{4} \tau^{(K,l)} \mathbb{E}_l^K, \quad ... \\
& ..., \quad \left(C_{13}\mathbb{D}_y^K + C_{14}\mathbb{D}_z^K + C_{16}\mathbb{D}_x^K\right), \quad \left(C_{23}\mathbb{D}_z^K + C_{24}\mathbb{D}_y^K + C_{25}\mathbb{D}_x^K\right), \quad ... \\
& ..., \quad \left(C_{33}\mathbb{D}_z^K + C_{34}\mathbb{D}_y^K + C_{35}\mathbb{D}_x^K\right), \quad \left(C_{36}\mathbb{D}_z^K + C_{46}\mathbb{D}_y^K + C_{56}\mathbb{D}_x^K\right), \quad ... \\
& ..., \quad \left(C_{35}\mathbb{D}_z^K + C_{45}\mathbb{D}_y^K + C_{55}\mathbb{D}_x^K\right), \quad \left(C_{34}\mathbb{D}_z^K + C_{44}\mathbb{D}_y^K + C_{45}\mathbb{D}_x^K\right)\Bigg]^T,
\end{aligned}$$

$$\mathbb{A}_4^K = \left[-\mathbb{D}_x^{K^T}, \ 0, \ 0, \ i\omega\mathbb{M}^K, \ 0, \ 0, \ 0, \ 0, \ 0\right]^T,$$

$$\mathbb{A}_5^K = \left[0, \ -\mathbb{D}_y^{K^T}, \ 0, \ 0, \ i\omega\mathbb{M}^K, \ 0, \ 0, \ 0, \ 0\right]^T,$$

$$\mathbb{A}_6^K = \left[0, \ 0, \ -\mathbb{D}_z^{K^T}, \ 0, \ 0, \ i\omega\mathbb{M}^K, \ 0, \ 0, \ 0\right]^T,$$

$$\mathbb{A}_7^K = \left[-\mathbb{D}_y^{K^T}, \ -\mathbb{D}_x^{K^T}, \ 0, \ 0, \ 0, \ 0, \ i\omega\mathbb{M}^K, \ 0, \ 0\right]^T,$$

$$\mathbb{A}_8^K = \left[-\mathbb{D}_z^{K^T},\ \ 0,\ \ -\mathbb{D}_x^{K^T},\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ i\omega\mathbb{M}^K,\ \ 0\right]^T,$$

$$\mathbb{A}_9^K = \left[0,\ \ -\mathbb{D}_z^{K^T},\ \ -\mathbb{D}_y^{K^T},\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ i\omega\mathbb{M}^K\right]^T.$$

$$\mathbb{C}^K = -\left[\mathbb{C}_x^K \quad \mathbb{C}_y^K \quad \mathbb{C}_z^K\right],$$

where

$$\mathbb{C}_x^K = \left[C_{x,1}^K,\ \ C_{x,2}^K,\ \ C_{x,3}^K,\ \ C_{x,4}^K\right],\ \text{ with }\ C_{x,l}^K = \begin{bmatrix} \tau^{(K,l)}\mathbb{F}_l^K \\ 0 \\ 0 \\ \left(C_{11}\mathbb{Q}_{xl}^K + C_{15}\mathbb{Q}_{zl}^K + C_{16}\mathbb{Q}_{yl}^K\right) \\ \left(C_{12}\mathbb{Q}_{xl}^K + C_{25}\mathbb{Q}_{zl}^K + C_{26}\mathbb{Q}_{yl}^K\right) \\ \left(C_{13}\mathbb{Q}_{xl}^K + C_{35}\mathbb{Q}_{zl}^K + C_{36}\mathbb{Q}_{yl}^K\right) \\ \left(C_{16}\mathbb{Q}_{xl}^K + C_{56}\mathbb{Q}_{zl}^K + C_{66}\mathbb{Q}_{yl}^K\right) \\ \left(C_{15}\mathbb{Q}_{xl}^K + C_{55}\mathbb{Q}_{zl}^K + C_{56}\mathbb{Q}_{yl}^K\right) \\ \left(C_{14}\mathbb{Q}_{xl}^K + C_{45}\mathbb{Q}_{zl}^K + C_{46}\mathbb{Q}_{yl}^K\right) \end{bmatrix},\ l = 1,2,3,4;$$

$$\mathbb{C}_y^K = \left[C_{y,1}^K,\ \ C_{y,2}^K,\ \ C_{y,3}^K,\ \ C_{y,4}^K\right],\ \text{ with }\ C_{y,l}^K = \begin{bmatrix} 0 \\ \tau^{(K,l)}\mathbb{F}_l^K \\ 0 \\ \left(C_{12}\mathbb{Q}_{yl}^K + C_{14}\mathbb{Q}_{zl}^K + C_{16}\mathbb{Q}_{xl}^K\right) \\ \left(C_{22}\mathbb{Q}_{yl}^K + C_{24}\mathbb{Q}_{zl}^K + C_{26}\mathbb{Q}_{xl}^K\right) \\ \left(C_{23}\mathbb{Q}_{yl}^K + C_{34}\mathbb{Q}_{zl}^K + C_{36}\mathbb{Q}_{xl}^K\right) \\ \left(C_{26}\mathbb{Q}_{yl}^K + C_{46}\mathbb{Q}_{zl}^K + C_{66}\mathbb{Q}_{xl}^K\right) \\ \left(C_{25}\mathbb{Q}_{yl}^K + C_{45}\mathbb{Q}_{zl}^K + C_{56}\mathbb{Q}_{xl}^K\right) \\ \left(C_{24}\mathbb{Q}_{yl}^K + C_{44}\mathbb{Q}_{zl}^K + C_{46}\mathbb{Q}_{xl}^K\right) \end{bmatrix},\ l = 1,2,3,4;$$

$$\mathbb{C}_z^K = \left[C_{z,1}^K,\ \ C_{z,2}^K,\ \ C_{z,3}^K,\ \ C_{z,4}^K\right],\ \text{ with }\ C_{z,l}^K = \begin{bmatrix} 0 \\ 0 \\ \tau^{(K,l)}\mathbb{F}_l^K \\ \left(C_{13}\mathbb{Q}_{zl}^K + C_{14}\mathbb{Q}_{yl}^K + C_{16}\mathbb{Q}_{xl}^K\right) \\ \left(C_{23}\mathbb{Q}_{zl}^K + C_{24}\mathbb{Q}_{yl}^K + C_{25}\mathbb{Q}_{xl}^K\right) \\ \left(C_{33}\mathbb{Q}_{zl}^K + C_{34}\mathbb{Q}_{yl}^K + C_{35}\mathbb{Q}_{xl}^K\right) \\ \left(C_{36}\mathbb{Q}_{zl}^K + C_{46}\mathbb{Q}_{yl}^K + C_{56}\mathbb{Q}_{xl}^K\right) \\ \left(C_{35}\mathbb{Q}_{zl}^K + C_{45}\mathbb{Q}_{yl}^K + C_{55}\mathbb{Q}_{xl}^K\right) \\ \left(C_{34}\mathbb{Q}_{zl}^K + C_{44}\mathbb{Q}_{yl}^K + C_{45}\mathbb{Q}_{xl}^K\right) \end{bmatrix},\ l = 1,2,3,4.$$

The transmission condition on a face $F_j = \partial K^+ \cap \partial K^-$ is unchanged and its local discretization reads as

$$\mathbb{B}^K \underline{W}^K + \mathbb{L}^K \underline{\Lambda}^K + \mathcal{R}^K = 0.$$

Matrices $\mathbb{B}^K$ and $\mathbb{L}^K$ and the vector $\mathcal{R}^K$ are the same than in the isotropic case.

Finally considering a source term non-equal to zero and using the trace space operator $\mathcal{A}_{HDG}$, we obtain a similar global system as (5.4.12) in 3D and as (3.3.13) in 2D

$$\sum_{K\in\mathcal{T}_h} (\mathcal{A}_{HDG}^K)^T \left[-\mathbb{B}^K(\mathbb{A}^K)^{-1}\mathbb{C}^K + \mathbb{L}^K\right] \mathcal{A}_{HDG}^K \underline{\Lambda} = \sum_{K\in\mathcal{T}_h} -(\mathcal{A}_{HDG}^K)^T \mathbb{B}^K(\mathbb{A}^K)^{-1}\mathbb{S}^K.$$

5.4.3 Boundary conditions

The discretization of the boundary conditions in 3D is similar to the discretization presented in section 3.3.3. We remind the boundary conditions (3.1.4) and (3.1.6)

- Free surface condition over Γ_l : $\quad \underline{\underline{\sigma}} \cdot \mathbf{n} = 0$,
- Absorbing boundary condition over Γ_a : $\underline{\underline{\sigma}} \cdot \mathbf{n} - PA(\theta', \phi')P^T \mathbf{v} = 0$.

As in 2D, the boundary conditions are included in the last equation of the global formulation (5.3.6) and modify the expression of the transmission condition

$$\begin{aligned}\sum_{F\in\mathcal{F}_h\setminus\Gamma_l\cup\Gamma_a}\int_F\left(\widehat{\underline{\underline{\sigma}}}_h\cdot\mathbf{n}\right)\cdot\eta+\sum_{F\in\Gamma_l}\int_F\left(\widehat{\underline{\underline{\sigma}}}_h\cdot\mathbf{n}\right)\cdot\eta+\sum_{F\in\Gamma_a}\int_F\left(\widehat{\underline{\underline{\sigma}}}_h\cdot\mathbf{n}-PA(\theta',\phi')P^T\lambda_h\right)\cdot\eta\\ =\sum_{F\in\Gamma_l}\int_F g_l\cdot\eta+\sum_{F\in\Gamma_a}\int_F g_a\cdot\eta.\end{aligned} \tag{5.4.16}$$

where in our case, $g_l = 0$ and $g_a = 0$. Replacing $\widehat{\underline{\underline{\sigma}}}_h \cdot \mathbf{n}$ by the expression (3.2.10), we obtain

$$\sum_{F\in\mathcal{F}_h}\int_F\left(\underline{\underline{\sigma}}_h\cdot\mathbf{n}\right)\cdot\eta-\sum_{F\in\mathcal{F}_h}\int_F\mathbf{S}\left(\mathbf{v}_h-\lambda_h\right)\cdot\eta-\sum_{F\in\Gamma_a}\int_F\left(PA(\theta',\phi')P^T\lambda_h\right)\cdot\eta=0. \tag{5.4.17}$$

Taking now into account the boundary conditions in the transmission condition, the HDG formulation is written:
find $(\mathbf{v}_h, \underline{\underline{\sigma}}_h, \lambda_h) \in \mathbf{V}_h^p \times \mathbf{\Sigma}_h^p \times \mathbf{M}_h^0$ such as: $\forall (\mathbf{w}, \underline{\underline{\xi}}, \eta) \in \mathbf{V}^p(K) \times \mathbf{\Sigma}^p(K) \times \mathbf{M}_h^0$,

$$\left\{\begin{array}{rcl}\displaystyle\sum_{K\in\mathcal{T}_h}\int_K i\omega\rho_K\mathbf{v}_h\cdot\mathbf{w}-\sum_{K\in\mathcal{T}_h}\int_K\left(\nabla\cdot\underline{\underline{\sigma}}_h\right)\cdot\mathbf{w}+\sum_{K\in\mathcal{T}_h}\int_{\partial K}\mathbf{S}\left(\mathbf{v}_h-\lambda_h\right)\cdot\mathbf{w} &=& 0,\\ \displaystyle\sum_{K\in\mathcal{T}_h}\int_K i\omega\underline{\underline{\sigma}}_h:\underline{\underline{\xi}}+\sum_{K\in\mathcal{T}_h}\int_K\mathbf{v}_h\cdot\nabla\cdot\left(\underline{\underline{C}}_K\underline{\underline{\xi}}\right)-\sum_{K\in\mathcal{T}_h}\int_{\partial K}\lambda_h\cdot\underline{\underline{C}}_K\underline{\underline{\xi}}\cdot\mathbf{n} &=& \displaystyle\sum_{K\in\mathcal{T}_h}\int_K\mathbf{f}:\underline{\underline{\xi}},\\ \displaystyle\sum_{F\in\mathcal{F}_h}\int_F\left(\underline{\underline{\sigma}}_h\cdot\mathbf{n}\right)\cdot\eta-\sum_{F\in\mathcal{F}_h}\int_F\mathbf{S}\left(\mathbf{v}_h-\lambda_h\right)\cdot\eta-\sum_{F\in\Gamma_a}\int_F\left(PA(\theta',\phi')P^T\lambda_h\right)\cdot\eta &=& 0.\end{array}\right. \tag{5.4.18}$$

For the sake of clarity, we write $Bc = PA(\theta', \phi')P^T$ in the discretization of the last equation of (5.4.18) on the boundary Γ_a

$$\left\{\begin{array}{l}\mathbb{Q}_{xl}^{K^T}\underline{\sigma}_{xx}^K+\mathbb{Q}_{yl}^{K^T}\underline{\sigma}_{xy}^K+\mathbb{Q}_{zl}^{K^T}\underline{\sigma}_{xz}^K-\tau^{(K,l)}\mathbb{F}_l^{K^T}\underline{v}_x^K+\tau^{(K,l)}\mathbb{G}^j\underline{\lambda}_x^{\beta(K,l)}-Bc_{11}\mathbb{G}^F\underline{\lambda}_x^{\beta(k,l)}\\ \qquad\qquad -Bc_{12}\mathbb{G}^F\underline{\lambda}_y^{\beta(k,l)}-Bc_{13}\mathbb{G}^F\underline{\lambda}_z^{\beta(k,l)}=0,\\ \mathbb{Q}_{xl}^{K^T}\underline{\sigma}_{xy}^K+\mathbb{Q}_{yl}^{K^T}\underline{\sigma}_{yy}^K+\mathbb{Q}_{zl}^{K^T}\underline{\sigma}_{yz}^K-\tau^{(K,l)}\mathbb{F}_l^{K^T}\underline{v}_y^K+\tau^{(K,l)}\mathbb{G}^j\underline{\lambda}_y^{\beta(K,l)}-Bc_{21}\mathbb{G}^F\underline{\lambda}_x^{\beta(k,l)}\\ \qquad\qquad -Bc_{22}\mathbb{G}^F\underline{\lambda}_y^{\beta(k,l)}-Bc_{23}\mathbb{G}^F\underline{\lambda}_z^{\beta(k,l)}=0,\\ \mathbb{Q}_{xl}^{K^T}\underline{\sigma}_{xz}^K+\mathbb{Q}_{yl}^{K^T}\underline{\sigma}_{yz}^K+\mathbb{Q}_{zl}^{K^T}\underline{\sigma}_{zz}^K-\tau^{(K,l)}\mathbb{F}_l^{K^T}\underline{v}_z^K+\tau^{(K,l)}\mathbb{G}^j\underline{\lambda}_z^{\beta(K,l)}-Bc_{31}\mathbb{G}^F\underline{\lambda}_x^{\beta(k,l)}\\ \qquad\qquad -Bc_{32}\mathbb{G}^F\underline{\lambda}_y^{\beta(k,l)}-Bc_{33}\mathbb{G}^F\underline{\lambda}_z^{\beta(k,l)}=0.\end{array}\right. \tag{5.4.19}$$

Then, for an element having an absorbing face, as in 2D, the inclusion of the boundary terms does not modify the expression of matrix $\mathbb{B}^K$ but affects only the values of some coefficients in matrix $\mathbb{L}^K$. For example if the face $l = 1$ of K is an absorbing face, $\mathbb{L}^K$ becomes

$$\mathbb{L}^K = \begin{bmatrix}\mathbb{L}_x^K & \mathbb{L}_y^K & \mathbb{L}_z^K\end{bmatrix},$$

$$\text{where } \mathbb{L}_x^K = \begin{bmatrix} \left(\tau^{(K,1)} - Bc_{11}^K\right)\mathbb{G}^{\beta(K,1)} & 0 & 0 & 0 \\ 0 & \tau^{(K,2)}\mathbb{G}^{\beta(K,2)} & 0 & 0 \\ 0 & 0 & \tau^{(K,3)}\mathbb{G}^{\beta(K,3)} & 0 \\ 0 & 0 & 0 & \tau^{(K,4)}\mathbb{G}^{\beta(K,4)} \\ -Bc_{21}^K\mathbb{G}^{\beta(K,1)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -Bc_{31}^K\mathbb{G}^{\beta(K,1)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbb{L}_y^K = \begin{bmatrix} -Bc_{12}^K\mathbb{G}^{\beta(K,1)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \left(\tau^{(K,1)} - Bc_{22}^K\right)\mathbb{G}^{\beta(K,1)} & 0 & 0 & 0 \\ 0 & \tau^{(K,2)}\mathbb{G}^{\beta(K,2)} & 0 & 0 \\ 0 & 0 & \tau^{(K,3)}\mathbb{G}^{\beta(K,3)} & 0 \\ 0 & 0 & 0 & \tau^{(K,4)}\mathbb{G}^{\beta(K,4)} \\ -Bc_{32}^K\mathbb{G}^{\beta(K,1)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbb{L}_z^K = \begin{bmatrix} -Bc_{13}^K\mathbb{G}^{\beta(K,1)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -Bc_{23}^K\mathbb{G}^{\beta(K,1)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \left(\tau^{(K,1)} - Bc_{33}^K\right)\mathbb{G}^{\beta(K,1)} & 0 & 0 & 0 \\ 0 & \tau^{(K,2)}\mathbb{G}^{\beta(K,2)} & 0 & 0 \\ 0 & 0 & \tau^{(K,3)}\mathbb{G}^{\beta(K,3)} & 0 \\ 0 & 0 & 0 & \tau^{(K,4)}\mathbb{G}^{\beta(K,4)} \end{bmatrix}.$$

