



HAL
open science

Méthode de visualisation adaptée aux simulations d'ordre élevé. Application à la compression-reconstruction de champs rayonnés pour des ondes harmoniques

Matthieu Maunoury

► **To cite this version:**

Matthieu Maunoury. Méthode de visualisation adaptée aux simulations d'ordre élevé. Application à la compression-reconstruction de champs rayonnés pour des ondes harmoniques. Physique mathématique [math-ph]. Université Toulouse 3 Paul Sabatier, 2019. Français. NNT : . tel-02274979

HAL Id: tel-02274979

<https://hal.science/tel-02274979>

Submitted on 30 Aug 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le 30/01/2019 par :

MATTHIEU MAUNOURY

Méthode de visualisation adaptée aux simulations d'ordre élevé.
Application à la compression-reconstruction de champs rayonnés pour
des ondes harmoniques.

JURY

| | | |
|--------------------------|----------------------------|-----------------------|
| FRANÇOIS ALOUGES | CMAP - École Polytechnique | Rapporteur |
| CHRISTOPHE BESSE | Université Paul Sabatier | Directeur de thèse |
| VIRGINIE BONNAILLIE-NOEL | ENS Paris | Présidente |
| FRANCK BOYER | Université Paul Sabatier | Examineur |
| CHRISTOPHE GEUZAINÉ | Université de Liège | Rapporteur |
| ADRIEN LOSEILLE | INRIA - Saclay | Examineur |
| VINCENT MOUYSSSET | ONERA - Toulouse | Co-directeur de thèse |
| SÉBASTIEN PERNET | ONERA - Toulouse | Co-directeur de thèse |

École doctorale et spécialité :

MITT : Domaine Mathématiques : Mathématiques appliquées

Unité de Recherche :

Equipe d'accueil ONERA-ISAE, MOIS : MODélisation et Ingénierie des Systèmes

Directeur(s) de Thèse :

Christophe BESSE, Vincent MOUYSSSET et Sébastien PERNET

Rapporteurs :

François ALOUGES et Christophe GEUZAINÉ

Remerciements

Je souhaite remercier toutes les personnes qui ont participé, de près ou de loin, à cette aventure. J'ai passé trois superbes années à Toulouse grâce à de nombreuses personnes que j'ai rencontré pendant mon doctorat et d'autres que je connais depuis plus longtemps.

Tout d'abord, je remercie les personnes qui m'ont encadrés scientifiquement (et pas que!) : la « Team Visu » ! Je remercie chaleureusement mon directeur de thèse, Christophe Besse pour son encadrement, sa confiance et son recul qui nous a permis de (quasiment toujours) tenir les *deadlines*. Au-delà de sa maîtrise scientifique, ses conseils¹ auront été très précieux. Je remercie profondément Vincent Mouysset et Sébastien Pernet pour leur investissement presque quotidien mais également pour toutes les discussions scientifiques ou non. Leurs apports dans ces travaux sont naturellement démentiels.

J'exprime une immense gratitude à François Alouges et Christophe Geuzaine d'avoir accepté de rapporter ma thèse. Je suis très content que François Alouges se soit intéressé à mes recherches, puisque ses sujets de prédilection sont *a priori* différents, mais sa curiosité a fait qu'il s'est intéressé à ce sujet. Je suis également ravi que Christophe Geuzaine ait rapporté ma thèse puisqu'il connaît très bien le sujet et s'est posé des questions de visualisation de simulation d'ordre élevé il y a déjà une dizaine d'années !

Je suis content que Virginie Bonnaillie-Noël ait présidée ce jury et ait parfaitement mené les échanges. Je suis ravi que Franck Boyer compose également ce jury et se soit intéressé à ces travaux. Enfin, je suis très content que Adrien Loseille complète ce jury puisque cette problématique lui tient également à cœur. Je suis ravi de la confiance qu'il m'accorde et je pense que notre collaboration² sera fructueuse.

Je remercie tous les doctorants, post-docs et stagiaires rencontrés à l'ONERA pour tous les moments passés ensemble : Emanuele, Nadir, César, Matthieu, Marichka, Nour, Kostas, Jeanne, Hugo, Guillaume, Kévin, David, Oumar, Marius, Marc, Jonathan, Asma, Paul, Lucien, Adrian, Hakon... J'en profite pour saluer Chiara et Daniela. La vie du centre de recherche serait bien plus fade sans tous ces jeunes ! Merci donc pour les discussions interminables sur des sujets toujours très variés et parfois inattendus. Merci pour les pauses cafés, les afterworks, les repas, les futsals, la transmission des légendes... Si je garde un bon souvenir de ma thèse, c'est en grande partie grâce à vous !

Je fais une dédicace spéciale à Nicolas qui a trouvé le cadeau de thèse le plus stylé possible³. Merci pour les week-ends à Toulouse (ou à Bordeaux). Merci également à Korantin et Adélie pour les moments passés ensemble et pour la coloc temporaire fort sympathique qui m'a permis

1. Voir traduction du mot *advisor*.

2. Au moment où j'écris ces lignes, j'ai commencé à travailler à l'INRIA et développe une plateforme de visualisation.

3. Cette remarque est tout à fait objective...

de préparer au mieux ma soutenance. Sans oublier les chats de la coloc (Quiqui tmtc). J'en profite pour saluer Jean-Marie qui est également venu à Toulouse malgré la longue route!

Merci aux « Bee Gees » : Broq', Chachou, Séb, Anne ($S \cap A = \emptyset$, $S \cup A = B$, qui l'a?), Kiki, Doudou, Flo, Marion, Lourdeau, Kelly, Jojo, Emma, Gabi, Valou, Pignet, qu'ils soient à Paris (plus ou moins 25 kms), Bordeaux, Toulouse, Voiron, Gramat, Cannes, Montréal, Saint Pierre... pour tous les moments géniaux que l'on passe ensemble.

Merci à tous mes autres amis toulousains et en particulier Francis, Kazu et Chopp. Je pense également à la coloc bordelaise et aux bordelais : Lucas, Marie, Tube, Clem, Gentien, Jean (Quoi? Il n'habite pas à Bordeaux?!). Les soirées, les jeux de société, les randonnées dans les Pyrénées m'ont souvent aidé à me ressourcer pendant ces trois années. Merci à mes amis de Mysterb, et en particulier Biture, Marie, Dédé, Willou, Valou...

Merci à mes amis normands Steeve, Marie (et Rose!), Manu, Gaët pour les diners, les belotes...

Enfin, je remercie ma famille, et en particulier mon frère, ma sœur et mes parents, pour leur soutien inconditionnel même si les problématiques qui m'ont intéressé pendant ma thèse (et mes études en général) ont pu leur paraître si obscures, si lointaines...

Table des matières

| | | |
|----------|---|-----------|
| 1 | Introduction | 11 |
| 1.1 | Les méthodes numériques de résolution d'EDP | 11 |
| 1.2 | Les logiciels de visualisation | 12 |
| 1.3 | La visualisation de résultats de simulations d'ordre élevé | 17 |
| 1.3.1 | L'approche naïve pour visualiser ces résultats | 18 |
| 1.3.2 | La création d'un logiciel de visualisation dédié aux méthodes d'ordre élevé | 18 |
| 1.3.3 | Le remaillage en une solution de bas ordre | 21 |
| 1.4 | Objectifs de cette thèse | 27 |
| 1.5 | Organisation du manuscrit | 28 |
| 2 | Conception d'une méthode de visualisation assurant le contrôle ponctuel des erreurs de visualisation sur des éléments droits | 31 |
| 2.1 | Formalisation des objectifs de la méthode de visualisation | 32 |
| 2.2 | Cadre des méthodes d'ordre élevé | 33 |
| 2.3 | Conception d'une méthodologie pour répondre aux objectifs | 36 |
| 2.3.1 | Réalisation de l'objectif 1 | 36 |
| 2.3.2 | Réalisation de l'objectif 2 | 38 |
| 2.3.3 | Réalisation de l'objectif 3 | 40 |
| 2.3.4 | Résumé de la méthodologie | 44 |
| 2.4 | Justifications mathématiques de la méthode proposée | 45 |
| 2.4.1 | Réalisation de l'objectif 1 | 45 |
| 2.4.2 | Réalisation de l'objectif 2 | 47 |
| 2.4.3 | Réalisation de l'objectif 3 | 48 |
| 2.5 | Algorithme pour générer une représentation satisfaisant les objectifs | 51 |
| 2.5.1 | Un algorithme de recherche d'extrema : DIRECT | 51 |
| 2.5.2 | Algorithme complet pour les éléments droits | 54 |
| 2.5.3 | Théorème sur la réponse aux objectifs | 56 |
| 2.6 | Conclusions du chapitre | 57 |
| 3 | Extension de la méthode de visualisation au traitement des surfaces courbes | 59 |
| 3.1 | Problèmes liés à l'utilisation d'éléments courbes | 60 |
| 3.2 | Extension de la méthode aux éléments courbes | 62 |
| 3.2.1 | Identification et caractérisation des zones de retournements | 63 |
| 3.2.2 | Construction d'un estimateur <i>a posteriori</i> spécifique | 74 |
| 3.3 | Modifications de l'algorithme générant la représentation | 76 |
| 3.3.1 | Modifications dans l'algorithme général | 76 |
| 3.3.2 | Théorème de convergence pour les éléments courbes | 78 |
| 3.3.3 | Problème d'optimisation pour calculer le paramètre d'avancement maximal | 78 |

| | | |
|----------|---|------------|
| 3.3.4 | Utilisation de DIRECT pour estimer l'indicateur spécifique | 80 |
| 3.4 | Extension au traitement des surfaces courbes dans l'espace | 87 |
| 3.5 | Conclusions du chapitre | 90 |
| 4 | Validations numériques de la méthode de visualisation | 91 |
| 4.1 | Validations pour le traitement des éléments droits | 92 |
| 4.1.1 | Comparaison des contrôles en normes L^2 et L^∞ | 92 |
| 4.1.2 | Comparaison du raffinement adaptatif avec un raffinement uniforme : Fonction de base Gauss Q^3 | 94 |
| 4.1.3 | Comparaison du raffinement adaptatif avec un raffinement uniforme : Pro- blème d'élasticité résolu par la méthode Galerkin Discontinue | 95 |
| 4.1.4 | Exemples volumiques : modes sur hexaèdres. | 102 |
| 4.2 | Validations pour le traitement des éléments courbes | 104 |
| 4.2.1 | Fonction de base Gauss-Lobatto Q^6 sur un quadrangle Q^2 | 104 |
| 4.2.2 | Retournements dans l'espace | 107 |
| 4.2.3 | Problème de propagation d'onde acoustique. | 109 |
| 4.3 | Amélioration de la qualité des éléments en considérant le voisinage de la frontière | 111 |
| 4.4 | Conclusions du chapitre | 115 |
| 5 | Reconstruction et visualisation de champs rayonnés pour des ondes harmo- niques | 117 |
| 5.1 | Positionnement du problème | 118 |
| 5.2 | Construction de cartographies | 121 |
| 5.2.1 | Rayonnements sur grilles cartésiennes | 122 |
| 5.2.2 | Rayonnements par interpolations d'ordre élevé | 122 |
| 5.3 | Compression de l'information : interpolation directionnelle | 124 |
| 5.3.1 | Rayonnements par interpolations du noyau de Green | 124 |
| 5.3.2 | Interpolation directionnelle avec une direction | 125 |
| 5.3.3 | Interpolation directionnelle avec plusieurs directions | 127 |
| 5.4 | Construction de champs rayonnés | 128 |
| 5.4.1 | Subdivision du domaine à visualiser | 128 |
| 5.4.2 | Subdivision de la frontière de l'objet diffractant | 128 |
| 5.4.3 | Critères de construction des champs rayonnés | 129 |
| 5.4.4 | Visualisation des champs reconstruits | 132 |
| 5.5 | Un algorithme pour reconstruire des champs rayonnés | 132 |
| 5.5.1 | Critère d'admissibilité | 133 |
| 5.5.2 | Subdivision du domaine | 134 |
| 5.5.3 | Assignation des ordres d'interpolation. | 136 |
| 5.6 | Validations numériques des champs rayonnés reconstruits. | 138 |
| 5.6.1 | Interpolation directionnelle avec une seule direction | 138 |
| 5.6.2 | BEM en 2D et rayonnements dans le plan | 139 |
| 5.6.3 | BEM en 3D et rayonnements dans des plans | 148 |
| 5.6.4 | BEM en 3D et rayonnements dans des volumes | 149 |
| 5.7 | Conclusions du chapitre | 155 |
| 6 | Conclusions et perspectives | 159 |

| | | |
|----------|---|------------|
| A | La génération de maillages en simplexes pour créer un maillage de représentation | 165 |
| A.1 | Les méthodes basées sur des arbres | 166 |
| A.2 | Les méthodes par avancée de front | 168 |
| A.3 | Les méthodes de maillage de Delaunay | 168 |
| A.4 | Bilan sur les techniques de génération de maillage | 170 |
| B | Fonctions de base et transformations géométriques usuelles | 171 |
| B.1 | Interpolation de Lagrange, polynômes de Gauss, de Gauss-Lobatto, de Chebyshev | 171 |
| B.2 | Quelques transformations géométriques usuelles | 176 |
| B.3 | Quelques fonctions de base usuelles | 182 |
| C | Détails des calculs de la section 3.3.3 | 185 |
| C.1 | L'algorithme de Gauss-Newton | 185 |
| C.2 | Rappels de calcul différentiel | 186 |
| C.3 | Cas où les surfaces se trouvent dans le plan | 187 |
| | Table des figures | 191 |
| | Bibliographie | 197 |

Notations

Sur la solution numérique :

f_{num} : Solution numérique.

d : Dimension du problème.

X : Domaine de calcul : sous-espace d'un espace $\mathbb{X} = \mathbb{R}^d$.

\mathbb{Y} : \mathbb{K} -espace vectoriel où sont définis les champs de la solution numérique.

\mathcal{T}_h : Maillage de calcul.

K : Éléments constituant le maillage de calcul.

\widehat{K} : Cellule de référence associée à un élément K .

g_K : Transformation géométrique bijective reliant \widehat{K} à K (cf. figure 2.1).

f_{num}^K : Solution numérique locale à K (cf. (2.1)).

Sur la représentation :

\widetilde{K} : Approximation d'un élément K (cf. (2.4)).

$f_{vis}^{\widetilde{K}}$: Représentation locale (cf. (2.3)).

$\mathcal{T}(\widehat{K})$: Maillage de représentation de la cellule de référence \widehat{K} (cf. figure 2.7).

$\mathcal{T}(K)$: Maillage de représentation de l'élément K (cf. figure 2.7).

$P^1 g_{\mathcal{T}(K)}$: Approximation de la transformation géométrique (cf. figure 2.2).

$d_H^{\mathbb{X} \times \mathbb{Y}}$: Distance de Hausdorff sur $\mathbb{X} \times \mathbb{Y}$ (cf. (2.6)).

α, β : Coefficients de pondération définis dans la distance de Hausdorff (cf. (2.8)).

$\Delta_{\mathcal{T}(K)}$: Indicateur d'erreurs *a posteriori* pour un élément droit K (cf. (2.9)).

Γ_S^K : Indicateur d'erreurs *a posteriori* pour une arête S courbe associée à un élément K (cf. (3.9)).

$\mathcal{T}_i D$: Éléments du maillage de calcul de dimension i (cf. (2.14)).

$\mathcal{T}_3 D$: Ensemble des volumes (cf. (2.15)).

$\mathcal{T}_2 D$: Ensemble des faces (cf. (2.16)).

$\mathcal{T}_1 D$: Ensemble des arêtes (cf. (2.17)).

Abréviations :

BEM : Boundary Element Method (Méthode des éléments finis de frontière).

DG : Discontinuous Galerkin (Méthode de Galerkin Discontinue).

FD : Finite Differences (Méthode des différences Finies).

FEM : Finite Element Method (Méthode des éléments finis).

FVM : Finite Volume Method (Méthode des volumes finis).

Chapitre 1

Introduction

La simulation numérique permet de prédire le comportement de phénomènes physiques sans avoir à passer par des prototypes / expérimentations : crash tests pour l'automobile, essais dans le nucléaire ou l'aérospatial par exemple. En effet, ces essais peuvent être coûteux ou difficilement réalisables. La simulation est donc un enjeu important dans de nombreux domaines d'applications aux niveaux académique, industriel et militaire. Parmi les applications, on peut notamment citer la mécanique des fluides ou des solides (résistance des structures, optimisation de formes pour l'automobile ou l'aéronautique par exemple), la modélisation thermique, l'acoustique ou encore l'électromagnétisme (problèmes de furtivité radar)...

La modélisation de nombreux phénomènes physiques fait appel à la résolution d'équations aux dérivées partielles (EDP). Afin de simuler un système complexe, la chaîne classique de calcul est composée de trois étapes définies sur la figure 1.1. Tout d'abord, un pré-traitement est nécessaire afin de générer la géométrie et le maillage de calcul. Pour cela, un logiciel de CAO (Conception Assistée par Ordinateur) peut être utilisé pour construire la géométrie. Un mailleur permet de créer le maillage. La deuxième étape est la résolution du problème où la méthode numérique utilisée s'appuie sur ce maillage. Enfin, la dernière étape concerne les post-traitements des résultats. La visualisation est une brique importante qui permet d'analyser les résultats mais également facilite leur communication.

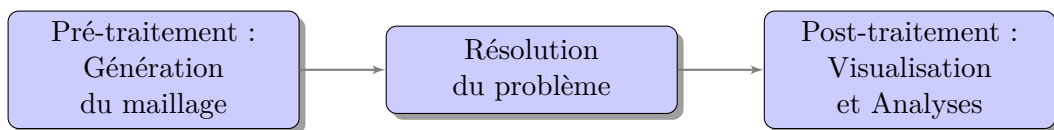


FIGURE 1.1 – La chaîne classique de calcul

1.1 Les méthodes numériques de résolution d'EDP

Il existe plusieurs familles de méthodes numériques pour résoudre des EDP. La méthode des différences finies [Forsythe and Wasow, 1960, Smith, 1985, Richtmyer and Morton, 1994], abrégée FD pour *Finite Differences*, fut à l'origine élaborée sur des maillages cartésiens, où le pas de maillage est constant, et la solution est construite uniquement à l'aide des nœuds du maillage. L'intérêt de cette méthode est qu'elle est facile à mettre en place mais la géométrie du domaine de calcul doit être simple. Pour des géométries plus complexes, la méthode des éléments

finis [Zienkiewicz and Taylor, 1977, Ciarlet, 1978], appelée FEM pour *Finite Element Method*, est plus adaptée. Celle-ci fut introduite avec des éléments finis P^1 , c'est-à-dire des fonctions de base de degré 1 construites à partir des nœuds du maillage. Dans ce cas, les éléments du maillage sont des simplexes, à savoir des segments, des triangles ou des tétraèdres suivant la dimension. Pour traiter des lois de conservations, la méthode des volumes finis [LeVeque, 2002], notée FVM pour *Finite Volume Method*, est particulièrement adaptée. Ceci est dû à la façon dont ces schémas ont été introduits qui sont naturellement conservatifs. Pour des domaines de calcul infinis, la méthode des éléments finis de frontière [Nédélec, 2001, Colton and Kress, 2012] appelée BEM pour *Boundary Element Method* est particulièrement intéressante. Dans leurs versions originelles, les solutions obtenues par toutes ces méthodes étaient de bas ordre.

La complexité croissante des phénomènes à simuler nécessite le développement de codes de calcul toujours plus performants (précision, coût...). Dans cette optique, des méthodes dites *hp* pour *hautes précisions* (polynômes d'ordre élevé, fonctions spéciales...) ont été proposées et analysées ces dernières décennies : méthodes des éléments finis d'ordre élevé, spectrales, isogéométriques [Hughes et al., 2005], ou encore de Galerkin Discontinue d'ordre élevé DG pour *Discontinuous Galerkin* [Reed and Hill, 1973, Cockburn et al., 2000a, Arnold et al., 2002]. Le principe de ces méthodes est d'avoir des fonctions de base plus riches que celles de bas ordre. Souvent, mais ce n'est pas toujours le cas, les données de ces méthodes sont polynomiales avec un degré, ou ordre, qui peut devenir élevé.

L'essor de ces méthodes implique l'émergence de nouveaux besoins en terme d'exploitation et de post-traitement des résultats. Or, les logiciels standards dédiés à cette tâche n'ont pas assez évolué afin d'exploiter toute la richesse donnée par ces nouvelles méthodes.

1.2 Les logiciels de visualisation

Pour alimenter les logiciels de visualisation, comme ParaView [Ayachit, 2015], Gmsh [Remacle et al., 2007], VisIt [Childs et al., 2012] ou encore Tecplot, les données de la solution numérique (sur les fonctions et sur la géométrie) sont transmises.

Il existe classiquement deux modes de fonctionnement des logiciels de visualisation :

- Le mode « valeurs nodales » : on fournit dans ce cas au logiciel des valeurs de la solution numérique sur une grille de points ou à certains nœuds, qui peuvent être différents des cellules géométriques (triangles, quadrangles...). Par exemple, pour des fonctions d'ordre deux, les centres des arêtes sont utilisés comme degrés de liberté. Ce mode de représentation est naturellement adapté aux méthodes FD/FEM/DG.
- Le mode « valeurs centrées aux cellules » : le fichier est alors composé de cellules géométriques simples auxquelles des constantes de la solution numérique sont associées. La représentation est alors constante par morceaux. Ceci est par exemple bien adapté pour les FVM.

Suivant la nature des données, les formats varient en deux types : pour des maillages structurés (sous forme de grille où le pas est régulier) ou non-structurés.

Lorsque le maillage est structuré, aucune table de connectivité n'est nécessaire. Les éléments sont repérés par des indices et la connaissance des voisins se fait simplement en incrémentant ou décrémentant ces indices. Il suffit de connaître le nombre de points et de donner les valeurs (à chaque nœud ou associées à un élément, suivant le mode considéré) correspondantes aux points.

Lorsque le maillage est non structuré, plus d'informations sont nécessaires. En général, trois blocs permettent de décrire la solution numérique. Dans un premier bloc, les coordonnées des

nœuds sont données. Un deuxième bloc explicite la connectivité associée à chaque élément. Suivant le type d'élément, le nombre de nœuds varie. Par exemple, pour un quadrangle d'ordre un, 4 nœuds interviennent tandis que pour l'ordre deux, il y en a 9 (les 4 sommets, les 4 centres des arêtes et le centre de la face). Dans ce dernier cas, l'élément considéré n'est plus un quadrangle, puisqu'un quadrangle est un quadrilatère, mais nous utilisons quand même l'appellation quadrangle d'ordre élevé. Le dernier bloc attribue les valeurs associées à chaque nœud.

A partir de toutes les informations qu'on lui donne, le logiciel de visualisation représente la solution numérique sur tout le domaine. Pour cela, des interpolations affines sont généralement réalisées. Lorsque l'on visualise une fonction affine sur un simplexe, la représentation est toujours exacte : le rendu graphique coïncide avec la fonction à visualiser. La figure 1.2 illustre cela pour une fonction affine définie sur un triangle quelconque avec ParaView (tous les résultats provenant de ParaView ont été générés avec la version 3.14).

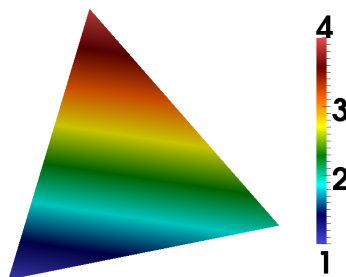


FIGURE 1.2 – Exemple de représentation d'une fonction affine sur un triangle avec ParaView 3.14.

Cependant, même dans des configurations simples que n'importe quel logiciel de visualisation est capable de traiter, le rendu graphique peut être mauvais. C'est le cas lorsque l'on s'intéresse à la visualisation de champs sur une grille cartésienne pour la méthode des différences finies par exemple. Comme nous l'avons précisé ci-dessus (mode 1), le logiciel doit souvent se contenter des « pauvres » informations qui lui sont données pour créer une représentation affine sur simplexes. Cette façon de faire peut induire des biais dans l'interprétation des résultats.

Pour illustrer cela, nous nous intéressons à la représentation de fonctions définies sur le carré unité à partir de valeurs aux 4 sommets. Nous nous focalisons sur 4 cas, illustrés sur la figure 1.3, où à chaque fois les valeurs de la fonction valent un à un sommet et sont nulles aux trois autres. A partir des valeurs aux nœuds, le logiciel génère le rendu sur tout l'élément. Les représentations obtenues sur la figure 1.4 avec ParaView sont très différentes alors qu'il suffirait *a priori* d'effectuer une rotation autour du centre du carré pour obtenir les images suivantes à partir de la première. On en déduit qu'implicitement, le carré est découpé selon une diagonale en deux triangles et une interpolation affine est effectuée sur chaque triangle. Ce découpage, bien qu'invisible et non contrôlé par l'utilisateur, est finalement logique puisqu'un logiciel de visualisation effectue des interpolations affines. Une interpolation affine est unique sur un triangle mais il n'y a pas unicité sur un carré. En conséquence, un logiciel traite ce genre d'élément mais le rendu obtenu peut être différent de ce qui est attendu.

Par rapport aux informations données au logiciel de visualisation (4 valeurs sur un carré), plusieurs représentations du cas 1 seraient admissibles. Sur la figure 1.5, deux configurations de découpage ont été effectuées. A chaque fois, on fournit au logiciel les données sur des triangles. Suivant le sens de la diagonale de découpage, la représentation varie mais les valeurs aux sommets sont respectées. Le choix de découpage (ou non), de l'élément a donc un impact important sur

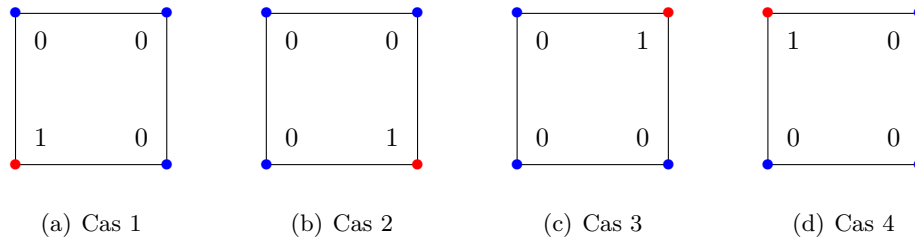


FIGURE 1.3 – Les 4 cas où une fonction est définie par les valeurs (0 ou 1) aux sommets sur le carré unité.

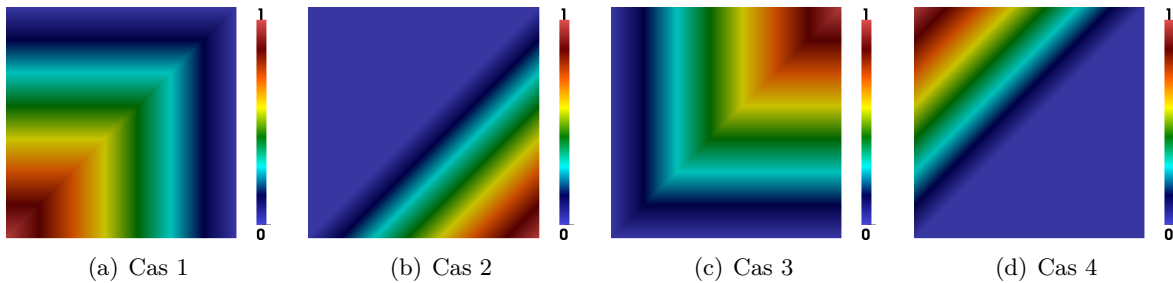


FIGURE 1.4 – Représentation avec ParaView 3.14 sur le carré unité de différents cas où la fonction est définie par les valeurs aux sommets.

le rendu final.

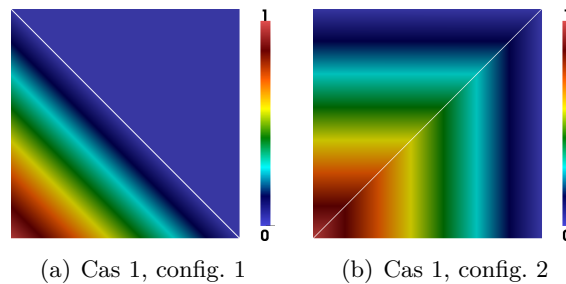


FIGURE 1.5 – Deux représentations possibles du cas 1 de la figure 1.3.

Le logiciel de visualisation devrait être consistant sur la représentation, en respectant la rotation observée. Si pour le cas 1, la représentation obtenue est celle donnée dans la figure 1.5.a (resp. figure 1.5.b), les suivantes devraient être obtenues par rotation comme sur la figure 1.6 (resp. figure 1.7). Pour respecter cette rotation, les configurations de découpe alternent. Deux observations apparaissent. Si on souhaite être consistant avec cette rotation, il est nécessaire d'adapter la configuration de découpage, c'est-à-dire le maillage de représentation. Deuxièmement, il fallait savoir *a priori* quelle représentation on souhaite obtenir : figure 1.5.a ou 1.5.b ? En effet, il n'y a pas assez d'informations données au logiciel de visualisation. Ainsi, les deux rendus sont admissibles mais ne correspondent pas nécessairement à ce que l'on souhaitait visualiser.

Les différents cas proposés par la figure 1.3 correspondent typiquement à une fonction Q^1

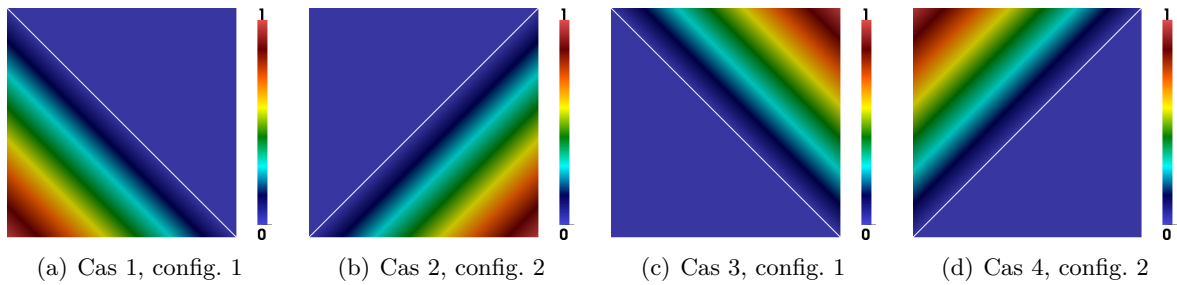


FIGURE 1.6 – Représentations obtenues des 4 cas en suivant la figure 1.5.a.

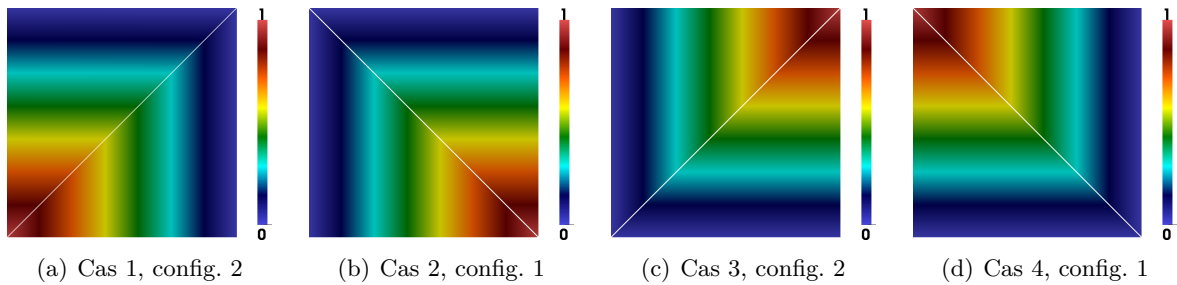
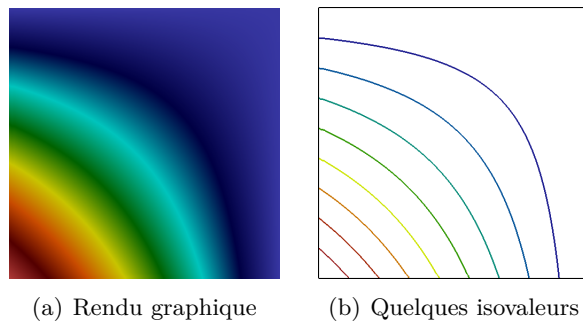


FIGURE 1.7 – Représentations obtenues des 4 cas en suivant la figure 1.5.b.

sur un carré, qui est donc définie à partir des 4 valeurs aux sommets et est illustré sur la figure 1.8. Dans ce cas, la fonction n'est pas affine, et aucune des représentations proposées jusqu'à présent n'est correcte. De plus, la figure 1.8.b montre des isovaleurs pour cette fonction qui, on va le voir, sont différentes des isovaleurs obtenues dans les autres configurations à cause de la non-linéarité de la fonction. Pour que ces données soient satisfaisantes, il aurait fallu donner plus d'informations au logiciel de visualisation.

FIGURE 1.8 – Représentation d'une fonction Q^1 sur le carré unité.

Outre la représentation, un aspect central dans le post-traitement est l'extraction des données. Sur la figure 1.9, des isovaleurs (ou courbes de niveaux ou *isolines*), ainsi que des gradients ont été extraits des représentations obtenues sur le carré (figure 1.4). Plusieurs incohérences apparaissent. Tout d'abord, les isovaleurs ne correspondent pas avec les représentations pour les cas 1 et 3. En effet, les isovaleurs affichées sont des segments alors que d'après les figures 1.4.a

et 1.4.c, deux segments devraient apparaître pour chaque isovaleur formant un angle droit. Des gradients sont également visibles. Ceux-ci ne sont pas cohérents avec les isovaleurs. Sur les figures 1.9.b et 1.9.d, les 3 gradients devraient être colinéaires et suivre une diagonale mais ce n'est pas le cas pour deux d'entre eux. Tous ces post-traitements sont en fait effectués indépendamment les uns des autres.

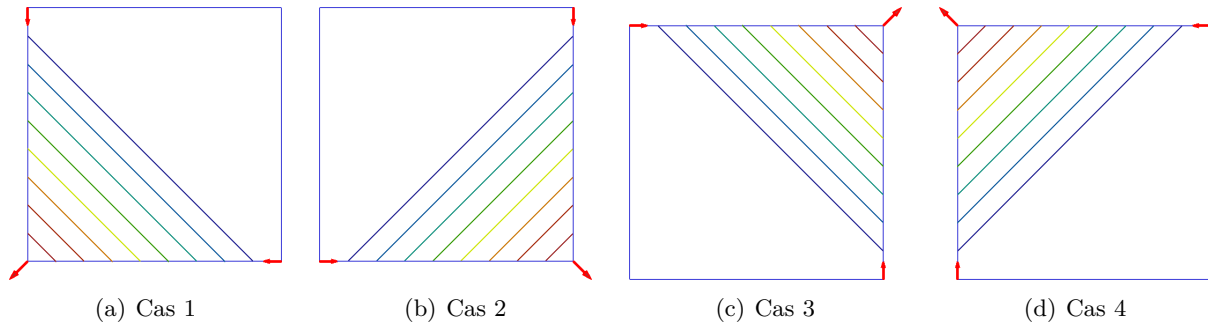


FIGURE 1.9 – Post-traitements dans ParaView 3.14 de la figure 1.4 où la fonction est définie par les valeurs aux sommets du carré unité. Des isovaleurs ainsi que des gradients (en rouge) sont affichés.

La génération des isovaleurs est reprise sur la figure 1.10. Nous comparons les différences lorsque le carré est utilisé avec les deux configurations où le carré est découpé en deux pour le cas 1. Comme nous l'avons remarqué précédemment, les isovaleurs de la figure 1.10.a ne correspondent pas à la représentation. La raison vient de la façon dont les isovaleurs sont construites. En effet, les valeurs recherchées sont extraites des arêtes de chaque élément. Comme la fonction est supposée affine, il est facile de trouver, sur chaque arête, l'éventuelle valeur cherchée : il suffit de la comparer aux valeurs des extrémités de chaque arête. A partir des valeurs trouvées sur ces arêtes, et comme la fonction est supposée être affine sur l'élément, les isovaleurs sont construites comme des segments. Dans les deux autres configurations, les isovaleurs sont bâties de la même manière. La différence est que, comme les éléments sont des simplexes, les isovaleurs correspondent bien à la représentation. Lorsque des simplexes sont utilisées, les post-traitements effectués sont cohérents avec les représentations. Ceci est important car dans le cas contraire, l'interprétation peut être entachée d'erreurs de jugement.

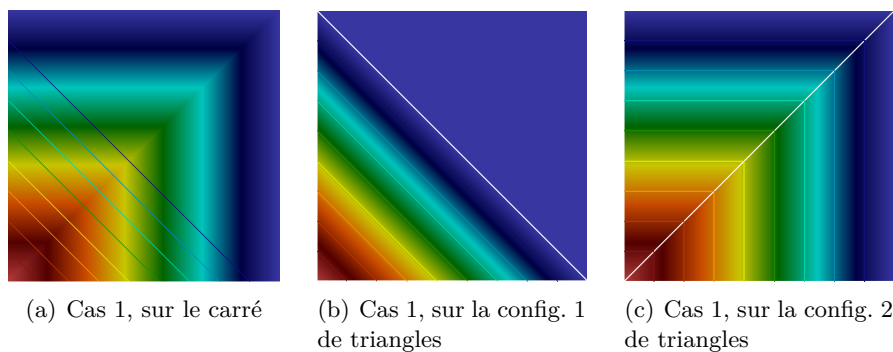


FIGURE 1.10 – Comparaison des représentations et isovaleurs obtenues pour le cas 1 sur le carré unité et deux configurations de triangles.

La nécessité d'utiliser des simplexes apparait dans la plupart des logiciels de visualisation. C'est également le cas de Gmsh, qui est illustré sur la figure 1.11. Nous avons représenté sur le carré unité les 4 cas où la fonction est définie à partir des sommets. Le redécoupage en deux triangles est bien visible sur la figure 1.11. Contrairement à ParaView, les isovaleurs correspondent à la représentation générée. Le redécoupage, est pris en compte lors de leurs générations.

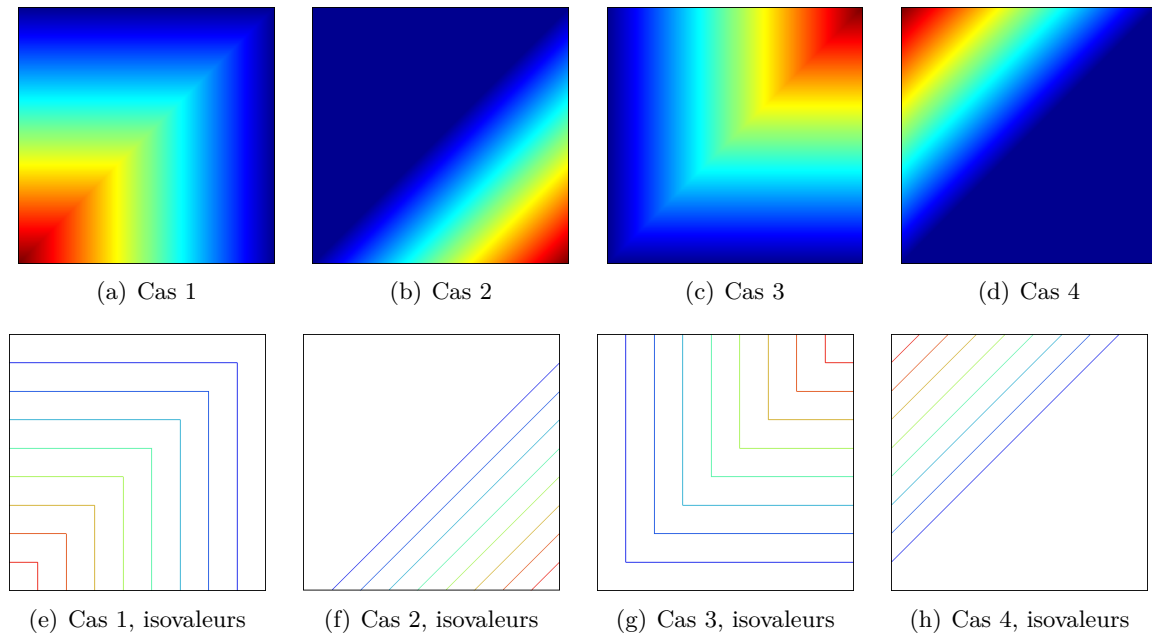


FIGURE 1.11 – Représentation avec Gmsh 2.4 sur le carré unité de différents cas où la fonction est définie par les valeurs aux sommets.

La conséquence est qu'il est crucial d'utiliser des simplexes pour générer des représentations et des post-traitements. Toutes ces problématiques sont également présentes en dimension 3.

1.3 La visualisation de résultats de simulations d'ordre élevé

La plupart des logiciels de visualisation ne sont pas capables de prendre en charge des résultats issus de méthodes d'ordre élevé. En effet, certains cas peuvent être traités par quelques logiciels de visualisation mais lorsque les degrés polynomiaux sont (très) élevé, en général il n'existe pas de format pour les gérer. Par exemple, actuellement, Gmsh dans sa version 4.0 est capable de traiter des éléments jusqu'à l'ordre 5 (voir documentation [Geuzaine and Remacle, 2008]). Avec la librairie VTK version 4.2, The Visualization Toolkit [Schroeder et al., 2004], des éléments (et fonctions) quadratiques peuvent être traités mais il n'est pas possible d'utiliser des données d'ordre supérieur à 2. Tecplot (version 2018) ne se limite qu'à des éléments de bas ordre : les éléments quadratiques ne sont pas gérés. Dans un cadre plus large, plusieurs stratégies, que nous détaillons maintenant, permettent de visualiser des résultats provenant de simulations d'ordre élevé.

1.3.1 L’approche naïve pour visualiser ces résultats

Pour visualiser des solutions hp , une approche « naïve » consiste à créer une grille de représentation et y projeter la solution numérique. Plus précisément, chaque élément du maillage est découpé de manière régulière et en chaque point de la grille, la valeur de la solution numérique est calculée. A partir de ces valeurs et à l’aide d’interpolations affines effectuées par le logiciel de visualisation, une approximation de la solution numérique peut être tracée. Une erreur de visualisation, correspondant à l’écart entre la solution numérique et son approximation affichée à l’écran, est ainsi introduite.

La figure 1.12 représente un polynôme de degré trois à l’aide de différentes grilles de représentation. Le problème est de savoir comment fixer le paramètre de redécoupage de chaque élément (i.e. combien de fois doit-on le découper). Si la grille est grossière, comme sur la colonne de gauche de la figure 1.12 avec une grille 4×4 , la représentation risque d’être médiocre voire très mauvaise. Le tableau 1.1 quantifie les erreurs de visualisation générées par différentes grilles cartésiennes. Une grille fine permet d’avoir une « bonne » représentation mais demande beaucoup d’efforts en temps CPU et coûts mémoire. L’erreur de visualisation est mal contrôlée puisque les grilles de visualisation sont définies *a priori*. Ce manque de maîtrise, à la fois sur la qualité de la représentation et sur la quantité de données générées, devrait être contrôlé pour avoir une méthode de visualisation efficace. Notons que pour les raisons évoquées précédemment, même si des grilles sont définies, un découpage en simplexes est utilisé (deuxième ligne de la figure 1.12).

| Grille de représentation | 5×5 | 10×10 | 20×20 | 50×50 |
|--------------------------|--------------|----------------|----------------|----------------|
| Erreur relative | 42.9 % | 19.2 % | 6.5 % | 1.2 % |

TABLE 1.1 – Erreurs de visualisation (erreurs d’interpolation en norme L^∞) des représentations obtenues sur la figure 1.12 sur différentes grilles de représentation.

1.3.2 La création d’un logiciel de visualisation dédié aux méthodes d’ordre élevé

Une première « vraie » stratégie pour visualiser des résultats de méthodes d’ordre élevé est de créer un logiciel dédié qui prend nativement en compte leurs formats. L’intérêt majeur vient du fait que, contrairement à l’approche précédente, il n’y a pas d’erreur de visualisation à contrôler et donc pas d’artefacts de représentation dus à celle-ci.

La plupart des systèmes de visualisation dédiés aux méthodes d’ordre élevé sont basés sur la représentation pixel exacte ou *pixel exact rendering*. Le principe est d’associer à chaque pixel de l’image, sa couleur calculée exactement. Pour cela, des lancers de rayons ou *ray casting* peuvent être utilisés [Levoy, 1988, Glassner, 1989, Levoy, 1990, Ray et al., 1999]. La figure 1.13, extraite de [Levoy, 1990] illustre la technique. Sur la figure 1.13, ces rayons sont parallèles et régulièrement espacés. Chaque rayon est à son tour découpé régulièrement de sorte qu’en chaque pixel, la couleur et l’opacité sont calculées à partir des données (valeurs de la solution à visualiser).

Des chercheurs à l’université d’Utah ont développé le logiciel *The Element Visualizer* (EIVis) qui est conçu spécifiquement pour visualiser des solutions scalaires éléments finis d’ordre élevé [Nelson et al., 2012, Nelson, 2012, Peiro et al., 2015]. EIVis n’introduit pas d’erreur sur l’image générée puisque les algorithmes sont directement basés sur la connaissance de la solution d’ordre élevé. La méthode est basée sur la résolution de deux problèmes non-linéaires (la recherche de

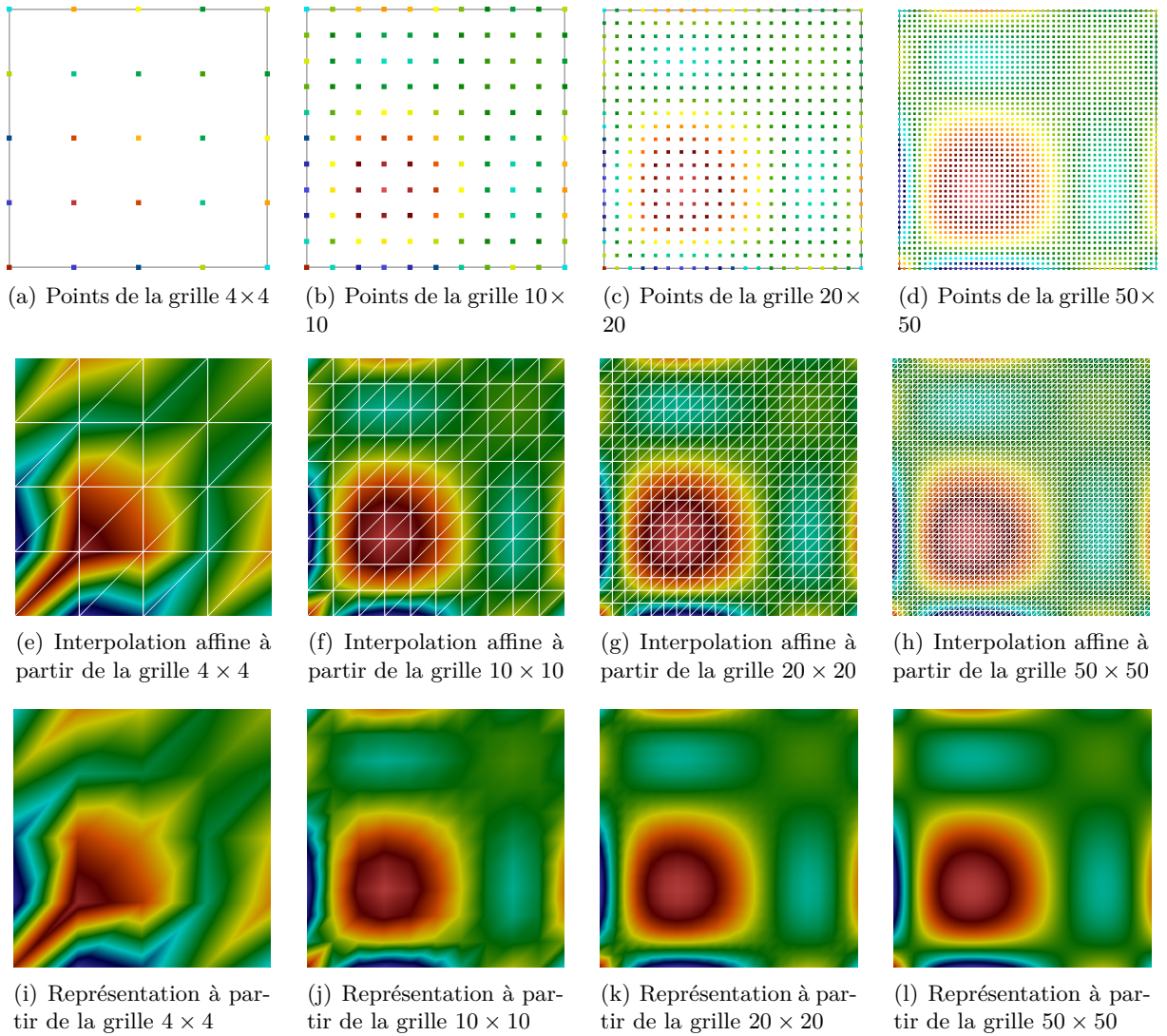


FIGURE 1.12 – Représentation sous ParaView d'un polynôme d'ordre élevé sur différentes grilles de visualisation.

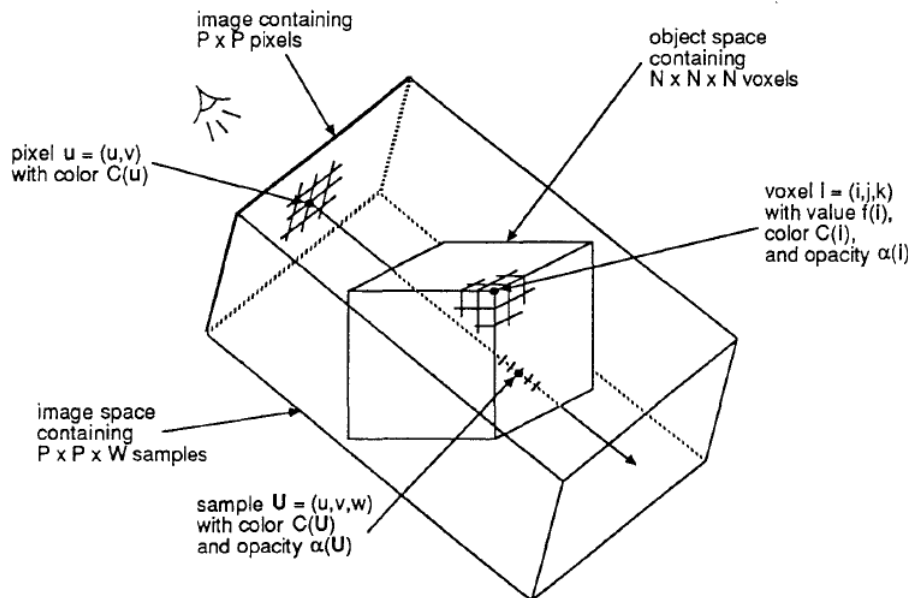


FIGURE 1.13 – Illustration du *Ray Tracing*. Image extraite de [Levoy, 1990].

racines et l'inversion de la transformation géométrique) qui sont très coûteux. Pour cette raison, la puissance des cartes graphiques, *Graphics Processing Units* (GPU) est utilisée. Néanmoins, en terme de ressources de calculs nécessaires, les auteurs [Nelson et al., 2012] expliquent que les coûts liés à l'utilisation de données d'ordre élevé dans un logiciel dédié sont beaucoup plus grands que si on utilise des simples interpolations affines. Or, l'interactivité est capitale lors de l'exploitation de donnée volumiques. En effet, si l'environnement (le domaine de calcul) est tridimensionnel, comme la représentation à l'écran est bidimensionnelle, celle-ci n'est compréhensible que dans un cadre interactif où l'utilisateur peut interagir avec son modèle.

D'autres travaux utilisent la représentation pixel exacte. Les travaux de Zhou et al. [Zhou et al., 2004, Zhou and Garland, 2006] portent sur la visualisation de solutions d'ordre élevé sur des tétraèdres et ils ont développé une méthode pour évaluer de manière pixel exacte la solution en utilisant des GPU. Le rendu sur tétraèdres basé sur des GPU a également été étudié par Weiler et al. [Weiler et al., 2003]. Uffinger et al. [Üffinger et al., 2010] utilisent du *raycasting* sur GPU pour visualiser des éléments finis d'ordre élevé volumiques. La recherche d'un post-traitement à la fois précis et interactif est crucial. Dans cette optique, des travaux très récents de Loseille et al. [Loseille and Feuillet, 2018] utilisent OpenGL 4.0 [Wolff, 2011] afin d'avoir un rendu pixel exact lorsque des éléments droits sont considérés ou presque pixel exact (*almost pixel-exact rendering*) dans le cas d'éléments courbes. Leurs travaux sont possibles grâce à la nouvelle version d'OpenGL. En effet, l'ancienne version n'aurait pas été capable de gérer leurs problèmes à cause d'une mémoire insuffisante. Il faut noter qu'il n'y a pas de lancer de rayon dans ce cas.

Outre les coûts, une autre difficulté vient des lourds développements nécessaires pour créer un logiciel complet ayant également à sa disposition tous les outils (*plugins*) nécessaires au post-traitement des résultats. Ces derniers sont tout aussi importants que le rendu graphique. On peut lister de manière non exhaustive :

- L'extraction d'isovaleurs (lignes de niveau).
- La génération de lignes de courant (*streamlines* en anglais). Une ligne de courant est une courbe qui en chacun de ces points est tangente au vecteur vitesse. C'est un outil parti-

culièrement intéressant en mécanique des fluides qui permet de représenter la trajectoire d'une ou plusieurs particules se déplaçant dans un écoulement.

- Le calcul de quantités usuelles telles que le gradient, le rotationnel ou la divergence. De manière plus générale, toutes les opérations qui peuvent être faites sur les données sont intéressantes comme par exemple le calcul d'une énergie à partir de plusieurs champs.
- La génération de coupes (*clip* ou *slice*). C'est une fonctionnalité indispensable en dimension 3 ainsi que le fait de pouvoir faire ces coupes en temps réel dans le logiciel.

Pour la plupart de ces outils, le fait de pouvoir les utiliser *à la volée* est primordial et donc directement dans le logiciel de visualisation. Or, dans le cadre des lancers de rayons, il est nécessaire de relancer des calculs pour effectuer les post-traitements comme pour extraire une valeur. A contrario, dans un logiciel de visualisation standard, grâce à la notion d'élément, et comme les fonctions sont affines, ce post-traitement est immédiat car les algorithmes sont simples et robustes.

1.3.3 Le remaillage en une solution de bas ordre

Pour pallier les problèmes de l'approche naïve, une autre stratégie, dite de *low order remeshing*, consiste à construire une approximation en bas ordre de la solution numérique en contrôlant l'erreur de visualisation afin de garantir la pertinence du rendu graphique. Pour cela, un sous-maillage, dit maillage de représentation, est construit pour chaque cellule du maillage de calcul, à partir duquel une interpolation de bas ordre de la fonction à visualiser est définie. Ce type de construction est basée sur un raffinement de maillage auto-adaptatif guidé par l'erreur d'interpolation, dite erreur de visualisation dans ce contexte. L'approximation générée est ainsi constituée de fonctions affines et peut ainsi être post-traitée par un logiciel de visualisation standard efficacement et interactivement. Ainsi, un utilisateur peut se servir d'un logiciel de visualisation habituel qui a l'avantage d'être robuste et mature dans ce cadre : tous les outils de post-traitement sont disponibles. Pour cette stratégie, une étape supplémentaire est donc nécessaire avant le post-traitement. Elle consiste à « convertir » les résultats obtenus par le code de calcul en données exploitables pour un logiciel de visualisation standard. Par rapport à la chaîne de calcul, qui est composée des trois étapes classiques (pré-traitement, résolution, post-traitement), présentée sur la figure 1.1, une étape supplémentaire se situe donc juste en amont de la dernière étape comme le montre la figure 1.14.

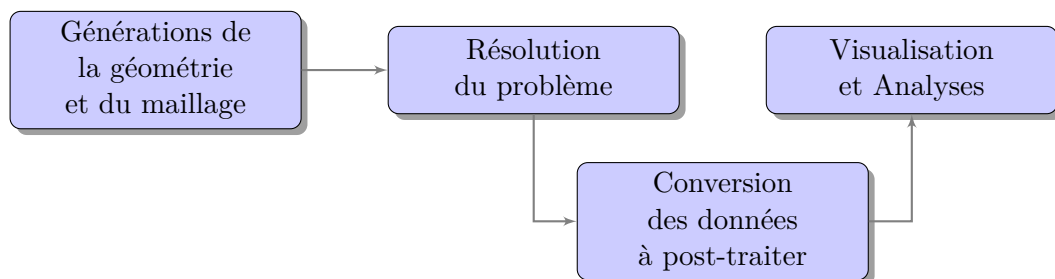


FIGURE 1.14 – Positionnement dans la chaîne de calcul

Les différences entre les travaux sur cette problématique sont de trois types :

- La méthode numérique traitée : méthodes des éléments finis, de Galerkin Discontinue, etc...

- La technique de construction / raffinement du maillage de représentation utilisée.
- La façon de définir et de contrôler l'erreur de visualisation.

Nous explicitons à présent certains de ces travaux.

Leone et al. [Leone et al., 1998] ont proposé une méthodologie pour visualiser des champs discontinus issus de méthodes DG en dimensions deux et trois pour des éléments et fonctions quadratiques. Le raffinement du maillage de représentation est effectué par la méthode dite de *Longest Side Midpoint Insertion*. A chaque étape de la méthode itérative, une cellule à raffiner est découpée suivant son arête la plus longue. L'erreur de visualisation est calculée en ne prenant en compte que les points de contrôle, c'est-à-dire les points où sont définis les degrés de liberté (dans ce cas les sommets et les centres des arêtes). Pour chaque élément du maillage de calcul, elle est définie comme le maximum entre une erreur géométrique et une erreur sur les fonctions. L'erreur géométrique est calculée comme la distance maximale pour chaque point de contrôle entre le point physique de l'élément et celui donné par l'approximation linéaire de l'élément (distance entre les points I et J sur la figure 1.15). De la même manière, l'erreur sur les fonctions est calculée comme le maximum de la valeur absolue de la différence entre la valeur réelle et la valeur interpolée linéairement en chacun des points de contrôle. Cette erreur s'écrit

$$\text{err} = \max \left(\left| F_4 - \frac{F_2 + F_3}{2} \right|, \left| F_5 - \frac{F_1 + F_3}{2} \right|, \left| F_6 - \frac{F_1 + F_2}{2} \right| \right), \quad (1.1)$$

où F_i est la valeur du champ au i -ème point de contrôle P_i .

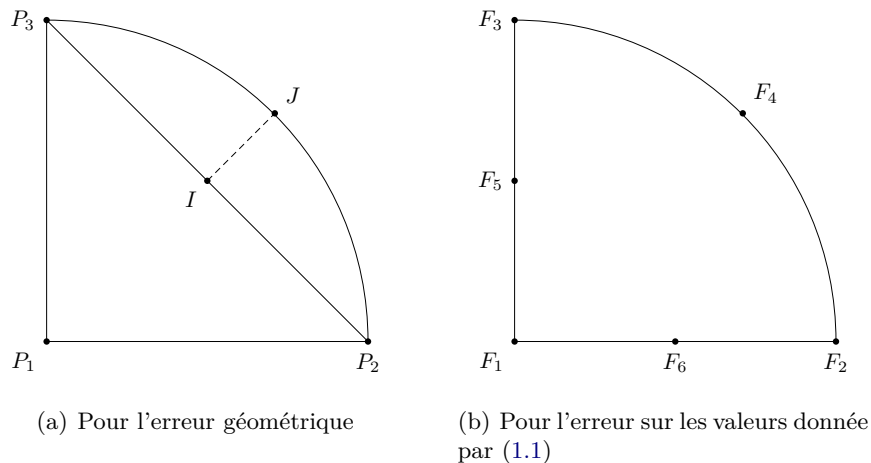


FIGURE 1.15 – Notations pour l'estimation des erreurs dans [Leone et al., 1998]

La simplicité de ces critères fait qu'ils sont très faciles à mettre en place et très rapides à estimer. Néanmoins, plusieurs limitations apparaissent. L'erreur de visualisation n'est définie qu'à partir des centres des arêtes et des sommets, ce qui signifie qu'elle est pertinente pour des fonctions et éléments géométriques quadratiques mais n'assure pas un véritable contrôle pour des fonctions d'ordre plus élevé. En outre, le choix, à savoir l'arête la plus longue, et la technique de découpage des arêtes est également très simple mais n'est pas directement liée à l'erreur de visualisation. On aurait pu décider de raffiner l'arête générant le plus d'erreur. Ce choix n'est donc *a priori* pas optimal pour maîtriser la quantité de données générées pour la

représentation. Néanmoins, le choix de l'arête la plus longue permet de générer des éléments qui ne sont pas trop déformés.

Par ailleurs, Nikishkov [Nikishkov, 2003, Nikishkov, 2011] a proposé une approche adaptative pour avoir une visualisation surfacique de champs issus de méthodes FEM et BEM pour des éléments et fonctions quadratiques. La visualisation étant surfacique, et non volumique, les éléments considérés sont de dimension deux. La méthode est basée sur une subdivision régulière des arêtes des éléments, puis le maillage de représentation des faces en est déduit. Le nombre de subdivisions N de chaque arête est défini localement et dépend de deux facteurs : la courbure de l'arête et l'amplitude des valeurs des fonctions. Des critères sont donnés pour traiter des éléments quadratiques. Le nombre de subdivisions géométriques N_g est proportionnel à un paramètre de courbure qui lui-même dépend du rayon de courbure et des longueurs des arêtes considérées :

$$N_g = k_g \rho, \quad (1.2)$$

où k_g est un coefficient géométrique empirique et ρ un paramètre de courbure donné par

$$\rho = \sin \alpha = \sqrt{1 - \frac{l_2^2}{4l_3^2}}, \quad (1.3)$$

avec α , l_2 et l_3 présentés sur la figure 1.16. Le nombre de subdivisions liées aux valeurs de la

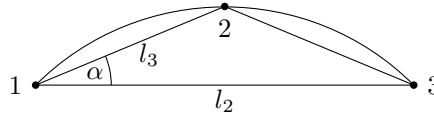


FIGURE 1.16 – Notations pour définir le nombre de subdivisions géométriques.

fonction N_v dépend des extrema de la fonction sur l'arête et est défini par

$$N_v = k_\tau \tau, \quad (1.4)$$

où k_τ est un paramètre empirique et τ est donné par

$$\tau = \frac{\max(|F_1 - F_2|, |F_3 - F_2|)}{F_{\max} - F_{\min}}. \quad (1.5)$$

Dans cette expression, F_i correspond à la valeur au point de contrôle i et F_{\max} (resp. F_{\min}) correspond à la valeur maximale (resp. minimale) de la fonction. Les paramètres empiriques k_g et k_τ permettent d'adimensionnaliser les grandeurs. Le nombre de subdivisions de l'arête est ensuite défini comme le maximum entre les deux quantités N_g et N_v :

$$N = \max(N_g, N_v). \quad (1.6)$$

Une fois que la subdivision régulière des arêtes est générée, des points à l'intérieur de chaque élément sont choisis afin que la distance entre ces points soit égale à la distance entre les points sur le bord de l'arête à proximité. Un algorithme de Delaunay, présenté dans la section A.3, permet d'obtenir le maillage de représentation à partir de ces points. La figure 1.17 illustre un exemple de construction de maillage. Le nombre de points sur les arêtes est fixé par (1.6) qui

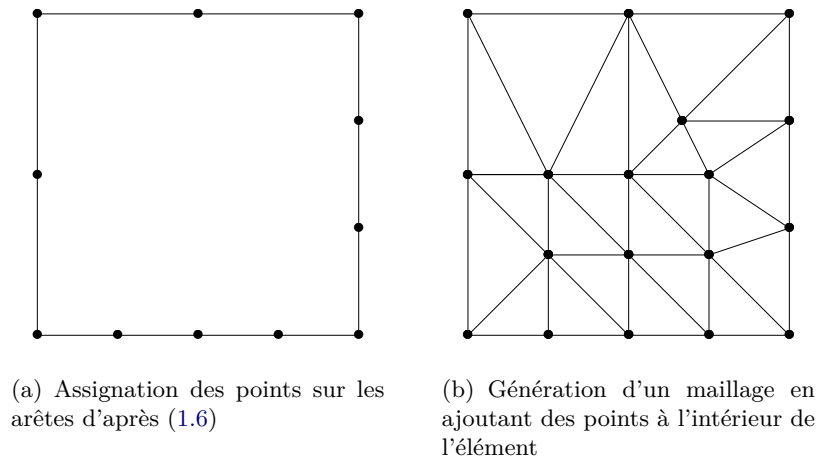


FIGURE 1.17 – Construction des maillages de représentation dans [Nikishkov, 2003]

comprennent, selon l'arête considérée, 5, 4, 3 ou 3 points sur la figure 1.17. Des points sont ensuite choisis à l'intérieur de l'élément et le maillage est obtenu.

Contrairement à l'approche précédente (et aux suivantes), cette méthode n'est pas itérative. Il n'y a pas de tolérance à atteindre. La précision sur la représentation obtenue est donc cachée dans les paramètres empiriques. Leurs choix, qui peuvent paraître anecdotiques, ne sont pas anodins. La construction *a priori* des maillages de représentation est simple et rapide. Cependant, même si le choix de découpe des arêtes est local et dépend de la courbure géométrique et des variations des traces de la solution numérique, cela reste isotrope et donc *a priori* n'est pas le plus adapté. De plus, la construction du maillage de l'intérieur des faces ne prend pas en compte les variations de la solution numérique à l'intérieur. Cette stratégie est adaptée pour des éléments quadratiques où tous les degrés de liberté sont définis sur les arêtes. Ceci est problématique pour des fonctions d'ordre plus élevé car il n'y a pas de contrôle sur l'approximation générée à l'intérieur des faces. Or, dans le cas où la trace de la fonction sur le bord varie peu, contrairement à l'intérieur, une partie de la richesse de la fonction peut être perdue lors de sa représentation.

Schroeder et al. [Schroeder et al., 2005, Schroeder et al., 2006] ont proposé une autre méthode pour visualiser des résultats provenant de méthodes éléments finis d'ordre élevé. Le maillage de représentation est créé à l'aide d'un processus de raffinement basé sur un découpage des arêtes (*edge-based tessellation*) en ajoutant un point au milieu de celles-ci [Chung and Field, 2000, Ruprecht and Müller, 1998]. Ils mettent en place un cadre où plusieurs mesures de l'erreur de visualisation sont définies afin de s'assurer de la convergence de la représentation vers la fonction à représenter. Ces mesures peuvent éventuellement être combinées. Ils présentent quelques critères adaptés pour des fonctions quadratiques :

- Erreur sur la géométrie (*Object-Based Geometric Error Measure*). Cette erreur est égale à la distance entre le centre de l'arête paramétrique donné par la transformation géométrique réelle et le segment passant par les deux extrémités de l'arête. La distance est obtenue à partir du point du segment le plus proche du centre de l'arête paramétrique et est notée d sur la figure 1.18.a qui illustre où C est le centre de l'arête paramétrique.
- Erreur sur l'aplatissement (*Object-Based Flatness Error Measure*). Cette erreur est basée

sur le calcul de l'angle formé à partir des deux extrémités de l'arête et du centre paramétrique de l'arête. Sur la figure 1.18.a, il s'agit de l'angle \widehat{ACB} .

- Erreur sur les valeurs (*Attribute-Based Error Measure*). Cette erreur est égale à la distance entre la valeur interpolée linéairement et la vraie valeur au centre de chaque arête. La figure 1.18.b donne les notations et l'erreur vaut $|F_c - (F_A + F_B)/2|$.
- Erreur liée à la vue (*Image-Based Geometric Error Measure*). Cette erreur est liée aux projections faites lorsque la caméra est changée par des rotations des objets dans un logiciel de visualisation.

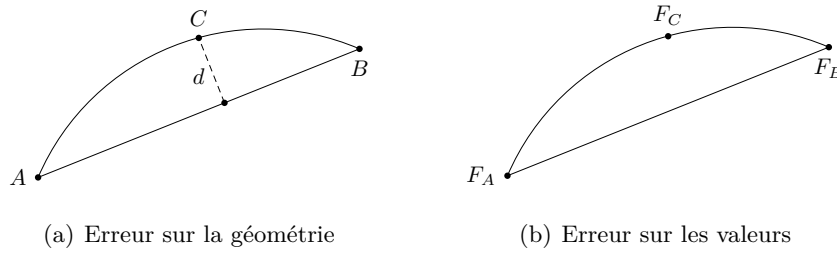


FIGURE 1.18 – Estimation des erreurs dans [Schroeder et al., 2006]

Toutes ces erreurs sont mesurées uniquement à l'aide des centres des arêtes, celles-ci sont donc adaptées à des fonctions et/ou éléments quadratiques mais pas pour des données d'ordres plus élevés. Le traitement de cas où le degré est supérieur à deux est laissé en perspective dans [Schroeder et al., 2005, Schroeder et al., 2006]. La méthode proposée est proche des travaux de Leone et al. [Leone et al., 1998] avec une erreur de visualisation basée sur des points de contrôle quadratiques et un raffinement par découpage d'arêtes. Cependant, contrairement à Leone et al. [Leone et al., 1998], le choix de l'arête est fixé par l'erreur de visualisation, et non par un choix purement géométrique. Ceci est plus pertinent, bien que limité à un contrôle partiel.

Remacle et al. [Remacle et al., 2007, Remacle et al., 2010] ont étendu leurs travaux pour pouvoir traiter des résultats provenant de méthodes FEM et DG d'ordre élevé. Originellement, leur logiciel de post-traitement (et de pré-traitement) Gmsh [Geuzaine and Remacle, 2009] a été développé pour manipuler du bas ordre. Une structure d'arbre permet de construire le maillage de représentation potentiellement jusqu'à un niveau maximal fixé *a priori*. Un algorithme récursif permet de sélectionner, à l'aide d'une estimation d'erreur et d'une tolérance cible, les éléments générant le maillage de représentation qui sont subdivisés isotropiquement. Un nombre minimal de découpe est fixé afin d'éviter de rater des variations importantes de la fonction dans des zones de taille inférieure à la longueur caractéristique des éléments. L'indicateur d'erreur est le même que celui utilisé par Leone et al. [Leone et al., 1998] et est donné par (1.1). Il est important de noter qu'à partir d'un champ élément fini continu, cette procédure de raffinement non-conforme peut produire des données de visualisation discontinues. Les auteurs indiquent néanmoins [Remacle et al., 2007] qu'il est possible d'assurer que la représentation reste continue en introduisant un processus de raffinement plus complexe afin d'avoir un maillage de représentation conforme. Un exemple de représentation obtenue par Gmsh est affiché sur la figure 1.19. La fonction est d'ordre 3, la tolérance de visualisation a été réglée sur 10^{-3} et le nombre maximal de découpages à 7.

L'estimation d'erreur et le h-raffinement mis en place sont rapides et permettent de mieux

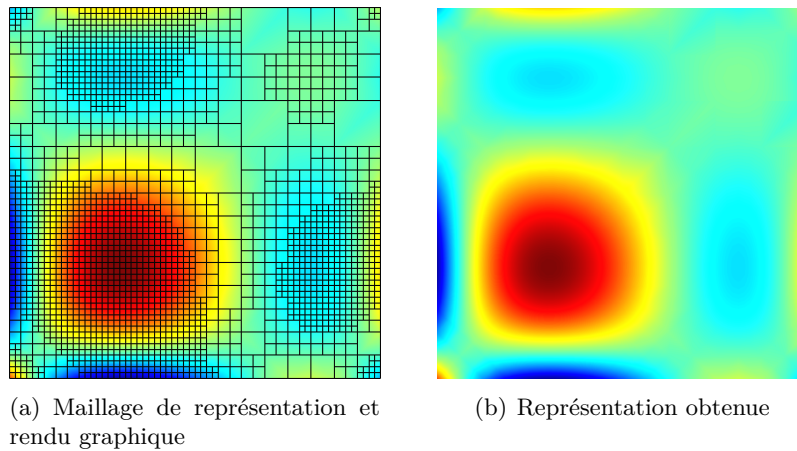


FIGURE 1.19 – Exemple de représentation obtenue dans Gmsh pour une fonction d'ordre 3.

contrôler l'erreur de visualisation que les travaux précédents puisque la construction du maillage est guidée par l'estimation d'erreur et est récursive. Néanmoins, le motif des découpages est fixé *a priori* lorsque le niveau maximal de découpe est choisi par l'utilisateur. La méthode n'est pas totalement adaptative.

La thèse de master de Walfisch [Walfisch, 2007] a porté sur la visualisation de solutions DG d'ordre élevé en mécanique des fluides (CFD). La construction des maillages de représentation est adaptative jusqu'à convergence sous la tolérance et est basée sur l'ajout de points choisis où l'erreur de visualisation est maximale. Après chaque ajout de point, la zone est réarrangée pour minimiser l'erreur de visualisation en utilisant des retournements d'arêtes, également appelé *swaps* en anglais [George, 1991, Dobrzynski, 2005]. L'erreur de visualisation est calculée aux points de quadrature comme la différence en valeur absolue entre la valeur de la solution numérique et son approximation linéaire. L'introduction de points de quadrature vient de la thèse de Quattrochi [Quattrochi, 2006], qui définit l'erreur de visualisation comme l'écart en norme L^1 ou L^∞ entre la solution et sa représentation. Dans les deux cas, des points de quadrature sont définies pour évaluer ces quantités.

Le raffinement des maillages de représentation est guidé par l'erreur de représentation et permet une adaptation anisotrope. Néanmoins, ce contrôle est basé uniquement sur les points de quadrature mais n'est donc, là encore, que partiel. La méthode n'est pas adaptée pour des champs FEM car des discontinuités sur la représentation peuvent être introduites entre les éléments à cause des possibles *hanging nodes* (maillage non-conforme) dans le maillage de représentation [Walfisch, 2007].

Xu et al. [Xu et al., 2017] s'intéressent à la visualisation de solutions provenant de méthode DG d'ordre élevé pour des éléments quadratiques et montrent l'intérêt de les utiliser à la place d'éléments linéaires. Le raffinement des maillages de représentation suit une structure d'arbre où les éléments sont découpés de manière isotrope si l'erreur de visualisation est supérieure à une tolérance. L'erreur de visualisation est définie comme l'écart en norme L^1 entre la solution numérique et sa représentation sur chaque élément. Afin d'estimer l'erreur de visualisation, des formules de quadratures sont utilisées avec un nombre de points de quadrature choisi en fonction de l'ordre des fonctions locales. Afin d'être performant, ils proposent une approche simple qui

consiste à diminuer l'ordre de la quadrature par un après chaque itération dans le processus adaptatif. Dans leur article [Xu et al., 2017], ils expliquent que cela est suffisant pour avoir une estimation précise de l'erreur de visualisation.

L'indicateur d'erreurs utilisé est facile et rapide à estimer mais le contrôle assuré de l'écart entre la solution et sa représentation n'est qu'en norme L^1 et n'est donc que partiel. En effet, cela signifie qu'en « moyenne » seulement, la représentation est correcte. Le choix de la norme L^1 utilisée se justifie car son évaluation est rapide mais il est possible de perdre une partie de la richesse de l'information contenue dans l'ordre élevé. Comme la norme L^1 est utilisée, un raffinement isotrope est mis en place mais n'est donc pas optimal.

Pour résumer, les travaux sur le *low order remeshing* sont basés sur la construction d'un maillage de représentation et une approximation de la solution sur celui-ci. Les principales différences entre les diverses approches viennent de la façon dont est définie l'erreur de visualisation, l'écart entre la solution numérique et son approximation, et comment celle-ci est évaluée. Or, le choix de l'indicateur d'erreurs est primordial puisque la qualité de la représentation ainsi que le coût de génération sont liés à ce choix. En général, ce choix n'est pas optimal.

1.4 Objectifs de cette thèse

L'objectif de cette thèse est de concevoir une méthode de visualisation de résultats provenant de codes de calcul utilisant des schémas d'ordre élevé. Outre la représentation des résultats, la possibilité d'utiliser les outils classiques de post-traitement (isovaleurs, lignes de courant, coupes etc...) est nécessaire.

Une première stratégie consiste à créer un logiciel dédié. Néanmoins, cela nécessite le développement complet d'une plateforme et de tous les outils nécessaires au post-traitement. Dans ce cas, des lancers de rayons peuvent être utilisés afin d'avoir une représentation pixel exacte. Ceci implique des calculs coûteux pour générer la représentation mais également lors de chaque opération, en particulier de post-traitement. L'interactivité peut donc être limitée. A contrario, la simplicité de construction de la représentation dans la stratégie de *low order remeshing* fait que tous les post-traitements sont immédiats dans un logiciel de visualisation standard.

Nous choisissons d'adopter la stratégie de remaillage pour construire une représentation de bas ordre évoquée précédemment mais nous souhaitons limiter les faiblesses de cette approche. Nous avons vu que les trois principales différences des travaux sur cette problématique sont les types d'applications, la technique de construction / raffinement du maillage de représentation ainsi que la façon de définir et de contrôler l'erreur de visualisation. A l'instar de Schroeder et al. [Schroeder et al., 2005, Schroeder et al., 2006], nous proposons un cadre général de conversion de données. Nous voulons que la méthode de visualisation proposée soit suffisamment robuste pour traiter n'importe quel schéma d'ordre élevé comme des méthodes continues (par exemple FEM) ou discontinues (DG). Il est primordial d'assurer le respect de ces discontinuités et de ne pas ajouter de sauts artificiels lorsque la solution est continue. Un effort important de notre méthode porte sur cet aspect. Étant donné que nous désirons pouvoir utiliser n'importe quel logiciel de visualisation, une approximation affine optimisée de la solution est générée et doit être contrôlée. Or, l'analyse des résultats est effectuée directement à partir de la représentation ou à l'aide d'outils utilisant celle-ci. Aussi, nous exigeons une maîtrise ponctuelle sur cet écart car des données ponctuelles peuvent être extraites de la représentation. Naturellement, le rendu graphique dépend du maillage de représentation généré. Il convient donc de lier la construction de ce dernier en effectuant une adaptation de maillage anisotrope basée sur cette erreur de

visualisation. Nous proposons une méthode de remaillage auto-adaptative « optimale » vis à vis de la représentation.

Nous souhaitons pouvoir traiter n'importe quel type d'éléments géométriques. Les éléments d'un maillage peuvent être regroupés en deux types : les éléments droits et les éléments courbes. La définition suivante explicite la différence.

Définition 1.1. *Un élément est dit droit s'il est possible de l'approcher exactement en utilisant uniquement une union de simplexes. Sinon, un élément est dit courbe.*

1.5 Organisation du manuscrit

Ce document synthétise les travaux qui ont été effectués pendant ces 3 années de thèse. Dans un premier temps (chapitres 2, 3 et 4), nous présentons une méthode de visualisation de solutions numériques d'ordre élevé. Ce travail a fait l'objet d'une publication [Maunoury et al., 2018b]. Dans un second temps (chapitre 5), nous nous intéressons à la reconstruction et à la visualisation de champs rayonnés. Ce travail a également été publié [Maunoury et al., 2018a]. Pour cette seconde partie, la méthode de visualisation est utilisée afin de représenter les champs reconstruits.

Chapitre 2. Le deuxième chapitre est dédié à la conception d'une méthode de visualisation sur des éléments droits. Ces derniers sont les éléments géométriques de bas ordre qui peuvent être approchés exactement par une union de simplexes (i.e. triangles ou tétraèdres). Dans ce chapitre, nous donnons un cadre mathématique aux méthodes que l'on souhaite traiter en précisant les hypothèses vérifiées par la solution numérique. Nous formalisons sous forme d'objectifs ce qui, selon nous, est une « bonne » visualisation en garantissant le contrôle ponctuel de l'erreur de visualisation introduite lors de l'approximation de la solution numérique et en assurant que les discontinuités sont bien gérées. La définition de ces objectifs est primordiale car c'est ce qui permettra de vérifier si la représentation générée est correcte ou non. Les principales difficultés de la méthodologie mise en place sont liées aux contraintes fixées par les objectifs. En particulier, un raffinement spécifique est mis en place de sorte à générer un maillage conforme afin de ne pas ajouter de discontinuité artificiel. Nous construisons un indicateur d'erreurs *a posteriori* qui contrôle ponctuellement l'erreur de visualisation et guide la construction d'un maillage de représentation. Un algorithme est développé et est explicité afin de générer une représentation conforme aux objectifs fixés. Finalement, nous discutons, dans le cas d'éléments droits, de la convergence de l'algorithme.

Chapitre 3. Le troisième chapitre est consacré à l'extension de la méthode de visualisation au traitement des éléments courbes. Ces derniers sont les éléments géométriques d'ordre (i.e. le degré de la transformation géométrique) strictement supérieur à un, qui ne peuvent donc pas être approchés exactement par une union de simplexes. Nous commençons par expliquer les problèmes liés à l'utilisation d'éléments courbes et montrons que la méthodologie mise en place dans le chapitre 2 ne peut pas être utilisée telle quelle. En particulier, des zones non contrôlées apparaissent dans ce cas et amènent notamment des recouvrements (dans la représentation, plusieurs valeurs sont associées à certains points du domaine de calcul) qui sont des artefacts à éviter. Pour pallier ces problèmes, un estimateur *a posteriori* spécifique est mis en place et guide la construction de la représentation. L'introduction de cet indicateur d'erreurs spécifique a des répercussions sur l'algorithme qui est modifié en conséquence. Tout ceci permet de généraliser la méthode de visualisation et de pouvoir traiter n'importe quel type d'élément. Finalement, nous

montrons que même avec des éléments courbes, si l'algorithme converge, alors les objectifs sont vérifiés.