5.5 Numerical results

This section is devoted to a preliminary numerical study of the performances of the proposed HDG method in 3D. We first consider the simple test problem of a plane wave propagating in an homogeneous medium. Then, we consider a geophysics test case the Epati model.
The 3D HDG algorithm has been implemented in a Fortran 90 software. As for the 2D algorithm, the sparse direct solver MUMPS is used for the resolution of the linear system. The simulations are performed on a computer system whose characteristics are:

- 2 x 12-core Haswell
- Frequency : 2,50 GHz

- Intel® Xeon® E5-2680 v3
- Cache L3: 30 Mb
- RAM: 128 Gb
- Infiniband QDR : 40Gb/s
- Ethernet : 10Gb/s
- AVX2

5.5.1 Plane wave propagation in an homogeneous medium

This analytical solution of this problem is the 3D extension of the problem of the propagation of a plane wave in an homogeneous medium treated in chapter 2, 3 and 4. For this study, the computational domain Ω considered is a plate (fig. 5.5.1). The characteristics of the medium are a mass density ρ= 1 kg.m^{-3} and Lamé's coefficients $\lambda_L =$ and $\mu =$ involving a velocity v_p=4000 m.s^{-1} and a velocity v_s=2000 m.s^{-1}. On the boundaries we impose an absorbing condition such that the exact solution is a plane wave. We choose a plane wave which propagates in the (x, y) plane

$$U = \nabla e^{i(k_x \cos\theta x + k_y \sin\theta y)} = \begin{pmatrix} V_{x0} \\ V_{y0} \\ V_{z0} \\ \sigma_{xx0} \\ \sigma_{yy0} \\ \sigma_{zz0} \\ \sigma_{xy0} \\ \sigma_{xz0} \\ \sigma_{yz0} \end{pmatrix} e^{i(k_x \cos\theta x + k_y \sin\theta y)},$$

where $k = \sqrt{k_x^2 + k_y^2 + k_z^2} = \frac{\omega}{v_p}$ is the wavenumber. The incidence angle θ is equal to 0° and the frequency f is equal to 2 Hz, and thus the angular frequency $\omega \simeq 12,56$. If we choose arbitrarly V_{x0}, we can express the other components as

$$\begin{cases} V_{y0} = \dfrac{k_x k_y (\lambda + \mu)}{\rho\omega^2 - k_x^2\mu - k_y^2(\lambda + 2\mu)} V_{x0}, \\ V_{z0} = \dfrac{k_x k_z (\lambda + \mu)}{\rho\omega^2 - k_x^2\mu - k_z^2(\lambda + 2\mu)} V_{x0}, \\ \sigma_{xx0} = \dfrac{-1}{\omega} \left(k_x (\lambda + 2\mu) V_{x0} + \lambda k_y V_{y0} + \lambda k_z V_{z0}\right), \\ \sigma_{yy0} = \dfrac{-1}{\omega} \left(k_y (\lambda + 2\mu) V_{y0} + \lambda k_x V_{x0} + \lambda k_z V_{z0}\right), \\ \sigma_{zz0} = \dfrac{-1}{\omega} \left(\lambda k_x V_{x0} + \lambda k_y V_{y0} + (\lambda + 2\mu) k_z V_{z0}\right), \\ \sigma_{xy0} = \dfrac{-\mu}{\omega} \left(k_y V_{x0} + k_x V_{y0}\right), \\ \sigma_{xz0} = \dfrac{-\mu}{\omega} \left(k_z V_{x0} + k_x V_{z0}\right), \\ \sigma_{yz0} = \dfrac{-\mu}{\omega} \left(k_z V_{y0} + k_y V_{z0}\right). \end{cases}$$

The computational domain Ω is discretized by three meshes whose characteristics are presented tab. 5.5.1.

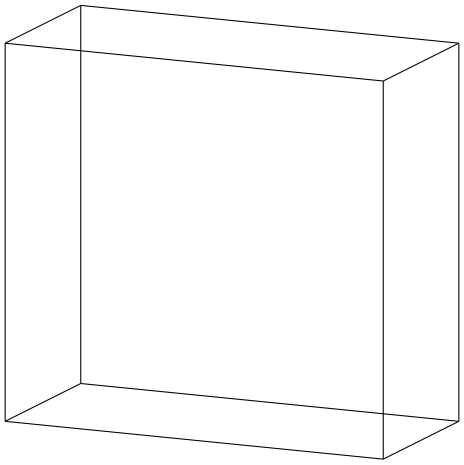


Fig. 5.5.1: Plane wave propagation in a 3D homogeneous medium: configuration of the computational domain Ω .

Mesh	# Mesh elements	# Mesh vertices	# Mesh faces	h_{min} (m)	h_{max} (m)	h_{max}/h_{min}
M1	5 150	1 350	11 300	292	955	3.27
M2	10 000	2 550	21 700	231	811	3.51
M3	15 600	3 600	33 500	156	709	4.54

Tab. 5.5.1: Plane wave propagation in a 3D homogeneous medium: characteristics of the three meshes

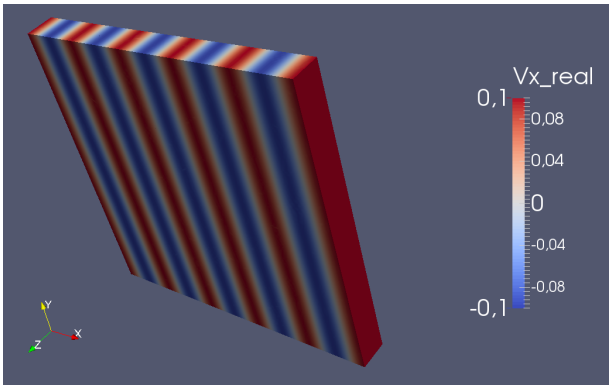
In order to validate the 3D HDG formulation, we present in tab. 5.5.2 the relative error on V_x and σ_{xx} components. The stabilization parameter has been taken equal to v_p^2, as we have observed in the 2D study that it seems to be an optimal value. We observe that for the interpolation orders considered, the numerical convergence on V_x is maintained, i.e. we converge with order $p+1$. For σ_{xx}, the numerical convergence appears to be suboptimal for the three meshes considered. This suboptimal convergence can be explained by the fact that in 3D, it is difficult to refine correctly the mesh. In our example, the finer mesh is not a refinement, strictly speaking, of the coarse mesh.

On fig. 5.5.2b we plot the numerical solution that we obtain with the HDG-$\mathbb{P}_3$ scheme used on the third mesh. A 2D view on the plan $z = 0$ is given on fig. 5.5.3b. We compare this solution to the exact one (fig. 5.5.2a and fig. 5.5.3a for the 2D view) on figs. 5.5.2c and 5.5.3c.

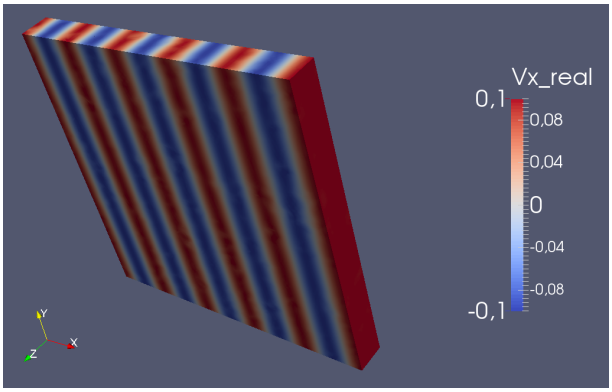
On tabs. 5.5.3, 5.5.4 and 5.5.5, we compare the 3D HDG computational performances to the ones of the 3D IPDG formulation. We remark that for $p = 1$ and $p = 2$, both methods have similar number of degrees of freedom and require the same memory space and computational time for the global system construction. For the resolution time, the HDG method starts to need less time for $p = 2$. The memory consumption begins to be more important with the IPDG scheme for $p = 3$. Moreover, from $p = 3$, we remark that the HDG global matrix construction requires more time than the IPDG global matrix, but taking into account the resolution time, the HDG method is more performant. Finally, the IPDG-$\mathbb{P}_4$ scheme has not been used for the second and third meshes because of high memory consumption.

# elements	Interpolation degree	Relative Error V_x (%)	Relative Error σ_{xx} (%)	Convergence order V_x	Convergence order σ_{xx}
5 150		28.7	45.97	-	-
10 000	1	20.07	36.1	2.0	1.5
15 600		15.97	30.5	1.93	1.3
5 150		6.4	8.75	-	-
10 000	2	2.9	5.0	4.8	3.4
15 600		2.2	4.0	2.1	1.7
5 150		7.2	7.8	-	-
10 000	3	3.19	3.85	4.98	4.3
15 600		2.4	3.18	3.5	1.4

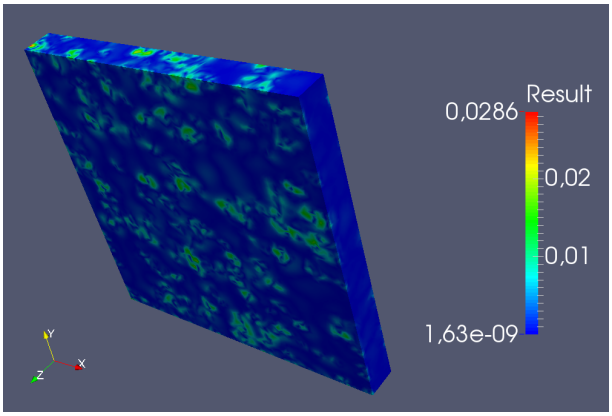
Tab. 5.5.2: Plane wave propagation in a 3D homogeneous medium: relative error on V_x and σ_{xx} and convergence order.



(a) Exact solution



(b) HDG-$\mathbb{P}_3$ solution

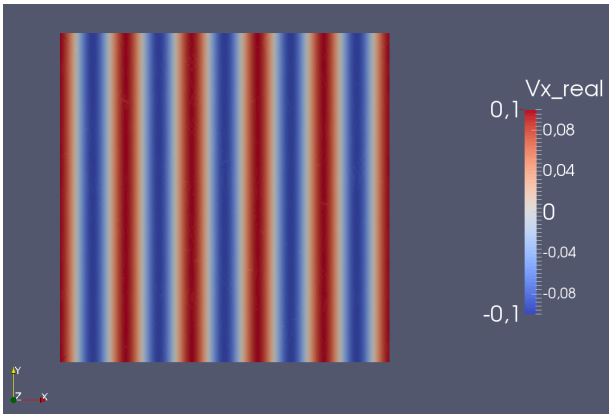


(c) Absolute error

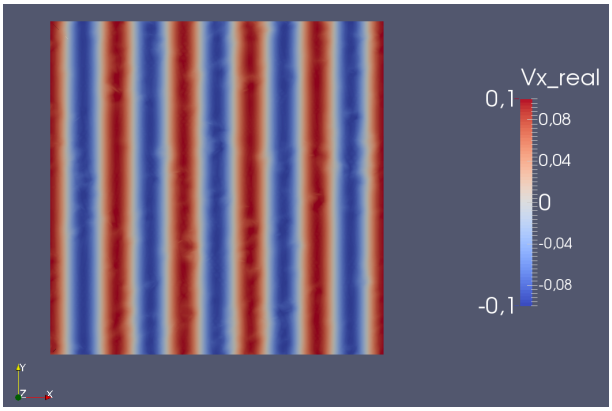
Fig. 5.5.2: Plane wave propagation in a 3D homogeneous medium: V_x component, mesh M3

5.5.2 Epati

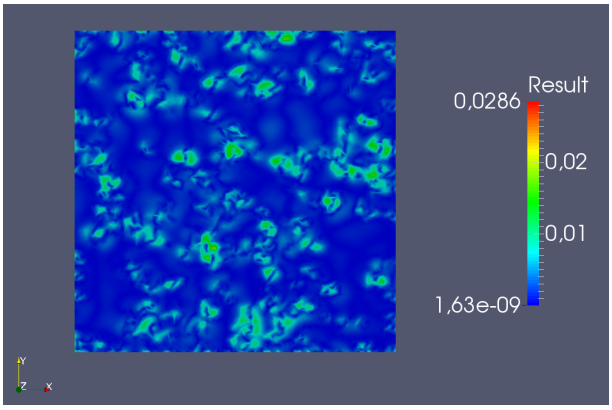
Finally, we test the 3D HDG formulation on a geophysical benchmark: the Epati test case. The configuration of the problem is presented fig. 5.5.4. We discretize the computational domain into two meshes whose characteristics are given on tab. 5.5.6.



(a) Exact solution



(b) HDG-$\mathbb{P}_3$ solution



(c) Absolute error

Fig. 5.5.3: Plane wave propagation in a 3D homogeneous medium: 2D view of V_x component, mesh M3

The numerical solution computed with the HDG-$\mathbb{P}_3$ is presented on figs. 5.5.5a to 5.5.5c. We compare the memory consumption of the HDG method to the one of a classical FE method, the classical Lagrange FE method (tab. 5.5.7). On the first mesh, the memory consumption of both methods is similar, but when we refine the mesh the HDG method requires more memory space than the FE method for a same interpolation order.

# Mesh elements	Interpolation degree	Non-zeros terms		# dof		fill rate (%)	
		HDG	IPDG	HDG	IPDG	HDG	IPDG
5 150	1	6.7e+06	3.4e+06	1.0e+05	6.2e+04	0.065	0.090
10 000		1.3e+07	6.7e+06	1.9e+05	1.2e+05	0.034	0.047
15 600		2.0e+07	1.1e+07	3.0e+05	1.9e+05	0.022	0.030
5 150	2	2.7e+07	2.1e+07	2.0e+05	1.5e+05	0.065	0.090
10 000		5.2e+07	4.2e+07	3.9e+05	3.0e+05	0.034	0.047
15 600		8.1e+07	6.6e+07	6.0e+05	4.7e+05	0.022	0.030
5 150	3	7.4e+07	8.6e+07	3.4e+05	3.1e+05	0.065	0.090
10 000		1.4e+08	1.7e+08	6.5e+05	6.0e+05	0.034	0.047
15 600		2.2e+08	2.7e+08	1.0e+06	9.4e+05	0.022	0.030
5 150	4	1.7e+08	2.6e+08	5.1e+05	5.4e+05	0.065	0.090
10 000		3.2e+08	-	9.7e+05	-	0.034	-
15 600		5.1e+08	-	1.5e+06	-	0.022	-

Tab. 5.5.3: Plane wave propagation in a 3D homogeneous medium: number of non-zero terms in the global matrix.

# Mesh elements	Interpolation degree	Memory (MB)		Memory ratio	
		HDG	IPDG	HDG	IPDG.
5 150	1	3.5e+03	3.2e+03	1	0.93
10 000		4.9e+03	4.4e+03	1	0.89
15 600		7.1e+03	6.3e+03	1	0.88
5 150	2	6.7e+03	7.5e+03	1	1.11
10 000		1.2e+04	1.4e+04	1	1.18
15 600		2.1e+04	2.6e+04	1	1.23
5 150	3	1.4e+04	2.3e+04	1	1.61
10 000		2.8e+04	4.9e+04	1	1.74
15 600		5.3e+04	9.8e+04	1	1.85
5 150	4	2.7e+04	8.5e+04	1	3.11
10 000		5.9e+04	-	1	-
15 600		1.2e+05	-	1	-

Tab. 5.5.4: Plane wave propagation in a 3D homogeneous medium: memory consumption.

Conclusion

In this last chapter, we have extended the HDG formulation to the 3D elastodynamics equations. The preliminary results on a simple test case that we have presented confirmed the numerical convergence of the method. They also show that when compared to IPDG method, the HDG method is still performant in memory and computational time. On a more realistic test case, we have compared the HDG method to classical FE method, and in this case, it seems to be less competitive. However in this preliminary 3D study, all the properties of the HDG scheme have not been used. For example, we have not test the p-adaptivity, which can improve the HDG costs by, as we have seen in 2D, reducing the computational costs to $p-1$ or $p-2$ computational costs and giving a solution with the same

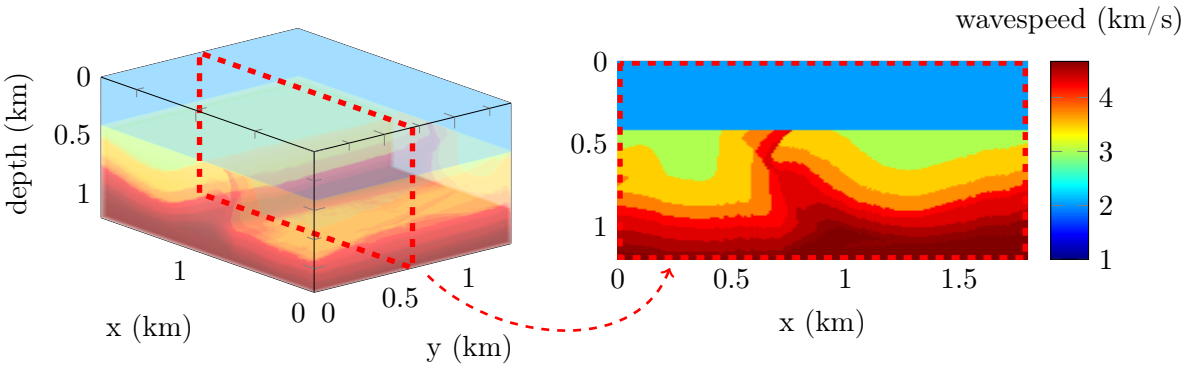
# Mesh elements	Interpolation degree	Construction time (s)		Solution time (s)	
		HDG	IPDG	HDG	IPDG
5 150	1	3.8	3.7	24.9	18.2
10 000		5.0	6.9	51.7	39.7
15 600		7.4	10.2	96.1	111.7
5 150	2	19.5	15.3	103	108
10 000		37.7	29.1	238	373
15 600		58.7	44.6	640	1172
5 150	3	135	51	388	769
10 000		229	99	976	2435
15 600		348	155	3245	5676
5 150	4	512	147	1313	9868
10 000		1008	286	3935	-
15 600		1395	-	14270	-

Tab. 5.5.5: Plane wave propagation in a 3D homogeneous medium: time required for the global matrix construction and for the system resolution.

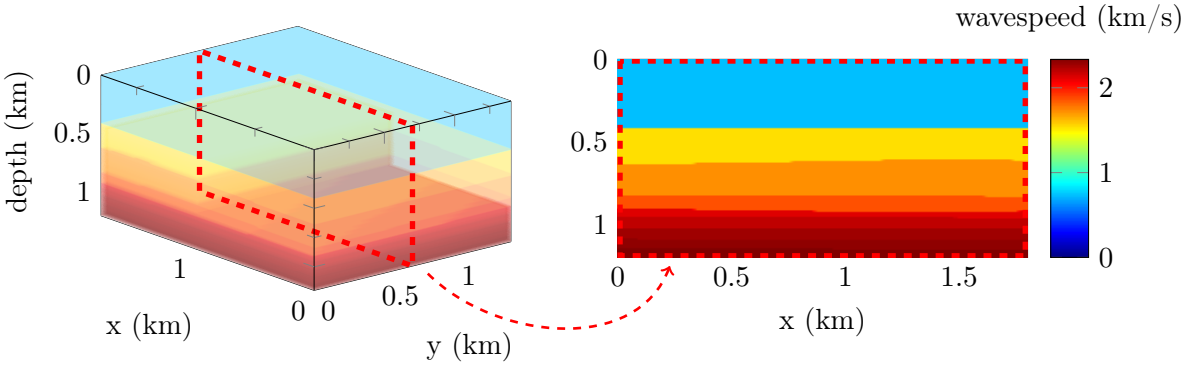
Mesh	# Mesh elements	# Mesh vertices	# Mesh faces
M1	500	150	1 100
M2	6 400	1 370	13 500

Tab. 5.5.6: Epati test case: characteristics of the two meshes.

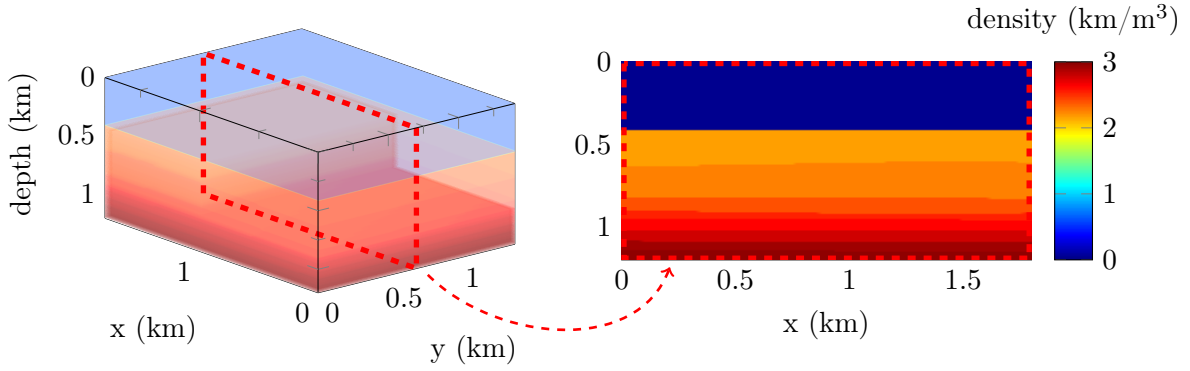
accuracy than a HDG-$\mathbb{P}_p$ solution.



(a) P-wavespeed.



(b) S-wavespeed.

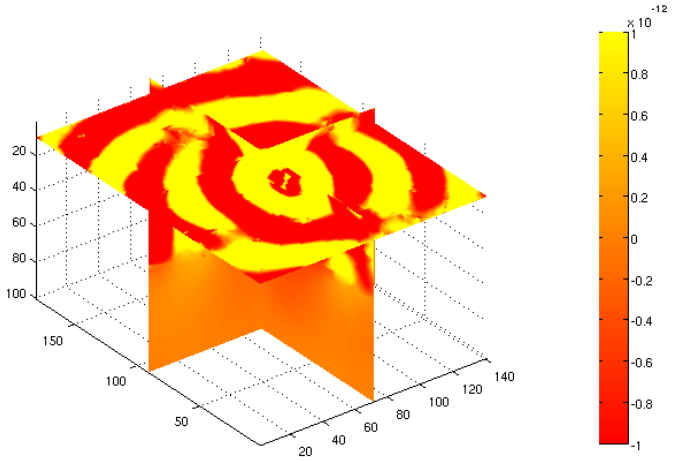


(c) Density.

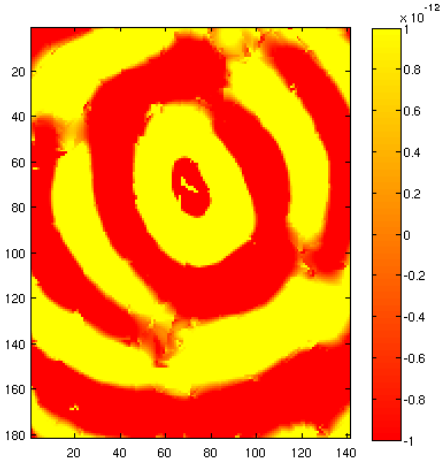
Fig. 5.5.4: Epati test case: Models of dimension $1.8 \times 1.4 \times 1.2$km. 3D representation (left) and vertical section at $y = 700$m (right).

# Mesh elements	Interpolation degree	# dof		Non-zero terms		Memory (MB)	
		HDG	FE	HDG	FE	HDG	FE
500	1	1.0e+04	3.5e+02	5.8e+05	1.1e+04	32	57
6 400		1.2e+05	3.6e+03	7.3e+06	1.4e+05	7124	332
500	2	2.0e+04	2.7e+03	2.3e+06	1.8e+05	125	311
6 400		2.4e+05	2.9e+04	2.7e+07	2.2e+06	1.7e+04	909
500	3	3.4e+04	8.3e+03	6.5e+06	1.0e+06	354	451
6 400		4.0e+05	9.4e+04	8.1e+07	1.3e+07	3.7e+04	3350
500	4	5.1e+04	1.9e+04	1.4e+07	3.7e+06	794	749
6 400		6.1e+05	2.2e+05	1.8e+08	4.6e+07	6.1e+04	1.0e+04
500	5	7.2e+04	3.5e+04	2.9e+07	1.0e+07	1613	1293
6 400		8.5e+05	4.2e+05	3.6e+08	1.3e+08	1.2e+05	2.4e+04

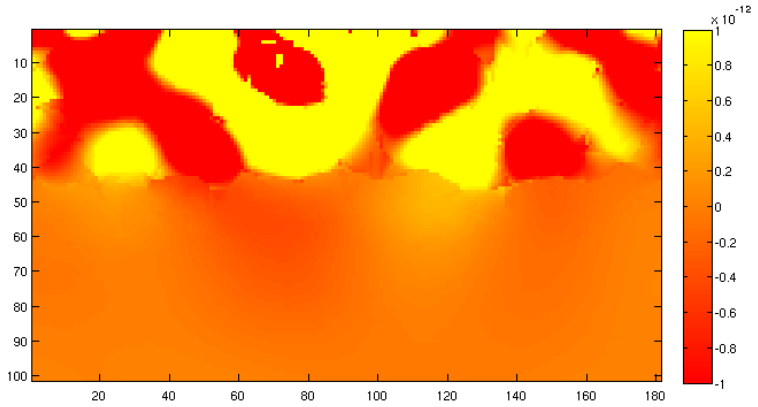
Tab. 5.5.7: Epati test case: number of degrees of freedom, non-zero terms in the global matrix and memory consumption.



(a)



(b) Horizontal section at $z = 10$ m.



(c) Vertical section at $y = 700$ m.

Fig. 5.5.5: Epati test case: HDG-$\mathbb{P}_3$ numerical solution on mesh M2, with a frequency $f = 2$.

General Conclusion

In this thesis we have proposed an hybridizable discontinuous Galerkin method for solving the anisotropic elastodynamics wave equations in harmonic domain, in two and three dimensions.
We have analyzed the performance of the proposed method in 2D on simple test case and compared it to classical DG methods. We have shown that the HDG method provides a more accurate solution for less computational cost provided that the order is high enough. The next step will be the comparison with classical FE methods in order to emphasize the efficiency of HDG.
We have also noted that, for scattering problems, the HDG scheme is more sensitive to the curvature of the obstacles, which could be resolved using curved elements.
We have illustrated the usefulness of the p-adaptivity in 2D, which allows to reach the accuracy of a global method of degree p for the costs of a global method of degree $p-1$ or $p-2$. This feature is already implemented in the 3D code. We now have to determine an accuracy criteria for assigning an order to a given cell, similar to the criteria we proposed in 2D.
For the numerical analysis of the scheme, we have shown that the HDG method could be rewritten as an upwind fluxes DG method and one of our perspectives is to use this equivalence in order to perform a dispersion analysis following the work of Ainsworth, Monk and Muniz [64].
We have observed the impact of the stabilization matrix on the accuracy of the HDG solution. In this thesis we have considered a diagonal matrix and it could be interesting to study another form for the stabilization matrix especially for the 3D case where we did not obtain an optimal convergence. Once this is done, we could extend the 3D to higher order and make a performance analysis similar to the 2D case.
Another improvement would to apply a post-processing to the Lagrange multiplier in order to increase the accuracy of the solution following the work of Cockburn, Guzman and Wang [74].
We have shown that HDG could be used for 2D simulation on geophysical benchmark, and we will now implement the method in a Reverse Time Migration software, the ultimate goal being to couple HDG method with a full waveform inversion solver. In order to tackle more realistic test cases in 3D, it will be mandatory to improve the linear solver and we are now considering the use of an hybrid solver such as Maphys developed by the INRIA team-project HIEPACS.
We plan on long-term to extend the HDG method to elasto-acoustic coupling in order to tackle offshore seismic imaging.

Conclusion Générale

Dans cette thèse, nous avons proposé une méthode de Galerkine discontinue hybride pour résoudre les équations d'ondes élastiques anisotropes en domaine fréquentiel, en deux et trois dimensions.
Nous avons analysé les performances de la méthode proposée en 2D sur un cas test simple et nous l'avons comparé à des méthodes DG classiques. Sur un maillage donné et pour un même ordre d'interpolation, la méthode HDG permet d'obtenir une solution aussi précise mais en utilisant des coûts de calcul plus faibles. La prochaine étape sera de la comparer avec des méthodes FE classiques dans le but de démontrer l'efficacité de la méthode HDG.
Nous avons aussi remarqué, pour les problèmes de diffraction, que le schéma HDG est plus sensible aux obstacles courbés, ce qui peut être résolu en utilisant des éléments courbes.
Nous avons illustré l'utilité de la p-adaptivité en 2D, qui permet d'obtenir la précision d'une méthode globale de degré p avec des coûts de calcul d'une méthode global de degré $p-1$ ou $p-2$. Cette caractéristique est déjà implémentée dans le code 3D. Nous devons maintenant déterminer un critère de précision pour assigner un ordre à une cellule donnée, de manière similaire au critère que nous avons proposé en 2D.
Au niveau de l'analyse numérique du schéma, nous avons montré que la méthode HDG pouvait être réécrite comme une méthode DG à flux décentrés et une de nos perspectives est d'utiliser cette équivalence dans le but d'effectuer une analyse de dispersion similaire aux travaux de Ainsworth, Monk et Muniz [64].
Nous avons observé l'impact de la matrice de stabilisation sur la précision de la solution HDG. Dans cette thèse nous avons considéré une matrice diagonale et il pourrait être intéressant d'étudier une autre forme de matrice de stabilisation, en particulier pour le cas 3D où nous n'avons pas obtenu une convergence optimale. Une fois cela fait, nous pourrons étendre le 3D aux ordres élevés et effectuer une analyse de performance identique à celle que nous avons fait pour le cas 2D.
Une autre amélioration serait d'appliquer un post-processing au multiplicateur de Lagrange afin d'améliorer la précision de la solution, comme cela est fait dans le travail de Cockburn, Guzman et Wang [74].
Nous avons montré que la méthode HDG pouvait être utilisée pour la simulation 2D de benchmarks géophysiques, et nous sommes actuellement en train de la mettre en oeuvre dans un logiciel de RTM, le but ultime étant de coupler la méthode HDG avec un solveur FWI. Dans le but de traiter des cas tests 3D plus réalistes, il devient indispensable d'améliorer le solveur linéaire et nous sommes actuellement en train de considérer l'utilisation d'un solveur hybride tel que Maphys développé par l'équipe-projet INRIA HIEPACS.
A long terme, nous souhaiterions étendre la méthode HDG au couplage élasto-acoustique afin de traiter de plus larges problèmes d'imagerie sismique.

Acknowledgements

This PhD. has been supported by the INRIA-TOTAL strategic action DIP (dip.inria.fr).

Experiments presented in this thesis were carried out using the PLAFRIM experimental testbed, being developed under the INRIA PlaFRIM development action with support from Bordeaux INP, LABRI and IMB and other entities: Conseil Régional d'Aquitaine, Université de Bordeaux and CNRS (and ANR in accordance to the programme d'investissements d'Avenir (see https://plafrim.bordeaux.inria.fr/)).

Bibliography

[1] B. L. Biondi. *Concepts and applications in 3D seismic imaging : 2007 Distinguished Instructor Short course*. Distinguished instructor series. Tulsa, Okla. Society of Exploration Geophysicists, 10, 2007, Tulsa, Okla., 2007. Sponsorisé par la SEG / EAGE.

[2] E. Baysal, D. D. Kosloff, and J. W. Sherwood. Reverse time migration. *Geophysics*, 48(11):1514–1524, 1983.

[3] N. D. Whitmore et al. Iterative depth migration by backward time propagation. In *1983 SEG Annual Meeting*. Society of Exploration Geophysicists, 1983.

[4] J. Virieux and S. Operto. *An overview of full-waveform inversion in exploration geophysics*, volume 74. Geophysics, November-December 2009.

[5] J. Brittan, J. Bai, H. Delome, C. Wong, and D. Yingst. *Full Waveform Inversion - the state of the art*, volume 31. First Break, October 2013.

[6] M. Kern. *Problèmes inverses: aspects numériques*. 2002-2003.

[7] E. Baysal and D. D. Kosloff. Forward modeling by a Fourier method. *Geophysics*, 47(10):1402–1412, 1982.

[8] R. H. Stolt. Migration by Fourier transform. *Geophysics*, 43(1):23–48, 1978.

[9] R. Brossier, S. Operto, and J. Virieux. Seismic imaging of complex onshore strucutres by 2D elastic frequency-domain full-waveform inversion. *Geophysics*, 74(6):WCC105–WCC118, 2009.

[10] Oliver G Ernst and Martin J Gander. Why it is difficult to solve helmholtz problems with classical iterative methods. In *Numerical analysis of multiscale problems*, pages 325–363. Springer, 2012.

[11] Mohamed El Bouajaji, Bertrand Thierry, Xavier Antoine, and Christophe Geuzaine. A quasi-optimal domain decomposition algorithm for the time-harmonic maxwell's equations. *Journal of Computational Physics*, 294:38–57, 2015.

[12] Yassine Boubendir, Xavier Antoine, and Christophe Geuzaine. A quasi-optimal non-overlapping domain decomposition algorithm for the helmholtz equation. *Journal of Computational Physics*, 231(2):262–280, 2012.

[13] B Stupfel and B Despres. A domain decomposition method for the solution of large electromagnetic scattering problems. *Journal of Electromagnetic waves and applications*, 13(11):1553–1568, 1999.

[14] Victorita Dolean, Martin J Gander, Stephane Lanteri, Jin-Fa Lee, and Zhen Peng. Effective transmission conditions for domain decomposition methods applied to the time-harmonic curl–curl maxwell's equations. *Journal of computational physics*, 280:232–247, 2015.