Chapitre 4. Le quatrième chapitre concerne les validations numériques de la méthode de visualisation. Dans un premier temps, nous confrontons notre approche qui assure un contrôle L^∞ à un contrôle en norme L^2 uniquement. L'intérêt de demander un contrôle ponctuel est ainsi mis en évidence. Ensuite, nous comparons notre approche adaptative à l'approche naïve qui consiste à redécouper *a priori* chaque élément du maillage de calcul afin de visualiser une approximation de la solution dans un logiciel de visualisation standard et montrons les gains sur les rendus graphiques générés. Puis, la démarche est validée pour des éléments droits ainsi que des éléments courbes. Des exemples simples avec très peu d'éléments dans le maillage de calcul sont d'abord examinés, puis nous nous intéressons à des exemples concrets où les données proviennent de codes de calcul. Le potentiel de la méthode de visualisation est mis en exergue par le traitement de problèmes de type industriel. Ce chapitre permet donc de valider la méthode de visualisation qui est également utilisée dans le chapitre suivant.

Chapitre 5. Le cinquième chapitre traite de la visualisation de champs rayonnés dans des grands domaines. Nous nous intéressons à des problèmes de propagation d'ondes. Plus précisément, le but est, à partir de solutions calculées sur la frontière d'un objet diffractant par la méthode des éléments de frontière par exemple, de reconstruire les champs rayonnés dans un domaine à visualiser. Ces derniers sont ensuite représentés et post-traités par la méthode de visualisation. La solution fondamentale qui intervient dans les problèmes de propagation qui nous intéresse est connue pour être oscillante. Nous choisissons de construire les champs comme une combinaison d'ondes planes afin de bien capter les oscillations et de polynômes d'ordre élevé pour mieux approcher le champ rayonné. Notre but est de compresser l'information, c'est-à-dire limiter le nombre de degrés de liberté tout en assurant une bonne précision dans la reconstruction. Afin d'ajouter des ondes planes dans les fonctions de base des champs reconstruits, des vecteurs appelés directions sont introduits et choisis avec attention : on parle d'interpolation directionnelle. Nous proposons un algorithme générant la reconstruction des champs rayonnés basé sur deux étapes : le domaine est d'abord subdivisé afin d'introduire des directions et des ondes planes, puis les ordres des fonctions de base sont assignés. Afin de subdiviser le domaine, un critère d'admissibilité est défini. Ce dernier quantifie les oscillations pour un élément de la subdivision en comparant les longueurs de l'élément considéré à une longueur d'onde modifiée. L'assignation des ordres est fixée par des règles *a priori* à partir de ratios qui comparent également les longueurs de l'élément à une longueur d'onde modifiée. Enfin, des validations numériques mettent en évidence les avantages de l'interpolation directionnelle et montrent le potentiel de notre algorithme.

Chapitre 2

Conception d'une méthode de visualisation assurant le contrôle ponctuel des erreurs de visualisation sur des éléments droits

Contents

| | | |
|------------|--|-----------|
| 2.1 | Formalisation des objectifs de la méthode de visualisation | 32 |
| 2.2 | Cadre des méthodes d'ordre élevé | 33 |
| 2.3 | Conception d'une méthodologie pour répondre aux objectifs | 36 |
| 2.3.1 | Réalisation de l'objectif 1 | 36 |
| 2.3.2 | Réalisation de l'objectif 2 | 38 |
| 2.3.3 | Réalisation de l'objectif 3 | 40 |
| 2.3.4 | Résumé de la méthodologie | 44 |
| 2.4 | Justifications mathématiques de la méthode proposée | 45 |
| 2.4.1 | Réalisation de l'objectif 1 | 45 |
| 2.4.2 | Réalisation de l'objectif 2 | 47 |
| 2.4.3 | Réalisation de l'objectif 3 | 48 |
| 2.5 | Algorithme pour générer une représentation satisfaisant les objectifs | 51 |
| 2.5.1 | Un algorithme de recherche d'extrema : DIRECT | 51 |
| 2.5.2 | Algorithme complet pour les éléments droits | 54 |
| 2.5.3 | Théorème sur la réponse aux objectifs | 56 |
| 2.6 | Conclusions du chapitre | 57 |

Ce chapitre est consacré à la mise en place d'une méthode de visualisation de résultats de méthode d'ordre élevé sur des éléments droits. Ce chapitre se limite aux traitements des éléments droits qui sont les éléments le plus souvent utilisés comme les éléments triangulaires, quadrangulaires, tétraédriques ou hexaédriques simples et qui sont plus faciles à traiter. La définition des éléments droits est rappelée dans l'introduction en définition 1.1. Le traitement spécifique des éléments courbes est reporté au chapitre 3. La gestion des éléments droits donne néanmoins les principales idées qui sont utilisées pour générer une « bonne » représentation.

Dans un premier temps, nous proposons une définition de ce qui, selon nous, est une « bonne » représentation de simulations d'ordre élevé. L'approche que nous proposons est basée sur la

stratégie de remaillage en bas ordre présentée dans l'introduction de ce manuscrit (section 1.3.3). Nous souhaitons néanmoins surmonter les limites que l'on peut reprocher à cette stratégie. Nous formalisons la définition d'une « bonne » représentation sous forme d'objectifs dans un cadre général, c'est-à-dire pour tout type d'éléments, qu'ils soient droits ou courbes. Nous justifions également pourquoi ceux-ci sont pertinents.

Dans un deuxième temps, nous explicitons le cadre des méthodes d'ordre élevé que nous souhaitons pouvoir traiter. Des hypothèses sur le maillage de calcul et sur les fonctions de base de la solution numérique sont données. L'éventail des méthodes numériques que nous voulons traiter est large FEM/DG/BEM...

Ensuite, nous proposons une méthodologie répondant aux objectifs. Celle-ci est présentée en deux temps : les idées sont d'abord données puis les aspects mathématiques sont justifiés.

Enfin, nous présentons un algorithme générant une « bonne » représentation dans le cas d'éléments droits. Nous montrons que, sous condition de convergence, l'algorithme proposé assure la construction d'une représentation pertinente et fidèle à la solution numérique à visualiser.

2.1 Formalisation des objectifs de la méthode de visualisation

Nous souhaitons proposer une méthode reposant sur la stratégie dite de *low order remeshing* afin de générer une représentation « pertinente » f_{vis} , d'une solution numérique d'ordre élevé f_{num} . Cette représentation sera visualisée dans un logiciel de visualisation standard. Pour cela, nous formalisons notre stratégie par l'intermédiaire de trois objectifs que la représentation doit vérifier pour qu'elle soit considérée correcte.

Tout d'abord, nous voulons que la représentation puisse être post-traitée/exploitée par n'importe quel logiciel de visualisation. Or, ces logiciels savent uniquement représenter *exactement* (i.e. la représentation affichée correspond exactement à la solution numérique) les fonctions affines sur simplexes. Dans les autres cas, une erreur de visualisation plus ou moins maîtrisée est générée. De manière générale, les fonctions que nous souhaitons visualiser ne sont pas affines et donc, afin d'éviter ce type d'erreur, nous choisissons de construire la représentation à l'aide de fonctions affines sur simplexes uniquement. Néanmoins, cette construction produit une erreur de visualisation induite par la projection de f_{num} sur l'espace des fonctions affines sur simplexes. Plus précisément, le support de f_{num} est approché par des simplexes et il y a donc généralement deux contributions à cette erreur. La première est due à l'approximation des fonctions tandis que la seconde est due à l'approximation de la géométrie. Nous cherchons à contrôler ces deux sources d'erreurs lors de la construction de la représentation.

Contrairement à ce qui est fait actuellement, nous désirons assurer un contrôle ponctuel de l'erreur de représentation, c'est à dire en norme L^∞ . Ce dernier est crucial car un utilisateur qui post-traite ses résultats dans un logiciel de visualisation peut vouloir récupérer des informations ponctuelles extraites directement à partir de l'image affichée. Par exemple, un utilisateur va extraire une valeur, faire des coupes, ce qui est intéressant lorsque les simulations sont faites en dimension 3, ou encore sortir des lignes de niveau (isovaleurs), qui peuvent par exemple correspondre à des seuils. Or, les outils de post-traitement des logiciels de visualisation se basent uniquement sur la représentation qui est exploitée telle quelle. Il est donc nécessaire d'avoir un rendu graphique pertinent en maîtrisant correctement l'erreur de visualisation due aux approximations des fonctions et de la géométrie.

Enfin, nous exigeons que la représentation n'introduise pas de saut artificiel lorsque la fonction à représenter est continue. Un saut artificiel est une discontinuité qui apparaît sur la représentation entre deux éléments du maillage de calcul alors que la solution numérique est localement continue. Un tel artefact donnerait une information fautive qui doit donc être évitée. A contrario, si la solution numérique comporte des sauts entre les éléments, leur représentation est primordiale. Ces sauts peuvent venir de la physique, lors d'un changement de matériau par exemple, ou de la méthode numérique, lors de l'utilisation de schémas discontinus de type DG.

Pour résumer, les objectifs que la représentation doit vérifier pour qu'elle soit considérée comme correcte sont les suivants.

- O₁** : La représentation est obtenue à l'aide de fonctions affines sur simplexes.
- O₂** : L'erreur de visualisation entre la solution numérique et sa représentation est majorée par une tolérance $\varepsilon > 0$ en norme L^∞ .
- O₃** : La représentation est localement continue si la solution numérique est localement continue.

Formulé de cette manière, l'objectif **O₃** préconise que de ne pas ajouter de saut artificiel. Or, avec en plus la maîtrise d'un contrôle ponctuel demandé par l'objectif **O₂**, la représentation des discontinuités est également assurée lorsque la solution numérique est discontinue.

2.2 Cadre des méthodes d'ordre élevé

Dans cette section, nous explicitons les propriétés des solutions numériques que nous souhaitons visualiser. Nous donnons des hypothèses que le maillage et les fonctions de base doivent vérifier. Dans notre cas, nous nous intéressons à la visualisation de solutions issues de tout type de méthodes numériques présentant une discrétisation spatiale d'ordre élevé. Nous ne nous limitons pas à la visualisation de fonctions polynomiales. En effet, par exemple des fonctions spéciales comme des ondes planes ou des splines utilisées notamment pour des éléments isogométriques [Hughes et al., 2005] peuvent également être traitées. Nous nous intéressons particulièrement au post-traitement de résultats obtenus par des méthodes FEM et DG d'ordre élevé.

Nous considérons que le problème à l'origine consistait à calculer l'approximation d'une fonction f définie comme l'image d'un sous-espace X d'un espace $\mathbb{X} = \mathbb{R}^d$, où d est la dimension à savoir 1, 2 ou 3, vers un \mathbb{K} -espace vectoriel \mathbb{Y} , où $\mathbb{K} = \mathbb{R}$ ou \mathbb{C} . Ce sont des espaces de Banach équipés de normes notées respectivement $\|\cdot\|_{\mathbb{X}}$ et $\|\cdot\|_{\mathbb{Y}}$. Les champs de la solution numérique peuvent être scalaires ou vectorielles, suivant la dimension de l'espace \mathbb{Y} . La norme $\|\cdot\|_{\mathbb{Y}}$ dépend de l'espace de Banach en question. Par exemple, lorsque l'on résout les équations de Maxwell, plusieurs fonctions numériques peuvent être envisagées. Tout d'abord, si le champ électrique est considéré, $f_{num} = E$, le choix naturel est la norme euclidienne : $\|f_{num}\|_{\mathbb{Y}} = \|E\|_{\mathbb{R}^d}$ où d est la dimension. Si maintenant les champs électrique et magnétique sont considérés en même temps, $f_{num} = (E, H)$, la norme peut être une énergie $\|f_{num}\|_{\mathbb{Y}}^2 = \varepsilon_0 \|E\|_{\mathbb{R}^d}^2 + \mu_0 \|H\|_{\mathbb{R}^d}^2$ où ε_0 et μ_0 sont la permittivité électrique du vide et la perméabilité magnétique du vide.

La solution numérique, calculée par un code de calcul, est notée f_{num} . La résolution du problème est effectuée à partir d'un maillage, noté \mathcal{T}_h , du domaine X . L'indice h fait référence à la taille caractéristique de chaque élément qui en général varie d'un élément à un autre. Le

maillage est raffiné à certains endroits du domaine en fonction des contraintes géométriques et des phénomènes physiques à observer. Ce maillage est supposé vérifier les propriétés suivantes :

- H₁** : Le maillage est constitué d'éléments K fermés et bornés tels que $\bigcup_{K \in \mathcal{T}_h} K = X$.
- H₂** : Il n'y a pas de recouvrements entre éléments. L'intersection de l'intérieur de deux éléments est vide.
- H₃** : Chaque élément du maillage $K \in \mathcal{T}_h$ est l'image d'une cellule de référence \widehat{K} , exprimée dans un espace de référence $\widehat{\mathbb{X}}_K$, par une transformation bijective notée g_K . La dimension de $\widehat{\mathbb{X}}_K$, qui est la dimension de l'élément K dans \mathbb{X} , est notée d_K .

Ces hypothèses sont tout à fait classiques et sont motivées par les observations suivantes.

La première hypothèse exprime simplement le fait que l'ensemble des éléments recouvre entièrement le domaine de calcul. Dans le cas où cette hypothèse n'est pas vérifiée ($\bigcup_{K \in \mathcal{T}_h} K \neq X$), il convient de remplacer dans la suite X par $\bigcup_{K \in \mathcal{T}_h} K$. L'hypothèse **H₁** n'est donc pas cruciale mais permet de simplifier la suite sans perdre de généralité.

La seconde hypothèse assure que le maillage de calcul est bien défini. Les maillages peuvent être conformes (la définition est rappelée en annexe A.2) ou non. Un maillage conforme est en général nécessaire lorsque la méthode FEM est utilisée. Dans ce cas, il est interdit d'avoir des *hanging nodes*. Ceci signifie qu'un nœud appartenant à plusieurs éléments doit être un sommet de chacun de ces éléments. Lorsque la méthode DG est utilisée, il est possible d'avoir un maillage non-conforme. Cela autorise plus de liberté lors de la génération du maillage et, en particulier, facilite la construction d'un maillage localement raffiné.

La troisième hypothèse est liée à la description des fonctions de base qui sont, en général, définies sur un élément de référence. En effet, afin de simplifier les calculs, un élément de référence et une transformation géométrique g_K associée à chaque élément K sont souvent introduits. Cette transformation est nécessairement bijective entre les éléments de référence et réel. Ce que l'on appelle élément réel est l'élément K du maillage à qui un élément (ou cellule) de référence \widehat{K} est associé. Il est également possible de travailler directement sur l'élément réel, sans passer par l'élément de référence, et ainsi la transformation géométrique est l'identité et les éléments de référence et réel sont identiques. Sinon, l'introduction de la transformation géométrique devient particulièrement intéressante lorsque la dimension des éléments est plus petite que la dimension de l'espace : lorsque le domaine est une surface courbe maillée par des triangles ou quadrangles dans l'espace par exemple (la dimension des éléments est 2 tandis que la dimension de l'espace est 3). Des exemples de transformations géométriques se trouvent en annexe B. Dans toute la suite, lorsque l'on parle d'un élément de référence, celui-ci est donc rattaché à un élément réel par une transformation géométrique.

Les fonctions de base sont la clef de voûte des méthodes numériques puisqu'elles sont le socle de la construction des champs de la solution numérique. Leur choix est motivé par les propriétés de celles-ci et des problèmes traités. Elles peuvent être scalaires ou vectorielles voire même tensorielles et être associées à des points (polynômes de Lagrange par exemple) ou non (onde plane par exemple). De plus, certaines fonctions de base comme les éléments de Raviart-Thomas [Raviart and Thomas, 1977] ou de Nédélec [Nédélec, 1980] peuvent être intéressantes pour leurs propriétés sur les continuités des composantes normales ou tangentielles. En effet, les éléments de Raviart-Thomas préservent la continuité des composantes normales, ce qui peut être utile pour des problèmes de propagation d'ondes acoustiques. Les éléments de Nédélec conservent la continuité des composantes tangentielles, ce qui peut être utile pour les équations de Maxwell

[Nédélec, 1980]. Des exemples de fonctions de base sont donnés en annexe B. Les fonctions de base sont donc définies localement sur chaque élément K de \mathcal{T}_h et sont notées $\{\varphi_i^K\}_{i=1,\dots,N_K}$, où N_K est le nombre local de degrés de liberté, et vérifient les hypothèses suivantes :

H₄ : Sur chaque élément $K \in \mathcal{T}_h$, $\{\varphi_i^K\}_{i=1,\dots,N_K} \in C^0(\widehat{K}, \mathbb{Y})$, où $C^0(\widehat{K}, \mathbb{Y})$ est l'ensemble des fonctions continues sur \widehat{K} ,

H₅ : Sur chaque élément $K \in \mathcal{T}_h$, la définition locale de f_{num} sur K , notée f_{num}^K , est construite à l'aide de degrés de liberté $f_i^K \in \mathbb{K}$ telle que :

$$\forall x \in K, \quad f_{num}^K(x) = \sum_{i=1}^{N_K} f_i^K \varphi_i^K \left(g_K^{-1}(x) \right). \quad (2.1)$$

La régularité demandée par l'hypothèse **H₄** est naturelle afin de construire la solution numérique et n'est pas restrictive car la plupart des méthodes sont basées sur des fonctions de base polynomiales. Dans tous les cas, les fonctions de base sont continues sur chaque cellule de référence.

L'hypothèse **H₅** décrit la manière usuelle de construire la solution numérique localement à partir de fonctions de bases et de degrés de liberté.

Remarque 2.1. *Sur chaque élément du domaine de calcul, plusieurs types de fonctions de base peuvent cohabiter, chacune associée à un champ. Sur un élément, pour chaque type de fonction de base, une cellule de référence différente peut donc lui être associée.*

La figure 2.1 reprend les notations permettant de définir la solution numérique et la cellule de référence. Sur le domaine de calcul X , la solution numérique f_{num} s'écrit donc

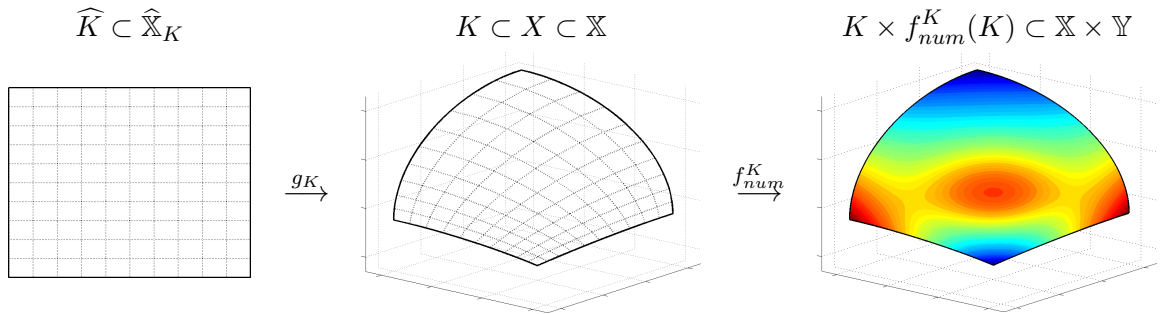


FIGURE 2.1 – Description des notations utilisées.

$$\forall x \in X, \quad f_{num}(\{x\}) = \left\{ f_{num}^K(x) / \forall K \in \mathcal{T}_h, x \in K \right\}. \quad (2.2)$$

Il est important de noter que f_{num} n'est pas une fonction sur X mais est une application. En effet, d'après la définition de f_{num} donnée par (2.2), à l'interface entre les éléments du maillage, en un point donné, plusieurs valeurs peuvent lui être associées. Cette définition permet de représenter des champs discontinus comme c'est le cas avec des schémas DG. Cependant, par les hypothèses **H₂** à **H₅**, pour chaque élément du maillage, f_{num}^K est bien définie et continue lorsque la restriction à l'intérieur de chaque élément est considérée.

2.3 Conception d'une méthodologie pour répondre aux objectifs

La solution numérique étant définie localement sur chaque élément du maillage de calcul, une approximation locale est construite. Cette approche est particulièrement intéressante car elle permet de traiter les éléments indépendamment les uns des autres. Sur chaque élément $K \in \mathcal{T}_h$, la solution numérique locale f_{num}^K est définie par l'équation (2.1). Une représentation, notée $f_{vis}^{\tilde{K}}$, est construite de manière locale. L'exposant \tilde{K} vient du fait que le support de cette fonction peut être différent du support K de f_{num}^K (voir (2.4) ci-dessous). Un maillage de représentation en simplexes, noté $\mathcal{T}(K)$, est construit et la représentation locale $f_{vis}^{\tilde{K}}$ va vérifier :

$$\forall S \in \mathcal{T}(K), f_{vis}^{\tilde{K}}|_S \in \mathbb{P}^1(S), \quad (2.3)$$

où $\mathbb{P}^1(S)$ correspond à l'espace des polynômes de degré au plus égal à 1. Une approximation de l'élément K est générée. En effet, K n'est *a priori* pas subdivisible en simplexes et donc il peut arriver que

$$\tilde{K} := \bigcup_{S \in \mathcal{T}(K)} S \neq K. \quad (2.4)$$

Cette différence de supports peut apparaître lorsque la transformation géométrique g_K n'est pas affine. C'est le cas par exemple lorsque des éléments courbes sont utilisés. Ce type d'éléments nécessite des traitements spécifiques qui sont développés dans le chapitre 3 de ce manuscrit. Nous notons $\tilde{X} := \bigcup_{K \in \mathcal{T}_h} \tilde{K}$ l'union de ces simplexes qui, *a priori*, n'est ni incluse ni contenue dans X où \tilde{K} est défini par (2.4). De la même manière, il peut arriver que

$$f_{vis}^{\tilde{K}}(\tilde{K}) \neq f_{num}^K(K). \quad (2.5)$$

La fonction numérique locale étant définie à partir de l'élément de référence, nous choisissons de créer un maillage de représentation, qui sera le support de la représentation $f_{vis}^{\tilde{K}}$, à partir d'un maillage de l'élément de référence noté $\mathcal{T}(\tilde{K})$. En effet, les fonctions de base sont définies sur l'élément de référence en suivant les hypothèses **H₄** et **H₅** données dans la description des méthodes d'ordre élevé (section 2.2). Le maillage de représentation de l'élément de référence $\mathcal{T}(\tilde{K})$ est composé de simplexes, qui sont construits comme l'enveloppe convexe de $d_K + 1$ points. Les nœuds du maillage de représentation $\mathcal{T}(K)$ sont définis comme image par la transformation géométrique g_K des nœuds de $\mathcal{T}(\tilde{K})$. Le maillage de représentation est ainsi défini à partir de ce maillage de l'élément de référence, en suivant la même connectivité. Par construction, $\mathcal{T}(K)$ est donc également composé de simplexes. Cette façon de faire induit l'interpolation \mathbb{P}^1 de Lagrange de type C^0 de g_K construit à partir de $\mathcal{T}(\tilde{K})$ notée $P^1 g_{\mathcal{T}(K)}$. La figure 2.2 illustre l'ensemble des notations et image le processus décrit dans cette section.

2.3.1 Réalisation de l'objectif 1

La méthode est basée sur la construction d'un maillage de représentation à partir de la cellule de référence, et cela pour chaque élément du domaine de calcul. La création de ces maillages utilise un algorithme de Bowyer-Watson [Bowyer, 1981, Watson, 1981, Rebay, 1993]. C'est une méthode adaptative de génération de maillage qui fonctionne de la façon suivante [George, 1992]. Pour chaque élément $K \in \mathcal{T}_h$, un maillage de sa cellule de référence est construit (étapes 1, 2 et 3 de l'algorithme 1). L'algorithme commence par la construction d'un maillage du bord de l'élément. Plus précisément, un ensemble de points se trouvant sur le bord de la cellule de référence est créé en suivant un critère spécifique (donné dans l'algorithme 3). A

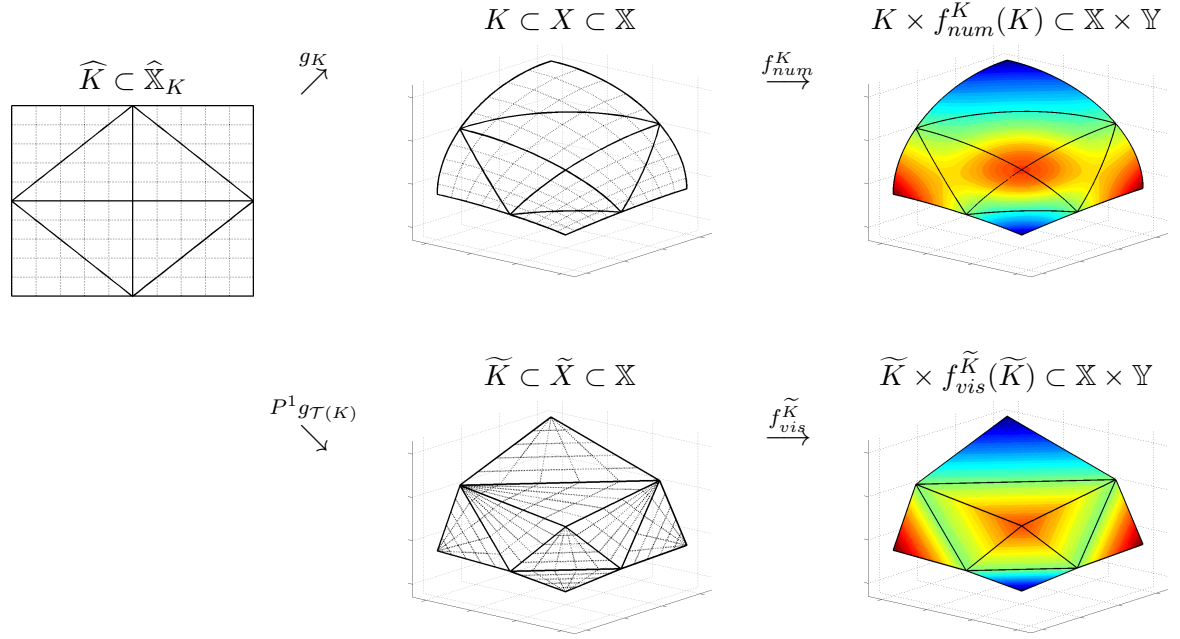


FIGURE 2.2 – Description des notations utilisées.

partir de ces points du bord, une première tessellation de la cellule de référence est générée (deuxième étape de l'algorithme 1). Cela permet de construire une première fonction $f_{vis}^{\widehat{K}}$ qui peut néanmoins être une mauvaise reproduction de f_{num}^K . Alors, afin d'avoir une représentation précise, suivant un critère donné dans la section suivante (algorithme 2), des points sont choisis et ajoutés à la tessellation (troisième étape de l'algorithme 1) si celle-ci a besoin d'être raffinée. Finalement, à partir de ce maillage de la cellule de référence et en suivant la même connectivité, le maillage de représentation de l'élément est déduit. La fonction $f_{vis}^{\widetilde{K}}$ est construite à l'aide de ces simplexes et via l'introduction de fonctions affines définies telles qu'en chaque nœud N du maillage de représentation $\mathcal{T}(K)$, la valeur de la représentation coïncide avec la valeur de la solution numérique : $f_{vis}^{\widetilde{K}}(N) = f_{num}^K(N)$. Cette construction permet de vérifier le premier objectif.

Algorithme 1 : Construction d'une représentation locale $f_{vis}^{\widetilde{K}}$ satisfaisant \mathbf{O}_1 .

1. Construction d'un maillage de la frontière notée $\partial\widehat{K}$ de la cellule de référence \widehat{K} (détaillé plus tard dans l'algorithme 3).
 2. Construction d'un premier maillage $\mathcal{T}(\widehat{K})$ de \widehat{K} .
 3. Mise à jour de $\mathcal{T}(\widehat{K})$: Ajout de nouveaux points pour obtenir une représentation précise (détaillé dans l'algorithme 2).
 4. Construction de $f_{vis}^{\widetilde{K}}$:
 - 4.1. A partir de $\mathcal{T}(\widehat{K})$, construction du maillage de représentation $\mathcal{T}(K)$.
 - 4.2. Interpolation affine de f_{num}^K sur chaque simplexe de $\mathcal{T}(K)$.
-

La justification de la réalisation de l'objectif \mathbf{O}_1 est donnée en section 2.4.1.

2.3.2 Réalisation de l'objectif 2

La construction du maillage de représentation est guidée par le contrôle de l'erreur de visualisation. Deux sources contribuent à cette erreur : la géométrie et les fonctions qui sont respectivement approchées par des simplexes et des fonctions affines. Nous introduisons à présent une quantification de l'erreur entre la solution numérique et sa représentation. Dans la stratégie de remaillage en bas ordre, cette erreur est, en général, contrôlée avec différents critères [Leone et al., 1998, Remacle et al., 2007, Schroeder et al., 2005] adaptés seulement pour des fonctions quadratiques, car basés uniquement sur le centre des arêtes ou alors en norme L^1 [Xu et al., 2017]. Nous demandons un contrôle L^∞ de l'erreur de visualisation. Les définitions de f_{num}^K et $f_{vis}^{\tilde{K}}$ étant locales à chaque élément par les hypothèses \mathbf{H}_4 et \mathbf{H}_5 , cette erreur peut également être estimée localement. La principale difficulté vient des potentielles non-correspondances entre les supports des fonctions, comme le montre l'équation (2.4), et entre les valeurs des fonctions, comme le montre l'équation (2.5). Pour cette raison, nous utilisons la distance de Hausdorff sur $\mathbb{X} \times \mathbb{Y}$ qui permet de mesurer l'erreur entre le graphe de deux fonctions

$$d_H^{\mathbb{X} \times \mathbb{Y}} : (f, \tilde{f}) \in (C^0(\mathbb{X}, \mathbb{Y}))^2 \mapsto \max \left(\sup_{x \in \text{Supp} f} \inf_{\tilde{x} \in \text{Supp} \tilde{f}} d \left((x, f(x)), (\tilde{x}, \tilde{f}(\tilde{x})) \right), \sup_{\tilde{x} \in \text{Supp} \tilde{f}} \inf_{x \in \text{Supp} f} d \left((x, f(x)), (\tilde{x}, \tilde{f}(\tilde{x})) \right) \right), \quad (2.6)$$

où d est une distance sur $\mathbb{X} \times \mathbb{Y}$, faisant intervenir deux coefficients positifs α et β

$$d : ((x, y), (\tilde{x}, \tilde{y})) \in (\mathbb{X} \times \mathbb{Y})^2 \mapsto \max(\alpha \|x - \tilde{x}\|_{\mathbb{X}}, \beta \|y - \tilde{y}\|_{\mathbb{Y}}). \quad (2.7)$$

Les deux constantes α et β sont des paramètres d'adimensionnement qui permettent une comparaison pertinente entre les valeurs de \mathbb{X} et celles de \mathbb{Y} , nous décidons de les fixer comme

$$\alpha = \frac{1}{\|\max_{x \in \mathbb{X}} x - \min_{x \in \mathbb{X}} x\|_{\mathbb{X}}}, \quad \beta = \begin{cases} \frac{1}{\|\max_{y \in \mathbb{Y}} y - \min_{y \in \mathbb{Y}} y\|_{\mathbb{Y}}}, & \text{si } \left\| \max_{y \in \mathbb{Y}} y - \min_{y \in \mathbb{Y}} y \right\|_{\mathbb{Y}} \neq 0, \\ 1, & \text{sinon.} \end{cases} \quad (2.8)$$

Nous choisissons de définir ces poids de pondérations α et β en introduisant les différences d'amplitudes maximales pour comparer des quantités du même ordre de grandeur. Nous supposons, dans la définition de α que le dénominateur ne s'annule pas. Ce dénominateur s'annule lorsque l'élément est restreint à un point. Nous ne considérons pas ce type d'élément dégénéré.

La distance de Hausdorff étant définie par deux optimisations imbriquées, son utilisation en pratique s'avère donc très coûteuse. Pour pallier cette difficulté, un indicateur d'erreurs *a posteriori* (nous montrons qu'il s'agit bien d'un indicateur *a posteriori* dans la section 2.4 justifications mathématiques) est introduit pour estimer cette distance : pour chaque élément $K \in \mathcal{T}_h$ et son maillage de représentation $\mathcal{T}(K)$ associé, nous définissons l'application suivante

$$\Delta_{\mathcal{T}(K)} : (f, \tilde{f}) \in C^0(K, \mathbb{Y}) \times C^0(\tilde{K}, \mathbb{Y}) \mapsto \sup_{\hat{x} \in \hat{K}} \delta_{\mathcal{T}(K)}(f, \tilde{f}; \hat{x}), \quad (2.9)$$

où

$$\delta_{\mathcal{T}(K)}(f, \tilde{f}; \hat{x}) = \max \left(\alpha \left\| g_K(\hat{x}) - P^1 g_{\mathcal{T}(K)}(\hat{x}) \right\|_{\mathbb{X}}, \right. \\ \left. \beta \left\| (f \circ g_K)(\hat{x}) - (\tilde{f} \circ P^1 g_{\mathcal{T}(K)})(\hat{x}) \right\|_{\mathbb{Y}} \right). \quad (2.10)$$

Comme l'indicateur (2.9) est défini sur toute la cellule de référence, un contrôle ponctuel est assuré : en tout point $\hat{x} \in \widehat{K}$. De plus, les deux sources d'erreurs de visualisation, sur la géométrie et sur les valeurs, sont prises en compte dans $\delta_{\mathcal{T}(K)}$. Dans l'équation (2.10), la partie $\alpha \left\| g_K(\hat{x}) - P^1 g_{\mathcal{T}(K)}(\hat{x}) \right\|_{\mathbb{X}}$ mesure l'erreur sur la géométrie tandis que $\beta \left\| (f \circ g_K)(\hat{x}) - (\tilde{f} \circ P^1 g_{\mathcal{T}(K)})(\hat{x}) \right\|_{\mathbb{Y}}$ estime l'erreur sur les valeurs des fonctions.

L'algorithme 2 montre comment la mise à jour du maillage de représentation est effectuée (étape 3 de l'algorithme 1). Les poids α et β étant globaux, ils sont calculés une fois pour toute au lancement du code. Pour diminuer globalement la distance de Hausdorff, nous choisissons de raffiner le maillage de représentation en ajoutant un point réalisant l'erreur maximale. Ce choix va permettre d'obtenir la convergence, y compris dans le cas d'éléments courbes. Comme nous allons le voir dans la suite, un point épineux de la méthode est que, pour un élément $K \in \mathcal{T}_h$, $P^1 g_{\mathcal{T}(K)}$ peut ne pas être injective. Cependant, si pour chaque élément $K \in \mathcal{T}_h$, la fonction $P^1 g_{\mathcal{T}(K)}$ est injective, l'indicateur d'erreurs $\Delta_{\mathcal{T}(K)}$ contrôle la distance de Hausdorff (ceci est démontré avec la proposition 2.6 dans la section 2.4 consacrée aux justifications mathématiques) qui est le critère pour évaluer l'erreur de visualisation. Comme $\Delta_{\mathcal{T}(K)}$ est localisé sur l'élément, cela fournit un outil pour mesurer la qualité de la représentation $f_{vis}^{\widehat{K}}$ à chaque étape de la construction. La convergence de cet indicateur (sous la tolérance ε) assure alors que l'objectif **O₂** est satisfait. Il est important de noter que cette convergence prend en compte à la fois la différence des valeurs de ces fonctions mais également de leurs supports. L'indice $\mathcal{T}(K)$ introduit dans la notation de $\Delta_{\mathcal{T}(K)}$ montre la dépendance de ce dernier au maillage de représentation de K .

Algorithme 2 : Raffinement du maillage de représentation (étape 3 de l'algorithme 1)
pour vérifier **O₂**.

Pré-requis : Calcul des coefficients de pondération α et β (global et unique pour tout le domaine).

tant que l'indicateur $\Delta_{\mathcal{T}(K)}$ est plus grand que la tolérance ε fixée **faire**

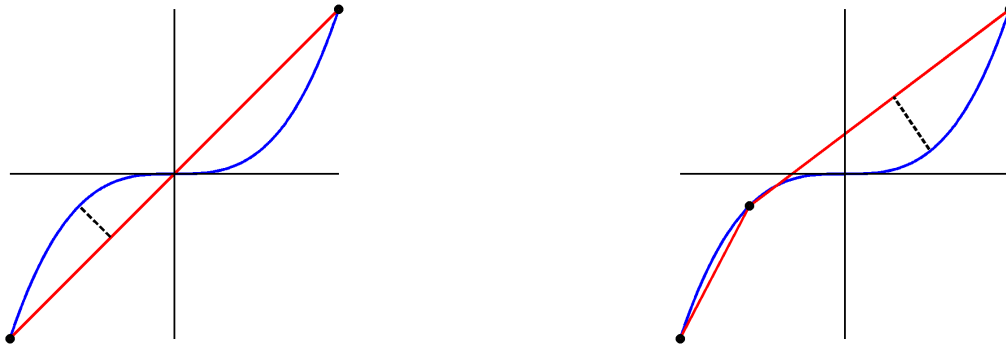
Calcul de l'indicateur *a posteriori* $\Delta_{\mathcal{T}(K)}$ associé au maillage de représentation $\mathcal{T}(\widehat{K})$.

Choisir $\hat{x} = \arg \max \Delta_{\mathcal{T}(K)}$ comme un nouveau point pour raffiner $\mathcal{T}(\widehat{K})$ en suivant l'algorithme de Bowyer-Watson.

fin

Remarque 2.2. *Le choix du point ajouté, à savoir un point réalisant l'erreur maximale, bien que naturel, n'induit pas nécessairement une décroissance de l'erreur de visualisation en norme L^∞ . Par exemple, sur la figure 2.3, la fonction cube est approchée par deux représentations affines par morceaux. La première (figure 2.3.a) est construite à partir de deux points d'interpolation tandis que la seconde est obtenue en ajoutant le point réalisant l'erreur maximale dans le premier cas. Nous constatons que l'erreur en distance de Hausdorff a augmenté malgré le raffinement. Dans ce cas, le seul point qui ne fait pas augmenter la distance de Hausdorff est l'origine. Ce point n'est pas du tout intuitif car l'erreur de visualisation est nulle en ce point. Néanmoins, même*

si la distance de Hausdorff a augmentée après cette première itération, celle-ci va naturellement diminuer quand d'autres points seront ajoutés. Une diminution à chaque itération n'est en effet pas assurée mais il y a une diminution globale si plusieurs itérations sont considérées.



(a) Représentation affine (en rouge) à partir de deux points

(b) Représentation affine (en rouge) à partir de trois points

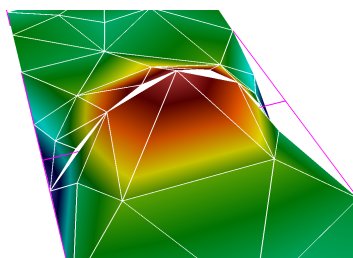
FIGURE 2.3 – Approximation de la fonction cube (en bleu) sur le segment $[-1,1]$ par des représentations affines par morceaux (en rouge). La distance de Hausdorff (en pointillé) augmente dans le cas où le maillage de représentation est raffiné (droite).

La justification de la réalisation de l'objectif \mathbf{O}_2 est donnée dans la section 2.4.2.

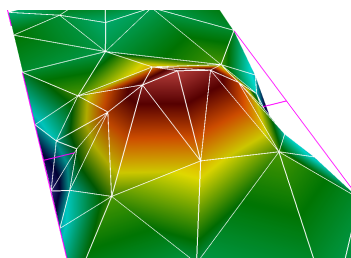
2.3.3 Réalisation de l'objectif 3

En utilisant l'indicateur d'erreurs *a posteriori* donné par (2.9), il est possible de construire un maillage de représentation et donc un rendu graphique. La figure 2.4 montre trois représentations de fonctions définies sur deux éléments qui sont deux carrés. Les figures 2.4.d, 2.4.e et 2.4.f sont les vues de dessus des fonctions qui se situent respectivement au-dessus de celles-ci à savoir les figures 2.4.a, 2.4.b et 2.4.c. Les figures 2.4.a et 2.4.b sont deux représentations de la même fonction qui est continue tandis que la dernière est discontinue. Le premier maillage de représentation est non-conforme (voir figure 2.4.d). La conséquence est la création d'un saut artificiel. En effet, sur la figure 2.4.a, alors que la fonction à visualiser est continue, sa représentation est localement discontinue à l'interface entre les deux éléments. Dans ce cas, l'objectif \mathbf{O}_3 n'est pas vérifié. Ceci est à éviter car un tel saut peut avoir des conséquences néfastes sur l'analyse que l'on pourrait faire en observant une telle discontinuité. En effet, on pourrait par exemple interpréter cette situation comme un changement de milieu/matériau alors qu'il n'en est rien. Le second maillage de représentation est coïncident (figure 2.4.e) et dans ce cas, pour la même fonction à visualiser, la représentation est continue (figure 2.4.b). Une solution simple pour éviter l'ajout de saut artificiel est donc de fixer un maillage unique de l'arête entre les éléments du maillage de calcul. La troisième fonction est discontinue et le saut est bien rendu dans ce cas (figure 2.4.c). Pour cette dernière configuration, le même maillage de représentation (figure 2.4.f) que la deuxième figure a été utilisé.

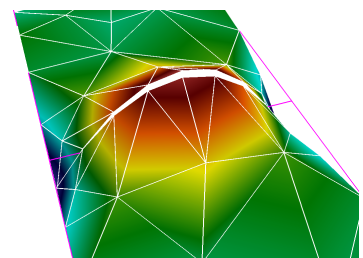
Pour fixer un maillage de représentation unique aux interfaces entre les éléments, un algorithme « simple » est le suivant :



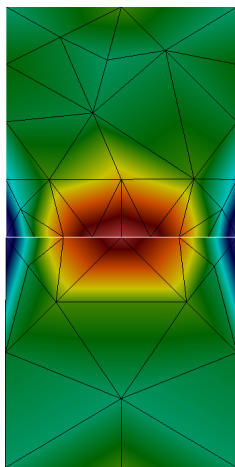
(a) Une fonction continue sur un maillage de représentation non-coïncident (blanc)



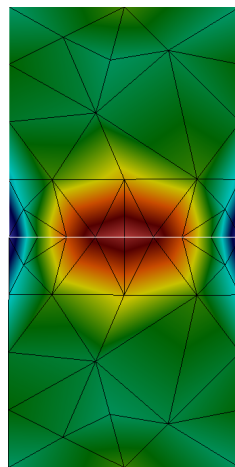
(b) Une fonction continue sur un maillage de représentation coïncident (blanc)



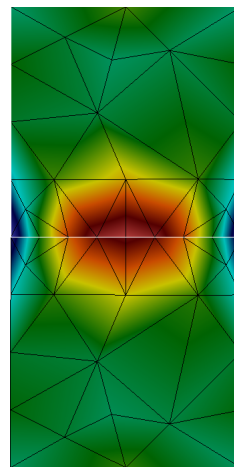
(c) Une fonction discontinue sur un maillage de représentation coïncident (blanc)



(d) Vue de dessus de (a)



(e) Vue de dessus de (b)



(f) Vue de dessus de (c)

FIGURE 2.4 – Représentation sur deux éléments de fonctions continues (a et b) et d'une fonction discontinue (c) sur un maillage de représentation non-coïncident (a) et des maillages coïncidents (b et c). Les vues de dessus sont également affichées (d, e et f).

- Construire un maillage unique de chacune des arêtes, notées $\tilde{\mathcal{T}}_{1D}$, des éléments.
- Construire un maillage unique de chacune des faces, notées $\tilde{\mathcal{T}}_{2D}$, des éléments, à partir des maillages des arêtes $\tilde{\mathcal{T}}_{1D}$.
- Construire les maillages des éléments volumiques, notées $\tilde{\mathcal{T}}_{3D}$, à partir des maillages des faces $\tilde{\mathcal{T}}_{2D}$.

Afin de suivre ce pseudo-algorithme, il est nécessaire d'extraire les faces et les arêtes de tous les éléments. Pour cela, les éléments sont décomposés en dimensions inférieures comme l'illustre la figure 2.5.

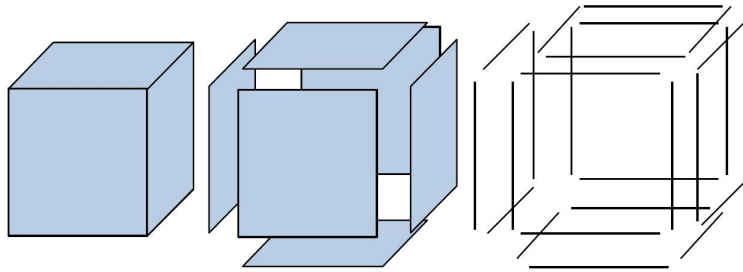


FIGURE 2.5 – Illustration de la décomposition des éléments en dimensions inférieures

Pour avoir un maillage unique aux interfaces entre les éléments du domaine de calcul, la construction des maillages du bord suit l'algorithme 3. Nous regroupons tous les éléments dans les ensembles \mathcal{T}_{iD} suivants leur dimension respective à savoir 1, 2 ou 3. De plus, nous définissons des ensembles plus larges $\tilde{\mathcal{T}}_{2D}$ et $\tilde{\mathcal{T}}_{1D}$. L'ensemble $\tilde{\mathcal{T}}_{2D}$ est constitué d'éléments 2D purs et de faces d'éléments 3D. L'ensemble $\tilde{\mathcal{T}}_{1D}$ est quant à lui composé d'éléments 1D purs et d'arêtes d'éléments de dimensions 2 et 3. Cette décomposition du bord de chaque élément en dimensions inférieures constitue la première étape de l'algorithme 3. La deuxième étape de l'algorithme 3 est d'associer à chaque arête les possibles traces de fonctions 2D et 3D ainsi que les fonctions 1D supportées par celle-ci. De la même manière, nous associons à chaque face, l'ensemble des traces de fonctions 3D et les fonctions 2D qui sont supportées par celle-ci. Cela nous permet de proposer des indicateurs (définis ci-après) qui prennent en compte toutes les contributions dans les étapes 3 et 4 de l'algorithme 3 afin de pouvoir générer les maillages de représentation.

La figure 2.6 montre le processus décrit précédemment dans le cas d'éléments 2D. Dans ce cas, l'élément 1 étant un segment, il ne peut pas être décomposé en dimension inférieure. Par contre, les éléments 2 et 3 qui sont des quadrangles sont décomposés (figure 2.6.b). Un maillage de chaque arête de $\tilde{\mathcal{T}}_{1D}$ est ensuite construit (figure 2.6.c). Il est important de noter qu'un seul maillage est créé pour chaque arête. Ainsi, l'arête entre les quadrangles 2 et 3 n'est traitée qu'une seule fois. A partir de ces maillages, les maillages des éléments de $\tilde{\mathcal{T}}_{2D}$ sont générés (figure 2.6.e). S'il y avait eu des éléments de dimension 3, ceux-ci auraient été construits à partir des maillages des faces. Finalement, des fonctions affines sont construites sur chacun des simplexes pour obtenir la représentation (figure 2.6.f).

Afin d'assurer la convergence globale, il est nécessaire de définir des indicateurs d'erreurs pour chaque dimension permettant de contrôler l'erreur de visualisation. Ainsi, pour une face $F \in \tilde{\mathcal{T}}_{2D}$,

Algorithme 3 : Construction des maillages du bord (étape 1 de l'algorithme 1) pour vérifier \mathbf{O}_3

1. Décomposition des arêtes et des faces des éléments.
2. Création des traces des fonctions associées aux arêtes et aux faces créées en 1.
3. Construction d'un maillage $\mathcal{T}(\hat{E})$ pour chaque arête E comme la somme de toutes les contraintes des traces sous-jacentes.

tant que $(\Delta_{\mathcal{T}(E)} > \varepsilon)$ **faire**

 | Choisir $\hat{x} = \arg \max \Delta_{\mathcal{T}(E)}$ comme un nouveau point pour raffiner $\mathcal{T}(\hat{E})$.

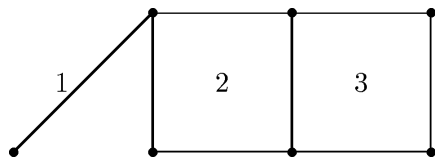
fin

4. Construction d'un maillage $\mathcal{T}(\hat{F})$ pour chaque face F comme la somme de toutes les contraintes des traces sous-jacentes.

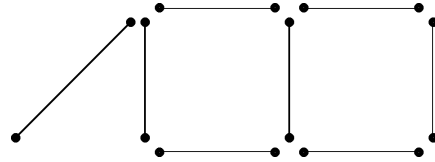
tant que $(\Delta_{\mathcal{T}(F)} > \varepsilon)$ **faire**

 | Choisir $\hat{x} = \arg \max \Delta_{\mathcal{T}(F)}$ comme un nouveau point pour raffiner $\mathcal{T}(\hat{F})$.

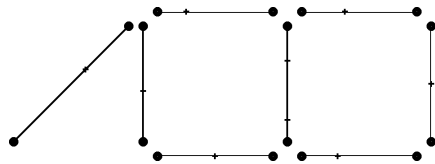
fin



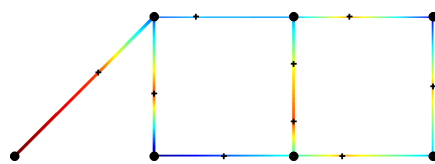
(a) Maillage de calcul : un segment et deux quadrangles



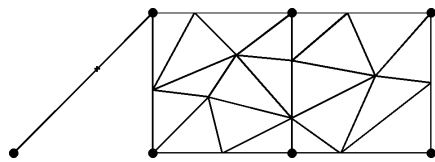
(b) Décomposition en dimensions inférieures



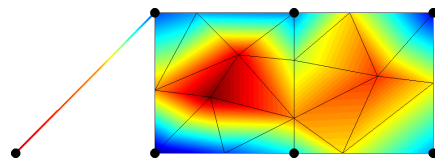
(c) Maillage des arêtes $\tilde{\mathcal{T}}_{1D}$



(d) Représentation sur les arêtes $\tilde{\mathcal{T}}_{1D}$



(e) Maillage des éléments $\tilde{\mathcal{T}}_{2D}$



(f) Représentation sur les éléments $\tilde{\mathcal{T}}_{2D}$

FIGURE 2.6 – Illustration du processus en dimension 2 : décomposition des éléments en dimensions inférieures (b), maillage et représentation des arêtes (c et d) et des éléments (e et f).

nous définissons $\Delta_{\mathcal{T}(F)}$ qui dépend donc du maillage $\mathcal{T}(F)$ de cette face. Les contributions à prendre en compte pour cet estimateur sont de deux types. Cette face peut appartenir à un ou plusieurs éléments 2D pur \mathcal{T}_{2D} , qui font partis du maillage de calcul. Cette face peut également être le support de la trace d'une ou de plusieurs fonctions 3D d'un élément \mathcal{T}_{3D} . L'indicateur d'erreurs s'écrit donc

$$\Delta_{\mathcal{T}(F)} = \max_{S \in \mathcal{T}(F)} \Delta_S = \max_{S \in \mathcal{T}(F)} \max \left(\max_{\{S \in \mathcal{T}_{2D}\}} \Delta_{\mathcal{T}(S)}, \max_{\{K \in \mathcal{T}_{3D}, S \in \mathcal{T}(\partial K)\}} \Delta_{\mathcal{T}(S)}^K \right), \quad (2.11)$$

où ∂K désigne la frontière de K et $\mathcal{T}(\partial K)$ le maillage de ce bord. Dans (2.11), deux indicateurs apparaissent : $\Delta_{\mathcal{T}(S)}$ et $\Delta_{\mathcal{T}(S)}^K$. Dans le premier cas, comme il s'agit d'un élément du maillage de calcul, cet indicateur est (2.9). Dans le second cas, la face S considérée est une partie de la frontière d'un élément volumique K du maillage de calcul. L'indicateur $\Delta_{\mathcal{T}(S)}^K$ s'écrit

$$\begin{aligned} \Delta_{\mathcal{T}(S)}^K : (f, \tilde{f}) &\in C^0(S, \mathbb{Y}) \times C^0(\tilde{S}, \mathbb{Y}) \\ &\mapsto \sup_{\hat{x} \in \hat{S}} \max \left(\alpha \left\| g_S(\hat{x}) - P^1 g_{\mathcal{T}(S)}(\hat{x}) \right\|_{\mathbb{X}}, \beta \left\| (f \circ g_S)(\hat{x}) - (\tilde{f} \circ P^1 g_{\mathcal{T}(S)})(\hat{x}) \right\|_{\mathbb{Y}} \right), \end{aligned} \quad (2.12)$$

où g_S (resp. $P^1 g_{\mathcal{T}(S)}$) est la restriction de g_K (resp. $P^1 g_{\mathcal{T}(K)}$) sur S (resp. $\tilde{S} = P^1 g_{\mathcal{T}(S)}(\hat{S})$) et f (resp. \tilde{f}) est la trace de la fonction f_{num}^K (resp. $f_{vis}^{\tilde{K}}$) sur S (resp. $\tilde{S} = P^1 g_{\mathcal{T}(S)}(\hat{S})$).

Pour une arête $E \in \tilde{\mathcal{T}}_{1D}$, un indicateur $\Delta_{\mathcal{T}(E)}$ est défini et dépend également uniquement que du maillage de cette arête $\mathcal{T}(E)$. Cette fois-ci, cet indicateur doit prendre en compte trois types de configurations. Cette arête peut appartenir à un ou plusieurs éléments 1D pur \mathcal{T}_{1D} . Cette arête peut également être la trace de fonctions 2D (d'éléments 2D \mathcal{T}_{2D}) ou de fonctions 3D (d'éléments 3D \mathcal{T}_{3D}). L'indicateur s'écrit donc

$$\Delta_{\mathcal{T}(E)} = \max_{S \in \mathcal{T}(E)} \Delta_S = \max_{S \in \mathcal{T}(E)} \max \left(\max_{\{S \in \mathcal{T}_{1D}\}} \Delta_{\mathcal{T}(S)}, \max_{\{K \in \mathcal{T}_{2D}, S \in \tilde{\mathcal{T}}_{1D}\}} \Delta_{\mathcal{T}(S)}^K, \max_{\{K \in \mathcal{T}_{3D}, S \in \tilde{\mathcal{T}}_{1D}\}} \Delta_{\mathcal{T}(S)}^K \right), \quad (2.13)$$

où là encore $\Delta_{\mathcal{T}(S)}$ correspond à (2.9) et $\Delta_{\mathcal{T}(S)}^K$ correspond à (2.12) où l'élément $K \in \mathcal{T}_h$ peut être de dimension deux ou trois.

La justification de la réalisation de l'objectif \mathbf{O}_3 est donnée dans la section 2.4.3.

2.3.4 Résumé de la méthodologie

Pour résumer, la méthodologie mise en place pour répondre aux objectifs est basée sur les points suivants :

- O₁** : Pour chaque élément, un maillage de représentation composé de simplexes est construit. Sur chacun de ces simplexes, une représentation affine est générée à partir des valeurs de la fonction numérique aux nœuds de ces simplexes.
- O₂** : Un indicateur d'erreurs *a posteriori* est mis en place pour guider la construction du maillage de représentation afin de contrôler l'erreur de visualisation. La convergence de cet estimateur en-dessous de la tolérance assure que l'objectif **O₂** est bien vérifié. Il est important de noter, que dans le cas où des éléments courbes sont utilisés, un indicateur spécifique est mis en œuvre dans le chapitre 3.
- O₃** : Une décomposition de la frontière des éléments en dimensions inférieures et un maillage par dimensions croissantes sont introduits pour vérifier l'objectif **O₃**.

2.4 Justifications mathématiques de la méthode proposée

Cette section est dédiée aux justifications mathématiques de la méthode. Son organisation suit la structure de la section précédente.

2.4.1 Réalisation de l'objectif 1

Le maillage de la cellule de référence $\mathcal{T}(\widehat{K})$ est construit de la manière suivante. Chaque élément \widehat{S} de $\mathcal{T}(\widehat{K})$ est un d_K -simplexe (i.e. l'enveloppe convexe de $d_K + 1$ points $\widehat{\mathcal{N}} := (\widehat{N}_i)_{i=1, \dots, d_K+1}$). Nous définissons l'application $\Lambda : \mathcal{N} \mapsto \Lambda(\mathcal{N})$, qui à un ensemble de $d_K + 1$ points $\mathcal{N} = (N_i)_{i=1, \dots, d_K+1}$ associe le d_K -simplexe $\Lambda(\mathcal{N})$. Nous introduisons de plus

$$\widehat{\Theta}_K := \{\widehat{\mathcal{N}} := (\widehat{N}_i)_{i=1, \dots, d_K+1} : \Lambda(\widehat{\mathcal{N}}) \in \mathcal{T}(\widehat{K})\}.$$

Alors, le maillage de \widehat{K} est défini par

$$\mathcal{T}(\widehat{K}) := \{\Lambda(\widehat{\mathcal{N}}) : \widehat{\mathcal{N}} \in \widehat{\Theta}_K\}.$$

La construction de $\mathcal{T}(K)$ est réalisée de la même manière. Nous définissons l'ensemble des nœuds de $\mathcal{T}(K)$ ainsi que l'ensemble

$$\Theta_K := \{\mathcal{N} = (N_i)_{i=1, \dots, d_K+1} := g_K(\widehat{\mathcal{N}}) : \widehat{\mathcal{N}} \in \widehat{\Theta}_K\},$$

où $g_K(\widehat{\mathcal{N}}) := (g_K(\widehat{N}_i))_{i=1, \dots, d_K+1}$ et $\widehat{\mathcal{N}} := (\widehat{N}_i)_{i=1, \dots, d_K+1}$. Le maillage en simplexes de K est défini par

$$\mathcal{T}(K) := \{\Lambda(\mathcal{N}) : \mathcal{N} \in \Theta_K\}.$$

Le lien entre le maillage de représentation de l'élément de référence et le maillage de représentation de l'élément réel est donné par le lemme 2.3. Pour cela, une dernière hypothèse est nécessaire.

H₆ : Les éléments de référence sont des polyèdres.

D'après **H₆**, il existe donc un maillage de représentation $\mathcal{T}(\widehat{K})$ qui satisfait $\widehat{K} = \bigcup_{\widehat{S} \in \mathcal{T}(\widehat{K})} \widehat{S}$ tel que pour chaque couple $(S, S') \in \mathcal{T}(\widehat{K})^2$, $S \neq S'$, $\text{Int}(S) \cap \text{Int}(S') = \emptyset$. Cette hypothèse stipule que chaque cellule de référence peut être entièrement recouverte par une union de simplexes. Dans le cas contraire, des éléments plus complexes auraient dû être utilisés.

Lemme 2.3 (Identification de $\mathcal{T}(K)$). *Soit un élément $K \in \mathcal{T}_h$, sous les hypothèses **H₃** à **H₆**, l'interpolation \mathbb{P}^1 de g_K construite à partir de $\mathcal{T}(\widehat{K})$, notée $P^1 g_{\mathcal{T}(K)}$, est*

1. une fonction bijective entre $\mathcal{T}(\widehat{K})$ et $\mathcal{T}(K)$,
2. une fonction surjective entre \widehat{K} et K .

Preuve. Soit $\Lambda(g_K(\widehat{\mathcal{N}}))$ un simplexe de dimension d_K de $\mathcal{T}(K)$, où $\Lambda(\widehat{\mathcal{N}})$ est un simplexe de $\mathcal{T}(\widehat{K})$ et $\mathcal{N} = (N_i)_{i=1, \dots, d_K+1}$ est l'ensemble des nœuds de ce simplexe. Par définition, l'approximation \mathbb{P}^1 de g_K vérifie

$$\forall \widehat{N}_i \in \widehat{\mathcal{N}}, \quad P^1 g_{\mathcal{T}(K)}(\widehat{N}_i) = g_K(\widehat{N}_i) = N_i.$$

Nous en déduisons que

$$\begin{aligned} P^1 g_{\mathcal{T}(K)}(\Lambda(\widehat{\mathcal{N}})) &= P^1 g_{\mathcal{T}(K)} \left(\left\{ \sum_{i=1}^{d_K+1} x_i \widehat{N}_i, (x_i) \in [0, 1]^{d_K+1} \right\} \right) \\ &= \left\{ \sum_{i=1}^{d_K+1} x_i N_i, (x_i) \in [0, 1]^{d_K+1} \right\} \\ &= \Lambda(g_K(\widehat{\mathcal{N}})). \end{aligned}$$

La surjectivité de $P^1 g_{\mathcal{T}(K)}$ de $\mathcal{T}(\widehat{K})$ vient par construction car $\mathcal{T}(K) = \{\Lambda(g_K(\widehat{\mathcal{N}})), \Lambda(\widehat{\mathcal{N}}) \in \mathcal{T}(\widehat{K})\}$. L'injectivité est prouvée en notant que $\mathcal{T}(\widehat{K})$ et $\mathcal{T}(K)$ ont la même cardinalité.

La surjectivité de $P^1 g_{\mathcal{T}(K)}$ entre \widehat{K} et \widetilde{K} vient de la surjectivité de $P^1 g_{\mathcal{T}(K)}$ entre $\mathcal{T}(\widehat{K})$ et $\mathcal{T}(K)$, et de l'hypothèse \mathbf{H}_6 : $\widetilde{K} = \bigcup_{\Lambda(\widehat{\mathcal{N}}) \in \mathcal{T}(\widehat{K})} \Lambda(\widehat{\mathcal{N}})$. \square

La construction est basée sur une transformation classique entre l'élément de référence et l'élément réel dite de *plane to surface* [Zienkiewicz and Phillips, 1971] dans le cas d'éléments surfaciques ($d_K = 2$). Un exemple de construction de ces maillages est montré sur la figure 2.7.

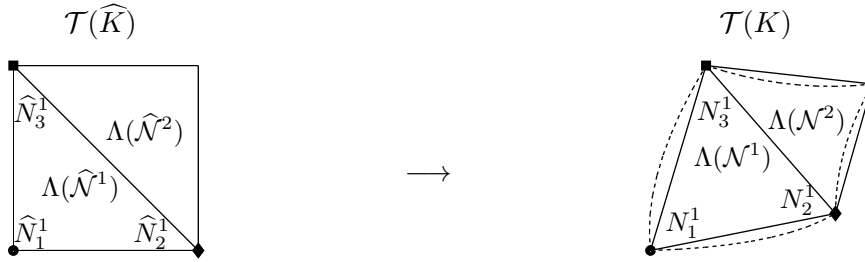


FIGURE 2.7 – Construction du maillage de représentation de l'élément de référence $\mathcal{T}(\widehat{K})$ (gauche) et du maillage de représentation $\mathcal{T}(K)$ (droite) pour un élément $K \in \mathcal{T}_h$ (en pointillé).

Malheureusement, le lemme 2.3 n'assure pas la bijectivité entre l'élément de référence et l'approximation de l'élément réel. En effet, la définition de $\mathcal{T}(K)$ n'interdit pas la création de simplexes dont l'intersection de leurs intérieurs n'est pas nulle. La conséquence de cette perte de bijectivité est que la représentation locale à l'élément n'est plus une fonction comme indiqué dans la proposition 2.4 ci-dessous. Or, il est important que la représentation locale $f_{vis}^{\widetilde{K}}$ soit bien une fonction, afin que la distance de Hausdorff (2.6) soit définie. Dans le cas contraire, l'estimation *a posteriori* est inutilisable même si l'indicateur d'erreurs est bien défini.

Proposition 2.4. *Soit un élément $K \in \mathcal{T}_h$ et $\mathcal{T}(K)$ son maillage de représentation. Sous les hypothèses \mathbf{H}_1 à \mathbf{H}_6 et en suivant les notations précédentes, pour une fonction f_{num}^K non triviale, $f_{vis}^{\widetilde{K}}$ est une fonction de \widetilde{K} si et seulement si $P^1 g_{\mathcal{T}(K)}$ est une fonction injective de \widehat{K} vers \widetilde{K} .*

Preuve. Soit $x \in \widetilde{K}$ et $P^1 g_{\mathcal{T}(K)}$ une fonction injective. Par le lemme 2.3, $P^1 g_{\mathcal{T}(K)}$ est donc une fonction bijective et il existe au moins un simplexe contenant x . Il y a deux possibilités :

- Cas 1 : soit le point appartient à l'intérieur de ce simplexe,
- Cas 2 : soit il se trouve sur sa frontière.

Dans le premier cas, si x appartient à l'intérieur d'un simplexe $\Lambda(\mathcal{N})$, alors grâce à la bijectivité de $P^1g_{\mathcal{T}(K)}$ et l'hypothèse **H₆**, ce simplexe est unique et $f_{vis}^{\tilde{K}}(x) = f_{vis}^{\Lambda(\mathcal{N})}(x)$.

Dans le second cas, x appartient à la frontière d'un ou de plusieurs simplexes. Plusieurs configurations sont possibles :

- Configuration 1 : x est un nœud d'un simplexe,
- Configuration 2 : x appartient à une arête,
- Configuration 3 : x appartient à une face.

Si x est un nœud d'un simplexe, $x = N$, alors $f_{vis}^{\tilde{K}}(x) = f_{num}^K(N)$.

Si x appartient à une arête tel que $x = x_1N_1 + x_2N_2$ où $(N_i)_{i=1,2}$ et $(x_i)_{i=1,2}$ sont uniques par bijectivité de $P^1g_{\mathcal{T}(K)}$, alors $f_{vis}^{\tilde{K}}(x) = \sum_{i=1}^2 x_i f_{num}^K(N_i)$ et cela prouve que $f_{vis}^{\tilde{K}}$ est une fonction sur \tilde{K} .

Le même résultat est vrai dans le cas où x appartient à une face sauf qu'il y a maintenant trois sommets au lieu de deux.

Réciproquement, si $P^1g_{\mathcal{T}(K)}$ n'est pas une fonction injective, un élément x de \tilde{K} peut être l'image de différents points $\hat{x}_j \in \widehat{K}$ où $j > 1$, $j \in \mathbb{N}$. Nous en déduisons que le point x peut appartenir à plusieurs simplexes $\Lambda(\mathcal{N}^j)$ tel que

$$\{x\} = \left\{ \sum_{i=1}^{d_K+1} x_i^j N_i^j / x \in \Lambda(\mathcal{N}^j), \mathcal{N}^j = (N_i^j)_{i=1,\dots,d_K+1}, (x_i^j)_j \in [0, 1]^{d_K+1} \right\}.$$

Il vient que

$$f_{vis}^{\tilde{K}}(\{x\}) = \left\{ f_{vis}^{\Lambda(\mathcal{N})}(x) / \forall \Lambda(N) \in \mathcal{T}(K), x \in \Lambda(N) \right\} = \left\{ \sum_{i=1}^{d_K+1} x_i^j f_{num}^K(N_i^j) \right\},$$

peut avoir plusieurs valeurs puisque f_{num}^K n'est pas triviale. Cela implique que $f_{vis}^{\tilde{K}}$ n'est pas une fonction sur \tilde{K} . \square

2.4.2 Réalisation de l'objectif 2

Le lien entre la distance de Hausdorff (2.6) et l'application $\Delta_{\mathcal{T}(K)}$ définie par (2.9) est donné par le lemme suivant.

Lemme 2.5. *Soit $K \in \mathcal{T}_h$, sous les hypothèses **H₃** et **H₆**, la fonction $\Delta_{\mathcal{T}(K)}$, donné par (2.9), vérifie*

$$\forall (f, \tilde{f}) \in C^0(K, \mathbb{Y}) \times C^0(\tilde{K}, \mathbb{Y}), d_H^{\mathbb{X} \times \mathbb{Y}}(f, \tilde{f}) \leq \Delta_{\mathcal{T}(K)}(f, \tilde{f}).$$

Preuve. Tout d'abord, soit un point arbitraire $x \in K$. Par l'hypothèse **H₃**, il existe $\hat{x} = g_K^{-1}(x)$. Comme $P^1g_{\mathcal{T}(K)}(\hat{x}) \in \tilde{K}$, nous en déduisons que

$$\inf_{\tilde{x} \in \tilde{K}} d\left((x, f(x)), (\tilde{x}, \tilde{f}(\tilde{x}))\right) \leq d\left((g_K(\hat{x}), f \circ g_K(\hat{x})), (P^1g_{\mathcal{T}(K)}(\hat{x}), \tilde{f} \circ P^1g_{\mathcal{T}(K)}(\hat{x}))\right),$$

ce qui implique que

$$\sup_{x \in K} \inf_{\tilde{x} \in \tilde{K}} d\left((x, f(x)), (\tilde{x}, \tilde{f}(\tilde{x}))\right) \leq \sup_{\hat{x} \in \hat{K}} d\left((g_K(\hat{x}), f \circ g_K(\hat{x})), (P^1 g_{\mathcal{T}(K)}(\hat{x}), \tilde{f} \circ P^1 g_{\mathcal{T}(K)}(\hat{x}))\right).$$

Réciproquement, soit $\tilde{x} \in \tilde{K}$ un point arbitraire. Par le lemme 2.3 et la surjectivité de $P^1 g_{\mathcal{T}(K)}$, il existe $\hat{x} \in \hat{K}$ tel que $\tilde{x} = P^1 g_{\mathcal{T}(K)}(\hat{x})$. Comme $g_K(\hat{x}) \in K$, il vient que

$$\inf_{x \in K} d\left((x, f(x)), (\tilde{x}, \tilde{f}(\tilde{x}))\right) \leq d\left((g_K(\hat{x}), f \circ g_K(\hat{x})), (\tilde{x}, \tilde{f}(\tilde{x}))\right),$$

et donc

$$\sup_{\tilde{x} \in \tilde{K}} \inf_{x \in K} d\left((x, f(x)), (\tilde{x}, \tilde{f}(\tilde{x}))\right) \leq \sup_{\hat{x} \in \hat{K}} d\left((g_K(\hat{x}), f \circ g_K(\hat{x})), (P^1 g_{\mathcal{T}(K)}(\hat{x}), \tilde{f} \circ P^1 g_{\mathcal{T}(K)}(\hat{x}))\right).$$

Finalement, les deux parties de la distance d_H sont bornées par $\Delta_{\mathcal{T}(K)}(f, \tilde{f})$ et nous obtenons le résultat. \square

Dans la proposition suivante, nous montrons que $\Delta_{\mathcal{T}(K)}$ est bien un indicateur d'erreurs *a posteriori*.

Proposition 2.6 (Estimation *a posteriori*). *Sous les hypothèses \mathbf{H}_1 à \mathbf{H}_6 , si $P^1 g_{\mathcal{T}(K)}$ est une fonction injective pour tout élément $K \in \mathcal{T}_h$, alors*

$$d_H^{\mathbb{X} \times \mathbb{Y}}(f_{num}, f_{vis}) = \max_{K \in \mathcal{T}_h} d_H^{\mathbb{X} \times \mathbb{Y}}\left(f_{num}^K, f_{vis}^{\tilde{K}}\right) \leq \max_{K \in \mathcal{T}_h} \Delta_{\mathcal{T}(K)}\left(f_{num}^K, f_{vis}^{\tilde{K}}\right).$$

Preuve. Soit $K \in \mathcal{T}_h$ un élément arbitraire vérifiant les hypothèses \mathbf{H}_1 à \mathbf{H}_3 . Il suffit de vérifier que f_{num}^K et $f_{vis}^{\tilde{K}}$ sont continues et alors le résultat vient en appliquant le lemme 2.5 sur K (l'hypothèse \mathbf{H}_6 est satisfaite). Or, f_{num}^K est continue sur K car les hypothèses \mathbf{H}_4 et \mathbf{H}_5 sont vérifiées. Par hypothèse, $P^1 g_{\mathcal{T}(K)}$ est une fonction injective, donc d'après la proposition 2.4, la fonction $f_{vis}^{\tilde{K}}$ est continue sur \tilde{K} . \square

Dans la proposition 2.6, une hypothèse sur l'injectivité de la fonction qui approxime la transformation géométrique apparait. Garantir que celle-ci reste vraie est primordiale car sinon la représentation locale qui est créée n'est pas bien définie comme le montre la proposition 2.4. Sous cette hypothèse, la proposition 2.6 établit une estimation *a posteriori* et montre que l'indicateur d'erreurs peut guider la construction de la représentation. Notons que $\Delta_{\mathcal{T}(K)}$ peut toujours être utilisé pour construire la représentation car cet indicateur est dans tous les cas bien défini. Nous détaillons dans la section 2.5.3 cette discussion sur l'éventuelle perte de bijectivité qui aurait pu entraver la convergence de l'algorithme.

2.4.3 Réalisation de l'objectif 3

Pour garantir l'objectif \mathbf{O}_3 , notre approche est basée sur la décomposition de toutes les frontières des éléments en dimensions inférieures. Pour cela, les éléments du maillage de calcul sont regroupés dans des ensembles $\mathcal{T}_{iD} \subset \mathcal{T}_h$, pour $i = 1, 2$ ou 3 , suivants leur dimension

$$\mathcal{T}_{iD} := \{K \in \mathcal{T}_h, d_K = i\}. \quad (2.14)$$

Nous définissons maintenant les ensembles $\tilde{\mathcal{T}}_iD$ où $i = 1, 2$ ou 3 . L'ensemble $\tilde{\mathcal{T}}_3D$ est le plus simple car il ne fait intervenir que \mathcal{T}_{3D}

$$\tilde{\mathcal{T}}_3D = \mathcal{T}_{3D}. \quad (2.15)$$

En effet, comme ce sont les éléments de dimension maximale, cet ensemble n'est pas enrichi. Nous définissons maintenant l'ensemble des faces $\mathcal{F}(X)$ par

$$\begin{aligned} \mathcal{F}(X) = & \left\{ F \subset \partial X \cap \partial K, K \in \mathcal{T}_{3D} \right\} \\ & \cup \left\{ \Sigma, \exists!(K, K') \in \mathcal{T}_{3D}^2, K \neq K', \Sigma = \partial K \cap \partial K', \text{Int}_{2D}(\Sigma) \neq \emptyset \right\}, \end{aligned}$$

où ∂X (resp. ∂K) désigne la frontière de l'ensemble X (resp. de K) et Int_{2D} désigne l'intérieur d'un élément 2D. A l'aide de cet ensemble, nous pouvons maintenant définir $\tilde{\mathcal{T}}_2D$ comme

$$\tilde{\mathcal{T}}_2D = \mathcal{T}_{2D} \cup \mathcal{F}(X). \quad (2.16)$$

De la même manière, nous définissons l'ensemble des arêtes $\mathcal{E}(X)$ par

$$\begin{aligned} \mathcal{E}(X) = & \left\{ E \subset \partial X \cap \partial K, K \in \tilde{\mathcal{T}}_2D \right\} \\ & \cup \left\{ \Sigma, \exists!(K, K') \in \tilde{\mathcal{T}}_2D^2, K \neq K', \Sigma = \partial K \cap \partial K', \text{Int}_{1D}(\Sigma) \neq \emptyset \right\}, \end{aligned}$$

où Int_{1D} désigne l'intérieur d'un élément 1D. Ces arêtes contiennent des frontières d'éléments 2D ou proviennent de faces qui elles-mêmes sont des frontières d'éléments 3D. Nous en déduisons l'ensemble $\tilde{\mathcal{T}}_1D$ par

$$\tilde{\mathcal{T}}_1D = \mathcal{T}_{1D} \cup \mathcal{E}(X). \quad (2.17)$$

Nous allons maintenant montrer que le maillage par dimensions croissantes permet de remplir l'objectif 3. Pour cela, il est important qu'une fois le maillage du bord construit, on n'y touche plus. Sinon, il est possible de générer des sauts artificiels. Dans le lemme suivant, nous montrons que si le maillage a bien été traité, lors du passage à la dimension supérieure, tous les points ajoutés au maillage de représentation se trouvent à l'intérieur de l'élément et donc pas sur le bord.

Lemme 2.7. *Soit $K \in \mathcal{T}_{iD}$ où $i = 2$ ou 3 , f_{num}^K une solution numérique et $\mathcal{T}(K)$ un maillage de représentation. Nous supposons que les constructions des représentations des éléments de $\tilde{\mathcal{T}}_{(i-1)D} \cap K$ ont convergé.*

Si $\Delta_{\mathcal{T}(K)}(f_{num}^K, f_{vis}^{\tilde{K}}) > \varepsilon$, alors $\arg \max \Delta_{\mathcal{T}(K)} \in \text{Int}(\hat{K})$.

Preuve. L'indicateur *a posteriori* peut être décomposé en deux parties

$$\Delta_{\mathcal{T}(K)}(f_{num}^K, f_{vis}^{\tilde{K}}) = \max \left(\sup_{\hat{x} \in \partial \hat{K}} \delta_{\mathcal{T}(K)}(f_{num}^K, f_{vis}^{\tilde{K}}; \hat{x}), \sup_{\hat{x} \in \text{Int}(\hat{K})} \delta_{\mathcal{T}(K)}(f_{num}^K, f_{vis}^{\tilde{K}}; \hat{x}) \right),$$

où la première partie concerne le bord de l'élément tandis que la seconde se concentre sur l'intérieur. Nous notons $\hat{y} \in \partial \hat{K}$ le point réalisant le maximum sur la frontière (première partie sous le max). Alors, il existe un élément $\Sigma \in \tilde{\mathcal{T}}_{(i-1)D}$ tel que $\hat{y} \in \hat{\Sigma}$. Or, par hypothèse, les constructions des bords ont convergé et il vient que

$$\delta_{\mathcal{T}(K)}(f_{num}^K, f_{vis}^{\tilde{K}}; \hat{y}) \leq \Gamma_{\mathcal{T}(\Sigma)} \leq \varepsilon.$$

Donc,

$$\sup_{\hat{x} \in \partial \widehat{K}} \delta_{\mathcal{T}(K)}(f_{num}^K, f_{vis}^{\widehat{K}}; \hat{x}) \leq \varepsilon < \sup_{\hat{x} \in \text{Int}(\widehat{K})} \delta_{\mathcal{T}(K)}(f_{num}^K, f_{vis}^{\widehat{K}}; \hat{x}).$$

Finalement,

$$\arg \max \Delta_{\mathcal{T}(K)} \in \text{Int}(\widehat{K}). \quad (2.18)$$

Le point se trouve nécessairement à l'intérieur de l'élément. \square

Le lemme 2.7 assure que les points sont ajoutés à l'intérieur de l'élément et non pas sur le bord si le maillage du bord a déjà convergé. La proposition suivante montre maintenant que la méthodologie permet de vérifier l'objectif 3.

Proposition 2.8 (Réalisation de l'objectif \mathbf{O}_3). *Sous les hypothèses \mathbf{H}_1 à \mathbf{H}_5 , soit $(\mathcal{T}(\Sigma))_{\Sigma \in \widetilde{\mathcal{T}}_{iD}, i=1,3}$ un ensemble de maillages en simplexes satisfaisant*

$$\forall K \in \mathcal{T}_{2D}, \mathcal{T}(K)|_{\partial \widetilde{K}} = \mathcal{T}(K) \cap \partial \widetilde{K} = \bigcup_{\Sigma \in \mathcal{E}(K)} \mathcal{T}(\Sigma), \quad (2.19)$$

$$\forall K \in \mathcal{T}_{3D}, \mathcal{T}(K)|_{\partial \widetilde{K}} = \mathcal{T}(K) \cap \partial \widetilde{K} = \bigcup_{F \in \mathcal{F}(K)} \mathcal{T}(F), \quad (2.20)$$

$$\forall K \in \mathcal{T}_{3D}, \mathcal{T}(K)|_{1D} = \bigcup_{\Sigma \in \mathcal{E}(K)} \mathcal{T}(\Sigma), \quad (2.21)$$

où $\widetilde{K} = \bigcup_{S \in \mathcal{T}(K)} S$, $\mathcal{F}(K) = \widetilde{\mathcal{T}}_{2D} \cap K$, $\mathcal{E}(K) = \widetilde{\mathcal{T}}_{1D} \cap K$ et $\mathcal{T}(K)|_{1D}$ pour les éléments de dimension 3 désigne leurs arêtes (bord de dimension un). Alors, la représentation f_{vis} de f_{num} construite pour chaque élément $K \in \mathcal{T}_h$ satisfait l'objectif \mathbf{O}_3 .

Preuve. Le lemme 2.7 prouve que les propriétés données par les équations (2.19), (2.20) et (2.21) restent vraies durant le processus de maillage de chaque élément. Nous voulons regarder si des discontinuités artificielles apparaissent aux interfaces entre les éléments. Ces interfaces peuvent être de deux types : des faces ou des arêtes.

Soit $F \in \widetilde{\mathcal{T}}_{2D} \setminus \partial X$. Nous considérons une face qui n'appartient pas au bord du domaine. En effet, sinon, si $F \in \partial X$, elle appartient nécessairement à un seul élément et donc il n'est pas possible d'avoir de saut. Comme $F \in \widetilde{\mathcal{T}}_{2D}$, il existe deux éléments, K et K' de \mathcal{T}_h , tels que $K \neq K'$ et $F = \partial K \cap \partial K'$. Supposons que $f_{num|F}^K = f_{num|F}^{K'}$ sur F où $f_{num|F}^K$ désigne la restriction de f_{num}^K à F . Par hypothèse, il existe un maillage $\mathcal{T}(F)$ de F . Soit S un triangle arbitraire de $\mathcal{T}(F)$. Par construction, $f_{vis|S}^{\widetilde{K}}$ et $f_{vis|S}^{\widetilde{K}'}$ sont fonctions affines définies à partir des valeurs de f_{num}^K et de $f_{num}^{K'}$ aux sommets de ce triangle. Or, par continuité de f_{num} sur F , en chaque sommet les valeurs de f_{num}^K et $f_{num}^{K'}$ sont identiques. Ceci implique que $f_{vis|S}^{\widetilde{K}} = f_{vis|S}^{\widetilde{K}'}$. De plus, cette propriété est vraie pour n'importe quel triangle de la face. Finalement, $f_{vis|F}^{\widetilde{K}} = f_{vis|F}^{\widetilde{K}'}$ et aucune discontinuité artificielle n'est ajoutée.