[15] Martin J Gander. Optimized schwarz methods. *SIAM Journal on Numerical Analysis*, 44(2):699–731, 2006.

[16] Alexandre Vion and Christophe Geuzaine. Double sweep preconditioner for optimized schwarz methods applied to the helmholtz problem. *Journal of Computational Physics*, 266:171–190, 2014.

[17] Martin J Gander, Frédéric Magoules, and Frédéric Nataf. Optimized schwarz methods without overlap for the helmholtz equation. *SIAM Journal on Scientific Computing*, 24(1):38–60, 2002.

[18] J. Virieux. P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method. *Geophysics*, 51:889–901, 1986.

[19] R. W. Graves. Simulating seismic wave propagation in 3D elastic media using staggered-grid finite differences. *Bulletin of the Seismological Society of America*, 86(4):1091–1106, 1996.

[20] J. O. A. Robertsson. A numerical free-surface condition for elastic/viscoelastic finite-difference modeling in the presence of topography. *Geophysics*, 61(6):1921–1934, 1996.

[21] T. Ohminato and B. A. Chouet. A free-surface boundary condition for including 3D topography in the finite-difference method. *Bulletin of the Seismological Society of America*, 87(2):494–515, 1997.

[22] K. Marfurt. Accuracy of finite-difference and finite-elements modeling of the scalar and elastic wave equation. *Geophysics*, 49:533–549, 1984.

[23] J.-D. Min, C. Shin, R. G. Pratt, and H.S. Yoo. Weighted-averaging finite-element method for 2d elastic wave equations in the frequency domain. *Bulletin of the Seismological Society of America*, 93:904–921, 2003.

[24] D. Komatitsch and J. Tromp. Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophys. J. Int.*, 139:806–822, 1999.

[25] J. D. De Basabe and M. K. Sen. New developments in the finite-element method for seismic modeling. *The Leading Edge*, 28(5):562–567, 2009.

[26] J. Virieux, V. Etienne, V. Cruz-Atienza, and et al. Modelling Seismic Wave Propagation for Geophysical Imaging. *Seismic Waves, Research and Analysis*, pages 253–304, 2012.

[27] V. Etienne, E. Chaljub, J. Virieux, and N. Glinsky. An hp-adaptive discontinuous Galerkin finite-element method for 3-D elastic wave modelling. *Geophysical Journal International*, 183(2):941–962, 2010.

[28] B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of discontinuous Galerkin, mixed and continuous Galerkin methods for second order elliptic problems. *SIAM Journal on Numerical Analysis*, 47:1319–1365, 2009.

[29] S. Lanteri, L. Li, and R. Perrussel. Numerical investigation of a high order hybridizable discontinuous Galerkin method for 2d time-harmonic Maxwell's equations. *COMPEL*, 32(3):1112–1138, 2013.

[30] N.C. Nguyen, J. Peraire, and B. Cockburn. Hybridizable discontinuous Galerkin methods for the time-harmonic Maxwell's equations. *Journal of Computational Physics*, 230:7151–7175, 2011.

[31] N. C. Nguyen, J. Peraire, and B. Cockburn. A hybridizable discontinuous Galerkin method for Stokes flow. *Comput. Methods Appl. Mech. Engrg.*, 199:582–597, 2010.

[32] N. C. Nguyen and J. Peraire. Hybridizable discontinuous Galerkin methods for partial differential equations in continuum mechanics. *Journal of Computational Physics*, (231):5955–5988, 2012.

[33] N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations. *Journal of Computational Physics*, 228:8841–8855, 2009.

[34] N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations. *Journal of Computational Physics*, 228:3232–3254, 2009.

[35] N. C. Nguyen, J. Peraire, and B. Cockburn. High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. *Journal of Computational Physics*, (230):3695–3718, 2011.

[36] L. Thomsen. Weak elastic anisotropy. *Geophysics*, 51(10):1954–1966, October 1986.

[37] L. Boillot. *Contributions à la modélisation mathématique et à l'algorithmique parallèle pour l'optimisation d'un propagateur d'ondes élastiques en milieu anisotrope*. PhD thesis, Université de Pau et des Pays de l'Adour, 2014.

[38] P. G. Ciarlet. *The finite element method for elliptic problems*, volume 40. Siam, 2002.

[39] T. J. R. Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[40] I. Babuška, F. Ihlenburg, E. T. Paik, and S. A. Sauter. A generalized finite element method for solving the Helmholtz equation in two dimensions with minimal pollution. *Computer Methods in Applied Mechanics and Engineering*, 128(3):325–359, 1995.

[41] I. M. Babuska and S. A. Sauter. Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? *SIAM review*, 42(3):451–484, 2000.

[42] K. Gerdes and F. Ihlenburg. On the pollution effect in FE solutions of the 3D-Helmholtz equation. *Computer Methods in Applied Mechanics and Engineering*, 170(1):155–172, 1999.

[43] D. Komatitsch and J.P. Vilotte. The spectral-element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures. *Bulletin of the Seismological Society of America*, 88(2):368–392, 1998.

[44] D. Komatitsch, J.-P. Vilotte, R. Vai, J. M. Castillo-Covarrubias, and F. J. Sanchez-Sesma. The spectral element method for elastic wave equations-application to 2-D and 3-D seismic problems. *International Journal for numerical methods in engineering*, 45(9):1139–1164, 1999.

[45] M. A. Taylor and B. A. Wingate. A generalized diagonal mass matrix spectral element method for non-quadrilateral elements. *Applied Numerical Mathematics*, 33(1):259–265, 2000.

[46] J. Shen, L.-L. Wang, and H. Li. A triangular spectral element method using fully tensorial rational basis functions. *SIAM Journal on Numerical Analysis*, 47(3):1619–1650, 2009.

[47] M. D. Samson, H. Li, and L.-L. Wang. A new triangular spectral element method I: implementation and analysis on a triangle. *Numerical Algorithms*, 64(3):519–547, 2013.

[48] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos National Laboratory, 1973.

[49] B. Cockburn. Discontinuous Galerkin methods. *ZAMM - Journal of Applied Mathematics and Mechanics*, 83(11):731–754, 2003.

[50] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779 (electronic), 2001/02.

[51] L. C. Wilcox, G. Stadler, C. Burstedde, and O. Ghattas. A high-order discontinuous Galerkin method for wave propagation through coupled elastic-acoustic media. *Journal of Computational Physics*, 229:9373–9396, 2010.

[52] M. Käser and M. Dumbser. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes - I. The two-dimensional isotropic case with external source terms. *Geophysical Journal International*, 166(2):855–877, 2006.

[53] M. Dumbser and M. Käser. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes - II; The three-dimensional isotropic case. *Geophysical Journal International*, 167:319–336, 2006.

[54] J. de la Puente, M. Käser, M. Dumbser, H. Igel, et al. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes - IV. Anisotropy. *Geophysical Journal International*, 169:1210–1228, 2007.

[55] S. Delcourte, L. Fezoui, and N. Glinsky-Olivier. A high order discontinuous Galerkin method for the seismic wave propagation. *ESAIM: Proceedings*, 27:70–89, May 2009.

[56] M.J. Grote, A. Schneebeli, and D. Schötzau. Discontinuous Galerkin finite element method for the wave equation. *SIAM J. Numer. Anal.*, 44(6):2408–2431 (electronic), 2006.

[57] C. Agut and J. Diaz. Stability analysis of the Interior Penalty Discontinuous Galerkin method for the wave equation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(3):903–932, 2013.

[58] R. Brossier. *Imagerie sismique à deux dimensions des milieux visco-élastiques par inversion des formes d'ondes: développements méthodologiques et applications.* PhD thesis, Université de Nice-Sophia Antipolis, November 2009.

[59] R. Brossier, V. Etienne, S. Operto, and J. Virieux. Frequency-domain Numerical Modelling of Visco-Acoustic Waves with Finite-Difference and Finite-Element Discontinuous Galerkin Methods. September 2010.

[60] S. Lanteri and M. El Bouajaji. High order discontinuous Galerkin method for the solution of 2D time-harmonic Maxwell's equations. *Applied Mathematics and Computation*, 219:7241–7251, 2013.

[61] P.R. Amestoy, I.S. Duff, and J.-Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computationnal Methods in Applied Mechanics and Engineering*, 184:501–520, 2000.

[62] J. Ahrens, B. Geveci, and C. Law. *ParaView: An End-User Tool for Large Data Visualization.* Number ISBN-13: 978-0123875822. Visualization Handbook, elsevier edition, 2005.

[63] Ayachit and Utkarsh. *The ParaView Guide: A Parallel Visualization Application.* Kitware, isbn 978-1930934306 edition, 2015.

[64] M. Ainsworth, P. Monk, and W. Muniz. Dispersive and Dissipative Properties of Discontinuous Galerkin Finite Element Methods for the Second-Order Wave Equation. *Journal of Scientific Computing*, 27(1-3):5–40, June 2006.

[65] B.M. Fraejis de Veubeke. Displacement and equilibrium models in the finite element method. In O. C. Zienkiewicz and G. S. Holister, editors, *Stress Analysis*, chapter 9, pages 145–197. Wiley, 1977.

[66] Raviart P. A. and J.-M. Thomas. A mixed finite element method for second order elliptic problems. In I. Galligani and E. Magenes, editors, *Mathematical Aspects of Finite Element Method*, volume 606 of *Lectures notes in Mathematics*, pages 292–315. Springer-Verlag, 1977.

[67] F. Brezzi, J. Douglas, and L. D. Marini. Two families of mixed finite elements for second order elliptic problems. *Numer. Math.*, 47:1267–1275, 1985.

[68] B. Cockburn and J.Gopalakrishnan. A characterization of hybridized mixed methods for second order elliptic problems. *SIAM J. Numer. Anal.*, 42(1):203–301, 2004.

[69] J. Carrero, B. Cockburn, and D. Schötzau. Hybridized, globally divergence-free LDG methds. Part I: The Stokes problem. *Math. Comp.*, 75:533–563, 2006.

[70] B. Cockburn and J.Gopalakrishnan. New hybridization techniques. *GAMM-Mitt.*, 2:154–183, 2005.

[71] G. Giorgiani. *Adaptative Hybrid Discontinuous Methods For Fluid and Wave Problems*. PhD thesis, April 2013.

[72] B. Cockburn, J.Gopalakrishnan, and F.-J. Sayas. A projection-based error analysis of HDG methods. *Math. Comp.*, 79:1351–1367, 2010.

[73] R. Griesmaier and P. Monk. Error analysis for a Hybridizable Discontinuous Galerkin method for the Helmholtz equation. *J. Sc. Computing*, 49:291–310, 2011.

[74] B. Cockburn, B.Dong, and J. Guzman. A superconvergent LDG-hybridizable Galerkin method for second order elliptic problems. *Math. Comp.*, 77:1887–1916, 2008.

[75] R. M. Kirby and S. J. Sherwin. To CG or to HDG: A comparative study. *Journal of Scientific Computing*, 51:183–212, May 2011.

[76] C. Farhat, I. Harari, and U. Hetmaniuk. A discontinuous Galerkin method with Lagrange multipliers for the solution of Helmholtz problems in the mid-frequency regime. *Comput. Methods Appl. Mech. Engrg.*, 192:1389–1419, 2003.

[77] G. Benitez Alvarez, A. F. Dourado Loula, E. Gomes Dutra do Carmo, and F. Alves Rochinha. A discontinuous finite element formulation for Helmholtz equation. *Computer Methods in Applied Mechanics and Engineering*, 195(33):4018–4035, 2006.

[78] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM journal on numerical analysis*, 19(4):742–760, 1982.

[79] B. Cockburn, J. Guzman, and H. Wang. Superconvergent discontinuous Galerkin methods for second-order elliptic problems. *Math. Comp.*, 78:1–24, 2009.

[80] D.N. Arnold and F. Brezzi. Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates. *RAIRO Modélisation mathématique et analyse numérique*, 19(1):7–32, 1985.

A Analytical expression of the plane wave in an homogeneous medium

We remind that the incident field of a plane wave reads as:

$$U = \begin{pmatrix} V_{x0} \\ V_{z0} \\ \sigma_{xx0} \\ \sigma_{zz0} \\ \sigma_{xz0} \end{pmatrix} e^{-i(xk_x+zk_z)} \tag{A.1}$$

U is solution of the wave equation (2.1.1). Replacing $(v_x, v_z, \sigma_{xx}, \sigma_{zz}, \sigma_{xz})^T$ in system (2.1.5), we obtain:

$$\begin{cases} i\omega V_{x0}e^{-i(xk_x+zk_z)} &= -\dfrac{i}{\rho}(k_x\sigma_{xx0} + k_z\sigma_{xz0})e^{-i(xk_x+zk_z)} \\ i\omega V_{z0}e^{-i(xk_x+zk_z)} &= -\dfrac{i}{\rho}(k_x\sigma_{xz0} + k_z\sigma_{zz0})e^{-i(xk_x+zk_z)} \\ i\omega\sigma_{xx0}e^{-i(xk_x+zk_z)} &= -i[(\lambda+2\mu)k_xV_{x0} + \lambda k_zV_{z0}]e^{-i(xk_x+zk_z)} \\ i\omega\sigma_{zz0}e^{-i(xk_x+zk_z)} &= -i[\lambda k_xV_{x0} + (\lambda+2\mu)k_zV_{z0}]e^{-i(xk_x+zk_z)} \\ i\omega\sigma_{xz0}e^{-i(xk_x+zk_z)} &= -i\mu(k_xV_{z0} + k_zV_{x0})e^{-i(xk_x+zk_z)} \end{cases} \tag{A.2}$$

If we replace σ_{xx0}, σ_{zz0} and σ_{xz0} in the two first equations by their expressions given by the three last, we get:

$$\begin{cases} \rho\omega^2V_{x0} &= V_{x0}[k_x^2(\lambda+2\mu) + \mu k_z^2] + V_{z0}k_xk_z(\lambda+\mu) \\ \rho\omega^2V_{z0} &= V_{z0}[k_z^2(\lambda+2\mu) + \mu k_x^2] + V_{xz0}k_xk_z(\lambda+\mu) \end{cases} \tag{A.3}$$

And so:

$$\begin{cases} V_{z0} &= \dfrac{k_zk_x(\lambda+\mu)}{\rho\omega^2 - k_x^2\mu - k_z^2(\lambda+2\mu)}V_{x0} \\ \rho\omega^2V_{x0} &= [\dfrac{(k_zk_x(\lambda+\mu))^2}{\rho\omega^2 - k_z^2(\lambda+2\mu) - k_x^2\mu} + k_x^2(\lambda+2\mu) + k_z^2\mu]V_{x0} \end{cases} \tag{A.4}$$

According to the second equation of the above system, we have:

$$(k_x^2+k_z^2)^2(c_pc_s\rho)^2 - \rho^2\omega^2(c_p^2+c_s^2)(k_x^2+k_z^2) + \rho^2\omega^4 = 0 \tag{A.5}$$

$$k_x^2+k_z^2 = \begin{cases} (\dfrac{\omega}{c_p})^2 \\ (\dfrac{\omega}{c_s})^2 \end{cases} \longleftrightarrow \begin{cases} k_x = \dfrac{\omega}{c_p}\cos(\theta) \\ k_z = \dfrac{\omega}{c_p}\sin(\theta) \end{cases} \text{ or } \begin{cases} k_x = \dfrac{\omega}{c_s}\cos(\theta) \\ k_z = \dfrac{\omega}{c_s}\sin(\theta) \end{cases} \tag{A.6}$$

Consequently, we can choose arbitrarily V_{x0} and compute the others components of U.