Soit $E \in \widetilde{\mathcal{T}}_{1D}$ tel que $E \not\subset \partial X$, sinon cette arête n'appartient qu'à un seul élément et donc il ne peut pas y avoir de problème. Ainsi, cette arête appartient à au moins deux éléments. Notons K et K' , deux éléments tels que $E = \partial K \cap \partial K'$. De la même manière que pour les faces, supposons que $f_{num|E}^K = f_{num|E}^{K'}$ sur E . Soit S une arête de $\mathcal{T}(E)$ dont les extrémités sont notées a et b . Par construction, $f_{vis|S}^{\widetilde{K}}$ et $f_{vis|S}^{\widetilde{K}'}$ sont des fonctions affines définies de $f_{num}^K(a)$ vers $f_{num}^K(b)$, par

continuité de f_{num} sur E . Donc, $f_{vis|S}^{\tilde{K}} = f_{vis|S}^{\tilde{K}'}$. Nous en déduisons que cette propriété est vraie pour chaque arête S de $\mathcal{T}(E)$. Finalement, $f_{vis|E}^{\tilde{K}} = f_{vis|E}^{\tilde{K}'}$ et aucune discontinuité artificielle n'est ajoutée. \square

2.5 Algorithme pour générer une représentation satisfaisant les objectifs

Un pseudo-algorithme basé sur un maillage par dimensions croissantes a été mis en place dans la section 2.3. Schématiquement, ce principe est illustré sur la figure 2.8.



FIGURE 2.8 – Principe général, maillage par dimensions croissantes

Le maillage de chaque élément, quelque soit la dimension, suit la même approche. Tout d'abord, un premier maillage de représentation en récupérant les maillages des bords qui ont déjà été maillés est généré. Comme indiqué dans la figure 2.9, un indicateur *a posteriori* est utilisé pour évaluer l'erreur de visualisation. Ensuite, un processus auto-adaptatif permet de raffiner le maillage de représentation si nécessaire.

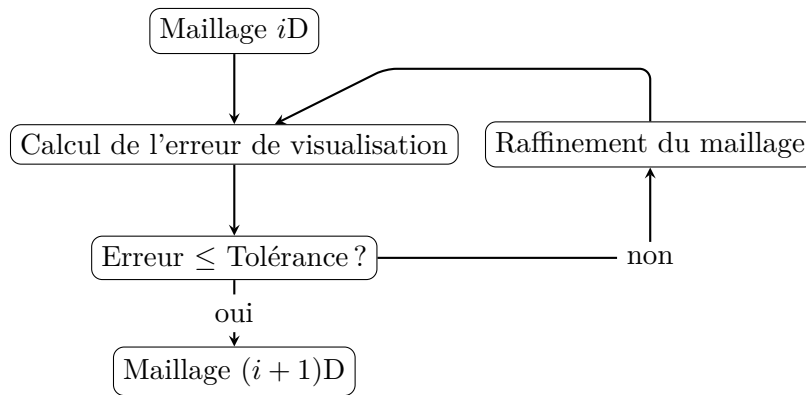


FIGURE 2.9 – Principe de la génération du maillage de représentation à une dimension donnée

Afin d'estimer les indicateurs d'erreurs définis précédemment, il est nécessaire d'évaluer des maxima. Pour cela, nous utilisons un algorithme classique que nous rappelons dans la section suivante. Ensuite, nous donnons l'algorithme complet permettant de traiter les éléments droits.

2.5.1 Un algorithme de recherche d'extrema : DIRECT

Pour évaluer les indicateurs d'erreurs, de nombreuses recherches d'extrema sont nécessaires. Cette recherche est basée sur un algorithme d'optimisation globale déterministe : DIRECT. Cet algorithme, initialement proposé par Jones et al. [Jones et al., 1993], a été développé pour résoudre des problèmes d'optimisation globale. C'est un processus déterministe de type séparation et évaluation ou *branch and bound* en anglais. L'appellation DIRECT, de l'acronyme anglais DIviding RECTangles, vient du but originel qui était de traiter des hyperrectangles et du fait que la procédure est basée sur une subdivision de certains des éléments. Dans notre

cas, les domaines de recherche sont des simplexes, ce qui implique une légère modification [Clausen and Žilinskas, 2002] de l’algorithme initial. Le but est donc de trouver les valeurs maximales d’une ou de plusieurs fonctions dans un domaine. Pour cela, le domaine est récursivement divisé en sous-domaines et les fonctions sont évaluées au barycentre de chaque sous-domaine pour orienter la découpe des éléments. Finalement, DIRECT renvoie le point où le maximum est atteint et la valeur de ce maximum.

Le principe général de DIRECT est présenté dans l’algorithme 4. Les étapes sont explicitées dans la suite.

Algorithme 4 : Principe de fonctionnement de l’algorithme DIRECT

Entrées : Fonction dont on cherche le maximum, élément supportant la fonction.

Sorties : Maximum et point réalisant ce maximum.

Initialiser le premier élément.

tant que Critères d’arrêt non vérifiés **faire**

 | Sélectionner des éléments à découper.

 | Découper ces éléments.

fin

Définition 2.9 (Niveau d’un élément). *Le niveau d’un élément correspond à son niveau de découpe. Le premier élément est initialisé au niveau 0. Lorsqu’un élément de niveau i est découpé, ses éléments fils sont de niveau $i + 1$.*

Sélection des éléments à découper dans DIRECT

Pour chaque niveau, du niveau courant (niveau maximum de découpe effectué) au niveau 0, on cherche parmi les éléments qui n’ont pas de fils, un qui réalise l’extremum et on le découpe. L’intérêt est que l’on réalise à la fois une recherche locale mais également globale. Le fait de continuer à tester tous les niveaux permet de découper des éléments des premiers niveaux, qui sont potentiellement assez gros. Cela permet de sélectionner des zones non explorées, qui pourraient être « oubliées » autrement.

Découpage des éléments dans DIRECT

Nous commençons par créer des nœuds fils au milieu des arêtes. Ainsi, lorsqu’un segment (resp. triangle) est découpé, 2 (resp. 3) nœuds fils sont créés. Lorsqu’un tétraèdre est découpé, 6 nœuds fils sont créés [Žilinskas and Žilinskas, 2002]. A partir de ces nœuds fils, les éléments fils (2 en 1D, 4 en 2D, 8 en 3D) sont créés. Les premières étapes de l’algorithme DIRECT en dimension deux sont montrés sur la figure 2.10.

La figure 2.11 illustre les découpages en dimension trois. L’intérêt de ces subdivisions est que tous les points sont définis par des coordonnées barycentriques par rapport aux sommets de l’élément initial et sont donc très faciles à générer et repérer.

Critères d’arrêt de DIRECT

Nous définissons plusieurs critères d’arrêt de l’algorithme DIRECT.

- Le premier critère est basé sur le niveau de découpage maximal possible. Lorsqu’un élément a atteint le niveau maximal donné par l’utilisateur, l’algorithme s’arrête. Il y a donc un premier paramètre à fixer par l’utilisateur.

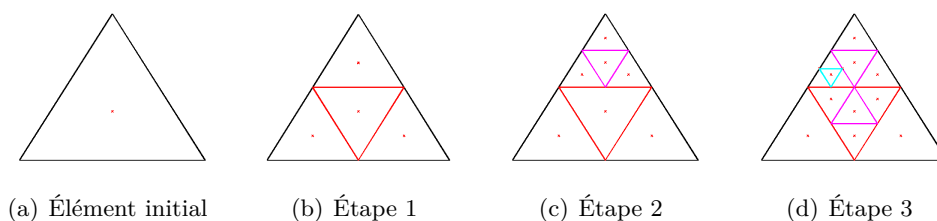


FIGURE 2.10 – Premières étapes de l'algorithme DIRECT sur un triangle

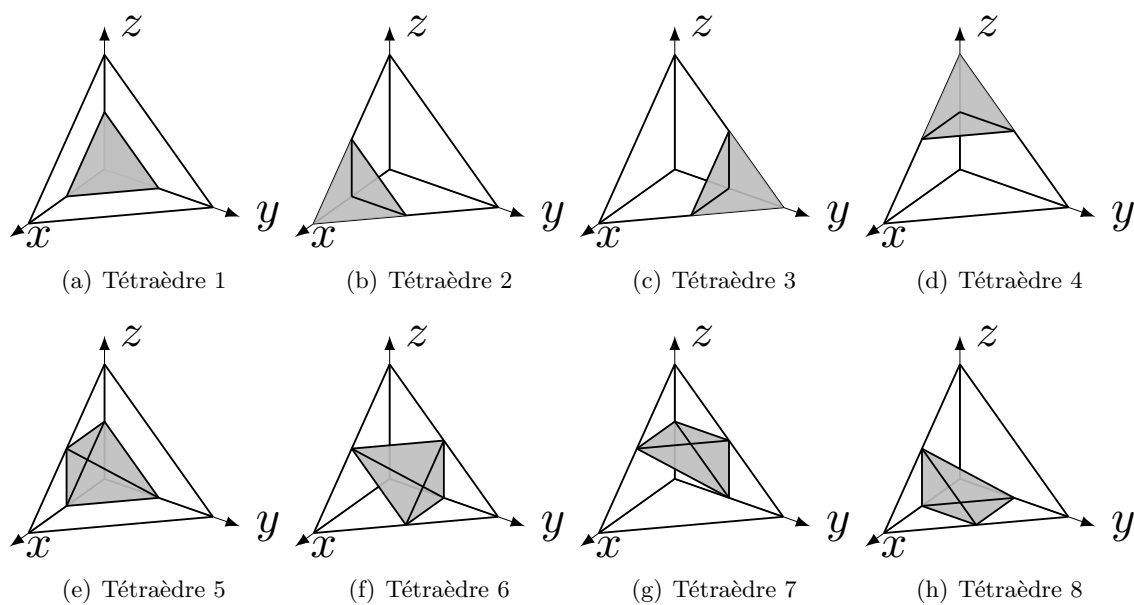


FIGURE 2.11 – Un tétraèdre est découpé régulièrement en 8 tétraèdres

- Le second critère est basé sur la stagnation des valeurs trouvées par l'algorithme. En effet, si entre deux découpes successives les extrema sont quasiment identiques, et à condition que le niveau de découpe ait atteint un niveau minimum, l'algorithme s'arrête. Imposer un niveau minimum permet d'assurer une exploration suffisamment poussée de tout le domaine.

En plus de ces niveaux minimum et maximum, nous pouvons définir un niveau de découpe initial qui donne une configuration de départ différente. L'intérêt est que pour tous les éléments de niveau inférieur à ce niveau initial, il est inutile d'évaluer les fonctions à leurs barycentres ni de les tester, puisqu'ils vont être découpés par la suite. Cela permet de limiter le nombre d'évaluations.

2.5.2 Algorithme complet pour les éléments droits

L'algorithme complet est donné dans l'algorithme 5. Nous décrivons maintenant les spécificités de chaque étape.

Calcul des extrema initiaux

Plusieurs extrema doivent être calculés. Pour la ou les fonctions, le maximum et le minimum de chaque composante de chaque fonction sont calculés. Pour la géométrie, nous calculons également ces extrema. Cela permet de définir les constantes de pondérations α et β , de l'équation (2.8), introduits dans la distance de Hausdorff. Ces coefficients sont globaux. Néanmoins, les extrema sont locaux à chaque élément du maillage de calcul donc cette recherche peut se faire de manière naturellement parallèle. Un seul DIRECT est utilisé pour calculer tous ces extrema. Cela signifie que toutes les contraintes (correspondant à chaque extrema à évaluer) sont empilées. Le nombre d'appels de l'algorithme DIRECT est ainsi limité ainsi que le nombre d'évaluations effectuées dans DIRECT.

Décomposition de la frontière en dimensions inférieures

Cette étape est centrale afin de réaliser l'objectif 3. Les faces et arêtes de chaque élément sont créées si elles n'existent pas. Chaque interface n'est créée qu'une seule fois.

Assignment des traces

Sur les arêtes et faces créées précédemment, les traces des fonctions sous-jacentes sont associées.

Maillage des éléments

Lorsque les éléments sont de dimensions deux ou trois, le maillage de représentation est initialisé à partir des maillages des bords qui viennent d'être construits. La construction des maillages des éléments $\tilde{\mathcal{T}}_{iD}$ suit un algorithme de Bowyer-Watson [Bowyer, 1981, Watson, 1981, Rebay, 1993]. C'est une méthode de remaillage adaptative guidée par un indicateur *a posteriori* qui suivant la dimension considérée est $\Delta_{\mathcal{T}(E)}$ (2.13), $\Delta_{\mathcal{T}(F)}$ (2.11) ou $\Delta_{\mathcal{T}(K)}$ (2.9).

Génération des sorties

La dernière étape de l'algorithme est la sortie des données au format spécifique d'un logiciel de visualisation (par exemple Gmsh [Remacle et al., 2007], Paraview [Ayachit, 2015], VisIt

Algorithme 5 : Construction de la représentation f_{vis} pour des éléments droits

Entrées : Maillage \mathcal{T}_h , solution numérique f_{num} , tolérance ε .

Sorties : Représentation f_{vis} .

1. Calcul des extrema des fonctions et de la géométrie. Évaluation des poids α et β , équation (2.8), introduits pour la distance de Hausdorff (2.6).
2. Décomposition de la frontière des éléments en dimensions inférieures : 3D \rightarrow 2D \rightarrow 1D. Création des arêtes et des faces des éléments.
3. Assignment des traces des fonctions associées aux arêtes et aux faces créées en 2.
4. Maillage des éléments 1D et des arêtes (notés $\tilde{\mathcal{T}}_{1D}$) :

pour chaque arête $E \in \tilde{\mathcal{T}}_{1D}$ **faire**

 Évaluation de l'indicateur *a posteriori* $\Delta_{\mathcal{T}(E)}$.

tant que $\Delta_{\mathcal{T}(E)} > \varepsilon$ **faire**

 Raffinement du maillage de l'élément de référence $\mathcal{T}(\hat{E})$, du maillage de représentation $\mathcal{T}(E)$ et mise à jour de $\Delta_{\mathcal{T}(E)}$.

fin

fin

5. Maillage des éléments 2D et des faces (notés $\tilde{\mathcal{T}}_{2D}$) :

pour chaque face $F \in \tilde{\mathcal{T}}_{2D}$ **faire**

 Construction d'un maillage initial $\mathcal{T}(\hat{F})$ à partir des maillages des arêtes $\hat{E} \in \mathcal{E}(\hat{F})$.

 Évaluation de l'indicateur *a posteriori* $\Delta_{\mathcal{T}(F)}$.

tant que $\Delta_{\mathcal{T}(F)} > \varepsilon$ **faire**

 Raffinement du maillage de l'élément de référence $\mathcal{T}(\hat{F})$, du maillage de représentation $\mathcal{T}(F)$ et mise à jour de $\Delta_{\mathcal{T}(F)}$.

fin

fin

6. Maillage des éléments 3D (notés $\tilde{\mathcal{T}}_{3D}$) :

pour chaque élément $K \in \tilde{\mathcal{T}}_{3D}$ **faire**

 Construction d'un maillage initial $\mathcal{T}(\hat{K})$ à partir des maillages des faces $\hat{F} \in \mathcal{F}(\hat{K})$.

 Évaluation de l'indicateur *a posteriori* $\Delta_{\mathcal{T}(K)}$.

tant que $\Delta_{\mathcal{T}(K)} > \varepsilon$ **faire**

 Raffinement du maillage de l'élément de référence $\mathcal{T}(\hat{K})$, du maillage de représentation $\mathcal{T}(K)$ et mise à jour de $\Delta_{\mathcal{T}(K)}$.

fin

fin

7. Génération des sorties.
-

[Childs et al., 2012]). Pour chaque élément du maillage de calcul, un maillage de représentation de l'élément a d'abord été construit. Ensuite, un maillage de l'élément réel en a été déduit. Finalement, des fonctions affines ont été introduites sur chacun des simplexes du maillage de représentation. Sur une interface, une arête ou une face, entre plusieurs éléments, un maillage unique est généré mais plusieurs nœuds sont créés, à savoir un pour chaque élément et leurs valeurs peuvent être différentes. C'est en particulier le cas lorsque la fonction n'est pas continue à cette interface. Cela permet notamment d'observer des sauts numériques dues aux schémas discontinus ou des sauts physiques. Cette duplication des nœuds est également utilisée dans [Leone et al., 1998] pour représenter des champs discontinus. Finalement, pour chaque élément du maillage de calcul, nous disposons d'une représentation locale, affine par morceaux, qui peut être écrite de façon à être lue par n'importe quel logiciel de visualisation.

2.5.3 Théorème sur la réponse aux objectifs

Dans cette section, nous établissons un théorème assurant qu'à condition de convergence de l'algorithme 5, la représentation satisfait les objectifs pour le traitement d'éléments droits. Pour cela, les résultats démontrés dans la section 2.4 sont utilisés. La proposition 2.6 demande que l'approximation de la transformation géométrique $P^1g_{\mathcal{T}(K)}$ soit bijective pour chaque élément du maillage. Or, dans la construction proposée, nous ne l'assurons pas. Néanmoins, lors du traitement d'éléments droits, nous n'avons pas observé de problèmes en pratique dans le chapitre 4 dédié aux validations numériques. Dans la cas où on détecterait une perte de bijectivité (voir notamment le chapitre 3), un autre point à ajouter au maillage de représentation peut être choisi. En effet, comme la transformation géométrique g_K est bijective, $P^1g_{\mathcal{T}(K)}$ tend donc, pour un maillage de représentation suffisamment raffiné, vers cette même fonction bijective. Il existe donc des maillages sans perte de bijectivité. Ce petit problème afin d'assurer que $P^1g_{\mathcal{T}(K)}$ reste une fonction bijective pour des éléments droits n'a pas été étudié pendant la thèse. Afin d'avoir un théorème rigoureux, $P^1g_{\mathcal{T}(K)}$ est supposé être bijectif. Même si, dans le cas où dans le processus de maillage, à un certain moment la fonction $P^1g_{\mathcal{T}(K)}$ n'est pas bijective, l'indicateur d'erreurs $\Delta_{\mathcal{T}(K)}$ est toujours défini et donc cela n'empêche pas la convergence de l'algorithme. De plus, l'ajout de points et les opérations effectuées pour avoir un maillage de Delaunay peuvent permettre de « retrouver » une fonction $P^1g_{\mathcal{T}(K)}$ bijective.

Théorème 2.10. *Soit ε une tolérance fixée et f_{num} une solution numérique appropriée vérifiant les hypothèses \mathbf{H}_1 à \mathbf{H}_6 . On suppose que l'ensemble des éléments du maillages sont des éléments droits et que $P^1g_{\mathcal{T}(K)}$ est une fonction bijective pour chaque élément K . Si l'algorithme 5 a convergé, alors la représentation f_{vis} satisfait les objectifs \mathbf{O}_1 - \mathbf{O}_2 - \mathbf{O}_3 .*

Preuve. L'objectif \mathbf{O}_1 est vérifié par construction car chaque élément du maillage de calcul est approché par une union de simplexes et des fonctions affines sont définies sur chacun de ces simplexes.

Plusieurs ingrédients sont nécessaires pour vérifier l'objectif \mathbf{O}_2 . Pour chaque élément, la représentation locale générée doit être une fonction pour que la distance de Hausdorff soit bien définie. La proposition 2.4 montre que la représentation locale $f_{vis}^{\tilde{K}}$ est une fonction si et seulement si $P^1g_{\mathcal{T}(K)}$ est une fonction bijective, ce qui est le cas par hypothèse. Finalement, la distance de Hausdorff est bien définie à chaque étape de l'algorithme et est contrôlée par l'indicateur *a posteriori* (2.9) par la proposition 2.6. L'objectif \mathbf{O}_2 est donc satisfait si pour chaque élément, l'indicateur est borné par la tolérance.

L'objectif \mathbf{O}_3 est rempli par la proposition 2.8. □

Remarque 2.11. Dans le théorème 2.10, la convergence de l'algorithme 5 n'est pas prouvée puisque l'approximation de la transformation géométrique peut ne pas être bijective. Cependant, comme l'espace des fonctions linéaires continues par morceaux est dense dans l'espace des fonctions continues, lorsque le maillage devient suffisamment fin, la fonction $P^1g_{\mathcal{T}(K)}$ tend vers g_K qui est bijective par hypothèse.

2.6 Conclusions du chapitre

Dans ce chapitre, nous avons tout d'abord défini ce qu'est une « bonne » visualisation de solutions d'ordre élevé. Nous avons choisi de construire notre approche en se basant sur la stratégie qui consiste à construire une représentation de bas ordre approchant la solution numérique. Afin d'avoir une « bonne » visualisation, il est nécessaire de bien contrôler cette approximation. Nous avons formalisé sous forme d'objectifs les conditions synonymes de succès. Le premier aspect concerne la nécessité de construire une représentation à l'aide de fonctions affines sur simplexes. Pour cela, une méthode de Delaunay est utilisée afin de créer un maillage de représentation composé de simplexes. Ce processus est guidé par un indicateur d'erreurs *a posteriori* qui mesure l'écart entre la solution numérique et sa représentation. Ce contrôle, qui est ponctuel, est le deuxième aspect. Enfin, un dernier point concerne la gestion des (dis)continuités. Pour cela, les éléments sont décomposés en dimensions inférieures et le processus de maillage est basé sur un traitement par dimensions croissantes. Le but est de définir un maillage unique aux interfaces (arêtes ou faces) des éléments. Dans cette optique, pour chaque dimension, un indicateur d'erreurs *a posteriori* est défini à partir de la solution numérique ou de ces traces. Un algorithme complet générant une « bonne » représentation est donné. Un algorithme d'optimisation globale déterministe, DIRECT, permet de rechercher les extrema et d'évaluer les indicateurs d'erreurs *a posteriori*. Le cadre de ce premier chapitre est néanmoins limité aux éléments droits.

Dans le chapitre suivant, nous étendons notre travail aux éléments courbes. Nous montrons les problèmes liés à l'utilisation de ce type d'éléments. Ensuite, nous modifions notre approche pour pouvoir les gérer.

Chapitre 3

Extension de la méthode de visualisation au traitement des surfaces courbes

Contents

| | | |
|------------|---|-----------|
| 3.1 | Problèmes liés à l'utilisation d'éléments courbes | 60 |
| 3.2 | Extension de la méthode aux éléments courbes | 62 |
| 3.2.1 | Identification et caractérisation des zones de retournements | 63 |
| 3.2.2 | Construction d'un estimateur <i>a posteriori</i> spécifique | 74 |
| 3.3 | Modifications de l'algorithme générant la représentation | 76 |
| 3.3.1 | Modifications dans l'algorithme général | 76 |
| 3.3.2 | Théorème de convergence pour les éléments courbes | 78 |
| 3.3.3 | Problème d'optimisation pour calculer le paramètre d'avancement maximal | 78 |
| 3.3.4 | Utilisation de DIRECT pour estimer l'indicateur spécifique | 80 |
| 3.4 | Extension au traitement des surfaces courbes dans l'espace | 87 |
| 3.5 | Conclusions du chapitre | 90 |

Dans le chapitre 2, nous avons proposé une méthodologie et un algorithme pour traiter les éléments droits. Dans ce chapitre, nous nous intéressons au traitement spécifique des surfaces courbes. Dans un premier temps, nous montrons qu'une utilisation « telle quelle » de la méthode proposée au chapitre précédent n'est pas adaptée pour des éléments courbes. Après avoir identifié les problèmes liés à l'utilisation de ces éléments, nous modifions la méthodologie pour pouvoir traiter n'importe quel type d'éléments. Les principes de la méthode proposée, à savoir la décomposition des bords en dimensions inférieures, le maillage par dimensions croissantes et la construction d'un maillage de représentation à l'aide d'un estimateur *a posteriori* sont inchangés. Néanmoins, des indicateurs d'erreurs *a posteriori* spécifiques sont construits pour pouvoir traiter tout type d'éléments. Cela a pour conséquence des modifications sur l'algorithme qui a été mis en place au chapitre précédemment. Finalement, nous montrons que cette construction permet de répondre aux objectifs \mathbf{O}_1 , \mathbf{O}_2 et \mathbf{O}_3 .

Contrairement au chapitre 2 qui propose une méthodologie capable de traiter des éléments surfaciques ou volumiques, ce chapitre se limite au cas des surfaces courbes. Cependant, même si uniquement la gestion des surfaces courbes est explicitée, le cas volumique est également évoqué.

3.1 Problèmes liés à l'utilisation d'éléments courbes

En suivant la méthodologie proposée au chapitre 2 et en particulier l'algorithme 5, un élément courbe est traité sur la figure 3.1. L'élément considéré est un triangle P^2 : ce triangle est donc construit à l'aide de 6 points et de fonctions de base d'ordre 2 (quelques exemples de fonctions de base se trouvent en annexe B). Sur ce triangle vit une fonction (figures 3.1.a et 3.1.b) que nous souhaitons représenter. Celle-ci n'est pas affine et a la particularité d'être nulle sur le bord. C'est typiquement le cas lorsque nous considérons une fonction de type Gauss-Lobatto d'ordre élevé dont les degrés de liberté associés au bord sont nuls. Un maillage de représentation de l'élément de référence est construit et une représentation sur l'élément de référence en découle (figure 3.1.c). A partir de cela, une représentation sur l'élément réel (figure 3.1.d) est déduite. Sur cette dernière représentation, nous observons des recouvrements, c'est-à-dire que l'intersection des intérieurs de certains triangles n'est pas vide. La conséquence est que la représentation générée n'est plus une fonction car plusieurs valeurs sont associées à certains points. Certaines zones de la représentation sont donc « cachées ».

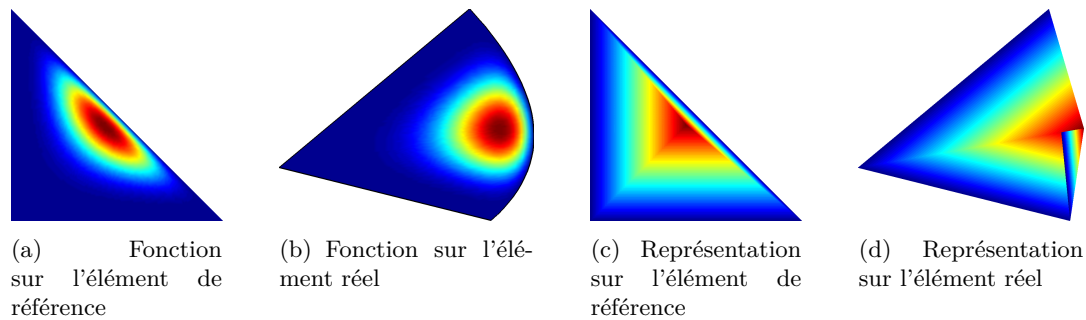


FIGURE 3.1 – Représentation (c et d) d'une fonction (a et b) sur l'élément de référence (a et c) et l'élément réel (b et d).

Sur la figure 3.2, les notations de la figure 2.2 du chapitre 2 sont reprises mais cette-fois-ci l'élément courbe présenté sur la figure 3.1 est considéré. Contrairement à la transformation géométrique g_K qui est bien bijective par définition, l'approximation \mathbb{P}^1 de la transformation ne l'est pas. Cette non-bijectivité se caractérise par l'apparition de recouvrements. Sur l'élément réel, une partie d'une arête (trait clair sur la figure 3.2) est recouverte par un des triangles.

Sur la figure 3.3, les étapes de construction qui ont permis d'obtenir les résultats de la figure 3.1 sont présentées. La solution à visualiser est affichée sur l'élément de référence (figure 3.3.a) et sur l'élément réel (figure 3.3.e).

Le processus utilisé ici, basé sur une construction par dimensions croissantes, commence par la création des maillages de représentation sur l'élément de référence des trois arêtes. Comme la fonction est nulle sur le bord de l'élément, la partie de l'erreur de visualisation due aux contributions des valeurs est nulle. La seconde partie de l'erreur de visualisation, l'erreur géométrique, est nulle sur deux des trois arêtes qui ne sont pas courbes (arêtes droites). L'erreur de visualisation pour ces deux arêtes est donc nulle et ainsi quelque soit la tolérance demandée, il n'y a pas besoin de raffiner leurs maillages 1D de représentation. La troisième arête, qui est courbe, peut avoir besoin d'être raffinée, suivant la tolérance demandée, à cause de l'approximation de la géométrie qui est introduite. Dans le cas de la figure 3.3, aucun point supplémentaire n'a été

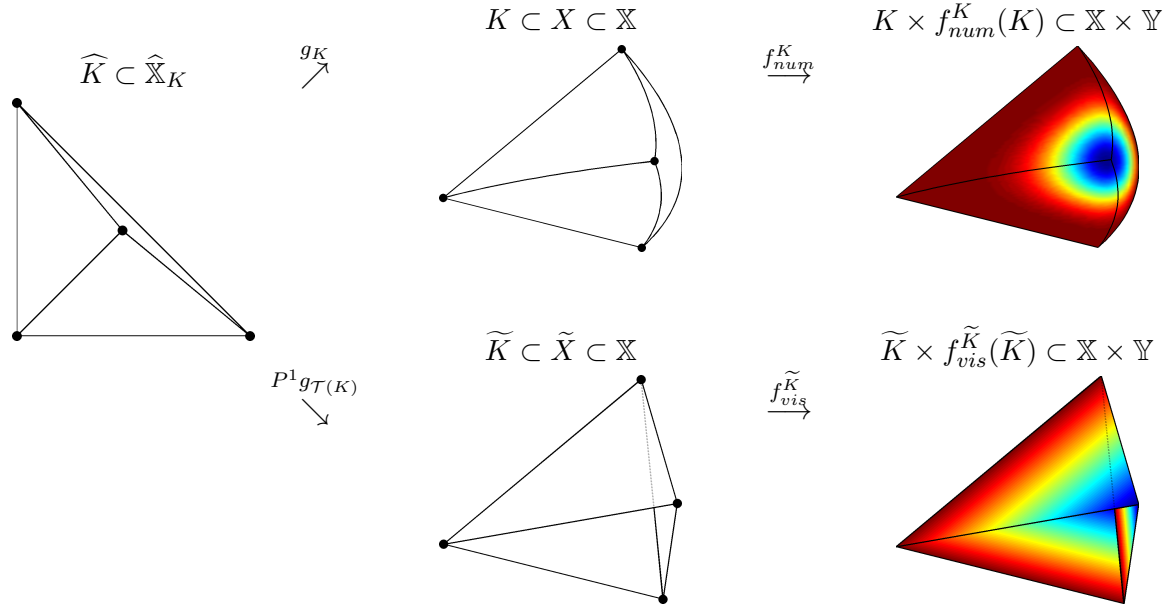


FIGURE 3.2 – Visualisation d’une fonction numérique f_{num}^K sur un élément K et sa représentation $f_{vis}^{\tilde{K}}$ construite sur un élément \tilde{K} . L’approximation \mathbb{P}^1 de la transformation géométrique n’est pas bijective de \tilde{K} vers K .

ajouté sur le bord pour représenter la fonction. Le maillage 1D est représenté en figure 3.3.b sur l’élément de référence et en figure 3.3.f sur l’élément réel.

Puis, l’algorithme passe à la dimension suivante. A partir de ces maillages des éléments de bord, le maillage de l’élément de référence est généré en figure 3.3.c. Le triangle initial est formé par les sommets de la cellule de référence. Un quatrième point est ajouté au maillage de référence, celui-ci se trouve à l’intérieur du triangle initial et est le lieu où les variations de la fonction sont les plus fortes (figure 3.3.c). Les nœuds du maillage sont transportés par la transformation géométrique g_K . Les triangles sont formés en suivant la connectivité de l’élément de référence. Le maillage de représentation sur l’élément réel en est déduit (figure 3.3.g). Cependant, sur l’élément réel, le quatrième point ne se trouve pas à l’intérieur du triangle initial. En effet, sur la figure 3.3.g, le point D se trouve en dehors du triangle ABC. A cause de ce sommet qui est à l’extérieur du triangle initial et de la connectique ainsi fixée, l’approximation de la transformation géométrique $\mathbb{P}^1 g_{\mathcal{T}(K)}$ n’est pas bijective. La conséquence est qu’un recouvrement apparaît sur l’élément réel (figure 3.3.h). Sur l’élément de référence, il n’y a pas de recouvrement (figure 3.3.d) car le maillage est bien conforme. Des zones (en orange) sont affichées sur la figure 3.3. Dans ces zones, si on ajoute un point, alors un recouvrement apparaît. Il est important de noter que cette zone ne dépend que du maillage 1D et que sa frontière sur l’élément réel est constituée du bord réel de l’élément et de l’approximation de celui-ci. La quatrième point qui a été ajouté se trouve dans cette zone (figures 3.3.c et 3.3.g), ce qui implique l’apparition d’un recouvrement.

Les zones où peuvent apparaître des recouvrements sont appelées *zones de retournements* et sont définies dans la suite du manuscrit. Nous les appelons ainsi puisque dans un cadre plus général, par exemple, pour des surfaces courbes dans l’espace (et non pas dans le plan), il peut ne

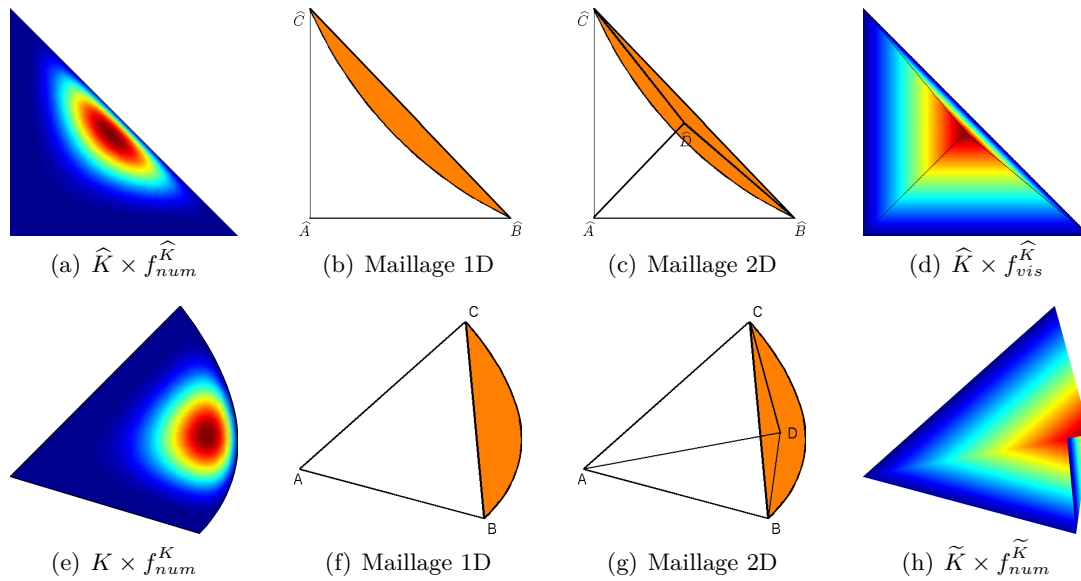


FIGURE 3.3 – Étapes de construction et mise en évidence des recouvrements lors du traitement d'un élément courbe.

pas avoir de recouvrements mais des zones faisant apparaître des *retournements* que l'on illustre dans la section 4.2.2. Comme pour la figure 3.3, ces zones de retournements ne dépendent que de l'approximation du bord (maillage 1D). Aussi, pour un maillage du bord différent, ces zones changent. En particulier, en raffinant le maillage du bord, la taille de ces zones est réduite.

En reprenant le même format, nous montrons sur la figure 3.4 ce qui se passe en ajoutant un point supplémentaire sur le bord. La fonction à représenter est identique (figures 3.4.a et 3.4.e). Les maillages des arêtes sont construits de la même manière que précédemment (figures 3.4.b et 3.4.f). La seule différence par rapport à la figure 3.3 est qu'un point supplémentaire a été ajouté. La conséquence est que les zones de retournements potentielles (en orange toujours) sont réduites par rapport à précédemment.

Avant de s'intéresser au maillage de l'élément 2D, un premier maillage est construit à partir de l'approximation du bord de l'élément. Cette fois-ci, le point qui est ajouté à l'endroit où la fonction varie beaucoup se trouve à l'intérieur de cette première tessellation sur l'élément réel, et donc à l'extérieur des zones de retournements sur l'élément réel (figure 3.4.g). La conséquence est qu'il n'y a pas de recouvrement sur l'élément réel (figure 3.4.h).

3.2 Extension de la méthode aux éléments courbes

A travers les exemples montrés dans la section 3.1, l'apparition de recouvrements et de zones de retournements ont été exhibés. Afin d'éviter d'avoir de tels problèmes, nous définissons un nouvel indicateur d'erreurs dont le principe est de raffiner les maillages de représentation des bords des éléments afin de réduire la taille de ces zones. De cette manière, aucun point n'est ajouté dans ces zones et aucun retournement n'apparaît lors du maillage en dimensions supérieures. Tout d'abord, nous commençons donc par identifier et caractériser ces zones. Ensuite, un estimateur, basé sur une recherche non pas uniquement sur le bord mais dans un certain voisinage de celui-ci, permet d'avoir un maillage suffisamment raffiné du bord.

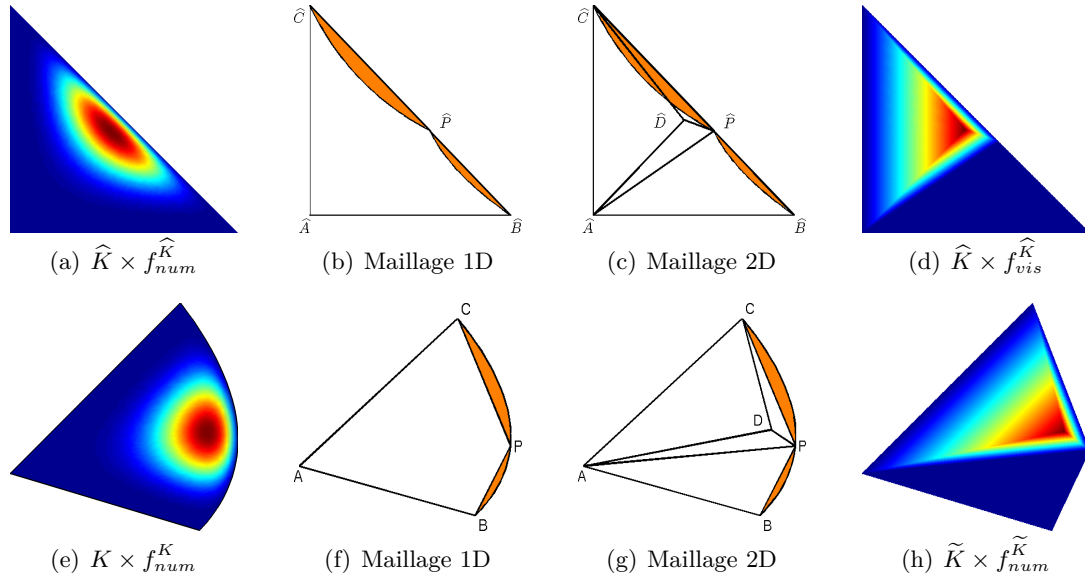


FIGURE 3.4 – Étapes de construction et *by-pass* de recouvrements par l'ajout d'un point sur le bord de l'élément.

Remarque 3.1. *Dans la suite, nous explicitons les caractérisations à partir du cas 2D plan pour lequel les définitions sont bien définies. Nous étendons les caractérisations afin de pouvoir également traiter des surfaces courbes dans l'espace.*

3.2.1 Identification et caractérisation des zones de retournements

Lorsque nous considérons des surfaces dans le plan, l'ajout d'un point $\hat{x} \in \widehat{K}$ peut amener des recouvrements. Les pertes de bijectivité sont décomposées en deux catégories :

1. **Intérieur** : le point est ajouté à l'intérieur du précédent \widetilde{K} (la dernière tessellation avant ajout de ce point).
2. **Extérieur** : le point est ajouté à l'extérieur du précédent \widetilde{K} .

Dans le premier cas, le point ajouté se trouve dans la tessellation mais pas dans le bon triangle. Ce premier cas peut souvent être résolu en utilisant des retournements d'arêtes (*swaps*). Sinon, l'ajout de points dans la tessellation à l'intérieur de l'élément suffit pour résoudre le problème. En effet, comme la transformation géométrique est bijective, son approximation tend nécessairement vers cette même fonction bijective. Ainsi, à partir du moment où le maillage de représentation est suffisamment raffiné, les pertes de bijectivité intérieures se règlent facilement.

La gestion du second cas est beaucoup plus délicate. Lorsque ce cas arrive, la véritable frontière de \widetilde{K} n'est plus $P^1 g_{\mathcal{T}(K)}(\partial \widehat{K})$ (voir figure 3.5). Tant que le point x choisi se trouve à l'extérieur de la tessellation, c'est-à-dire, en dehors du domaine délimité par $P^1 g_{\mathcal{T}(K)}(\partial \widehat{K})$, peu importe les *swaps* qui pourraient être effectués, $\partial \widetilde{K}$ sera différent de $P^1 g_{\mathcal{T}(K)}(\partial \widehat{K})$ et le problème reste. Ainsi, la seule solution est d'ajouter des points sur $\partial \widehat{K}$ pour assurer $\partial \widetilde{K} = P^1 g_{\mathcal{T}(K)}(\partial \widehat{K})$. Si cette opération est faite sans se soucier des éléments voisins qui partagent cette partie de

$g_K(\widehat{\partial K})$, alors des maillages non-coïncidents peuvent être construits et ainsi l'objectif \mathbf{O}_3 n'est plus vérifié. Pour contourner ce problème, il faudrait mettre à jour les maillages de frontière de tous les éléments adjacents au risque d'un coût très élevé et d'un algorithme potentiellement non convergent.

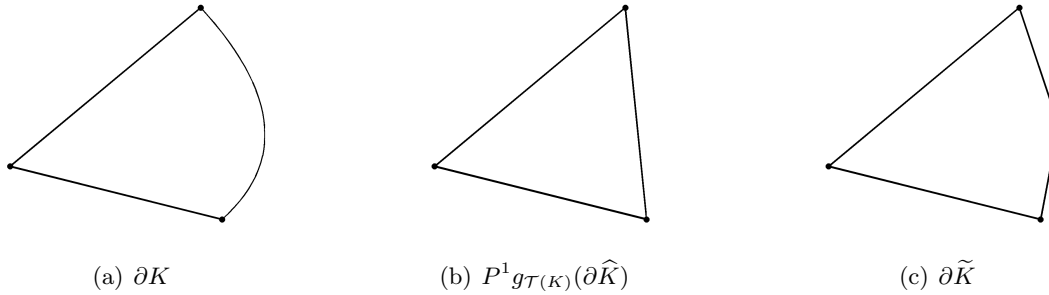


FIGURE 3.5 – Comparaison de ∂K (frontière de l'élément réel), de $P^1 g_{\mathcal{T}(K)}(\widehat{\partial K})$ (image par l'approximation de la transformation géométrique de la frontière de la cellule de référence) et $\partial \widetilde{K}$ (frontière de l'approximation de l'élément réel) de la figure 3.2.

Notre stratégie est de construire un maillage du bord suffisamment raffiné pour que les pertes de bijectivité extérieures n'apparaissent jamais. Pour cela, nous allons identifier *a priori* les zones de retournements pour assurer la préservation du maillage du bord de l'élément (i.e. $\widehat{\partial K} = P^1 g_{\mathcal{T}(K)}(\widehat{\partial K})$) durant tout le processus. Ensuite, nous prouverons que l'objectif \mathbf{O}_3 reste vérifié. Finalement, les problèmes liés à ces insertions de points peuvent être évités et ainsi $P^1 g_{\mathcal{T}(K)}$ reste injective.

Nous avons vu dans la section 3.1 que les zones de retournements se caractérisent comme l'ensemble des points à l'extérieur de la tessellation courante. cela se traduit de la façon suivante. Soit $K \in \mathcal{T}_h$ et $\mathcal{T}(K)$ un maillage associé à cet élément défini tel que $P^1 g_{\mathcal{T}(K)}$ est une fonction injective de \widehat{K} vers $\widetilde{K} = \bigcup_{S \in \mathcal{T}(K)} S$. Nous définissons les zones de retournements comme

$$\mathcal{R}_{\mathcal{T}(\partial K)}^K = K \setminus \text{Int}(\widetilde{K}). \quad (3.1)$$

L'indice $\mathcal{T}(\partial K)$ indique que ces zones ne dépendent que du maillage du bord. Ces zones peuvent également être définies sur l'élément de référence comme

$$\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K = g_K^{-1} \left(K \setminus \text{Int}(\widetilde{K}) \right).$$

La proposition 3.2 fait le lien entre les zones de retournements et la non injectivité de l'approximation \mathbb{P}^1 de la transformation géométrique.

Proposition 3.2. *Sous les hypothèses \mathbf{H}_1 à \mathbf{H}_6 , on considère un élément $K \in \mathcal{T}(X)$. Soit $\mathcal{T}(K)$ un maillage de K défini tel que $P^1 g_{\mathcal{T}(K)}$ soit injective de \widehat{K} vers \widetilde{K} . Soit $\hat{x} \in \widehat{K}$ un point arbitraire et on note $\mathcal{T}_{\hat{x}}(\widehat{K})$ le maillage raffiné par l'ajout du nouveau sommet \hat{x} dans $\mathcal{T}(\widehat{K})$.*

Si $x = g_K(\hat{x})$ est un sommet extérieur (i.e. $x \in \mathcal{R}_{\mathcal{T}(\partial K)}^K$), alors $P^1 g_{\mathcal{T}_{\hat{x}}(K)}$, la fonction associée à $\mathcal{T}_{\hat{x}}(K)$, n'est pas injective.

Preuve. Soit $\hat{x} \in \widehat{S}$ un sommet extérieur (sommet en rouge sur la figure 3.6), qui vérifie donc $\hat{x} \in \widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$ où $\widehat{S} \in \mathcal{T}(\widehat{K})$. En suivant la définition de $\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$ donnée par (3.1), $x = g_K(\hat{x}) \notin \text{Int}(\widetilde{K})$, donc il y a deux cas :

- Cas 1 : $x \in \partial\widetilde{K}$ (figure 3.6.e).
- Cas 2 : $x \notin P^1g_{\mathcal{T}(K)}(\widehat{S})$ (figure 3.6.f).

Dans le premier cas, comme $x \in \partial\widetilde{K}$, deux éléments sont plats sur l'élément réel (voir figure 3.6.e) et donc $P^1g_{\mathcal{T}_x(K)}$ n'est pas injective, même si ces éléments ne sont pas plats sur la cellule de référence (voir figure 3.6.b).

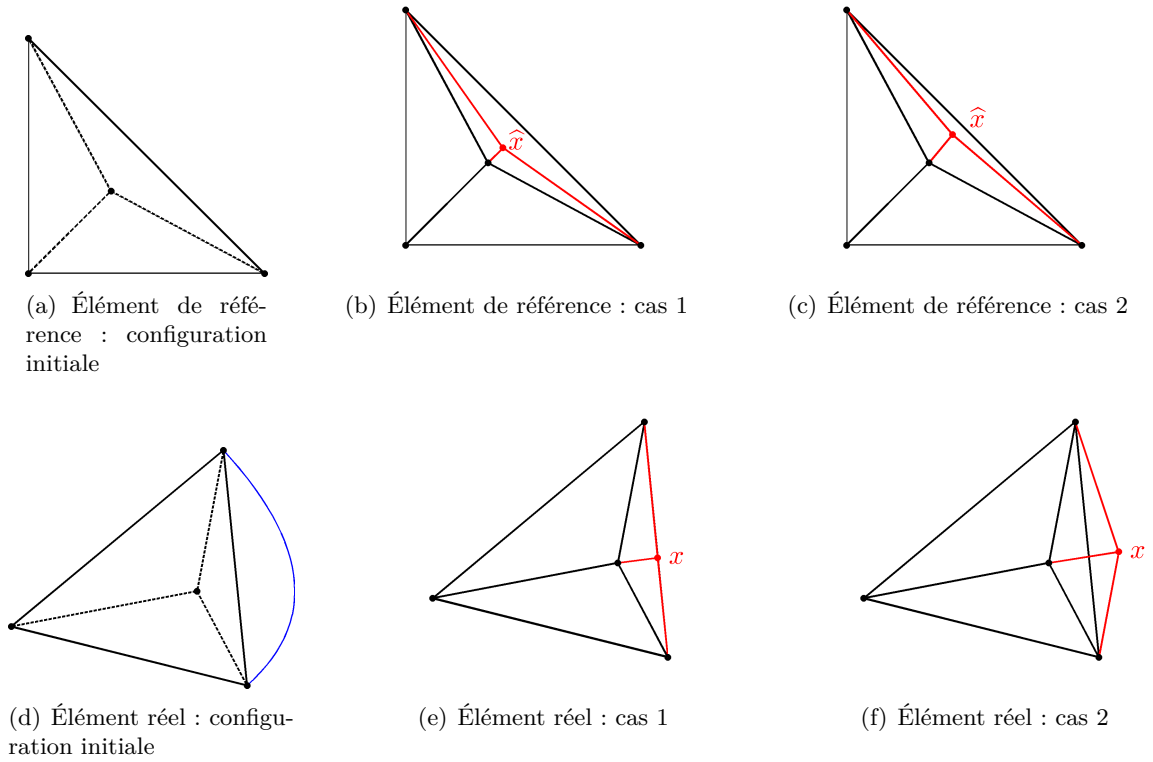


FIGURE 3.6 – Ajout d'un sommet extérieur (en rouge) dans une configuration initiale (a et d) pour un élément courbe (frontière réelle en bleu).

Dans le second cas, trois configurations données dans [Walfisch, 2007] sont possibles (voir figure 3.7) :

- Configuration 1 : \hat{x} appartient à l'intérieur d'un simplexe \widehat{S} qui est unique par \mathbf{H}_6 ,
- Configuration 2 : \hat{x} appartient à l'arête commune de deux simplexes,
- Configuration 3 : \hat{x} appartient à la frontière de \widehat{K} .

Suivant la configuration, le nombre de simplexes créés varie. Nous considérons la première configuration, la preuve étant similaire pour les deux autres. Nous notons \widehat{S}_1 , \widehat{S}_2 et \widehat{S}_3 les simplexes créés à partir de \hat{x} et des trois sommets \hat{a} , \hat{b} et \hat{c} de \widehat{S} et S_1 , S_2 et S_3 leurs images par $P^1g_{\mathcal{T}_x(K)}$. Ces notations sont illustrées sur la figure 3.8. Prouvons qu'il existe i et k , avec $i \neq k$, tel que $Int(S_i) \cap Int(S_k) \neq \emptyset$, ce qui signifie qu'il y a un recouvrement.

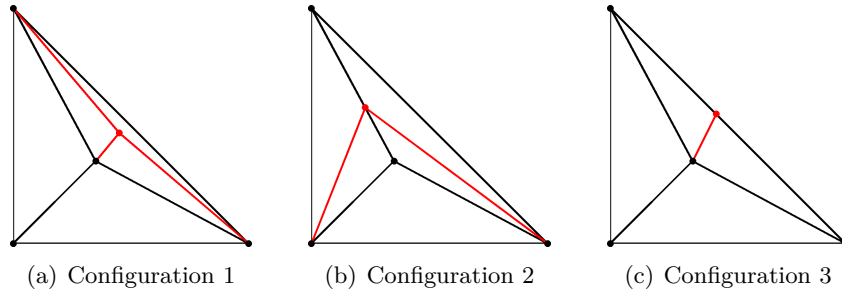


FIGURE 3.7 – Configurations possibles lors de l'ajout d'un nouveau sommet au maillage.

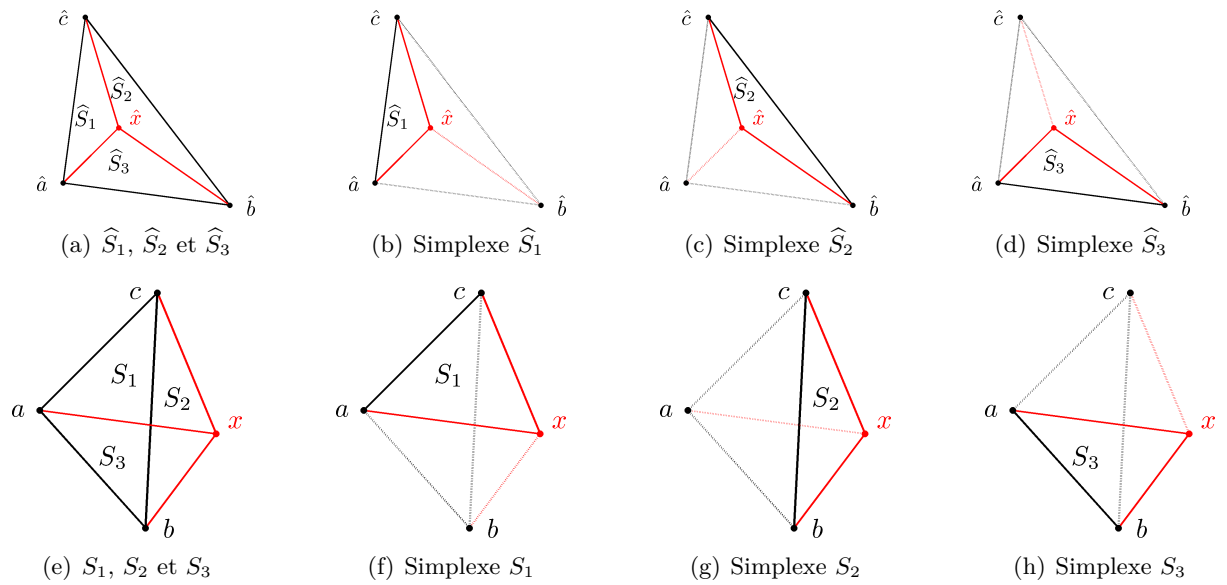


FIGURE 3.8 – Notations utilisées lors du découpage d'un simplexe en 3 simplexes.

Nous le montrons par contradiction. Nous supposons que $Int(S_i) \cap Int(S_k) = \emptyset$ pour chaque $i \neq k$. Dans ce cas, $\bigcup_{i=1}^3 S_i = S$ et $x = g_K(\hat{x}) \in S$, ce qui est absurde. Ainsi, il existe $y \in Int(S_i) \cap Int(S_k)$, $i \neq k$, tel que $y = P^1 g_{\mathcal{T}_{\hat{x}}(K)}(\hat{y}_i) = P^1 g_{\mathcal{T}_{\hat{x}}(K)}(\hat{y}_k)$ avec $\hat{y}_i \in \hat{S}_i$ et $\hat{y}_k \in \hat{S}_k$.

Finalement, $P^1 g_{\mathcal{T}_{\hat{x}}(K)}$ n'est pas injective car il y a un recouvrement. \square

Le problème de la définition des zones de retournements donnée par (3.1) est qu'elle est implicite et est donc inutilisable en pratique. Au lieu d'utiliser (3.1), nous souhaitons définir une zone qui englobe $\mathcal{R}_{\mathcal{T}(\partial K)}^K$ mais qui soit définie à partir de la frontière de la cellule. Ainsi, à chaque arête $\hat{S} \subset \hat{K}$, nous associons

$$\hat{\mathcal{V}}_S^K = \left\{ \hat{x} + t\hat{n}_x, \hat{x} \in \hat{S}, 0 \leq t \leq t_{\hat{x}} \right\}, \quad (3.2)$$

où \hat{n}_x est un vecteur unitaire défini à partir d'un point \hat{x} de l'arête pointant vers l'intérieur de l'élément \hat{K} et $t_{\hat{x}}$ est un paramètre d'avancement maximal d'investigation en partant du point \hat{x} . L'idée est de suivre le vecteur \hat{n}_x jusqu'à la dernière rencontre avec

$$\hat{S}' = g_K^{-1}(P^1 g_{\mathcal{T}(K)}(\hat{S}) \cap K),$$

qui est une partie du bord de $\hat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$. Les notations sont reportées sur la figure 3.9. L'image de \hat{S}' par g_K est S' et correspond à l'approximation de S qui est un segment. Le but de cette caractérisation est de parcourir la zone de retournements d'un bord \hat{S} à un autre bord \hat{S}' sur l'élément de référence (figure 3.9.a). Sur l'élément réel, cela se traduit par le passage du bord réel S à son approximation S' (figure 3.9.b).

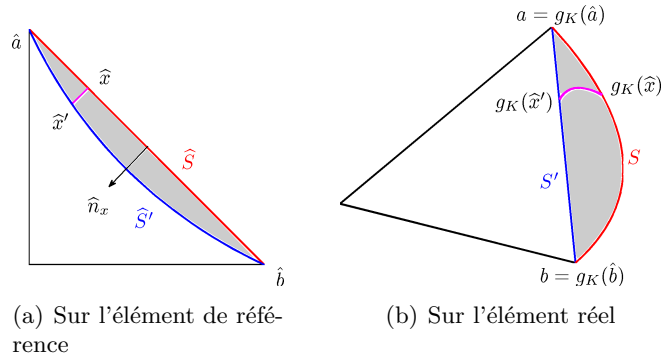


FIGURE 3.9 – Identification des zones de retournements sur l'élément de référence (gauche) et l'élément réel (droite).

Les définitions dans le cas général sont données dans la suite. Le principal avantage de ces zones est qu'elles sont constructibles. Il faut noter que sur la cellule de référence, la paramétrisation dans l'équation (3.2) est effectuée à l'aide d'un segment (en magenta sur la figure 3.9) entre un point de la frontière $\hat{x} \in \hat{S}$ et un point $\hat{x}' = \hat{x} + t_{\hat{x}}\hat{n}_x \in \hat{S}'$ afin de caractériser la zone. Remarquons cependant que comme la transformation géométrique g_K peut être non-linéaire, l'image de ce segment peut être différente d'un segment (en magenta sur la figure 3.9).

Dans le cas général, la définition 3.3 introduit les quantités qui caractérisent les zones de retournements. En particulier, nous donnons une définition rigoureuse du paramètre d'avancée maximale.

Définition 3.3. Soit $K \in \mathcal{T}_{2D}$, $\mathcal{T}(K)$ un maillage et $\widehat{S} = [\widehat{a}\widehat{b}] \in \mathcal{T}(\partial\widehat{K})$ un segment de la frontière. Nous notons $\widehat{S}' = g_K^{-1}(P^1g_{\mathcal{T}(K)}(\widehat{S}) \cap K)$ et \widehat{n} la normale unitaire de \widehat{S} pointant vers l'intérieur de \widehat{K} . Nous définissons les quantités suivantes dont la signification est expliquée ci-après.

1. Nous notons \widehat{n}_a et \widehat{n}_b , les vecteurs associés aux sommets \widehat{a} et \widehat{b} ,

$$\widehat{n}_a = \underset{\widehat{u} \in \left\{ \widehat{n}, \frac{\widehat{a}\widehat{y}}{\|\widehat{a}\widehat{y}\|} \text{ où } \widehat{y} \in \widehat{S}' \setminus \{\widehat{a}\} \right\}}{\arg \min} \left(\frac{\widehat{u} \cdot \widehat{a}\widehat{b}}{\|\widehat{a}\widehat{b}\|} \right) \quad \text{et} \quad \widehat{n}_b = \underset{\widehat{u} \in \left\{ \widehat{n}, \frac{\widehat{b}\widehat{y}}{\|\widehat{b}\widehat{y}\|} \text{ où } \widehat{y} \in \widehat{S}' \setminus \{\widehat{b}\} \right\}}{\arg \min} \left(\frac{\widehat{u} \cdot \widehat{b}\widehat{a}}{\|\widehat{b}\widehat{a}\|} \right). \quad (3.3)$$

2. Soit $\widehat{x} = (1 - \omega)\widehat{a} + \omega\widehat{b} \in \widehat{S}$ et $\widehat{n}_x = (1 - \omega)\widehat{n}_a + \omega\widehat{n}_b$, où $\omega \in [0, 1]$. Soit $\widehat{y}^* \in \widehat{S} \setminus g_K^{-1}(\widehat{K})$ un point arbitraire, nous introduisons

$$\begin{cases} t_{\widehat{x}} = \max \left\{ t, \widehat{x} + t\widehat{n}_x \in \partial\widehat{K} \right\}, & \text{si } (ab \wedge ag_K(\widehat{y}^*)) \cdot (ab \wedge ag_K(\widehat{x} + t_{\widehat{x}}\widehat{n}_x)) > 0, \\ t_{\widehat{x}} = \max \left\{ \left\{ t, \widehat{x} + t\widehat{n}_x \in \widehat{K}, ab \wedge ag_K(\widehat{x} + t_{\widehat{x}}\widehat{n}_x) = 0 \right\} \cup \{0\} \right\}, & \text{sinon.} \end{cases} \quad (3.4)$$

3. Finalement, avec ces notations, nous posons

$$\widehat{\mathcal{V}}_S^K = \left\{ \widehat{x} + t\widehat{n}_x, \widehat{x} = (1 - \omega)\widehat{a} + \omega\widehat{b} \in \widehat{S}, \widehat{n}_x = (1 - \omega)\widehat{n}_a + \omega\widehat{n}_b, \omega \in [0, 1], 0 \leq t \leq t_{\widehat{x}} \right\}. \quad (3.5)$$

Les vecteurs \widehat{n}_a et \widehat{n}_b sont définis comme des vecteurs réalisant le minimum de l'équation (3.3). Dans ces notations, \widehat{u} , \widehat{n} , $\widehat{a}\widehat{y}$, $\widehat{a}\widehat{b}$, $\widehat{b}\widehat{a}$ désignent des vecteurs et \cdot le produit scalaire usuel. Le vecteur \widehat{n}_a est construit en cherchant le minimum du produit scalaire entre les vecteurs \widehat{u} et $\widehat{a}\widehat{b}$ où \widehat{u} est l'ensemble des éléments \widehat{n} et $\frac{\widehat{a}\widehat{y}}{\|\widehat{a}\widehat{y}\|}$ avec $\widehat{y} \in \widehat{S}' \setminus \{\widehat{a}\}$. Dans cette recherche, $\widehat{y} \neq \widehat{a}$, sinon le vecteur \widehat{n}_a pourrait être le vecteur nul. Géométriquement, le vecteur \widehat{n}_a est choisi pour maximiser l'angle entre \widehat{u} et $\widehat{a}\widehat{b}$ car

$$\frac{\widehat{a}\widehat{y}}{\|\widehat{a}\widehat{y}\|} \cdot \frac{\widehat{a}\widehat{b}}{\|\widehat{a}\widehat{b}\|} = \cos \left(\widehat{y}\widehat{a}\widehat{b} \right).$$

Le vecteur \widehat{n}_b est défini de façon similaire. Les vecteurs \widehat{n}_a et \widehat{n}_b peuvent être égaux à la normale \widehat{n} mais ce n'est pas toujours le cas puisque qu'ils peuvent être construits également à partir d'un point de \widehat{S}' , qui est illustré en pointillé vert sur la figure 3.10. En se restreignant uniquement à la normale, il pourrait y avoir des cas où une partie de la zone n'est pas prise en compte (cas 1 de la figure 3.10 par exemple).

De manière classique \wedge désigne le produit vectoriel. Remarquons que dans le cas où l'espace est de dimension 3, le produit vectoriel entre deux vecteurs est en vecteur. Le produit vectoriel entre deux vecteurs de dimension 2 donne un scalaire qui est obtenu en considérant les vecteurs comme s'ils étaient de dimension 3 et en prenant la troisième composante du résultat (les autres étant nulles).

Le point \widehat{x} est repéré à l'aide d'une coordonnée barycentrique ω entre \widehat{a} et \widehat{b} . Le vecteur \widehat{n}_x est construit comme une combinaison linéaire en ω afin de balayer tout l'élément comme le montre la figure 3.11.

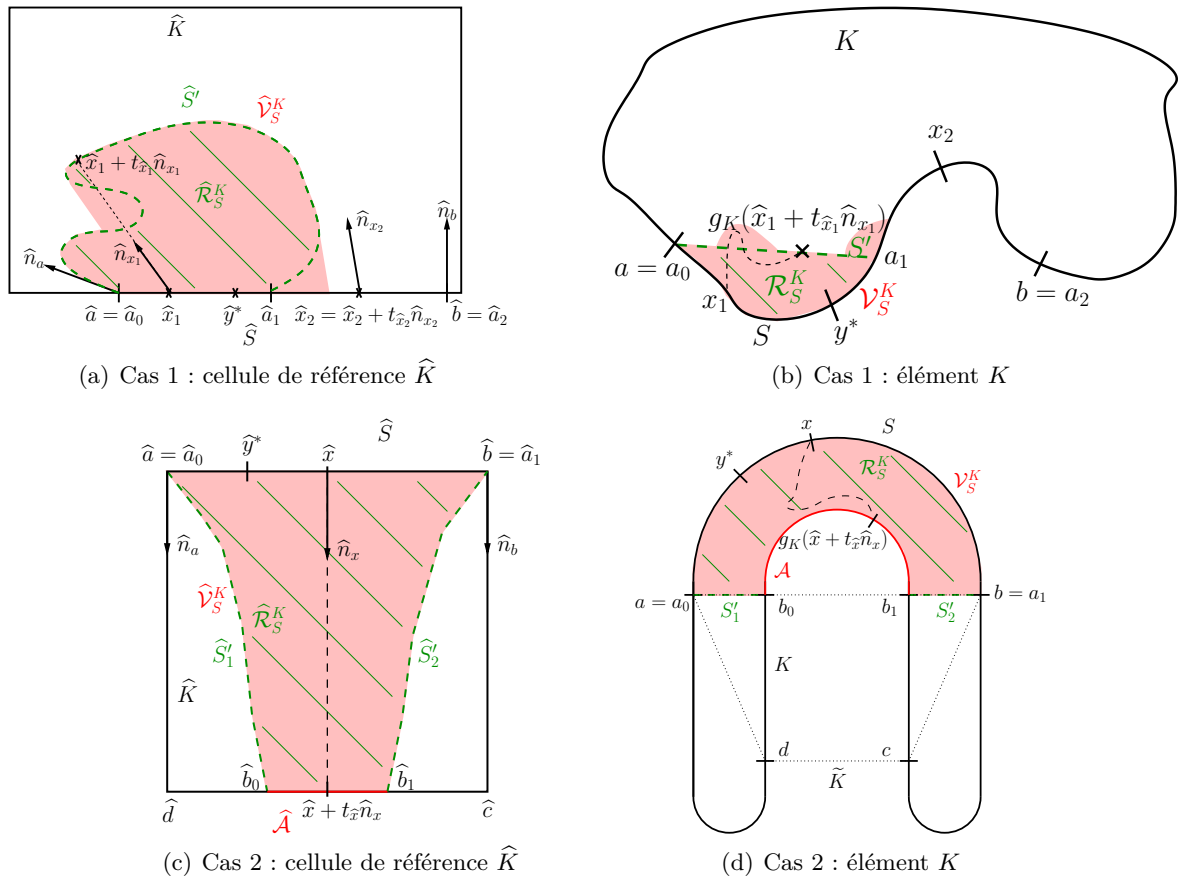


FIGURE 3.10 – Illustration des zones de retournements et des notations utilisées sur deux cas.

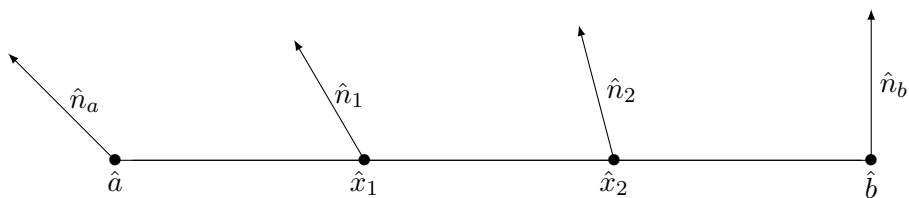


FIGURE 3.11 – Les vecteurs \hat{n}_x sont construits comme une combinaison affine entre \hat{n}_a et \hat{n}_b .

Notons que le premier cas de $t_{\hat{x}}$ est bien défini car \widehat{K} est un ensemble convexe et donc il y a au plus deux points sur sa frontière $\partial\widehat{K}$ qui intersectent la ligne partant de \hat{x} avec la direction \hat{n}_x (ces deux points peuvent être égaux, ce qui correspond à $t = 0$, ce qui arrive par exemple lorsque \hat{x} est un sommet d'un triangle \widehat{K}). Ce premier cas est illustré sur les figures 3.10.c et 3.10.d. L'ensemble $\widehat{S} \setminus g_K^{-1}(\widehat{K})$ utilisé pour choisir \hat{y}^* peut être vide. Néanmoins, cela signifie que dans ce cas $\widehat{\mathcal{V}}_S^K = \widehat{S}$, et donc naturellement $t_{\hat{x}} = 0$ puisqu'il n'y a pas d'avancée à l'intérieur de l'élément.

Le second cas de $t_{\hat{x}}$ cherche le vecteur $ag_K(\hat{x} + t_{\hat{x}}\hat{n}_x)$ colinéaire à ab en regardant quand le produit vectoriel est nul et est illustré sur les figures 3.10.a et 3.10.b. Ceci permet de trouver un point appartenant à l'approximation de l'arête.

Remarque 3.4. *Tous les vecteurs faisant apparaître n dans leur nom (\hat{n} , \hat{n}_x , n' , etc...) sont unitaires. En particulier, les normales sont choisies unitaires.*

La définition des zones données par l'équation (3.5) est utile car cela permet d'inclure les zones de retournements comme le montre le lemme suivant.

Lemme 3.5. *Soit $K \in \mathcal{T}_{2D}$, $\mathcal{T}(K)$ un maillage de K , $\mathcal{R}_{\mathcal{T}(\partial K)}^K$ et $\mathcal{V}_S^K = g_K(\widehat{\mathcal{V}}_S^K)$ donnés respectivement par (3.1) et (3.5), alors*

$$\mathcal{R}_{\mathcal{T}(\partial K)}^K \subset \bigcup_{S \in \mathcal{T}(\partial K)} \mathcal{V}_S^K.$$

Preuve. Pour prouver le résultat de ce lemme, la preuve est décomposée en 3 étapes.

- Tout d'abord, nous introduisons pour chaque arête S , un ensemble \mathcal{R}_S^K , qui est utilisé pour identifier les composantes connexes de $\mathcal{R}_{\mathcal{T}(\partial K)}^K$ et suit une décomposition à partir des arêtes.
- Ensuite, nous prouvons que ces ensembles \mathcal{R}_S^K vérifient $\mathcal{R}_{\mathcal{T}(\partial K)}^K \subset \bigcup_{S \in \mathcal{T}(\partial K)} \mathcal{R}_S^K$.
- Enfin, pour démontrer le résultat, nous montrons que pour chaque arête S , $\mathcal{R}_S^K \subset \mathcal{V}_S^K$.

Étape 1. Soit $\widehat{S} = [\hat{a}\hat{b}] \in \mathcal{T}(\partial\widehat{K})$ un segment appartenant à la frontière de l'élément et $S = g_K(\widehat{S})$. Nous définissons les points $\{a_i\}_{i=0,N}$ comme les extrémités des composantes connexes de $g_K(\widehat{S}) \cap P^1 g_{\mathcal{T}(K)}(\widehat{S})$. Le nombre de ces points est donc limité et ceux-ci vérifient

$$a_0 = a, a_N = b, \exists \omega_i \in [0, 1], a_i = (1 - \omega_i)a + \omega_i b = g_K(\hat{a}_i)$$

où les réels ω_i sont classés par ordre croissant (voir figure 3.10). Sur l'élément réel, ces points $\{a_i\}_{i=0,N}$ ont la particularité d'appartenir à la fois au bord S mais aussi à son approximation S' .

Maintenant, nous définissons

$$\widehat{\Sigma}_S^K = \bigcup_{i=0}^{N-1} \widehat{\Sigma}_{S,i}^K,$$

où

$$\widehat{\Sigma}_{S,i}^K = \begin{cases} g_K^{-1}([a_i a_{i+1}]) \cup [\hat{a}_i \hat{a}_{i+1}], & \text{si } [a_i a_{i+1}] \subset K, \\ g_K^{-1}([a_i a_{i+1}] \cap K) \cup \widehat{\mathcal{A}} \cup [\hat{a}_i \hat{a}_{i+1}], & \text{sinon,} \end{cases} \quad (3.6)$$

où $\widehat{\mathcal{A}}$ est un arc ou une union d'arcs tel que $\widehat{\mathcal{A}} \subset \partial\widehat{K}$ (voir figure 3.10). Chaque arc de $\widehat{\mathcal{A}}$ est composé d'arcs ou de segments $[\hat{b}_j \hat{b}_{j+1}]$ tel que $b_j = g_K(\hat{b}_j) \in (\partial K \setminus S) \cap P^1 g_{\mathcal{T}(K)}(\widehat{S})$, où les

points $\{\hat{b}_j\}$ sont classés par ordre croissant (de la même manière que pour les réels $\{\hat{a}_i\}$). Ainsi, chaque $\widehat{\Sigma}_{S,i}^K$ est fermé.

Explicitons ces courbes pour les deux exemples de la figure 3.10.

Pour le premier cas, il y a 3 points $\{a_i\}_{i=0,2}$ et donc deux courbes : $\widehat{\Sigma}_{S,0}^K$ et $\widehat{\Sigma}_{S,1}^K$. Comme le segment $[a_0a_1] \subset K$, nous en déduisons que

$$\widehat{\Sigma}_{S,0}^K = \widehat{S}' \cup [\hat{a}_0\hat{a}_1].$$

Par contre, le segment $[a_1a_2] \not\subset K$, on en déduit, d'après (3.6), que

$$\widehat{\Sigma}_{S,1}^K = [\hat{a}_1\hat{a}_2].$$

Pour le cas 2, il n'y a que deux points a_0 et a_1 , définis tels que $[a_0a_1] \not\subset K$. Or, $[a_0a_1] \cap K = S'_1 \cup S'_2$, nous en déduisons que $g_K^{-1}([a_0a_1] \cap K) = \widehat{S}'_1 \cup \widehat{S}'_2$. De plus, un arc $\widehat{\mathcal{A}}$ est défini afin de fermer $\widehat{\Sigma}_{S,0}^K$ tel que

$$\widehat{\Sigma}_{S,0}^K = \widehat{S}'_1 \cup \widehat{S}'_2 \cup \widehat{\mathcal{A}} \cup [\hat{a}_0\hat{a}_1].$$

Pour un ensemble $C \subset X$, nous notons maintenant $\mathcal{B}(C)$ les composantes connexes bornées du complémentaire de C dans X . Nous introduisons

$$\widehat{\mathcal{R}}_S^K = \left(\mathcal{B}(\widehat{\Sigma}_S^K) \cup \widehat{\Sigma}_S^K \right) \setminus g_K^{-1} \left(\text{Int}(\widetilde{K}) \right).$$

Ces zones $\widehat{\mathcal{R}}_S^K$ sont affichées en vert sur la figure 3.10.

Étape 2. Nous voulons montrer que

$$\mathcal{R}_{\mathcal{T}(\partial K)}^K \subset \bigcup_{S \in \mathcal{T}(\partial K)} \mathcal{R}_S^K.$$

Or, nous remarquons que la frontière des zones de retournements $\partial \mathcal{R}_{\mathcal{T}(\partial K)}^K = \partial \left(K \setminus \text{Int}(\widetilde{K}) \right)$ peut être décomposée en

$$\begin{aligned} \partial \mathcal{R}_{\mathcal{T}(\partial K)}^K &= \partial \left(K \cap \text{Int}(\widetilde{K})^c \right) \\ &= \left(\partial K \cap \text{Int}(\widetilde{K})^c \right) \cup \left(K \cap \partial \widetilde{K} \right) \\ &= \bigcup_{\widehat{S} \in \mathcal{T}(\partial \widehat{K})} \left(\left(g_K(\widehat{S}) \setminus \text{Int}(\widetilde{K}) \right) \cup \left(P^1 g_{\mathcal{T}(K)}(\widehat{S}) \cap K \right) \right). \end{aligned} \quad (3.7)$$

Alors, par construction de chaque $\Sigma_S^K = g_K(\widehat{\Sigma}_S^K)$ et l'équation (3.7),

$$\partial \mathcal{R}_{\mathcal{T}(\partial K)}^K \subset \bigcup_{S \in \mathcal{T}(\partial K)} \Sigma_S^K.$$

Prouvons maintenant que

$$\text{Int}(\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K) \subset \bigcup_{S \in \mathcal{T}(\partial K)} \mathcal{R}_S^K.$$

Nous décomposons $\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$ en $\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K = \bigcup_{\widehat{C}} \widehat{C}$ où \widehat{C} sont les composantes connexes de $\widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$. Considérons maintenant une composante connexe \widehat{C} telle que $\text{Int}(\widehat{C}) \neq \emptyset$ (sinon $\widehat{C} \subset \partial \widetilde{K}$ et le résultat est immédiat). Alors, $\exists \widehat{S}_i = [\hat{a}_i\hat{a}_{i+1}] \subset \widehat{S} \subset \partial \widetilde{K}$, $\text{Int}_{1D}(\widehat{S}_i) \cap \widehat{C} \neq \emptyset$, où $\text{Int}_{1D}(\widehat{S}_i)$ désigne l'intérieur 1D du segment \widehat{S}_i , et tel que $g_K(\widehat{S}_i) \neq P^1 g_{\mathcal{T}(K)}(\widehat{S}_i)$. Trois cas peuvent *a priori* apparaître.

- Si $\widehat{C} \subset \mathcal{B}(\widehat{\Sigma}_{S,i}^K) \cup \widehat{\Sigma}_{S,i}^K$, alors $\widehat{C} \subset \bigcup_{S \in \mathcal{T}(\partial K)} \widehat{\mathcal{R}}_S^K$.
 - Le deuxième cas serait que $\text{Int}(\widehat{C} \cap (\mathcal{B}(\widehat{\Sigma}_{S,i}^K) \cup \widehat{\Sigma}_{S,i}^K)) = \emptyset$.
 Dans ce cas, on aurait, $\widehat{C} \cap \text{Int}(\widehat{K}) = \emptyset$ ou $\widehat{\Sigma}_{S,i}^K \cap \text{Int}(\widehat{K}) = \emptyset$. Cependant, comme $\text{Int}(\widehat{C}) \neq \emptyset$, $\widehat{C} \cap \text{Int}(\widehat{K}) = \emptyset$ est impossible et comme $g_K(\widehat{S}_i) \neq P^1 g_{\mathcal{T}(K)}(\widehat{S}_i)$, $\widehat{\Sigma}_{S,i}^K \cap \text{Int}(\widehat{K}) = \emptyset$ est également impossible.
 Donc, $\text{Int}(\widehat{C} \cap (\mathcal{B}(\widehat{\Sigma}_{S,i}^K) \cup \widehat{\Sigma}_{S,i}^K)) \neq \emptyset$.
 - Le troisième cas serait que $\text{Int}(\widehat{C} \cap (\mathcal{B}(\widehat{\Sigma}_{S,i}^K) \cup \widehat{\Sigma}_{S,i}^K)) \neq \emptyset$ mais tel que $\widehat{C} \not\subset \mathcal{B}(\widehat{\Sigma}_{S,i}^K)$.
 Dans ce cas, il existerait une composante connexe $\widehat{C}' \subset (\widehat{C} \cap \mathcal{B}(\widehat{\Sigma}_{S,i}^K))$ telle que sa frontière $\partial \widehat{C}'$ vérifie $\partial \widehat{C}' \subset \partial \widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$. Donc, \widehat{C}' serait une composante connexe de $\widehat{\mathcal{R}}_{\partial K}^K$ strictement incluse dans \widehat{C} : ceci est impossible.
- Il vient que n'importe quelle composante connexe $\widehat{C} \subset \widehat{\mathcal{R}}_{\mathcal{T}(\partial K)}^K$ satisfait

$$\widehat{C} \subset \bigcup_{S \in \mathcal{T}(\partial K)} \widehat{\mathcal{R}}_S^K$$

et donc

$$\mathcal{R}_{\mathcal{T}(\partial K)}^K \subset \bigcup_{S \in \mathcal{T}(\partial K)} \mathcal{R}_S^K.$$

Étape 3. La dernière étape de cette preuve consiste à montrer que pour chaque $S \subset \partial K$,

$$\mathcal{R}_S^K \subset \mathcal{V}_S^K.$$

Considérons un point $\hat{z} \in \widehat{\mathcal{R}}_S^K$, pour prouver que $\hat{z} \in \widehat{\mathcal{V}}_S^K$, nous montrons d'abord que \hat{z} peut s'écrire comme $\hat{z} = \hat{x} + t\hat{n}_x$ où $t \geq 0$, $\hat{x} \in \widehat{S} = [\hat{a}\hat{b}]$. Ensuite, nous montrons que $t \leq t_{\hat{x}}$. Pour cela, nous introduisons l'application

$$\varphi(\omega, t) \mapsto (1 - \omega)(\hat{a} + t\hat{n}_a) + \omega(\hat{b} + t\hat{n}_b)$$

et l'ensemble

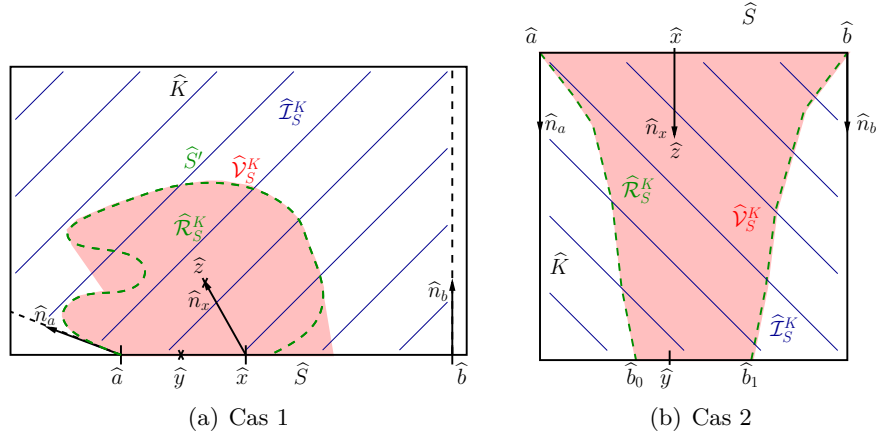
$$\widehat{\mathcal{I}}_S^K = \varphi([0, 1] \times \mathbb{R}^+) \cap \widehat{K}$$

qui est un ensemble convexe. Ces ensembles $\widehat{\mathcal{I}}_S^K$ sont illustrés sur la figure 3.12 pour les deux cas et sont construits en parcourant l'élément en partant d'un point \hat{x} et en suivant un vecteur \hat{n}_x .

Prouvons que $\hat{z} \in \widehat{\mathcal{I}}_S^K$. Il est évident que $t \geq 0$, sinon pour n'importe quel point $\hat{x} \in \widehat{S}$, $\hat{x} + t\hat{n}_x \notin \widehat{K}$ comme \hat{n}_x pointe vers l'intérieur de l'élément.

Prouvons maintenant que $\hat{n}_a = \arg \min_{\hat{u} \in \{\hat{n}, \frac{\hat{a}\hat{y}}{\|\hat{a}\hat{y}\|} \text{ où } \hat{y} \in \partial \widehat{\mathcal{R}}_S^K\}} \left(\frac{\hat{u} \cdot \hat{a}\hat{b}}{\|\hat{a}\hat{b}\|} \right)$. Nous considérons $\hat{y} \in \partial \widehat{\mathcal{R}}_S^K$, alors il y a trois possibilités, $\hat{y} \in \widehat{S}$, $\hat{y} \in \widehat{\mathcal{A}}$ ou $\hat{y} \in \widehat{S}'$ (voir définition (3.6) de $\widehat{\Sigma}_{S,i}^K$).

- Si $\hat{y} \in \widehat{S}$, $\hat{a}\hat{y} \cdot \hat{a}\hat{b} \geq 0$, alors $\left(\frac{\hat{a}\hat{y} \cdot \hat{a}\hat{b}}{\|\hat{a}\hat{y}\| \|\hat{a}\hat{b}\|} \right) \geq 0 \geq \min_{\hat{u} \in \{\hat{n}, \frac{\hat{a}\hat{y}}{\|\hat{a}\hat{y}\|} \text{ où } \hat{y} \in \widehat{S}' \setminus \{\hat{a}\}\}} \left(\frac{\hat{u} \cdot \hat{a}\hat{b}}{\|\hat{a}\hat{b}\|} \right)$ comme $\hat{n} \cdot \hat{a}\hat{b} = 0$.
- Si $\hat{y} \in \widehat{\mathcal{A}}$, il existe deux points \hat{b}_j, \hat{b}_{j+1} qui sont des extrémités de $\widehat{\mathcal{A}}$, et \hat{y} se trouve sur la frontière $\partial \widehat{K}$ entre \hat{b}_j et \hat{b}_{j+1} , tel que $\left| \widehat{b}_j \hat{a} \hat{y} \right| \leq \left| \widehat{b}_j \hat{a} \hat{b}_{j+1} \right|$ et $\left| \widehat{y} \hat{a} \hat{b}_{j+1} \right| \leq \left| \widehat{b}_j \hat{a} \hat{b}_{j+1} \right|$ (voir figure 3.12) où $\left| \widehat{bac} \right|$ désigne la valeur absolue de l'angle formé entre b et c à partir du point a .

FIGURE 3.12 – Illustration des zones $\hat{\mathcal{I}}_S^K$ sur deux cas

Ces deux inégalités montrent que $\arg \min_{\hat{y} \in \hat{\mathcal{A}}} \left(\frac{\hat{a}\hat{y} \cdot \hat{a}\hat{b}}{\|\hat{a}\hat{y}\| \|\hat{a}\hat{b}\|} \right) = \arg \min_{\hat{y} \in \{\hat{b}_j, \hat{b}_{j+1}\}} \left(\frac{\hat{a}\hat{y} \cdot \hat{a}\hat{b}}{\|\hat{a}\hat{y}\| \|\hat{a}\hat{b}\|} \right)$.
Cependant, comme \hat{b}_j et \hat{b}_{j+1} se trouvent sur \hat{S}' ,

$$\arg \min_{\hat{u} \in \{\hat{n}, \frac{\hat{a}\hat{y}}{\|\hat{a}\hat{y}\|} \text{ où } \hat{y} \in \partial \hat{\mathcal{R}}_S^K\}} \left(\frac{\hat{u} \cdot \hat{a}\hat{b}}{\|\hat{a}\hat{b}\|} \right) = \hat{n}_a.$$

Nous avons le même résultat pour \hat{n}_b .

— Ceci montre que le point réalisant le minimum se trouve sur \hat{S}' .

Alors, la définition de \hat{n}_a comme un argmin sur \hat{S}' donnée dans la définition 3.3 est la même que celle évaluée sur la frontière $\partial \hat{\mathcal{R}}_S^K$. De plus, par continuité de la fonction à minimiser sur un ensemble fermé, l'argmin de cette fonction sur $\hat{\mathcal{R}}_S^K$ se trouve donc sur sa frontière également. Donc, $\hat{z} \in \hat{\mathcal{I}}_S^K$ et \hat{z} peut s'exprimer comme $\hat{z} = \hat{x} + t_{\hat{z}} \hat{n}_x$ où $\hat{x} \in \hat{S}$ et $t_{\hat{z}} \geq 0$.

Pour terminer cette preuve, montrons que

$$t_{\hat{z}} \leq t_{\hat{x}}.$$

Deux cas doivent être considérés : $\hat{z} \in \partial \hat{\mathcal{R}}_S^K$ et $\hat{z} \in \text{Int}(\hat{\mathcal{R}}_S^K)$.

Si $\hat{z} \in \partial \hat{\mathcal{R}}_S^K$, alors il y a les trois mêmes possibilités que précédemment : $\hat{z} \in \hat{S}$, $\hat{z} \in \hat{\mathcal{A}}$ ou $\hat{z} \in \hat{S}'$.

• Si $\hat{z} \in \hat{S}$, alors $t_{\hat{z}} = 0$ et $t_{\hat{z}} \leq t_{\hat{x}}$.

• Si $\hat{z} \in \hat{\mathcal{A}}$, $t_{\hat{z}} = t_{\hat{x}}$ comme $t_{\hat{x}}$ est défini par le premier cas dans la définition 3.3, et $t_{\hat{x}}$ est la borne supérieure (autrement le point est à l'extérieur de \hat{K}).

• Si $\hat{z} \in \hat{S}' = g_K^{-1}(P^1 g_{\mathcal{T}(K)} \cap K)$, alors $ab \wedge ag_K(\hat{x} + t_{\hat{z}} \hat{n}_x) = 0$.

Donc, $t_{\hat{z}} \in \left\{ t, \hat{x} + t \hat{n}_x \in \hat{K}, ab \wedge ag_K(\hat{x} + t \hat{n}_x) = 0 \right\}$ et il vient que $t_{\hat{z}} \leq t_{\hat{x}}$.

Finalement, dans tous les cas, si $\hat{z} \in \partial \hat{\mathcal{R}}_S^K$, alors $t_{\hat{z}} \leq t_{\hat{x}}$.

Considérons maintenant le cas $\hat{z} \in \text{Int}(\hat{\mathcal{R}}_S^K)$. En suivant la ligne partant de \hat{x} et guidée par \hat{n}_x , il existe $\hat{y} = \hat{x} + t_{\hat{y}} \hat{n}_x$ tel que $\hat{y} \in \partial \hat{\mathcal{R}}_S^K$ et $t_{\hat{z}} \leq t_{\hat{y}}$. Or, par le cas précédent, nous savons que $t_{\hat{y}} \leq t_{\hat{x}}$, et donc $t_{\hat{z}} \leq t_{\hat{x}}$. Finalement, $\hat{z} \in \hat{\mathcal{V}}_S^K$. \square

Ce lemme montre que les zones que l'on construit permettent de contrôler les zones de retournements.

3.2.2 Construction d'un estimateur *a posteriori* spécifique

Maintenant que nous avons une caractérisation des zones de retournements, nous voulons contrôler la distance de Hausdorff (2.6), non plus uniquement sur l'arête lors de la construction du maillage 1D mais dans cette zone élargie. Pour un élément $K \in \mathcal{T}_h$, une arête $\widehat{S} \subset \widehat{E}$, $\widehat{E} \in \mathcal{T}(\partial K)$, l'équation (2.12) définit pour des éléments droits un indicateur d'erreurs Δ_S^K sur cette arête, à savoir,

$$\begin{aligned} \Delta_S^K : (f, \tilde{f}) &\in C^0(S, \mathbb{Y}) \times C^0(\tilde{S}, \mathbb{Y}) \\ &\mapsto \sup_{\hat{x} \in \widehat{S}} \max \left(\alpha \left\| g_S(\hat{x}) - P^1 g_{\mathcal{T}(S)}(\hat{x}) \right\|_{\mathbb{X}}, \beta \left\| (f \circ g_S)(\hat{x}) - \left(\tilde{f} \circ P^1 g_{\mathcal{T}(S)} \right)(\hat{x}) \right\|_{\mathbb{Y}} \right). \end{aligned} \quad (3.8)$$

Dans la zone $\widehat{\mathcal{V}}_S^K$, nous définissons maintenant l'application suivante :

$$\begin{aligned} \Gamma_S^K : (f_{num}^K, f_{vis}^{\tilde{S}}) &\in C^0(K, \mathbb{Y}) \times C^0(\tilde{S}, \mathbb{Y}) \\ &\mapsto \sup_{\hat{x} \in \widehat{S}, t \in [0, t_{\hat{x}}]} \max \left(\alpha \left\| g_K(\hat{x} + t \hat{n}_x) - G(f_{num}^K, f_{vis}^{\tilde{S}}; \hat{x}, t) \right\|_{\mathbb{X}}, \right. \\ &\quad \left. \beta \left\| \left(f_{num}^K \circ g_K \right)(\hat{x} + t \hat{n}_x) - F(f_{num}^K, f_{vis}^{\tilde{S}}; \hat{x}, t) \right\|_{\mathbb{Y}} \right), \end{aligned} \quad (3.9)$$

où les fonctions F et G sont définies ci-après par (3.10) et (3.12). Chaque composante de G , notée G_i ($i = 1, \dots, 3$), est définie par :

$$G_i(f_{num}^K, f_{vis}^{\tilde{S}}; \hat{x}, t) = \arg \max_{\{G_i^\ell, \ell=1, \dots, 4\}} \left(\left| (g_K)_i(\hat{x} + t \hat{n}_x) - G_i^\ell(f_{num}^K, f_{vis}^{\tilde{S}}; \hat{x}, t) \right| \right), \quad i = 1, \dots, 3. \quad (3.10)$$

Chaque fonction G^ℓ , dont les composantes sont notées G_i^ℓ , ($i = 1, \dots, 3$), est donnée par

$$G^\ell(f_{num}^K, f_{vis}^{\tilde{S}}; \hat{x}, t) = P^1 g_{\mathcal{T}(E)}(\hat{x}) + t J_{g_K}^\ell \hat{n}_x, \quad \ell = 1, \dots, 4, \quad (3.11)$$

où $J_{g_K}^\ell$ sont équivalentes à des matrices jacobiniennes et sont construites à partir des maxima et/ou des minima des dérivées partielles de g_K :

$$\begin{aligned} J_{g_K}^1 &= \begin{pmatrix} \partial_1 g_1^{\max} & \partial_2 g_1^{\max} \\ \partial_1 g_2^{\max} & \partial_2 g_2^{\max} \\ \partial_1 g_3^{\max} & \partial_2 g_3^{\max} \end{pmatrix}, \quad J_{g_K}^2 = \begin{pmatrix} \partial_1 g_1^{\min} & \partial_2 g_1^{\min} \\ \partial_1 g_2^{\min} & \partial_2 g_2^{\min} \\ \partial_1 g_3^{\min} & \partial_2 g_3^{\min} \end{pmatrix}, \\ J_{g_K}^3 &= \begin{pmatrix} \partial_1 g_1^{\min} & \partial_2 g_1^{\max} \\ \partial_1 g_2^{\min} & \partial_2 g_2^{\max} \\ \partial_1 g_3^{\min} & \partial_2 g_3^{\max} \end{pmatrix}, \quad J_{g_K}^4 = \begin{pmatrix} \partial_1 g_1^{\min} & \partial_2 g_1^{\min} \\ \partial_1 g_2^{\min} & \partial_2 g_2^{\min} \\ \partial_1 g_3^{\min} & \partial_2 g_3^{\min} \end{pmatrix}, \end{aligned}$$

avec $\partial_j g_i^{\max}$ (resp. $\partial_j g_i^{\min}$) le maximum (resp. minimum) de la dérivée partielle de la i -ème composante de g_K dans la j -ème direction. Notons que dans (3.11), $J_{g_K}^\ell \hat{n}_x$ désigne un produit matrice vecteur.

De la même manière, nous définissons les fonctions F et F^ℓ ($\ell = 1, \dots, 4$) en substituant g_K par $f_{num}^K \circ g_K$ et $P^1 g_{\mathcal{T}(E)}$ par $f_{vis}^{\tilde{S}} \circ P^1 g_{\mathcal{T}(E)}$ et chaque $J_{f_{num} \circ g_K}^\ell$ est défini de la même façon que $J_{g_K}^\ell$ dans (3.10) et (3.11).

Les composantes F_i ($i = 1, \dots, 3$) de la fonction F sont donc données par

$$F_i(f_{num}^K, f_{vis}^{\tilde{S}}; \hat{x}, t) = \arg \max_{\{F_i^\ell, \ell=1, \dots, 4\}} \left(\left| (f_{num}^K \circ g_K)_i(\hat{x} + t \hat{n}_x) - F_i^\ell(f_{num}^K, f_{vis}^{\tilde{S}}; \hat{x}, t) \right| \right), \quad i = 1, \dots, 3. \quad (3.12)$$

Chaque fonction F^ℓ , dont les composantes sont notées F_i^ℓ , ($i = 1, \dots, 3$), est donnée par

$$F^\ell(f_{num}^K, f_{vis}^{\tilde{S}}; \hat{x}, t) = \left(f_{vis}^{\tilde{S}} \circ P^1 g_{\mathcal{T}(E)} \right) (\hat{x}) + t J_{f_{num} \circ g_K}^\ell \hat{n}_x, \quad \ell = 1, \dots, 4. \quad (3.13)$$

Les matrices $J_{f_{num} \circ g_K}^\ell$ sont également définies à l'aide de dérivées partielles :

$$J_{f_{num} \circ g_K}^1 = \begin{pmatrix} \partial_1 f_1^{\max} & \partial_2 f_1^{\max} \\ \partial_1 f_2^{\max} & \partial_2 f_2^{\max} \\ \partial_1 f_3^{\max} & \partial_2 f_3^{\max} \end{pmatrix}, \quad J_{f_{num} \circ g_K}^2 = \begin{pmatrix} \partial_1 f_1^{\max} & \partial_2 f_1^{\min} \\ \partial_1 f_2^{\max} & \partial_2 f_2^{\min} \\ \partial_1 f_3^{\max} & \partial_2 f_3^{\min} \end{pmatrix},$$

$$J_{f_{num} \circ g_K}^3 = \begin{pmatrix} \partial_1 f_1^{\min} & \partial_2 f_1^{\max} \\ \partial_1 f_2^{\min} & \partial_2 f_2^{\max} \\ \partial_1 f_3^{\min} & \partial_2 f_3^{\max} \end{pmatrix}, \quad J_{f_{num} \circ g_K}^4 = \begin{pmatrix} \partial_1 f_1^{\min} & \partial_2 f_1^{\min} \\ \partial_1 f_2^{\min} & \partial_2 f_2^{\min} \\ \partial_1 f_3^{\min} & \partial_2 f_3^{\min} \end{pmatrix},$$

où $\partial_j f_i^{\max}$ (resp. $\partial_j f_i^{\min}$) désigne le maximum (resp. minimum) de la dérivée partielle de la i -ème composante de f_{num} dans la j -ème direction.

La proposition 3.6 ci-dessous prouve que cet indicateur *a posteriori* spécifique contrôle la distance de Hausdorff dans ces zones d'intérêt.

Proposition 3.6. *Avec les notations précédentes, soit $K \in \mathcal{T}_h$ et $\hat{S} \in \mathcal{T}(\widehat{\partial K})$. Alors,*

$$\forall \hat{z} \in \hat{\mathcal{V}}_S^K, \quad \delta_{\mathcal{T}(K)} \left(f_{num}^K, f_{vis}^{\tilde{K}}; \hat{z} \right) \leq \Gamma_S^K \left(f_{num}^K, f_{vis}^{\tilde{S}} \right).$$

Preuve. Pour montrer le résultat, il suffit de prouver que pour n'importe quel maillage $\mathcal{T}(K)$ de K , tel que $\mathcal{T}(K)|_{\tilde{S}} = \tilde{S}$, G satisfait, pour tout $\hat{z} = \hat{x} + t \hat{n}_x \in \hat{\mathcal{V}}_S^K$,

$$\left\| g_K(\hat{z}) - P^1 g_{\mathcal{T}(K)}(\hat{z}) \right\|_{\mathbb{X}} \leq \left\| g_K(\hat{z}) - G(f_{num}^K, f_{vis}^{\tilde{S}}; \hat{x}, t) \right\|_{\mathbb{X}}.$$

Et, avec des arguments similaires, il suffit de montrer que

$$\left\| \left(f_{num}^K \circ g_K \right) (\hat{z}) - \left(f_{vis}^{\tilde{K}} \circ P^1 g_{\mathcal{T}(K)} \right) (\hat{z}) \right\|_{\mathbb{Y}} \leq \left\| \left(f_{num}^K \circ g_K \right) (\hat{z}) - F(f_{num}^K, f_{vis}^{\tilde{S}}; \hat{x}, t) \right\|_{\mathbb{Y}}.$$

Prenons un point arbitraire $\hat{z} = \hat{x} + t \hat{n}_x \in \hat{\mathcal{V}}_S^K$, la fonction $P^1 g_{\mathcal{T}(K)}(\hat{z})$ peut être exprimée comme :

$$P^1 g_{\mathcal{T}(K)}(\hat{z}) = P^1 g_{\mathcal{T}(K)}(\hat{x} + t \hat{n}_x) = P^1 g_{\mathcal{T}(E)}(\hat{x}) + t J_{P^1 g_{\mathcal{T}(K)}}(\hat{x}) \hat{n}_x,$$

où $J_{P^1 g_{\mathcal{T}(K)}}$ désigne la jacobienne associée à $P^1 g_{\mathcal{T}(K)}$ et où $P^1 g_{\mathcal{T}(E)}$ ne dépend uniquement que du maillage de l'arête E . De plus, pour chaque composante,

$$\min_{\ell=1, \dots, 4} \left(J_{g_K}^\ell \hat{n}_x \right)_i \leq \left(J_{P^1 g_{\mathcal{T}(K)}}(\hat{x}) \hat{n}_x \right)_i \leq \max_{\ell=1, \dots, 4} \left(J_{g_K}^\ell \hat{n}_x \right)_i, \quad i = 1, \dots, 3,$$

et

$$\left| (g_K)_i(\hat{z}) - (P^1 g_{\mathcal{T}(K)})_i(\hat{z}) \right| \leq \left| (g_K)_i(\hat{z}) - G(f_{num}^K, f_{vis}^{\tilde{S}}; \hat{x}, t) \right|, \quad i = 1, \dots, 3.$$

Nous en déduisons le résultat. Les arguments sont identiques pour traiter l'approximation des fonctions. \square

Pour satisfaire l'objectif \mathbf{O}_3 , et de manière similaire à ce qui a été fait dans le cas d'éléments droits avec l'indicateur $\Delta_{\mathcal{T}(E)}$ défini dans l'équation (2.13), l'indicateur $\Gamma_{\mathcal{T}(E)}$ doit prendre en compte toutes les contributions de toutes les traces des fonctions qui ont pour support l'arête considérée. Ainsi, pour une arête E , l'indicateur s'écrit :

$$\Gamma_{\mathcal{T}(E)} = \max_{S \in \mathcal{T}(E)} \Gamma_S = \max_{S \in \mathcal{T}(E)} \max_{\{K \in \mathcal{T}(X), S \in \mathcal{T}(\partial K)\}} \Gamma_S^K. \quad (3.14)$$

Cela induit la création d'une liste de fonctions connectées à E . Ces fonctions sont soit des fonctions 1D pures (dans le cas où il y a des segments), soit des traces de fonctions 2D ou 3D dont la frontière contient E comme support.

Il faut noter que l'évaluation de cet estimateur Γ_S^K est plus coûteuse que Δ_S^K puisque qu'au lieu de l'estimer sur une arête, une zone plus large est utilisée. Néanmoins, lorsque nous savons *a priori* qu'il ne peut pas y avoir de zones de retournements (dans le cas d'éléments droits), l'indicateur Δ_S^K défini dans (2.12) est utilisé.

3.3 Modifications de l'algorithme générant la représentation

Dans cette section, nous présentons les modifications apparaissant dans l'algorithme générant la représentation suite à la définition d'un indicateur d'erreurs spécifique (3.14).

3.3.1 Modifications dans l'algorithme général

En utilisant l'estimation *a posteriori* donnée par la proposition 3.6, il est possible, lors de la génération des maillages des frontières des éléments, de contrôler avec précision la représentation qui sera construite dans les zones de retournements pour n'importe quel maillage 2D reposant sur ce maillage de frontière. Aussi, une modification de l'algorithme sur le bord est nécessaire avec l'algorithme 6. Il est important de noter que la construction des maillages 2D est inchangée (algorithmes 1 et 2).

L'algorithme complet est donné dans l'algorithme 7.

Remarque 3.7. *Comme les éléments volumiques courbes ne sont pas traités, l'algorithme 7 se limite à la gestion des surfaces courbes. Néanmoins, il est toujours possible de traiter des volumes courbes par l'algorithme 5 donné dans le chapitre 2. Le seul problème est que dans ce cas, la bijectivité de l'approximation de la transformation géométrique n'est pas assurée. Cependant, si les maillages des bords sont suffisamment fins, il est tout à fait envisageable d'essayer de traiter des volumes courbes. En particulier, les maillages de bords seront assez fins si la trace de la fonction n'est pas triviale (i.e. si les variations de la fonction sont importantes).*

Algorithme 6 : Modification du maillage du bord (étape 3 de l'algorithme 3) dans le cas d'éléments courbes.

```

tant que ( $\Gamma_{\mathcal{T}(E)} > \varepsilon$ ) faire
  pour ( $S \in \mathcal{T}(E)$  et  $K \in \mathcal{T}(X)$  tels que  $S \in \mathcal{T}(\partial K)$ ) faire
    si ( $\Gamma_S^K > \varepsilon$ ) alors
      Évaluations des fonctions  $J_{g_K}^\ell$  et  $J_{f_{num} \circ g_K}^\ell$ , où  $\ell = 1, \dots, 4$ , définies dans
      (3.11).
      Définition de la zone  $\mathcal{V}_S^K$  dans (3.2).
      Calcul de  $\Gamma_S^K$  dans (3.9).
    fin
  fin
  Choisir  $\hat{x} = \arg \max \Gamma_{\mathcal{T}(E)}$  comme un nouveau point pour raffiner  $\mathcal{T}(\hat{E})$ .
fin

```

Algorithme 7 : Construction de la représentation f_{vis} pour des éléments courbes

Entrées : Maillage \mathcal{T}_h , solution numérique f_{num} , tolérance ε

Sorties : Représentation f_{vis}

1. Calcul des extrema des fonctions et de la géométrie. Évaluation des poids α et β , équation (2.8), introduits pour la distance de Hausdorff (2.6).
2. Décomposition de la frontière des éléments en dimension inférieure : 2D \rightarrow 1D.
3. Maillage des éléments 1D et des arêtes (notés $\tilde{\mathcal{T}}_{1D}$) :


```

pour chaque arête  $E \in \tilde{\mathcal{T}}_{1D}$  faire
  Évaluation de l'estimateur a posteriori  $\Gamma_{\mathcal{T}(E)}$ .
  tant que  $\Gamma_{\mathcal{T}(E)} > \varepsilon$  faire
    Raffinement du maillage de l'élément de référence  $\mathcal{T}(\hat{E})$ , du maillage de
    représentation  $\mathcal{T}(E)$  et mise à jour de  $\Gamma_{\mathcal{T}(E)}$ .
  fin
fin

```
4. Maillage des éléments 2D et des faces (notés $\tilde{\mathcal{T}}_{2D}$) :


```

pour chaque face  $F \in \tilde{\mathcal{T}}_{2D}$  faire
  Construction d'un maillage initial  $\mathcal{T}(\hat{F})$  à partir des maillages des arêtes  $\hat{E} \in \mathcal{E}(\hat{F})$ .
  Évaluation de l'estimateur a posteriori  $\Delta_{\mathcal{T}(F)}$ .
  tant que  $\Delta_{\mathcal{T}(F)} > \varepsilon$  faire
    Raffinement du maillage de l'élément de référence  $\mathcal{T}(\hat{F})$ , du maillage de
    représentation  $\mathcal{T}(F)$  et mise à jour de  $\Delta_{\mathcal{T}(F)}$ .
  fin
fin

```
5. Génération des sorties.

3.3.2 Théorème de convergence pour les éléments courbes

De la même manière que pour les éléments droits avec le théorème 2.10 donné dans le chapitre 2, nous établissons maintenant dans le théorème 3.8 un résultat sur la vérification des objectifs dans le cas courbe.

Théorème 3.8. *Soit ε une tolérance fixée et f_{num} une solution numérique appropriée (vérifiant les hypothèses \mathbf{H}_1 à \mathbf{H}_6). Si l'algorithme 7 a convergé, alors la représentation f_{vis} satisfait les objectifs \mathbf{O}_1 - \mathbf{O}_2 - \mathbf{O}_3 .*

Preuve. L'objectif \mathbf{O}_1 est toujours vérifié par construction puisque la représentation est définie à l'aide de fonctions affines sur simplexes.

Le point clef pour vérifier l'objectif \mathbf{O}_2 est toujours lié à la proposition 2.4 qui montre que la représentation locale $f_{vis}^{\tilde{K}}$ est une fonction si et seulement si $P^1g_{\mathcal{T}(K)}$ est une fonction bijective. Sans cela, la distance de Hausdorff n'est pas bien définie. Les zones, dites de retournements, qui peuvent amener des pertes de bijectivité, ont été définies et caractérisées. Des zones constructibles et locales ont ensuite été définies. Le lemme 3.5 a montré que ces dernières zones permettent d'englober les zones de retournements. La proposition 3.6 a ensuite montré que dans ces zones, l'estimateur $\delta_{\mathcal{T}(K)}$ est majorée par un indicateur d'erreur spécifique Γ_S^K que nous avons défini. Or, $\delta_{\mathcal{T}(K)}$ majore lui-même la distance de Hausdorff (proposition 2.6). Comme le point qui est ajouté est celui qui réalise l'erreur maximale, si l'estimateur Γ_S^K est borné par la tolérance, aucun point ne sera ajouté dans les zones de retournements. Finalement, la distance de Hausdorff est bien définie à chaque étape de l'algorithme et peut donc être contrôlée par $\Delta_{\mathcal{T}(K)}$. La proposition 2.6 montre donc que \mathbf{O}_2 est satisfait si la tolérance est une borne supérieure à l'estimateur $\Delta_{\mathcal{T}(K)}$ pour chaque élément K .

L'objectif \mathbf{O}_3 est toujours rempli par la proposition 2.8. □

3.3.3 Problème d'optimisation pour calculer le paramètre d'avancement maximal

Pour calculer le paramètre d'avancement maximal, nous distinguons deux cas suivant que les surfaces considérées se trouvent dans le plan ou dans l'espace. Nous commençons par traiter le cas plan que l'on généralise dans la section 3.4. Pour cela, un algorithme de Gauss-Newton est utilisé et est rappelé dans l'annexe C.1.

Nous nous intéressons au calcul du paramètre d'avancement maximal dans la zone de retournements dans le cas où l'élément est plan. Nous considérons un segment \hat{S} appartenant à la cellule de référence \hat{K} . On considère un point $\hat{x} \in \hat{S}$ et on cherche donc $t_{\hat{x}}$ le paramètre maximal d'avancement à l'intérieur de l'élément.

La figure 3.13 illustre la recherche de ce paramètre $t_{\hat{x}}$. Sur l'élément de référence, la trajectoire suivie à partir de \hat{x} , selon \hat{n}_x , jusqu'à atteindre la frontière \hat{S}' au point $\hat{x} + t_{\hat{x}}\hat{n}_x$ est rectiligne (en bleu sur la figure 3.13 à gauche). Sur l'élément réel, l'image par g_K de cette trajectoire n'est pas rectiligne (en bleu sur la figure 3.13 à droite) puisque la transformation géométrique g_K n'est pas linéaire. Nous cherchons un point $g_K(\hat{x} + t_{\hat{x}}\hat{n}_x)$ appartenant au segment $[g_K(\hat{a})g_K(\hat{b})]$. Cela se traduit par

$$\hat{x} + t_{\hat{x}}\hat{n}_x \in \hat{S}' \Leftrightarrow \exists \hat{x}' \in \hat{S}, \quad g_K(\hat{x} + t_{\hat{x}}\hat{n}_x) = P^1g_{\mathcal{T}(K)}(\hat{x}'). \quad (3.15)$$

Sur la figure 3.13, nous illustrons l'équation (3.15). Sur l'élément réel, S' correspond à un segment (en vert sur la figure 3.13 à droite). C'est l'image par g_K de \hat{S}' qui *a priori* n'est pas un segment

(également en vert sur la figure 3.13 à gauche). Nous en déduisons que la définition de \widehat{S}' dans le cas 2D plan est donnée par

$$\widehat{S}' = \left\{ \hat{x} + t_{\hat{x}} \hat{n}_x, \hat{x} \in \widehat{S} \mid \exists \hat{x}' \in \widehat{S}, \exists t_{\hat{x}} \geq 0, g_K(\hat{x} + t_{\hat{x}} \hat{n}_x) = P^1 g_{\mathcal{T}(K)}(\hat{x}') \right\} \quad (3.16)$$

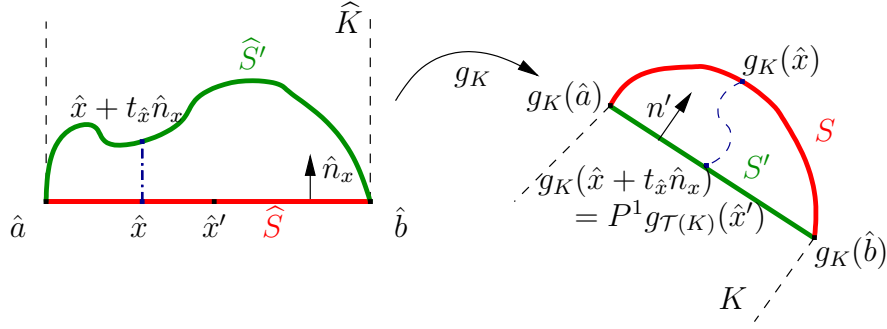


FIGURE 3.13 – Illustration de la recherche du paramètre d'avancement maximal.

Le problème de la définition donnée par l'équation (3.16) est qu'elle est implicite. Pour un point \hat{x} donné et un paramètre $t_{\hat{x}}$, il n'est pas facile de savoir si le point $g_K(\hat{x} + t_{\hat{x}} \hat{n}_x)$ appartient au segment, à savoir s'il existe un point $\hat{x}' \in \widehat{S}'$ tel que $g_K(\hat{x} + t_{\hat{x}} \hat{n}_x) = P^1 g_{\mathcal{T}(K)}(\hat{x}')$. Aussi, nous souhaitons nous affranchir de \hat{x}' en définissant une autre caractérisation de \widehat{S}' . Pour cela, nous introduisons une normale (entrante ou sortante) n' au segment $[g_K(\hat{a})g_K(\hat{b})]$ sur l'élément réel. Comme $P^1 g_{\mathcal{T}(K)}(\hat{x}') \in [g_K(\hat{a})g_K(\hat{b})]$, $g_K(\hat{a})P^1 g_{\mathcal{T}(K)}(\hat{x}') \perp n'$, on a :

$$\begin{aligned} \hat{x} + t_{\hat{x}} \hat{n}_x \in \widehat{S}' &\Leftrightarrow g_K(\hat{a})g_K(\hat{x} + t_{\hat{x}} \hat{n}_x) \perp n' \\ &\Leftrightarrow \langle g_K(\hat{x} + t_{\hat{x}} \hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}} = 0. \end{aligned}$$

Sur la figure 3.13, le vecteur $g_K(\hat{a})g_K(\hat{x} + t_{\hat{x}} \hat{n}_x)$ est orthogonal à n' . Nous en déduisons une nouvelle caractérisation de \widehat{S}' :

$$\widehat{S}' = \left\{ \hat{x} + t_{\hat{x}} \hat{n}_x, \hat{x} \in \widehat{S} \mid \exists t_{\hat{x}} \geq 0, \langle g_K(\hat{x} + t_{\hat{x}} \hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}} = 0 \right\}. \quad (3.17)$$

A partir de cette dernière caractérisation, la définition 3.9 donne, dans le cas 2D plan, une façon de calculer le paramètre qui nous intéresse.

Définition 3.9 (Calcul de $t_{\hat{x}}$ en 2D plan). *Considérons une arête $\widehat{S} = [\hat{a}\hat{b}]$ et un point $\hat{x} \in \widehat{S}$. Dans le cas où l'élément K se trouve dans le plan, $t_{\hat{x}}$ est défini par :*

$$t_{\hat{x}} = \arg \min_{t \geq 0} \frac{1}{2} \langle g_K(\hat{x} + t \hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}}^2, \quad (3.18)$$

où n' est la normale entrante ou sortante sur l'élément réel (qui vérifie $n' \perp g_K(\hat{a})g_K(\hat{b})$).

Un algorithme de Gauss-Newton est utilisé pour résoudre ce problème d'optimisation et est rappelé en annexe C.1. Les détails des calculs permettant d'obtenir les résultats suivants se trouvent en annexe C.3. Dans cette section, nous donnons directement les équations afin de réaliser un algorithme de Gauss-Newton pour résoudre notre problème.

La fonction à minimiser est la suivante :

$$h(t) = \frac{1}{2} \langle g_K(\hat{x} + t \hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}}^2. \quad (3.19)$$

Sa dérivée vaut

$$h'(t) = \langle g_K(\hat{x} + t\hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}} \langle J_{g_K}(\hat{x} + t\hat{n}_x)\hat{n}_x, n' \rangle_{\mathbb{X}}. \quad (3.20)$$

où J_{g_K} est la jacobienne de g_K . La fonction h' fait intervenir 2 produits scalaires. Le premier correspond à la condition d'orthogonalité entre le vecteur $g_K(\hat{a})g_K(\hat{x} + t\hat{n}_x)$ et la normale n' . Le second fait intervenir la normale n' et un vecteur résultant du produit matrice-vecteur entre la matrice jacobienne évaluée au point $\hat{x} + t\hat{n}_x$ et la normale \hat{n} . Nous cherchons donc le point t , noté $t_{\hat{x}}$, qui annule h' . Pour cela, on s'intéresse à la dérivée de h' :

$$h''(t) = \langle J_{g_K}(\hat{x} + t\hat{n}_x)\hat{n}_x, n' \rangle_{\mathbb{X}}^2 + \langle g_K(\hat{x} + t\hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}} \left\langle \left(H_{g_K}(\hat{x} + t\hat{n}_x)\hat{n}_x \right) \cdot \hat{n}_x, n' \right\rangle, \quad (3.21)$$

où H_{g_K} est la Hessienne de g_K dont les composantes sont données par :

$$\left(\left(H_{g_K}(\hat{x} + t\hat{n}_x)\hat{n}_x \right) \cdot \hat{n}_x \right)_i = \sum_{j=1}^2 \sum_{k=1}^2 \frac{\partial^2 g_i(\hat{x} + t\hat{n}_x)}{\partial \hat{x}_k \partial \hat{x}_j} \hat{n}_k \hat{n}_j, \quad i = 1, 3. \quad (3.22)$$

Notons que, dans ce cas plan, la troisième composante est toujours nulle. De manière très synthétique, l'algorithme est initialisé à

$$t_0 = 0,$$

et les itérations se font de la façon suivante

$$t_{n+1} = t_n - \frac{h'(t_n)}{h''(t_n)},$$

où h' et h'' sont données par (3.20) et (3.21). L'algorithme complet est donné dans l'algorithme 14 en annexe C.1.

3.3.4 Utilisation de DIRECT pour estimer l'indicateur spécifique

Maintenant que nous avons une caractérisation des zones de retournements et qu'un estimateur spécifique a été défini afin de résoudre ce problème, nous explicitons comment en pratique nous pouvons les utiliser.

Une paramétrisation de la zone pour estimer l'indicateur spécifique

Considérons une arête $S \in \tilde{\mathcal{T}}_{1D}$ issue d'un élément $K \in \mathcal{T}_h$. L'indicateur Γ_S^K donné par l'équation (3.9) est utilisé. Dans sa formulation, il est nécessaire de réaliser une optimisation sur un sous ensemble $\hat{\mathcal{V}}_S^K$ de \hat{K} . La difficulté vient du fait que $\hat{\mathcal{V}}_S^K$ est lui-même défini par un autre problème d'optimisation. Cela a pour conséquence que cette estimation devrait être compliquée et coûteuse. Aussi, pour pallier cette difficulté, le domaine $\hat{\mathcal{V}}_S^K$ qui n'est pas explicite est remplacé par un domaine arbitraire \mathcal{V}^0 beaucoup plus approprié pour réaliser cette optimisation avec DIRECT. Nous choisissons pour \mathcal{V}^0 un triangle isocèle dont les sommets sont $A(0,0), B(1,0), C(0.5,0.5)$. Néanmoins, une transformation supplémentaire est nécessaire pour passer du triangle isocèle à la zone d'intérêt. Un point est référencé sur le triangle isocèle par ces coordonnées (ω, τ) avec

$$\omega \in [0, 1], \quad \tau \in \begin{cases} [0, \omega], & \text{si } \omega \leq 0.5, \\ [0, 1 - \omega], & \text{sinon.} \end{cases}$$

Un point $\hat{z} \in \widehat{\mathcal{V}}_S^K$ peut être décomposé comme $\hat{z} = \hat{x} + t\hat{n}_x$ où $\hat{x} \in \widehat{S}$ et nous notons \hat{a} et \hat{b} les extrémités de \widehat{S} . Nous introduisons la bijection ψ entre \mathcal{V}^0 et $\widehat{\mathcal{V}}_S^K$ définie par

$$\psi : (\omega, \tau) \mapsto \hat{z} = \hat{a} + \omega\hat{a}\hat{b} + t\hat{n}_x \quad \text{avec} \quad t = \begin{cases} \frac{\tau}{\omega}t_{\hat{x}}, & \text{si } 0 \leq \omega \leq 0.5, \\ \frac{\tau}{1-\omega}t_{\hat{x}}, & \text{sinon.} \end{cases}$$

Avec cette nouvelle paramétrisation, l'algorithme DIRECT peut être utilisé sur le simplexe \mathcal{V}^0 . Cet algorithme renvoie la valeur de l'estimateur et les coordonnées (ω, τ) du point dans le triangle \mathcal{V}^0 où l'erreur est maximale. Puis, le point $\hat{z} \in \widehat{\mathcal{V}}_S^K$ est obtenu à l'aide de ψ . Pour cela, il est nécessaire de connaître la valeur de $t_{\hat{x}}$ qui est associée à un point $\hat{x} = (1 - \omega)\hat{a} + \omega\hat{b} \in \widehat{S}$. Celle-ci est obtenue en résolvant un problème d'optimisation qui est détaillé dans la section 3.3.3. Ce dernier calcul est coûteux donc il est très intéressant de limiter le nombre de calculs de ces paramètres $t_{\hat{x}}$, qui ne dépendent que du point \hat{x} associé. Or, la forme spécifique du domaine à explorer \mathcal{V}^0 , qui est un triangle isocèle et l'utilisation de l'algorithme DIRECT permettent de drastiquement diminuer le nombre d'évaluations de $t_{\hat{x}}$ à faire. Ceci vient du fait que beaucoup de points d'évaluation de l'algorithme DIRECT vont avoir la même valeur de $t_{\hat{x}}$ (voir figure 3.14). On rappelle qu'un élément est découpé par l'algorithme en 4 éléments de mêmes tailles et que la valeur sur chaque élément est évaluée au barycentre. Sur la figure 3.14, différents barycentres sont affichés en rouge et sont alignés verticalement (pointillé bleu).

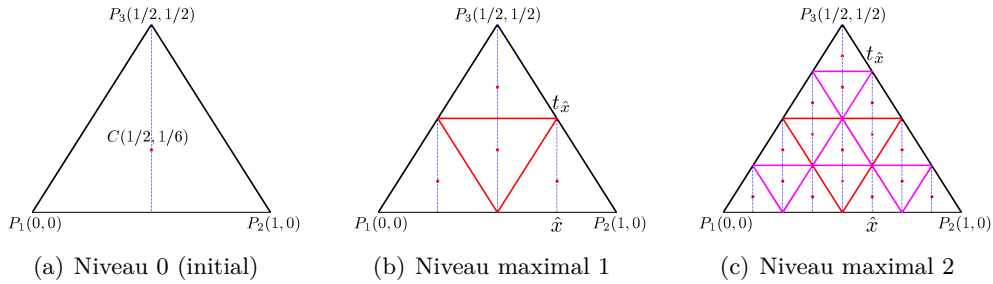


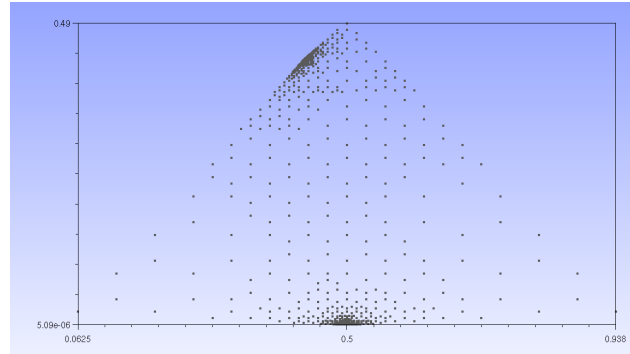
FIGURE 3.14 – Triangle initial exploré par DIRECT (a), et découpages jusqu'au niveau 1 (b) et 2 (c).

La figure 3.15 illustre certains points testés par l'algorithme DIRECT mis en place. Tout d'abord, les points sont testés par DIRECT sur le triangle isocèle \mathcal{V}^0 (figure 3.15.a). La bijection ψ permet ensuite de passer du triangle isocèle à l'élément de référence (figure 3.15.b). Enfin, la transformation géométrique g_K est utilisée pour revenir à l'élément réel (figure 3.15.c).

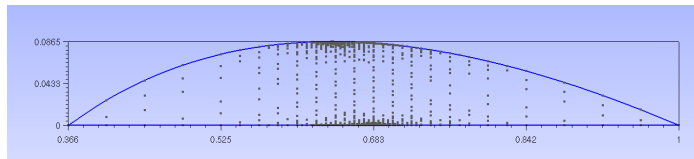
Alignement des nœuds

Nous souhaitons montrer que les nœuds (les sommets des éléments de DIRECT et leurs centres de gravité) sont alignés. Ensuite, nous allons introduire un algorithme de numérotation des sommets qui va nous permettre de stocker les valeurs $t_{\hat{x}}$ et donc éviter de refaire le problème d'optimisation.

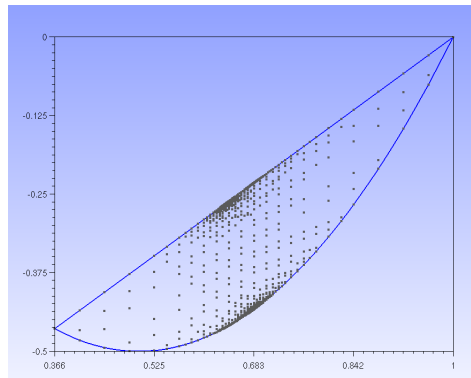
Le découpage d'un triangle suit toujours le même schéma. Un triangle T formé par les sommets P_1, P_2, P_3 est découpé en 4 triangles. Pour cela, trois nouveaux points (fils) P_{f_1}, P_{f_2} et P_{f_3} sont définis :



(a) Sur le triangle isocèle



(b) Sur la cellule de référence



(c) Sur l'élément réel

FIGURE 3.15 – Points testés par DIRECT sur le triangle isocèle (a), sur la cellule de référence (b) et sur l'élément réel (c).

- P_{f_1} est défini comme l'isobarycentre de P_2 et P_3 : $P_{f_1} = \frac{1}{2}(P_2 + P_3)$,
- P_{f_2} l'isobarycentre de P_1 et P_3 : $P_{f_2} = \frac{1}{2}(P_1 + P_3)$,
- P_{f_3} l'isobarycentre de P_1 et P_2 : $P_{f_3} = \frac{1}{2}(P_1 + P_2)$.

Les 4 triangles sont tels que :

- T_1 est formé par P_1, P_{f_2}, P_{f_3} ,
- T_2 par P_2, P_{f_1}, P_{f_3} ,
- T_3 par P_3, P_{f_1}, P_{f_2} ,
- T_4 par P_{f_1}, P_{f_2} et P_{f_3} .

Lemme 3.10 (Découpage d'un triangle isocèle en triangles isocèles). *Un triangle isocèle dont la base est parallèle à la base du triangle isocèle initial est découpé, par le découpage précédent, en 4 triangles isocèles dont les bases respectives sont également parallèles à la base du triangle isocèle initial.*

Preuve. Soit un triangle T formé par les triangles P_1, P_2 et P_3 isocèle en P_3 et tel que (P_1P_2) soit parallèle à la base du triangle initial (c'est-à-dire l'axe des abscisses).

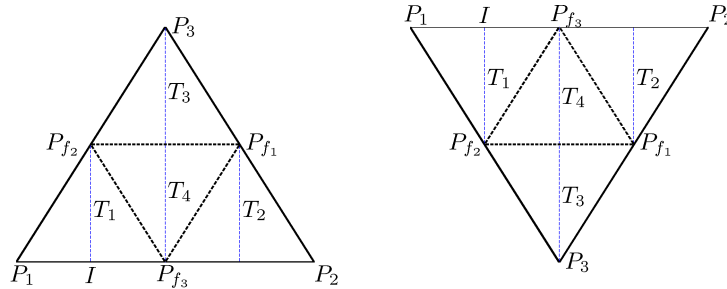


FIGURE 3.16 – Découpage de triangles isocèles en triangles isocèles

Les trois nœuds fils P_{f_1}, P_{f_2} et P_{f_3} sont définis comme précédemment. Comme T est isocèle en P_3 et que P_{f_1} et P_{f_2} sont définis comme les milieux respectifs de $[P_2P_3]$ et $[P_1P_3]$, nous en déduisons que $(P_{f_1}P_{f_2})$ est parallèle à (P_1P_2) par le théorème de Thalès.

Nous introduisons le point I qui est le milieu de $[P_1P_{f_3}]$. Comme P_{f_2} est le milieu de $[P_1P_3]$, nous en déduisons que $[IP_{f_2}]$ est parallèle à $[P_3P_{f_3}]$. Alors, $P_1P_{f_2} = P_{f_3}P_{f_2}$. Finalement, le triangle T_1 est isocèle en P_{f_2} et sa base $P_1P_{f_3}$ est parallèle à P_1P_2 car P_{f_3} est le milieu de P_1P_2 .

Par symétrie, $P_2P_{f_1} = P_{f_3}P_{f_1}$ et donc $P_{f_1}P_{f_3} = P_{f_2}P_{f_3}$.

On en déduit :

- T_2 est isocèle en P_{f_1} et sa base $P_2P_{f_3}$ est parallèle à P_1P_2 ,
- T_3 est isocèle en P_3 et sa base $P_{f_1}P_{f_2}$ est parallèle à P_1P_2 ,
- T_4 est isocèle en P_{f_3} et sa base $P_{f_2}P_{f_1}$ est parallèle à P_1P_2 .

Les notations utilisées sont reportées sur la figure 3.16. □

Corollaire 3.11. *Les centres de gravité des triangles isocèles ainsi créés ont la même abscisse (même ω) que les sommets de ces triangles isocèles.*

Démonstration. Comme tous les triangles sont isocèles et que leurs bases respectives sont parallèles à l'axe des abscisses, la hauteur issue du sommet du triangle isocèle passe par le centre de gravité et est orthogonale à l'axe des abscisses. Nous en déduisons que ces deux points, le centre de gravité et le sommet du triangle isocèle, ont la même abscisse. \square

Dans toute la suite, nous noterons indifféremment le paramètre d'avancement maximal $t_{\hat{x}}$ ou t_{ω} où ω est l'abscisse associée au point \hat{x} .

Définition 3.12. On note A^i l'ensemble $A^i = \left\{ \omega^i = \frac{k}{2^{i+1}} \mid k \in \llbracket 0, 2^{i+1} \rrbracket \right\}$, où i correspond à un niveau (entier naturel) considéré et $\llbracket \cdot \rrbracket$ désigne l'ensemble des entiers dans cet intervalle, à savoir dans ce cas les éléments $k \in \mathbb{N}$, tels que $0 \leq k \leq 2^{i+1}$.

Lemme 3.13 (Alignement des sommets des triangles). *L'abscisse ω d'un sommet d'un triangle de niveau i vérifie $\omega \in A^i$ (ω est donc un nombre rationnel).*

Preuve. Par récurrence, nous montrons qu'au niveau i , les abscisses des sommets des triangles appartiennent à l'ensemble A^i et que les sommets d'un triangle ont pour abscisses $k/(2^{i+1})$, $(k+1)/(2^{i+1})$ et $(k+2)/(2^{i+1})$ où $0 \leq k \leq 2^{i+1} - 2$.

L'ensemble d'exposant 0 est : $A^0 = \{ \omega_k^0 = k/2 \mid k \in \llbracket 0, 2 \rrbracket \} = \{0, 1/2, 1\}$. Au niveau 0, il y a un seul élément dont les 3 sommets sont définis en $P_1(0 \ 0)$, $P_2(1 \ 0)$, $P_3(0.5 \ 0.5)$, les abscisses de ces sommets appartiennent donc bien à A^0 et sont de la forme $k/2$, $(k+1)/2$ et $(k+2)/2$ où $k = 0$.

L'ensemble A^1 est : $A^1 = \{ \omega_k^1 = k/4 \mid k \in \llbracket 0, 2^2 \rrbracket \} = \{0, 1/4, 1/2, 3/4, 1\}$. Pour passer au niveau 1, le triangle initial est découpé en 4. Pour cela, trois nouveaux sommets sont créés à partir des sommets P_1 , P_2 et P_3 : $P_{f_1}(3/4 \ 1/4)$, $P_{f_2}(1/4 \ 1/4)$, $P_{f_3}(1/2 \ 0)$. Les abscisses de ces 6 sommets appartiennent bien à A^1 . Les 4 triangles sont formés comme sur la figure 3.16. On vérifie sans difficulté que pour chaque triangles, les abscisses des sommets sont de la forme $k/(2^2)$, $(k+1)/(2^2)$ et $(k+2)/(2^2)$ où $k = 0$ ou $k = 1$.

Supposons maintenant qu'au niveau i , les abscisses de tous les sommets appartiennent à A^i et que les sommets de chaque triangle ont pour abscisses $k/(2^{i+1})$, $(k+1)/(2^{i+1})$ et $(k+2)/(2^{i+1})$ où $0 \leq k \leq 2^{i+1} - 2$.

Considérons un triangle de niveau i de sommets P_1 , P_2 et P_3 dont les abscisses valent respectivement $\omega_1 = k/(2^{i+1})$, $\omega_2 = (k+2)/(2^{i+1})$ et $\omega_3 = (k+1)/(2^{i+1})$ où $k \in \llbracket 0, 2^{i+1} - 2 \rrbracket$. En suivant les mêmes notations que précédemment, 3 sommets fils sont créés et notés P_{f_1} , P_{f_2} et P_{f_3} ainsi que 4 triangles T_1 , T_2 , T_3 et T_4 . Les abscisses de P_{f_1} , P_{f_2} et P_{f_3} sont respectivement notées ω_{f_1} , ω_{f_2} et ω_{f_3} et vérifient :

$$\begin{aligned} \omega_{f_1} &= \frac{\omega_2 + \omega_3}{2} = \frac{1}{2} \left(\frac{k+2}{2^{i+1}} + \frac{k+1}{2^{i+1}} \right) = \frac{2k+3}{2^{i+1+1}}, \\ \omega_{f_2} &= \frac{\omega_1 + \omega_3}{2} = \frac{1}{2} \left(\frac{k}{2^{i+1}} + \frac{k+1}{2^{i+1}} \right) = \frac{2k+1}{2^{i+1+1}}, \\ \omega_{f_3} &= \frac{\omega_1 + \omega_2}{2} = \frac{1}{2} \left(\frac{k}{2^{i+1}} + \frac{k+2}{2^{i+1}} \right) = \frac{2k+2}{2^{i+1+1}}. \end{aligned} \tag{3.23}$$

Tout d'abord, notons que toutes ces abscisses sont de la forme

$$\omega_{k'}^{i+1} = \frac{k'}{2^{(i+1)+1}},$$

où $1 \leq k' \leq 2^{i+1+1} - 1$. On en déduit que tous ces nouveaux sommets appartiennent à A_{i+1} . De plus, tous les sommets déjà créés avant ce stade vérifient $\omega_k^i = \frac{k}{2^{i+1}}$ avec $k \in \llbracket 0, 2^{i+1} \rrbracket$, et vérifient également

$$\omega_k^i = \frac{k}{2^{i+1}} = \frac{2k}{2^{(i+1)+1}} = \omega_{k'}^{i+1},$$

où k' est un entier pair tel que $0 \leq k' = 2k \leq 2^{(i+1)+1}$. Ces sommets appartiennent donc également à A_{i+1} .

Nous voulons maintenant montrer que pour chaque triangle, les abscisses des sommets sont de la forme $k'/(2^{i+1+1})$, $(k' + 1)/(2^{i+1+1})$ et $(k' + 2)/(2^{i+1+1})$ où $0 \leq k' \leq 2^{i+1+1} - 2$. Nous récapitulons les différents cas possibles dans le tableau 3.1 qui nous permet de vérifier que l'hypothèse est vraie.

| Triangle | Sommet 1 ω_1 | Sommet 2 ω_2 | Sommet 3 ω_3 | Lien $k'-k$ | Amplitude k' |
|----------|--|--|--|---------------|--|
| T_1 | P_1 | P_{f_2} | P_{f_3} | $k' = 2k$ | $\llbracket 0, 2^{i+1} - 4 \rrbracket$ |
| | $\frac{k}{2^{i+1}} = \frac{2k}{2^{i+1+1}}$ | $\frac{2k+1}{2^{i+1+1}}$ | $\frac{2k+2}{2^{i+1+1}}$ | | |
| T_2 | P_{f_3} | P_{f_1} | P_2 | $k' = 2k + 2$ | $\llbracket 2, 2^{i+1} - 2 \rrbracket$ |
| | $\frac{2k+2}{2^{i+1+1}}$ | $\frac{2k+3}{2^{i+1+1}}$ | $\frac{k+2}{2^{i+1}} = \frac{2k+4}{2^{i+1+1}}$ | | |
| T_3 | P_{f_2} | P_3 | P_{f_1} | $k' = 2k + 1$ | $\llbracket 1, 2^{i+1} - 3 \rrbracket$ |
| | $\frac{2k+1}{2^{i+1+1}}$ | $\frac{k+1}{2^{i+1}} = \frac{2k+2}{2^{i+1+1}}$ | $\frac{2k+3}{2^{i+1+1}}$ | | |
| T_4 | P_{f_2} | P_{f_3} | P_{f_1} | $k' = 2k + 1$ | $\llbracket 1, 2^{i+1} - 3 \rrbracket$ |
| | $\frac{2k+1}{2^{i+1+1}}$ | $\frac{2k+2}{2^{i+1+1}}$ | $\frac{2k+3}{2^{i+1+1}}$ | | |

TABLE 3.1 – Configurations après subdivision du triangle.

□

Lemme 3.14 (Alignement des centres de gravité). *Au niveau i , les centres de gravité des éléments se situent en $\omega_k^i = \frac{k}{2^{i+1}}$ pour $k=1, \dots, 2^{i+1} - 1$.*

Preuve. D'après le corollaire 3.11, les centres de gravité des triangles ont la même abscisse que leurs sommets respectifs. Or, par le lemme 3.13, l'abscisse des sommets des triangles se situe en $\omega \in A^i$. Cependant les sommets d'abscisse 0 et 1 (qui ne sont autre que les points $P_1(0 \ 0)$ et $P_2(1 \ 0)$ du triangle initial), ne sont pas des sommets de triangles isocèles.

Les centres de gravité des éléments se situent en $A^i \setminus \{0, 1\}$, c'est-à-dire en $\omega_k^i = \frac{k}{2^{i+1}}$ pour $k = 1, \dots, 2^{i+1} - 1$. □

Remarque 3.15. *A priori, comme il y a 4^i éléments au maximum au niveau i , il y aurait $\sum_{j=0}^i 4^j$ évaluations de t_ω (associés aux centres de gravité des éléments) à faire au maximum (en considérant que tous les éléments sont découpés jusqu'au niveau i).*

Proposition 3.16. *Au niveau i , le nombre total de t_ω à calculer est $2^{i+1} - 1$. Ceux-ci sont associés aux abscisses $\omega_k = \frac{k}{2^{i+1}}$ pour $k = 1, \dots, 2^{i+1} - 1$. Pour passer du niveau i au niveau $i + 1$, 2^{i+1} nouveaux t_ω doivent être calculés au maximum.*

Preuve. Les fonctions étant évaluées aux centres de gravité des éléments, le nombre de t_ω à calculer (permettant de passer du triangle isocèle à l'élément de référence) est lié à l'abscisse des centres de gravité, qui se situent en $\omega_k^i = \frac{k}{2^{i+1}}$ pour $k = 1, \dots, 2^{i+1} - 1$. Il y a donc $2^{i+1} - 1$ différentes valeurs de t_ω à évaluer au niveau i . \square

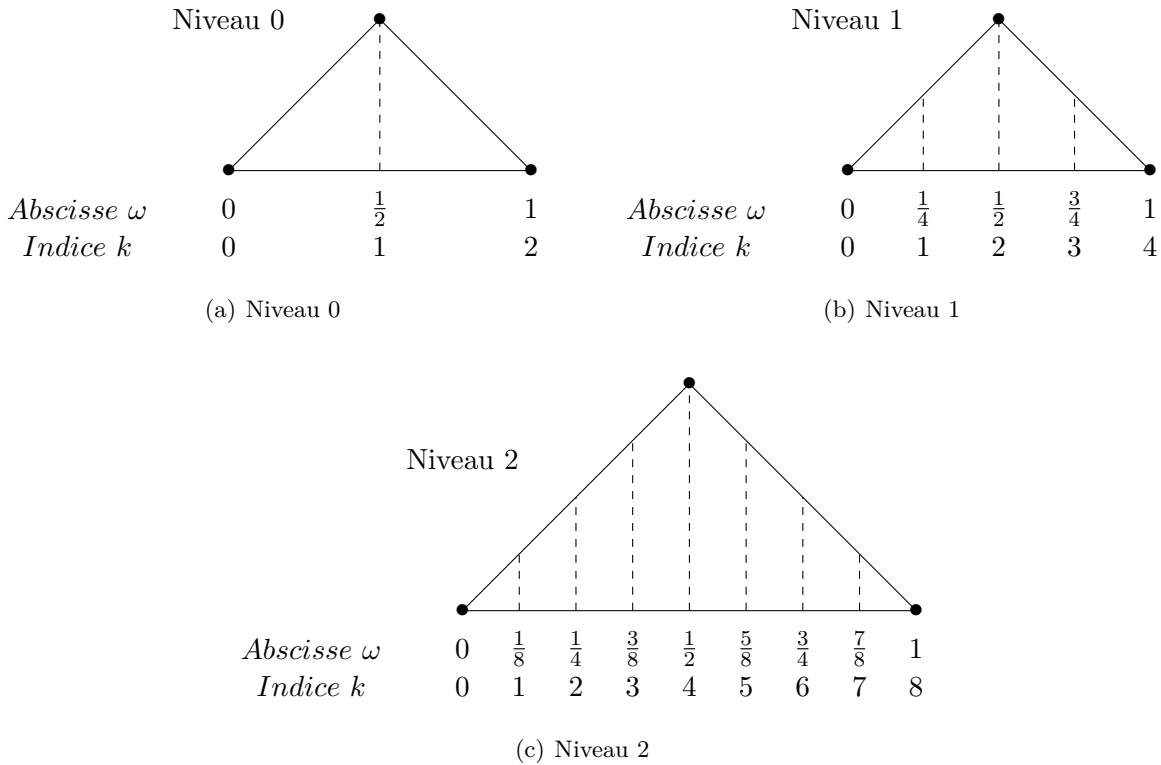


FIGURE 3.17 – Relations entre les abscisses ω et les indices k pour les premiers niveaux de DIRECT

Tout ceci permet de diminuer le nombre d'évaluations des paramètres d'avancée maximale de $2^{m+1} - 1$ au lieu de 4^m . La complexité passe donc de $\mathcal{O}(N)$ à $\mathcal{O}(\sqrt{N})$.

Stockage des paramètres d'avancée pour l'indicateur spécifique

Lors de l'utilisation de DIRECT, il est nécessaire pour chaque point considéré, de connaître la valeur de t_ω . Or, t_ω ne dépendant que de ω , si sa valeur a déjà été calculée, il est inutile de la recalculer, et on peut la récupérer si celle-ci a été stockée auparavant. Aussi, nous introduisons un stockage des t_ω . Pour cela, nous associons à chaque valeur de ω un entier k_ω . Le fait que k_ω soit entier est nécessaire pour pouvoir tester si la valeur t_ω a déjà été calculée. Dans le cas contraire, il y a un risque d'erreurs d'arrondis si on raisonne directement avec l'abscisse ω .

L'algorithme 8 donne un moyen facile de numérotter les sommets et donc d'avoir l'abscisse du centre de gravité d'un triangle. Ceci permet donc d'obtenir les coordonnées des points testés par

DIRECT pour évaluer l'estimateur. Le raffinement de $\widehat{S} \in \mathcal{T}(\widehat{E})$ est fait en divisant \widehat{S} autour d'un point réalisant le maximum de l'estimateur. L'abscisse d'un tel point peut directement être obtenue par l'algorithme 8.

Algorithme 8 : Numérotation des sommets dans DIRECT pour les éléments courbes.

Entrées : m le niveau maximal de DIRECT

1. Initialisation des indices associés aux sommets du triangle \mathcal{V}^0 : $A(k_A = 0)$, $B(k_B = 2^{m+1})$ et $C(k_C = 2^m)$.
2. Lorsqu'un triangle de sommets $V_1(k_{V_1})$, $V_2(k_{V_2})$ et $V_3(k_{V_3})$ est raffiné, nous notons V_4 , V_5 et V_6 les sommets des segments $[V_1V_2]$, $[V_1V_3]$ et $[V_2V_3]$, générant les triangles du niveau suivant $V_1V_4V_5$, $V_2V_4V_6$, $V_3V_5V_6$ et $V_4V_5V_6$. Les indices des nouveaux sommets sont :

$$k_{V_4} = (k_{V_1} + k_{V_2}) / 2, \quad k_{V_5} = (k_{V_1} + k_{V_3}) / 2, \quad k_{V_6} = (k_{V_2} + k_{V_3}) / 2.$$

3.4 Extension au traitement des surfaces courbes dans l'espace

Dans cette section, nous proposons une extension de ce travail au traitement des surfaces courbes dans l'espace. Les extensions s'inspirent directement du cas plan.

Dans le cas où l'élément ne se trouve pas dans le plan (i.e. surface dans l'espace), la caractérisation (3.16) ne peut pas être utilisée, car pour un point $\hat{x}' \in \widehat{S}$, il est possible d'avoir

$$\forall \hat{z} \in \widehat{K}, \quad P^1 g_{\mathcal{T}(K)}(\hat{x}') \neq g_K(\hat{z}). \quad (3.24)$$

Nous rajoutons donc une condition supplémentaire dans la définition 3.17.

Définition 3.17. *Considérons une arête $\widehat{S} = [\hat{a}\hat{b}]$. Dans le cas où l'élément K se trouve dans l'espace, S' est défini par :*

$$S' = \left\{ g_K(\hat{y}), \hat{y} = \arg \min_{\hat{z} \in \widehat{K} / \varphi_{\hat{x}^*}(\hat{z})=0} \left\| P^1 g_{\mathcal{T}(K)}(\hat{x}^*) - g_K(\hat{z}) \right\|_{\mathbb{X}} \mid \hat{x}^* \in \widehat{S} \right\}, \quad (3.25)$$

où,

$$\varphi_{\hat{x}^*}(\hat{z}) = \left\langle g_K(\hat{b}) - g_K(\hat{a}), P^1 g_{\mathcal{T}(K)}(\hat{x}^*) - g_K(\hat{z}) \right\rangle, \quad \hat{x}^* \in \widehat{S}.$$

La fonction $\varphi_{\hat{x}^*}$ introduite impose une condition d'orthogonalité. Nous souhaitons que le point cherché se trouve dans un plan orthogonal à l'arête de l'élément réel. Cette définition est cohérente avec le cas plan puisque dans ce cas $P^1 g_{\mathcal{T}(K)}(\hat{x}^*) = g(\hat{z})$ et donc $\varphi_{\hat{x}^*}(\hat{z}) = 0$.

Nous cherchons maintenant une caractérisation de la définition 3.17. On fixe $\hat{x} \in \widehat{S}$ et on cherche $\hat{z} \in \widehat{K}$, on introduit les fonctions suivantes :

$$\begin{aligned} \tilde{h}(\hat{z}) &= \left\| P^1 g_{\mathcal{T}(K)}(\hat{x}) - g_K(\hat{z}) \right\|_{\mathbb{X}}^2, \\ h(\hat{z}) &= P^1 g_{\mathcal{T}(K)}(\hat{x}) - g_K(\hat{z}). \end{aligned}$$

Nous avons donc un problème de minimisation sous contrainte [Bertsekas, 2014] à résoudre :

$$\begin{cases} \min_{\hat{z} \in \hat{K}} \tilde{h}(\hat{z}) \\ \text{sous } \varphi_{\hat{x}^*}(\hat{z}) = 0 \end{cases} \quad (3.26)$$

Nous posons le lagrangien de ce problème :

$$\mathcal{L}(\lambda, \hat{z}) = \tilde{h}(\hat{z}) + \lambda \varphi(\hat{z}), \quad (3.27)$$

où $\lambda \in \mathbb{R}$ est un multiplicateur de Lagrange. Soit (λ^*, \hat{z}^*) le couple vérifiant :

$$\mathcal{L}(\lambda^*, \hat{z}^*) = \min_{(\lambda, \hat{z}) \in \mathbb{R} \times \hat{K}} \mathcal{L}(\lambda, \hat{z})$$

Ce couple (λ^*, \hat{z}^*) satisfait également

$$\nabla \tilde{h}(\hat{z}^*) + \lambda^* \nabla \varphi(\hat{z}^*) = 0. \quad (3.28)$$

Or,

$$\begin{aligned} \nabla \varphi(\hat{z}) &= -Jg_K^T(\hat{z}) \left(g_K(\hat{b}) - g_K(\hat{a}) \right), \\ \nabla \tilde{h}(\hat{z}) &= -2Jg_K^T(\hat{z})h(\hat{z}), \end{aligned}$$

où $\nabla \varphi$ est obtenu par un produit matrice-vecteur entre $Jg_K^T(\hat{z})$ la transposée de la jacobienne de la transformation géométrique g_K au point \hat{z} et le vecteur $g_K(\hat{a})g_K(\hat{b})$. L'équation (3.28) peut être interprétée comme une condition de colinéarité entre les deux vecteurs $\nabla \tilde{h}$ et $\nabla \varphi$. La solution de ce problème vérifie donc

$$Jg_K^T(\hat{z}^*) \left(g_K(\hat{b}) - g_K(\hat{a}) \right) \wedge Jg_K^T(\hat{z}^*)h(\hat{z}^*) = 0. \quad (3.29)$$

Nous pouvons de plus remarquer que

$$\varphi_{\hat{x}^*}(\hat{z}) = 0 \Leftrightarrow P^1 g_{\mathcal{T}(K)}(\hat{x}^*) = g_K(\hat{a}) + \langle g_K(\hat{z}) - g_K(\hat{a}), u \rangle u, \quad (3.30)$$

où

$$u = \frac{g_K(\hat{b}) - g_K(\hat{a})}{\|g_K(\hat{b}) - g_K(\hat{a})\|_{\mathbb{X}}}.$$

En utilisant l'équation (3.30), nous pouvons expliciter $h(\hat{z}^*)$:

$$h(\hat{z}^*) = g_K(\hat{a}) + \langle g_K(\hat{z}^*) - g_K(\hat{a}), u \rangle u - g_K(\hat{z}^*). \quad (3.31)$$

En utilisant (3.29) et (3.31), il vient que

$$\begin{aligned} Jg_K^T(\hat{z}^*) \left(g_K(\hat{b}) - g_K(\hat{a}) \right) \wedge \left(Jg_K^T(\hat{z}^*) \left(g_K(\hat{z}^*) - g_K(\hat{a}) \right) \right. \\ \left. + \langle g_K(\hat{z}^*) - g_K(\hat{a}), u \rangle Jg_K^T(\hat{z}^*)u \right) = 0. \end{aligned} \quad (3.32)$$

Finalement, comme \hat{u} est colinéaire au vecteur $g_K(\hat{a})g_K(\hat{b})$, on obtient

$$Jg_K^T(\hat{z}^*) \left(g_K(\hat{b}) - g_K(\hat{a}) \right) \wedge Jg_K^T(\hat{z}^*) \left(g_K(\hat{z}^*) - g_K(\hat{a}) \right) = 0. \quad (3.33)$$

Remarque 3.18. *La matrice jacobienne est une matrice rectangulaire puisque la cellule de référence est de dimension deux et l'élément réel de dimension trois. L'équation (3.33) peut être interprétée comme une condition nécessaire de colinéarité dans le plan tangent à la surface en \hat{z}^* .*

Remarque 3.19 (Lien avec la condition donnée en 2D plan). *Dans le cas 2D plan, la matrice jacobienne Jg_K est une matrice carrée qui est inversible, la condition s'écrit :*

$$\left(g_K(\hat{b}) - g_K(\hat{a})\right) \wedge \left(g_K(\hat{z}^*) - g_K(\hat{a})\right) = 0$$

Nous obtenons une simple condition de colinéarité équivalente à la condition donnée pour le cas 2D plan.

Comme précédemment, $\hat{z} \in \hat{K}$ s'écrit à partir d'un point $\hat{x} \in \hat{S}$ et de \hat{n}_x :

$$\hat{z} = \hat{x} + t\hat{n}_x \quad (\text{avec } 0 \leq t).$$

En remplaçant dans l'équation (3.33), nous obtenons

$$Jg_K^T(\hat{x} + t\hat{n}_x)(b - a) \wedge Jg_K^T(\hat{x} + t\hat{n}_x)(g_K(\hat{x} + t\hat{n}_x) - a) = 0, \quad (3.34)$$

où on a utilisé les identités $a = g_K(\hat{a})$ et $b = g_K(\hat{b})$.

Définition 3.20 (Calcul de $t_{\hat{x}}$ pour des surfaces dans l'espace). *Considérons une arête $\hat{S} = [\hat{a}\hat{b}]$ et un point $\hat{x} \in \hat{S}$. Dans le cas où l'élément K se trouve dans l'espace, $t_{\hat{x}}$ est défini par :*

$$t_{\hat{x}} = \max \left\{ \arg \min_{t \geq 0} \left\| Jg_K^T(\hat{x} + t\hat{n}_x) \left(g_K(\hat{b}) - g_K(\hat{a}) \right) \wedge Jg_K^T(\hat{x} + t\hat{n}_x) \left(g_K(\hat{x} + t\hat{n}_x) - g_K(\hat{a}) \right) \right\|_{\mathbb{X}}^2 \right\}. \quad (3.35)$$

Pour déterminer $t_{\hat{x}}$ défini par (3.35), nous utilisons un algorithme de Gauss-Newton et nous introduisons les fonctions Ψ et f définies par

$$\begin{aligned} \Psi(t) &= \|f(t)\|_{\mathbb{X}}^2, \\ f(t) &= Jg_K^T(\hat{x} + t\hat{n}_x) (b - a) \wedge Jg_K^T(\hat{x} + t\hat{n}_x) (g_K(\hat{x} + t\hat{n}_x) - a) \\ &= u(t) \wedge v(t), \end{aligned}$$

avec,

$$\begin{aligned} u(t) &= Jg_K^T(\hat{x} + t\hat{n}_x) (b - a), \\ v(t) &= Jg_K^T(\hat{x} + t\hat{n}_x) (g_K(\hat{x} + t\hat{n}_x) - a). \end{aligned}$$

Les composantes de u sont notées u_k et vérifient

$$u_k(t) = \sum_{i=1}^3 \frac{\partial g_K(\hat{x} + t\hat{n}_x)}{\partial \hat{x}_k} (ab)_i, \quad k = 1, 2.$$

On en déduit les composantes u'_k de la dérivée de u' :

$$u'_k(t) = \sum_{i=1}^3 \left(\sum_{j=1}^2 \frac{\partial^2 g_K(\hat{x} + t\hat{n}_x)}{\partial \hat{x}_k \partial \hat{x}_j} \hat{n}_j \right) (b_i - a_i), \quad k = 1, 2.$$

De la même façon, les composantes de v sont notées v_k

$$v_k(t) = \sum_{i=1}^3 \frac{\partial g_K(\hat{x} + t\hat{n}_x)}{\partial \hat{x}_k} (g_K(\hat{x} + t\hat{n}_x) - a)_i, \quad k = 1, 2.$$

De plus,

$$\begin{aligned} v'_k(t) &= \sum_{i=1}^3 \left(\sum_{j=1}^2 \frac{\partial^2 g_K(\hat{x} + t\hat{n}_x)}{\partial \hat{x}_k \partial \hat{x}_j} \hat{n}_j \right) (g_K(\hat{x} + t\hat{n}_x) - a)_i \\ &+ \underbrace{\sum_{i=1}^3 \frac{\partial g_K(\hat{x} + \hat{n}_x)}{\partial \hat{x}_k} \left(\sum_{j=1}^2 \frac{\partial g_K(\hat{x} + t\hat{n}_x)}{\partial \hat{x}_j} \hat{n}_j \right)}_{=(Jg_K^T(\hat{x} + t\hat{n}_x)Jg_K(\hat{x} + t\hat{n}_x)\hat{n}_x)_k}, \quad k = 1, 2. \end{aligned}$$

Finalement,

$$\begin{aligned} f'(t) &= u'(t) \wedge v(t) + u(t) \wedge v'(t). \\ \Psi'(t) &= 2 \langle f(t), f'(t) \rangle. \end{aligned}$$

L'algorithme s'écrit donc de la façon suivante

$$\begin{aligned} t_0 &= 0, \\ t_{n+1} &= t_n - \frac{\Psi(t_n)}{\Psi'(t_n)}. \end{aligned}$$

Ceci permet de calculer $t_{\hat{x}}$ et donc de caractériser les zones de retournements.

3.5 Conclusions du chapitre

Dans ce chapitre, nous avons modifié certains aspects de notre approche pour pouvoir traiter des éléments courbes. En particulier, un estimateur spécifique est introduit. Celui-ci permet de raffiner les maillages de bord afin d'éviter d'avoir des recouvrements ou retournements dus à une perte de bijectivité de l'approximation de la transformation géométrique. Pour cela, les zones de retournements sont identifiées et caractérisées. L'algorithme DIRECT est utilisé pour évaluer l'estimateur spécifique. Bien que la méthodologie et les résultats soient basés pour des surfaces courbes, il est envisageable de traiter des volumes courbes également.

Le chapitre suivant est consacré aux validations numériques de la méthode de visualisation définie à partir du chapitre précédent et étendue dans ce chapitre.

Chapitre 4

Validations numériques de la méthode de visualisation

Contents

| | | |
|------------|---|------------|
| 4.1 | Validations pour le traitement des éléments droits | 92 |
| 4.1.1 | Comparaison des contrôles en normes L^2 et L^∞ | 92 |
| 4.1.2 | Comparaison du raffinement adaptatif avec un raffinement uniforme : Fonction de base Gauss Q^3 | 94 |
| 4.1.3 | Comparaison du raffinement adaptatif avec un raffinement uniforme : Problème d'élasticité résolu par la méthode Galerkin Discontinue | 95 |
| 4.1.4 | Exemples volumiques : modes sur hexaèdres. | 102 |
| 4.2 | Validations pour le traitement des éléments courbes | 104 |
| 4.2.1 | Fonction de base Gauss-Lobatto Q^6 sur un quadrangle Q^2 | 104 |
| 4.2.2 | Retournements dans l'espace | 107 |
| 4.2.3 | Problème de propagation d'onde acoustique. | 109 |
| 4.3 | Amélioration de la qualité des éléments en considérant le voisinage de la frontière | 111 |
| 4.4 | Conclusions du chapitre | 115 |

Dans le chapitre 2, nous avons construit une méthodologie et des algorithmes pour pouvoir visualiser des méthodes d'ordre élevé sur des éléments droits, ceux-ci pouvant être des surfaces ou des volumes. Dans le chapitre 3, nous avons étendu la méthode pour pouvoir traiter des surfaces courbes.

Dans ce chapitre, nous validons sur des exemples numériques ce qui a été présenté dans les deux précédents chapitres.

Dans un premier temps, nous montrons l'intérêt de la méthode sur des éléments droits. Nous commençons par montrer pourquoi un contrôle ponctuel est crucial en comparant notre méthode basée sur une norme L^∞ à une approche qui serait guidée par une norme L^2 . Ensuite, nous comparons l'approche adaptative que nous proposons à des raffinements réguliers sur plusieurs exemples de complexité croissante.

Dans un deuxième temps, les traitements spécifiques liés aux éléments courbes sont mis en évidence. Pour cela, les deux types d'indicateurs d'erreurs, l'un ne prenant en compte que le bord des arêtes, l'autre prenant en compte un voisinage, sont utilisés dans différentes configurations.

Dans un troisième temps, nous constatons que la qualité des éléments du maillage de représentation peut être de piètre qualité dans certaines situations très spécifiques et nous proposons

de l'améliorer en nous basant sur ce qui a été fait pour les éléments courbes où un voisinage de la frontière est considéré.

4.1 Validations pour le traitement des éléments droits

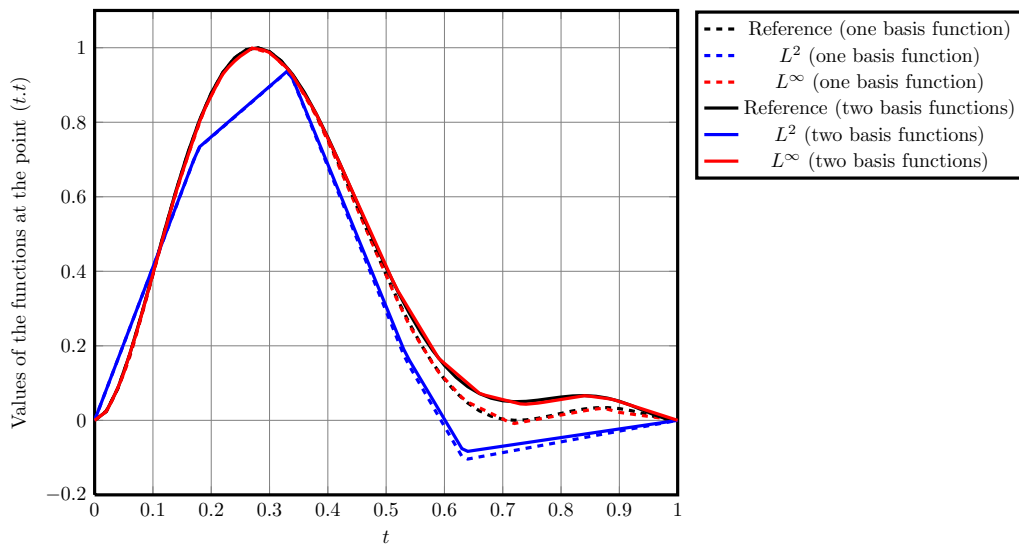
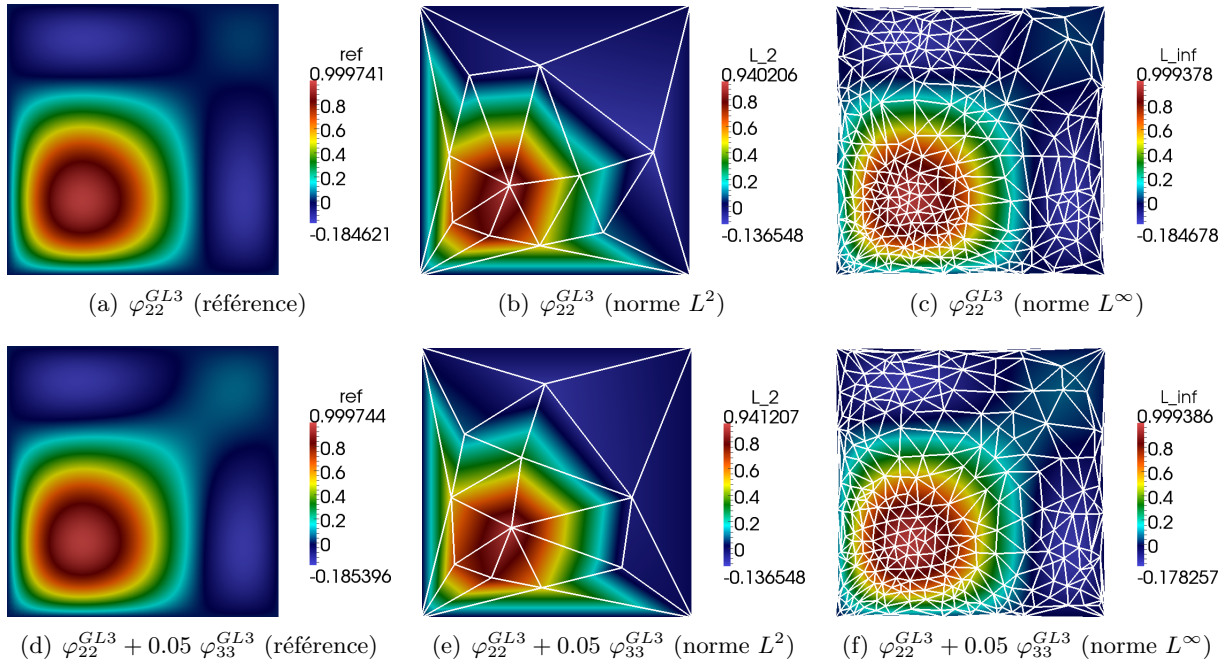
Dans cette section, nous nous intéressons à la validation de la démarche sur différents exemples numériques. Nous commençons par comparer un raffinement basé sur une norme L^2 à notre approche qui assure un contrôle ponctuel à l'aide de la norme L^∞ . On pourrait en effet être tenté d'utiliser une norme L^2 puisque l'évaluation de l'erreur de visualisation en norme L^2 est très facile et rapide à mettre en place. Ensuite, nous montrons les intérêts d'une approche adaptative par rapport à un redécoupage uniforme qui serait fait naïvement.