B Analytical solution of the disk-shaped scatterer problem

In this section, we recall the analytical expression of the solution of the scattering of a plane wave by an elastic disk-shaped configuration. This analytical solution is expressed in the form of Fourier series. In the case of an infinite solid domain, the total displacement field u can be expressed using two others displacement fields u^1 and u^2 as

$$u = u^1 + u^2. \tag{B.1}$$

Each of these displacement fields u^j, $j = 1, 2$, can be written with the help of two potentials ϕ^j and ψ^j

$$u^1 = \nabla\phi^1 + (-\mathbf{e_z}) \times \nabla\psi^1, \tag{B.2}$$
$$u^2 = \nabla\phi^2 + (-\mathbf{e_z}) \times \nabla\psi^2, \tag{B.3}$$

where

$$\phi^1 = \sum_{n=0}^{+\infty} A_n^1 H_n^{(1)}(k_p r)\cos(n\theta), \qquad \psi^1 = \sum_{n=0}^{+\infty} A_n^2 H_n^{(1)}(k_s r)\sin(n\theta),$$
$$\phi^2 = \sum_{n=0}^{+\infty} A_n^3 H_n^{(2)}(k_p r)\cos(n\theta), \qquad \psi^2 = \sum_{n=0}^{+\infty} A_n^4 H_n^{(2)}(k_s r)\sin(n\theta),$$

and $\mathbf{e_z}$ is the third vector of the cartesian basis. $H_n^{(1)}$ and $H_n^{(2)}$ respectively represent Hankel's functions of first and second kind, defined such as

$$\begin{aligned} H_n^{(1)}(x) &= J_n(x) + iY_n(x) \\ H_n^{(2)}(x) &= J_n(x) - iY_n(x). \end{aligned} \tag{B.4}$$

The Hankel function of second kind is the conjugate of the Hankel function of first kind. J_n and Y_n are respectively Bessel's functions of first and second kind. $k_p = \dfrac{\omega}{v_p}$ is the $P-$wave number and $k_s = \dfrac{\omega}{v_s}$ the $S-$wave number. In polar coordinates, we have

$$\nabla\phi^j = \frac{\partial\phi^j}{\partial r}\mathbf{e_r} + \frac{1}{r}\frac{\partial\phi^j}{\partial\theta}\mathbf{e}_\theta, \qquad j = 1, 2,$$

$$(-\mathbf{e_z}) \times \nabla\psi^j = \frac{1}{r}\frac{\partial\psi^j}{\partial\theta}\mathbf{e_r} - \frac{\partial\psi^j}{\partial r}\mathbf{e}_\theta, \qquad j = 1, 2.$$

This allows us to write

$$u^j = u_r^j \mathbf{e_r} + u_\theta^j \mathbf{e}_\theta, \tag{B.5}$$
$$u = \left(u_r^1 + u_r^2\right)\mathbf{e_r} + \left(u_\theta^1 + u_\theta^2\right)\mathbf{e}_\theta, \tag{B.6}$$

where $\quad u_r^j = \dfrac{\partial\phi^j}{\partial r} + \dfrac{1}{r}\dfrac{\partial\psi^j}{\partial\theta} \quad$ and $\quad u_\theta^j = \dfrac{1}{r}\dfrac{\partial\phi^j}{\partial\theta} - \dfrac{\partial\psi^j}{\partial r}.$

Since the polar basis vectors $\mathbf{e_r}$ and $\mathbf{e}_\theta$ are given in the cartesian basis by $(\cos\theta, \sin\theta)^t$ and $(-\sin\theta, \cos\theta)^t$ respectively, it follows that the components of the displacement field can be expressed in the cartesian basis as

$$\left\{ \begin{array}{rcccl} u_x &=& u_r\cos\theta - u_\theta\sin\theta &=& (u_r^1 + u_r^2)\cos\theta - (u_\theta^1 + u_\theta^2)\sin\theta, \\ u_z &=& u_r\sin\theta + u_\theta\cos\theta &=& (u_r^1 + u_r^2)\sin\theta + (u_\theta^1 + u_\theta^2)\cos\theta, \end{array} \right. \tag{B.7}$$

with

$$u_r^1 = \sum_{n=0}^{+\infty} \left[A_n^1 k_p H_n^{(1)'}(k_p r) + \frac{n}{r} A_n^2 H_n^{(1)}(k_s r) \right] \cos(n\theta),$$

$$u_\theta^1 = \sum_{n=0}^{+\infty} \left[-A_n^1 \frac{n}{r} H_n^{(1)}(k_p r) - A_n^2 k_s H_n^{(1)'}(k_s r) \right] \sin(n\theta),$$

$$u_r^2 = \sum_{n=0}^{+\infty} \left[A_n^3 k_p H_n^{(2)'}(k_p r) + \frac{n}{r} A_n^4 H_n^{(2)}(k_s r) \right] \cos(n\theta),$$

$$u_\theta^2 = \sum_{n=0}^{+\infty} \left[-A_n^3 \frac{n}{r} H_n^{(2)}(k_p r) - A_n^4 k_s H_n^{(2)'}(k_s r) \right] \sin(n\theta).$$

Knowing that derivatives of Hankel's functions are determined by one of the two recurrence relations

$$H_n^{(j)'}(x) = \begin{cases} -H_{n+1}^{(j)}(x) & \text{for } n = 0, \\ H_{n-1}^{(j)}(x) - \dfrac{n}{x} H_n^{(j)}(x) & \text{for } n > 0, \end{cases}$$

or

$$H_n^{(j)'}(x) = \begin{cases} -H_{n+1}^{(j)}(x) & \text{for } n = 0, \\ -H_{n+1}^{(j)}(x) + \dfrac{n}{x} H_n^{(j)}(x) & \text{for } n > 0, \end{cases}$$

we choose to write arbitrarily

$$H_n^{(1)'}(x) = \begin{cases} -H_{n+1}^{(1)}(x) & \text{for } n = 0, \\ -H_{n+1}^{(1)}(x) + \dfrac{n}{x} H_n^{(1)}(x) & \text{for } n > 0, \end{cases}$$

and

$$H_n^{(2)'}(x) = \begin{cases} -H_{n+1}^{(2)}(x) & \text{for } n = 0, \\ H_{n-1}^{(2)}(x) - \dfrac{n}{x} H_n^{(2)}(x) & \text{for } n > 0. \end{cases}$$

Finally this leads to

$$\begin{aligned} u_r &= \sum_{n=0}^{+\infty} \left[A_n^1 k_p H_n^{(1)'}(k_p r) + A_n^2 \frac{n}{r} H_n^{(1)}(k_s r) + A_n^3 k_p H_n^{(2)'}(k_p r) + A_n^4 \frac{n}{r} H_n^{(2)}(k_s r) \right] \cos(n\theta) \\ u_\theta &= \sum_{n=0}^{+\infty} \left[-A_n^1 \frac{n}{r} H_n^{(1)}(k_p r) - A_n^2 k_s H_n^{(1)'}(k_s r) - A_n^3 \frac{n}{r} H_n^{(2)}(k_p r) - A_n^4 k_s H_n^{(2)'}(k_s r) \right] \sin(n\theta). \end{aligned} \tag{B.8}$$

In order to determine coefficients A_n^1, A_n^2, A_n^3 and A_n^4, we have to use the boundary conditions. We get

$$u = -u^{inc} \qquad \text{on } \Gamma_a, \tag{B.9}$$

$$\underline{\underline{\sigma}}\mathbf{n} = v_p \left(\mathbf{v} \cdot \mathbf{n} \right) \mathbf{n} + v_s \left(\mathbf{v} \cdot \mathbf{t} \right) \mathbf{t} \qquad \text{on } \Gamma_b. \tag{B.10}$$

We recall the reader to refer to figure 4.3.2 for the definition of the boundaries Γ_a and Γ_b. u^{inc} is the incident wave which is written as

$$u^{inc} = \nabla \phi^{inc} + (-\mathbf{e_z}) \times \nabla \psi^{inc},$$

with $\psi^{inc} = 0$ and

$$\phi^{inc} = \sum_{n=0}^{+\infty} \varepsilon_n i^n J_n(k_p r) \cos(n\theta)$$

where $\varepsilon_n = \begin{cases} 1 & \text{if } n = 0, \\ 2 & \text{if } n \geq 1, \end{cases}$. And finally we write

$$\begin{aligned} u^{inc} &= \frac{\partial \phi^{inc}}{\partial r}\mathbf{e_r} + \frac{1}{r}\frac{\partial \phi^{inc}}{\partial \theta}\mathbf{e}_\theta \\ &= \sum_{n=0}^{+\infty} \varepsilon_n i^n \cos(n\theta) k_p J_n'(k_p r)\mathbf{e_r} + \sum_{n=0}^{+\infty} \frac{-n}{r}\varepsilon_n i^n J_n(k_p r)\sin(n\theta)\mathbf{e}_\theta. \end{aligned} \tag{B.11}$$

If we develop the equation

$$\underline{\underline{\sigma}}\mathbf{n} = v_p\left(\mathbf{v}\cdot\mathbf{n}\right)\mathbf{n} + v_s\left(\mathbf{v}\cdot\mathbf{t}\right)\mathbf{t},$$

in polar coordinates, we find

$$\begin{aligned} \sigma_{rr} &= i\omega\rho v_p u_r, \\ \sigma_{r\theta} &= i\omega\rho v_s u_\theta. \end{aligned} \tag{B.12}$$

σ_{rr} and $\sigma_{r\theta}$ can be expressed in terms of potentials ψ^j et ϕ^j

$$\begin{aligned} \sigma_{rr} &= \sigma_{rr}^1 + \sigma_{rr}^2 = \sum_{j=1,2} \sigma_{rr}^{\phi^j} + \sigma_{rr}^{\psi^j}, \\ \sigma_{r\theta} &= \sigma_{r\theta}^1 + \sigma_{r\theta}^2 = \sum_{j=1,2} \sigma_{r\theta}^{\phi^j} + \sigma_{r\theta}^{\psi^j}, \end{aligned} \tag{B.13}$$

where

$$\begin{aligned} \sigma_{rr}^{\phi^j} &= \lambda\Delta\phi^j + 2\mu\frac{\partial^2\phi^j}{\partial r^2}, \\ \sigma_{rr}^{\psi^j} &= 2\mu\left[\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial\psi^j}{\partial\theta}\right)\right], \\ \sigma_{r\theta}^{\phi^j} &= 2\mu\left(\frac{1}{r}\frac{\partial^2\phi^j}{\partial\theta\partial r} - \frac{1}{r^2}\frac{\partial\phi^j}{\partial\theta}\right), \\ \sigma_{r\theta}^{\psi^j} &= \mu\left[\frac{1}{r^2}\frac{\partial^2\psi^j}{\partial\theta^2} - r\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial\psi^j}{\partial r}\right)\right]. \end{aligned}$$

where ϕ^j satisfies the Helmholtz equation

$$\Delta\phi^j = -k_p^2\phi^j.$$

This yields

$$\Delta\phi^j = -k_p^2\sum_{n=0}^{+\infty} A_n^h H_n^{(j)}(k_p r)\cos(n\theta), \quad h = 1 \text{ or } 3.$$

Then, by some calculations, we obtain

$$\begin{aligned}
\frac{\partial^2\phi^j}{\partial r^2} &= \sum_{n=0}^{+\infty} A_n^h k_p^2 H_n^{(j)''}(k_p r)\cos(n\theta),\\
\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial\psi^j}{\partial\theta}\right) &= -\frac{1}{r^2}\frac{\partial\psi^j}{\partial\theta}+\frac{1}{r}\frac{\partial^2\psi^j}{\partial r\partial\theta}\\
&= -\frac{1}{r^2}\sum_{n=0}^{+\infty} A_n^l H_n^{(j)}(k_s r)(n\cos(n\theta))\\
&+ \tfrac{1}{r}\textstyle\sum_{n=0}^{+\infty} n A_n^l k_s H_n^{(j)'}(k_s r)\cos(n\theta), \quad l=2 \text{ ou } 4,\\
\frac{\partial^2\phi^j}{\partial\theta\partial r} &= \sum_{n=0}^{+\infty} A_n^h k_p H_n^{(j)'}(k_p r)\left(-n\sin(n\theta)\right),\\
\frac{\partial\phi^j}{\partial\theta} &= \sum_{n=0}^{+\infty} A_n^h H_n^{(j)}(k_p r)\left(-n\sin(n\theta)\right),\\
\frac{\partial^2\psi^j}{\partial\theta^2} &= \sum_{n=0}^{+\infty} A_n^l H_n^{(j)}(k_s r)(-n^2\sin(n\theta)),\\
\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial\psi^j}{\partial r}\right) &= -\frac{1}{r^2}\frac{\partial\psi^j}{\partial r}+\frac{1}{r}\frac{\partial^2\psi^j}{\partial r^2}\\
&= -\frac{1}{r^2}\sum_{n=0}^{+\infty} A_n^l k_s H_n^{(j)'}(k_s r)\sin(n\theta)+\frac{1}{r}\sum_{n=0}^{+\infty} A_n^l k_s^2 H_n^{(j)''}(k_s r)\sin(n\theta).
\end{aligned}$$

In summary, for $\underline{\underline{\sigma}}^1$, we get

$$\left\{\begin{aligned}
\sigma_{rr}^1 &= \sum_{n=0}^{+\infty}\Big[A_n^1 k_p^2\left(-\lambda H_n^{(1)}(k_p r)+2\mu H_n^{(1)''}(k_p r)\right)+\\
&\qquad A_n^2\frac{2\mu n}{r}\left(-\frac{1}{r}H_n^{(1)}(k_s r)+k_s H_n^{(1)'}(k_s r)\right)\Big]\cos(n\theta),\\
\sigma_{r\theta}^1 &= \mu\sum_{n=0}^{+\infty}\Big[-2A_n^1\frac{n}{r}\left(k_p H_n^{(1)'}(k_p r)-\frac{1}{r}H_n^{(1)}(k_p r)\right)+\\
&\qquad A_n^2\left(-\frac{n^2}{r^2}H_n^{(1)}(k_s r)+\frac{1}{r}k_s H_n^{(1)'}(k_s r)-k_s^2 H_n^{(1)''}(k_s r)\right)\Big]\sin(n\theta).
\end{aligned}\right. \tag{B.14}$$

In addition, since $k_p^2(\lambda+2\mu)=k_s^2\mu$, we obtain $\lambda=\dfrac{k_s^2}{k_p^2}\mu-2\mu$. Moreover, we remark that

$$\begin{aligned}
H_n^{(1)''}(k_p r) &= -\frac{1}{k_p r}H_n^{(1)'}(k_p r)-(1-\frac{n^2}{(k_p r)^2})H_n^{(1)}(k_p r)\\
&= \frac{1}{k_p r}H_{n+1}^{(1)}(k_p r)-\frac{n}{(k_p r)^2}H_n^{(1)}(k_p r)-(1-\frac{n^2}{(k_p r)^2})H_n^{(1)}(k_p r).
\end{aligned}$$

It follows that

$$\left\{\begin{aligned}
\sigma_{rr}^1 &= \sum_{n=0}^{+\infty}\Big[A_n^1\frac{2\mu}{r^2}\left(\left(n^2-n-\frac{1}{2}k_p^2 r^2\right)H_n^{(1)}(k_p r)+k_p r H_{n+1}^{(1)}(k_p r)\right)+\\
&\qquad A_n^2\frac{2\mu n}{r^2}\left((n-1)H_n^{(1)}(k_s r)-k_s r H_{n+1}^{(1)}(k_p r)\right)\Big]\cos(n\theta),\\
\sigma_{r\theta}^1 &= \sum_{n=0}^{+\infty}\Big[-A_n^1\frac{2\mu n}{r^2}\left((n-1)H_n^{(1)}(k_p r)-k_p r H_{n+1}^{(1)}(k_p r)\right)-\\
&\qquad A_n^2\frac{2\mu}{r^2}\left(\left(n^2-n-\frac{1}{2}k_s^2 r^2\right)H_n^{(1)}(k_s r)+k_s r H_{n+1}^{(1)}(k_s r)\right)\Big]\sin(n\theta).
\end{aligned}\right. \tag{B.15}$$

We can do the same for $\underline{\underline{\sigma}}^2$ and we obtain

$$\left\{\begin{array}{rcl}\sigma_{rr}^2 &=& \displaystyle\sum_{n=0}^{+\infty}\Big[-A_n^3\mu\left(\left(k_s^2-2k_p^2\right)H_n^{(2)}(k_pr)+2k_p^2H_n^{(2)''}(k_pr)\right)+\\ &&\qquad 2A_n^4\mu\frac{n}{r}\left(\frac{-1}{r}H_n^{(2)}(k_sr)+k_sH_n^{(2)'}(k_sr)\right)\Big]\cos(n\theta),\\ \sigma_{r\theta}^2 &=& \displaystyle\sum_{n=0}^{+\infty}\Big[-2A_n^3\mu\frac{n}{r}\left(k_pH_n^{(2)'}(k_pr)-\frac{1}{r}H_n^{(2)}(k_pr)\right)+\\ &&\qquad A_n^4\mu\left(-\frac{n^2}{r^2}H_n^{(2)}(k_sr)+\frac{1}{r}k_sH_n^{(2)'}(k_sr)-k_s^2H_n^{(2)''}(k_sr)\right)\Big]\sin(n\theta).\end{array}\right. \tag{B.16}$$

If we sum these two systems, we finally get for $\underline{\underline{\sigma}}$

$$\left\{\begin{array}{rcl}\sigma_{rr} &=& \displaystyle\sum_{n=0}^{+\infty}\Big[A_n^1\frac{2\mu}{r^2}\left(\left(n^2-n-\frac{1}{2}k_p^2r^2\right)H_n^{(1)}(k_pr)+k_prH_{n+1}^{(1)}(k_pr)\right)+\\ &&\qquad A_n^2\frac{2\mu n}{r^2}\left((n-1)H_n^{(1)}(k_sr)-k_srH_{n+1}^{(1)}(k_pr)\right)-\\ &&\qquad A_n^3\mu\left(\left(k_s^2-2k_p^2\right)H_n^{(2)}(k_pr)+2k_p^2H_n^{(2)''}(k_pr)\right)+\\ &&\qquad 2A_n^4\mu\frac{n}{r}\left(\frac{-1}{r}H_n^{(2)}(k_sr)+k_sH_n^{(2)'}(k_sr)\right)\Big]\cos(n\theta),\\ \sigma_{r\theta} &=& \displaystyle\sum_{n=0}^{+\infty}\Big[-A_n^1\frac{2\mu n}{r^2}\left((n-1)H_n^{(1)}(k_pr)-k_prH_{n+1}^{(1)}(k_pr)\right)-\\ &&\qquad A_n^2\frac{2\mu}{r^2}\left(\left(n^2-n-\frac{1}{2}k_s^2r^2\right)H_n^{(1)}(k_sr)+k_srH_{n+1}^{(1)}(k_sr)\right)-\\ &&\qquad 2A_n^3\mu\frac{n}{r}\left(k_pH_n^{(2)'}(k_pr)-\frac{1}{r}H_n^{(2)}(k_pr)\right)+\\ &&\qquad A_n^4\mu\left(-\frac{n^2}{r^2}H_n^{(2)}(k_sr)+\frac{1}{r}k_sH_n^{(2)'}(k_sr)-k_s^2H_n^{(2)''}(k_sr)\right)\Big]\sin(n\theta).\end{array}\right. \tag{B.17}$$