4.1.1 Comparaison des contrôles en normes L^2 et L^∞

Plusieurs critères peuvent être utilisés pour contrôler l'erreur de visualisation comme nous l'avons vu dans la section 1.3.3. Le but de ces paragraphes est de justifier le besoin d'un contrôle ponctuel. Ceci est important puisque dans la suite, les erreurs de visualisation seront calculées en norme L^∞ dans cette optique. Sur la figure 4.1, nous comparons des représentations obtenues lorsque la construction est guidée en suivant les normes L^2 et L^∞ . Les fonctions considérées sont des fonctions de type Gauss-Lobatto d'ordre $p = 3$. Celles-ci sont notées φ_{ij}^{GLp} , $i, j = 1, \dots, p+1$ où p désigne l'ordre. Les points de Gauss-Lobatto $\xi_{ij} = (\xi_i, \xi_j)$ sont définis à partir des points de Gauss-Lobatto de dimension un ξ_i [Abramowitz and Stegun, 1964]. Le support de ces solutions est à chaque fois le carré unité, la représentation de la géométrie est donc triviale. Sur la figure 4.1.a, la fonction φ_{22}^{GL3} est représentée sur un maillage très fin, il s'agit de la visualisation de « référence » que l'on souhaite représenter.

La figure 4.1.b est obtenue par une stratégie adaptative mais guidée par la norme L^2 sous la tolérance 1%. Ceci signifie que pour chaque élément du maillage de représentation, l'erreur de visualisation (i.e. l'écart entre la solution numérique et sa représentation) est calculée en norme L^2 , et tant qu'un élément a une erreur plus grande que la tolérance fixée, ce dernier est raffiné par l'ajout d'un nœud au maillage de représentation au centre de gravité de l'élément en question. Même si la tolérance est assez fine, le rendu est très éloigné de la solution de référence. La figure 4.1.c est obtenue par notre approche, avec la norme L^∞ , pour la même tolérance de 1 %. La différence est que cette fois-ci, un contrôle ponctuel est assuré sur tout l'élément. Les maillages de représentation sur ces deux dernières figures sont affichés en blanc.

Sur les trois figures suivantes, 4.1.d, 4.1.e and 4.1.f, la fonction est la somme de deux fonctions de base de type Gauss-Lobatto : $\varphi_{22}^{GL3} + 0.05 \varphi_{33}^{GL3}$. Cette dernière est donc très proche de la première fonction de base de Gauss-Lobatto mais avec l'ajout d'une seconde fonction de base dont l'amplitude est de 5 % la première. Une fois de plus, la figure 4.1.d fait office de référence où la fonction est représentée sur un maillage très fin. Lorsque la norme L^2 est utilisée pour représenter cette fonction, figure 4.1.e, sous la même tolérance de 1 %, l'ajout de la seconde fonction de base n'est pas « détectée » et le rendu est quasiment similaire au premier cas où il n'y avait qu'une fonction de base. Cependant, on s'attendrait à ce que la perturbation $0.05 \varphi_{33}^{GL3}$ soit détectée car la précision cible de 1% est donc plus petite que le taux d'amplitude de 5% de cette perturbation. Une part significative de l'information est donc perdue. Le contrôle en norme L^∞ permet quant à lui d'éviter ce genre d'artefact de visualisation puisque, sur la figure 4.1.f, les deux fonctions de bases apparaissent bien.



(g) Un tracé par rapport à une diagonale du carré des 6 fonctions précédentes (*plot over line*)

FIGURE 4.1 – Représentations de fonctions de base de Gauss-Lobatto d'ordre 3 sur un carré (précision 1%).

Finalement, sur la figure 4.1.g, chacune des 6 fonctions précédentes est tracée par rapport à une diagonale du carré (*plot over line*). Les extrémités du segment sont deux points du carré : (0,0) et (1,1). Cette extraction confirme qu'avec la norme L^2 , le rendu ne détecte par la seconde fonction de base. Les deux tracés des représentations obtenues par la norme L^2 sont quasiment identiques. Lorsque la norme L^∞ est utilisée à la place, la différence est perçue et les courbes « suivent » la solution de référence.

Remarque 4.1. *Le contrôle ponctuel demandé à un coût. En effet, alors qu'il est très rapide de calculer l'erreur de visualisation en norme L^2 par exemple, l'estimation en norme L^∞ est évidemment beaucoup plus couteuse. Cette question des coûts n'est pas abordée dans ce chapitre où nous montrons les intérêts d'utiliser notre méthode de visualisation. Cependant, dans les conclusions de ce manuscrit, nous discuterons des coûts et donnerons des pistes pour accélérer ces estimations.*

4.1.2 Comparaison du raffinement adaptatif avec un raffinement uniforme : Fonction de base Gauss Q^3

Dans cette partie, nous montrons l'intérêt du raffinement adaptatif sur des cas concrets où l'utilisation d'une subdivision uniforme va nécessiter un nombre de points important sous peine d'avoir une précision faible sur la représentation, c'est à dire une erreur de visualisation non négligeable. Nous considérons d'abord une « simple » fonction de base puis un cas plus concret issu d'un résultat numérique.

Les fonctions de base sont les clefs de voûte des méthodes numériques. Il est donc naturel de s'intéresser à leur représentation puisqu'une solution numérique d'ordre élevé est construite à partir de fonctions de base et de degrés de liberté. Le premier cas qui nous intéresse est une fonction de base de type Gauss, d'ordre 3. Nous nous intéressons à une seule fonction de base définie sur le carré unitaire. L'approximation de la géométrie est triviale puisqu'il n'y a pas d'erreur géométrique à considérer. Les fonctions de base de type Gauss correspondent à des polynômes de Lagrange notés φ_{ij}^{Gp} , $i, j = 1, \dots, p+1$ (où p désigne l'ordre du polynôme) définies à partir des points de Gauss $\xi_{ij} = (\xi_i, \xi_j)$ où ξ_i sont les points de Gauss définis sur le segment unité [Abramowitz and Stegun, 1964]. Les points de Gauss sur l'élément carré de référence sont donc simplement définis par une tensorisation des points 1D. Plus de détails sur ces fonctions de base pourra se trouver dans l'annexe B.

Sur la figure 4.2, nous avons représenté sur un maillage de représentation très fin, une telle fonction de base, en l'occurrence φ_{22}^{G3} , définie sur le carré unité. Le maillage de représentation est montré sur la figure 4.2.b et comporte 80000 triangles. Ce nombre est bien entendu énorme pour représenter une « simple » fonction de base d'ordre 3 mais cela nous permet d'avoir une visualisation de référence.

Sur la figure 4.3, nous comparons les représentations obtenues lorsque le maillage de représentation est défini par la méthode adaptative avec des maillages de représentation uniformes. Pour une précision demandée de 1%, la méthode adaptative nécessite 700 triangles dans le maillage de représentation : ces résultats se trouvent sur les figures 4.3.a et 4.3.d. Ensuite, si nous générons un maillage de représentation régulier avec un nombre similaire d'éléments, à savoir 722 triangles, les représentations obtenus se trouvent sur les figures 4.3.b et 4.3.e et l'erreur de visualisation passe à 7.14 %. Pour atteindre la même précision de 1 % avec une subdivision uniforme, 6272 triangles sont nécessaires sur les figures 4.3 et 4.3.f au lieu des 700 dans le cas adaptatif. En comparant « grossièrement » les figures 4.3.a et 4.3.b, on peut penser

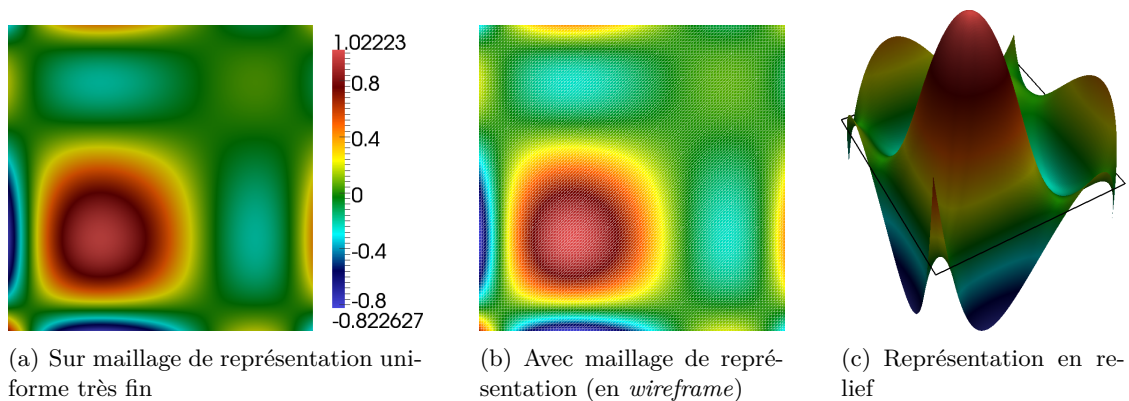


FIGURE 4.2 – Une fonction de base de type Gauss d’ordre 3 sur un élément carré.

qu’il n’y a pas beaucoup de différences. Cependant, en zoomant près des zones où les variations de la fonction sont importantes, on s’aperçoit sur les figures 4.4.a et 4.4.b que la subdivision régulière peut donner localement une information incorrecte.

La cartographie de l’erreur de visualisation se trouve sur la figure 4.5. Pour chaque élément du maillage de visualisation, l’écart entre la solution numérique et sa représentation est calculée. La figure 4.5.b, montre que l’erreur est localisée près de l’origine et que le maillage de représentation doit être raffiné dans cette zone. Aussi, le maillage de représentation obtenu par la méthode adaptative sur la figure 4.3.d est raffiné dans cette zone. La taille des triangles varie beaucoup en fonction de la richesse de la fonction à représenter. La figure 4.6 illustre les cartographies d’erreurs, comme la figure 4.5, mais une unique *colorbar* est utilisée, c’est à dire que la même étendue de valeur est utilisée afin de marquer les différences entre l’adaptatif et l’uniforme. Pour quasiment le même nombre d’éléments, la méthode adaptative permet d’avoir une erreur faible sur tout l’élément, contrairement à la subdivision régulière. La méthode auto-adaptative cherche à uniformiser l’erreur de visualisation.

Tous ces résultats sont regroupés dans le tableau 4.1 où d’autres précisions sont également investiguées. Pour la même fonction, la figure 4.7 montre les courbes de convergence avec la méthode adaptative et avec un découpage uniforme. Dans tous les cas, la méthode adaptative est plus optimale.

4.1.3 Comparaison du raffinement adaptatif avec un raffinement uniforme : Problème d’élasticité résolu par la méthode Galerkin Discontinue

Nous considérons maintenant un problème de propagation d’ondes élastiques dans un tri-milieu calculé par une méthode de Galerkin Discontinue [Dudouit, 2014]. Des fonctions de base de Gauss sont utilisées, celles-ci sont d’ordre 7. Le maillage de calcul est un maillage cartésien composé de 24×24 cellules (en magenta sur les figures qui suivent).

Avant de quantifier les erreurs de représentation, nous comparons pour un nombre d’éléments similaires les représentations obtenues avec les subdivisions régulières (gauche) et adaptatives (droite) sur la figure 4.8. Lorsque l’on regarde « de loin » les représentations obtenues sur les figures 4.8.a et 4.8.b, on ne voit pas forcément de différences et on pourrait même se demander

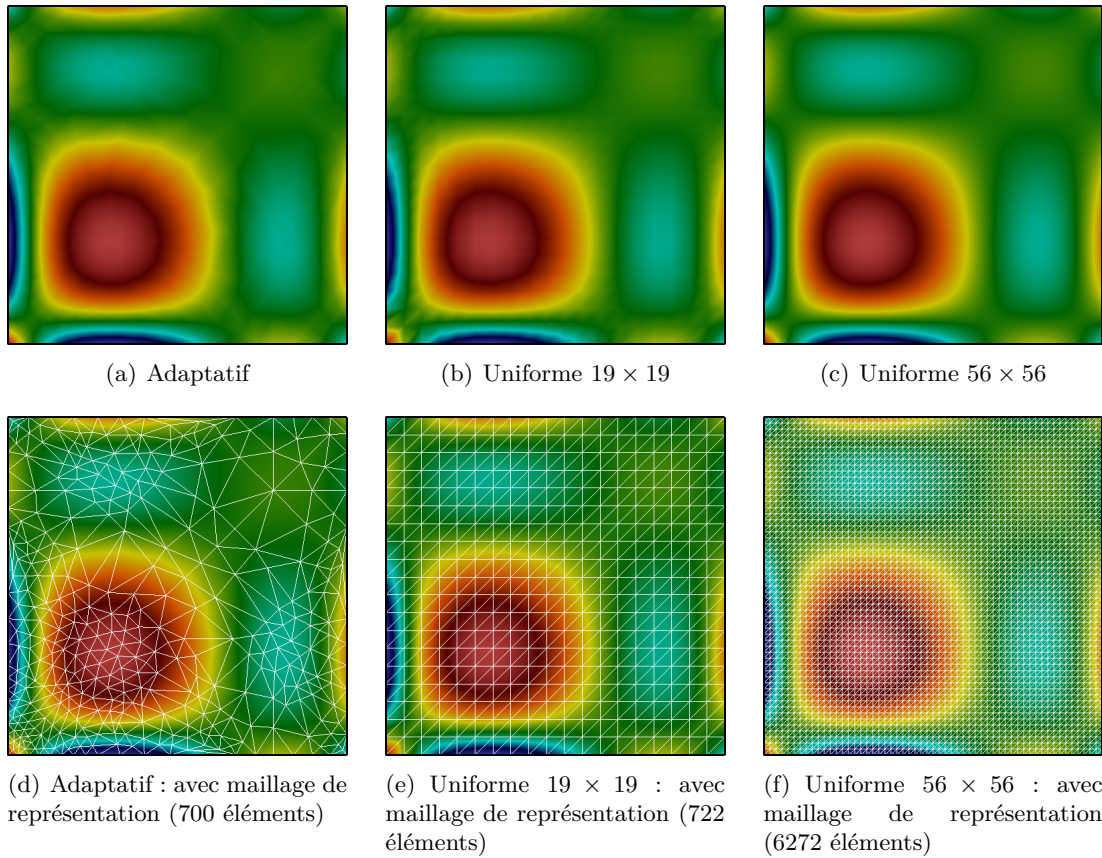


FIGURE 4.3 – Représentation d’une fonction de Gauss Q^3 avec la méthode adaptative (gauche), avec un découpage uniforme pour un nombre similaire d’éléments (milieu) et avec un découpage uniforme pour une précision similaire (droite). Le maillage de représentation est affiché en blanc sur la deuxième ligne.

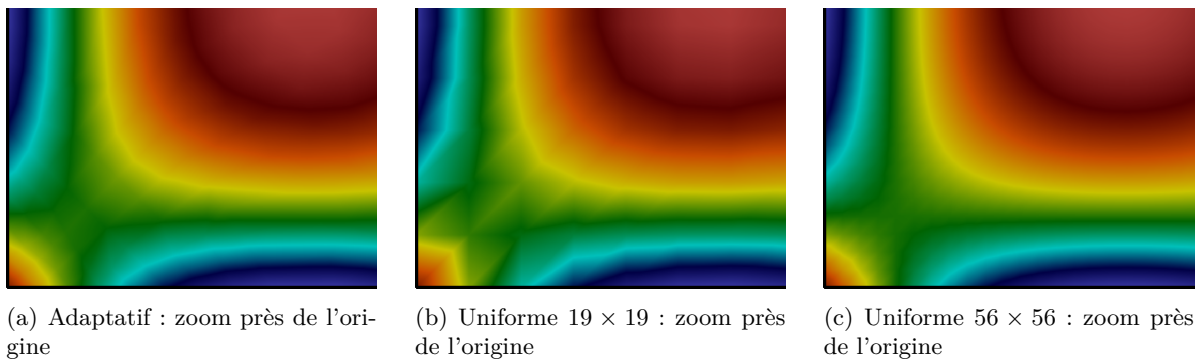


FIGURE 4.4 – Zoom près de l’origine de la figure 4.3 pour la représentation d’une fonction de Gauss Q^3 avec la méthode adaptative (gauche), avec un découpage uniforme pour un nombre similaire d’éléments (milieu) et avec un découpage uniforme pour une précision similaire (droite).

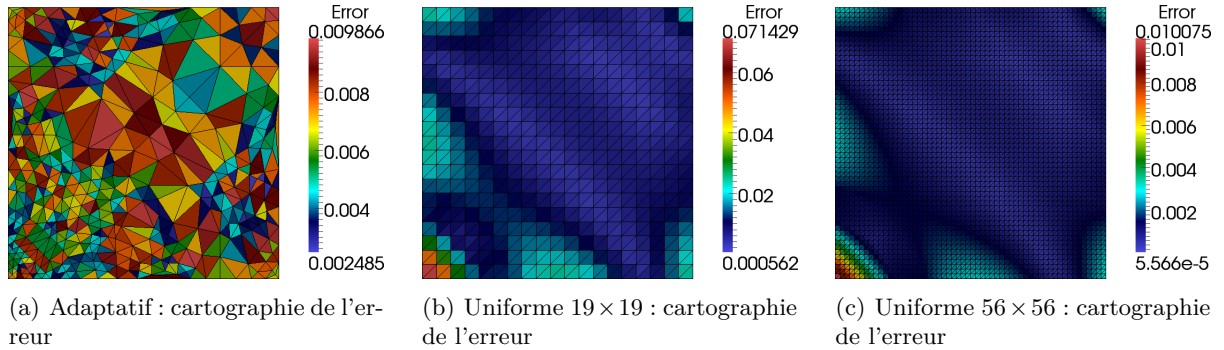


FIGURE 4.5 – Cartographie de l'erreur (voir figure 4.3) pour la représentation d'une fonction de Gauss Q^3 avec la méthode adaptative (gauche), avec un découpage uniforme pour un nombre similaire d'éléments (milieu) et avec un découpage uniforme pour une précision similaire (droite).

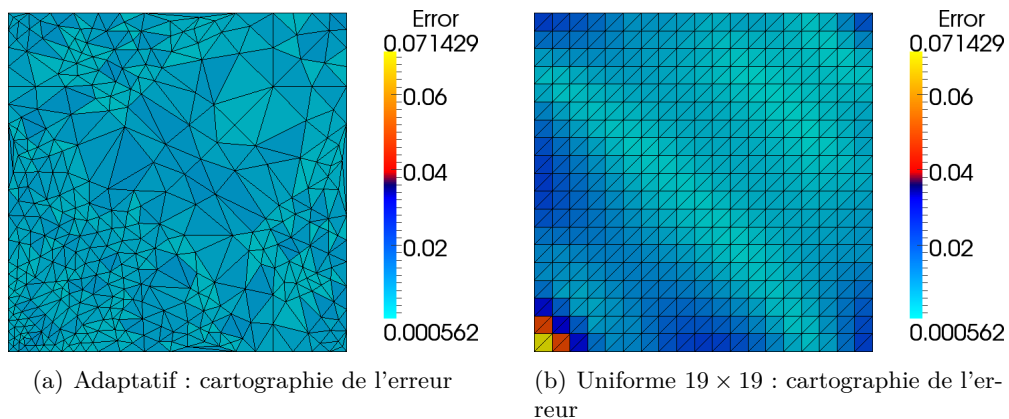


FIGURE 4.6 – Cartographie des erreurs de visualisation pour la fonction de gauss Q^3 entre les méthodes adaptatives et uniformes pour un nombre similaire d'éléments avec la même *colorbar* (celle du cas uniforme).

| Méthode utilisée | Nombre de triangles | Précision atteinte | Ratio (unif./adap.) |
|-------------------------------|---------------------|--------------------|---------------------|
| Adaptative (10) | 56 | 9.80 % | |
| Uniforme (6×6) | 72 | 34.37 % | 3.50 |
| Uniforme (15×15) | 450 | 10.53 % | 8.03 |
| Adaptative (1) | 700 | 0.99 % | |
| Uniforme (19×19) | 722 | 7.14 % | 7.21 |
| Uniforme (56×56) | 6272 | 1.01 % | 8.96 |
| Adaptative (0.1) | 7230 | 0.10 % | |
| Uniforme (60×60) | 7200 | 0.88 % | 8.80 |
| Uniforme (184×184) | 67712 | 0.10 % | 9.36 |

TABLE 4.1 – Comparaison de la méthode adaptative avec un raffinement uniforme pour une fonction de Gauss d'ordre 3.

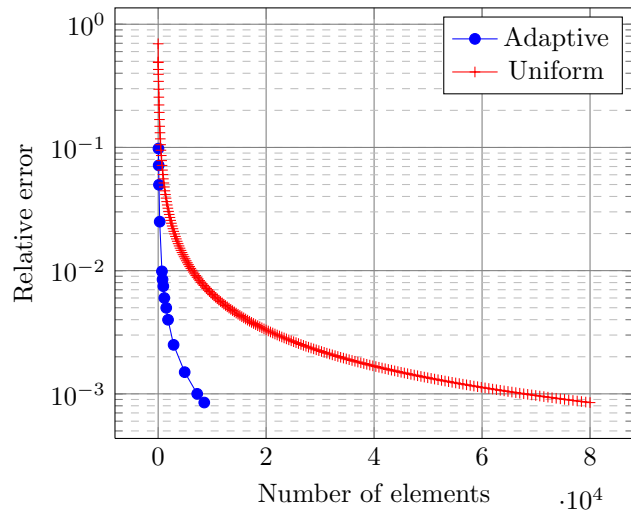
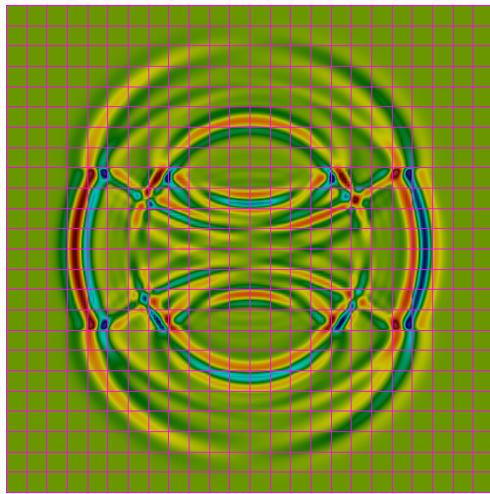
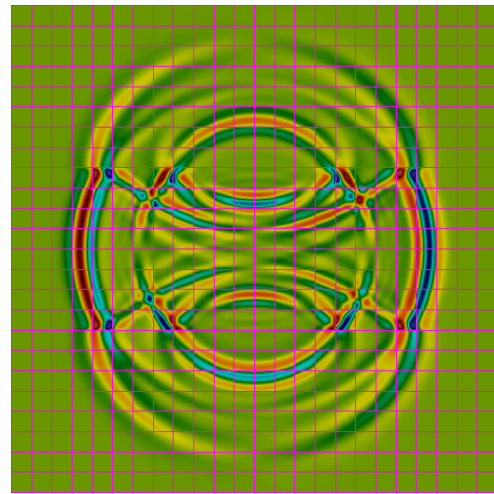


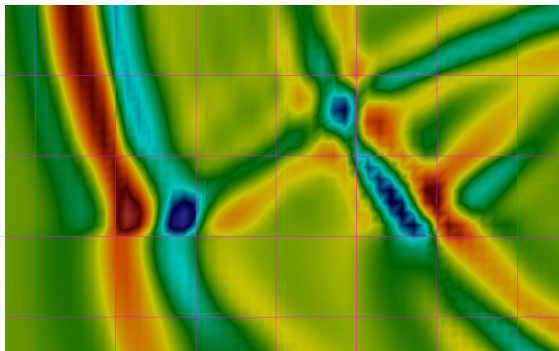
FIGURE 4.7 – Comparaison de la méthode adaptative avec un découpage uniforme pour une fonction de base Gauss d'ordre 3.



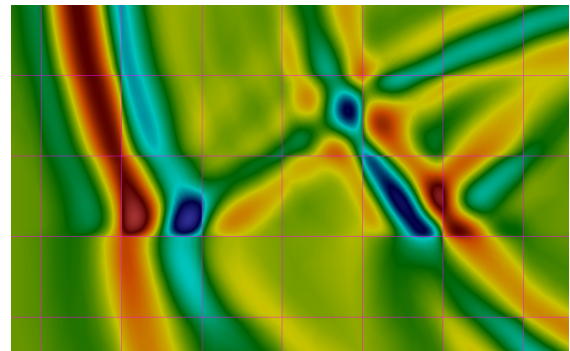
(a) Avec maillage de représentation uniforme 7×7 : 56448 éléments



(b) Avec maillage de représentation adaptatif (0.5 %) : 52108 éléments



(c) Un zoom de (a)



(d) Un zoom de (b)

FIGURE 4.8 – Propagation d'une onde élastique dans un tri-milieu calculée par une méthode Galerkin Discontinue Q^7 sur un maillage uniforme 24×24 . Représentation à 0.5 %

s'il y a un réel intérêt d'avoir un maillage de représentation adaptatif. Néanmoins, lorsque l'on regarde plus précisément avec les figures 4.8.c et 4.8.d, les défauts d'une subdivision régulière apparaissent clairement visuellement. Nous allons maintenant quantifier ces différences.

Sur la figure 4.9, plusieurs représentations de la solution numérique au temps final sont affichées. Sur la figure 4.9.a, la méthode adaptative est utilisée sous la tolérance fixée à 0.5 %. Le maillage de représentation y est affiché en blanc. Celui-ci est composé de 52108 éléments, ce qui implique une moyenne de 90 éléments par cellule. Nous nous intéressons en particulier à une petite zone qui est mise en évidence en pointillés noirs sur la figure 4.9.a. Cette zone est intéressante car la solution y est riche. En effet, le changement de milieu physique implique des recombinaisons d'ondes. Des zooms sont donc effectués sur les figures 4.9.b et 4.9.c où une subdivision uniforme est comparée avec notre méthode. Une méthode de Galerkin Discontinue étant utilisée, des discontinuités, correspondant aux sauts numériques sont affichées. Comme demandé par l'objectif \mathbf{O}_3 , la restitution de ces sauts est bien effectuée sur la figure 4.9.c.

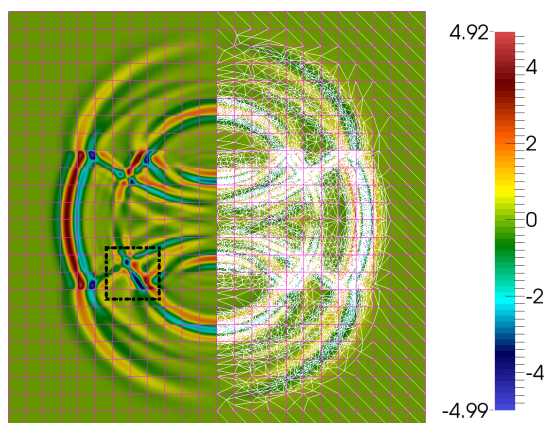
Sur la figure 4.9.b, chaque cellule du maillage de calcul est redécoupée de manière uniforme en 7×7 (correspondant à 98 triangles). Ce choix de découpage est fixé *a priori* et est motivé par le fait que l'ordre des fonctions est 7. Choisir comment redécouper de manière uniforme est délicat et dépend de la solution à représenter. Une « règle » choisie pour cet exemple est de calibrer le découpage en fonction de l'ordre des fonctions de base. Cependant, dans la plupart des cas, soit la représentation n'est pas assez précise, soit beaucoup de découpages sont nécessaires, ce qui implique de nombreux calculs et interpolations ainsi que des coûts en mémoire élevés. Dans tous les cas, comme la figure 4.9 le montre, même si sur chaque cellule du maillage de calcul, l'ordre est constant, la richesse de la solution peut varier énormément d'une cellule à l'autre. Ainsi, un maillage de représentation non-uniforme est nécessaire. Avec la subdivision fixée en 7×7 , l'erreur de visualisation est de 10,5 % en norme L^∞ . La représentation obtenue n'est donc pas précise.

De plus, tous les post-traitements, comme les sorties de lignes de niveau ou encore les coupes, qui sont effectués à partir de cette « mauvaise » représentation sont évidemment également erronés. En particulier, sur les figures 4.9.b et 4.9.c, dix lignes de niveau (*isolines*) sont affichées. On voit clairement que celles obtenues par le raffinement uniforme ne sont pas précises.

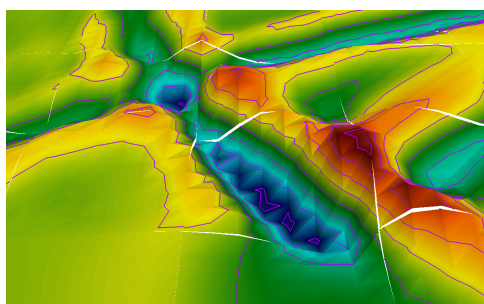
Nous nous intéressons maintenant à une cellule du maillage de calcul, noté \mathcal{C} , qui se trouve au milieu de la zone en pointillé noire. Cette dernière est particulièrement intéressante car elle contient beaucoup d'informations en raison du changement de milieu. Sur la figure 4.9.d, un zoom sur cette cellule est effectué où un raffinement uniforme 7×7 a été utilisée pour générer une grille de visualisation. L'erreur de visualisation dépasse les 10 %. Pour avoir une précision similaire, avec la méthode adaptative, seulement 23 éléments sont nécessaires (figure 4.9.e). Sur la figure 4.9.f, le même nombre global d'éléments est utilisé avec la méthode adaptative. La précision atteinte est inférieure à 0.5 %.

Le tableau 4.2 résume toutes ces informations. Le nombre d'éléments nécessaires pour avoir une précision similaire est comparé pour plusieurs cas. Le nombre total d'éléments pour avoir 10 % (respectivement 1 %) de précision est 25 (respectivement 33) fois plus grand avec une subdivision uniforme qu'avec la méthode adaptative.

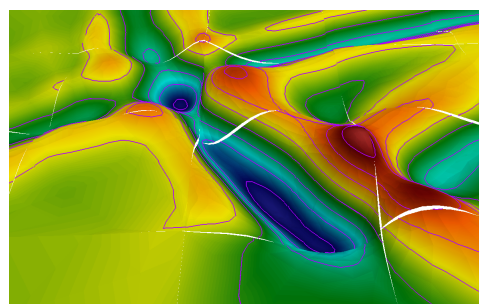
On comprend facilement que la méthode adaptative est nécessaire puisque sur certains éléments du domaine de calcul, il n'y a besoin de peu d'éléments dans le maillage de représentation tandis que pour d'autres éléments, les maillages de représentation sont raffinés afin de capter toute la richesse locale de la solution.



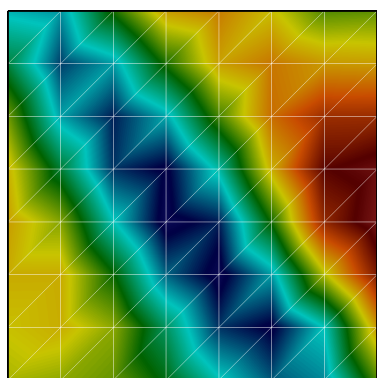
(a) Représentation de la solution numérique avec la méthode adaptative (0.5 %).



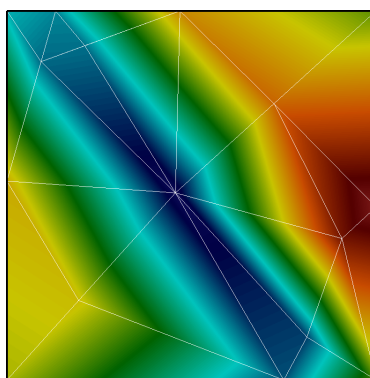
(b) Zoom sur la zone en pointillé noire de (a), subdivision uniforme 7×7 , 56448 éléments



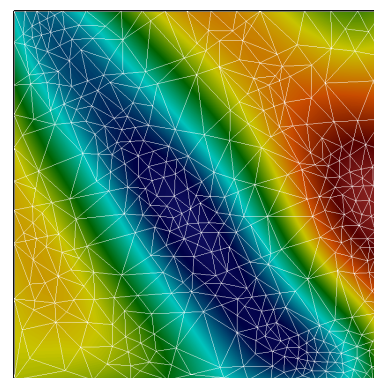
(c) Zoom sur la zone en pointillé noire de (a), méthode adaptative (0.5 %), 52108 éléments



(d) Focus sur la cellule centrale de la zone en pointillé noire, notée \mathcal{C} , uniforme 7×7 , 98 éléments



(e) Focus sur la cellule \mathcal{C} , adaptative 10 %, 23 éléments



(f) Focus sur la cellule \mathcal{C} , adaptative 0.5 %, 818 éléments

FIGURE 4.9 – Propagation d'une onde élastique dans un tri-milieu calculée par une méthode Galerkin Discontinue Q^7 sur un maillage uniforme 24×24 .

| Méthode | Local (cellule \mathcal{C}) | | Global (576 cellules) | |
|-------------------------|--------------------------------|----------------|-------------------------------------|----------------|
| | Nb. d'éléments | Préc. atteinte | Nb. d'éléments | Préc. atteinte |
| Adaptative (10) | 23 | 9.93 % | 2194 ($\approx 4/\text{cell.}$) | 9.93 % |
| Uniforme 7×7 | 98 | 10,4 % | 56448 (98/cell.) | 10,5 % |
| Adaptative (1) | 416 | 1.00 % | 24856 ($\approx 43/\text{cell.}$) | 1.00 % |
| Uniforme 27×27 | 1458 | 1.03 % | 839808 (1458/cell.) | 1.03 % |

TABLE 4.2 – Comparaison du nombre d'éléments des maillages de représentation générés avec une subdivision uniforme et avec la méthode adaptative pour plusieurs précisions sur une cellule du domaine (local, cellule \mathcal{C}) et sur le domaine en entier (global).

4.1.4 Exemples volumiques : modes sur hexaèdres.

Dans cette section, nous traitons deux exemples analytiques volumiques afin de montrer que notre méthode de visualisation permet de traiter des éléments de dimension 3. Les éléments géométriques considérés sont des hexaèdres simples, définis par 8 sommets.

Le premier mode est défini de la façon suivante

$$f(r, s, t) = \sin(2\pi r C_0) \sin(2\pi s C_1), \quad (4.1)$$

où C_0 et C_1 sont deux constantes.

Deux éléments hexaédriques composent le maillage. Le premier élément est le l'hexaèdre unité. Pour ce premier élément, les constantes sont fixées à $C_0 = C_1 = 2$. Le second élément est de même taille (longueurs des arêtes égales à un) et est son voisin. Pour ce dernier, les constantes sont fixées à $C_0 = C_1 = 3$.

Ce mode est représenté sur la figure 4.10 avec notre méthode de visualisation à 1 %. Le maillage de représentation est constitué de 27195 tétraèdres pour ces deux éléments. La figure 4.10.a montre le rendu obtenu. Les extrema de la fonction (-1 et 1) sont bien restitués. La fonction est nulle sur plusieurs faces étant donné que des sinusoides définissent la fonction. Cependant, la fonction n'est pas nulle à l'intérieur comme le montre la coupe effectuée sur la figure 4.10.b. Le plan de coupe, effectué sous ParaView, est affiché en rouge. Deux courbes de niveau sont extraites sur la figure 4.10.c correspondant aux valeurs -0.5 et 0.5. Des cylindres sont obtenus comme la fonction ne varie pas suivant la composante z (voir équation (4.1)). Le maillage de représentation est exhibé sur la figure 4.10.d. Le côté anisotrope est bien visible puisqu'il n'y a pas de points ajoutés à l'intérieur des deux hexaèdres comme le montre la figure 4.10.e. Cela montre la pertinence de l'algorithme (estimateur d'erreur et choix du point).

Nous considérons maintenant le mode suivant

$$f(r, s, t) = \sin(2\pi r C_0) \sin(2\pi s C_1) \sin(2\pi t C_2), \quad (4.2)$$

où C_0 , C_1 et C_2 sont des constantes.

Nous considérons un seul élément, l'hexaèdre unité et nous fixons les constantes comme $C_0 = C_1 = C_2 = 1$. Ce mode est représenté avec notre méthode à 0.5% sur la figure 4.11.

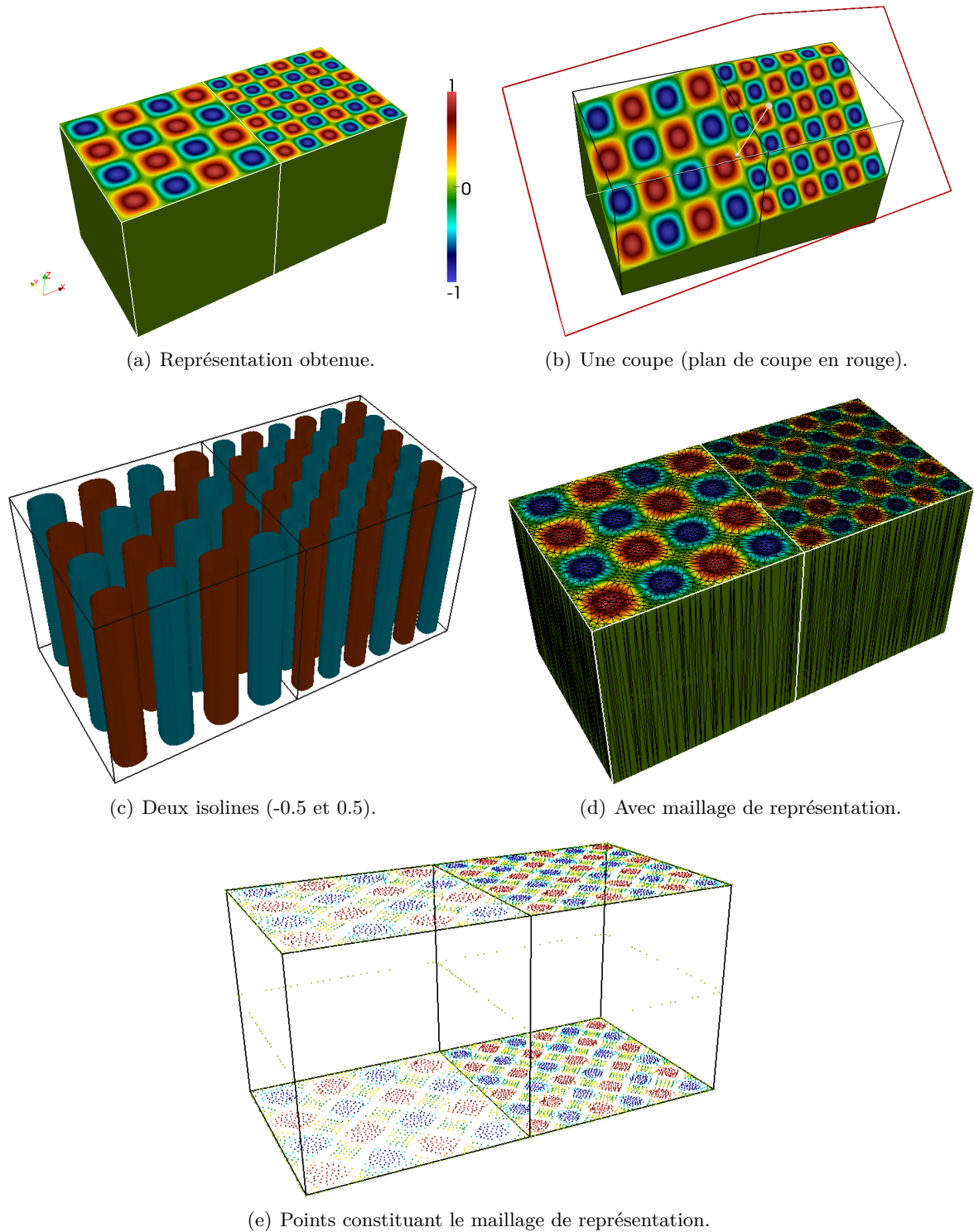


FIGURE 4.10 – Représentation d’un mode plan donné par (4.1) sur des hexaèdres avec notre méthode de visualisation à 1%.

Le maillage de représentation est composé de 140 546 tétraèdres. Sur la figure 4.11.a, nous avons effectué une coupe dans le logiciel ParaView afin de montrer le mode à l'intérieur de l'élément. Sur la figure 4.11.b, nous avons représenté deux isosurfaces (-0.5 et 0.5). Comme les trois constantes de (4.2) sont identiques, ces surfaces sont des boules de mêmes rayons. Par rapport au mode précédent, il y a naturellement beaucoup plus de tétraèdres car le maillage de représentation est cette fois-ci aussi composé de points à l'intérieur de l'hexaèdre.

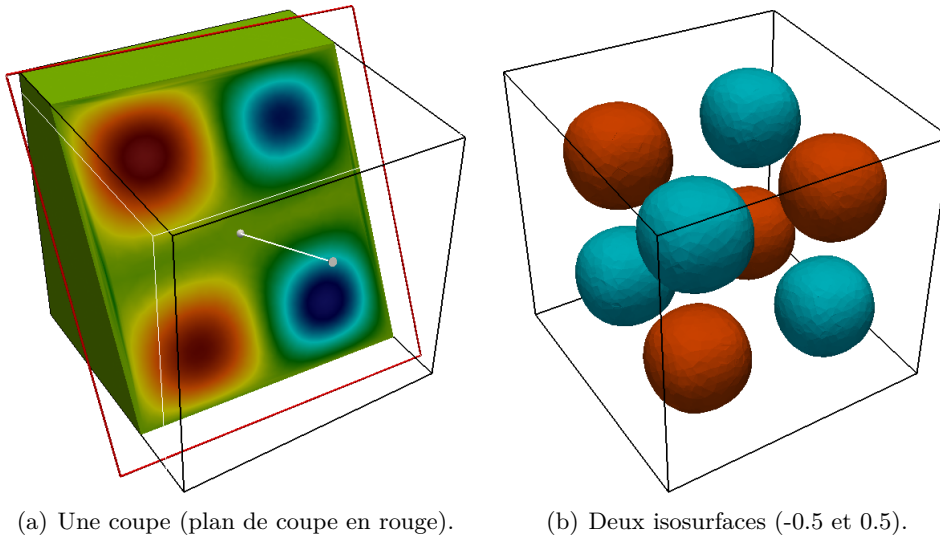


FIGURE 4.11 – Représentation d'un mode donné par (4.2) sur des hexaèdres avec notre méthode de visualisation à 0.5%.

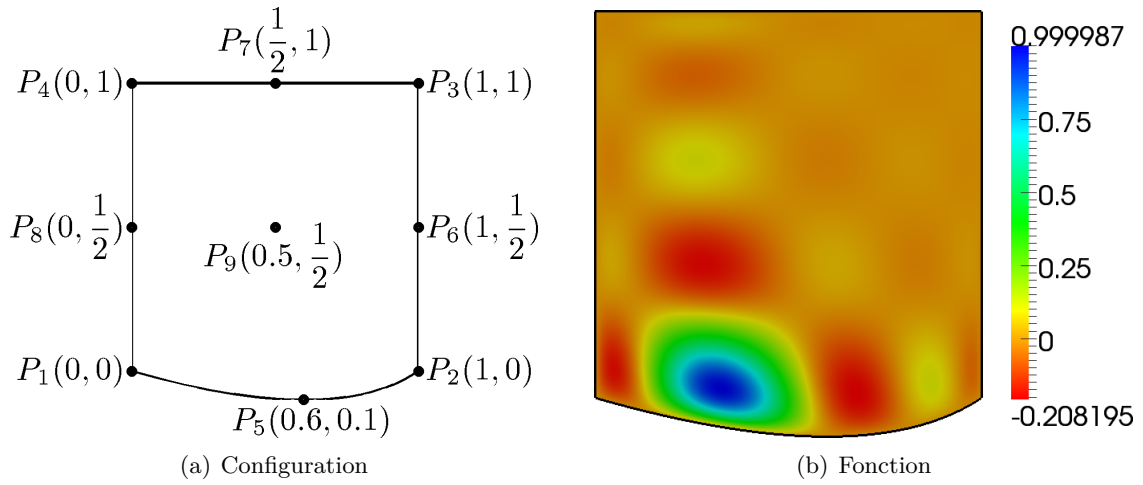
4.2 Validations pour le traitement des éléments courbes

Dans le chapitre 3, nous avons proposé une méthodologie pour traiter les éléments courbes. Nous avons vu que si l'indicateur d'erreurs défini dans le chapitre 2 est utilisé tel quel, la représentation obtenue peut faire apparaître des recouvrements. Pour éviter cela, un estimateur spécifique a été développé pour éliminer ces problèmes. Dans cette section, nous souhaitons donc montrer sur des exemples numériques les gains obtenus par ce dernier.

4.2.1 Fonction de base Gauss-Lobatto Q^6 sur un quadrangle Q^2

Nous nous intéressons à un cas très simple puisque qu'une seule cellule dans le domaine de calcul est considérée. La cellule en question est un quadrangle Q^2 qui est donc définie par 9 nœuds et dont la transformation géométrique g_K est d'ordre 2. La configuration de l'élément est donné sur la figure 4.12.a. Nous considérons une fonction de Gauss-Lobatto d'ordre 6 définie sur cet élément : φ_{32}^{GL6} où les notations précédentes sont reprises. Ceci signifie que la fonction est nulle sur le bord de l'élément car elle est construite à partir d'un degré de liberté intérieur. Cette fonction est donc nulle à tous les autres degrés de liberté. Cette fonction est visible sur la figure 4.12 sur un maillage de représentation très fin.

Sur la figure 4.13, les représentations obtenues à l'aide des deux estimateurs à savoir l'estimateur classique Δ_S^K (défini à partir de (3.8)) et l'estimateur spécifique Γ_S^K (défini à partir de

FIGURE 4.12 – Une fonction de base de type Gauss-Lobatto d’ordre 6 sur un quadrangle Q^2 .

(3.9)) sont comparées. Les maillages de représentation sont affichés ainsi que le maillage du bord qui est épaissi. La particularité de la fonction considérée est qu’elle est constante sur le bord.

Ceci signifie que lors du traitement des arêtes, si l’estimateur classique (pour les éléments droits) est utilisé, l’erreur due aux fonctions est nulle sur le bord et il n’y a donc que l’erreur due à l’approximation de la géométrie à contrôler. Notons qu’il y a trois arêtes où il n’y a pas besoin d’ajouter de points. En effet, l’erreur de visualisation y est nulle car ce sont des segments. Concernant la quatrième arête, qui est un segment sur la cellule de référence mais est courbe sur l’élément réel, certains points sont ajoutés sur le maillage de représentation afin d’approcher la géométrie à la précision demandée. Le nombre de segments sur chaque arête est répertorié dans le tableau 4.3. Lorsque l’indicateur classique (3.8) est utilisé, comme uniquement le bord est pris en compte, une zone de recouvrement apparaît sur les figures 4.13.a et 4.13.c. Ceci vient du fait que certains points du maillage 2D sont à l’extérieur de la cellule construite à partir de cette approximation du bord.

| Estimateur | Arête 1 | Arête 2 | Arête 3 | Arête 4 | Quadrangle Q^2 |
|-------------------------|---------|---------|---------|---------|------------------|
| Classique Δ_S^K | 1 | 2 | 1 | 1 | 221 |
| Spécifique Γ_S^K | 1 | 11 | 1 | 1 | 216 |

TABLE 4.3 – Nombre d’éléments sur les maillages de représentation de la figure 4.13 pour chaque arête et sur le quadrangle Q^2 .

Cependant, lorsque l’indicateur spécifique (3.9) est utilisé, celui-ci prend en compte les variations des fonctions au voisinage de chaque arête et ne se limite pas simplement à l’arête en question. La conséquence, comme le montre le tableau 4.3, est qu’un raffinement du maillage du bord est réalisé afin d’éviter les zones de recouvrements pour la précision cible. Ainsi, pour l’arête qui posait problème, le maillage de représentation passe de 3 à 12 points (c’est à dire de 2 à 11 segments). Par conséquent, il n’y a plus de points qui sortent (figures 4.13.b et 4.13.d) lors du traitement de l’intérieur. Ainsi, il n’y a plus de recouvrement. Il est important de noter que dans les deux cas, le nombre d’éléments est similaire et qu’avec

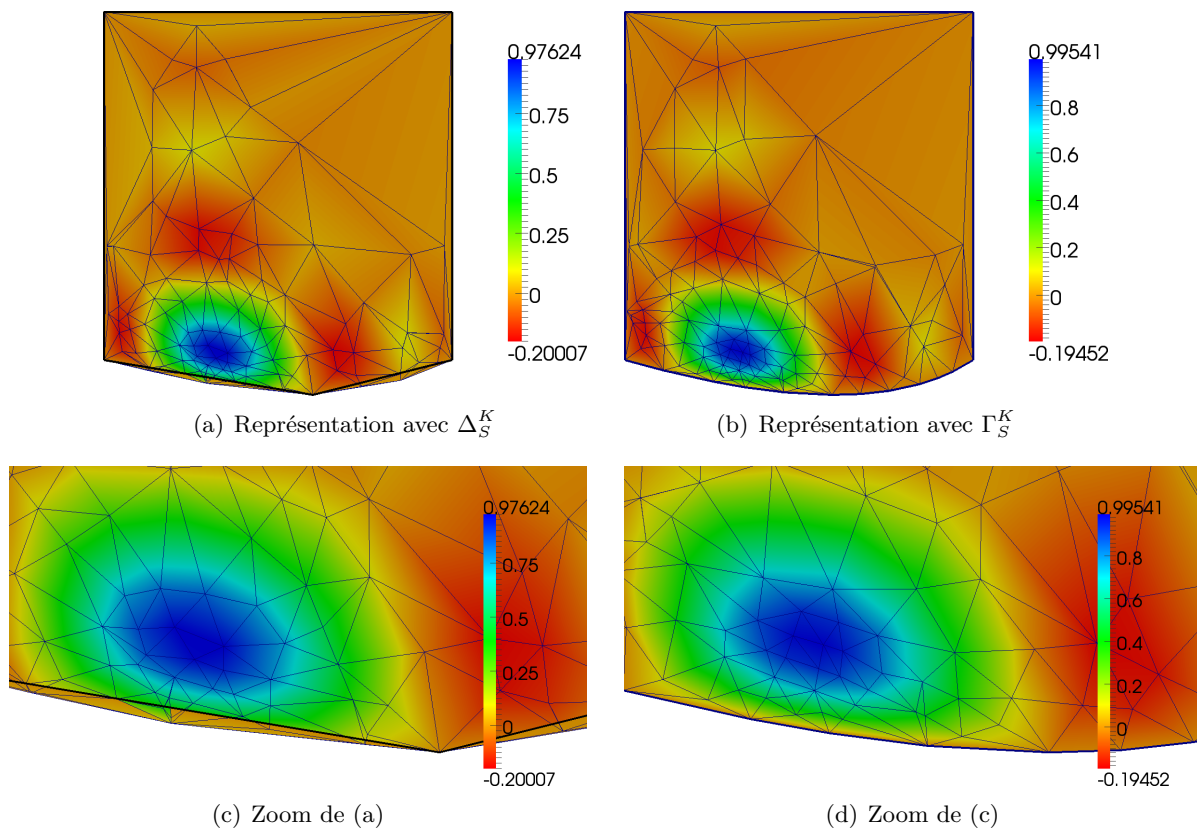


FIGURE 4.13 – Représentation à une précision de 3.5 % avec les deux indicateurs Δ_S^K (a et c) and Γ_S^K (b et d) d'une fonction de base Gauss-Lobatto Q^6 sur un quadrangle Q^2 .

l'estimateur spécifique, il y a même dans ce cas moins d'éléments. Cela signifie que même s'il y a un raffinement du bord, cela n'implique pas de surcoût sur le maillage de l'intérieur de l'élément.

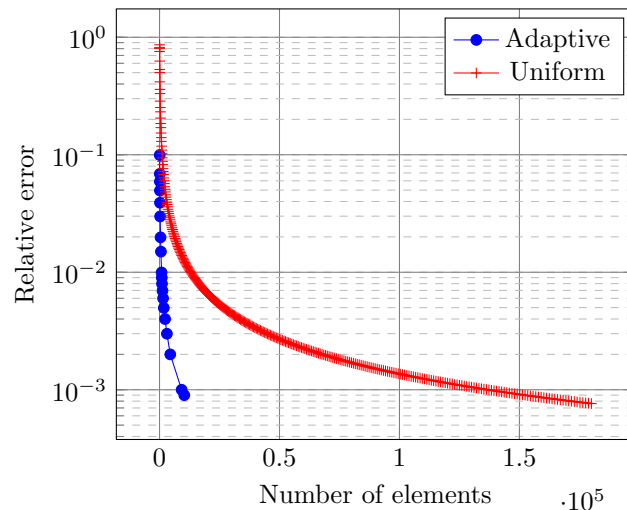


FIGURE 4.14 – Comparaison de la méthode adaptative avec un découpage uniforme pour une fonction de base Gauss d'ordre 6 sur un quadrangle Q^2 .

Enfin, à l'instar de ce que nous avons fait pour des éléments droits, nous comparons la méthode adaptative avec un raffinement uniforme sur la figure 4.14. Le tableau 4.4 répertorie quelques valeurs. Les ratios entre les méthodes uniforme et adaptative sont également précisés. A savoir, pour une précision donnée, le rapport entre le nombre d'éléments générés pour les deux méthodes, et vice et versa. Ces ratios augmentent avec l'augmentation de la taille du problème. De plus, par rapport à la section 4.1.2 où une fonction Q^3 sur un élément Q^1 était considérée, le fait d'être passé à une fonction Q^6 sur un élément Q^2 a également fait augmenter ces ratios.

4.2.2 Retournements dans l'espace

Nous nous intéressons maintenant à la représentation d'un élément courbe dans l'espace et nous illustrons les retournements qui peuvent apparaître. L'élément considéré est très proche du précédent, c'est un quadrangle Q^2 . La seule différence est que pour un point la troisième composante n'est pas nulle, tandis que pour tous les autres points de contrôle, cette troisième composante est nulle. La configuration est ainsi très proche de la figure 4.12.a, la seule différence est que la composante du cinquième point est fixée à 0.2. La fonction à représenter est la même que précédemment, à savoir φ_{32}^{GL6} .

La figure 4.15 compare les rendus obtenus avec l'estimateur Δ_S^K qui ne prend en compte que le bord lors du maillage de la frontière des éléments et Γ_S^K qui prend en compte le voisinage du bord. Les figures 4.15.a et 4.15.b sont des vues de dessus qui sont assez proches de la figure 4.13. Cette-fois-ci, il n'y a pas de recouvrement qui apparaît, ni de perte de bijectivité de l'approximation de la transformation géométrique, mais des retournements apparaissent sur la figure 4.15.c. Lorsque l'estimateur Γ_S^K est utilisé, il n'y a plus de retournements.

| Méthode utilisée | Nombre de triangles | Précision atteinte | Ratio (unif./adap.) |
|-------------------------------|---------------------|--------------------|---------------------|
| Adaptative (10) | 76 | 9.94% | 6.31 |
| Uniforme (6×6) | 72 | 62.68 % | |
| Uniforme (23×23) | 1058 | 10.22 % | |
| Adaptative (1) | 850 | 1.00 % | 11.83 |
| Uniforme (21×21) | 882 | 11.83 % | |
| Uniforme (80×80) | 12800 | 1.02 % | |
| Adaptative (0.1) | 9200 | 0.10 % | 13.90 |
| Uniforme (68×68) | 9248 | 1.39 % | |
| Uniforme (261×261) | 136242 | 0.10 % | |

TABLE 4.4 – Comparaison de la méthode adaptative avec un raffinement uniforme pour une fonction de Gauss d'ordre 6 sur un quadrangle Q^2 .

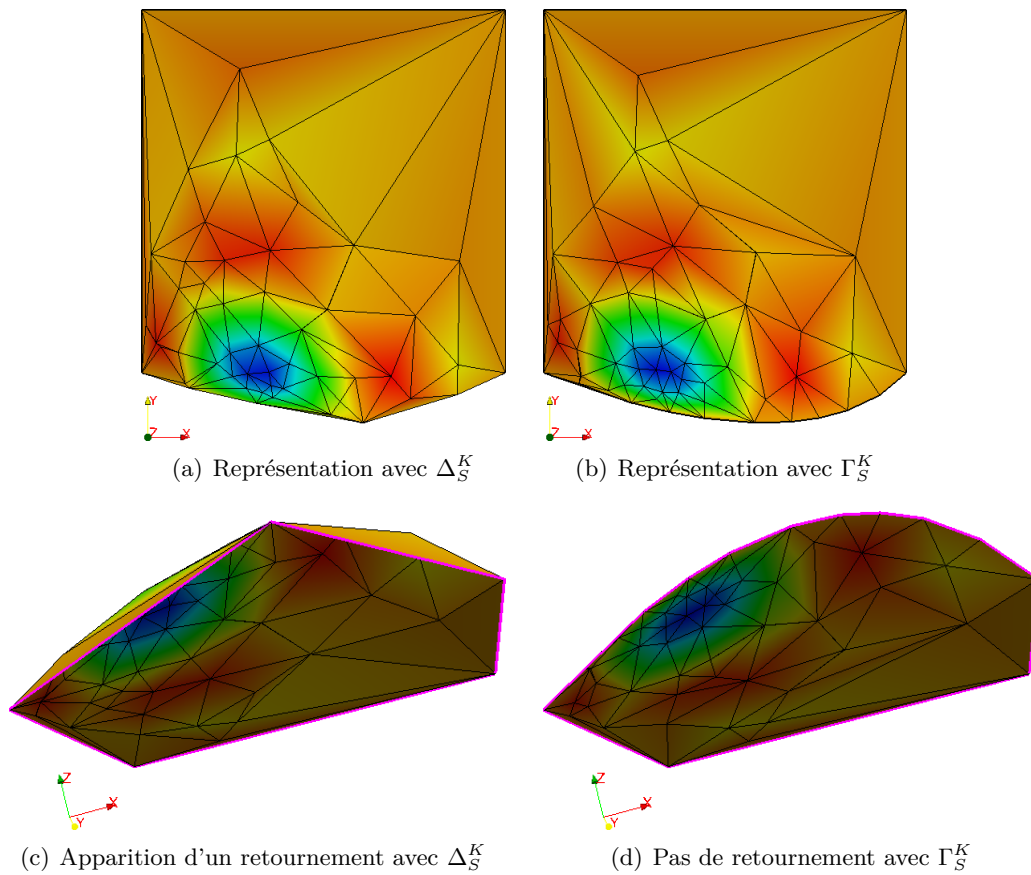


FIGURE 4.15 – Représentation à une précision de 7 % avec les deux indicateurs Δ_S^K (a et c) and Γ_S^K (b et d) d'une fonction de base Gauss-Lobatto Q^6 sur un quadrangle Q^2 dans l'espace.

Le tableau 4.5 récapitule le nombre d'éléments sur les maillages de représentation de chaque arête. Comme précédemment, l'utilisation de l'estimateur spécifique permet de raffiner le maillage de l'arête courbe qui passe de 2 à 9 éléments. Le bord de l'élément approché est affiché en magenta sur la figure 4.15. L'estimateur spécifique permet ainsi d'avoir une meilleure approximation du bord et surtout élimine les artefacts visuels. Notons enfin que le nombre d'élément dans le maillage de représentation sur la surface est similaire dans les deux cas : 101 et 104 (le maillage de l'intérieur n'est pas modifié).

| Estimateur | Arête 1 | Arête 2 | Arête 3 | Arête 4 | Quadrangle Q^2 |
|-------------------------|---------|---------|---------|---------|------------------|
| Classique Δ_S^K | 2 | 1 | 1 | 1 | 101 |
| Spécifique Γ_S^K | 9 | 1 | 1 | 1 | 104 |

TABLE 4.5 – Nombre d'éléments sur les maillages de représentation de la figure 4.15 pour chaque arête et sur le quadrangle Q^2 .

4.2.3 Problème de propagation d'onde acoustique.

Nous nous intéressons maintenant à un exemple numérique traitant la propagation d'une onde acoustique. Nous considérons une formulation basée sur le déplacement en dimension deux. Le système à résoudre est le suivant

$$\left\{ \begin{array}{ll} \rho(x) \frac{\partial^2 u}{\partial t^2}(x, t) - \nabla (\mu(x) \nabla \cdot u(x, t)) = f(x, t), & \text{dans } \Omega \times [0, T], \\ u(x, t) \cdot n = 0, & \text{sur } \Gamma \times [0, T], \\ u(x, 0) = 0, & \text{dans } \Omega, \\ \frac{\partial u}{\partial t}(x, 0) = 0, & \text{dans } \Omega, \end{array} \right. \quad (4.3)$$

où Ω est le domaine de calcul borné, Γ est sa frontière, n une normale à la frontière et T le temps final. Des notations classiques sont introduites : ∇ désigne le gradient et $\nabla \cdot$ la divergence. Le vecteur déplacement est noté u . Les paramètres ρ et μ dépendent du milieu considéré.

Le domaine de calcul considéré est visible sur la figure 4.16. Des éléments courbes sont notamment introduits pour approcher la géométrie. Pour cela, des éléments finis Raviart-Thomas sont introduits sur un maillage hybride. Des éléments triangulaires courbes d'ordre 2 (i.e. définis par 6 points) sont utilisés pour avoir une bonne approximation de la géométrie et des quadrangles d'ordre 1 sont utilisés lorsqu'un maillage Cartésien est possible. Le champ de déplacement est approximé par l'espace éléments finis suivant :

$$V_h^p = \{v_h \in H(\text{div}, \Omega) : \forall K \in \mathcal{T}(X), \mathcal{P}_K^{-1} v_{h|_K} \circ g_K \in RT^p(\widehat{K})\}, \quad (4.4)$$

où p est l'ordre des éléments de Raviart-Thomas utilisé et \mathcal{P}_K désigne la transformation de Piola [Ervin, 2012] donnée par

$$\mathcal{P}_K = \frac{Jg_K}{|Jg_K|},$$

avec Jg_K est la jacobienne de la transformation géométrique g_K et $|Jg_K|$ son déterminant pour un élément K donné. En fonction de la nature des éléments, deux cas doivent être considérés.

Les triangles sont définis par 6 nœuds tels que :

$$g_K \in P^2(\widehat{K})^2 \quad \text{et} \quad RT^p(\widehat{K}) = P^p(\widehat{K})^2 + xP^p(\widehat{K})$$

comme dans [Ervin, 2012] et sont présentés en annexe B.3.

Pour les quadrangles, la cellule de référence \widehat{K} est le carré unité et

$$g_K \in Q^1(\widehat{K})^2, \quad \text{et} \quad RT^p(\widehat{K}) = Q^{p+1,p}(\widehat{K}) \times Q^{p,p+1}(\widehat{K})$$

comme dans [Cohen and Pernet, 2016], où

$$Q^{\ell,m}(\widehat{K}) = P^{\ell}([0,1]) \otimes P^m([0,1])$$

et \otimes désigne le produit tensoriel.

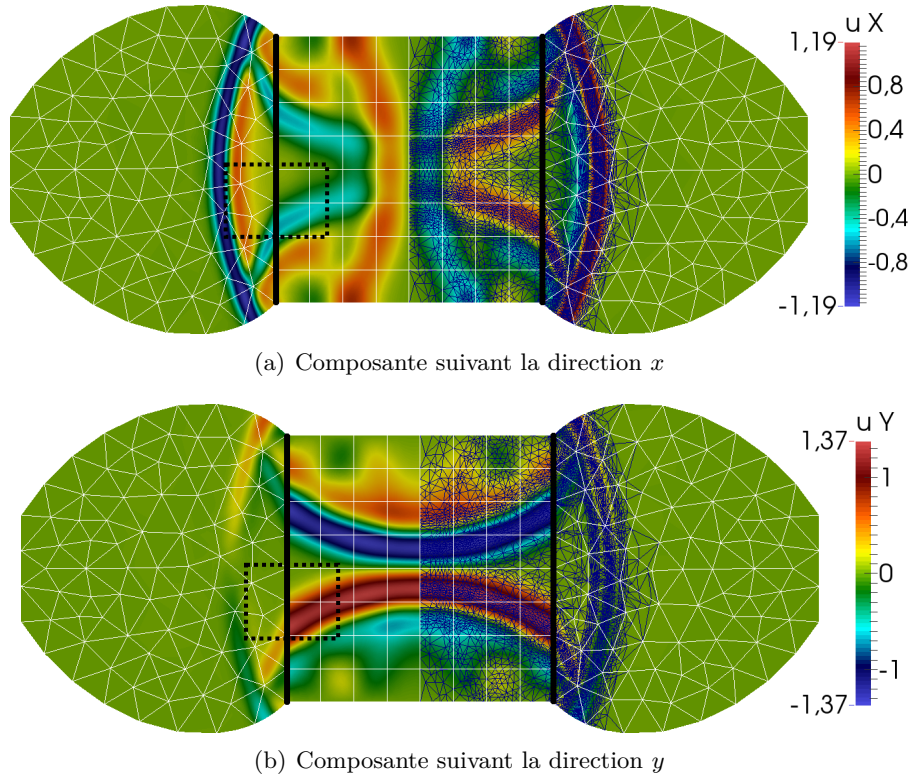


FIGURE 4.16 – Représentation de la solution de propagation d'une onde acoustique à un temps T .

Le maillage de calcul utilisé pour la figure 4.16 est composé de 64 quadrangles et de 306 triangles. L'ordre choisi sur chaque élément est $p = 5$. Le domaine est constitué de deux milieux. Le premier milieu est composé des triangles et ses paramètres sont les suivants : $\rho_1 = 4$ and $\mu_1 = 1$. Le second milieu est composé des quadrangles tel que $\rho_2 = 1$ et $\mu_2 = 1$. Comme $\rho_1 \neq \rho_2$, il y a une discontinuité physique due à ce changement de matériaux à l'interface entre les triangles et les quadrangles. Cependant, tout le champ n'est pas discontinu. En effet, seule la composante tangentielle l'est tandis que la composante normale reste continue à la

4.3. AMÉLIORATION DE LA QUALITÉ DES ÉLÉMENTS EN CONSIDÉRANT LE VOISINAGE DE LA FRONTIÈRE

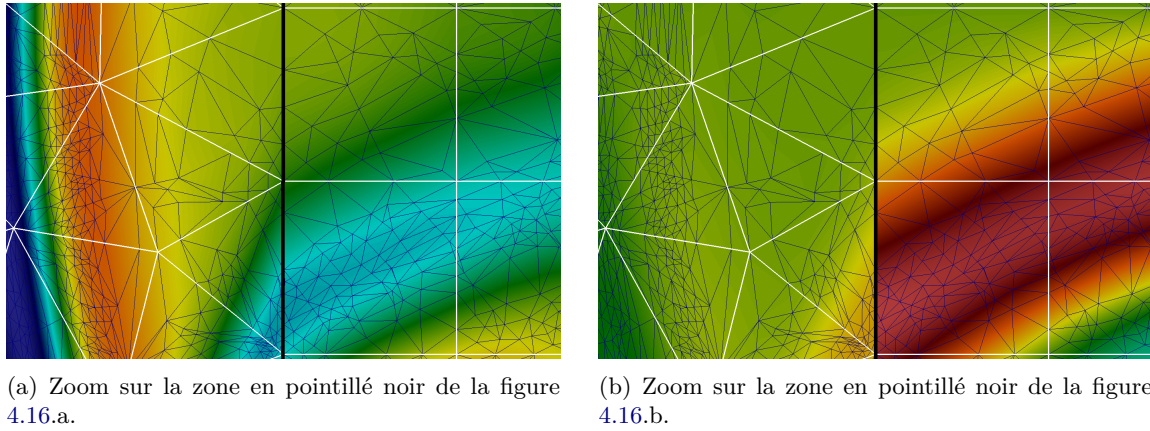


FIGURE 4.17 – Représentation de la solution de propagation d’une onde acoustique. Zoom de la figure 4.16.

traversée de l’interface. Les éléments fins de Raviart-Thomas sont justement choisis pour naturellement conserver cette continuité normale, tout en autorisant une discontinuité tangentielle.

Deux segments noirs épais montrent cette interface sur la figure 4.16. Comme cette interface est parallèle à l’axe- y , la composante- x est continue alors que la composante- y est discontinue. Il est important de noter qu’un seul maillage de représentation est généré. Ainsi, chaque composante de la solution a le même maillage de représentation. Aussi, lors de la création de ce dernier, chaque composante est prise en compte (accumulation de « contraintes »).

La figure 4.17 est un zoom d’une zone de la figure 4.16 qui est particulièrement intéressante puisqu’elle se trouve à la limite entre les deux milieux. Chaque arête entre deux éléments a un maillage de représentation unique. Ceci permet de représenter avec précision les sauts physiques sur la composante- y et les continuités sur la composante- x comme demandé par l’objectif \mathbf{O}_3 .

4.3 Amélioration de la qualité des éléments en considérant le voisinage de la frontière

Dans cette section, nous nous intéressons à un point permettant d’améliorer, pour certaines configurations, la qualité de la représentation obtenue.

Nous avons vu, notamment lors de l’utilisation d’éléments courbes, que le maillage par dimensions croissantes et le fait de ne plus pouvoir modifier le maillage du bord est une grosse contrainte. S’il y a peu d’éléments sur le maillage du bord, il est possible que des éléments *plats*, c’est à dire dont plusieurs angles sont très faibles et dont l’aire de l’élément est quasiment nulle, soient générés. Même si pour notre problématique de visualisation, la présence de tels éléments n’est pas aussi grave que lors de la génération d’un maillage de calcul, ceci peut avoir deux conséquences néfastes. Des artefacts de visualisation peuvent apparaître ainsi que des problèmes de précision. Afin de soulager les algorithmes, il est intéressant d’éliminer ce genre d’éléments. En outre, lors de la génération de maillages en 3D où les problèmes de précision sont beaucoup plus prépondérants.

Sur la figure 4.18, ces problèmes sont illustrés pour une fonction de base de Gauss-Lobatto définie sur un quadrangle Q^1 . L'estimateur spécifique n'est pas utilisé dans ce cas puisque l'élément est droit. La fonction est nulle sur le bord et donc il n'y a pas d'erreur de visualisation lors du maillage des arêtes. Aussi, le maillage de représentation est généré sans ajouter le moindre point sur le bord. Ceci signifie que certains éléments du maillage de représentation s'appuient ainsi sur les coins du quadrangle et peuvent donc tendre vers des éléments plats. Il est clair sur les figures 4.18.a et 4.18.c que les éléments près du bord sont quasiment plats. Ceci a une conséquence qui peut être préjudiciable lorsque l'on regarde la solution en relief comme le montre la figure 4.18.b.

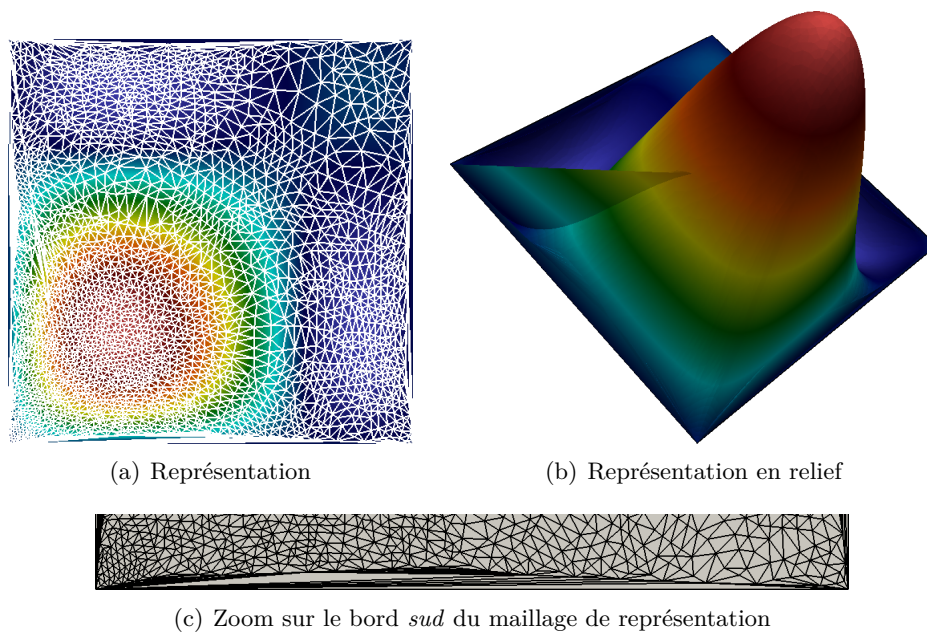


FIGURE 4.18 – Représentation d'une fonction Gauss-Lobatto à 0.1 % avec prise en compte uniquement du bord lors du maillage des arêtes. Des éléments plats apparaissent près du bord.

Pour toutes ces raisons et en s'inspirant de ce qui a été fait pour traiter les éléments courbes, nous souhaitons être capable d'avoir un maillage du bord éventuellement plus raffiné. Pour cela, l'idée est de travailler non pas uniquement sur le bord mais dans un certain voisinage autour du bord lors de la génération du maillage 1D.

Lorsqu'une arête est traitée, nous souhaitons pouvoir investiguer le problème dans une surface. Nous généraliserons ensuite à la dimension supérieure. La première question est de savoir comment définir cette zone. En particulier, si une zone trop large est définie, le risque est de trop raffiner le maillage du bord. Néanmoins, si cette zone est trop petite, le maillage risque de ne pas être modifié et donc la qualité inchangée. Il y aurait une infinité de possibilités pour définir cette zone. Les choix que nous faisons sont donc tout à fait arbitraires mais intuitifs. Cette arête est définie par ces extrémités à savoir deux points. La taille de cette zone doit d'une certaine façon être proportionnelle à la longueur de l'arête considérée. Nous définissons la zone sur la cellule de référence car les indicateurs d'erreurs sont également définis dessus. Nous lui associons une normale entrante à l'élément. A partir de ces deux extrémités et de cette normale, on est capable de définir un troisième point afin que la zone soit un triangle équilatéral. Ce choix est

motivé par plusieurs observations. Tout d'abord, la taille de cette zone est ainsi naturellement proportionnelle à la longueur de l'arête. De plus, la définition de ce troisième point, noté \hat{P}_3 , est triviale. En effet, si on note \hat{P}_1 et \hat{P}_2 les extrémités de l'arête, $|\hat{P}_1\hat{P}_2|$ la longueur de l'arête et \hat{n} la normale entrante, on a

$$\hat{P}_3 = \frac{\hat{P}_1 + \hat{P}_2}{2} + \left| \hat{P}_1\hat{P}_2 \right| \frac{\sqrt{3}}{2} \hat{n}. \quad (4.5)$$

La figure 4.19 illustre deux exemples de telles zones. Naturellement, plus le segment $[\hat{P}_1\hat{P}_2]$ est court, plus cette zone est petite. Cela permet la convergence de l'algorithme.

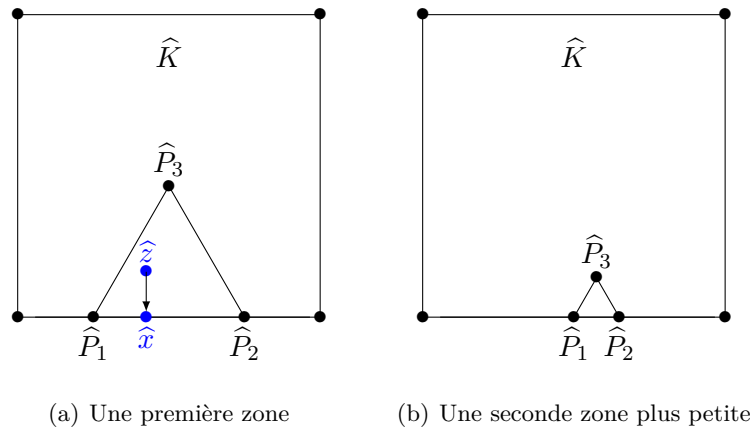


FIGURE 4.19 – Exemples de zones de voisinage en dimension 2.

Finalement, comme la zone considérée est un triangle, l'algorithme DIRECT peut être appliqué et l'indicateur d'erreurs classique 2D est estimé dans cette zone. Naturellement, et comme dans le cas d'éléments courbes, les points qui sont ajoutés au maillage de représentation le sont sur le bord, c'est à dire sur l'arête considérée. Pour cela, le point ajouté est choisi comme le projeté orthogonal sur l'arête. Par exemple, sur la figure 4.19, si l'algorithme a trouvé le maximum au point \hat{z} , le point \hat{x} , qui est son projeté orthogonal sur l'arête est ajouté au maillage 1D.

Si maintenant nous nous intéressons à une face, nous souhaitons définir un voisinage qui sera un tétraèdre délimité par les 3 sommets de cette face (qui en fait est un triangle) et un quatrième point à définir. De la même manière que précédemment, on note \hat{P}_1 , \hat{P}_2 et \hat{P}_3 les trois sommets de la face considérée, $|\hat{P}_1\hat{P}_2\hat{P}_3|$ l'aire de la face et \hat{n} la normale entrante. Les coordonnées du quatrième point sont données par

$$\hat{P}_4 = \frac{\hat{P}_1 + \hat{P}_2 + \hat{P}_3}{3} + \left| \hat{P}_1\hat{P}_2\hat{P}_3 \right| \frac{\sqrt{6}}{3} \hat{n}. \quad (4.6)$$

La figure 4.20 montre la représentation obtenue à 0.1 % lorsque l'on ne prend en compte que le bord lors du maillage de l'arête, celle-ci représente la même fonction que la figure 4.18. Sur la figure 4.21, la même fonction est représentée mais en prenant en compte le voisinage. Le maillage des arêtes est raffiné (voir tableau 4.6). Ceci permet d'éliminer les éléments presque plats comme on peut le voir sur la figure 4.21.b et d'éviter les artefacts lorsque l'on représente la solution en relief comme sur la figure 4.21.c. Nous tirons les mêmes conclusions lorsque la

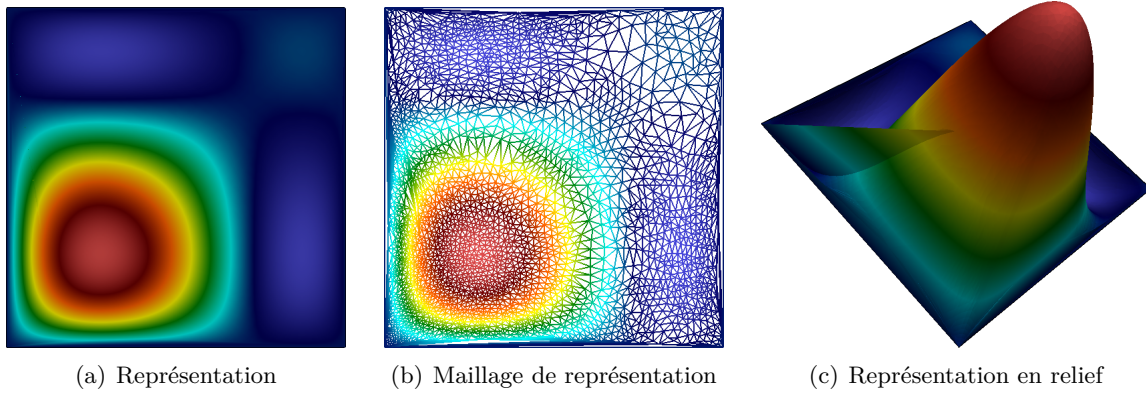


FIGURE 4.20 – Représentation d’une fonction Gauss-Lobatto à 0.1 % avec prise en compte uniquement du bord lors du maillage des arêtes.

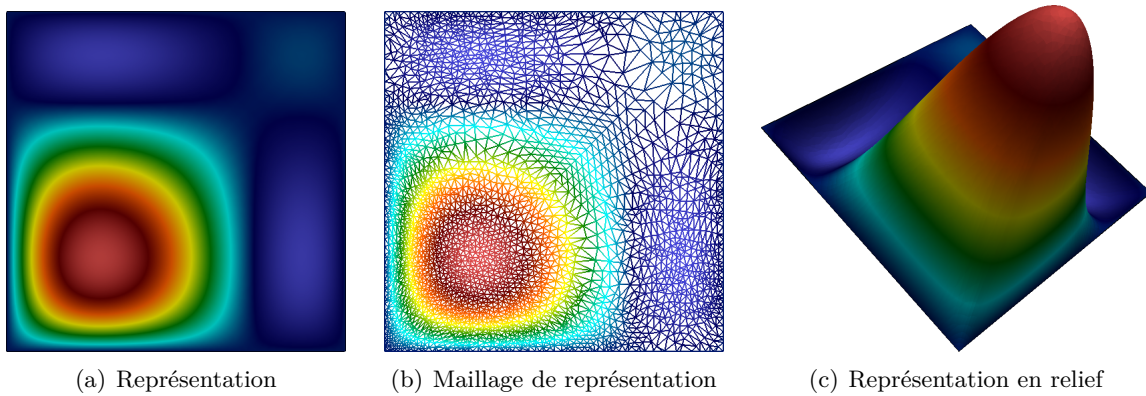


FIGURE 4.21 – Représentation d’une fonction Gauss-Lobatto à 0.1 % avec prise en compte du voisinage lors du maillage des arêtes.

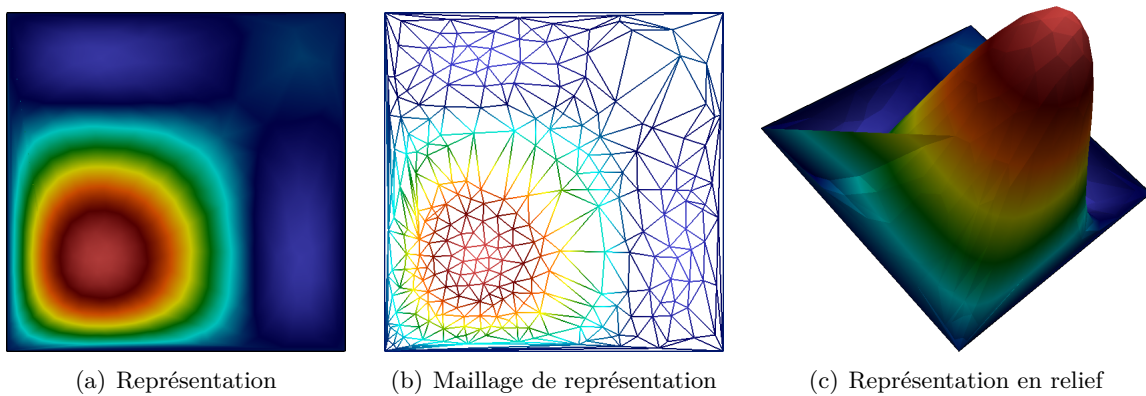


FIGURE 4.22 – Représentation d’une fonction Gauss-Lobatto à 1 % avec prise en compte uniquement du bord lors du maillage des arêtes.

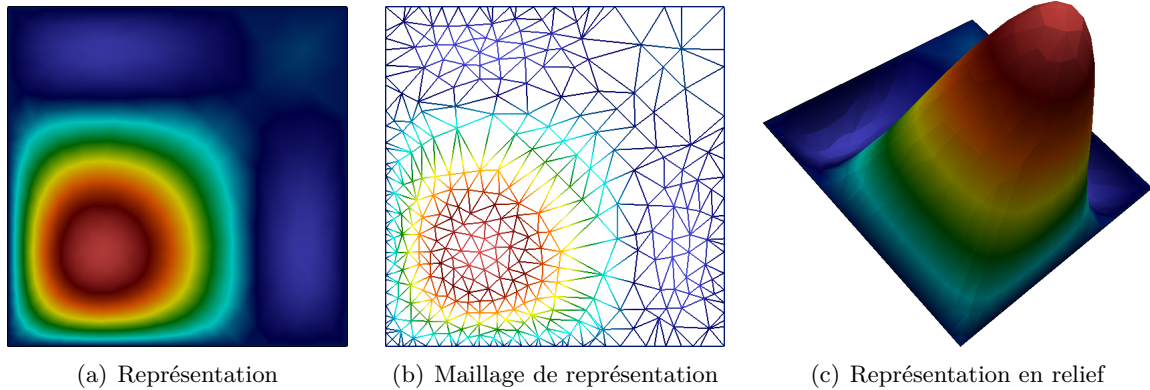


FIGURE 4.23 – Représentation d’une fonction Gauss-Lobatto à 1 % avec prise en compte du voisinage lors du maillage des arêtes.

tolérance est fixée à 1 % sur les figures 4.22 et 4.23.

| Méthode | Arête 1 | Arête 2 | Arête 3 | Arête 4 | Élément 2D |
|------------------|---------|---------|---------|---------|------------|
| Bord (1%) | 1 | 1 | 1 | 1 | 556 |
| Voisinage (1%) | 32 | 20 | 32 | 20 | 634 |
| Bord (0.1%) | 1 | 1 | 1 | 1 | 5406 |
| Voisinage (0.1%) | 67 | 64 | 67 | 64 | 5648 |

TABLE 4.6 – Nombre de segments sur les maillages des arêtes et de triangles sur le maillage de l’élément pour représenter la fonction Gauss-Lobatto

Nous pouvons néanmoins imaginer d’autres définitions de zones (voisinages). En particulier, la définition proposée a la particularité d’être purement géométrique (i.e. elle ne prend pas en compte les valeurs de la solution à visualiser) mais a l’avantage d’être très facile à mettre en place et efficace. A l’instar de ce qui a été fait pour le traitement des éléments courbes, une autre idée est, pour chaque point \hat{x} de l’arête \hat{S} d’associer un paramètre $t_{\hat{x}}$ qui correspond à un avancement maximal en suivant la normale \hat{n} . La zone ainsi définie est plus complexe mais en définissant une transformation comme on l’a fait pour les éléments courbes, on peut se ramener à un triangle isocèle et appliquer DIRECT dessus. Il reste à choisir comment définir $t_{\hat{x}}$. On pourrait par exemple faire intervenir dans ce choix plusieurs paramètres tels que la longueur de l’arête, la valeur de la dérivée normale au point considéré sur l’arête... De nombreux choix seraient donc envisageables mais nous utilisons uniquement le critère géométrique.

4.4 Conclusions du chapitre

Dans ce chapitre, nous avons montré le potentiel de la méthode de visualisation sur des exemples numériques de complexité croissante. Tout d’abord, nous avons montré l’intérêt d’uti-

liser un raffinement adaptatif par rapport à un redécoupage uniforme. Les gains sont non négligeables et sont d'autant plus importants pour l'interactivité dans le logiciel de visualisation. Des données trop lourdes peuvent ralentir ou empêcher le fonctionnement d'un logiciel de visualisation. Nous illustrons la nécessité de demander un contrôle ponctuel sur l'erreur de visualisation. Pour cela, nous comparons notre approche, qui assure un contrôle en norme L^∞ , à une approche guidée par la norme L^2 . Dans ce dernier cas, une partie de la richesse de l'information peut être perdue, ce qui est préjudiciable pour l'analyse et l'interprétation des résultats.

Dans un second temps, nous nous sommes intéressés au traitement avec notre méthode de surfaces courbes. En particulier, nous avons montré l'importance de l'indicateur d'erreurs *a posteriori* spécifique que l'on met en place. Nous avons également traité une surface courbe dans l'espace et exhiber l'apparition de retournements que nous avons pu éliminer en utilisant l'indicateur d'erreurs *a posteriori* spécifique.

Ensuite, nous avons proposé une stratégie pour améliorer la qualité des éléments en considérant le voisinage du bord des éléments. Cette idée s'inspire de la gestion des éléments courbes où le traitement des arêtes et des faces est faite en considérant non pas uniquement le bord mais en investiguant l'intérieur de l'élément afin de soulager la contrainte du maillage par dimensions croissantes.

Nous nous sommes également intéressé à des exemples volumiques. La méthode de visualisation est utilisée pour traiter une fonction analytique, un mode, défini sur des hexaèdres et nous avons montré sa capacité à traiter ce type d'éléments.

Un point limitant de notre méthode concerne les temps CPU qui peuvent être longs, notamment à cause du contrôle ponctuel demandé. Pour donner un ordre d'idée, pour traiter des milliers d'éléments volumiques, le temps écoulé peut dépasser plusieurs minutes voire des dizaines de minutes, même si évidemment, cela dépend de la complexité de la solution ainsi que du nombre de processeurs utilisé. La structure de notre méthodologie permet naturellement de se placer dans un cadre parallèle puisque chaque arête, face ou volume peut être traité après décomposition indépendamment des autres.

Maintenant que nous avons validé numériquement notre méthode de visualisation, nous allons l'utiliser dans le chapitre suivant où nous reconstruisons et visualisons des champs rayonnés en équations intégrales. La souplesse de notre méthode de visualisation permet de proposer une solution performante à cette problématique. En particulier, d'autres visualisations de cas volumiques, issus de codes de calcul, sont traités dans le chapitre 5.

Chapitre 5

Reconstruction et visualisation de champs rayonnés pour des ondes harmoniques

Contents

| | | |
|------------|--|------------|
| 5.1 | Positionnement du problème | 118 |
| 5.2 | Construction de cartographies | 121 |
| 5.2.1 | Rayonnements sur grilles cartésiennes | 122 |
| 5.2.2 | Rayonnements par interpolations d'ordre élevé | 122 |
| 5.3 | Compression de l'information : interpolation directionnelle | 124 |
| 5.3.1 | Rayonnements par interpolations du noyau de Green | 124 |
| 5.3.2 | Interpolation directionnelle avec une direction | 125 |
| 5.3.3 | Interpolation directionnelle avec plusieurs directions | 127 |
| 5.4 | Construction de champs rayonnés | 128 |
| 5.4.1 | Subdivision du domaine à visualiser | 128 |
| 5.4.2 | Subdivision de la frontière de l'objet diffractant | 128 |
| 5.4.3 | Critères de construction des champs rayonnés | 129 |
| 5.4.4 | Visualisation des champs reconstruits | 132 |
| 5.5 | Un algorithme pour reconstruire des champs rayonnés | 132 |
| 5.5.1 | Critère d'admissibilité | 133 |
| 5.5.2 | Subdivision du domaine | 134 |
| 5.5.3 | Assignation des ordres d'interpolation. | 136 |
| 5.6 | Validations numériques des champs rayonnés reconstruits. | 138 |
| 5.6.1 | Interpolation directionnelle avec une seule direction | 138 |
| 5.6.2 | BEM en 2D et rayonnements dans le plan | 139 |
| 5.6.2.1 | Diffraction par un disque | 139 |
| 5.6.2.2 | Diffraction autour d'un disque | 141 |
| 5.6.2.3 | Diffraction autour d'un objet <i>L-shape</i> | 144 |
| 5.6.3 | BEM en 3D et rayonnements dans des plans | 148 |
| 5.6.4 | BEM en 3D et rayonnements dans des volumes | 149 |
| 5.6.4.1 | Un rayonnement en dehors de l'obstacle diffractant | 149 |
| 5.6.4.2 | Un rayonnement où le domaine à visualiser englobe l'obstacle | 150 |
| 5.7 | Conclusions du chapitre | 155 |

Les formules de représentation intégrale permettent de paramétriser la solution d'une EDP dans un domaine à partir d'informations données sur une surface (au terme source près). Ces formules peuvent être utilisées à la fois pour construire des schémas numériques ou pour étendre la solution numérique dans un domaine d'intérêt plus vaste. Dans ce chapitre, nous nous plaçons sur cette dernière problématique. Plus précisément, notre objectif est de proposer une méthode rendant possible la représentation de champs d'ondes dans de grands domaines (par rapport à la longueur d'onde). Les deux verrous principaux pour réaliser cela sont d'une part, un nombre de points nécessaires pouvant être prohibitif pour la construction d'un rendu précis du champ et d'autre part, le calcul de la solution en chaque point qui nécessite l'évaluation de produits de convolution (mettant en jeu une solution élémentaire du problème). Ces contraintes font que, en général, les exploitations des résultats obtenus par ces méthodes sont limitées.

Nous souhaitons proposer une méthode précise et peu coûteuse pour reconstruire les champs rayonnés. Pour cela, nous allons utiliser une interpolation des champs, définie à partir de fonctions de base liées à la physique du problème. Ce choix va nous permettre de diminuer drastiquement le nombre de fonctions par rapport à une interpolation polynomiale classique. On parle de compression des données. Nous avons choisi de la construire à partir d'une interpolation directionnelle du noyau comme celle utilisée pour la compression algébrique de matrices issues de BEM. Cette stratégie est utilisée pour approcher des noyaux de Helmholtz à hautes fréquences dans le cadre des éléments finis de frontière [Börm, 2015, Börm and Melenk, 2017] et pour faire de la compression de matrices [Brandt, 1991, Engquist and Ying, 2007, Messner et al., 2012, Bebendorf et al., 2015] afin d'améliorer l'efficacité des calculs puisque les matrices sont pleines. Enfin, la méthode de visualisation présentée précédemment pourra être utilisée afin de représenter et post-traiter les champs ainsi reconstruits. Si nous n'avions pas la possibilité de visualiser ces champs, tout le travail de reconstruction serait beaucoup moins intéressant, voire inutile.

Ce chapitre est organisé de la façon suivante. Nous nous intéressons à la construction de cartographies de champs rayonnés qui constitue notre problématique. L'interpolation directionnelle est ensuite décrite et nous montrons l'intérêt de l'utiliser dans notre cas. Puis, nous décrivons notre approche qui permet de reconstruire efficacement et de manière compressée des champs rayonnés. Enfin, nous proposons un algorithme générant des champs reconstruits que nous validons numériquement ensuite.

5.1 Positionnement du problème

Nous commençons cette partie par une présentation abstraite et formelle du problème car la méthodologie proposée peut s'adapter à de nombreuses situations. Des exemples concrets seront donnés dans la suite.

Construction formelle du cadre de travail

Soient Ω_s un domaine borné suffisamment régulier de \mathbb{R}^d et L un opérateur différentiel. Nous nous intéressons aux solutions des problèmes de la forme :

$$Lu = 0 \text{ dans } \Omega_e := \mathbb{R}^d \setminus \overline{\Omega}_s. \quad (5.1)$$

Remarque 5.1. Une condition de fermeture du système (par exemple radiation à l'infini) peut être éventuellement ajoutée afin de fermer le problème.

On note γ_c l'opérateur « données de Cauchy » sur $\Gamma := \partial\bar{\Omega}_s$ associé à L . Par exemple, si $L = -\Delta$ alors $\gamma_c u = (u|_\Gamma, \partial u/\partial n)$ avec $\partial u/\partial n = (\nabla u)|_\Gamma \cdot n$ la dérivée normale sur Γ .

Si on étend une solution u de (5.1) par zéro dans Ω_s , alors cette dernière vérifie une équation de la forme :

$$Lu = T_\Gamma \text{ dans } \mathbb{R}^d, \quad (5.2)$$

où T_Γ est une distribution à support compact Γ (la frontière de Ω_s) dépendante de $\gamma_c u$ [Costabel, 1986].

Supposons à présent qu'une solution fondamentale \mathcal{S} de L est connue c'est-à-dire :

- pour toute distribution u à support compact : $\mathcal{S} L u = u = L \mathcal{S} u$,
- \mathcal{S} est un opérateur intégral :

$$\mathcal{S} v(x) = \int_{\mathbb{R}^d} G(x, y) v(y) dy, \quad (5.3)$$

avec un noyau G faiblement singulier pour $x = y$.

Dans ce contexte, il est alors connu que les solutions de (5.1) peuvent être paramétrisées à partir des données de Cauchy via la forme

$$u = G * T_\Gamma, \quad (5.4)$$

où $*$ désigne l'opérateur de convolution.

Ainsi, si le noyau G est de plus de classe C^∞ en dehors de la diagonale $x = y$, alors u est de classe C^∞ en dehors du support de T_Γ et en particulier

$$u(x) = \langle T_\Gamma, G(x, y) \rangle_{\mathcal{E}'(\mathbb{R}^d), \mathcal{E}(\mathbb{R}^d)}, \quad x \in \Omega_e, \quad (5.5)$$

où $\mathcal{E}'(\mathbb{R}^d)$ est l'espace des distributions à support compact de \mathbb{R}^d .

Cette dernière formule est très importante car elle permet d'une part de construire des schémas numériques (dans le cadre des éléments finis de frontières ou pour borner un domaine de calcul), et d'autre part, de construire (*a posteriori*) les champs solutions dans tout le domaine Ω_e lorsque $\gamma_c u$ est connu. C'est cette dernière situation qui nous intéresse ici. On peut imaginer qu'une méthode numérique (BEM, FEM, DG, FD etc...) soit utilisée pour calculer la solution dans une partie du domaine d'intérêt (zones où il y a des hétérogénéités par exemple) et nous fournit donc les données de Cauchy sur une frontière (d'un obstacle par exemple). A partir de ces données et de la formule (5.5), nous pouvons alors déterminer la solution dans le reste du domaine.

Dans ce chapitre, nous nous focalisons sur les phénomènes oscillants provenant des problèmes de propagation d'ondes. Nous explicitons donc à présent le cadre abstrait précédent pour l'acoustique et l'électromagnétisme. Il faut noter que les phénomènes mettant en jeu des ondes élastiques pourraient naturellement être traités par l'approche qui va être présentée.

L'équation de Helmholtz en acoustique

L'équation de Helmholtz est donnée par

$$Lu = \Delta u + k^2 u = 0, \quad \text{dans } \mathbb{R}^d \setminus \bar{\Omega}_s, \quad (5.6)$$

où Δ est le laplacien, k est le nombre d'onde (supposé constant), Ω_s est l'objet diffractant supposé Lipschitzien et u est le champ acoustique total.

En acoustique, la distribution T_Γ introduite dans (5.4) s'écrit indépendamment des conditions aux limites comme

$$T_\Gamma = -\frac{\partial u}{\partial n}\delta_\Gamma + u\delta'_\Gamma, \quad (5.7)$$

où les distributions $\frac{\partial u}{\partial n}\delta_\Gamma$ et $u\delta'_\Gamma$ sont définies¹ par : $\forall \varphi \in \mathcal{D}(\mathbb{R}^d, \mathbb{C})$,

$$\begin{aligned} \left\langle \frac{\partial u}{\partial n}\delta_\Gamma, \varphi \right\rangle &:= \int_\Gamma \frac{\partial u}{\partial n} \bar{\varphi} d\gamma. \\ \langle u\delta'_\Gamma, \varphi \rangle &:= \int_\Gamma u \frac{\partial \bar{\varphi}}{\partial n} d\gamma. \end{aligned} \quad (5.8)$$

Si nous ajoutons à (5.6) la condition de radiation de Sommerfeld :

$$\lim_{|x| \rightarrow +\infty} |x|^{\frac{d-1}{2}} \left(\frac{\partial}{\partial |x|} - ik \right) u(x) = 0 \quad (5.9)$$

pour sélectionner les ondes sortantes, le noyau fondamental associé est défini par [Nédélec, 2001] :

$$G(x, y) = \begin{cases} \frac{i}{4} H_0^{(1)}(k|x-y|), & \text{si } d = 2, \\ \frac{e^{ik|x-y|}}{4\pi|x-y|}, & \text{si } d = 3, \end{cases} \quad (5.10)$$

où $H_0^{(1)}$ est la fonction de Hankel de premier type et d'ordre 0 [Abramowitz and Stegun, 1964].

La formule de représentation de type (5.5) est alors donnée par :

$$u(x) = - \int_\Gamma G(x, y) \frac{\partial u}{\partial n_y}(y) d\gamma_y + \int_\Gamma \frac{\partial G}{\partial n_y}(x, y) u(y) d\gamma_y, \quad \text{si } x \in \Omega_e. \quad (5.11)$$

Les équations de Maxwell en électromagnétisme

Si maintenant on s'intéresse à un problème d'électromagnétisme, le système de Maxwell est donné par

$$\begin{cases} \nabla \times \mathbf{E} - ikZ\mu\mathbf{H} = 0 \text{ dans } \mathbb{R}^d \setminus \bar{\Omega}_s, \\ \nabla \times \mathbf{H} + ik\frac{1}{Z}\varepsilon\mathbf{E} = 0 \text{ dans } \mathbb{R}^d \setminus \bar{\Omega}_s, \\ \lim_{|x| \rightarrow +\infty} |x| \left(Z\mathbf{H}(x) \times \frac{x}{|x|} - \mathbf{E}(x) \right) = 0, \end{cases} \quad (5.12)$$

où plusieurs grandeurs physiques interviennent :

- un champ électrique : \mathbf{E} ,
- un champ magnétique : \mathbf{H} ,

Le milieu est supposé homogène et est caractérisé par les paramètres suivants :

- une permittivité relative : ε ,

1. On a supposé que les traces de Cauchy étaient au moins intégrables.

- une perméabilité relative : μ ,
- l'impédance du vide $Z = \sqrt{\frac{\mu_0}{\varepsilon_0}} \approx 100\pi$,
- le nombre d'onde (dans le vide) k .

Les formules de représentation sont alors dérivées en réécrivant le problème (5.12) sous forme d'un système d'équations de Helmholtz vectorielles (pour cela, on utilise les propriétés $\operatorname{div} \mathbf{E} = 0$ et $\operatorname{div} \mathbf{H} = 0$) et en utilisant le noyau fondamental G présenté dans la partie acoustique. Nous obtenons alors les formules suivantes [Nédélec, 2001].

Théorème 5.2 (Formules de représentation). *Les solutions du problème de Maxwell (5.12) vérifient les formules suivantes : pour $x \in \Omega_e$,*

$$\begin{cases} \mathbf{E}(x) &= ikZ(\mathcal{T}\mathbf{J})(x) + (\mathcal{K}\mathbf{M})(x), \\ \mathbf{H}(x) &= -(\mathcal{K}\mathbf{J})(x) + ikZ^{-1}(\mathcal{T}\mathbf{M})(x), \end{cases} \quad (5.13)$$

où, $\mathbf{J} = n \times \mathbf{H}|_{\Gamma}$ et $\mathbf{M} = \mathbf{E}|_{\Gamma} \times n$ sont respectivement les courants électrique et magnétique sur Γ et

$$\begin{cases} (\mathcal{T}\mathbf{J})(x) = \frac{1}{k^2} \nabla \int_{\Gamma} G(x, y) \nabla_{\Gamma} \cdot \mathbf{J}(y) d\gamma_y + \int_{\Gamma} G(x, y) \mathbf{J}(y) d\gamma_y, \\ (\mathcal{K}\mathbf{M})(x) = \int_{\Gamma} \nabla_y G(x, y) \times \mathbf{M}(y) d\gamma_y. \end{cases} \quad (5.14)$$

5.2 Construction de cartographies

Nous avons rappelé que même si les inconnues ne sont calculées uniquement que sur la frontière Γ , il est possible, grâce aux formules de représentation (théorèmes 5.1 et 5.2), de calculer la solution en n'importe quel point. C'est justement ce dernier aspect qui nous intéresse. En effet, nous souhaitons, à partir de solutions surfaciques, calculer le rayonnement sur un certain domaine noté $\mathcal{V} \subset \Omega_e$. Pour cela, une technique classique et naïve consiste à définir une grille cartésienne, à calculer la solution en chacun des points de la grille à l'aide des formules de représentation puis à utiliser un logiciel de visualisation afin d'avoir la solution dans le domaine considéré [Tzoulis and Eibert, 2006]. La figure 5.1 illustre ce processus.

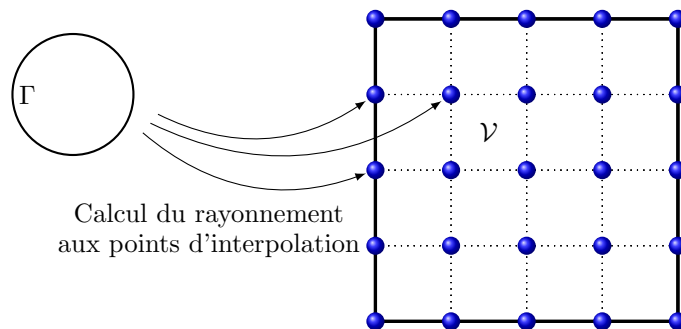


FIGURE 5.1 – Un rayonnement en utilisant une grille d'interpolation

5.2.1 Rayonnements sur grilles cartésiennes

Plusieurs champs rayonnés obtenus à partir de grilles cartésiennes et d'interpolations P^1 sont représentés sur la figure 5.2. La taille du domaine correspond à une longueur d'onde dans chaque direction. Naturellement, plus le nombre de points d'interpolation est élevé, plus la solution est précise mais plus la génération des données est coûteuse. En effet, comme les formules de représentation font intervenir une convolution entre le noyau de Green G et la solution calculée sur la surface (par exemple par un code équations intégrales), chaque appel à cette formule est coûteux et ce coût augmente avec le nombre d'éléments composant la frontière. Une interpolation affine permet ensuite d'avoir la solution sur tout le domaine, mais celle-ci peut-être imprécise. La figure 5.2.a est générée à l'aide de $5 \times 5 = 25$ éléments soit $6 \times 6 = 36$ points d'interpolation. La reconstruction obtenue est très imprécise. Si maintenant une grille plus raffinée est utilisée, comme sur les images suivantes de la figure 5.2, les champs rayonnés deviennent plus précis.

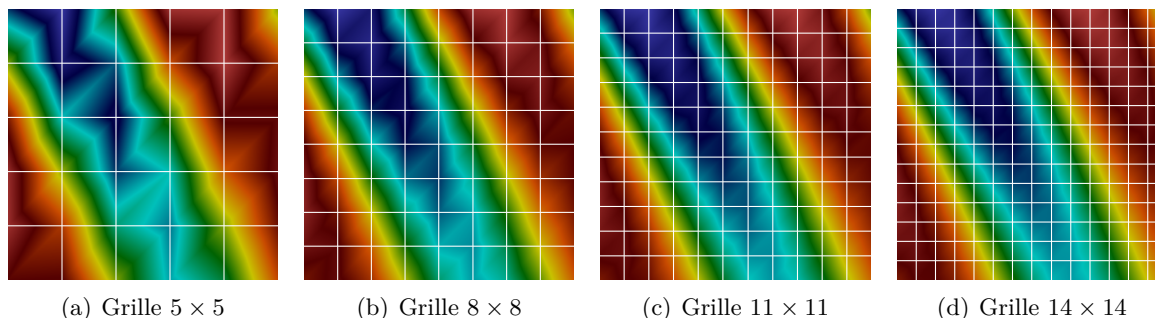


FIGURE 5.2 – Champs rayonnés obtenus avec différentes grilles de visualisation (interpolation P^1).

Afin de gagner en performance, les interpolations affines (sur grilles) peuvent être remplacées par des interpolations d'ordre élevé.

5.2.2 Rayonnements par interpolations d'ordre élevé

Sur la figure 5.3, des champs rayonnés obtenus par une interpolation P^1 sur des grilles cartésiennes (bas ordre) sont comparés avec une interpolation d'ordre élevé pour une même erreur d'interpolation de 1% en norme L^∞ . En utilisant des interpolations affines sur une grille, 289 degrés de liberté sont nécessaires pour avoir une précision de 1% (figure 5.3.a). Si maintenant, des fonctions de degré 6 sont considérées, seulement 49 degrés de liberté sont nécessaires pour avoir une précision similaire sur le même domaine. Les points d'interpolation sont affichés en blanc sur la figure 5.3. Il s'agit de points de Chebyshev, qui sont présentés en annexe B et sont connus pour avoir des bonnes propriétés d'interpolation en norme L^∞ .

Nous considérons ensuite un domaine plus large encore (de taille $3\lambda \times 3\lambda$ au lieu de $\lambda \times \lambda$ où λ est la longueur d'onde) sur les figures 5.3.c et 5.3.d. Le tableau 5.1 récapitule les résultats et montre les gains (ratios). Lorsque la taille du domaine augmente, l'ordre élevé permet de réduire encore plus le nombre de degrés de liberté.

Même si l'ordre élevé permet de réduire le nombre de degrés de liberté par rapport à l'interpolation P^1 , ce coût peut toujours être élevé. Ainsi, dans la section suivante, nous nous intéressons

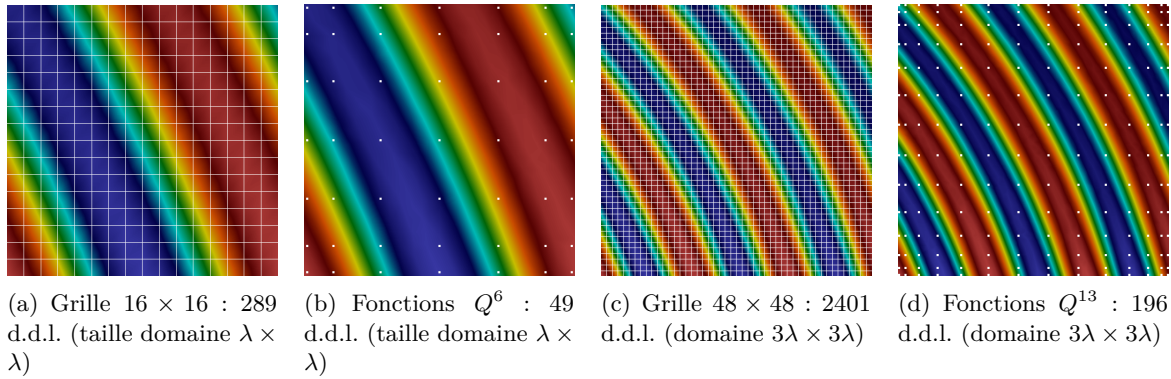


FIGURE 5.3 – Représentation de plusieurs champs rayonnés avec des fonctions de bas ordre (a et c) et des fonctions d'ordre élevé (b et d) pour une erreur d'interpolation L^∞ sous 1%.

| Précision 1% | Domaine $\lambda \times \lambda$ | Domaine $3\lambda \times 3\lambda$ |
|--------------------------|----------------------------------|------------------------------------|
| Ordre bas | 289 | 2401 |
| Ordre élevé | 49 | 196 |
| Ratio (Bas/Ordre) | 5.90 | 12.25 |

TABLE 5.1 – Récapitulatif du nombre de d.d.l. nécessaire pour avoir une interpolation L^∞ sous 1% de la figure 5.3.

à l'interpolation directionnelle qui, via l'introduction de directions bien choisies et en définissant les fonctions de base comme des combinaisons entre des polynômes d'ordre élevé et des ondes planes, réduit drastiquement le nombre de degrés de liberté nécessaires pour interpoler le noyau de Green.

5.3 Compression de l'information : interpolation directionnelle

Dans cette section, nous montrons comment nous pouvons compresser l'information en utilisant des interpolations directionnelles.

5.3.1 Rayonnements par interpolations du noyau de Green

Dans les travaux sur la compression de matrices- \mathcal{H}^2 [Börm, 2015, Börm and Melenk, 2017], l'interpolation directionnelle est utilisée pour approcher le noyau de Green. Nous allons, dans notre cadre, exploiter cette idée.

Lorsque les rayonnements sont obtenues sur des grilles cartésiennes (i.e. interpolation affine) ou par interpolation d'ordre élevé, la solution u est approchée de la manière suivante :

$$u(x) \approx \sum_{j=1}^{N_{\mathcal{V}}} u(\xi_j) \mathcal{L}_j(x), \quad \forall x \in \mathcal{V}, \quad (5.15)$$

où $N_{\mathcal{V}}$ désigne le nombre de points d'interpolation, $(\xi_j)_{j=1, \dots, N_{\mathcal{V}}}$ sont les points d'interpolation et $(\mathcal{L}_j)_{j=1, \dots, N_{\mathcal{V}}}$ sont les polynômes de Lagrange associés à ces points. Les degrés de liberté $(u(\xi_j))_{j=1, \dots, N_{\mathcal{V}}}$ sont calculés par les formules de représentation (5.4).

Cependant, au lieu d'interpoler le champ rayonné, il peut être plus judicieux d'interpoler le noyau G . En effet, son expression analytique est connue alors que le champ rayonné dépend de la solution calculée sur la surface. Ainsi, une interpolation du noyau peut être effectuée de manière tout à fait indépendante de la résolution du problème qui permet de calculer la solution sur la surface.

Si les mêmes fonctions de base et les mêmes points d'interpolations sont utilisés, ces deux interpolations donnent la même approximation sur u . En effet, si le noyau est interpolé dans les mêmes conditions que (5.15), cette approximation s'écrit :

$$G(x, y) \approx \sum_{j=1}^{N_{\mathcal{V}}} G(\xi_j, y) \mathcal{L}_j(x), \quad \forall x \in \mathcal{V}. \quad (5.16)$$

Alors, en prenant en compte cette approximation du noyau, nous en déduisons sur u

$$u(x) \approx \left\langle \mathbf{T}_{\Gamma}, \sum_{j=1}^{N_{\mathcal{V}}} G(\xi_j, y) \mathcal{L}_j(x) \right\rangle, \quad (5.17)$$

et

$$u(x) \approx \sum_{j=1}^{N_{\mathcal{V}}} \underbrace{\langle \mathbf{T}_{\Gamma}, G(\xi_j, y) \rangle}_{u(\xi_j)} \mathcal{L}_j(x), \quad \forall x \in \mathcal{V}. \quad (5.18)$$

Ainsi, pour contrôler l'approximation de la solution pour n'importe quelle onde incidente ou n'importe quelle source au problème, il suffit de contrôler l'approximation du noyau.

5.3.2 Interpolation directionnelle avec une direction

Sur la figure 5.4, un cas particulier du noyau de Green est considéré où le point y est fixé à l'origine notée O . Le domaine de visualisation \mathcal{V} est un carré dont les abscisses et ordonnées varient entre 5λ et 6λ où $\lambda = 2\pi/k$. La figure 5.4.a représente le noyau de Green donné par

$$G(x, O) = \frac{e^{ik\|x\|}}{4\pi\|x\|}, \quad \forall x \in \mathcal{V}. \quad (5.19)$$

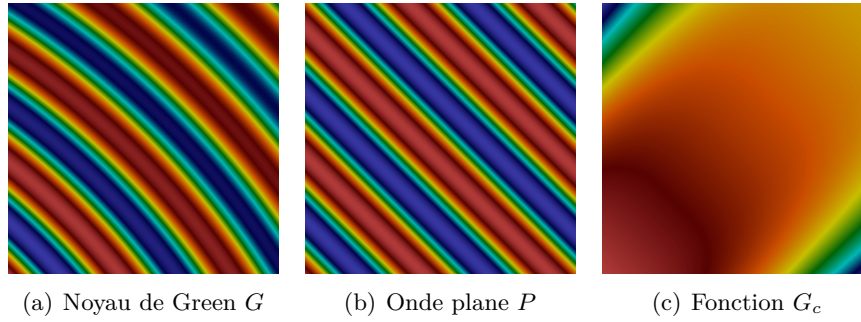


FIGURE 5.4 – Visualisation du noyau de Green G (a), d'une onde plane P (b) et de la fonction G_c (c).

Le comportement de ce noyau de Green est particulier (figure 5.4.a) et est proche d'une onde plane (figure 5.4.b). Nous introduisons une direction c :

$$c = \frac{x_c}{\|x_c\|}, \quad (5.20)$$

où x_c est le centre du carré. A partir de cette direction, nous définissons une onde plane P sur le même domaine de la façon suivante

$$P(x) = e^{ik\langle c, x \rangle}, \quad \forall x \in \mathcal{V}. \quad (5.21)$$

Cette onde plane est représentée sur la figure 5.4.b. Nous définissons maintenant un résidu G_c comme le quotient entre le noyau G et cette onde plane P :

$$G_c(x) = G(x, O)/P(x) = G(x, O)e^{-ik\langle c, x \rangle}, \quad \forall x \in \mathcal{V}.$$

Cette fonction G_c est représentée sur la figure 5.4.c et nous remarquons qu'elle oscille beaucoup moins que le noyau G donné sur la figure 5.4.a. Ainsi, beaucoup moins de degrés de liberté seront nécessaires pour interpoler cette fonction pour la même précision. Ce cas particulier montre l'intérêt de l'interpolation directionnelle.

De manière générale, nous définissons pour $c \in \mathbb{R}^3$ le résidu G_c comme

$$G_c(x, y) = G(x, y)e^{-ik\langle c, x-y \rangle} = \frac{e^{ik(\|x-y\| - \langle c, x-y \rangle)}}{4\pi\|x-y\|}, \quad \forall x \in \mathcal{V}, \forall y \in \Gamma. \quad (5.22)$$

L'approximation de G_c s'écrit donc

$$G_c(x, y) \approx \sum_{j=1}^{N_{\mathcal{V}}} G_c(\xi_j, y) \mathcal{L}_j(x), \quad (5.23)$$

où $(\mathcal{L}_j)_{j=1,\dots,N_V}$ sont les polynômes de Lagrange aux points $(\xi_j)_{j=1,\dots,N_V}$.
 En écrivant u comme

$$u(x) = \left\langle \mathbf{T}_\Gamma, G(x, y) e^{-ik\langle c, x-y \rangle} e^{ik\langle c, x-y \rangle} \right\rangle,$$

on en déduit

$$u(x) \approx \left\langle \mathbf{T}_\Gamma, \sum_{j=1}^{N_V} G(\xi_j, y) e^{-ik\langle c, \xi_j-y \rangle} \mathcal{L}_j(x) e^{ik\langle c, x-y \rangle} \right\rangle,$$

puis,

$$u(x) \approx \left\langle \mathbf{T}_\Gamma, \sum_{j=1}^{N_V} G(\xi_j, y) \mathcal{L}_j(x) e^{ik\langle c, x-\xi_j \rangle} \right\rangle.$$

Il vient que le champ u est maintenant approché de la façon suivante

$$u(x) \approx \sum_{j=1}^{N_V} \underbrace{\langle \mathbf{T}_\Gamma, G(\xi_j, y) \rangle}_{u(\xi_j)} \mathcal{L}_j(x) e^{ik\langle c, x-\xi_j \rangle},$$

que l'on écrit comme

$$u(x) \approx \sum_{j=1}^{N_V} u(\xi_j) \underbrace{\mathcal{L}_j(x) e^{ik\langle c, x-\xi_j \rangle}}_{\phi_j(x)}.$$

Finalement, les degrés de liberté $u(\xi_j)$ apparaissent toujours. Les fonctions de base ϕ_j sont cependant différentes puisqu'elles combinent des polynômes de Lagrange et des ondes planes intégrant la direction introduite. Le tableau 5.2 montre les gains de l'ajout de la direction. Le dernier cas de la figure 5.3 a été repris, et avec la direction introduite, seulement 25 degrés de liberté sont nécessaires pour avoir une erreur d'interpolation L^∞ inférieure à 1%.

| Précision 1% | Ordre bas (sans dir.) | Ordre élevé (sans dir.) | Ordre élevé (une dir.) |
|------------------|-----------------------|-------------------------|------------------------|
| Nombre de d.d.l. | 2401 | 196 | 25 |

TABLE 5.2 – Comparaison du nombre de degrés de liberté nécessaires après ajout d'une seule direction.

Pour que l'interpolation directionnelle soit intéressante, il faut pouvoir trouver une direction bien choisie. Considérons $y \in \Gamma$ donné et $x \in \mathbb{R}^d$ défini tel que $x = y + \alpha c + \epsilon$, où $c \in \mathbb{R}^d$ avec $\|c\| = 1$, $\epsilon \in \mathbb{R}^d$ et α est une constante (voir figure 5.5).

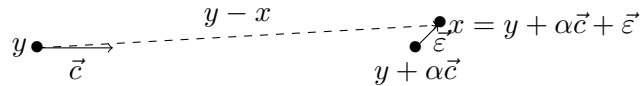


FIGURE 5.5 – Introduction d'une direction pour effectuer une interpolation directionnelle

Alors,

$$\|x - y\| = \left\langle x - y, \frac{x - y}{\|x - y\|} \right\rangle = \left\langle x - y, \frac{\alpha c + \epsilon}{\|\alpha c + \epsilon\|} \right\rangle \underset{\|\epsilon\| \rightarrow 0}{\approx} \left\langle x - y, \frac{c}{\|c\|} \right\rangle. \quad (5.24)$$

L'équation (5.24) montre le comportement lorsque la direction est choisie telle que $x = y + \alpha c + \epsilon$ et $\|\epsilon\|/\alpha \rightarrow 0$. Dans cette configuration, le noyau G a un comportement de type onde plane. Le choix de la direction est donc pertinent si

$$c \approx \frac{x - y}{\|x - y\|}, \quad \forall x \in \mathcal{V}, \forall y \in \Gamma. \quad (5.25)$$

Dans le cas où cette hypothèse est vérifiée, la quantité $(\|x - y\| - \langle c, x - y \rangle)$, qui apparaît dans la définition de G_c (5.22), est réduite et les oscillations de G_c sont donc limitées.

Si l'hypothèse (5.25) n'est *a priori* pas vérifiée, une subdivision du domaine \mathcal{V} (et éventuellement de la frontière Γ) est nécessaire et plusieurs directions sont définies.

5.3.3 Interpolation directionnelle avec plusieurs directions

Afin de respecter l'hypothèse (5.25) et faire « fonctionner » l'interpolation directionnelle, il peut être judicieux d'introduire plusieurs directions permettant de définir plusieurs ondes planes et plusieurs résidus. Des subdivisions du domaine \mathcal{V} et de la frontière Γ sont définies afin de générer ces directions. Des éléments notés $\mathcal{E}_\mathcal{V}$ et Γ_n constituent respectivement les subdivisions de \mathcal{V} et de Γ . Ces décompositions seront précisées dans la section 5.4. Pour chaque couple $(\mathcal{E}_\mathcal{V}, \Gamma_n)$, une direction notée c_n est associée ainsi qu'une fonction résidu, notée G_c^n , similaire au cas où il n'y avait qu'une seule direction (5.22), définie par

$$G_c^n(x, y) = \frac{e^{ik(\|x-y\| - \langle c_n, x-y \rangle)}}{4\pi \|x - y\|}, \quad (5.26)$$

où $x \in \mathcal{E}_\mathcal{V}$ est un point du domaine à visualiser \mathcal{V} , $y \in \Gamma_n$ est un point de la frontière Γ et c_n la direction associée à ce couple (x, y) .

De la même manière que précédemment, ce sont ces quantités qui sont approchées par interpolation. L'approximation de u est définie à l'aide de ces interpolations. Pour cela, la frontière Γ est décomposée en plusieurs portions notées Γ_n associées aux directions c_n . Il vient que

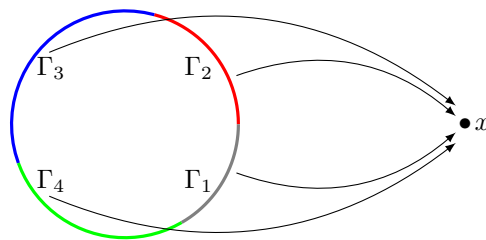
$$u(x) \approx \sum_{n=1}^{N_\Gamma} \sum_{j=1}^{N_\mathcal{V}} \langle T_{\Gamma_n}, G(\xi_j, y) \rangle \mathcal{L}_j(x) e^{ik\langle c_n, x - \xi_j \rangle}, \quad \forall x \in \mathcal{V}, \quad (5.27)$$

où N_Γ est le nombre de subdivision de Γ , $N_\mathcal{V}$ le nombre de degrés de liberté et T_{Γ_n} est la distribution dont le support est Γ_n . Notons que le nombre de degrés de liberté $N_\mathcal{V}$ ne dépend pas du découpage de Γ et donc ne dépend pas de N_Γ . L'approximation peut également s'écrire comme

$$u(x) \approx \sum_{n=1}^{N_\Gamma} \sum_{j=1}^{N_\mathcal{E}} u_{|\Gamma_n}(\xi_j) \mathcal{L}_j(x) e^{ik\langle c_n, x - \xi_j \rangle}, \quad \forall x \in \mathcal{V}, \quad (5.28)$$

où $u_{|\Gamma_n}(\xi_j)$ est le champ rayonné partiel calculé en un point d'interpolation (calculé par convolution sur la portion $\Gamma_n \subset \Gamma$). La figure 5.6 illustre un découpage possible pour une frontière donnée. Le rayonnement en un point x quelconque est calculé à partir des portions Γ_n , où $n = 1, \dots, 4$, pour cet exemple.

Il est très important de noter que pour un nombre de degrés de liberté donné, le coût de calcul en un point avec une formule de représentation est le même avec une ou plusieurs directions. En effet, en un point d'interpolation donné, la somme de toutes les contributions reste constante puisque chaque intégration ne se fait à chaque fois que sur une partie de la frontière et donc la somme correspond à la discrétisation totale de Γ .

FIGURE 5.6 – Un exemple de découpage de la frontière Γ .

5.4 Construction de champs rayonnés

Afin de reconstruire les champs rayonnés efficacement, et en particulier pour profiter de l'interpolation multi-directionnelle, une subdivision du domaine \mathcal{V} (en plus de celle de Γ) peut être effectuée. Cela permet de respecter l'hypothèse (5.25) et faire fonctionner l'interpolation directionnelle.

5.4.1 Subdivision du domaine à visualiser

Un maillage de \mathcal{V} , noté $\mathcal{T}(\mathcal{V})$, est construit et est composé d'éléments notés $\mathcal{E}_{\mathcal{V}}$ recouvrant le domaine

$$\bigcup_{\mathcal{E}_{\mathcal{V}} \in \mathcal{T}(\mathcal{V})} \mathcal{E}_{\mathcal{V}} = \mathcal{V}, \quad (5.29)$$

tel qu'il n'y ait pas de recouvrements entre les éléments. Sur chaque élément $\mathcal{E}_{\mathcal{V}}$, un champ local est reconstruit avec des directions et des fonctions de base locales. Outre la possibilité de pouvoir utiliser l'approche multi-directionnelle, d'autres raisons justifient la construction d'un découpage de \mathcal{V} . En effet, le domaine à visualiser est potentiellement très large et sans ce découpage, des ordres excessivement élevés seraient nécessaires. Or, ceci peut entraîner des problèmes lors de l'interpolation avec le phénomène de Runge (oscillations de l'approximation). De plus, créer un maillage permet de faire varier les ordres d'interpolation en fonction de la richesse de la solution (ou du noyau) à approcher.

Nous supposons que \mathcal{V} est une boîte, c'est à dire un rectangle en dimension deux ou un parallélépipède en dimension trois, où chaque arête est ainsi parallèle à un des axes. Dans le cas où le domaine est plus complexe, celui-ci peut être englobé par une telle boîte ou par une union de boîtes. Dans le cas où le domaine est composé de plusieurs boîtes, la méthodologie est appliquée à chacune d'entre elle indépendamment des autres. Dans la suite, nous considérons donc que \mathcal{V} est une seule boîte et nous cherchons à visualiser le champ rayonné dans cette zone où le champ est diffracté par un objet dont la frontière est notée Γ .

5.4.2 Subdivision de la frontière de l'objet diffractant

En plus de la subdivision du domaine \mathcal{V} , une subdivision de Γ peut être établie afin de respecter les hypothèses sur les directions (5.25). La frontière Γ est décomposée en $\Gamma = \bigcup_{\Gamma_n} \Gamma_n$, comme dans (5.28), tel qu'il n'y ait pas de recouvrements entre les éléments Γ_n . Ce découpage de la frontière est associé à chaque élément $\mathcal{E}_{\mathcal{V}}$. Cela signifie que le domaine \mathcal{V} est composé d'éléments $\mathcal{E}_{\mathcal{V}}$ et pour chaque élément $\mathcal{E}_{\mathcal{V}}$, des éléments Γ_n qui viennent d'un découpage de Γ

lui sont associés. Cela permet d'avoir une décomposition de la frontière adaptée à la zone du domaine \mathcal{V} considérée.

Afin de décomposer la frontière Γ , l'idée est de définir un maillage d'une boîte englobant celle-ci. Nous supposons donc qu'il existe une boîte \mathcal{G} qui contient Γ , ce qui est le cas car Γ est la frontière d'un domaine borné. Dans le cas où la géométrie est plus complexe, une union de boîtes pourrait être considérée et dans cette configuration, des directions pourraient alors être associées à chacune des boîtes, qui seraient également découpées. Pour le moment et afin de rester dans un cadre didactique, nous considérons qu'une seule boîte \mathcal{G} peut contenir tout Γ . La figure 5.7 illustre la problématique.

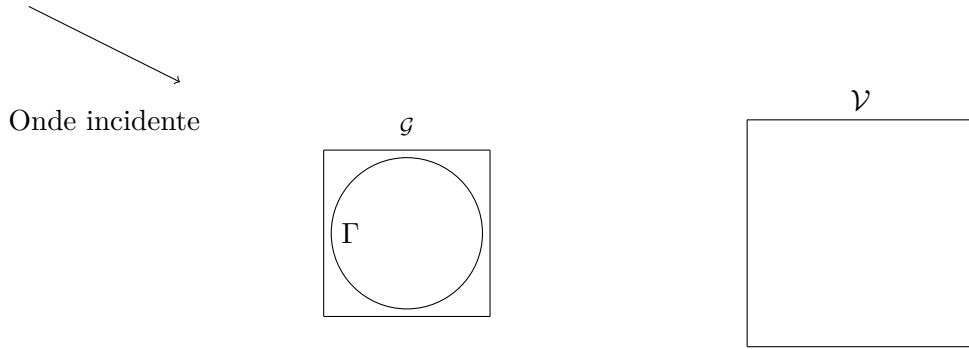


FIGURE 5.7 – Notations et problématique

Pour chaque élément $\mathcal{E}_\mathcal{V} \in \mathcal{T}(\mathcal{V})$, une subdivision de \mathcal{G} est construite telle que ces éléments notés $\mathcal{B}_\mathcal{E}$, vérifient

$$\bigcup_{\mathcal{B}_\mathcal{E} \subset \mathcal{G}} \mathcal{B}_\mathcal{E} = \mathcal{G}.$$

L'indice \mathcal{E} de $\mathcal{B}_\mathcal{E}$ montre la dépendance avec l'élément $\mathcal{E}_\mathcal{V}$ considéré. La frontière Γ peut être décomposée, en suivant la subdivision de \mathcal{G} , telle que

$$\Gamma = \bigcup_{\Gamma_n \subset \mathcal{B}_\mathcal{E}} \Gamma_n,$$

et

$$\Gamma_n = \Gamma \cap \mathcal{B}_\mathcal{E}.$$

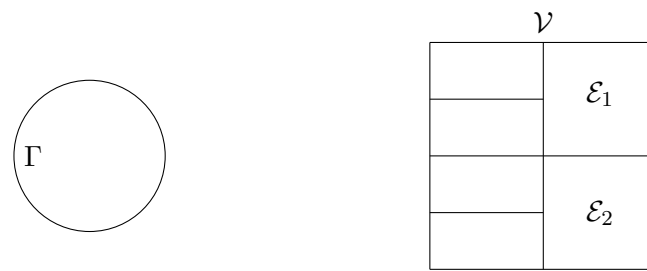
Nous illustrons sur la figure 5.8 les décompositions de \mathcal{V} et de Γ . En particulier, nous nous intéressons à deux éléments \mathcal{E}_1 et \mathcal{E}_2 de \mathcal{V} et montrons que les décompositions changent suivant l'élément considéré. Notons d'ailleurs que le nombre de portions peut également varier puisqu'il y a $N_\Gamma = 4$ portions dans le premier cas et $N_\Gamma = 3$ dans le second.

Pour chaque portion Γ_n incluse dans une boîte $\mathcal{B}_\mathcal{E}$, une direction c_n est définie à partir des centres x_c et y_c respectivement de $\mathcal{E}_\mathcal{V}$ et $\mathcal{B}_\mathcal{E}$, c'est-à-dire :

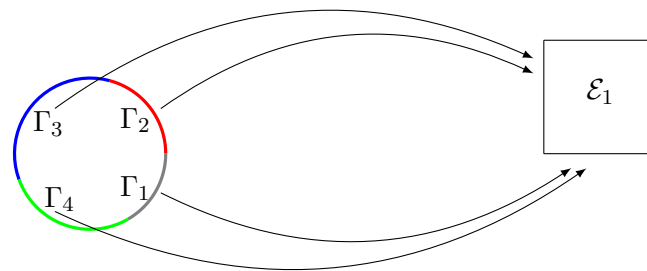
$$c_n = \frac{x_c - y_c}{\|x_c - y_c\|}. \quad (5.30)$$

5.4.3 Critères de construction des champs rayonnés

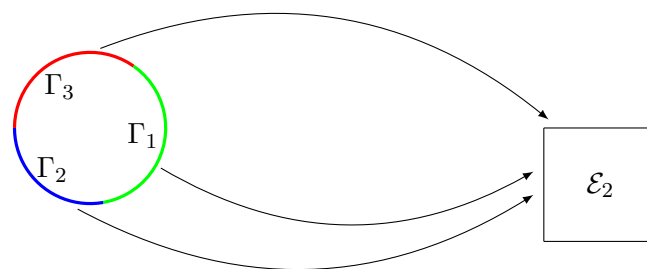
Le but est donc de construire des subdivisions de \mathcal{V} et de \mathcal{G} afin que l'interpolation directionnelle soit efficace. Or, pour un couple $(\mathcal{E}_\mathcal{V}, \mathcal{B}_\mathcal{E})$, d'après (5.25), la direction c_n donnée par



(a) Une subdivision du domaine \mathcal{V} à l'aide d'éléments \mathcal{E}_ν



(b) Une subdivision de la frontière associée à un élément \mathcal{E}_1



(c) Une subdivision de la frontière associée à un élément \mathcal{E}_2

FIGURE 5.8 – Illustration des subdivisions de la frontière Γ associées aux élément \mathcal{E}_ν de \mathcal{V} .

(5.30) est judicieuse si

$$c \approx \frac{x - y}{\|x - y\|}, \quad \forall x \in \mathcal{E}_\nu, \forall y \in \mathcal{B}_\varepsilon. \quad (5.31)$$

Notre approche est donc basée sur ces subdivisions et sur l'introduction d'ondes planes. Afin d'interpoler correctement les résidus, des polynômes de Lagrange d'ordre élevé sont également utilisés. Les valeurs aux points d'interpolation sont toujours calculées via les formules de représentation et ensuite une solution locale sur chaque élément du maillage de \mathcal{V} est construite. La génération du maillage et le choix des points à faire rayonner sont faits *a priori*, c'est à dire indépendamment du calcul des degrés de liberté via les formules de représentation. La figure 5.9 récapitule la méthodologie mise en place.

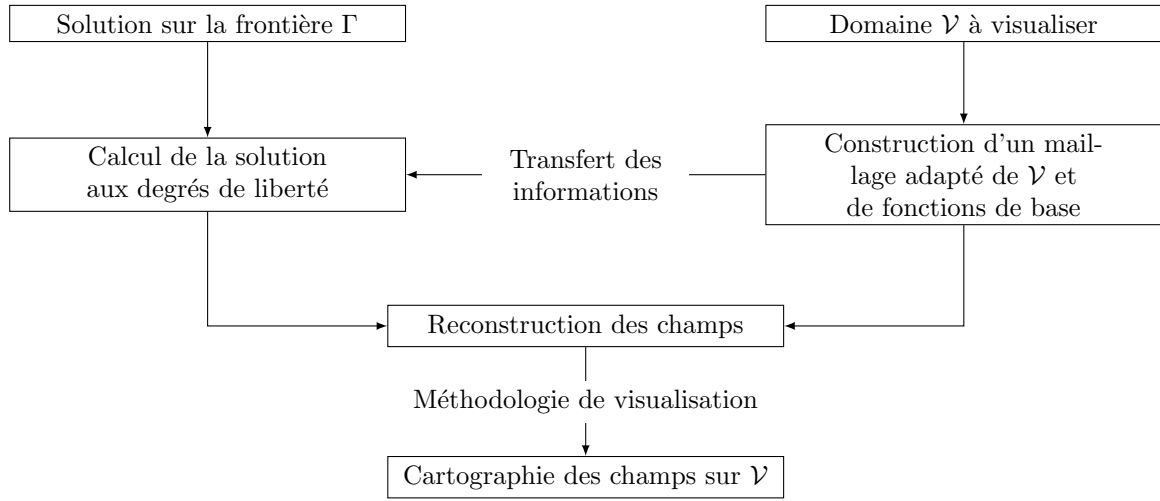


FIGURE 5.9 – Schéma général du rayonnement

Il y a plusieurs paramètres à prendre en compte lors de la création de ces maillages. Le nombre le plus important à contrôler est le nombre de points à faire rayonner, c'est à dire le nombre de points où les formules de représentation sont utilisées afin de calculer ponctuellement la solution. Ce nombre dépend du nombre d'éléments dans le maillage de \mathcal{V} et du nombre de points à faire rayonner dans chaque élément. En effet, une fois la subdivision de \mathcal{V} générée, les directions sont définies, et il reste à approcher le résidu à l'aide de polynômes éventuellement d'ordre élevé. Des points d'interpolation et des fonctions de base sont définis afin d'approcher ce résidu. Ceci se traduit par l'assignation d'un ordre suivant chaque direction afin de savoir le nombre de points nécessaires. Pour cela, on raisonne en dimension un puis on tensorise. La maîtrise du nombre de points d'interpolation est un point capital. Le nombre de points d'interpolation, qui correspond au nombre de degrés de liberté, à contrôler est égal à

$$\begin{aligned}
 N_{ddl} &= \sum_{\mathcal{E}_\nu \in \mathcal{T}(\mathcal{V})} N_1^{\mathcal{E}_\nu} N_2^{\mathcal{E}_\nu}, & \text{si } d = 2, \\
 N_{ddl} &= \sum_{\mathcal{E}_\nu \in \mathcal{T}(\mathcal{V})} N_1^{\mathcal{E}_\nu} N_2^{\mathcal{E}_\nu} N_3^{\mathcal{E}_\nu}, & \text{si } d = 3,
 \end{aligned} \quad (5.32)$$

où $N_i^{\mathcal{E}_\nu}$ correspond au nombre de points d'interpolation local à l'élément \mathcal{E}_ν dans la direction i .

Dans une moindre mesure, il est nécessaire de contrôler le nombre de découpages de \mathcal{G} . Au lieu d'avoir un découpage unique de \mathcal{G} , qui ferait augmenter le nombre de découpages et de directions afin de respecter les critères d'admissibilité de chaque couple, nous décidons d'associer un découpage de \mathcal{G} à chaque élément $\mathcal{E}_\nu \in \mathcal{T}(\mathcal{V})$.

5.4.4 Visualisation des champs reconstruits

Le travail mis en place dans ce chapitre est possible car nous avons à notre disposition un outil permettant de visualiser les champs rayonnés reconstruits. En effet, un maillage $\mathcal{T}(\mathcal{V})$ du domaine à visualiser est construit et sur chaque élément \mathcal{E}_ν du maillage, d'après (5.28), un champ reconstruit local noté $u_{rad}^{\mathcal{E}_\nu}$ est défini et s'écrit

$$u_{rad}^{\mathcal{E}_\nu}(x) = \sum_{i=1}^{N_{\mathcal{E}} \cdot N_{\Gamma}^{\mathcal{E}}} u_i^{\mathcal{E}_\nu} \phi_i^{\mathcal{E}_\nu}(x), \quad \forall x \in \mathcal{E}_\nu \subset \mathcal{V}, \quad (5.33)$$

où $u_i^{\mathcal{E}_\nu}$ sont les degrés de liberté et les fonctions de base $\phi_i^{\mathcal{E}_\nu}$ sont composées de polynômes d'ordre élevé et d'ondes planes comme dans (5.28). Le champ est donc de la même forme que le cadre prescrit dans la section 2.2 avec l'équation (2.1). La méthode de visualisation explicitée dans les chapitres précédents peut donc être utilisée. Les fonctions de base φ_i^K dans (2.1) sont définies sur la cellule de référence de l'élément tandis que les fonctions de base $\phi_i^{\mathcal{E}_\nu}$ de (5.33) sont définies directement sur l'élément. La relation entre ces fonctions est la suivante

$$\phi_i^{\mathcal{E}_\nu}(x) = (\varphi_i^K \circ g_K^{-1})(x), \quad \forall x \in \mathcal{E}_\nu, \quad (5.34)$$

où g_K est la transformation géométrique entre la cellule de référence et l'élément noté \mathcal{E}_ν ou K .

5.5 Un algorithme pour reconstruire des champs rayonnés

Nous proposons une méthodologie pour générer la subdivision de \mathcal{V} et pour chaque élément $\mathcal{E}_\nu \in \mathcal{T}(\mathcal{V})$, la subdivision de \mathcal{G} associée dans l'algorithme 9. Un critère est défini pour décider si un raffinement est nécessaire ou non. Nous fixons une constante arbitraire $R > 0$ et limitons les oscillations du noyau sur chaque élément en contrôlant le ratio entre les longueurs de l'élément et une longueur d'onde modifiée. Le but est de trouver le plus grand ratio noté τ_{\max} d'un certain couple $(\mathcal{E}_\nu^{\max}, \mathcal{B}_\mathcal{E}^{\max})$. Le calcul de ces ratios est donné dans la section 5.5.1 consacrée aux critères d'admissibilités. Puis, la façon dont les éléments sont découpez est précisée dans la section 5.5.2.

Algorithme 9 : Subdivision de \mathcal{V} and assignation des ordres

Initialiser le critère R , $\mathcal{E}_\nu = \mathcal{V}$ et $\mathcal{B}_\mathcal{E} = \mathcal{G}$.

Calculer τ_{\max} correspondant au couple $(\mathcal{E}_\nu^{\max}, \mathcal{B}_\mathcal{E}^{\max}) = (\mathcal{E}_\nu, \mathcal{B}_\mathcal{E})$.

tant que $\tau_{\max} > R$ **faire**

 | Subdiviser $(\mathcal{E}_\nu^{\max}, \mathcal{B}_\mathcal{E}^{\max})$.

 | Calculer τ_{\max} pour un nouveau couple $(\mathcal{E}_\nu^{\max}, \mathcal{B}_\mathcal{E}^{\max})$.

fin

pour chaque élément $\mathcal{E}_\nu \in \mathcal{T}(\mathcal{V})$ **faire**

 | Assigner les ordres d'interpolation dans chaque direction.

fin

La construction des découpages de \mathcal{V} et de \mathcal{G} est faite simultanément. La modification d'un seul découpage (sur \mathcal{V} ou \mathcal{G}) à la fois, c'est-à-dire sans changer le second, semble périlleuse puisqu'il n'est pas possible de maîtriser l'interpolation directionnelle en ne jouant que sur une subdivision. Ainsi, cela demanderait de nombreuses itérations entre les deux maillages pour les construire.

5.5.1 Critère d'admissibilité

Deux cas doivent être considérés suivant qu'il y ait des directions ou non.

Critère d'admissibilité sans direction

Considérons tout d'abord le cas sans direction et en particulier le critère d'admissibilité pour un élément arbitraire. La longueur d'onde est notée λ et les longueurs de l'élément sont notées L_i suivant chaque direction où $i = 1, \dots, d$ et d est la dimension. Nous définissons le ratio τ comme

$$\tau = \max_{i=1, \dots, d} L_i / \lambda. \quad (5.35)$$

Plus ce ratio est élevé, plus la fonction est oscillante dans le domaine considéré. On définit le critère d'admissibilité :

$$\begin{aligned} \mathcal{E}_{\mathcal{V}} \text{ admissible} &\Leftrightarrow \tau = \frac{\max(L_1, L_2)}{\lambda} \leq R, & \text{si } d = 2. \\ \mathcal{E}_{\mathcal{V}} \text{ admissible} &\Leftrightarrow \tau = \frac{\max(L_1, L_2, L_3)}{\lambda} \leq R, & \text{si } d = 3. \end{aligned} \quad (5.36)$$

Critère d'admissibilité avec directions

Dans le second cas, lorsque l'interpolation directionnelle est utilisée, pour chaque couple $(\mathcal{E}_{\mathcal{V}}, \mathcal{B}_{\mathcal{E}})$, une direction c_n est introduite et nous définissons le ratio $\tilde{\tau}$ comme

$$\tilde{\tau} = \max_{i=1, \dots, d} L_i / \tilde{\lambda}, \quad (5.37)$$

où

$$\tilde{\lambda} = \frac{2\pi}{\tilde{k}},$$

et \tilde{k} est défini par

$$\tilde{k} = k \max_{(x,y) \in (\mathcal{E}_{\mathcal{V}}, \mathcal{B}_{\mathcal{E}})} \left| 1 - \left\langle c_n, \frac{x-y}{\|x-y\|} \right\rangle \right|. \quad (5.38)$$

La définition de \tilde{k} est donc équivalente à un nombre d'onde modifié. En effet, le résidu est alors équivalent à

$$\frac{e^{i\tilde{k}r}}{4\pi r},$$

où r est la distance entre deux points donnés. Le ratio $\tilde{\tau}$ quantifie les oscillations après l'introduction des directions. On définit le critère d'admissibilité avec directions comme

$$\begin{aligned} \mathcal{E}_{\mathcal{V}} \text{ admissible} &\Leftrightarrow \forall \mathcal{B}_{\mathcal{E}} \subset \mathcal{G}, \tilde{\tau} = \frac{\max(L_1, L_2)}{\tilde{\lambda}} \leq R, & \text{si } d = 2. \\ \mathcal{E}_{\mathcal{V}} \text{ admissible} &\Leftrightarrow \forall \mathcal{B}_{\mathcal{E}} \subset \mathcal{G}, \tilde{\tau} = \frac{\max(L_1, L_2, L_3)}{\tilde{\lambda}} \leq R, & \text{si } d = 3. \end{aligned} \quad (5.39)$$

5.5.2 Subdivision du domaine

La construction des subdivisions de \mathcal{V} et de \mathcal{G} est basée sur une partition hiérarchique du domaine et donc une structure d'arbre est utilisée. Le niveau 0 correspond au domaine complet \mathcal{V} . Le niveau (*level* en anglais) $l + 1$ de l'arbre est obtenu à partir du niveau l en subdivisant une feuille en 2 feuilles de tailles égales. Pour chaque feuille, la direction de découpe, suivant la première, deuxième ou troisième direction, est choisie afin que le maximum des nouveaux ratios $\tilde{\tau}$ dans la nouvelle configuration soit le plus faible. Pour chaque élément $\mathcal{E}_\nu \subset \mathcal{V}$, une subdivision de \mathcal{G} est définie. Pour un couple donné, 4 (resp. 9) configurations (voir figures 5.10 et 5.11) sont testées en dimension 2 (resp. 3) puisque le nombre de directions possibles pour chaque feuille dépend de la direction (2 ou 3). L'algorithme 10 récapitule comment choisir la configuration de découpe. Les feuilles de la configuration choisie sont subdivisées en deux feuilles de tailles égales. La figure 5.12 montre les premières étapes de subdivisions possibles. On peut voir qu'après ces découpages successifs, l'hypothèse (5.25) devient vérifiée.

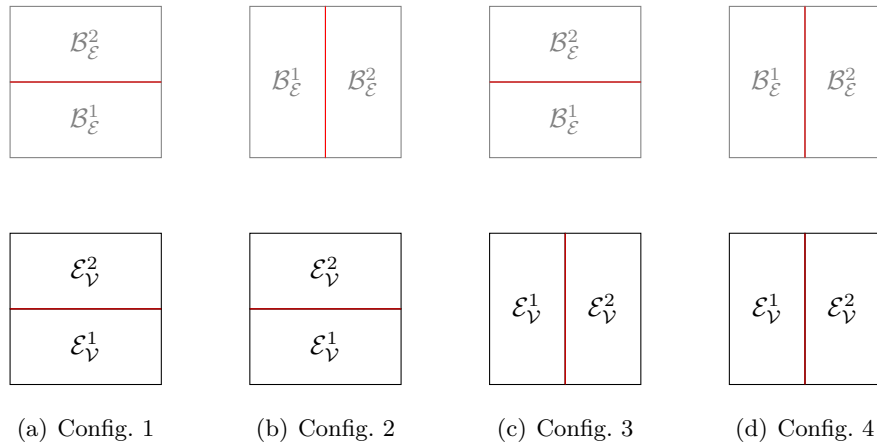


FIGURE 5.10 – Les 4 configurations de subdivisions en dimension deux

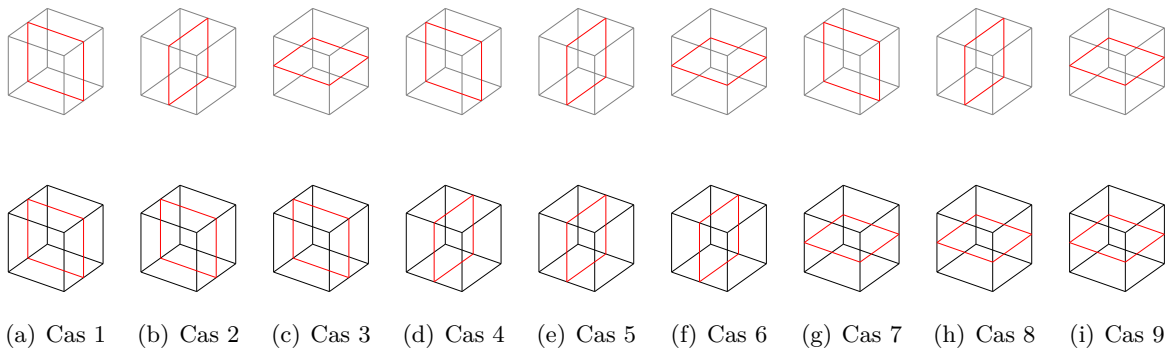


FIGURE 5.11 – Les 9 configurations de subdivisions en dimension trois

Lorsque le critère d'admissibilité est vérifié pour tous les éléments (ratio majoré par R), un maillage de \mathcal{V} est construit en prenant toutes les feuilles (éléments sans enfant) de l'arbre de \mathcal{V} .

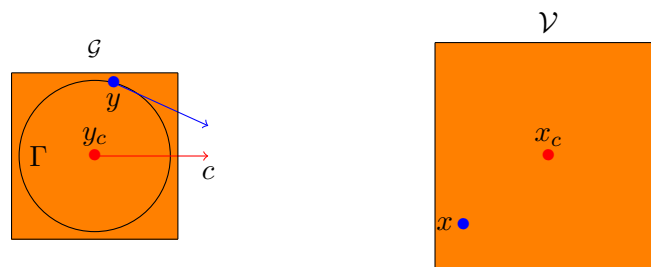
Algorithme 10 : Choix de la configuration de découpe

pour toutes les configurations **faire**

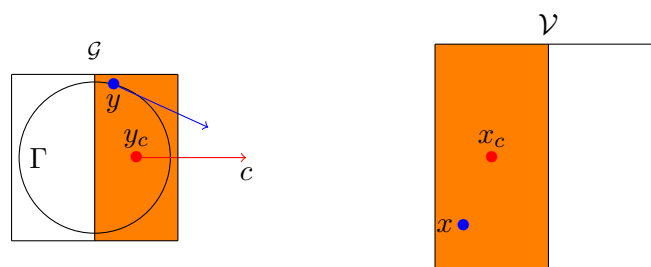
| Calculer le maximum des ratios ((5.35) ou (5.37)) des 4 couples de la configuration.

fin

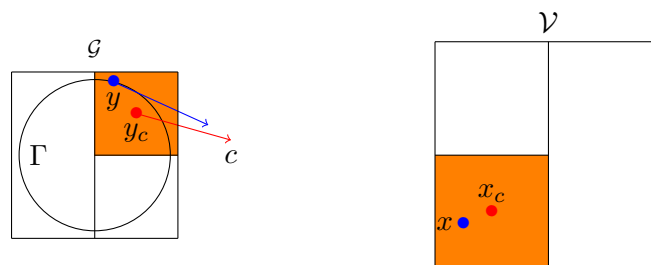
Choisir la configuration minimisant ces maxima.



(a) Configuration initiale



(b) Après une subdivision



(c) Après deux subdivisions

FIGURE 5.12 – Illustration des premières étapes de subdivisions (le couple à subdiviser est mis en évidence en orange).

La figure 5.13 illustre des exemples de subdivisions.

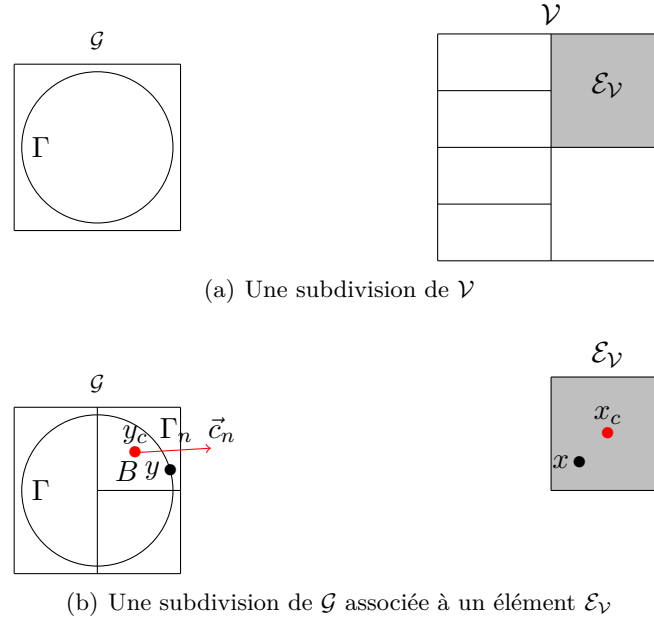


FIGURE 5.13 – Exemples de subdivisions du domaine à visualiser \mathcal{V} (a) et de la frontière \mathcal{G} (b).

5.5.3 Assignment des ordres d'interpolation.

La dernière étape dans l'algorithme 9 est l'assignation des ordres d'interpolation et donc le choix des points d'interpolation pour chaque élément. Un maillage de \mathcal{V} a été construit. Nous considérons un élément $\mathcal{E}_\mathcal{V} \subset \mathcal{V}$. Une décomposition de la frontière Γ lui est associée à partir des éléments $\mathcal{B}_\mathcal{E} \subset \mathcal{G}$. De la même manière que pour le critère d'admissibilité, nous définissons les ratios

$$\bar{\tau}_i = \frac{L_i}{\bar{\lambda}}, \quad i = 1, \dots, d,$$

où L_i la taille de l'élément dans la direction i et $\bar{\lambda}$ est défini de la façon suivante

$$\bar{\lambda} = \frac{2\pi}{\kappa_\mathcal{E}}.$$

Le paramètre $\kappa_\mathcal{E}$ est local à un élément $\mathcal{E}_\mathcal{V}$ et équivalent à un nombre d'onde modifié, comme \tilde{k} dans (5.38), et doit prendre en compte toutes les directions générées par la décomposition de la frontière

$$\kappa_\mathcal{E} = k \sup_{\{\mathcal{B}_\mathcal{E}\}} \max_{(x,y) \in (\mathcal{E}_\mathcal{V}, \mathcal{B}_\mathcal{E})} \left| 1 - \left\langle c, \frac{x-y}{\|x-y\|} \right\rangle \right|.$$

Les ratios $\bar{\tau}_i$ permettent d'estimer l'oscillation de la fonction à interpoler dans chaque direction et sont utilisés pour fixer les ordres des fonctions de base. Pour cela, des règles *a priori* fixent les ordres à assigner. Un problème en dimension un est résolu pour les définir. La méthode est très performante grâce à ces règles *a priori* puisque leur utilisation est très facile et rapide.

Résolution d'un problème d'interpolation 1D

Dans [Bériot et al., 2016], les auteurs utilisent également des règles *a priori* : des valeurs tabulées indiquent quel ordre prendre en fonction de la taille de l'élément et du nombre d'onde. Dans notre cas, nous voulons en fonction des ratios $\bar{\tau}$ savoir quel ordre choisir pour atteindre une précision donnée. Nous considérons le problème 1D suivant :

pour une tolérance donnée $\varepsilon > 0$, trouver le plus petit entier naturel N tel que

$$\|f - \tilde{f}_N\|_{\infty, [0,1]} \leq \varepsilon, \quad (5.40)$$

où $f(x) = \sin(2\pi\tau x)$, τ est le ratio et $\tilde{f}_N(x) = \sum_{j=1}^{N+1} f(\xi_j^N) L_j^N(x)$. Pour chaque ordre N , les $N+1$ points de Chebyshev sont notés $(\xi_j^N)_{j=1, \dots, N+1}$ (présentés en annexe B.1) et les polynômes de Lagrange associés sont notés L_j^N . Un problème sur le segment $[0, 1]$ est donc considéré, la fonction à interpoler f est une fonction sinusoidale dont l'oscillation dépend de τ . Ce choix est motivé afin d'obtenir une bonne restitution de la partie oscillante. L'approximation de f est notée \tilde{f}_N et si (5.40) est vérifié, cela assure un contrôle ponctuel sur l'erreur d'interpolation pour une précision demandée (par exemple $\varepsilon = 1\%$). La figure 5.14 donne le lien entre l'ordre nécessaire et le ratio pour avoir une précision de 1%. Quelques règles sont reportées dans le tableau 5.3.

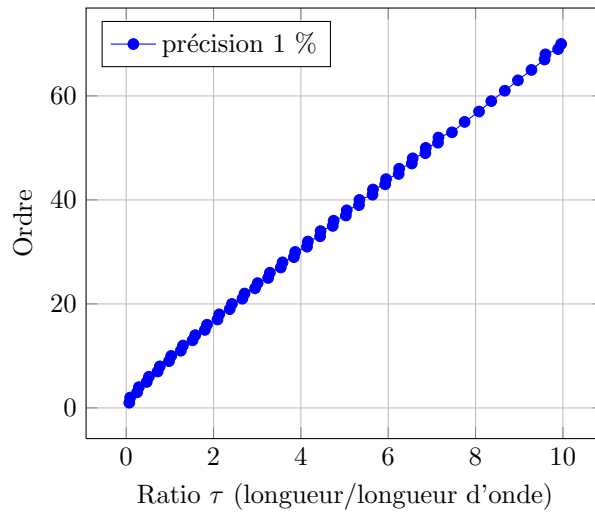


FIGURE 5.14 – Règles pour fixer l'ordre pour une précision donnée selon le ratio longueur / longueur d'onde à 1 %

TABLE 5.3 – Règles pour fixer l'ordre pour une précision donnée selon le ratio longueur / longueur d'onde.

| Ratio τ | 1/4 | 1/3 | 1/2 | 3/4 | 1 | 3/2 | 2 | 3 |
|-------------------------------|-----|-----|-----|-----|----|-----|----|----|
| Ordre ($\varepsilon = 1\%$) | 3 | 5 | 6 | 8 | 10 | 13 | 17 | 24 |

5.6 Validations numériques des champs rayonnés reconstruits.

Plusieurs cas sont exhibés afin de valider numériquement la méthode proposée. Nous commençons par exposer l'intérêt de l'ajout d'une direction. Pour cela, le domaine à visualiser n'est pas découpé et une seule direction est introduite. Nous montrons que l'interpolation directionnelle, même avec une seule direction, est très intéressante sous certaines configurations. Nous traitons ensuite des problèmes en dimension deux à l'aide de notre méthode de construction adaptative. Enfin, nous considérons des problèmes en dimension trois où des rayonnements sont surfaciques puis volumiques. Dans toutes les configurations, nous montrons que l'utilisation d'interpolations d'ordre élevé réduit le nombre de degré de liberté et que l'interpolation directionnelle permet de le réduire encore plus. Il est important d'avoir une solution compressée (i.e. avec peu de degrés de liberté pour qu'elle soit performante) et précise (i.e. approchant correctement le champ rayonné calculé par la formule de représentation).

5.6.1 Interpolation directionnelle avec une seule direction

Le premier cas concerne la propagation d'une onde avec diffraction autour d'un disque. Le problème de Helmholtz avec une condition de Dirichlet est traité et s'écrit comme

$$\begin{cases} \Delta u + k^2 u = 0, & \text{dans } \mathbb{R}^d \setminus \overline{\Omega}_s, \\ u = 0, & \text{sur } \Gamma = \partial \overline{\Omega}_s. \end{cases} \quad (5.41)$$

Une condition de radiation de Sommerfeld est utilisée à l'infini pour que le problème soit bien défini [Nédélec, 2001]. Nous souhaitons montrer sur ce cas l'intérêt de l'interpolation directionnelle. Pour cela, on se place dans une configuration très simple comme le montre la figure 5.15 et une seule direction est introduite. Les paramètres de la simulation sont les suivants :

- Obstacle : Disque de rayon 0.9 centré à l'origine.
- Nombre d'onde $k = 6\pi$, longueur d'onde $\lambda = 1/3$.
- Taille du domaine à visualiser : $3\lambda \times 3\lambda$.

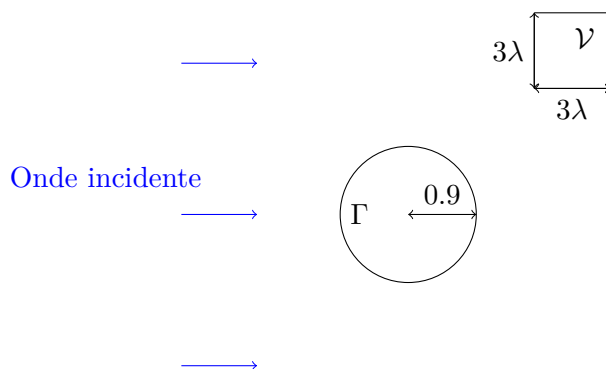


FIGURE 5.15 – Interpolation directionnelle avec une seule direction : configuration.