To summarize, using boundary conditions, we get

- At $r=a$,

$$\sum_{n=0}^{+\infty}\left[A_n^1k_pH_n^{(1)'}(k_pr)+A_n^2\frac{n}{r}H_n^{(1)}(k_sr)+A_n^3k_pH_n^{(2)'}(k_pr)+A_n^4\frac{n}{r}H_n^{(2)}(k_sr)\right]\cos(n\theta)=-\sum_{n=0}^{+\infty}\varepsilon_ni^nk_pJ_n'(k_pr)\cos(n\theta), \tag{B.18}$$

$$\sum_{n=0}^{+\infty}\left[-A_n^1\frac{n}{r}H_n^{(1)}(k_pr)-A_n^2k_sH_n^{(1)'}(k_sr)-A_n^3\frac{n}{r}H_n^{(2)}(k_pr)-A_n^4k_sH_n^{(2)'}(k_sr)\right]\sin(n\theta)=\sum_{n=0}^{+\infty}\frac{n}{r}\varepsilon_ni^nJ_n(k_pr)\sin(n\theta). \tag{B.19}$$

- At $r = b$,

$$\begin{aligned}
&\sum_{n=0}^{+\infty}\Bigg[A_n^1\frac{2\mu}{r^2}\left(\left(n^2-n-\frac{1}{2}k_p^2r^2\right)H_n^{(1)}(k_pr)+k_prH_{n+1}^{(1)}(k_pr)\right)+\\
&\qquad A_n^2\frac{2\mu n}{r^2}\left((n-1)\,H_n^{(1)}(k_sr)-k_srH_{n+1}^{(1)}(k_pr)\right)-\\
&\qquad A_n^3\mu\left(\left(k_s^2-2k_p^2\right)H_n^{(2)}(k_pr)+2k_p^2H_n^{(2)''}(k_pr)\right)+\\
&\qquad 2A_n^4\mu\frac{n}{r}\left(\frac{-1}{r}H_n^{(2)}(k_sr)+k_sH_n^{(2)'}(k_sr)\right)\Bigg]\cos(n\theta)=\\
&i\omega\rho v_p\sum_{n=0}^{+\infty}\left[A_n^1k_pH_n^{(1)'}(k_pr)+A_n^2\frac{n}{r}H_n^{(1)}(k_sr)+A_n^3k_pH_n^{(2)'}(k_pr)+A_n^4\frac{n}{r}H_n^{(2)}(k_sr)\right]\cos(n\theta),
\end{aligned}\tag{B.20}$$

$$\begin{aligned}
&\sum_{n=0}^{+\infty}\Bigg[-A_n^1\frac{2\mu n}{r^2}\left((n-1)\,H_n^{(1)}(k_pr)-k_prH_{n+1}^{(1)}(k_pr)\right)-\\
&\qquad A_n^2\frac{2\mu}{r^2}\left(\left(n^2-n-\frac{1}{2}k_s^2r^2\right)H_n^{(1)}(k_sr)+k_srH_{n+1}^{(1)}(k_sr)\right)-\\
&\qquad 2A_n^3\mu\frac{n}{r}\left(k_pH_n^{(2)'}(k_pr)-\frac{1}{r}H_n^{(2)}(k_pr)\right)+\\
&\qquad A_n^4\mu\left(-\frac{n^2}{r^2}H_n^{(2)}(k_sr)+\frac{1}{r}k_sH_n^{(2)'}(k_sr)-k_s^2H_n^{(2)''}(k_sr)\right)\Bigg]\sin(n\theta)\\
&=i\omega\rho v_s\sum_{n=0}^{+\infty}\Big[-A_n^1\frac{n}{r}H_n^{(1)}(k_pr)-A_n^2k_sH_n^{(1)'}(k_sr)-\\
&\qquad A_n^3\frac{n}{r}H_n^{(2)}(k_pr)-A_n^4k_sH_n^{(2)'}(k_sr)\Big]\sin(n\theta).
\end{aligned}\tag{B.21}$$

Since n represents modes of Fourier's serie, we compute the coefficients $\mathbf{A}_n^j$, $j = 1, 4$ by solving the following system at each Fourier mode n, since $\mathbf{C}_n$ is inversible

$$\mathbf{C}_n\mathbf{A}_n = \mathbf{B}_n, \tag{B.22}$$

where $\mathbf{C}_n = \begin{pmatrix}\mathbf{C}_n(:,1) & \mathbf{C}_n(:,2) & \mathbf{C}_n(:,3) & \mathbf{C}_n(:,4)\end{pmatrix}$

with $\mathbf{C}_n(:,1) = \begin{pmatrix} k_pH_n^{(1)'}(k_pa) \\ -\frac{n}{a}H_n^{(1)}(k_pa) \\ \left(\left(n^2-n-\frac{1}{2}k_p^2b^2\right)H_n^{(1)}(k_pb)+k_pbH_{n+1}^{(1)}(k_pb)\right)-i\omega^2\rho\frac{b^2}{2\mu}H_n^{(1)'}(k_pb) \\ -n\left((n-1)\,H_n^{(1)}(k_pb)-k_pbH_{n+1}^{(1)}(k_pb)\right)+i\omega\rho v_sn\frac{b}{2\mu}H_n^{(1)}(k_pb)\end{pmatrix}$,

$\mathbf{C}_n(:,2) = \begin{pmatrix} \frac{n}{a}H_n^{(1)}(k_sa) \\ -k_sH_n^{(1)'}(k_sa) \\ n\left((n-1)\,H_n^{(1)}(k_sb)-k_sbH_{n+1}^{(1)}(k_pb)\right)-i\omega\rho v_pn\frac{b}{2\mu}H_n^{(1)}(k_sb) \\ -\left(\left(n^2-n-\frac{1}{2}k_s^2b^2\right)H_n^{(1)}(k_sb)+k_sbH_{n+1}^{(1)}(k_sb)\right)+i\omega^2\rho\frac{b^2}{2\mu}H_n^{(1)'}(k_sb)\end{pmatrix}$,

$$\mathbf{C}_n(:,3) = \begin{pmatrix} k_p H_n^{(2)'}(k_p a) \\ -\dfrac{n}{a} H_n^{(2)}(k_p a) \\ \left(-\left(\left(k_s^2 - 2k_p^2\right) H_n^{(2)}(k_p b) + 2k_p^2 H_n^{(2)''}(k_p b) \right) - i\omega^2 \rho H_n^{(2)'}(k_p b) \right) \dfrac{b^2}{2} \\ -nb \left(k_p H_n^{(2)'}(k_p b) - \dfrac{1}{b} H_n^{(2)}(k_p b) \right) + i\omega\rho v_s n \dfrac{b}{2\mu} H_n^{(2)}(k_p b) \end{pmatrix},$$

$$\mathbf{C}_n(:,4) = \begin{pmatrix} \dfrac{n}{a} H_n^{(2)}(k_s a) \\ -k_s H_n^{(2)'}(k_s a) \\ nb \left(\dfrac{-1}{b} H_n^{(2)}(k_s b) + k_s H_n^{(2)'}(k_s b) \right) - i\omega\rho v_p n \dfrac{b}{2\mu} H_n^{(2)}(k_s b) \\ \left(\left(-\dfrac{n^2}{b^2} H_n^{(2)}(k_s b) + \dfrac{1}{b} k_s H_n^{(2)'}(k_s b) - k_s^2 H_n^{(2)''}(k_s b) \right) + i\omega^2 \rho \dfrac{1}{\mu} H_n^{(2)'}(k_s b) \right) \dfrac{b^2}{2} \end{pmatrix},$$

$$\mathbf{A}_n = \begin{pmatrix} A_n^1 \\ A_n^2 \\ A_n^3 \\ A_n^4 \end{pmatrix}, \qquad \mathbf{B}_n = \begin{pmatrix} -\varepsilon_n i^n k_p J_n'(k_p a) \\ \dfrac{n}{a} \varepsilon_n i^n J_n(k_p a) \\ 0 \\ 0 \end{pmatrix}.$$

C Analytical solution of the elastic solid scatterer problem

Like in the previous section, we recall here the analytical solution in the form of Fourier series of the elastic disk-shaped solid scatterer problem. The displacement field u is represented with the help to two potentials ϕ and ψ

$$u = \nabla\phi + (-\mathbf{e_z}) \times \nabla\psi, \tag{C.1}$$

where, in the case of a circle, like the domain Ω_a, we refer to figure 4.4.1 for the domains $\Omega, \Omega_a, \Omega_b$

$$\phi = \phi_3 = \sum_{n=0}^{+\infty} A_n^3 J_n(k_p r)\cos(n\theta),$$
$$\psi = \psi_3 = \sum_{n=0}^{+\infty} A_n^4 J_n(k_s r)\sin(n\theta),$$

and in the case of a ring like Ω_b

$$\phi = \phi_1 + \phi_2 = \sum_{n=0}^{+\infty} A_n^1 H_n^{(1)}(k_p r)\cos(n\theta) + \sum_{n=0}^{+\infty} A_n^5 H_n^{(2)}(k_p r)\cos(n\theta),$$
$$\psi = \psi_1 + \psi_2 = \sum_{n=0}^{+\infty} A_n^2 H_n^{(1)}(k_s r)\sin(n\theta) + \sum_{n=0}^{+\infty} A_n^6 H_n^{(2)}(k_s r)\sin(n\theta).$$

$\mathbf{e_z}$ is the third vector of the cartesian basis. $k_p = \dfrac{\omega}{v_p}$ et $k_s = \dfrac{\omega}{v_s}$. $H_n^{(1)}$ represents the Hankel's function of first kind defined such as

$$H_n^{(1)}(x) = J_n(x) + iY_n(x), \tag{C.2}$$

with J_n et Y_n Bessel's functions respectively of first and second kind. $H_n^{(2)}$ is the Hankel's function of second kind defined such as

$$H_n^{(2)}(x) = J_n(x) - iY_n(x). \tag{C.3}$$

In polar coordinates, we get

$$\nabla\phi = \frac{\partial\phi}{\partial r}\mathbf{e_r} + \frac{1}{r}\frac{\partial\phi}{\partial\theta}\mathbf{e}_\theta,$$
$$(-\mathbf{e_z}) \times \nabla\psi = \frac{1}{r}\frac{\partial\psi}{\partial\theta}\mathbf{e_r} - \frac{\partial\psi}{\partial r}\mathbf{e}_\theta.$$

It follows that

$$u = u_r\mathbf{e_r} + u_\theta\mathbf{e}_\theta, \tag{C.4}$$

where $\quad u_r = \dfrac{\partial\phi}{\partial r} + \dfrac{1}{r}\dfrac{\partial\psi}{\partial\theta} \quad$ and $\quad u_\theta = \dfrac{1}{r}\dfrac{\partial\phi}{\partial\theta} - \dfrac{\partial\psi}{\partial r}.$

Since polar vectors $\mathbf{e_r}$ and $\mathbf{e}_\theta$ are defined in cartesian basis by $(\cos\theta, \sin\theta)^t$ and $(-\sin\theta, \cos\theta)^t$ respectively, we obtain the components of the displacement field in the cartesian basis

$$\left\{ \begin{array}{rcl} u_x &=& u_r\cos\theta - u_\theta\sin\theta, \\ u_z &=& u_r\sin\theta + u_\theta\cos\theta. \end{array} \right. \tag{C.5}$$

Then, in Ω_a, the components of displacement field u_3 are written

$$\left\{ \begin{array}{rcl} u_{3x} &=& u_{3r}\cos\theta - u_{3\theta}\sin\theta, \\ u_{3z} &=& u_{3r}\sin\theta + u_{3\theta}\cos\theta, \end{array} \right. \tag{C.6}$$

where

$$u_{3r} = \sum_{n=0}^{+\infty} \Big[A_n^3 k_p J_n'(k_p r) + \frac{n}{r} A_n^4 J_n(k_s r) \Big] \cos(n\theta),$$

$$u_{3\theta} = \sum_{n=0}^{+\infty} \Big[-A_n^3 \frac{n}{r} J_n(k_p r) - A_n^4 k_s J_n'(k_s r) \Big] \sin(n\theta).$$

In Ω_b, the components of displacement field u_1 are written

$$\left\{ \begin{array}{rcrcl} u_{12x} & = & u_{1x} + u_{2x} & = & u_{1r}\cos\theta - u_{1\theta}\sin\theta + u_{2r}\cos\theta - u_{2\theta}\sin\theta \\ & & & = & (u_{1r} + u_{2r})\cos\theta - (u_{1\theta} + u_{2\theta})\sin\theta, \\ u_{12z} & = & u_{1z} + u_{2z} & = & u_{1r}\sin\theta + u_{1\theta}\cos\theta + u_{1r}\sin\theta + u_{1\theta}\cos\theta \\ & & & = & (u_{1r} + u_{2r})\sin\theta + (u_{1\theta} + u_{2\theta})\cos\theta, \end{array} \right. \tag{C.7}$$

where

$$u_{1r} = \sum_{n=0}^{+\infty} \Big[A_n^1 k_p H_n^{(1)'}(k_p r) + \frac{n}{r} A_n^2 H_n^{(1)}(k_s r) \Big] \cos(n\theta),$$

$$u_{1\theta} = \sum_{n=0}^{+\infty} \Big[-A_n^1 \frac{n}{r} H_n^{(1)}(k_p r) - A_n^2 k_s H_n^{(1)'}(k_s r) \Big] \sin(n\theta),$$

$$u_{2r} = \sum_{n=0}^{+\infty} \Big[A_n^5 k_p H_n^{(2)'}(k_p r) + \frac{n}{r} A_n^6 H_n^{(2)}(k_s r) \Big] \cos(n\theta),$$

$$u_{2\theta} = \sum_{n=0}^{+\infty} \Big[-A_n^5 \frac{n}{r} H_n^{(2)}(k_p r) - A_n^6 k_s H_n^{(2)'}(k_s r) \Big] \sin(n\theta),$$

so we get

$$u_{12r} = \sum_{n=0}^{+\infty} \Big[A_n^1 k_p H_n^{(1)'}(k_p r) + \frac{n}{r} A_n^2 H_n^{(1)}(k_s r) + A_n^5 k_p H_n^{(2)'}(k_p r) + \frac{n}{r} A_n^6 H_n^{(2)}(k_s r) \Big] \cos(n\theta),$$

$$u_{12\theta} = \sum_{n=0}^{+\infty} \Big[-A_n^1 \frac{n}{r} H_n^{(1)}(k_p r) - A_n^2 k_s H_n^{(1)'}(k_s r) - A_n^5 \frac{n}{r} H_n^{(2)}(k_p r) - A_n^6 k_s H_n^{(2)'}(k_s r) \Big] \sin(n\theta).$$

Derivatives of Hankel's function are determined by the following reccurence relations

$$H_n^{(1)'}(x) = \left\{ \begin{array}{ll} -H_{n+1}^{(1)}(x) & \text{for } n = 0, \\ -H_{n+1}^{(1)}(x) + \dfrac{n}{x} H_n^{(1)}(x) & \text{for } n > 0, \end{array} \right.$$

$$H_n^{(2)'}(x) = \left\{ \begin{array}{ll} -H_{n+1}^{(2)}(x) & \text{for } n = 0, \\ H_{n-1}^{(2)}(x) - \dfrac{n}{x} H_n^{(2)}(x) & \text{for } n > 0. \end{array} \right.$$

In the same way, the derivative of the Bessel's function of first kind is

$$J_n'(x) = \left\{ \begin{array}{ll} -J_{n+1}(x) & \text{for } n = 0, \\ -J_{n+1}(x) + \dfrac{n}{x} J_n(x) & \text{for } n > 0. \end{array} \right.$$