L'onde incidente considérée est plane et vient de la gauche. Nous choisissons de fixer le nombre d'onde à $k = 6\pi$ pour que le noyau de Green considéré soit oscillant (voir figure 5.16.a). La longueur d'onde est donc égale à $\lambda = 1/3$. Les abscisses et ordonnées du domaine à visualiser

\mathcal{V} varient entre 5λ et 8λ , c'est à dire entre $5/3$ et $8/3$, les longueurs du domaine dans chaque direction sont donc égales à 3λ . Le champ diffracté de référence est représenté sur la figure 5.16.a. Cette solution de référence est obtenue en définissant une grille de visualisation très fine, en calculant en chacun des nœuds la solution par la formule de représentation et en effectuant une interpolation affine. Toutes les solutions de référence dans cette section sont obtenues de cette façon. Nous définissons une seule direction c à partir du centre du domaine à visualiser et de l'origine, qui correspond au centre du disque, on a donc

$$c = (4.5\lambda, 4.5\lambda). \quad (5.42)$$

Le domaine à visualiser n'est pas découpé puisque l'on veut montrer qu'ajouter simplement cette direction dans la reconstruction du champ rayonné permet d'améliorer sensiblement sa qualité. Les ordres d'interpolation sont prescrit en suivant la section 5.5.3 et sont fixés à 5 dans chaque direction : le problème est symétrique et les longueurs sont identiques. Ainsi, 36 points d'interpolation sont définis pour reconstruire le champ rayonné.

La figure 5.16 compare les résultats obtenus avec ou sans cette direction à partir des mêmes points d'interpolation. Lorsqu'il n'y a pas de direction (figure 5.16.b), le champ reconstruit est très loin de la solution de référence puisqu'il n'y a pas assez de points d'interpolation pour capter la richesse de l'information. L'erreur absolue ponctuelle, correspondant à l'écart entre le champ rayonné et son approximation en tout point, est affichée sur la figure 5.16.d et il y a une erreur relative pouvant atteindre plus de 111 % en certains points. Avec une unique direction, le champ obtenu avec ces mêmes points se trouve sur la figure 5.16.c et la solution est beaucoup plus précise (erreur sur la figure 5.16.e). Dans ce cas, l'erreur relative est au maximum égale à 3%.

Maintenant que nous avons montré l'intérêt de l'interpolation directionnelle en n'ajoutant qu'une seule direction, sans le processus de découpages, nous nous concentrons sur des cas plus réalistes.

5.6.2 BEM en 2D et rayonnements dans le plan

Nous nous intéressons maintenant à des problèmes en dimension deux. Un code équations intégrales en dimension deux, développé par Marc Bakry durant sa thèse [Bakry, 2016], est utilisé. Les frontières des objets sont donc maillés à l'aide de segments puisque la méthode des éléments finis de frontière permet de réduire la dimension du problème par un. Nous souhaitons, à partir de la solution calculée par le code équations intégrales et des formules de représentation, et en utilisant notre algorithme, reconstruire des champs rayonnés dans des domaines donnés en dimension deux. Nous considérons d'abord la diffraction d'une onde plane autour d'un disque puis l'objet considéré est plus complexe avec une forme dite de L ou *L-shape* en anglais.

5.6.2.1 Diffraction par un disque

Pour ce problème, nous reprenons quasiment les mêmes paramètres que dans la section 5.6.1 mais le domaine à visualiser est beaucoup plus large (voir figure 5.17). Nous souhaitons montrer l'intérêt du découpage mis en place et l'introduction de plusieurs directions. On considère là encore le problème de Helmholtz avec une condition de Dirichlet. Les paramètres sont les suivants :

- Obstacle : Disque de rayon 0.9 centré à l'origine.
- Nombre d'onde $k = 6\pi$, longueur d'onde $\lambda = 1/3$.

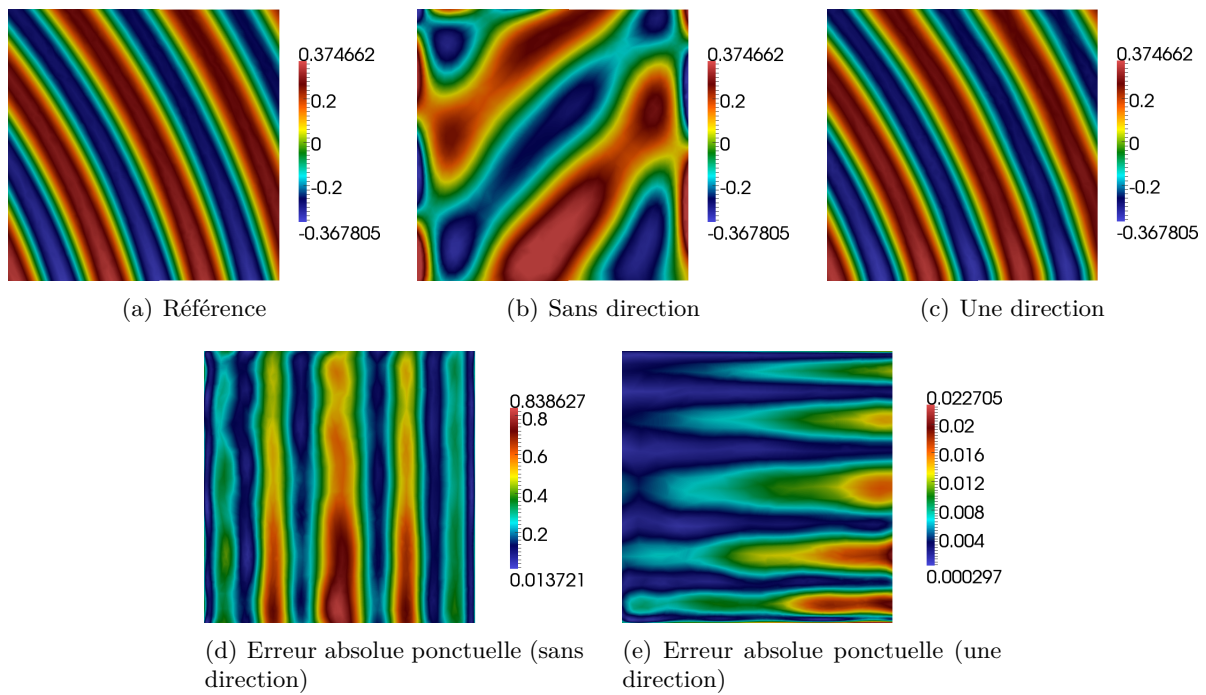


FIGURE 5.16 – Diffusions de champs diffractés (a, b et c) par un disque. Hormis la référence (a), les autres résultats sont obtenus avec des fonctions d'ordre 5 dans chaque direction. Comparaison de l'ajout d'une seule direction (c et e) au cas sans direction (b et d).

- Taille du domaine à visualiser : $12\lambda \times 24\lambda$.

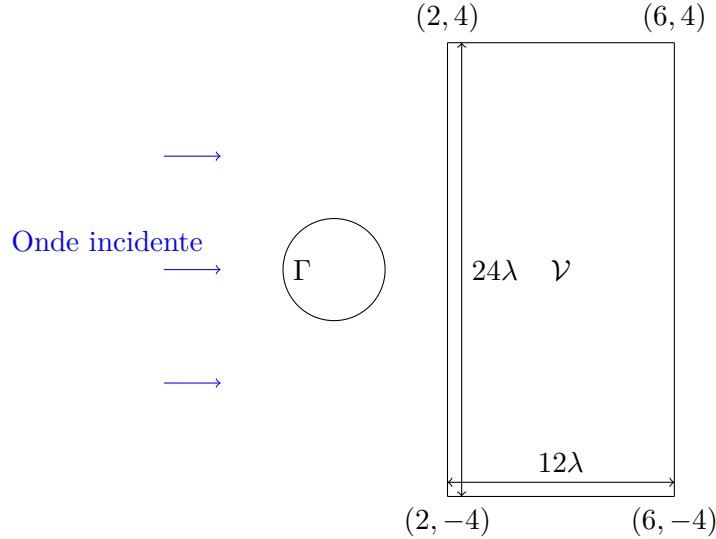


FIGURE 5.17 – Diffraction autour d'un disque : configuration.

Dans l'algorithme 9, un paramètre R permet de limiter les oscillations sur chaque élément. Pour ce cas, nous choisissons de le fixer comme $R = 0.1$. Ce nombre étant petit, cela signifie qu'il y aura beaucoup de découpages du domaine mais que sur chaque élément, avec les directions, le nombre de points d'interpolations nécessaire sera petit. La figure 5.18 montre la subdivision du domaine effectuée où 112 éléments sont définis et compare les ordres nécessaires, fixés en suivant la section 5.5.3, avec ou sans direction pour avoir une précision demandée à 1%. Lorsque l'approche directionnelle est utilisée, les ordres varient dans chaque direction entre 2 et 3 sur chaque élément et le nombre de points d'interpolation au total est égal à 1120. Sans direction, les ordres varient entre 9 et 26 dans chaque direction et pour chaque élément, et 32508 points d'interpolation sont nécessaires.

La figure 5.19 compare les visualisations de champs diffractés obtenues dans différentes configurations, certaines avec, d'autres sans direction. La figure 5.19.a est obtenue avec notre méthode adaptative où 1120 degrés de liberté sont utilisés. Pour la même subdivision et avec les mêmes degrés de liberté mais cette fois-ci sans direction, la figure 5.19.b permet de comprendre pourquoi l'approche directionnelle est intéressante. Sans direction, pour avoir une précision similaire, 32508 degrés de liberté sont nécessaires sur la figure 5.19.c. Un facteur 29 apparait et montre la compression de l'information obtenue grâce à l'interpolation directionnelle.

5.6.2.2 Diffraction autour d'un disque

Nous considérons le même type de problème mais avec deux changements notables : le nombre d'onde est plus élevé et le domaine de visualisation englobe l'obstacle. Les paramètres sont les suivants :

- Obstacle : Disque de rayon 1.0 centré à l'origine.
- Nombre d'onde $k = 25$, longueur d'onde $\lambda \approx 0.25$.

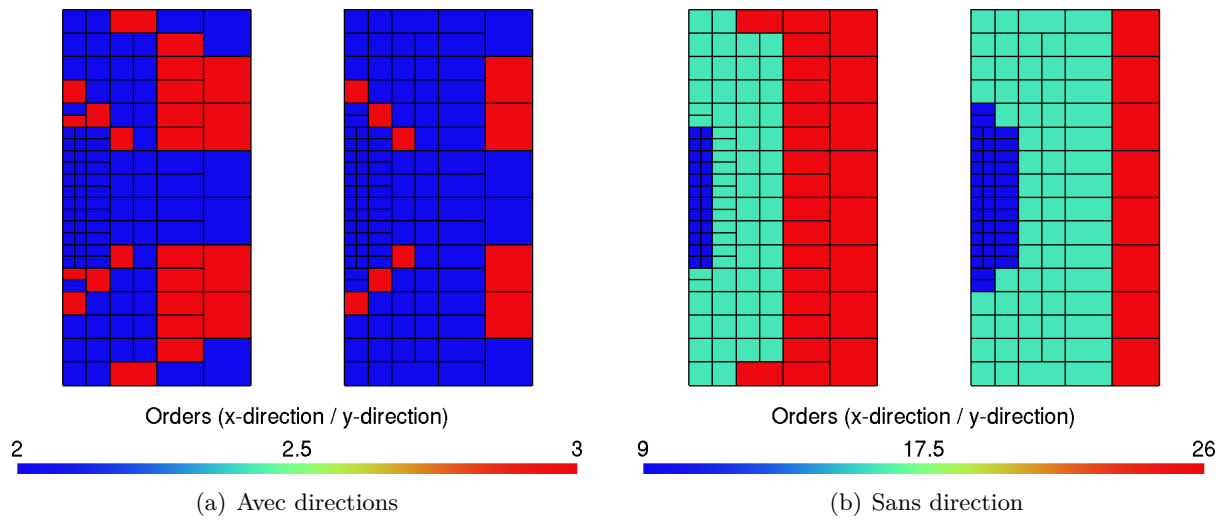


FIGURE 5.18 – Subdivision du domaine et comparaison des ordres nécessaires.

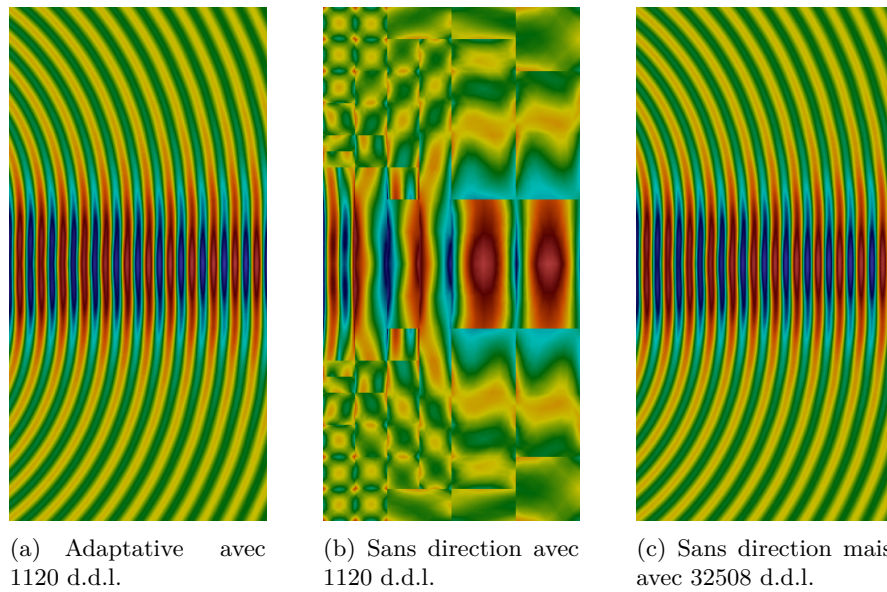


FIGURE 5.19 – Visualisation de champs diffractés. Comparaison de notre méthode avec le cas sans direction pour le même nombre d'éléments (a-b) et pour une précision similaire (a-c).

- Taille du domaine à visualiser : $8 \times 8 \approx 32\lambda \times 32\lambda$.

La configuration du problème est donnée par la figure 5.20. L'algorithme 9 est utilisé avec le paramètre $R = 1$. Le domaine est ainsi découpé en 620 éléments et les formules de représentation sont utilisées en 29268 points.

La spécificité du domaine à visualiser par rapport à précédemment est qu'il contient l'obstacle. La conséquence est que nous choisissons de le décomposer en deux parties :

- Une partie dite « proche » où l'interpolation directionnelle n'est pas utilisée.
- Une partie dite « lointaine » où des directions sont introduits et l'interpolation directionnelle est utilisée.

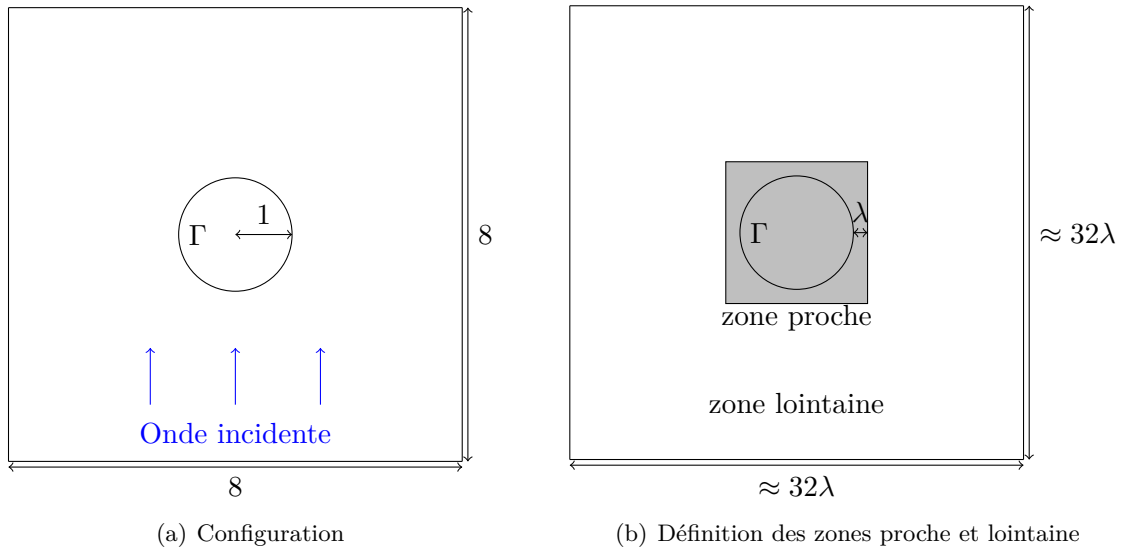


FIGURE 5.20 – Diffraction autour d'un disque

Dans la partie « lointaine », le comportement du champ rayonné s'approche de celui d'une onde plane et donc il est très intéressant d'utiliser l'interpolation directionnelle comme nous l'avons vu sur les exemples précédents. En outre, plus le domaine \mathcal{V} est loin de la frontière, plus l'approche directionnelle est efficace. C'est pour cette raison que les éléments les plus éloignés de la frontière sont de plus grandes tailles. Dans la partie « proche », mettre en place une interpolation directionnelle est beaucoup moins intéressante car afin d'avoir des directions pertinentes, il faut pour chaque couple $(\mathcal{E}_\mathcal{V}, \mathcal{B}_\mathcal{E})$ que la direction c vérifie

$$c \approx \frac{x - y}{\|x - y\|}, \quad \forall x \in \mathcal{E}_\mathcal{V}, \forall y \in \mathcal{B}_\mathcal{E}. \quad (5.43)$$

Pour que cette approximation soit vraie pour chaque couple considéré, dans le cas « proche », la seule possibilité est de découper énormément et donc d'introduire de très nombreuses directions. Dans le cas contraire, c'est-à-dire si l'approximation n'est pas vraie, il peut arriver que l'utilisation de directions fasse augmenter le nombre de points d'interpolation nécessaire par rapport au cas sans direction. Aussi, dans la zone proche, nous décidons de ne pas utiliser l'interpolation directionnelle mais nous utilisons des fonctions d'ordre élevé. Le critère d'admissibilité sans direction est utilisé et les ordres sont fixés en suivant la section 5.5.3. Nous

décidons que la zone est considérée comme lointaine lorsque la distance avec la frontière est plus grande qu'une longueur d'onde comme le montre la figure 5.20.b.

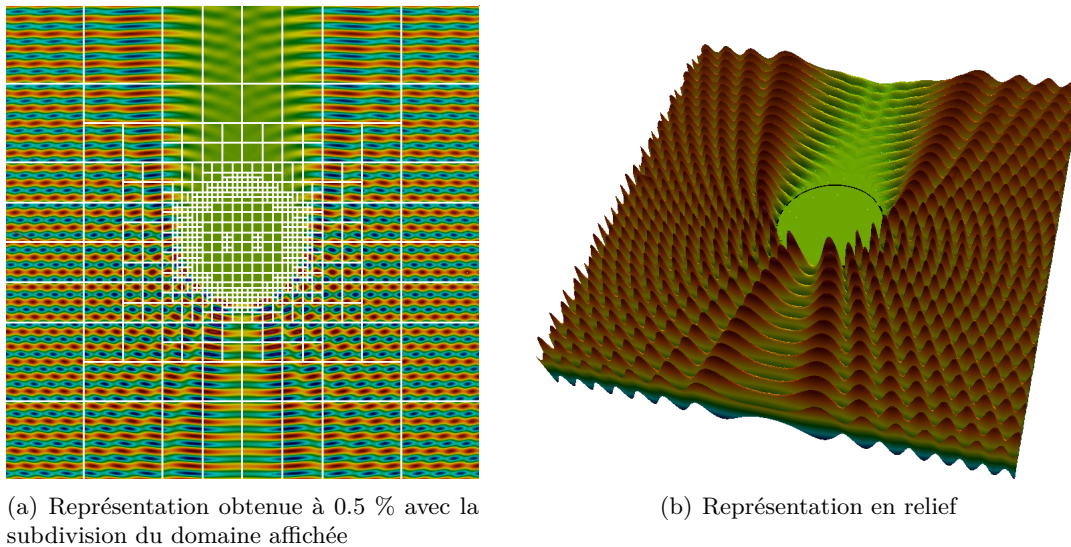


FIGURE 5.21 – Propagation d'une onde acoustique autour d'un disque pour un nombre d'onde élevé

La figure 5.21 donne la représentation obtenue pour une précision de 0.5 % avec notre méthode de visualisation. La subdivision du domaine est affichée sur la figure 5.21.a. Le maillage de représentation est composé de 427 962 triangles. La figure 5.21.b illustre bien le coté oscillant de l'onde sur cette représentation en relief.

5.6.2.3 Diffraction autour d'un objet L-shape

Nous considérons maintenant un cas plus réaliste avec un objet dont la frontière est en forme de L comme le montre la figure 5.22.a. Des zones proche et lointaine sont définies comme précédemment (voir figure 5.22.b). Les paramètres du problème sont les suivants :

- Obstacle en forme de L de longueur maximale 3λ .
- Nombre d'onde $k = 6\pi$, longueur d'onde $\lambda = 1/3$.
- Taille du domaine à visualiser : $12\lambda \times 12\lambda$.

La méthode adaptative est utilisée pour découper le domaine. Nous fixons le paramètre R à $R = 0.5$. Ainsi, 405 éléments constituent le maillage du domaine, avec certains en zone proche, d'autres en zone lointaine. Il y a 9596 points d'interpolation soit une moyenne légèrement inférieur à 24 degrés de liberté par éléments. Le découpage et les ordres par direction sont montrés sur la figure 5.23.

Le maillage, ainsi que le choix des ordres, ne dépend pas de l'onde incidente et est donc fixé une seule fois. Les mêmes points d'interpolations sont utilisés même si l'angle de l'onde incidente est modifiée. Les figures 5.24 et 5.25 montrent les champs rayonnés obtenus pour deux ondes planes incidentes différentes (respectivement 30 et 120 degrés). Dans les deux cas, la figure de gauche représente le champ total (champ incident + champ diffracté) rayonné. Afin

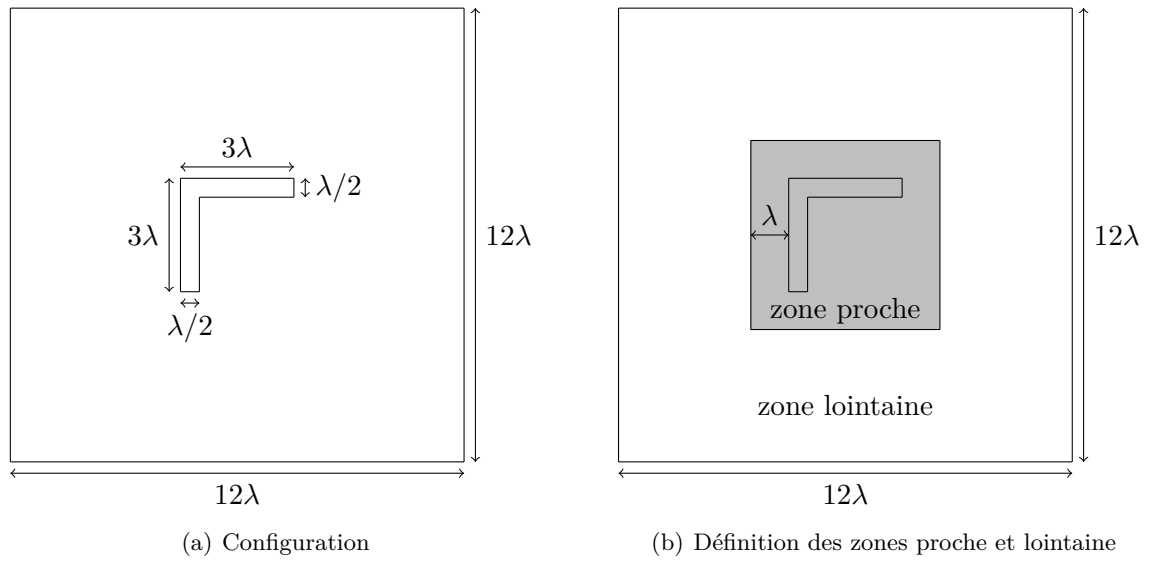


FIGURE 5.22 – Diffraction autour d'un objet *L-shape*

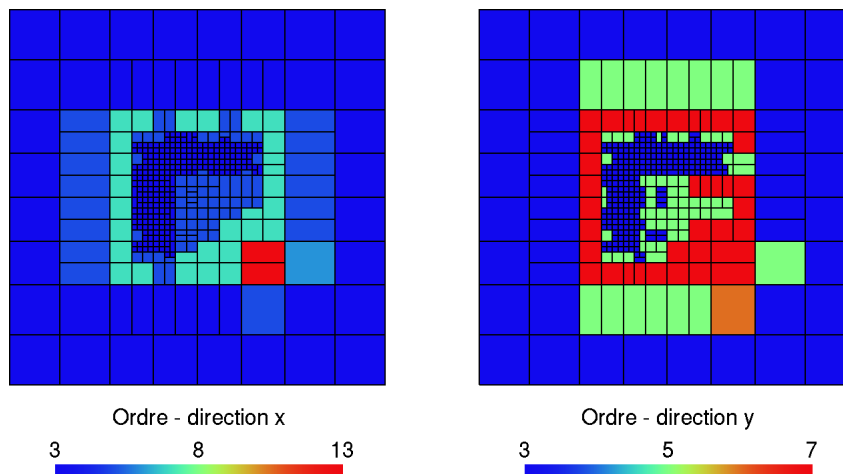


FIGURE 5.23 – Subdivision du domaine et ordres nécessaires par direction pour la configuration en forme de L.

de représenter le champ total, l'onde incidente, qui est de la forme $e^{i\nu x}$, où ν est la direction de l'onde plane, doit être ajoutée au champ diffracté. La figure de droite quant à elle affiche la subdivision du domaine (qui ne change pas) et le maillage de représentation utilisé pour une précision demandée à 1 %. Le maillage de représentation pour la figure 5.24 est constitué de 35009 éléments soit une moyenne de 86 triangles par élément. Pour la figure 5.25, il n'y a que 26978 triangles soit une moyenne de 67 triangles par élément.

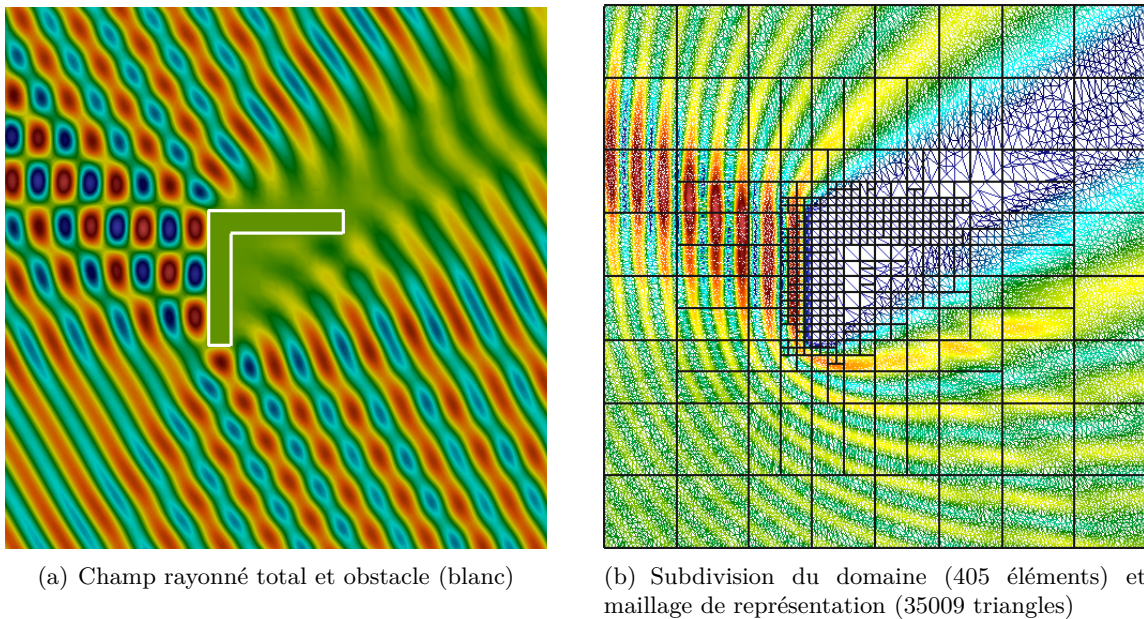


FIGURE 5.24 – Visualisation du champ rayonné total (parties réelle (a) et absolue (b)) diffracté par un objet en forme de L pour une onde plane incidente donnée (angle 30 degrés).

Augmentation du nombre d'onde

Nous reprenons les mêmes longueurs que le cas précédent mais en faisant varier le nombre d'onde, nous fixons celui-ci à $k = 100$. Les paramètres sont donc les suivants :

- Obstacle en forme de L de longueur maximale 1 soit environ 16λ .
- Nombre d'onde $k = 100$, longueur d'onde $\lambda \approx 0.06$.
- Taille du domaine à visualiser : $4 \times 4 \approx 64\lambda \times 64\lambda$.

Avec un critère toujours fixé à $R = 0.5$, le nombre de subdivisions du domaine est maintenant égal à 1644 (voir figure 5.26). Cette augmentation est logique car les longueurs n'ont pas changées mais la longueur d'onde est beaucoup plus petite donc il y a beaucoup plus de phénomènes à capturer (plus d'oscillations).

La figure 5.27 montre le champ rayonné total obtenu avec ces nouveaux paramètres. Pour une précision de visualisation demandée à 1 %, 545 098 triangles définissent le maillage de représentation soit une moyenne de 332 triangles par élément. La méthodologie mise en place permet de traiter n'importe quelle longueur d'onde, même lorsque que le nombre d'onde est élevé.

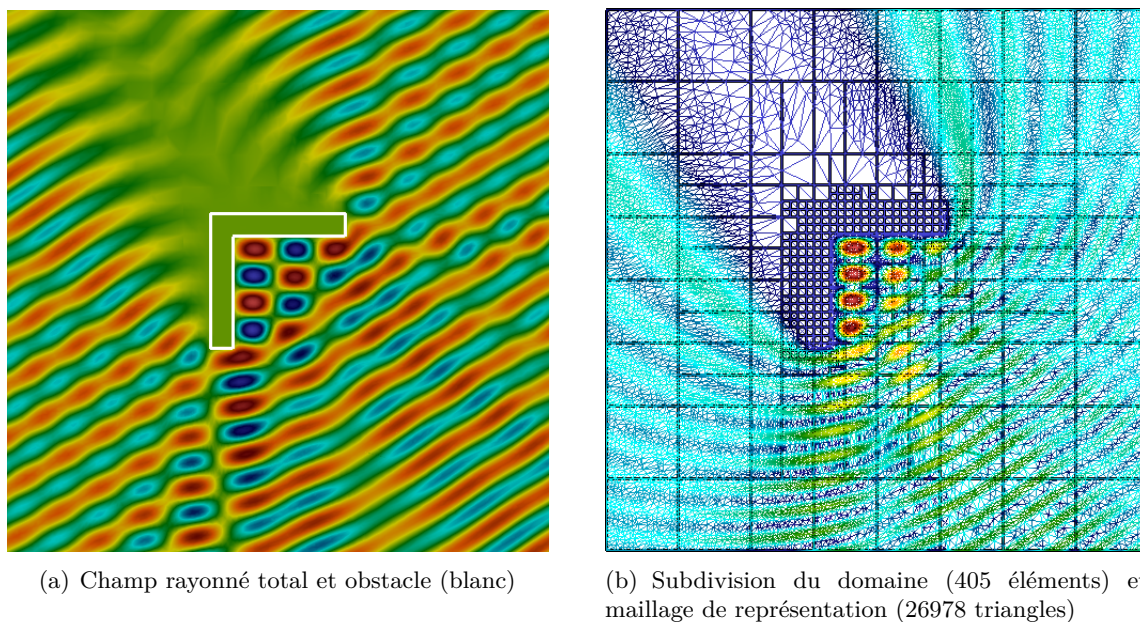


FIGURE 5.25 – Visualisation du champ rayonné total (parties réelle (a) et absolue (b)) diffracté par un objet en forme de L pour une seconde onde plane incidente (angle 120 degrés).

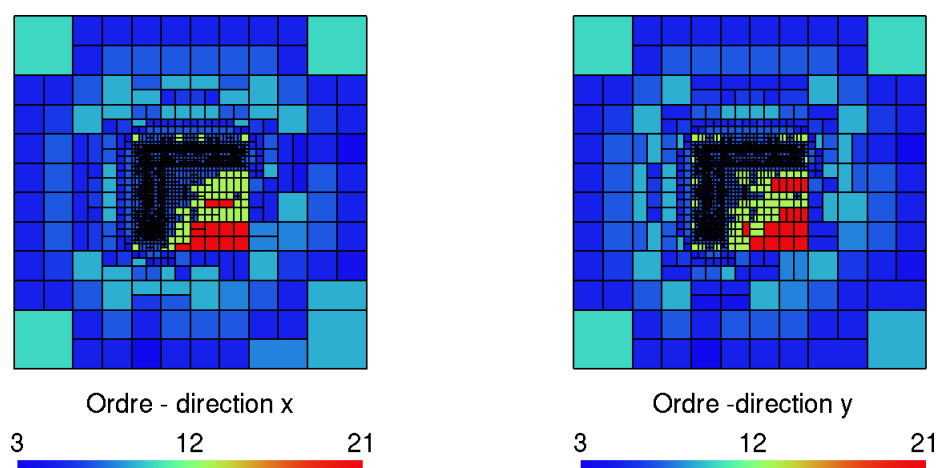


FIGURE 5.26 – Subdivision du domaine et ordres nécessaires par direction pour la configuration en forme de L pour $k = 100$.

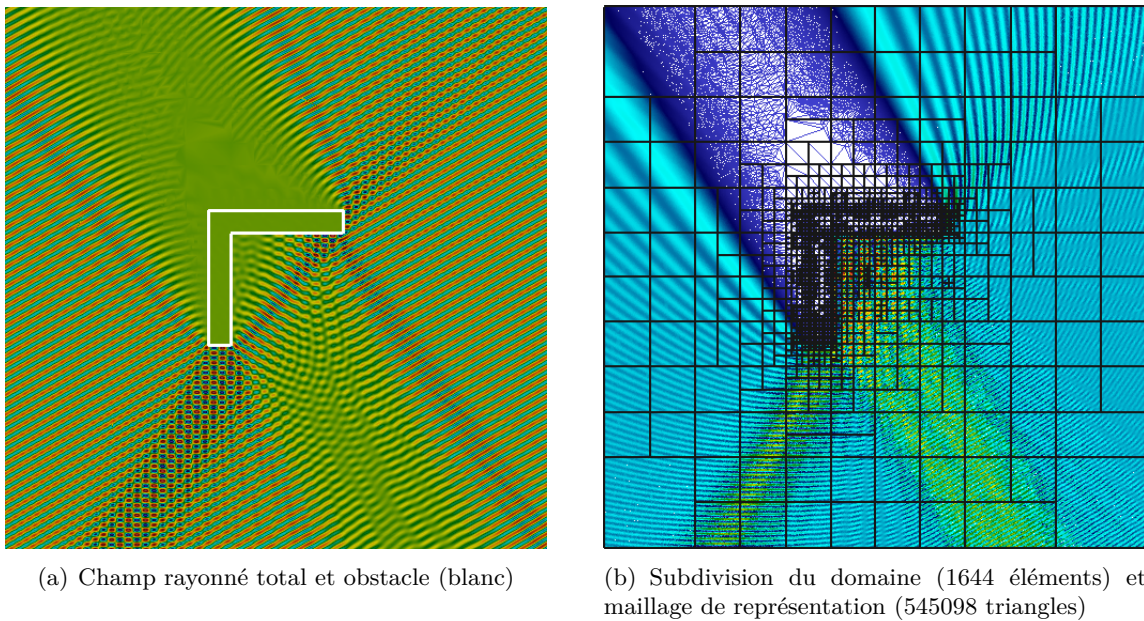


FIGURE 5.27 – Visualisation du champ rayonné total (parties réelle (a) et absolue (b)) diffracté par un objet en forme de L pour une onde plane incidente à $k = 100$.

5.6.3 BEM en 3D et rayonnements dans des plans

Dans cette section, nous présentons des résultats où la méthode des éléments finis de frontière est utilisée en dimension trois. L'objet est donc maillé par des éléments surfaciques comme sur la figure 5.28, mais les rayonnements sont effectués dans des plans. Notons que les axes ne sont pas classiques car par exemple l'altitude est désignée par Y (voir figure 5.28). Dans la section suivante, les rayonnements seront volumiques mais pour l'instant nous nous intéressons à des rayonnements dans des plans. Il s'agit d'un cas important puisque dans le cas volumique, l'utilisateur peut faire des coupes et donc se ramener à des plans. En travaillant directement dans un plan, la subdivision du domaine suit le processus en dimension deux. La génération de la visualisation est faite sur des éléments surfaciques et est donc plus aisée.

Nous nous intéressons à la propagation d'une onde autour d'un rafale (figure 5.28) dont les caractéristiques sont les suivantes :

- Longueur (direction Z) : 11.6.
- Largeur (direction X) : 7.
- Hauteur (direction Y) : 2.92.

La longueur d'onde est fixée à

$$k = 3. \quad (5.44)$$

Les caractéristiques des plans sur lesquels les champs sont reconstruits sont donnés dans le tableau 5.4. Le critère fixant le découpage du domaine est fixé à chaque fois à $R = 1$. Pour les plans respectivement 1, 2 et 3, le domaine à visualiser est découpé en respectivement 110, 148 et 24 éléments. Les découpages et maillages de représentation générés sont affichés sur la figure 5.29. Les champs rayonnés sont affichés sur la figure 5.30. Les nombres de triangles dans les

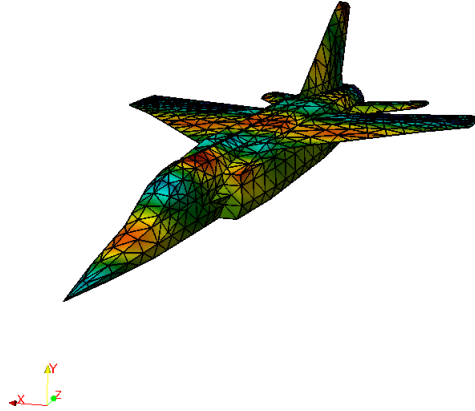


FIGURE 5.28 – Solution calculée par la méthode des éléments finis de frontière.

maillages de représentation sont 27761, 19915 et 27548 correspondants aux trois plans.

| Plan | X_{\min} | X_{\max} | Y_{\min} | Y_{\max} | Z_{\min} | Z_{\max} |
|--------|------------|------------|------------|------------|------------|------------|
| Plan 1 | 0 | 0 | -5 | 5 | -20 | 20 |
| Plan 2 | -10 | 10 | -2 | -2 | -20 | 20 |
| Plan 3 | -10 | -10 | -5 | 5 | 8 | 8 |

TABLE 5.4 – Caractéristiques des plans où les champs sont reconstruits.

5.6.4 BEM en 3D et rayonnements dans des volumes

Dans cette section, nous présentons des rayonnements volumiques. Nous considérons deux cas de propagation d'onde acoustique autour d'un rafale. Pour tous les exemples, l'objet diffractant est discrétisé à l'aide de triangles et nous souhaitons reconstruire la solution dans un volume. L'objet diffractant est exactement le même que dans la section 5.6.3. Tout d'abord, nous exhibons une configuration où le domaine \mathcal{V} est totalement disjoint de l'objet diffractant. L'interpolation directionnelle est donc utilisée afin de compresser l'information. Dans un second temps, le domaine à visualiser englobe le rafale.

5.6.4.1 Un rayonnement en dehors de l'obstacle diffractant

Pour ce premier cas, les caractéristiques du domaine \mathcal{V} sont les suivantes :

- $X_{\min} = 0$, $X_{\max} = 15$,
- $Y_{\min} = -10$, $Y_{\max} = 10$,
- $Z_{\min} = -20$, $Z_{\max} = -10$.

Ce domaine est « loin » de la surface où la méthode des éléments de finis de frontière est appliquée.

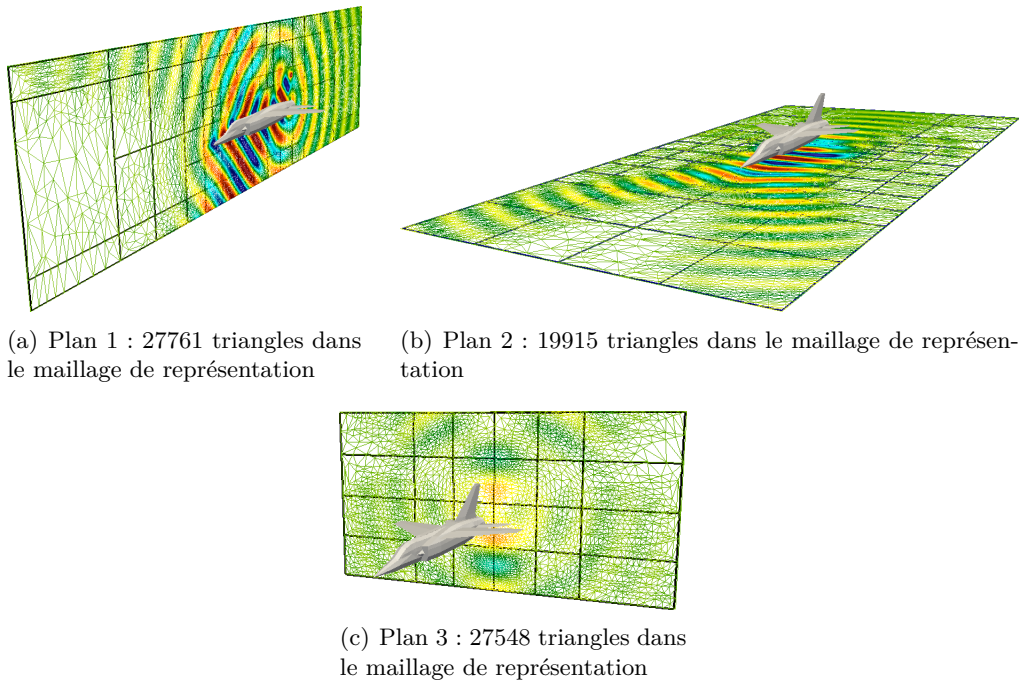


FIGURE 5.29 – Propagation d’une onde autour d’un rafale. Visualisation des champs rayonné dans des plans. Subdivision (en noir) des plans en question et maillages de représentation.

Le nombre d’onde est fixé à

$$k = 5.$$

L’algorithme de subdivision du domaine est appliqué avec comme paramètre $R = 0.2$. La figure 5.31 illustre la subdivision et exhibe les ordres assignés suivant chaque direction. Le domaine est décomposé en 308 éléments et il y a 14412 degrés de liberté où sont utilisées les formules de rayonnement. La méthode de visualisation permet de représenter la solution reconstruite où la précision demandée est fixée à 1 %. Le maillage de représentation est constitué de presque 8 millions de tétraèdres. La figure 5.32 montre le champ reconstruit, la subdivision du domaine ainsi que le maillage de représentation généré. Pour les mêmes degrés de liberté, la figure 5.33 compare les représentations obtenues avec et sans directions. Comme en dimension deux, l’approche directionnelle est très intéressante et permet d’obtenir une bien meilleure reconstruction.

5.6.4.2 Un rayonnement où le domaine à visualiser englobe l’obstacle

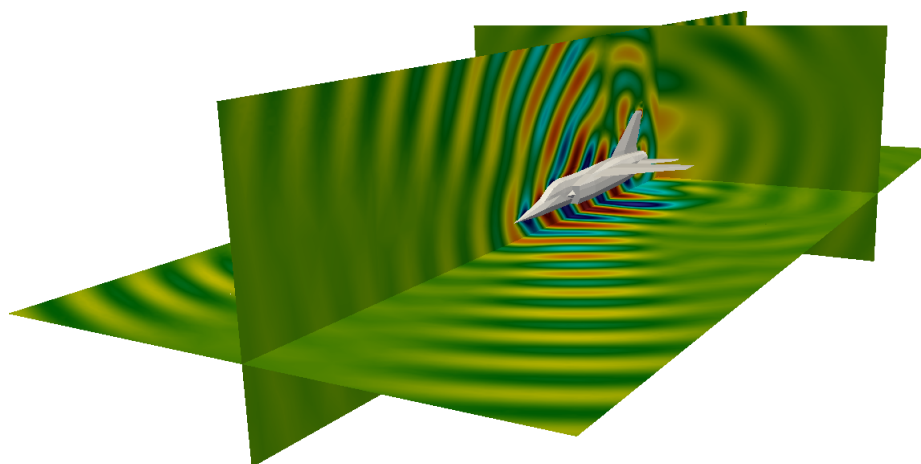
Pour ce second cas, la principale différence avec le précédent cas est que le domaine \mathcal{V} où nous souhaitons visualiser le champ reconstruit englobe l’obstacle. Ces caractéristiques sont :

- $X_{\min} = -20, X_{\max} = 20,$
- $Y_{\min} = -10, Y_{\max} = 10,$
- $Z_{\min} = -30, Z_{\max} = 30.$

Le nombre d’onde est fixé à

$$k = 2.$$

Notre algorithme de subdivisions est appliqué avec le paramètre $R = 1.0$ et 844 éléments sont générés comme le montre la figure 5.34. Comme précédemment, deux zones sont définies : une



(a) Champs reconstruits dans des plans

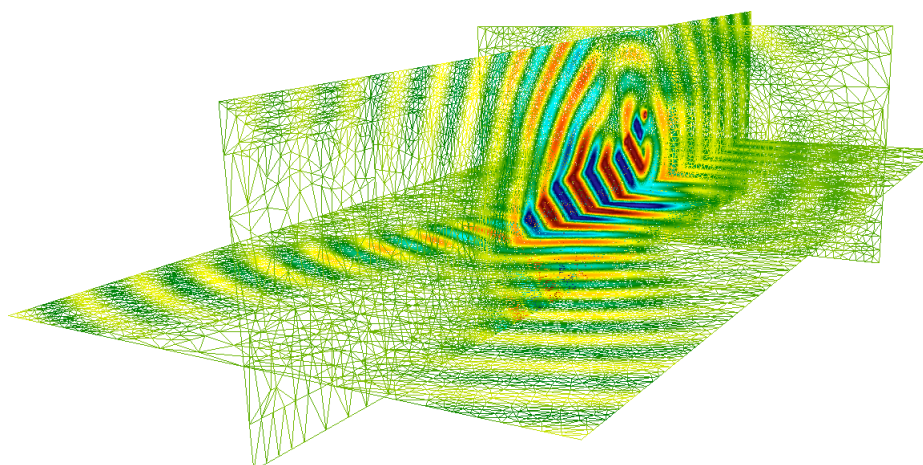
(b) Avec maillage de représentation (en *wireframe*)

FIGURE 5.30 – Visualisation des champs rayonné dans des plans. Propagation d’une onde autour d’un rafale.

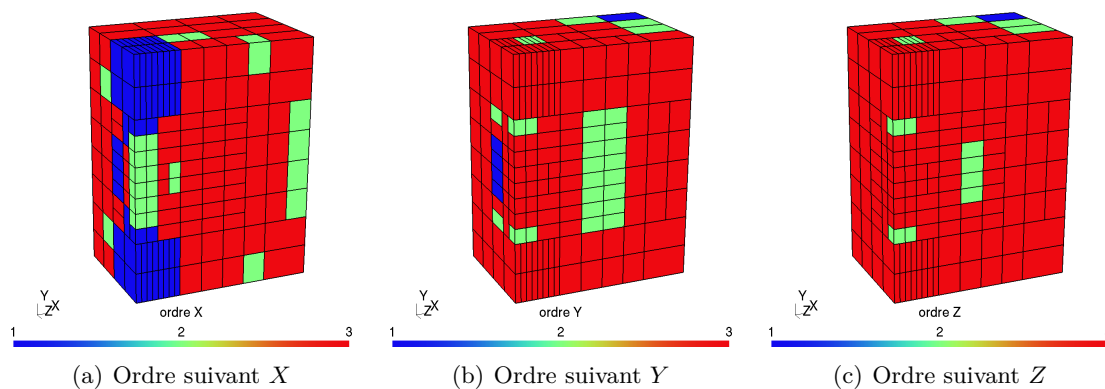


FIGURE 5.31 – Ordres suivant chaque direction pour le rayonnement 3D en dehors de l’obstacle diffractant

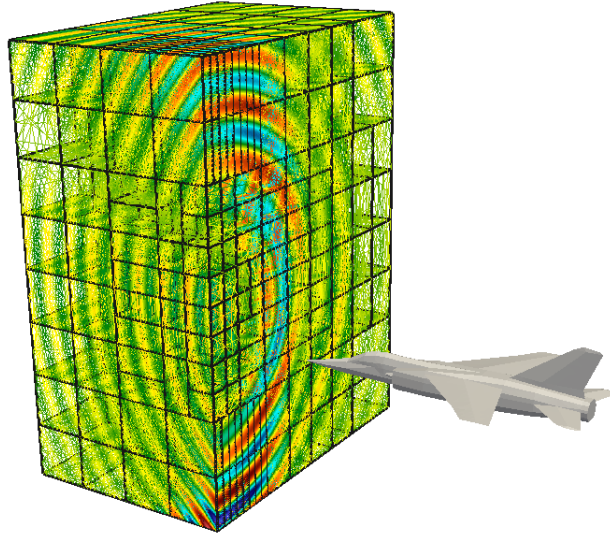


FIGURE 5.32 – Visualisation du champ rayonné reconstruit. La subdivision du domaine est affichée en noir (308 éléments), le maillage de représentation se trouve en *wireframe* (7 985 584 tétraèdres).

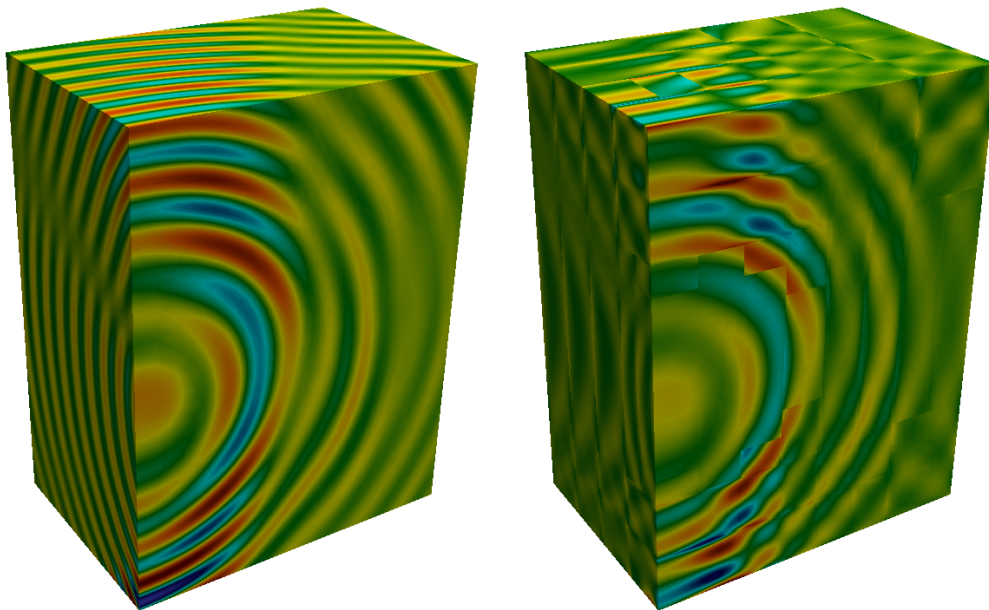


FIGURE 5.33 – Comparaison de la visualisation obtenue à 1 % avec la méthode adaptative (gauche) et sans direction (droite).

zone *proche*, en présence de l'obstacle, où l'approche directionnelle n'est pas utilisée et une zone *lointaine*, définie à partir d'une longueur d'onde de l'obstacle. Dans la zone proche, les éléments, au nombre de 648, sont de très petite taille tandis que les 196 éléments de la zone lointaine sont beaucoup plus grands grâce aux ondes planes introduites dans les fonctions de base.

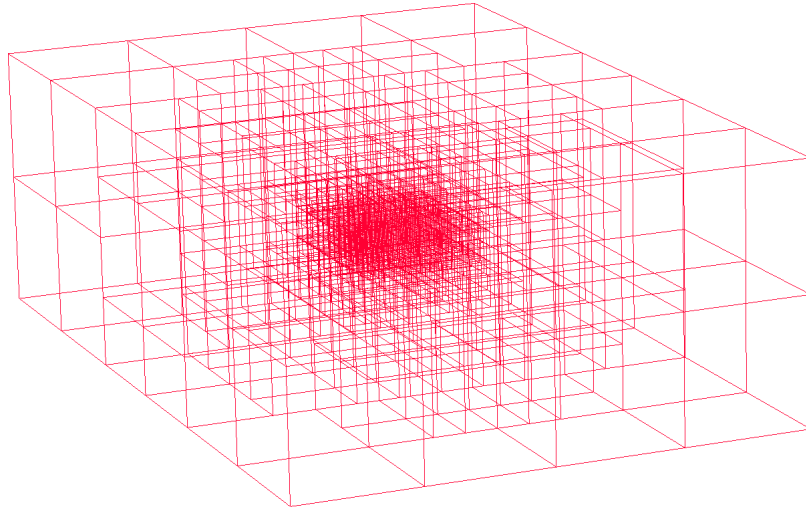


FIGURE 5.34 – Subdivision du domaine en 844 éléments.

Pour chaque élément de la subdivision, les ordres sont fixés en se basant sur la section 5.5.3. La figure 5.35 met en évidence ces ordres. Le nombre total de degrés de liberté est 275 576. Une coupe est réalisée afin de voir les ordres près de l'obstacle et est visible sur la figure 5.36.

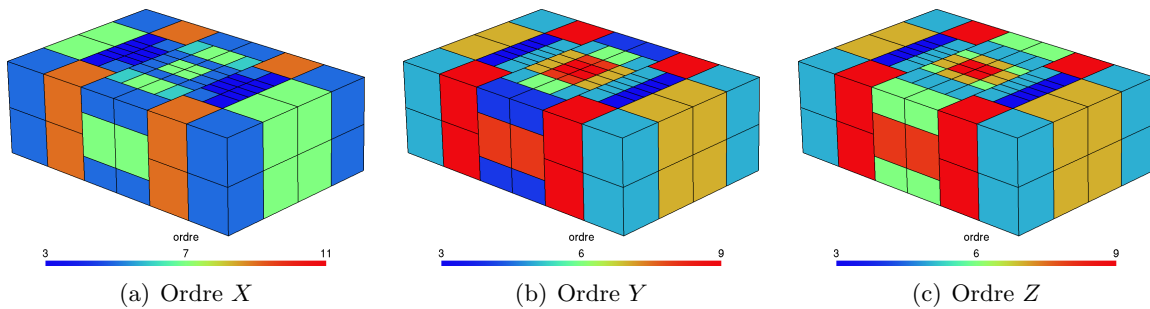


FIGURE 5.35 – Ordres suivant chaque direction pour le rayonnement 3D autour de l'obstacle.

Pour les mêmes degrés de liberté, la figure 5.37 compare les visualisations obtenue à 1 % avec notre méthode avec et sans les directions. Lorsque l'approche directionnelle est utilisée, la précision sur la reconstruction du champ et sur sa représentation est garantie. Dans le second cas, la différence se fait dans la zone *lointaine* où l'absence de direction est visible. La reconstruction est tellement mauvaise que l'on peut avoir l'impression qu'une seconde onde vient de l'autre sens et qu'il y a des recombinaisons d'ondes. En se basant sur cette représentation, l'analyse peut induire des erreurs de jugement.

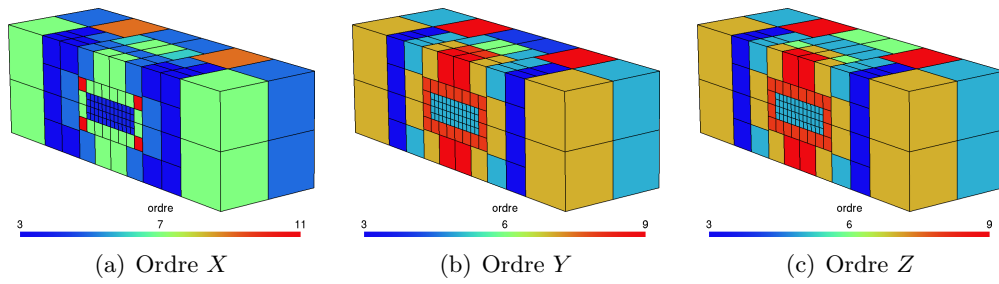


FIGURE 5.36 – Coupe de la figure 5.35 permettant de voir les ordres près de l'obstacle.

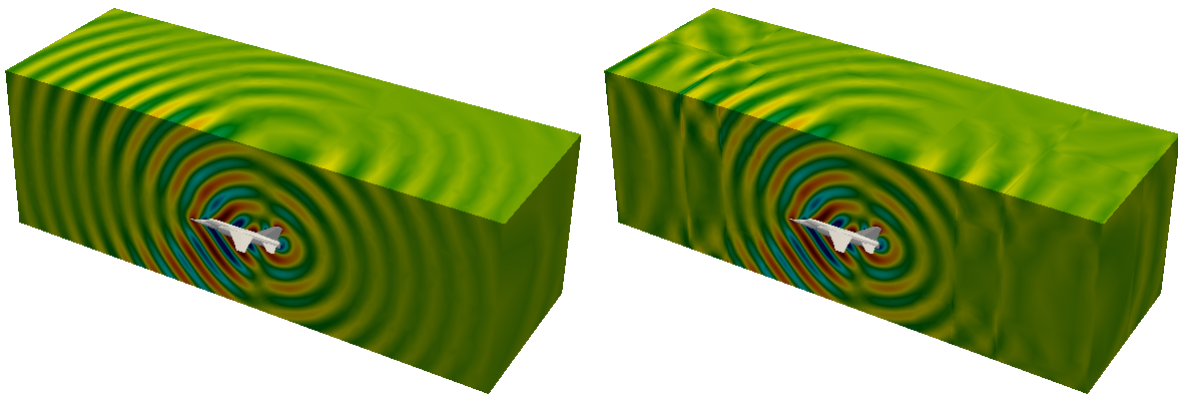


FIGURE 5.37 – Comparaison pour les mêmes degrés de liberté de la visualisation obtenue avec directions (gauche) et sans directions (droite).

La figure 5.38 se concentre sur la représentation obtenue avec notre méthode de visualisation pour une précision demandée à 1%. Le maillage de représentation est constitué de 1 623 153 tétraèdres et est affiché sur la figure 5.38.b. Naturellement, ce dernier est raffiné lorsque les variations de la solution sont les plus importantes et en particulier au niveau de l'obstacle. Dans notre cas, nous visualisons le champ diffracté, qui à l'intérieur de l'obstacle est égal à l'opposée de l'onde plane incidente puisque le champ total y est nul.

5.7 Conclusions du chapitre

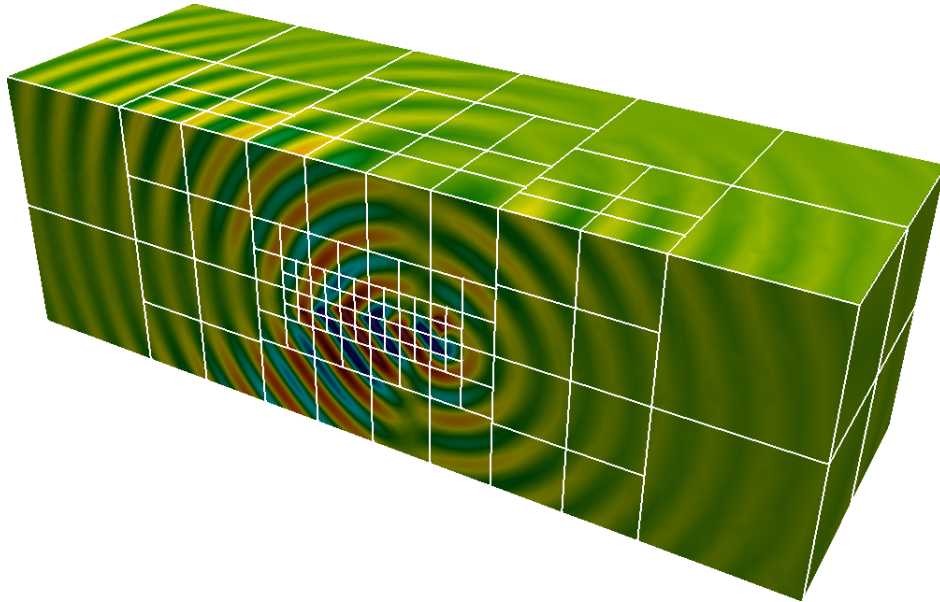
Dans ce chapitre, nous nous sommes intéressés à la reconstruction de champs rayonnés. Nous avons vu que, bien que la méthode des éléments finis de frontière soit pertinente pour des problèmes de propagation d'ondes, comme les équations de Helmholtz ou de Maxwell, la construction de cartographies, à l'aide des formules de représentation, n'est pas souvent effectuée en pratique car celle-ci est très coûteuse. Nous avons montré que l'utilisation de fonctions d'ordre élevé, au lieu de simples interpolations affines, permet de gagner énormément en performance. De plus, des interpolations directionnelles sont considérées et en particulier des domaines très larges peuvent être traités grâce à cela.

Nous avons présenté une méthodologie et un algorithme pour reconstruire des champs rayonnés, composés de polynômes d'ordre élevé et d'ondes planes, qui reposent sur deux étapes :

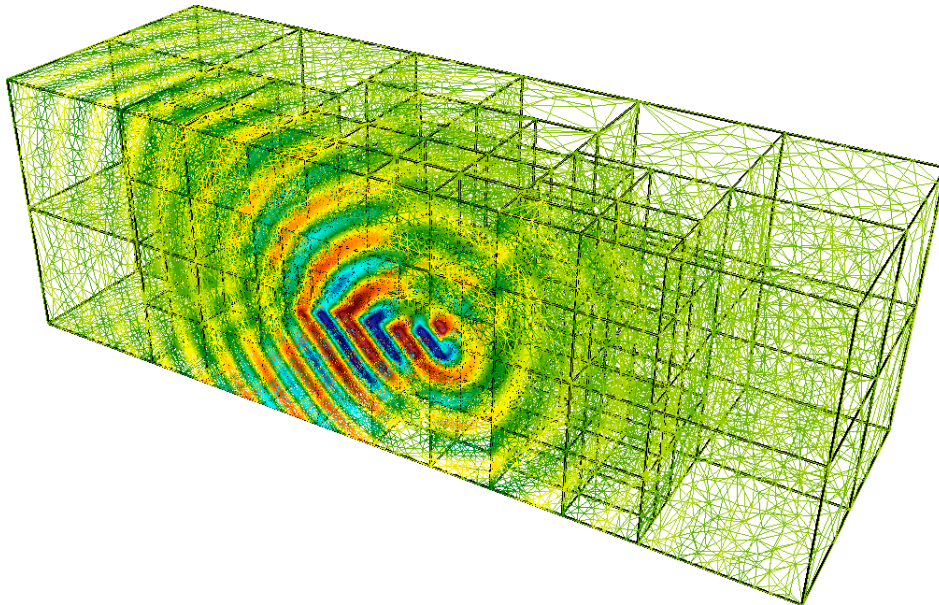
- **Subdivision du domaine à l'aide d'un critère d'admissibilité.** Un paramètre (ratio longueur/longueur d'onde modifié) est fixé pour limiter les oscillations de la fonction à interpoler dans chaque élément et pour chaque direction. Un critère d'admissibilité est introduit pour définir si un élément doit être raffiné et un processus de subdivision guidé par ce même critère est généré. Finalement, un maillage du domaine à visualiser est obtenu où tous les éléments sont admissibles. Des directions sont associées à chaque élément afin d'utiliser les propriétés de l'interpolation directionnelle en incluant des ondes planes dans les fonctions de base.
- **Assignment des ordres des fonctions de base.** Des règles *a priori* sont utilisées pour fixer les ordres dans les éléments qui ont été générés. Ces règles sont basées sur un problème d'interpolation en dimension un. Les ordres sont choisis afin que l'interpolation du résidu (la fonction à approcher est moins oscillante grâce à l'introduction des ondes planes) soit précise. Les fonctions de base s'écrivent finalement à l'aide de polynômes d'ordre élevé et d'ondes planes suivant les directions définies précédemment.

Dans une seconde partie de ce chapitre, plusieurs exemples numériques issus d'équations intégrales ont validé la méthodologie. Tout d'abord, nous avons montré l'intérêt de l'interpolation directionnelle sur un exemple très simple. Les gains obtenus avec l'ordre élevé par rapport au bas ordre étaient déjà grands, mais en introduisant des ondes planes, le nombre de degrés de liberté est encore largement réduit. La méthodologie mise en place (i.e. subdivision du domaine puis assignation des ordres) est ensuite testée sur différents exemples de dimensions deux ou trois où nous montrons, là encore, qu'un champ reconstruit compressé (avec moins de degrés de liberté) et précis est obtenu. La méthode de visualisation est utilisée pour représenter les champs obtenus.

Nous avons réfléchi à une modification de la méthodologie afin de minimiser le nombre de degrés de liberté. Au lieu de baser la méthode sur deux étapes disjointes (i.e. subdivision du domaine puis assignation des ordres), l'idée est de faire les deux en mêmes temps. En effet, dans la version proposée actuellement, les ordres fixés avec les règles *a priori* ne sont choisis



(a) Subdivision du domaine



(b) Maillage de représentation en *wireframe* et subdivision du domaine en noir

FIGURE 5.38 – Représentation des champs rayonnés autour d'un rafale avec nos méthodes de reconstruction et de visualisation.

que lorsque tous les éléments sont admissibles selon les critères donnés dans la section 5.5.1 mais ces règles pourraient s'appliquer à n'importe quel étape de l'algorithme. Ainsi, pour une configuration donnée, le nombre de degrés de liberté nécessaire pour atteindre une précision donnée est calculé. Ensuite, différents découpages sont testés et si une configuration fait diminuer le nombre de degrés de liberté, alors le découpage en question est effectué. Dans le cas où plusieurs configurations feraient diminuer le nombre total de d.d.l., celle qui fait diminuer le plus ce nombre est appliquée. Les configurations testées sont les mêmes que les découpages de la section 5.5.2. La méthodologie est résumée dans l'algorithme 11.

Algorithme 11 : Un choix de découpage pour minimiser le nombre de degrés de liberté

Initialiser $\mathcal{E}_{\mathcal{V}} = \mathcal{V}$ et $\mathcal{B}_{\mathcal{E}} = \mathcal{G}$ et N_{ddl} le nombre de d.d.l.

Tester les découpages et calculer N_{\min} comme le minimum de d.d.l. de ces nouvelles configurations.

tant que $N_{\min} < N_{ddl}$ **faire**

 Subdiviser $(\mathcal{E}_{\mathcal{V}}^{\min}, \mathcal{B}_{\mathcal{E}}^{\min})$ le couple réalisant la configuration N_{\min} .

 Mettre à jour N_{ddl} le nouveau nombre de d.d.l.

 Tester les découpages et calculer N_{\min} comme le minimum de d.d.l. de ces nouvelles configurations.

fin

L'avantage de l'algorithme 11 est que le nombre de points à faire rayonner peut être réduit, ce qui implique moins de coûts grâce à la réduction du nombre d'appels aux formules de représentation. L'interpolation directionnelle est toujours utilisée et les règles *a priori* prennent toujours en compte les directions qui apparaissent dans les ratios. Une précision similaire que l'algorithme 9 est obtenu avec cette stratégie puisque les mêmes règles *a priori* sont utilisées.

L'algorithme 11 comporte néanmoins quelques inconvénients. Tout d'abord, dans ce cas, les ordres peuvent être potentiellement très grands, avec peu de subdivisions néanmoins. Ceci est possible notamment si on ne considère pas de direction dans le cas où le domaine à visualiser est très proche de l'objet notamment. En pratique, la gestion d'ordre très élevé, par exemple supérieur à 100 dans chaque direction, est délicate. Il est en général plus intéressant d'avoir plus d'éléments avec des ordres moins grands. En outre, l'algorithme 11 est intéressant puisque le paramètre R n'est pas utilisé mais, à une itération donnée, il n'y a finalement que peu de configurations testées et il peut arriver qu'aucune de ces configurations ne fasse diminuer le nombre de degrés de liberté. Ainsi, cela peut produire l'arrêt de l'algorithme alors qu'il peut exister d'autres configurations qui feraient diminuer le nombre de degrés de liberté. En particulier, même si une configuration ne fait pas diminuer ce nombre, la suivante pourrait le faire. Il n'est pas possible de savoir *a priori* le découpage à effectuer.

Finalement, l'algorithme 11 a été testé sur certaines configurations mais manque de robustesse. Il pourrait être intéressant, à l'avenir, de l'améliorer.

Chapitre 6

Conclusions et perspectives

L'objectif premier de cette thèse était de proposer une méthode et des algorithmes efficaces afin de visualiser des simulations d'ordre élevé. Une grande partie du manuscrit (chapitres 2, 3 et 4) est ainsi dédiée à cette tâche. Dans une seconde partie (chapitre 5), nous nous sommes intéressés à la reconstruction et la visualisation de champs rayonnés pour des ondes en régime harmonique. Nous présentons ici des conclusions ainsi que quelques perspectives de ce travail.

Conclusions

Nous avons débuté ce mémoire en recensant les stratégies actuelles permettant de visualiser et de post-traiter des simulations d'ordre élevé. Une approche naïve consiste à redécouper de manière régulière chacun des éléments du maillage de calcul. Une grille de représentation est ainsi générée et la solution numérique est calculée en chacun des nœuds. Un logiciel de visualisation standard est ensuite utilisé pour représenter la solution sur tout le domaine en utilisant des interpolations affines à partir de cette grille. Un manque de maîtrise apparaît à la fois sur la précision de la représentation obtenue ainsi que sur la quantité de données générée.

Une première stratégie consiste à développer un logiciel dédié aux méthodes d'ordre élevé. Les principaux avantages sont qu'un tel logiciel est directement compatible avec des données provenant des méthodes d'ordre élevé et qu'il n'y a pas d'erreur de visualisation puisque dans ce cas, on ne crée pas de représentation approchant la solution numérique. Néanmoins, cette stratégie nécessite de lourds développements afin d'avoir une plateforme capable de générer des rendus graphiques mais également disposant de tous les outils nécessaires au post-traitement des résultats. Le peu d'investissement sur cette stratégie s'explique principalement par deux raisons. D'une part, pendant longtemps, il n'y avait pas de besoin en visualisation de simulations d'ordre élevé car ces méthodes sont relativement récentes. D'autre part, les outils graphiques, comme OpenGL, n'étaient pas aussi puissants qu'actuellement. A l'avenir, il semble que cette stratégie soit exploitable notamment en terme d'interactivité, ce qui n'était pas forcément le cas avant.

Une seconde stratégie consiste à générer des maillages de représentation adaptée à la solution numérique à visualiser. L'idée est ainsi de construire une représentation affine optimisée approchant la solution numérique afin de pouvoir la représenter à l'aide d'un logiciel de visualisation standard. Dans ce cas, une erreur de visualisation apparaît et doit être contrôlée. Les principales différences entre les travaux concernent la façon de définir et d'estimer cette erreur de visualisation ainsi que la technique de génération ou de raffinement des maillages de représentation utilisée. Dans ces travaux, un aspect limitant est que le contrôle de l'erreur de visualisation n'est que partiel alors que nous avons vu dans la section 4.1.1 qu'un contrôle ponctuel et global est nécessaire afin de ne pas perdre d'information.

Dans une première partie, nous avons conçu une méthode de visualisation, reposant sur la stratégie de remaillage en solution de bas ordre, permettant de générer une « bonne » représentation de simulation d'ordre élevé.

Dans cette optique, dans le chapitre 2, nous avons commencé par définir ce qu'est une « bonne » représentation et trois objectifs sont ressortis :

- la compatibilité avec les logiciels de visualisation standard,
- le contrôle ponctuel de l'erreur de visualisation,
- la restitution des continuités et des discontinuités.

Ensuite, nous avons mis en place une méthodologie et des algorithmes afin de répondre à ces objectifs. Pour chaque élément du maillage de calcul, un maillage de représentation est construit en utilisant un algorithme de Bowyer-Watson où le raffinement est guidé par un indicateur d'erreurs *a posteriori*. Ce dernier majore la distance de Hausdorff qui est le critère de convergence de la représentation vers la solution numérique en norme L^∞ . Enfin, pour respecter les (dis)continuités et en particulier pour éviter d'introduire des sauts artificiels entre les éléments, les éléments sont décomposés en dimensions inférieures et une approche par dimensions croissantes est utilisée. Le but est de définir un maillage unique sur chaque interface entre les éléments du maillage de calcul. Pour cela, des indicateurs d'erreurs *a posteriori* sont définis pour chaque dimension et prennent en compte toutes les contributions en attachant sur les interfaces considérées, les traces des fonctions sous-jacentes. Afin d'évaluer tous ces estimateurs, un algorithme déterministe de type séparation et évaluation est utilisé.

Dans le chapitre suivant, nous avons effectué des développements spécifiques afin que notre méthode de visualisation soit capable de traiter des surfaces courbes. En effet, si la méthode de visualisation est appliquée telle quelle, nous avons vu, notamment dans la section 4.2 consacrée à la validation de la méthode pour des éléments courbes, que des recouvrements ou des retournements peuvent apparaître. Ces artefacts sont dus à une perte de bijectivité de l'approximation de la transformation géométrique utilisée pour générer le maillage de représentation des éléments à partir du maillage de représentation de la cellule de référence associée. Ces problèmes viennent principalement de la contrainte de mailler par dimensions croissantes qui imposent de ne plus modifier les maillages des bords une fois traités. La conséquence est qu'une zone dite de retournements est introduite. Pour pallier ces problèmes, nous avons défini des indicateurs d'erreurs *a posteriori* spécifiques, qui lors du maillage des arêtes des éléments, assurent qu'aucun point ne sera ajouté dans ces zones lors du passage en dimension supérieure. Pour cela, nous avons défini un algorithme qui permet de caractériser ces zones de manière efficace : la complexité est réduite de $\mathcal{O}(N)$ à $\mathcal{O}(\sqrt{N})$.

Dans le chapitre 4, nous avons validé numériquement notre méthode de visualisation, sur des éléments droits et courbes, pour des problèmes de complexité croissante. éléments surfaciques et volumiques sont traités. Nous avons également amélioré la robustesse de la méthode, pour certains problèmes très spécifiques, en nous inspirant des développements effectués dans le cadre des éléments courbes. Le problème, du à la contrainte de maillage par dimensions croissantes, est que les maillages de représentation peuvent être relativement grossiers sur les bords (en fonction des variations de la trace des fonctions et de la tolérance demandée comme précision), ce qui peut générer des éléments quasiment plats. Afin d'éviter cela, l'idée a été de définir des voisinages autour de ces arêtes et de considérer ces zones lors du maillage des arêtes afin d'éventuellement raffiner les maillages sur le bord des éléments.

Dans une seconde partie de ce travail, nous nous sommes intéressés à la reconstruction et à

la visualisation de champs rayonnés pour des ondes en régime harmonique. Notre objectif était de proposer une méthode pour avoir une reconstruction précise, automatique et compressée des champs rayonnés. Ceci signifie que nous voulions visualiser ces champs rayonnés dans un certain volume à partir de données connues sur une surface. Ces données peuvent par exemple être calculées par FEM ou par BEM. Nous nous sommes plus particulièrement intéressé au cadre équations intégrales même si la méthodologie est générale. En remarquant que le noyau de la solution fondamentale est, sous certaines conditions, proche d'une onde plane, l'idée a été d'introduire une interpolation directionnelle du noyau afin de compresser l'information. Pour cela, les fonctions de base définies combinent des polynômes d'ordre élevé et des ondes planes. Nous avons mis en place un algorithme qui permet de créer une subdivision du domaine à visualiser et de définir ces fonctions de base sur chaque élément. Cet algorithme est très efficace grâce à la simplicité et la pertinence des critères utilisés. En effet, un critère d'admissibilité basé sur le ratio entre les tailles des éléments et des longueurs d'onde modifiées guide ces subdivisions. Des règles *a priori* permettent ensuite de fixer les degrés des polynômes d'ordre élevé sur chaque élément afin d'approcher précisément les champs rayonnés locaux. Les gains obtenus par la compression effectués peuvent être très importants avec des facteurs 30 sur le nombre de degrés de liberté entre les cas sans et avec directions. Ces facteurs sont encore plus grands si on compare notre méthodologie aux grilles cartésiennes. Finalement, la méthode de visualisation a été utilisée pour représenter ces champs et valider numériquement le potentiel de l'approche.

Discussions et perspectives

Tout d'abord, un point important concerne les temps CPU, qui actuellement sont longs notamment à cause du contrôle ponctuel demandé (résolution d'un problème d'optimisation). Nous avons réfléchi à des stratégies afin de limiter les appels à l'algorithme DIRECT. En effet, malgré le fait que la recherche effectuée par cet algorithme soit guidée et donc bien meilleure qu'une répartition régulière des points testés, ce processus est coûteux. Une idée a été d'utiliser une estimation d'erreur quadratique. Pour cela, la solution n'est calculée qu'aux degrés de liberté quadratiques à savoir les sommets et les centres des arêtes. L'erreur est dans ce cas définie comme la différence entre l'interpolation quadratique (i.e. de degré deux et définie à partir des degrés de liberté quadratiques) et l'interpolation affine (i.e. de degré un et définie uniquement à partir des sommets). L'évaluation de cette erreur est rapide et il est même possible, lorsque la fonction est réelle, de déterminer analytiquement un point réalisant l'erreur maximale (en calculant les gradients des fonctions et en résolvant un petit système linéaire). Une fois que cette estimation a convergé sous la tolérance, on peut utiliser l'algorithme DIRECT afin d'estimer la vraie erreur ponctuelle. L'intérêt de cette approche est que les recherches effectuées avec l'erreur quadratique sont très rapides à réaliser et permettent d'obtenir une première représentation facilement. Sur les quelques tests réalisés, cette stratégie semble prometteuse avec des temps de calcul pouvant être réduit de moitié par exemple.

De plus, l'approche basée sur une construction auto-adaptative proposée souffre d'un manque d'optimalité au niveau de la construction du maillage de représentation. En effet, le critère pour choisir le point à ajouter et l'utilisation de techniques de retournement d'arêtes standards rendent cet algorithme non optimal au sens de la meilleure décroissance de l'erreur. Il serait intéressant d'investiguer les techniques de raffinement de maillages anisotropes basées sur l'utilisation de métriques (adaptées à l'estimateur d'erreur) [D'Azevedo, 1991, Alauzet and Frey, 2003].

Le traitement des éléments courbes est actuellement limité aux surfaces. Il est tout à fait possible d'utiliser la méthode de visualisation sur des volumes courbes mais dans ce cas, il n'y

a pas de garantie que l'approximation de la transformation géométrique soit bijective. Ainsi, on ne peut pas assurer que la représentation générée soit une fonction. Il serait intéressant d'étendre les développements spécifiques aux volumes courbes. Néanmoins, la première question concerne la définition et la caractérisation des zones posant éventuellement des problèmes. Il faudrait ensuite définir un indicateur d'erreurs spécifique.

Nous avons proposé une stratégie de raffinement du maillage de bord dans la section 4.3 afin d'améliorer la qualité des maillages de représentation et également d'avoir un algorithme plus robuste. Une zone, correspondant à un voisinage, est ainsi définie. Le critère de construction de cette zone que nous utilisons est géométrique, simple et assure la convergence mais on pourrait en tester d'autres. La caractérisation de cette zone pourrait par exemple prendre en compte les variations des traces des fonctions dans leur construction.

Concernant la partie reconstruction de champs rayonnés, plusieurs perspectives seraient intéressantes. Tout d'abord, comme la méthodologie mise en place est générale, on pourrait s'intéresser à d'autres applications. Ces applications peuvent concerner d'autres domaines d'applications en BEM comme l'électromagnétisme qui n'a pas été testé numériquement ou de manière plus générale à la reconstruction de champs où les données seraient connues sur une surface en étant calculées à partir de FEM. Deuxièmement, on pourrait s'intéresser à des éléments plus compliqués comme par exemple des éléments courbes permettant de définir des domaines à visualiser plus complexes. Enfin, nous l'avons déjà évoqué mais il pourrait être pertinent de proposer une nouvelle façon de faire afin d'optimiser le nombre de degrés de liberté. Pour cela, une idée serait construire simultanément les subdivisions des domaines et les fonctions de base intégrant les polynômes d'ordres élevés et les ondes planes.

Un dernier aspect concerne la régularisation de champs dans le cadre de la méthode DG. En effet, la méthode de Galerkin Discontinu est basée sur les sauts entre les éléments et peut rendre discontinue une solution qui était continue. Or, d'un point de vue physique, ces sauts n'ont pas de sens. Ainsi, il pourrait être intéressant d'éliminer les discontinuités numériques (dues aux flux de la méthode) entre les éléments tout en conservant les discontinuités physiques (dues à la physique du problème). L'idée serait de construire une solution équivalente à celle à visualiser mais en lui donnant plus de sens physique. Pour cela, on pourrait ajouter comme données d'entrées du post-traitement, différentes informations provenant du problème (comme la nature de l'opérateur utilisé) ou encore du schéma numérique utilisé. En se donnant un indicateur de sous-partie du domaine de calcul (décrivant par exemple un changement de matériau), on peut guider la régularisation et éviter l'introduction de phénomènes non-physiques. Le but est de visualiser des discontinuités uniquement si elles existent. Cockburn et al. [Cockburn et al., 2000b] ont proposé une méthode appelée *SIAC* (*Smoothness-Increasing Accuracy-Conserving*) permettant de gagner en régularité et même d'améliorer l'ordre de la méthode dans certains cas. Ce post-traitement ne peut pas être appliqué à notre problème car il rendrait toutes les composantes continues. Pour régulariser la solution, une idée serait d'utiliser des fonctions de base d'éléments finis de type Raviart-Thomas ou Nédélec, selon les problèmes, qui ont naturellement de bonnes propriétés sur les composantes normales ou tangentielles. Celles-ci sont importantes car, pour certains problèmes, les discontinuités physiques peuvent être normales ou tangentielles.