Using these definitions we rewrite u_{ir} and $u_{i\theta}$, $i = 12, 3$

$$u_{3r} = \sum_{n=0}^{+\infty} \left[A_n^3 k_p \left(-J_{n+1}(k_p r) + \frac{n}{k_p r} J_n(k_p r) \right) + \frac{n}{r} A_n^4 J_n(k_s r) \right] \cos(n\theta), \tag{C.8}$$

$$u_{3\theta} = \sum_{n=0}^{+\infty} \left[-A_n^3 \frac{n}{r} J_n(k_p r) - A_n^4 k_s \left(-J_{n+1}(k_s r) + \frac{n}{k_s r} J_n(k_s r) \right) \right] \sin(n\theta), \tag{C.9}$$

$$\begin{aligned} u_{12r} = \sum_{n=0}^{+\infty} \Bigg[& A_n^1 k_p \left(-H_{n+1}^{(1)}(k_p r) + \frac{n}{k_p r} H_n^{(1)}(k_p r) \right) + \frac{n}{r} A_n^2 H_n^{(1)}(k_s r) + \\ & A_n^5 k_p \left(H_{n-1}^{(2)}(k_p r) - \frac{n}{k_p r} H_n^{(2)}(k_p r) \right) + \frac{n}{r} A_n^6 H_n^{(2)}(k_s r) \Bigg] \cos(n\theta), \end{aligned} \tag{C.10}$$

$$\begin{aligned} u_{12\theta} = \sum_{n=0}^{+\infty} \Bigg[& -A_n^1 \frac{n}{r} H_n^{(1)}(k_p r) - A_n^2 k_s \left(-H_{n+1}^{(1)}(k_s r) + \frac{n}{k_s r} H_n^{(1)}(k_s r) \right) - \\ & A_n^5 \frac{n}{r} H_n^{(2)}(k_p r) - A_n^6 k_s \left(H_{n-1}^{(2)}(k_s r) - \frac{n}{k_s r} H_n^{(2)}(k_s r) \right) \Bigg] \sin(n\theta). \end{aligned} \tag{C.11}$$

In order to determine coefficients A_n^i, $i = 1, .., 6$, we use boundary conditions. We get

$$\begin{cases} u_{12} + u^{inc} &= u_3 \\ \underline{\underline{\sigma}}_{12}\mathbf{n} + \underline{\underline{\sigma}}^{inc}\mathbf{n} &= \underline{\underline{\sigma}}_{3}\mathbf{n} \end{cases} \qquad \text{on } \Gamma_a, \tag{C.12}$$

$$\underline{\underline{\sigma}}_{12}\mathbf{n} = v_p \left(v_{12} \cdot \mathbf{n} \right) \mathbf{n} + v_s \left(v_{12} \cdot \mathbf{t} \right) \mathbf{t} \qquad \text{on } \Gamma_b. \tag{C.13}$$

We refer to figure 4.4.1 for the definition of boundaries Γ_a and Γ_b.

u^{inc} is the displacement field of the incident wave described by

$$u^{inc} = \nabla \phi^{inc} + (-\mathbf{e_z}) \times \nabla \psi^{inc},$$

with $\psi^{inc} = 0$ and

$$\phi^{inc} = \sum_{n=0}^{+\infty} \varepsilon_n i^n J_n(k_p r) \cos(n\theta),$$

where $\varepsilon_n = \begin{cases} 1 & \text{if } n = 0 \\ 2 & \text{if } n \geq 1 \end{cases}$.

In summary, we have

$$\begin{aligned} u^{inc} &= \frac{\partial \phi^{inc}}{\partial r} \mathbf{e_r} + \frac{1}{r} \frac{\partial \phi^{inc}}{\partial \theta} \mathbf{e}_\theta \\ &= \sum_{n=0}^{+\infty} \varepsilon_n i^n \cos(n\theta) k_p J_n'(k_p r) \mathbf{e_r} + \sum_{n=0}^{+\infty} \frac{-n}{r} \varepsilon_n i^n J_n(k_p r) \sin(n\theta) \mathbf{e}_\theta, \end{aligned} \tag{C.14}$$

The free surface condition on Γ_a is given by

$$\begin{cases} \sigma_{rr_{12}} + \sigma_{rr}^{inc} &= \sigma_{rr_3}, \\ \sigma_{r\theta_{12}} + \sigma_{r\theta}^{inc} &= \sigma_{r\theta_3}, \end{cases} \tag{C.15}$$

and the absorbing condition on Γ_b by

$$\begin{cases} \sigma_{rr_{12}} &= i\omega v_p u_{12r}, \\ \sigma_{r\theta_{12}} &= i\omega v_s u_{12\theta}. \end{cases} \tag{C.16}$$

The components σ_{rr} and $\sigma_{r\theta}$ of $\underline{\underline{\sigma}}$ are expressed such as

$$\left\{ \begin{array}{lcl} \sigma_{rr} & = & \sigma_{rr}^{\phi} + \sigma_{rr}^{\psi}, \\ \sigma_{r\theta} & = & \sigma_{r\theta}^{\phi} + \sigma_{r\theta}^{\psi}, \end{array} \right. \tag{C.17}$$

where

$$\begin{aligned} \sigma_{rr}^{\phi} &= \lambda \Delta\phi + 2\mu \frac{\partial^2 \phi}{\partial r^2}, \\ \sigma_{rr}^{\psi} &= 2\mu \left[\frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial \psi}{\partial \theta} \right) \right], \\ \sigma_{r\theta}^{\phi} &= 2\mu \left(\frac{1}{r} \frac{\partial^2 \phi}{\partial \theta \partial r} - \frac{1}{r^2} \frac{\partial \phi}{\partial \theta} \right), \\ \sigma_{r\theta}^{\psi} &= \mu \left(\frac{1}{r^2} \frac{\partial^2 \psi}{\partial \theta^2} - r \frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial \psi}{\partial r} \right) \right). \end{aligned}$$

ϕ satisfies the Helmholtz equation

$$\Delta\phi = -k_p^2 \phi,$$

so

$$\Delta\phi = -k_p^2 \sum_{n=0}^{+\infty} A_n^h C_n(k_p r) \cos(n\theta), h = 1, 3 \text{ or} 5,$$

and $C_n(k_p r) = H_n^{(1)}(k_p r)$, $J_n(k_p r)$ or $H_n^{(2)}(k_p r)$. For $\underline{\underline{\sigma}}_1$ we get

$$\left\{ \begin{array}{lcl} \sigma_{rr_1} & = & \displaystyle\sum_{n=0}^{+\infty} \left[A_n^1 k_p^2 \left(\left(-\lambda - \frac{2\mu}{(k_p r)^2} \left(n + (k_p r)^2 - n^2 \right) \right) H_n^{(1)}(k_p r) + 2\mu \frac{1}{k_p r} H_{n+1}^{(1)}(k_p r) \right) + \right. \\ & & \left. 2\mu A_n^2 \frac{n}{r} \left(\frac{-1}{r} (1-n) H_n^{(1)}(k_s r) - k_s H_{n+1}^{(1)}(k_s r) \right) \right] \cos(n\theta), \\ \sigma_{r\theta_1} & = & \mu \displaystyle\sum_{n=0}^{+\infty} \left[-2A_n^1 \frac{n}{r} \left(-k_p H_{n+1}^{(1)}(k_p r) + \frac{n-1}{r} H_n^{(1)}(k_p r) \right) + \right. \\ & & \left. A_n^2 \frac{2}{r} \left(\frac{1}{r} \left(-n^2 + n + \frac{1}{2}(k_s r)^2 \right) H_n^{(1)}(k_s r) - k_s H_{n+1}^{(1)}(k_s r) \right) \right] \sin(n\theta). \end{array} \right. \tag{C.18}$$

For $\underline{\underline{\sigma}}_2$

$$\left\{ \begin{array}{lcl} \sigma_{rr_2} & = & \displaystyle\sum_{n=0}^{+\infty} \left[A_n^5 k_p^2 \left(\left(-\lambda + \frac{2\mu}{(k_p r)^2} \left(n - (k_p r)^2 + n^2 \right) \right) H_n^{(2)}(k_p r) - 2\mu \frac{1}{k_p r} H_{n-1}^{(2)}(k_p r) \right) + \right. \\ & & \left. 2\mu A_n^6 \frac{n}{r} \left(\frac{-1}{r} (1+n) H_n^{(2)}(k_s r) + k_s H_{n-1}^{(2)}(k_s r) \right) \right] \cos(n\theta). \\ \sigma_{r\theta_2} & = & \mu \displaystyle\sum_{n=0}^{+\infty} \left[-2A_n^5 \frac{n}{r} \left(k_p H_{n-1}^{(2)}(k_p r) - \frac{n+1}{r} H_n^{(2)}(k_p r) \right) + \right. \\ & & \left. A_n^6 \frac{2}{r} \left(-\frac{1}{r} \left(n^2 + n - \frac{1}{2}(k_s r)^2 \right) H_n^{(2)}(k_s r) + k_s H_{n-1}^{(2)}(k_s r) \right) \right] \sin(n\theta). \end{array} \right. \tag{C.19}$$

For $\underline{\underline{\sigma}}_3$

$$\left\{\begin{array}{rcl}\sigma_{rr_3} &=& \displaystyle\sum_{n=0}^{+\infty}\left[A_n^3k_p^2\left(\left(-\lambda-\frac{2\mu}{(k_pr)^2}\left(n+(k_pr)^2-n^2\right)\right)J_n(k_pr)+2\mu\frac{1}{k_pr}J_{n+1}(k_pr)\right)+\right.\\ &&\left.2\mu A_n^4\frac{n}{r}\left(\frac{-1}{r}\left(1-n\right)J_n(k_sr)-k_sJ_{n+1}(k_sr)\right)\right]\cos(n\theta),\\ \sigma_{r\theta_3} &=& \mu\displaystyle\sum_{n=0}^{+\infty}\left[-2A_n^3\frac{n}{r}\left(-k_pJ_{n+1}(k_pr)+\frac{n-1}{r}J_n(k_pr)\right)+\right.\\ &&\left.A_n^4\frac{2}{r}\left(\frac{1}{r}\left(-n^2+n+\frac{1}{2}(k_sr)^2\right)J_n(k_sr)-k_sJ_{n+1}(k_sr)\right)\right]\sin(n\theta).\end{array}\right. \tag{C.20}$$

Finally for $\underline{\underline{\sigma}}^{inc}$

$$\left\{\begin{array}{rcl}\sigma_{rr}^{inc} &=& \displaystyle\sum_{n=0}^{+\infty}\varepsilon_ni^n\left[\left(-\lambda k_p^2-2\mu\frac{1}{r^2}\left(n-n^2+(k_pr)^2\right)\right)J_n(k_pr)+2\mu\frac{k_p}{r}J_{n+1}(k_pr)\right]\cos(n\theta),\\ \sigma_{r\theta}^{inc} &=& \displaystyle\sum_{n=0}^{+\infty}\left[-2\mu\varepsilon_ni^n\frac{n}{r}\left(-\frac{1-n}{r}J_n(k_pr)-k_pJ_{n+1}(k_pr)\right)\right]\sin(n\theta).\end{array}\right. \tag{C.21}$$

For $r=a$ we have

$$\begin{aligned}&\sum_{n=0}^{+\infty}\left[A_n^1k_p\left(-H_{n+1}^{(1)}(k_pr)+\frac{n}{k_pr}H_n^{(1)}(k_pr)\right)+\frac{n}{r}A_n^2H_n^{(1)}(k_sr)+\right.\\ &\qquad A_n^5k_p\left(H_{n-1}^{(2)}(k_pr)-\frac{n}{k_pr}H_n^{(2)}(k_pr)\right)+\frac{n}{r}A_n^6H_n^{(2)}(k_sr)-\\ &\qquad\left.A_n^3k_p\left(-J_{n+1}(k_pr)+\frac{n}{k_pr}J_n(k_pr)\right)-\frac{n}{r}A_n^4J_n(k_sr)\right]\cos(n\theta)\\ &=\sum_{n=0}^{+\infty}-\varepsilon_ni^nk_p\left(-J_{n+1}(k_pr)+\frac{n}{k_pr}J_n(k_pr)\right)\cos(n\theta),\end{aligned}$$

$$\begin{aligned}&\sum_{n=0}^{+\infty}\left[-A_n^1\frac{n}{r}H_n^{(1)}(k_pr)-A_n^2k_s\left(-H_{n+1}^{(1)}(k_sr)+\frac{n}{k_sr}H_n^{(1)}(k_sr)\right)-A_n^5\frac{n}{r}H_n^{(2)}(k_pr)-\right.\\ &\qquad A_n^6k_s\left(H_{n-1}^{(2)}(k_sr)-\frac{n}{k_sr}H_n^{(2)}(k_sr)\right)+A_n^3\frac{n}{r}J_n(k_pr)+\\ &\qquad\left.A_n^4k_s\left(-J_{n+1}(k_sr)+\frac{n}{k_sr}J_n(k_sr)\right)\right]\sin(n\theta)\\ &=\sum_{n=0}^{+\infty}\frac{n}{r}\varepsilon_ni^nJ_n(k_pr)\sin(n\theta)\end{aligned}$$

$$\sum_{n=0}^{+\infty}\Bigg[A_n^1k_p^2\left(\left(-\lambda-\frac{2\mu}{(k_pr)^2}\left(n+(k_pr)^2-n^2\right)\right)H_n^{(1)}(k_pr)+2\mu\frac{1}{k_pr}H_{n+1}^{(1)}(k_pr)\right)+$$
$$2\mu A_n^2\frac{n}{r}\left(\frac{-1}{r}\left(1-n\right)H_n^{(1)}(k_sr)-k_sH_{n+1}^{(1)}(k_sr)\right)+$$
$$A_n^5k_p^2\left(\left(-\lambda+\frac{2\mu}{(k_pr)^2}\left(n-(k_pr)^2+n^2\right)\right)H_n^{(2)}(k_pr)-2\mu\frac{1}{k_pr}H_{n-1}^{(1)}(k_pr)\right)+$$
$$2\mu A_n^6\frac{n}{r}\left(\frac{-1}{r}\left(1+n\right)H_n^{(2)}(k_sr)+k_sH_{n-1}^{(2)}(k_sr)\right)-$$
$$A_n^3k_p^2\left(\left(-\lambda-\frac{2\mu}{(k_pr)^2}\left(n+(k_pr)^2-n^2\right)\right)J_n(k_pr)+2\mu\frac{1}{k_pr}J_{n+1}(k_pr)\right)-$$
$$2\mu A_n^4\frac{n}{r}\left(\frac{-1}{r}\left(1-n\right)J_n(k_sr)-k_sJ_{n+1}(k_sr)\right)\Bigg]\cos(n\theta),$$
$$=-\sum_{n=0}^{+\infty}\varepsilon_ni^n\left[\left(-\lambda k_p^2-2\mu\frac{1}{r^2}\left(n-n^2+(k_pr)^2\right)\right)J_n(k_pr)+2\mu\frac{k_p}{r}J_{n+1}(k_pr)\right]\cos(n\theta),$$

$$\mu\sum_{n=0}^{+\infty}\Bigg[-2A_n^1\frac{n}{r}\left(-k_pH_{n+1}^{(1)}(k_pr)+\frac{n-1}{r}H_n^{(1)}(k_pr)\right)+$$
$$A_n^2\frac{2}{r}\left(\frac{1}{r}\left(-n^2+n+\frac{1}{2}(k_sr)^2\right)H_n^{(1)}(k_sr)-k_sH_{n+1}^{(1)}(k_sr)\right)-$$
$$2A_n^5\frac{n}{r}\left(k_pH_{n-1}^{(2)}(k_pr)-\frac{n+1}{r}H_n^{(2)}(k_pr)\right)+$$
$$A_n^6\frac{2}{r}\left(-\frac{1}{r}\left(n^2+n-\frac{1}{2}(k_sr)^2\right)H_n^{(2)}(k_sr)+k_sH_{n-1}^{(2)}(k_sr)\right)+$$
$$2A_n^3\frac{n}{r}\left(-k_pJ_{n+1}(k_pr)+\frac{n-1}{r}J_n(k_pr)\right)-$$
$$A_n^4\frac{2}{r}\left(\frac{1}{r}\left(-n^2+n+\frac{1}{2}(k_sr)^2\right)J_n(k_sr)-k_sJ_{n+1}(k_sr)\right)\Bigg]\sin(n\theta)$$
$$=\sum_{n=0}^{+\infty}\left[-2\mu\varepsilon_ni^n\frac{n}{r}\left(-\frac{1-n}{r}J_n(k_pr)-k_pJ_{n+1}(k_pr)\right)\right]\sin(n\theta).$$

For $r = b$, we have

$$\begin{aligned}
&\sum_{n=0}^{+\infty}\Bigg[A_n^1\left(k_p^2\left(\left(-\lambda-\frac{2\mu}{(k_pr)^2}\left(n+(k_pr)^2-n^2\right)\right)H_n^{(1)}(k_pr)+2\mu\frac{1}{k_pr}H_{n+1}^{(1)}(k_pr)\right)\right.\\
&\quad\left.-i\omega v_pk_p\left(-H_{n+1}^{(1)}(k_pr)+\frac{n}{k_pr}H_n^{(1)}(k_pr)\right)\right)\\
&\quad+A_n^2\left(2\mu\frac{n}{r}\left(\frac{-1}{r}\left(1-n\right)H_n^{(1)}(k_sr)-k_sH_{n+1}^{(1)}(k_sr)\right)-i\omega v_p\frac{n}{r}H_n^{(1)}(k_sr)\right)\\
&\quad+A_n^5\left(k_p^2\left(\left(-\lambda+\frac{2\mu}{(k_pr)^2}\left(n-(k_pr)^2+n^2\right)\right)H_n^{(2)}(k_pr)-2\mu\frac{1}{k_pr}H_{n-1}^{(2)}(k_pr)\right)\right.\\
&\quad\left.-i\omega v_pk_p\left(H_{n-1}^{(2)}(k_pr)-\frac{n}{k_pr}H_n^{(2)}(k_pr)\right)\right)+\\
&\quad A_n^6\left(2\mu\frac{n}{r}\left(\frac{-1}{r}\left(1+n\right)H_n^{(2)}(k_sr)+k_sH_{n-1}^{(2)}(k_sr)\right)-i\omega v_p\frac{n}{r}H_n^{(2)}(k_sr)\right)\Bigg]\cos(n\theta)=0
\end{aligned}\tag{C.22}$$

n representing modes of Fourier's serie, we get for each Fourier mode n. For $r = a$

$$\begin{aligned}
&\left[A_n^1k_p\left(-H_{n+1}^{(1)}(k_pr)+\frac{n}{k_pr}H_n^{(1)}(k_pr)\right)+\frac{n}{r}A_n^2H_n^{(1)}(k_sr)+A_n^5k_p\left(H_{n-1}^{(2)}(k_pr)-\frac{n}{k_pr}H_n^{(2)}(k_pr)\right)\right.\\
&\quad\left.+\frac{n}{r}A_n^6H_n^{(2)}(k_sr)-A_n^3k_p\left(-J_{n+1}(k_pr)+\frac{n}{k_pr}J_n(k_pr)\right)-\frac{n}{r}A_n^4J_n(k_sr)\right]\cos(n\theta)\\
&=-\varepsilon_ni^nk_p\left(-J_{n+1}(k_pr)+\frac{n}{k_pr}J_n(k_pr)\right)\cos(n\theta),
\end{aligned}\tag{C.23}$$

$$\begin{aligned}
&\left[-A_n^1\frac{n}{r}H_n^{(1)}(k_pr)-A_n^2k_s\left(-H_{n+1}^{(1)}(k_sr)+\frac{n}{k_sr}H_n^{(1)}(k_sr)\right)-A_n^5\frac{n}{r}H_n^{(2)}(k_pr)-\right.\\
&\qquad A_n^6k_s\left(H_{n-1}^{(2)}(k_sr)-\frac{n}{k_sr}H_n^{(2)}(k_sr)\right)+A_n^3\frac{n}{r}J_n(k_pr)+\\
&\qquad\left.A_n^4k_s\left(-J_{n+1}(k_sr)+\frac{n}{k_sr}J_n(k_sr)\right)\right]\sin(n\theta)\\
&\quad=\frac{n}{r}\varepsilon_ni^nJ_n(k_pr)\sin(n\theta)
\end{aligned}\tag{C.24}$$

$$\begin{aligned}
&\left[A_n^1k_p^2\left(\left(-\lambda-\frac{2\mu}{(k_pr)^2}\left(n+(k_pr)^2-n^2\right)\right)H_n^{(1)}(k_pr)+2\mu\frac{1}{k_pr}H_{n+1}^{(1)}(k_pr)\right)+\right.\\
&\qquad 2\mu A_n^2\frac{n}{r}\left(\frac{-1}{r}\left(1-n\right)H_n^{(1)}(k_sr)-k_sH_{n+1}^{(1)}(k_sr)\right)+\\
&\qquad A_n^5k_p^2\left(\left(-\lambda+\frac{2\mu}{(k_pr)^2}\left(n-(k_pr)^2+n^2\right)\right)H_n^{(2)}(k_pr)-2\mu\frac{1}{k_pr}H_{n-1}^{(1)}(k_pr)\right)+\\
&\qquad 2\mu A_n^6\frac{n}{r}\left(\frac{-1}{r}\left(1+n\right)H_n^{(2)}(k_sr)+k_sH_{n-1}^{(2)}(k_sr)\right)-\\
&\qquad A_n^3k_p^2\left(\left(-\lambda-\frac{2\mu}{(k_pr)^2}\left(n+(k_pr)^2-n^2\right)\right)J_n(k_pr)+2\mu\frac{1}{k_pr}J_{n+1}(k_pr)\right)-\\
&\qquad\left.2\mu A_n^4\frac{n}{r}\left(\frac{-1}{r}\left(1-n\right)J_n(k_sr)-k_sJ_{n+1}(k_sr)\right)\right]\cos(n\theta)\\
&=-\varepsilon_ni^n\left[\left(-\lambda k_p^2-2\mu\frac{1}{r^2}\left(n-n^2+(k_pr)^2\right)\right)J_n(k_pr)+2\mu\frac{k_p}{r}J_{n+1}(k_pr)\right]\cos(n\theta),
\end{aligned}\tag{C.25}$$

$$
\begin{aligned}
&\mu\left[-2A_n^1\frac{n}{r}\left(-k_pH_{n+1}^{(1)}(k_pr)+\frac{n-1}{r}H_n^{(1)}(k_pr)\right)+\right.\\
&\quad A_n^2\frac{2}{r}\left(\frac{1}{r}\left(-n^2+n+\frac{1}{2}(k_sr)^2\right)H_n^{(1)}(k_sr)-k_sH_{n+1}^{(1)}(k_sr)\right)-\\
&\quad 2A_n^5\frac{n}{r}\left(k_pH_{n-1}^{(2)}(k_pr)-\frac{n+1}{r}H_n^{(2)}(k_pr)\right)+\\
&\quad A_n^6\frac{2}{r}\left(-\frac{1}{r}\left(n^2+n-\frac{1}{2}(k_sr)^2\right)H_n^{(2)}(k_sr)+k_sH_{n-1}^{(2)}(k_sr)\right)+\\
&\quad 2A_n^3\frac{n}{r}\left(-k_pJ_{n+1}(k_pr)+\frac{n-1}{r}J_n(k_pr)\right)-\\
&\quad \left.A_n^4\frac{2}{r}\left(\frac{1}{r}\left(-n^2+n+\frac{1}{2}(k_sr)^2\right)J_n(k_sr)-k_sJ_{n+1}(k_sr)\right)\right]\sin(n\theta)\\
&=\left[-2\mu\varepsilon_ni^n\frac{n}{r}\left(-\frac{1-n}{r}J_n(k_pr)-k_pJ_{n+1}(k_pr)\right)\right]\sin(n\theta).
\end{aligned}
\tag{C.26}
$$

For $r=b$, we have

$$
\begin{aligned}
&\left[A_n^1\left(k_p^2\left(\left(-\lambda-\frac{2\mu}{(k_pr)^2}\left(n+(k_pr)^2-n^2\right)\right)H_n^{(1)}(k_pr)+2\mu\frac{1}{k_pr}H_{n+1}^{(1)}(k_pr)\right)\right.\\
&-i\omega v_pk_p\left(-H_{n+1}^{(1)}(k_pr)+\frac{n}{k_pr}H_n^{(1)}(k_pr)\right)\Big)\\
&+A_n^2\left(2\mu\frac{n}{r}\left(\frac{-1}{r}(1-n)H_n^{(1)}(k_sr)-k_sH_{n+1}^{(1)}(k_sr)\right)-i\omega v_p\frac{n}{r}H_n^{(1)}(k_sr)\right)\\
&+A_n^5\left(k_p^2\left(\left(-\lambda+\frac{2\mu}{(k_pr)^2}\left(n-(k_pr)^2+n^2\right)\right)H_n^{(2)}(k_pr)-2\mu\frac{1}{k_pr}H_{n-1}^{(2)}(k_pr)\right)\right.\\
&-i\omega v_pk_p\left(H_{n-1}^{(2)}(k_pr)-\frac{n}{k_pr}H_n^{(2)}(k_pr)\right)\Big)+\\
&\left.A_n^6\left(2\mu\frac{n}{r}\left(\frac{-1}{r}(1+n)H_n^{(2)}(k_sr)+k_sH_{n-1}^{(2)}(k_sr)\right)-i\omega v_p\frac{n}{r}H_n^{(2)}(k_sr)\right)\right]\cos(n\theta)=0
\end{aligned}
\tag{C.27}
$$

$$
\begin{aligned}
&\left[A_n^1\left(-2\mu\frac{n}{r}\left(-k_pH_{n+1}^{(1)}(k_pr)+\frac{n-1}{r}H_n^{(1)}(k_pr)\right)+i\omega v_s\frac{n}{r}H_n^{(1)}(k_pr)\right)+\right.\\
&A_n^2\left(\mu\frac{2}{r}\left(\frac{1}{r}\left(-n^2+n+\frac{1}{2}(k_sr)^2\right)H_n^{(1)}(k_sr)-k_sH_{n+1}^{(1)}(k_sr)\right)\right.\\
&\left.+i\omega v_sk_s\left(-H_{n+1}^{(1)}(k_sr)+\frac{n}{k_sr}H_n^{(1)}(k_sr)\right)\right)\\
&A_n^5\left(-2\mu\frac{n}{r}\left(k_pH_{n-1}^{(2)}(k_pr)-\frac{n+1}{r}H_n^{(2)}(k_pr)\right)+i\omega v_s\frac{n}{r}H_n^{(2)}(k_pr)\right)\\
&+A_n^6\left(\mu\frac{2}{r}\left(-\frac{1}{r}\left(n^2+n-\frac{1}{2}(k_sr)^2\right)H_n^{(2)}(k_sr)+k_sH_{n-1}^{(2)}(k_sr)\right)\right.\\
&\left.\left.+i\omega v_sk_s\left(H_{n-1}^{(2)}(k_sr)-\frac{n}{k_sr}H_n^{(2)}(k_sr)\right)\right)\right]\sin(n\theta)=0.
\end{aligned}
\tag{C.28}
$$

Since $k_p^2(\lambda+2\mu)=k_s^2\mu$, we replace λ by $\dfrac{k_s^2}{k_p^2}\mu-2\mu$ and rewriting equations (C.23) to (C.28) under matrix form, we finally get

$$
\mathbf{C}_n\mathbf{A}_n=\mathbf{B}_n, \tag{C.29}
$$

where

$$\mathbf{C}_n = \left(\mathbf{C}_n(:,1),\quad \mathbf{C}_n(:,2),\quad \mathbf{C}_n(:,3),\quad \mathbf{C}_n(:,4),\quad \mathbf{C}_n(:,5),\quad \mathbf{C}_n(:,6)\right),$$

with

$$\mathbf{C}_n(:,1) = \begin{pmatrix} nH_n^{(1)}(k_{p_a}a) - k_{p_a}aH_{n+1}^{(1)}(k_{p_a}a) \\ -nH_n^{(1)}(k_{p_a}a) \\ \left(n^2 - n - \frac{1}{2}k_{s_a}^2a^2\right)H_n^{(1)}(k_{p_a}a) + k_{p_a}aH_{n+1}^{(1)}(k_{p_a}a) \\ -n\left((n-1)H_n^{(1)}(k_{p_a}a) - k_{p_a}aH_{n+1}^{(1)}(k_{p_a}a)\right) \\ \left(n^2 - n - \frac{1}{2}k_s^2b^2\right)H_n^{(1)}(k_pb) + k_pbH_{n+1}^{(1)}(k_pb) - i\omega v_p\frac{b}{2\mu}\left(nH_n^{(1)}(k_pb) - k_pbH_{n+1}^{(1)}(k_pb)\right) \\ -n\left((n-1)H_n^{(1)}(k_pb) - k_pbH_{n+1}^{(1)}(k_pb)\right) - i\omega v_s\frac{r}{2\mu_b}H_n^{(1)}(k_pb) \end{pmatrix},$$

$$\mathbf{C}_n(:,2) = \begin{pmatrix} nH_n^{(1)}(k_{s_a}a) \\ -(nH_n^{(1)}(k_{s_a}a) - k_{s_a}aH_{n+1}^{(1)}(k_{s_a}a)) \\ n\left((n-1)H_n^{(1)}(k_{s_a}a) - k_{s_a}aH_{n+1}^{(1)}(k_{s_a}a)\right) \\ \left(-(n^2 - n - \frac{1}{2}k_{s_a}^2a^2\right)H_n^{(1)}(k_{s_a}a) - k_{s_a}aH_{n+1}^{(1)}(k_{s_a}a) \\ n\left((n-1)H_n^{(1)}(k_sb) - k_sbH_{n+1}^{(1)}(k_sb)\right) - i\omega v_p\frac{b}{2\mu}H_n^{(1)}(k_sb) \\ \left(-n^2 + n + \frac{1}{2}k_s^2b^2\right)H_n^{(1)}(k_sb) - k_sbH_{n+1}^{(1)}(k_sb) + i\omega v_s\frac{b}{2\mu}\left(nH_n^{(1)}(k_sb) - k_sbH_{n+1}^{(1)}(k_sb)\right) \end{pmatrix},$$

$$\mathbf{C}_n(:,3) = \begin{pmatrix} -nJ_n(k_{p_a}a) + k_{p_a}aJ_{n+1}(k_{p_a}a) \\ nJ_n(k_{p_a}a) \\ -\left(n^2 - n - \frac{1}{2}k_{s_a}^2a^2\right)J_n(k_{p_a}a) + k_{p_a}aJ_{n+1}(k_{p_a}a) \\ n\left((n-1)J_n(k_{p_a}a) - k_{p_a}aJ_{n+1}(k_{p_a}a)\right) \\ 0 \\ 0 \end{pmatrix},$$

$$\mathbf{C}_n(:,4) = \begin{pmatrix} -nJ_n(k_{s_a}a) \\ nJ_n(k_{s_a}a) - k_{s_a}aJ_{n+1}(k_{s_a}a) \\ -n\left((n-1)J_n(k_{s_a}a) - k_{s_a}aJ_{n+1}(k_{s_a}a)\right) \\ -\left(n^2 - n - \frac{1}{2}k_{s_a}^2a^2\right)J_n(k_{s_a}a) + k_{s_a}aJ_{n+1}(k_{s_a}a) \\ 0 \\ 0 \end{pmatrix},$$

$$\mathbf{C}_n(:,5) = \begin{pmatrix} -nH_n^{(2)}(k_{p_a}a) + k_{p_a}aH_{n-1}^{(2)}(k_{p_a}a) \\ -nH_n^{(2)}(k_{p_a}a) \\ \left(n^2 + n - \frac{1}{2}k_{s_a}^2a^2\right)H_n^{(2)}(k_{p_a}a) - k_{p_a}aH_{n-1}^{(2)}(k_{p_a}a) \\ -n\left(-(n+1)H_n^{(2)}(k_{p_a}a) + k_{p_a}aH_{n-1}^{(2)}(k_{p_a}a)\right) \\ (n^2 + n - \frac{1}{2}k_s^2b^2)H_n^{(2)}(k_pb) - k_pbH_{n-1}^{(2)}(k_pb) - i\omega v_p\frac{b}{2\mu}\left(k_pbH_{n-1}^{(2)}(k_pb) - nH_n^{(2)}(k_pb)\right) \\ -n\left((n+1)H_n^{(2)}(k_pb) - k_pbH_{n-1}^{(2)}(k_pb)\right) + i\omega v_s\frac{b}{2\mu_b}H_n^{(2)}(k_pb) \end{pmatrix},$$

$$\mathbf{C}_n(:,6) = \begin{pmatrix} nH_n^{(2)}(k_{s_a}a) \\ -(-nH_n^{(2)}(k_{s_a}a) + k_{s_a}aH_{n-1}^{(2)}(k_{s_a}a)) \\ n\left(-(n+1)H_n^{(2)}(k_{s_a}a) + k_{s_a}aH_{n-1}^{(2)}(k_{s_a}a)\right) \\ \left(-(n^2+n-\frac{1}{2}k_{s_a}^2a^2\right)H_n^{(2)}(k_{s_a}a) + k_{s_a}aH_{n-1}^{(2)}(k_{s_a}a) \\ n\left(-(n+1)H_n^{(2)}(k_sb) + k_sbH_{n-1}^{(2)}(k_sb)\right) - i\omega v_p\frac{b}{2\mu}H_n^{(2)}(k_sb) \\ -(n^2+n-\frac{1}{2}k_s^2b^2)H_n^{(2)}(k_sb) + k_sbH_{n-1}^{(2)}(k_sb) + i\omega v_s\frac{b}{2\mu}\left(k_sbH_{n-1}^{(2)}(k_sb) - nH_n^{(2)}(k_sb)\right) \end{pmatrix};$$

$$\mathbf{A}_n = \begin{pmatrix} A_n^1 \\ A_n^2 \\ A_n^3 \\ A_n^4 \\ A_n^5 \\ A_n^6 \end{pmatrix},$$

$$\mathbf{B}_n = \begin{pmatrix} -\varepsilon_n i^n\left(-k_paJ_{n+1}(k_pa) + nJ_n(k_pa)\right) \\ n\varepsilon_n i^n J_n(k_pa) \\ -\varepsilon_n i^n\left[-\left(n-n^2+\frac{1}{2}k_s^2a^2\right)J_n(k_pa) - k_paJ_{n+1}(k_pa)\right] \\ -\varepsilon_n i^n n\left((n-1)J_n(k_pa) - k_paJ_{n+1}(k_pa)\right) \\ 0 \\ 0 \end{pmatrix}.$$

Since $\mathbf{C}_n$ is inversible, we have to solve the following system to find coefficients $\mathbf{A}_n^j$, $j = 1, 6$

$$\mathbf{A}_n = \mathbf{C}_n^{-1}\mathbf{B}_n. \tag{C.30}$$