Annexes

Annexe A

La génération de maillages en simplexes pour créer un maillage de représentation

Afin de générer une approximation de la solution numérique, un maillage de représentation composé de simplexes doit être construit. Aussi, nous faisons dans ce chapitre quelques rappels succincts sur les maillages et présentons des méthodes de génération de maillages en simplexes. Nous choisissons celle qui semble la plus pertinente vis à vis de notre problématique et motivons ce choix.

Toutes les méthodes numériques traitées par notre méthode de visualisation sont basées sur la discrétisation du domaine de calcul et la génération d'un maillage. Cette étape de génération du maillage est une « vraie » étape dans le processus de calcul dans le sens où, en général, la construction d'un maillage nécessite l'utilisation d'un *mailleur* (i.e. un logiciel dédié). Un maillage est une partition du domaine en un ensemble de cellules élémentaires. La qualité du maillage est primordiale car la précision de la solution numérique dépend du maillage utilisée. En fonction de la méthode numérique choisie, le maillage peut être soumis à certaines contraintes. En différences finies, des maillages structurés (voir définition A.1 ci-dessous), construits sous forme de grilles et donc cartésien, sont souvent utilisés. Ces maillages sont très faciles à générer mais ne peuvent traiter que des géométries simples. Une autre limitation de ce type de maillage vient du fait que le raffinement local du maillage n'est pas possible, ce qui peut impliquer un nombre important d'éléments et de degrés de liberté. Des maillages non-structurés sont donc généralement utilisés, en particulier pour les méthodes des éléments finis et de Galerkin Discontinue.

Définition A.1. *Un maillage \mathcal{T}_h d'un domaine Ω est dit **structuré** si ces nœuds sont définis à partir d'une grille (en dimension 2 ou 3). Cela signifie que les éléments sont repérés par des indices et que la connaissance de ces voisins se fait simplement en incrémentant ou décrémentant ces indices. Il n'y a donc pas de table de connectivité et quasiment pas de stockage nécessaire. Cependant, les domaines considérés doivent être très simples et on ne peut pas raffiner le maillage sans modifier sa taille dans tout le domaine.*

*Un maillage **non-structuré** est défini à partir d'un tableau des coordonnées de tous les nœuds du maillage et d'une table de connectivité permettant de définir chaque élément par les numéros de ces sommets. Il est en général nécessaire de stocker les voisins de chaque élément. Ce type de maillage permet de mailler des géométries complexes et d'avoir des zones de raffinement. Cependant, ils sont plus longs à générer et les coûts de stockage sont plus élevés.*

La figure A.1 montre un maillage structuré et un autre non structuré.

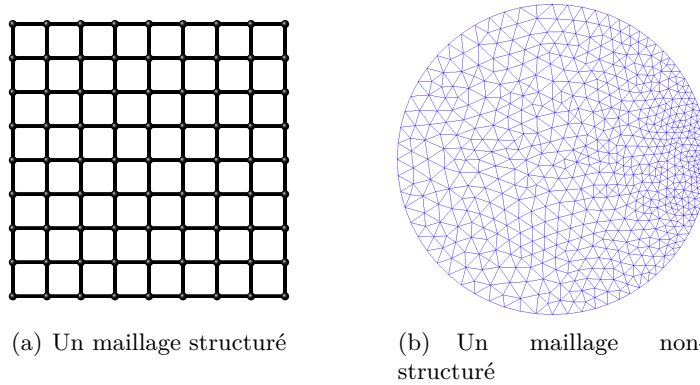


FIGURE A.1 – Exemples de maillages structuré (gauche) et non-structuré (droite).

Les maillages des méthodes des éléments finis doivent être conformes selon la définition A.2 tandis que les maillages des méthodes de Galerkin Discontinue peuvent être non-conformes. Ceci offre une flexibilité supplémentaire pour la construction de ces derniers. La figure A.2 illustre la différence entre des maillages conforme et non-conforme. Dans le second cas, on observe la présence d'un *hanging node* entouré en rouge sur la figure. Ce point appartient à trois triangles. C'est un des sommets pour deux d'entre eux, mais ce n'est pas un sommet du dernier triangle.

Définition A.2. Un maillage \mathcal{T}_h d'un domaine Ω est dit **conforme** si les conditions suivantes sont vérifiées :

- Les éléments recouvrent le domaine $\bar{\Omega} = \bigcup_{K \in \mathcal{T}_h} K$,
- Chaque élément a un intérieur non-vide,
- L'intersection de deux éléments du maillage est soit :
 - l'ensemble vide,
 - un point,
 - un arête,
 - une face,

dont les sommets sont des nœuds de ces deux éléments.

Il existe 3 grandes familles [Owen, 1998, Frey and George, 2008] d'algorithmes de génération automatisée de maillages non-structurés.

- Les méthodes qui utilisent des arbres (Quadrees en 2D et Octrees en 3D).
- Les méthodes par avancée de front.
- Les méthodes de type Delaunay.

A.1 Les méthodes basées sur des arbres

La première famille est la génération de maillages basés sur une grille (Quadrees en dimension 2 et Octrees en dimension 3). Ces méthodes ont été développées à partir des années

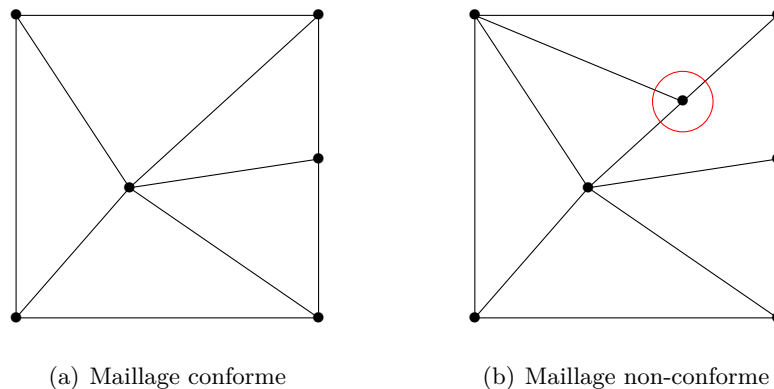


FIGURE A.2 – Exemples de maillages conforme (gauche) et non-conforme (droite).

80 [Yerry and Shephard, 1984, Cheng et al., 1988]. Elles sont basées sur des arbres dont le principe est que les feuilles sont récursivement subdivisées jusqu'à ce qu'une taille requise soit atteinte.

Le point de départ de ces méthodes est une géométrie et une description discrète de la géométrie (i.e. une frontière formée par des points, des segments ou des triangles suivant la dimension). Le principe de la méthode est, en fonction de cette description, de définir un arbre raffiné à certains endroits. Ensuite, à partir de cet arbre, un maillage est construit. La méthode peut être résumée en plusieurs étapes [George, 1992] :

1. *Création d'une boîte englobante.* Tous les éléments de la géométrie à mailler sont contenus dans cette boîte.
2. *Création d'un arbre.* Un arbre est créé en fonction de la position des sommets et des éléments. Pour cela, tous les points de la frontière sont parcourus, la feuille de l'arbre le contenant est recherchée et cette dernière est subdivisée en 4 (resp. 8) éléments de tailles identiques en dimension 2 (resp. 3). De la même manière, les éléments de la frontière sont parcourus et les feuilles qui contiennent plus d'un élément sont subdivisées. Finalement, chaque feuille de l'arbre ne doit contenir qu'au plus un seul sommet de la géométrie (elle peut ne pas en contenir) et ne doit contenir qu'au plus un élément.
3. *Équilibrage de l'arbre.* L'écart de niveau (de découpe) entre deux éléments voisins de l'arbre ne doit pas être plus grand qu'un. Si ce n'est pas le cas, certains éléments sont subdivisés. Cet équilibrage est utile pour avoir un maillage avec des tailles équilibrées.
4. *Création du maillage.* Le maillage de la boîte est ensuite généré en prenant en compte les sommets de la géométrie et les feuilles de l'arbre. La dernière étape consiste à éliminer l'extérieur, c'est à dire les éléments de la boîte qui n'appartiennent pas à la géométrie. Cette dernière est facile à réaliser puisque la frontière est préservée et donc les éléments sont soit à l'intérieur, soit à l'extérieur.

La construction par cette méthode est coûteuse car beaucoup de recherches dans l'arbre sont nécessaires. Pour rester performante, la gestion par structures de données est primordiale. Il faut également noter qu'à part l'équilibrage de l'arbre qui permet d'avoir des éléments de tailles similaires, il n'y a aucun critère de qualité pris en compte lors de la construction du maillage. Il

peut y avoir donc des éléments assez plats par exemple. Une étape d'optimisation de ce maillage est donc en général nécessaire.

A.2 Les méthodes par avancée de front

La deuxième famille concerne les méthodes par avancée de front. Ce type de méthode a été introduit dans la thèse de John Alan George en 1971 [George, 1971], puis longuement étudié notamment par Lo [Lo, 1985, Lo, 1991a, Lo, 1991b] et Löhner [Löhner and Parikh, 1988, Löhner, 1996, Löhner, 1997]. Certains maillages sont basés sur cette méthode [Schöberl, 1997]. Adapté à n'importe quelle géométrie arbitraire, le maillage est composé de triangles ou de tétraèdres et est construit à partir du contour du domaine. Dans un premier temps, un maillage du bord de l'objet est construit, à partir d'un critère sur la taille des éléments. Ce maillage du bord constitue le front initial. Ensuite, l'objet est rempli par des simplexes ajoutés un à un et construits à partir du front. Ainsi, à chaque étape de l'algorithme, le front est mis à jour et sépare la partie de l'objet maillée de celle qui ne l'est pas encore. L'algorithme s'arrête lorsque l'ensemble des simplexes recouvre entièrement l'objet. A l'inverse des méthodes basées sur des arbres présentées précédemment, les méthodes frontales permettent de construire des maillages dont les éléments sont de bonne qualité près des bords de l'objet. Néanmoins, afin d'éviter que deux fronts de tailles différentes [George and Seveno, 1994] ne se rencontrent au centre de l'objet (sinon l'algorithme n'a pas d'autre choix que de créer des éléments de mauvaise qualité), il est nécessaire d'avoir un maillage du bord bien adapté. Cette méthode n'est néanmoins pas la plus performante et peut être en difficulté pour traiter des cas en dimension trois. De plus, si le domaine est fortement non-convexe ou en présence de nombreux trous ou obstacles (la frontière est composée de plusieurs composantes connexes), cette approche peut ne pas converger.

A.3 Les méthodes de maillage de Delaunay

La troisième famille concerne les méthodes permettant de générer des maillages par raffinement de triangulations de Delaunay [Delaunay, 1934]. Un maillage initial est d'abord construit à partir de la triangulation de Delaunay d'un ensemble de points. Ce maillage « grossier » est ensuite raffiné par ajout de points. Cette méthode est basée sur le critère de la *sphère vide* [Delaunay, 1934, George, 1991, Dobrzynski, 2005]. Ce dernier est rappelé dans la définition A.3 qui explicite ce qu'est une triangulation de Delaunay. L'algorithme de Bowyer-Watson [Bowyer, 1981, Watson, 1981], développé et publié indépendamment la même année (1981) par Bowyer et Watson, fournit un moyen efficace pour calculer les triangulations de Delaunay en dimension quelconque.

Définition A.3. Soit \mathcal{T}_h un maillage du domaine. Cette tessellation est une triangulation de Delaunay si les boules ouvertes $\mathcal{B}(K)$ circonscrites aux éléments K ne contiennent aucun autre sommet du maillage.

Le rapport de George [George, 1991] explique comment construire un maillage par la méthode de Delaunay en dimension 2. La méthodologie générale est donnée dans l'algorithme 12.

La méthode repose sur deux aspects fondamentaux qui sont l'insertion de points dans le maillage respectant le critère de Delaunay (i.e. il y a éventuellement une modification locale du maillage autour du sommet ajouté) et la recherche de la frontière du domaine.

Nous décrivons maintenant comment fonctionne l'insertion d'un point P dans un maillage existant \mathcal{T}_h . Nous supposons que \mathcal{T}_h est un maillage de Delaunay. Cela signifie d'après la définition

Algorithme 12 : La méthode de Delaunay

1. *Création d'un nuage de points associé aux données.* Des points sur la géométrie du domaine à mailler sont créés en se basant sur une taille caractéristique.
 2. *Création d'une boîte englobante.* Tous les éléments de la géométrie à mailler sont contenus dans cette boîte.
 3. *Maillage de cette boîte à l'aide de simplexes.*
 4. *Insertion, un à un, des points du nuage pour définir un maillage initial.*
 5. *Recherche dans le maillage précédent des arêtes du contour donné.* Il peut arriver que, lorsque le critère de Delaunay n'est pas vérifié, la frontière de l'objet soit modifiée. Cette étape vise donc à modifier le maillage pour recréer éventuellement les arêtes manquantes et donc retrouver la géométrie initiale.
 6. *Marquage des éléments du maillage par rapport à la géométrie du domaine.* Cette étape permet de classer les éléments entre ceux qui sont à l'intérieur du domaine et ceux qui sont à l'extérieur.
 7. *Création des points internes et insertion de ceux-ci.*
 8. *Élimination des éléments extérieurs.*
-

A.3 que le cercle circonscrit de chacun des éléments ne contient aucun autre point que ses propres sommets. L'algorithme permettant de construire le nouveau maillage par raffinement de l'ancien via l'ajout d'un nouveau point est décrit dans l'algorithme 13. Il est basé sur la construction d'une cavité S constituée de tous les éléments dont le cercle circonscrit contient le point P (étapes 2, 3 et 4 de l'algorithme 13). Une fois que la cavité est repérée, toutes les arêtes externes de celle-ci sont définies, formant ainsi un contour fermé. Ce contour est ensuite utilisé pour modifier le maillage (étape 5 de l'algorithme 13).

Algorithme 13 : Insertion d'un point dans un maillage en respectant le critère de Delaunay (en dimension 2).

1. Trouver l'élément (s'il se trouve à l'intérieur) ou les éléments (s'il se trouve sur le bord) du maillage contenant le point P .
 2. Mettre ce(s) élément(s) dans une pile S .
 3. Compléter récursivement S en ajoutant tous les éléments du maillage dont le cercle circonscrit contient P .
 4. Construire la cavité. Trouver $F_1 \dots F_p$ toutes les arêtes externes de S c'est à dire les arêtes non communes à plusieurs éléments.
 5. Construire le nouveau maillage comme $\mathcal{T}_h = (\mathcal{T}_h \setminus S) \cup \{F_j, P\}_{j=1, \dots, p}$.
-

Le deuxième aspect fondamental concerne la recherche de la frontière du domaine. En effet, à cause des retournements d'arêtes [Dobrzynski, 2005] qui sont effectués et de l'insertion de points avec la cavité qui est généré, il est possible que le bord ne soit plus respecté. Une étape est donc nécessaire pour « retrouver » la frontière. Pour cela, une idée simple [George, 1991] en dimension deux est de définir un polygone contenant tous les triangles coupant l'arête à retrouver et d'effectuer des retournements d'arêtes successifs jusqu'à ce que l'arête soit retrouvée. L'idée

est similaire pour la dimension trois. Il est possible de générer un maillage en conservant la frontière du domaine [George et al., 1991], mais cela impose d'autres contraintes et il est plus simple de faire cette étape pour retrouver le contour du domaine.

A.4 Bilan sur les techniques de génération de maillage

En ce qui nous concerne, nous nous intéressons aux techniques de maillage puisque pour chaque élément du domaine de calcul, nous allons construire un maillage de représentation de l'élément de référence. Nous nous intéressons à une méthode itérative basée sur un processus d'ajouts de points. Ainsi, la méthode de Delaunay semble tout à fait pertinente. Les méthodes par avancée de front ne semblent pas facilement utilisables car on ne sait pas *a priori* où se trouvent les points à ajouter. Ainsi, l'introduction d'un front qui avance n'est pas envisageable car on ne peut pas assurer ne pas revenir « en arrière ». Les méthodes basées sur des arbres ne sont pas non plus envisageables car il n'y a pas de géométrie au départ puisque seule la cellule de référence est connue lorsque l'algorithme de construction du maillage de représentation débute. Pour toutes ces raisons, l'algorithme de Delaunay, qui est principalement basé sur un processus d'ajouts de points dans une tessellation existante, est choisi.

Annexe B

Fonctions de base et transformations géométriques usuelles

Dans cette annexe, nous présentons quelques fonctions de base et transformations géométriques usuelles. L'élément de référence est défini selon la fonction de base considérée. Ainsi, pour un même élément où plusieurs champs sont définis dessus, pour chaque champ, un élément de référence différent peut lui être associé. Il convient donc de bien préciser l'élément de référence utilisé et donc la transformation géométrique utilisée lorsque nous définissons une fonction de base.

B.1 Interpolation de Lagrange, polynômes de Gauss, de Gauss-Lobatto, de Chebyshev

Parmi les fonctions de base, les plus classiques sont les polynômes de Lagrange. Un ensemble de points (qui peuvent être les nœuds de l'élément est choisi) est introduit et pour chacun de ces points, un polynôme de Lagrange lui est associé. La particularité de ce dernier est qu'il vaut 1 au point qui lui est associé et 0 à tous les autres.

Interpolation de Lagrange

Soit x_0, x_1, \dots, x_n $n + 1$ réels distincts. Nous définissons $n + 1$ polynômes l_i , pour $i = 0$ à n par

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}. \quad (\text{B.1})$$

Aux points considérés, ces polynômes vérifient

$$l_i(x_j) = \delta_{ij}, \quad (\text{B.2})$$

où δ est le symbole de Kronecker. Les polynômes de Gauss, de Gauss-Lobatto ou encore de Chebyshev sont des polynômes de Lagrange définis en des points particuliers choisis pour vérifier certaines propriétés.

Points de Gauss

Les points de Gauss sur le segment $[-1,1]$ sont donnés dans les tableaux B.1, B.2. et B.3 jusqu'à l'ordre 10.

| | Ordre 1 | Ordre 2 | Ordre 3 | Ordre 4 |
|-------|-----------------------|-----------------------|---|---|
| x_0 | $-\frac{1}{\sqrt{3}}$ | $-\sqrt{\frac{3}{5}}$ | $-\sqrt{\frac{3}{7} + \frac{2}{35}\sqrt{30}}$ | $-\sqrt{\frac{5}{9} + \frac{2}{63}\sqrt{70}}$ |
| x_1 | $\frac{1}{\sqrt{3}}$ | 0. | $-\sqrt{\frac{3}{7} - \frac{2}{35}\sqrt{30}}$ | $-\sqrt{\frac{5}{9} - \frac{2}{63}\sqrt{70}}$ |
| x_2 | | $\sqrt{\frac{3}{5}}$ | $\sqrt{\frac{3}{7} - \frac{2}{35}\sqrt{30}}$ | 0. |
| x_3 | | | $\sqrt{\frac{3}{7} + \frac{2}{35}\sqrt{30}}$ | $\sqrt{\frac{5}{9} - \frac{2}{63}\sqrt{70}}$ |
| x_4 | | | | $\sqrt{\frac{5}{9} + \frac{2}{63}\sqrt{70}}$ |

TABLE B.1 – Points de Gauss sur le segment $[-1,1]$ jusqu'à l'ordre 4.

| | Ordre 5 | Ordre 6 | Ordre 7 |
|-------|---------------------|---------------------|---------------------|
| x_0 | -0.9324695142031521 | -0.9491079123427585 | -0.9602898564975363 |
| x_1 | -0.6612093864662645 | -0.7415311855993945 | -0.7966664774136267 |
| x_2 | -0.2386191860831969 | -0.4058451513773972 | -0.5255324099163290 |
| x_3 | 0.2386191860831969 | 0. | -0.1834346424956498 |
| x_4 | 0.6612093864662645 | 0.4058451513773972 | 0.1834346424956498 |
| x_5 | 0.9324695142031521 | 0.7415311855993945 | 0.5255324099163290 |
| x_6 | | 0.9491079123427585 | 0.7966664774136267 |
| x_7 | | | 0.9602898564975363 |

TABLE B.2 – Points de Gauss sur le segment $[-1,1]$ pour les ordres 5, 6 et 7.

| | Ordre 8 | Ordre 9 | Ordre 10 |
|----------|---------------------|---------------------|---------------------|
| x_0 | -0.9681602395076261 | -0.9739065285171717 | -0.9782286581460570 |
| x_1 | -0.8360311073266358 | -0.8650633666889845 | -0.8870625997680953 |
| x_2 | -0.6133714327005904 | -0.6794095682990244 | -0.7301520055740494 |
| x_3 | -0.3242534234038089 | -0.4333953941292472 | -0.5190961292068118 |
| x_4 | 0. | -0.1488743389816312 | -0.2695431559523450 |
| x_5 | 0.3242534234038089 | 0.1488743389816312 | 0. |
| x_6 | 0.6133714327005904 | 0.4333953941292472 | 0.2695431559523450 |
| x_7 | 0.8360311073266358 | 0.6794095682990244 | 0.5190961292068118 |
| x_8 | 0.9681602395076261 | 0.8650633666889845 | |
| x_9 | | 0.9739065285171717 | 0.8870625997680953 |
| x_{10} | | | 0.9782286581460570 |

TABLE B.3 – Points de Gauss sur le segment $[-1,1]$ pour les ordres 8, 9 et 10.

Points de Gauss-Lobatto

| | Ordre 1 | Ordre 2 | Ordre 3 | Ordre 4 |
|-------|---------|---------|-----------------------|-----------------------|
| x_0 | -1. | -1. | -1. | -1. |
| x_1 | 1. | 0. | $-\frac{1}{\sqrt{5}}$ | $-\sqrt{\frac{3}{7}}$ |
| x_2 | | 1. | $\frac{1}{\sqrt{5}}$ | 0. |
| x_3 | | | 1. | $\sqrt{\frac{3}{7}}$ |
| x_4 | | | | 1. |

TABLE B.4 – Points de Gauss-Lobatto sur le segment $[-1,1]$ jusqu'à l'ordre 4.

Les points de Gauss-Lobatto sont donnés sur le segment $[-1,1]$ dans les tableaux B.4, B.5. et B.6 jusqu'à l'ordre 10.

Points de Chebyshev

Les polynômes de Chebyshev sont très utilisés en interpolation polynomiale car ils ont la propriété de minimiser les effets du phénomène de Runge. Les points de Chebyshev (avec n points) sont donnés sur le segment $[-1,1]$ par la formule suivante

$$x_k = \cos\left(\frac{2k-1}{2n}\pi\right), \quad \forall k = 1, \dots, n. \quad (\text{B.3})$$

| | Ordre 5 | Ordre 6 | Ordre 7 |
|-------|---------------------------|---------------------------|---------------------------|
| x_0 | -1. | -1. | -1. |
| x_1 | -0.765055323929464692851 | -0.830223896278566929872 | -0.8717401485096066153374 |
| x_2 | -0.2852315164806450963142 | -0.4688487934707142138038 | -0.5917001814331423021445 |
| x_3 | 0.2852315164806450963142 | 0. | -0.2092992179024788687687 |
| x_4 | 0.765055323929464692851 | 0.4688487934707142138038 | 0.2092992179024788687687 |
| x_5 | 1. | 0.830223896278566929872 | 0.5917001814331423021445 |
| x_6 | | 1. | 0.8717401485096066153374 |
| x_7 | | | 1. |

TABLE B.5 – Points de Gauss-Lobatto sur le segment $[-1,1]$ pour les ordres 5, 6 et 7.

| | Ordre 8 | Ordre 9 | Ordre 10 |
|----------|---------------------------|---------------------------|---------------------------|
| x_0 | -1. | -1. | -1. |
| x_1 | -0.8997579954114601573123 | -0.9195339081664588138289 | -0.9340014304080591343323 |
| x_2 | -0.6771862795107377534459 | -0.7387738651055050750031 | -0.7844834736631444186224 |
| x_3 | -0.3631174638261781587108 | -0.4779249498104444956612 | -0.565235326996205006471 |
| x_4 | 0. | -0.1652789576663870246262 | -0.2957581355869393914319 |
| x_5 | 0.3631174638261781587108 | 0.1652789576663870246262 | 0. |
| x_6 | 0.6771862795107377534459 | 0.4779249498104444956612 | 0.2957581355869393914319 |
| x_7 | 0.8997579954114601573123 | 0.7387738651055050750031 | 0.565235326996205006471 |
| x_8 | 1. | 0.9195339081664588138289 | 0.7844834736631444186224 |
| x_9 | | 1. | 0.9340014304080591343323 |
| x_{10} | | | 1. |

TABLE B.6 – Points de Gauss-Lobatto sur le segment $[-1,1]$ pour les ordres 8, 9 et 10.



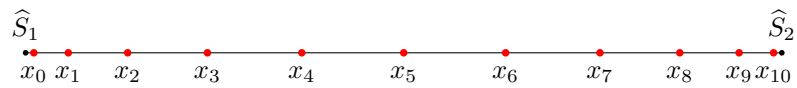
(a) Points de Gauss d'ordre 5



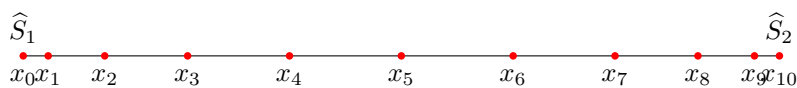
(b) Points de Gauss Lobatto d'ordre 5



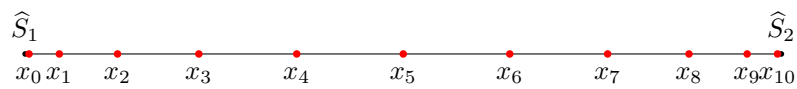
(c) Points de Chebyshev d'ordre 5



(d) Points de Gauss d'ordre 10



(e) Points de Gauss Lobatto d'ordre 10



(f) Points de Chebyshev d'ordre 10

FIGURE B.1 – Localisation de points de Gauss, de Gauss-Lobatto et de Chebyshev sur le segment de référence.

La localisation des points de Gauss, de Gauss-Lobatto et de Chebyshev sur le segment de référence pour les ordre 5 et 10 est donnée sur la figure B.1.

B.2 Quelques transformations géométriques usuelles

Dans cette section, nous rappelons quelques transformations géométriques usuelles. Il est important, quand on définit une fonction de base de donner l'élément de référence et la transformation géométrique associée.

Quelques transformations géométriques en dimension 1

Nous considérons tout d'abord l'élément de référence \widehat{K} défini comme le segment unité de coordonnées $\widehat{S}_1 = (0)$ et $\widehat{S}_2 = (1)$ et le segment K défini à partir de deux sommets $S_1 = (x_1, y_1, z_1)$ et $S_2 = (x_2, y_2, z_2)$. La transformation géométrique (voir figure B.2) est dans ce cas linéaire et est donnée par :

$$g_K : \left\{ \begin{array}{l} \widehat{K} = [0, 1] \longrightarrow K \\ t \longmapsto \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \varphi_1(t) + \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} \varphi_2(t), \end{array} \right. \quad (\text{B.4})$$

où φ_1 et φ_2 sont les fonctions de base P^1 suivantes :

- $\varphi_1(t) = 1 - t$,
- $\varphi_2(t) = t$.

On peut aisément vérifier que $\varphi_1(\widehat{S}_1) = 1$, $\varphi_1(\widehat{S}_2) = 0$, $\varphi_2(\widehat{S}_1) = 0$ et $\varphi_2(\widehat{S}_2) = 1$. Le choix du segment unité comme cellule de référence est intéressante car elle permet de manipuler des fonctions de base très simples. Néanmoins, il est tout à fait possible de faire d'autres choix.

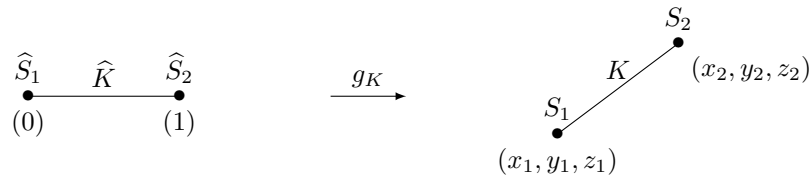


FIGURE B.2 – Transformation géométrique entre le segment unité et un segment quelconque.

Un second choix par exemple est de définir l'élément de référence comme le segment $[-1, 1]$ (voir figure B.3). La transformation géométrique est donc toujours affine mais est modifiée de la façon suivante

$$g_K : \left\{ \begin{array}{l} \widehat{K} = [-1, 1] \longrightarrow K \\ t \longmapsto \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \varphi_1(t) + \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} \varphi_2(t), \end{array} \right. \quad (\text{B.5})$$

où cette fois-ci les fonctions φ_1 et φ_2 sont

- $\varphi_1(t) = \frac{1-t}{2}$,
- $\varphi_2(t) = \frac{1+t}{2}$.

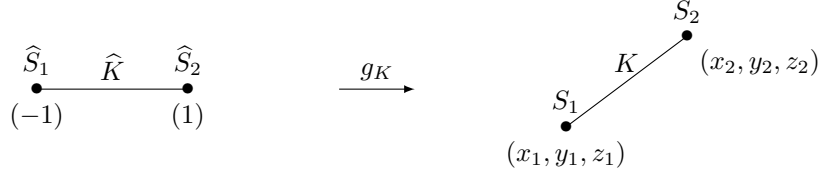


FIGURE B.3 – Transformation géométrique entre la cellule de référence $[-1,1]$ et un segment quelconque.

Quelques transformations géométriques en dimension 2

De manière générale, lorsque l'élément de référence est de dimension 2, la transformation géométrique s'écrit :

$$g_K : \left\{ \begin{array}{l} \widehat{K} \longrightarrow K \\ \begin{pmatrix} r \\ s \end{pmatrix} \longmapsto \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \sum_{i=1}^N \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \varphi_i(r, s), \end{array} \right. \quad (\text{B.6})$$

où \widehat{K} est l'élément de référence qui varie suivant le cas considéré, N est le nombre de points de contrôle et $\{\varphi_i\}_{i=1,\dots,N}$ sont les fonctions de base associées aux points $S_i = (x_i, y_i, z_i)$.

Nous explicitons à présent quelques exemples de transformations géométriques usuelles de dimension 2.

Éléments triangulaires

La première transformation usuelle est le triangle défini par $N = 3$ points qui sont donc les sommets du triangle. Cette transformation, dite P^1 , est affine. La figure B.4 illustre une telle transformation dans le cas où la cellule de référence est le triangle unité.

Les 3 fonctions de base P^1 , que l'on note ici $\{\lambda_i\}_{i=1,\dots,3}$, sont les suivantes :

- $\lambda_1(r, s) = 1 - r - s$,
- $\lambda_2(r, s) = r$,
- $\lambda_3(r, s) = s$.

Si maintenant on s'intéresse à la transformation P^2 sur triangle, celle-ci est définie à partir de $N = 6$ points comme le montre la figure B.5.

La transformation géométrique g_K n'est plus affine. Les 6 fonctions de base, peuvent se construire à l'aide des 3 fonctions de base P^1 données précédemment, et s'écrivent :

- $\varphi_1(r, s) = 2\lambda_1(\lambda_1 - 0.5)$,

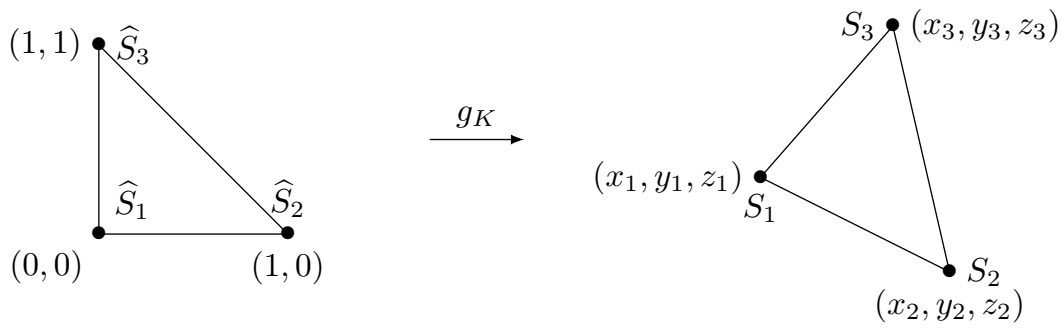


FIGURE B.4 – Transformation géométrique entre le triangle unité et un triangle P^1 quelconque.

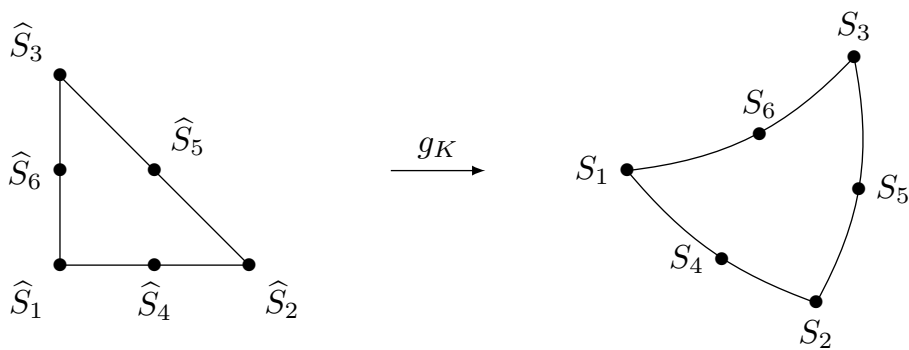


FIGURE B.5 – Transformation géométrique entre le triangle de référence et un triangle P^2 quelconque.

- $\varphi_2(r, s) = 2\lambda_2(\lambda_2 - 0.5)$,
- $\varphi_3(r, s) = 2\lambda_3(\lambda_3 - 0.5)$,
- $\varphi_4(r, s) = 4\lambda_1\lambda_2$,
- $\varphi_5(r, s) = 4\lambda_2\lambda_3$,
- $\varphi_6(r, s) = 4\lambda_1\lambda_3$.

Éléments quadrangulaires

L'élément quadrangulaire le plus simple est l'élément fini quadrilatéral de degré 1, le Q^1 à $N = 4$ nœuds [George and Borouchaki, 2012]. L'élément réel considéré est composé de 4 sommets $S_i = (x_i, y_i)_{i=1, \dots, 4}$ et la transformation géométrique g_K vérifie $S_i = g_K(\widehat{S}_i)$ pour $i = 1, \dots, 4$. La transformation géométrique n'est pas affine. La figure B.6 illustre cette transformation géométrique.

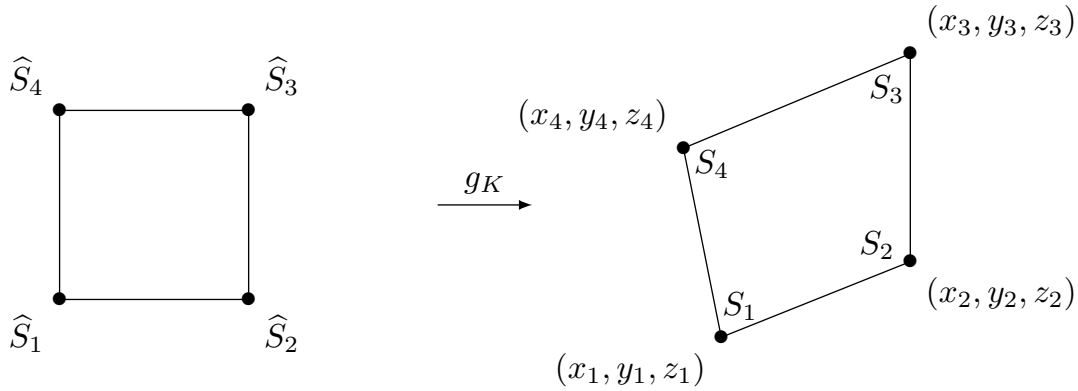


FIGURE B.6 – Transformation géométrique entre la cellule de référence et un quadrangle Q^1 quelconque.

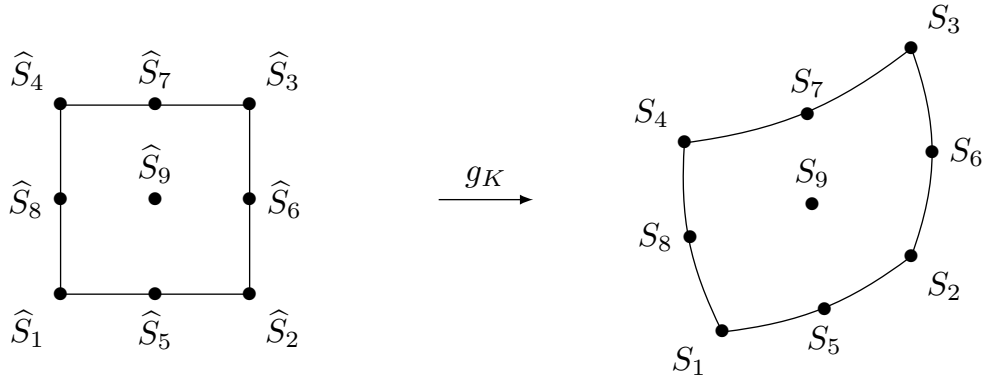
Le tableau B.7 donne les fonctions de base φ_i pour deux cellules de référence : le carré unité et le carré $[-1, 1] \times [-1, 1]$. Le choix de la cellule de référence dépend de la solution numérique. Par exemple, si des polynômes de Gauss sont utilisés, la cellule de référence est le carré unité alors que pour des polynômes de Gauss-Lobatto, la cellule de référence est le carré $[-1, 1] \times [-1, 1]$.

| \widehat{K} | $\varphi_1(r, s)$ | $\varphi_2(r, s)$ | $\varphi_3(r, s)$ | $\varphi_4(r, s)$ |
|--------------------------|------------------------|------------------------|------------------------|------------------------|
| $[0, 1] \times [0, 1]$ | $(1-r)(1-s)$ | $r(1-s)$ | rs | $r(1-s)$ |
| $[-1, 1] \times [-1, 1]$ | $\frac{(1-r)(1-s)}{4}$ | $\frac{(1+r)(1-s)}{4}$ | $\frac{(1+r)(1+s)}{4}$ | $\frac{(1-r)(1+s)}{4}$ |

TABLE B.7 – Construction des transformations géométriques pour l'élément quadrangulaire Q^1 .

Lorsque l'on considère un quadrangle Q^2 , $N = 9$ points sont pris en compte (cf. figure B.7). Les fonctions de base dépendent toujours de l'élément de référence. Par exemple, pour $\widehat{K} = [0, 1] \times [0, 1]$, elles sont données par :

- $\varphi_1(r, s) = \frac{rs(1-r)(1-s)}{4}$,


 FIGURE B.7 – Transformation géométrique entre le carré unité et un quadrangle Q^2 quelconque.

- $\varphi_2(r, s) = \frac{-rs(1+r)(1-s)}{4}$,
- $\varphi_3(r, s) = \frac{rs(1+r)(1+s)}{4}$,
- $\varphi_4(r, s) = \frac{-rs(1-r)(1+s)}{4}$,
- $\varphi_5(r, s) = \frac{-(1+r)s(1-r)(1-s)}{2}$,
- $\varphi_6(r, s) = \frac{r(1+s)(1+r)(1-s)}{2}$,
- $\varphi_7(r, s) = \frac{(1+r)s(1-r)(1+s)}{2}$,
- $\varphi_8(r, s) = \frac{-r(1+s)(1-r)(1-s)}{2}$,
- $\varphi_9(r, s) = (1+r)(1+s)(1-r)(1-s)$.

Quelques transformations géométriques en dimension 3

Lorsque l'élément de référence est de dimension 3, la transformation géométrique s'écrit de manière générale comme

$$g_K : \left| \begin{array}{l} \widehat{K} \longrightarrow K \\ \begin{pmatrix} r \\ s \\ t \end{pmatrix} \longmapsto \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \sum_{i=1}^N \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \varphi_i(r, s), \end{array} \right. \quad (\text{B.7})$$

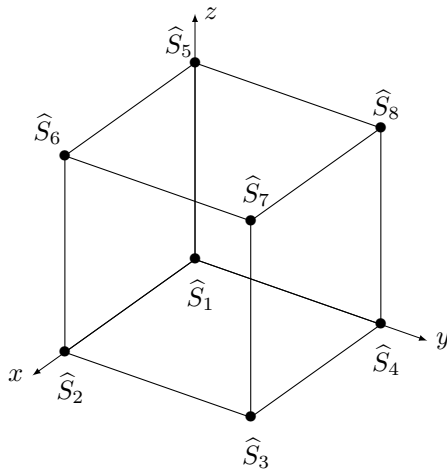
où \widehat{K} est l'élément de référence, N le nombre de points de contrôle et $\{\varphi_i\}_{i=1, \dots, N}$ sont les fonctions de base associées aux points $S_i = (x_i, y_i, z_i)$.

Nous explicitons les fonctions de base pour deux cas hexaédriques Q^1 . La numérotation des sommets est donnée sur la figure B.8. Pour le premier cas, l'élément de référence est le cube unité, les 8 fonctions de base s'écrivent alors

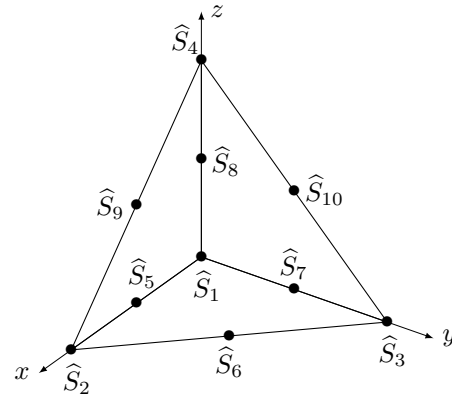
- $\varphi_1(r, s, t) = (1 - r)(1 - s)(1 - t)$,
- $\varphi_2(r, s, t) = r(1 - s)(1 - t)$,
- $\varphi_3(r, s, t) = rs(1 - t)$,
- $\varphi_4(r, s, t) = (1 - r)s(1 - t)$,
- $\varphi_5(r, s, t) = (1 - r)(1 - s)t$,
- $\varphi_6(r, s, t) = r(1 - s)t$,
- $\varphi_7(r, s, t) = rst$,
- $\varphi_8(r, s, t) = (1 - r)st$.

Si maintenant la cellule de référence est le cube $[-1, 1]^3$, les 8 fonctions de base s'écrivent

- $\varphi_1(r, s, t) = (1 - r)(1 - s)(1 - t)$,
- $\varphi_2(r, s, t) = (1 - r)(1 - s)(1 + t)$,
- $\varphi_3(r, s, t) = (1 - r)(1 + s)(1 + t)$,
- $\varphi_4(r, s, t) = (1 - r)(1 + s)(1 - t)$,
- $\varphi_5(r, s, t) = (1 + r)(1 - s)(1 - t)$,
- $\varphi_6(r, s, t) = (1 + r)(1 - s)(1 + t)$,
- $\varphi_7(r, s, t) = (1 + r)(1 + s)(1 + t)$,
- $\varphi_8(r, s, t) = (1 + r)(1 + s)(1 - t)$.



(a) Pour l'hexaèdre



(b) Pour le tétraèdre

FIGURE B.8 – Numérotations des sommets pour l'hexaèdre et le tétraèdre.

Nous donnons maintenant les 4 fonctions de base P_1 notées $(\lambda_i)_{i=1,\dots,4}$ associées aux 4 sommets du tétraèdre numérotés comme sur la figure B.8.b :

- $\lambda_1(r, s, t) = 1 - r - s - t$,
- $\lambda_2(r, s, t) = r$,
- $\lambda_3(r, s, t) = s$,
- $\lambda_4(r, s, t) = t$.

A partir de ces fonctions de base, et en introduisant les centres des arêtes comme sur la figure B.8.b, on peut construire les fonctions de base P_2 notées $(\varphi_i)_{i=1,\dots,10}$ associées aux 10 nœuds.

- $\varphi_1(r, s, t) = (1 - r - s - t)(1 - 2r - 2s - 2t)$,
- $\varphi_2(r, s, t) = r(2r - 1)$,
- $\varphi_3(r, s, t) = s(2s - 1)$,
- $\varphi_4(r, s, t) = t(2t - 1)$,
- $\varphi_5(r, s, t) = 4(1 - r - s - t)r$,
- $\varphi_6(r, s, t) = 4rs$,
- $\varphi_7(r, s, t) = 4(1 - r - s - t)s$,
- $\varphi_8(r, s, t) = 4(1 - r - s - t)t$,
- $\varphi_9(r, s, t) = 4rt$,
- $\varphi_{10}(r, s, t) = 4st$.

B.3 Quelques fonctions de base usuelles

Certaines solutions numériques sont construites à partir d'éléments isoparamétriques. Dans ce cas, les fonctions de base qui décrivent les champs sont les mêmes que les fonctions de base qui décrivent la géométrie (par exemple triangles P^1 , P^2 , quadrangles Q^1 , Q^2 ...). Cependant, souvent les fonctions de base qui décrivent les champs sont plus riches que les fonctions de base géométriques. En effet, en général, les transformations géométriques sont au plus quadratiques ou cubiques. Par contre, pour approcher les champs, des fonctions plus complexes sont souvent utilisées comme des polynômes de Gauss, de Gauss-Lobatto...

Ces fonctions étant classiques, nous ne donnons pas plus de détails. Nous rappelons simplement comment sont définis les fonctions de Raviart-Thomas d'ordre p sur triangle P^1 que nous avons utilisé dans la section 4.2.3. Le triangle de référence est défini classiquement par les trois points $\hat{S}_1 = (0, 0)$, $\hat{S}_2 = (1, 0)$ et $\hat{S}_3 = (0, 1)$. Les arêtes sont numérotées par rapport au numéro du sommet auxquelles elles sont opposées.

Les fonctions de base de Raviart-Thomas d'ordre 0 sont définies comme

- $\hat{e}_1(\xi, \eta) = \begin{pmatrix} \xi \\ \eta \end{pmatrix}$,
- $\hat{e}_2(\xi, \eta) = \begin{pmatrix} \xi - 1 \\ \eta \end{pmatrix}$,
- $\hat{e}_3(\xi, \eta) = \begin{pmatrix} \xi \\ \eta - 1 \end{pmatrix}$,

et sont représentées sur la figure B.9.

Pour définir les autres fonctions, il est nécessaire d'introduire deux fonctions :

- $\hat{e}_4(\xi, \eta) = \eta \begin{pmatrix} \xi \\ \eta - 1 \end{pmatrix}$,

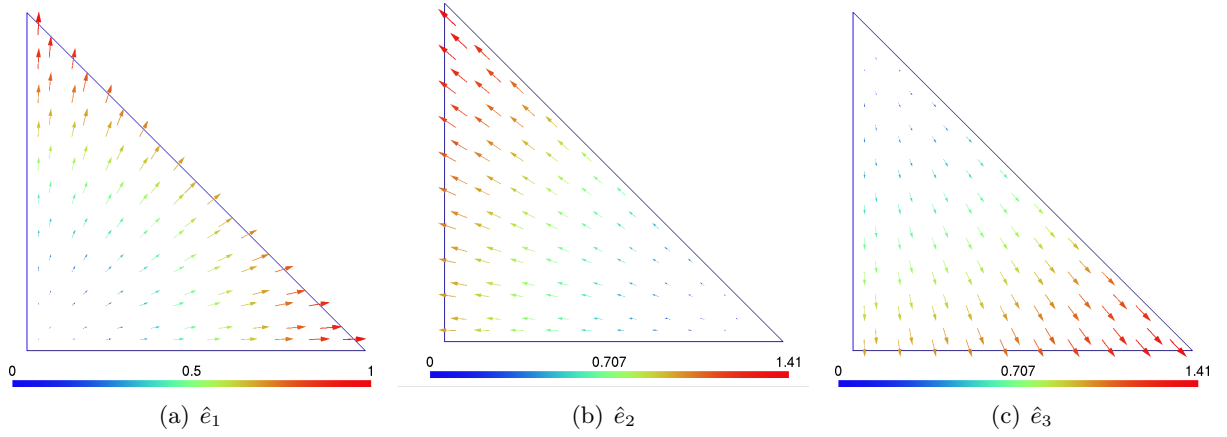


FIGURE B.9 – Fonctions de Raviart-Thomas d'ordre 0.

$$\bullet \hat{e}_5(\xi, \eta) = \xi \begin{pmatrix} \xi - 1 \\ \eta \end{pmatrix}.$$

Les fonctions de base de Raviart-Thomas d'ordre p sont construites d'une manière similaire. Pour cela, on introduit les $p+1$ points de Gauss sur $[0,1]$ que l'on note x_l . On note l_j le polynôme de Lagrange de degré p tel que $l_j(x_l) = \delta_{jl}$. Les fonctions de base normales de Raviart Thomas sur l'élément de référence sont définies à partir des polynômes de Lagrange l_j et des fonctions \hat{e}_1 , \hat{e}_2 et \hat{e}_3 :

- $\hat{\Phi}_j^1(\xi, \eta) = l_j(\eta)\hat{e}_1(\xi, \eta), \quad j = 1, \dots, p+1,$
- $\hat{\Phi}_j^2(\xi, \eta) = l_{p+2-j}(\eta)\hat{e}_2(\xi, \eta), \quad j = 1, \dots, p+1,$
- $\hat{\Phi}_j^3(\xi, \eta) = l_j(\xi)\hat{e}_3(\xi, \eta), \quad j = 1, \dots, p+1.$

Les fonctions de base d'ordre p non-normales sont définies comme

- $\hat{\Phi}_j^4(\xi, \eta) = b_j(\eta, \eta)\hat{e}_4(\xi, \eta), \quad j = 1, \dots, p(p+1)/2,$
- $\hat{\Phi}_j^5(\xi, \eta) = b_j(\eta, \eta)\hat{e}_5(\xi, \eta), \quad j = 1, \dots, p(p+1)/2,$

où $\{b_j\}_{j=1, \dots, p(p+1)/2}$ désigne une base de $P^p(\hat{T})$, \hat{T} étant le triangle de référence.

Ces fonctions de base sont définies sur l'élément de référence, la transformation de Piola permet d'avoir les fonctions sur l'élément réel [Ervin, 2012].

Annexe C

Détails des calculs de la section 3.3.3

Dans cette annexe, nous détaillons les calculs de la section 3.3.3. Nous commençons par rappeler le fonctionnement de l'algorithme de Gauss-Newton. Ensuite, nous faisons quelques rappels de calcul différentiel qui seront utiles pour la suite. Puis, nous nous intéressons au problème d'optimisation dans le plan de la section 3.3.3.

C.1 L'algorithme de Gauss-Newton

La méthode de Gauss-Newton (ou Newton-Raphson en dimension un) est une méthode utilisée pour calculer les racines (également appelées zéros) d'une fonction à valeur réelle et définie par une seule variable [Ypma, 1995]. Il s'agit d'une méthode itérative qui cherche à chaque itération une meilleure approximation des racines. Notons f la fonction dont on cherche la racine et que l'on suppose dérivable en tout point. On part d'un point x_0 arbitraire. Le choix de ce point est important et, si possible, nous le choisissons proche de la racine cherchée (des estimations grossières par dichotomie peuvent être faites par exemple). Afin de construire une bonne approximation de ces racines, un développement de Taylor au premier ordre est utilisé. La fonction est approchée par sa tangente en ce point, à savoir :

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0). \quad (\text{C.1})$$

Comme nous cherchons les racines de f , nous calculons l'intersection de la droite tangente avec l'axe des racines :

$$0 = f(x_0) + f'(x_0)(x - x_0). \quad (\text{C.2})$$

Aussi, nous construisons par récurrence la suite

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad (\text{C.3})$$

où f' désigne la dérivée de f et $f'(x_k)$ est supposé être non nul puisque nous cherchons justement à l'annuler. Le point x_{k+1} est bien solution de l'équation affine

$$f(x_k) + f'(x_k)(x - x_k) = 0. \quad (\text{C.4})$$

En itérant de la sorte, on finit par trouver la solution à ce problème notée x^* . Il est nécessaire de fixer des critères d'arrêt. On peut en fixer plusieurs :

- Un critère sur la valeur de la fonction $|f(x_k)| < \varepsilon_1$, où $\varepsilon_1 \in \mathbb{R}^+$ est un paramètre à fixer assurant que la valeur au point considéré soit quasi-nulle.

- Un critère sur la stagnation des valeurs $|x_{k+1} - x_k| < \varepsilon_2$ qui permet d'arrêter l'algorithme dans le cas où les points que l'on trouve ne bougent quasiment plus. Nous devons donc fixer un deuxième paramètre $\varepsilon_2 \in \mathbb{R}^+$ qui en général est très petit.
- Un critère sur le nombre d'itérations maximales $k > N_{\max}$ où $N_{\max} \in \mathbb{N}$ représente le nombre d'itérations maximales.

L'algorithme 14 décrit, dans ce cadre général, l'algorithme de Newton-Raphson.

Algorithme 14 : Algorithme de Newton-Raphson

```

Données :  $x_0$  ; /* point de départ */
Résultat :  $x^*$  ; /* racine */
Initialisation des paramètres :  $\varepsilon_1, \varepsilon_2, N_{\max}, k = 0$  ;
Calcul de  $f(x_k)$  ;
tant que ( $|f(x_k)| \geq \varepsilon_1$ ) et ( $|x_{k+1} - x_k| \geq \varepsilon_2$ ) et ( $k < N_{\max}$ ) faire
  | Calcul de  $f(x_k)$  ;
  | Calcul de  $f'(x_k)$  ;
  | si  $f'(x_k) \neq 0$  alors
  |   |  $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$  ;
  |   |  $x_k = x_{k+1}$  ;
  | sinon
  |   |  $x_{k+1} = x_k$  ;
  | fin
  |  $k = k + 1$  ;
fin
 $x^* = x_k$  ;

```

Le principal intérêt de l'algorithme de Newton-Raphson est que sa convergence est quadratique. Il présente néanmoins quelques inconvénients. Il est nécessaire de connaître la valeur de la dérivée (ou une approximation) de la fonction considérée aux points considérés. Dans notre cas, ce n'est pas un problème puisque les dérivées des quantités manipulées sont connues. Lorsqu'il y a plusieurs racines, on ne va pouvoir en récupérer qu'une seule, en fonction du point de départ choisi, ce qui explique l'importance de ce choix.

C.2 Rappels de calcul différentiel

Les rappels donnés dans cette section s'inspirent du cours de calcul différentiel de Jean Pierre Raymond. Dans cette section, X et Y désignent des espaces vectoriels réels normés, f désigne une application d'un ouvert $U \subset X$ à valeurs dans Y . La première définition concerne la notion de dérivée.

Définition C.1 (Dérivée en un point suivant une direction). *Soient $a \in U$ et $d \in X$. La limite*

$$\lim_{\lambda \rightarrow 0} \frac{f(a + \lambda d) - f(a)}{\lambda} \tag{C.5}$$

lorsqu'elle existe est appelée dérivée de f au point a suivant la direction d et est notée $f'(a; d)$.

La notion de différentiabilité est introduite dans la définition suivante.

Définition C.2 (Dérivée de Fréchet). *Soit $a \in U$. Supposons qu'il existe un opérateur linéaire et continu $L \in \mathcal{L}(X; Y)$ et une application ε de X dans Y tels que*

$$f(a + d) = f(a) + Ld + \|d\|_X \varepsilon(d), \quad \text{avec } \lim_{\|d\|_X \rightarrow 0} \varepsilon(d) = 0. \quad (\text{C.6})$$

L'opérateur L est appelé différentielle de Fréchet de f au point a et f est dite Fréchet-différentiable (ou différentiable ou différentiable au sens de Fréchet) au point a . La différentielle de f au point a est également notée $Df(a)$.

Si l'application f est différentiable en a , alors l'opérateur L est unique ainsi que ε . Une autre définition de différentiabilité (au sens de Gâteaux) existe et fait intervenir les dérivées de Gâteaux. Les deux notions sont proches mais ne sont pas équivalentes. Dans la suite, nous n'utilisons que les différentielles au sens de Fréchet que l'on appelle simplement différentielles.

Remarque C.3 (Définition équivalente de la différentiabilité au sens de Fréchet). *L'application f est différentiable en a si et seulement si il existe un opérateur linéaire $L \in \mathcal{L}(X; Y)$ tel que*

$$\lim_{\|d\|_X \rightarrow 0} \frac{\|f(a + d) - f(a) - Ld\|_Y}{\|d\|_X} = 0. \quad (\text{C.7})$$

Les deux théorèmes suivants explicitent la différentielle d'une combinaison linéaire et la composée de deux applications différentiables.

Théorème C.4 (Différentielle d'une combinaison linéaire). *Soient X et Y des espaces vectoriels normés. Soient U un ouvert de X , f et g deux applications de U dans Y , $a \in U$ et $\lambda \in \mathbb{R}$.*

Si f et g sont différentiables en a , alors $\lambda f + g$ est différentiable en a et

$$D(\lambda f + g)(a) = \lambda Df(a) + Dg(a). \quad (\text{C.8})$$

Théorème C.5 (Différentielle de la composée de deux applications différentiables). *Soient X , Y et Z des espaces vectoriels normés. Soient U un ouvert de X , V un ouvert de Y , f une application de U dans V et g une application de V dans Z .*

Si f est différentiable en $a \in U$ et g est différentiable en $b = f(a)$, alors $g \circ f$ est différentiable en a et

$$D(g \circ f)(a) = Dg(f(a)) \circ Df(a). \quad (\text{C.9})$$

C.3 Cas où les surfaces se trouvent dans le plan

Tout d'abord, dans le plan, nous rappelons que la fonction à minimiser est la suivante :

$$h(t) = \langle g_K(\hat{x} + t\hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}}^2. \quad (\text{C.10})$$

où $t \in \mathbb{R}$, $\hat{x} \in \widehat{K}$, $\hat{n}_x \in \widehat{K}$, $\hat{a} \in \widehat{K}$, $g_K : \widehat{K} \rightarrow \mathbb{X}$ et $n' \in \mathbb{X}$. Nous cherchons à calculer sa dérivée h' . Pour cela, nous introduisons deux fonctions f et m telles que :

$$f : \begin{cases} \mathbb{R} & \longrightarrow \mathbb{X} \\ t & \longmapsto g_K(\hat{x} + t\hat{n}_x) - g_K(\hat{a}) \end{cases} \quad (\text{C.11})$$

et

$$m : \begin{cases} \mathbb{X} & \longrightarrow \mathbb{R} \\ x & \longmapsto \langle x, n' \rangle_{\mathbb{X}} \end{cases} \quad (\text{C.12})$$

La fonction h est donc définie comme une convolution :

$$h = m \circ f. \quad (\text{C.13})$$

Sa différentielle (voir le théorème C.5 pour la formule de convolution) est donnée par

$$Dh(t)(\tilde{t}) = Dm(f(t))(Df(t)(\tilde{t})), \quad \forall \tilde{t} \in \mathbb{R}. \quad (\text{C.14})$$

Nous nous intéressons donc maintenant aux différentielles de f et m . Or, la différentielle de f s'écrit simplement

$$Df(t)(\tilde{t}) = Jg_K(\hat{x} + t\hat{n}_x)\hat{n}_x\tilde{t}, \quad \forall \tilde{t} \in \mathbb{R}. \quad (\text{C.15})$$

La différentielle de m s'écrit quant à elle

$$Dm(x)(\tilde{x}) = 2\langle x, n' \rangle_{\mathbb{X}} \langle \tilde{x}, n' \rangle_{\mathbb{X}}, \quad \forall \tilde{x} \in \mathbb{X}. \quad (\text{C.16})$$

Pour s'en convaincre, on revient à la définition de la différentielle et on a

$$\begin{aligned} m(x + \tilde{x}) &= \langle x + \tilde{x}, n' \rangle_{\mathbb{X}}^2 = \langle x, n' \rangle_{\mathbb{X}}^2 + 2\langle x, n' \rangle_{\mathbb{X}} \langle \tilde{x}, n' \rangle_{\mathbb{X}} + \langle \tilde{x}, n' \rangle_{\mathbb{X}}^2 \\ &= m(x) + 2\langle x, n' \rangle_{\mathbb{X}} \langle \tilde{x}, n' \rangle_{\mathbb{X}} + m(\tilde{x}) \end{aligned} \quad (\text{C.17})$$

Or, nous pouvons facilement montrer que $m(\tilde{x}) = \mathcal{O}(\|\tilde{x}\|)$ et nous en déduisons la différentielle de m (C.16). En utilisant (C.14), (C.15), et (C.16), nous pouvons en déduire la différentielle de h :

$$Dh(t)(\tilde{t}) = 2\langle g_K(\hat{x} + t\hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}} \langle Jg_K(\hat{x} + t\hat{n}_x)\hat{n}_x\tilde{t}, n' \rangle_{\mathbb{X}}, \quad \forall \tilde{t} \in \mathbb{R}. \quad (\text{C.18})$$

Nous en déduisons simplement que la dérivée de h vaut

$$h'(t) = 2\langle g_K(\hat{x} + t\hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}} \langle Jg_K(\hat{x} + t\hat{n}_x)\hat{n}_x, n' \rangle_{\mathbb{X}}. \quad (\text{C.19})$$

Nous cherchons donc $t_{\tilde{h}}$ qui annule h' . Pour cela, nous appliquons la méthode de Gauss-Newton à une fonction \tilde{h} :

$$\tilde{h}(t) = \langle g_K(\hat{x} + t\hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}} \langle Jg_K(\hat{x} + t\hat{n}_x)\hat{n}_x, n' \rangle_{\mathbb{X}}. \quad (\text{C.20})$$

Nous souhaitons calculer la dérivée de \tilde{h} . Nous nous intéressons donc, pour un réel arbitraire \tilde{t} , à

$$\tilde{h}(t + \tilde{t}) = \langle g_K(\hat{x} + (t + \tilde{t})\hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}} \langle Jg_K(\hat{x} + (t + \tilde{t})\hat{n}_x)\hat{n}_x, n' \rangle_{\mathbb{X}}, \quad \forall \tilde{t} \in \mathbb{R}. \quad (\text{C.21})$$

Or,

$$\begin{aligned} g_K(\hat{x} + (t + \tilde{t})\hat{n}_x) &= g_K(\hat{x} + t\hat{n}_x) + Jg_K(\hat{x} + t\hat{n}_x)\tilde{t}\hat{n}_x + \mathcal{O}(\|\tilde{t}\|). \\ Jg_K(\hat{x} + (t + \tilde{t})\hat{n}_x)\hat{n}_x &= Jg_K(\hat{x} + t\hat{n}_x)\hat{n}_x + D(Jg_K(\hat{x} + t\hat{n}_x)\hat{n}_x)(\tilde{t}\hat{n}_x) + \mathcal{O}(\|\tilde{t}\|). \end{aligned} \quad (\text{C.22})$$

En utilisant les équations de (C.22) dans (C.21), nous obtenons :

$$\begin{aligned} \tilde{h}(t + \tilde{t}) &= \langle g_K(\hat{x} + t\hat{n}_x) - g_K(\hat{a}) + Jg_K(\hat{x} + t\hat{n}_x)\tilde{t}\hat{n}_x, n' \rangle_{\mathbb{X}} \cdot \langle Jg_K(\hat{x} + t\hat{n}_x)\hat{n}_x \\ &\quad + D(Jg_K(\hat{x} + t\hat{n}_x)\hat{n}_x)(\tilde{t}\hat{n}_x), n' \rangle_{\mathbb{X}} + \mathcal{O}(\|\tilde{t}\|), \end{aligned}$$

que l'on peut exprimer de la façon suivante

$$\begin{aligned}\tilde{h}(t + \tilde{t}) &= \langle g_K(\hat{x} + t\hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}} \cdot \langle Jg_K(\hat{x} + t\hat{n}_x) \hat{n}_x, n' \rangle_{\mathbb{X}} \\ &\quad + \tilde{t} \langle g_K(\hat{x} + t\hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}} \cdot \langle D(Jg_K(\hat{x} + t\hat{n}_x) \hat{n}_x)(\hat{n}_x), n' \rangle_{\mathbb{X}} \\ &\quad + \tilde{t} \langle Jg_K(\hat{x} + t\hat{n}_x) \hat{n}_x, n' \rangle_{\mathbb{X}} \cdot \langle Jg_K(\hat{x} + t\hat{n}_x) \hat{n}_x, n' \rangle_{\mathbb{X}} \\ &\quad + \tilde{t}^2 \langle Jg_K(\hat{x} + t\hat{n}_x) \hat{n}_x, n' \rangle_{\mathbb{X}} \cdot \langle D(Jg_K(\hat{x} + t\hat{n}_x) \hat{n}_x)(\hat{n}_x), n' \rangle_{\mathbb{X}} \\ &\quad + \mathcal{O}(\|\tilde{t}\|),\end{aligned}$$

Nous en déduisons que

$$\begin{aligned}D\tilde{h}(t)(\tilde{t}) &= \tilde{t} \langle g_K(\hat{x} + t\hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}} \cdot \langle D(Jg_K(\hat{x} + t\hat{n}_x) \hat{n}_x)(\hat{n}_x), n' \rangle_{\mathbb{X}} \\ &\quad + \tilde{t} \langle Jg_K(\hat{x} + t\hat{n}_x) \hat{n}_x, n' \rangle_{\mathbb{X}} \cdot \langle Jg_K(\hat{x} + t\hat{n}_x) \hat{n}_x, n' \rangle_{\mathbb{X}}\end{aligned}\tag{C.23}$$

et la dérivée de \tilde{h} :

$$\tilde{h}'(t) = \langle Jg_K(\hat{x} + t\hat{n}_x) \hat{n}_x, n' \rangle_{\mathbb{X}}^2 + \langle g_K(\hat{x} + t\hat{n}_x) - g_K(\hat{a}), n' \rangle_{\mathbb{X}} \cdot \langle D(Jg_K(\hat{x} + t\hat{n}_x) \hat{n}_x)(\hat{n}_x), n' \rangle_{\mathbb{X}}.\tag{C.24}$$

Nous devons maintenant préciser le terme $D(Jg_K(\hat{x} + t\hat{n}_x) \hat{n}_x)(\hat{n}_x)$. Pour cela, nous posons la fonction \mathcal{M} définie par

$$\mathcal{M} \left\{ \begin{array}{l} \widehat{K} \longrightarrow \mathbb{X} \\ \hat{y} \longmapsto Jg_K(\hat{y}) \hat{n}_x \end{array} \right.\tag{C.25}$$

que l'on peut écrire comme

$$\mathcal{M}(\hat{y}) = \begin{pmatrix} \partial_1 g_1(\hat{y}) & \partial_2 g_1(\hat{y}) \\ \partial_1 g_2(\hat{y}) & \partial_2 g_2(\hat{y}) \\ \partial_1 g_3(\hat{y}) & \partial_2 g_3(\hat{y}) \end{pmatrix} \begin{pmatrix} \hat{n}_1 \\ \hat{n}_2 \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^2 \partial_k g_1(\hat{y}) \cdot \hat{n}_k \\ \sum_{k=1}^2 \partial_k g_2(\hat{y}) \cdot \hat{n}_k \\ \sum_{k=1}^2 \partial_k g_3(\hat{y}) \cdot \hat{n}_k \end{pmatrix}\tag{C.26}$$

où on note g_i la i -ème composante de g_K , pour $i = 1, 2, 3$ afin de simplifier les notations. On s'intéresse aux différentielles :

$$D(Jg_K(\hat{x} + t\hat{n}_x) \hat{n}_x)(\hat{n}) = D\mathcal{M}(\hat{x} + t\hat{n}_x)(\hat{n}_x).\tag{C.27}$$

Or,

$$D\mathcal{M}(\hat{y})(\hat{z}) = J\mathcal{M}(\hat{y})\hat{z} \quad \forall \hat{z} \in \widehat{K},\tag{C.28}$$

où $J\mathcal{M}$ désigne la jacobienne de \mathcal{M} qui est donc définie par

$$J\mathcal{M}(\hat{y}) = \begin{pmatrix} \partial_1 \mathcal{M}_1(\hat{y}) & \partial_2 \mathcal{M}_1(\hat{y}) \\ \partial_1 \mathcal{M}_2(\hat{y}) & \partial_2 \mathcal{M}_2(\hat{y}) \\ \partial_1 \mathcal{M}_3(\hat{y}) & \partial_2 \mathcal{M}_3(\hat{y}) \end{pmatrix},\tag{C.29}$$

que l'on peut écrire de manière abrégée comme

$$J\mathcal{M}(\hat{y})_{ij} = \frac{\partial \mathcal{M}_i}{\partial \hat{x}_j}(\hat{y}), \quad i = 1, \dots, 3, j = 1, \dots, 2.\tag{C.30}$$

En utilisant (C.26), on a

$$J\mathcal{M}(\hat{y})_{ij} = \sum_{k=1}^2 \frac{\partial^2 g_i(\hat{y})}{\partial \hat{x}_k \partial \hat{x}_j} \hat{n}_k. \quad (\text{C.31})$$

En combinant maintenant (C.27), (C.28) et (C.31) on en déduit que

$$D(Jg_K(\hat{x} + t\hat{n}_x)\hat{n}_x)(\hat{n}_x) = J\mathcal{M}(\hat{x} + t\hat{n}_x)\hat{n}_x = \left(\sum_{j=1}^2 \sum_{k=1}^2 \frac{\partial^2 g_i(\hat{x} + t\hat{n}_x)}{\partial \hat{x}_k \partial \hat{x}_j} \hat{n}_k \hat{n}_j \right)_{i=1, \dots, 3}. \quad (\text{C.32})$$

Table des figures

| | | |
|------|---|----|
| 1.1 | La chaîne classique de calcul | 11 |
| 1.2 | Exemple de représentation d’une fonction affine sur un triangle avec ParaView 3.14. | 13 |
| 1.3 | Les 4 cas où une fonction est définie par les valeurs (0 ou 1) aux sommets sur le carré unité. | 14 |
| 1.4 | Représentation avec ParaView 3.14 sur le carré unité de différents cas où la fonction est définie par les valeurs aux sommets. | 14 |
| 1.5 | Deux représentations possibles du cas 1 de la figure 1.3. | 14 |
| 1.6 | Représentations obtenues des 4 cas en suivant la figure 1.5.a. | 15 |
| 1.7 | Représentations obtenues des 4 cas en suivant la figure 1.5.b. | 15 |
| 1.8 | Représentation d’une fonction Q^1 sur le carré unité. | 15 |
| 1.9 | Post-traitements dans ParaView 3.14 de la figure 1.4 où la fonction est définie par les valeurs aux sommets du carré unité. Des isovaleurs ainsi que des gradients (en rouge) sont affichés. | 16 |
| 1.10 | Comparaison des représentations et isovaleurs obtenues pour le cas 1 sur le carré unité et deux configurations de triangles. | 16 |
| 1.11 | Représentation avec Gmsh 2.4 sur le carré unité de différents cas où la fonction est définie par les valeurs aux sommets. | 17 |
| 1.12 | Représentation sous ParaView d’un polynôme d’ordre élevé sur différentes grilles de visualisation. | 19 |
| 1.13 | Illustration du <i>Ray Tracing</i> . Image extraite de [Levoy, 1990]. | 20 |
| 1.14 | Positionnement dans la chaîne de calcul | 21 |
| 1.15 | Notations pour l’estimation des erreurs dans [Leone et al., 1998] | 22 |
| 1.16 | Notations pour définir le nombre de subdivisions géométriques. | 23 |
| 1.17 | Construction des maillages de représentation dans [Nikishkov, 2003] | 24 |
| 1.18 | Estimation des erreurs dans [Schroeder et al., 2006] | 25 |
| 1.19 | Exemple de représentation obtenue dans Gmsh pour une fonction d’ordre 3. | 26 |
| 2.1 | Description des notations utilisées. | 35 |
| 2.2 | Description des notations utilisées. | 37 |
| 2.3 | Approximation de la fonction cube (en bleu) sur le segment $[-1,1]$ par des représentations affines par morceaux (en rouge). La distance de Hausdorff (en pointillé) augmente dans le cas où le maillage de représentation est raffiné (droite). | 40 |
| 2.4 | Représentation sur deux éléments de fonctions continues (a et b) et d’une fonction discontinue (c) sur un maillage de représentation non-coïncident (a) et des maillages coïncidents (b et c). Les vues de dessus sont également affichées (d, e et f). | 41 |
| 2.5 | Illustration de la décomposition des éléments en dimensions inférieures | 42 |

| | | |
|------|---|----|
| 2.6 | Illustration du processus en dimension 2 : décomposition des éléments en dimensions inférieures (b), maillage et représentation des arêtes (c et d) et des éléments (e et f). | 43 |
| 2.7 | Construction du maillage de représentation de l'élément de référence $\mathcal{T}(\widehat{K})$ (gauche) et du maillage de représentation $\mathcal{T}(K)$ (droite) pour un élément $K \in \mathcal{T}_h$ (en pointillé). | 46 |
| 2.8 | Principe général, maillage par dimensions croissantes | 51 |
| 2.9 | Principe de la génération du maillage de représentation à une dimension donnée | 51 |
| 2.10 | Premières étapes de l'algorithme DIRECT sur un triangle | 53 |
| 2.11 | Un tétraèdre est découpé régulièrement en 8 tétraèdres | 53 |
| 3.1 | Représentation (c et d) d'une fonction (a et b) sur l'élément de référence (a et c) et l'élément réel (b et d). | 60 |
| 3.2 | Visualisation d'une fonction numérique f_{num}^K sur un élément K et sa représentation $f_{vis}^{\widetilde{K}}$ construite sur un élément \widetilde{K} . L'approximation \mathbb{P}^1 de la transformation géométrique n'est pas bijective de \widetilde{K} vers \widehat{K} | 61 |
| 3.3 | Étapes de construction et mise en évidence des recouvrements lors du traitement d'un élément courbe. | 62 |
| 3.4 | Étapes de construction et <i>by-pass</i> de recouvrements par l'ajout d'un point sur le bord de l'élément. | 63 |
| 3.5 | Comparaison de ∂K (frontière de l'élément réel), de $P^1 g_{\mathcal{T}(K)}(\partial \widehat{K})$ (image par l'approximation de la transformation géométrique de la frontière de la cellule de référence) et $\partial \widetilde{K}$ (frontière de l'approximation de l'élément réel) de la figure 3.2. | 64 |
| 3.6 | Ajout d'un sommet extérieur (en rouge) dans une configuration initiale (a et d) pour un élément courbe (frontière réelle en bleu). | 65 |
| 3.7 | Configurations possibles lors de l'ajout d'un nouveau sommet au maillage. | 66 |
| 3.8 | Notations utilisées lors du découpage d'un simplexe en 3 simplexes. | 66 |
| 3.9 | Identification des zones de retournements sur l'élément de référence (gauche) et l'élément réel (droite). | 67 |
| 3.10 | Illustration des zones de retournements et des notations utilisées sur deux cas. | 69 |
| 3.11 | Les vecteurs \hat{n}_x sont construits comme une combinaison affine entre \hat{n}_a et \hat{n}_b | 69 |
| 3.12 | Illustration des zones $\widehat{\mathcal{I}}_S^K$ sur deux cas | 73 |
| 3.13 | Illustration de la recherche du paramètre d'avancement maximal. | 79 |
| 3.14 | Triangle initial exploré par DIRECT (a), et découpages jusqu'au niveau 1 (b) et 2 (c). | 81 |
| 3.15 | Points testés par DIRECT sur le triangle isocèle (a), sur la cellule de référence (b) et sur l'élément réel (c). | 82 |
| 3.16 | Découpage de triangles isocèles en triangles isocèles | 83 |
| 3.17 | Relations entre les abscisses ω et les indices k pour les premiers niveaux de DIRECT | 86 |
| 4.1 | Représentations de fonctions de base de Gauss-Lobatto d'ordre 3 sur un carré (précision 1%). | 93 |
| 4.2 | Une fonction de base de type Gauss d'ordre 3 sur un élément carré. | 95 |
| 4.3 | Représentation d'une fonction de Gauss Q^3 avec la méthode adaptative (gauche), avec un découpage uniforme pour un nombre similaire d'éléments (milieu) et avec un découpage uniforme pour une précision similaire (droite). Le maillage de représentation est affiché en blanc sur la deuxième ligne. | 96 |

| | | |
|------|---|-----|
| 4.4 | Zoom près de l'origine de la figure 4.3 pour la représentation d'une fonction de Gauss Q^3 avec la méthode adaptative (gauche), avec un découpage uniforme pour un nombre similaire d'éléments (milieu) et avec un découpage uniforme pour une précision similaire (droite). | 96 |
| 4.5 | Cartographie de l'erreur (voir figure 4.3) pour la représentation d'une fonction de Gauss Q^3 avec la méthode adaptative (gauche), avec un découpage uniforme pour un nombre similaire d'éléments (milieu) et avec un découpage uniforme pour une précision similaire (droite). | 97 |
| 4.6 | Cartographie des erreurs de visualisation pour la fonction de gauss Q^3 entre les méthodes adaptatives et uniformes pour un nombre similaire d'éléments avec la même <i>colorbar</i> (celle du cas uniforme). | 97 |
| 4.7 | Comparaison de la méthode adaptative avec un découpage uniforme pour une fonction de base Gauss d'ordre 3. | 98 |
| 4.8 | Propagation d'une onde élastique dans un tri-milieu calculée par une méthode Galerkin Discontinue Q^7 sur un maillage uniforme 24×24 . Représentation à 0.5 % | 99 |
| 4.9 | Propagation d'une onde élastique dans un tri-milieu calculée par une méthode Galerkin Discontinue Q^7 sur un maillage uniforme 24×24 | 101 |
| 4.10 | Représentation d'un mode plan donné par (4.1) sur des hexaèdres avec notre méthode de visualisation à 1%. | 103 |
| 4.11 | Représentation d'un mode donné par (4.2) sur des hexaèdres avec notre méthode de visualisation à 0.5%. | 104 |
| 4.12 | Une fonction de base de type Gauss-Lobatto d'ordre 6 sur un quadrangle Q^2 . . . | 105 |
| 4.13 | Représentation à une précision de 3.5 % avec les deux indicateurs Δ_S^K (a et c) and Γ_S^K (b et d) d'une fonction de base Gauss-Lobatto Q^6 sur un quadrangle Q^2 . . . | 106 |
| 4.14 | Comparaison de la méthode adaptative avec un découpage uniforme pour une fonction de base Gauss d'ordre 6 sur un quadrangle Q^2 | 107 |
| 4.15 | Représentation à une précision de 7 % avec les deux indicateurs Δ_S^K (a et c) and Γ_S^K (b et d) d'une fonction de base Gauss-Lobatto Q^6 sur un quadrangle Q^2 dans l'espace. | 108 |
| 4.16 | Représentation de la solution de propagation d'une onde acoustique à un temps T . | 110 |
| 4.17 | Représentation de la solution de propagation d'une onde acoustique. Zoom de la figure 4.16. | 111 |
| 4.18 | Représentation d'une fonction Gauss-Lobatto à 0.1 % avec prise en compte uniquement du bord lors du maillage des arêtes. Des éléments plats apparaissent près du bord. | 112 |
| 4.19 | Exemples de zones de voisinage en dimension 2. | 113 |
| 4.20 | Représentation d'une fonction Gauss-Lobatto à 0.1 % avec prise en compte uniquement du bord lors du maillage des arêtes. | 114 |
| 4.21 | Représentation d'une fonction Gauss-Lobatto à 0.1 % avec prise en compte du voisinage lors du maillage des arêtes. | 114 |
| 4.22 | Représentation d'une fonction Gauss-Lobatto à 1 % avec prise en compte uniquement du bord lors du maillage des arêtes. | 114 |
| 4.23 | Représentation d'une fonction Gauss-Lobatto à 1 % avec prise en compte du voisinage lors du maillage des arêtes. | 115 |
| 5.1 | Un rayonnement en utilisant une grille d'interpolation | 121 |
| 5.2 | Champs rayonnés obtenus avec différentes grilles de visualisation (interpolation P^1). | 122 |

| | | |
|------|---|-----|
| 5.3 | Représentation de plusieurs champs rayonnés avec des fonctions de bas ordre (a et c) et des fonctions d'ordre élevé (b et d) pour une erreur d'interpolation L^∞ sous 1%. | 123 |
| 5.4 | Visualisation du noyau de Green G (a), d'une onde plane P (b) et de la fonction G_c (c). | 125 |
| 5.5 | Introduction d'une direction pour effectuer une interpolation directionnelle . . . | 126 |
| 5.6 | Un exemple de découpage de la frontière Γ | 128 |
| 5.7 | Notations et problématique | 129 |
| 5.8 | Illustration des subdivisions de la frontière Γ associées aux élément \mathcal{E}_V de \mathcal{V} | 130 |
| 5.9 | Schéma général du rayonnement | 131 |
| 5.10 | Les 4 configurations de subdivisions en dimension deux | 134 |
| 5.11 | Les 9 configurations de subdivisions en dimension trois | 134 |
| 5.12 | Illustration des premières étapes de subdivisions (le couple à subdiviser est mis en évidence en orange). | 135 |
| 5.13 | Exemples de subdivisions du domaine à visualiser \mathcal{V} (a) et de la frontière \mathcal{G} (b). . | 136 |
| 5.14 | Règles pour fixer l'ordre pour une précision donnée selon le ratio longueur / longueur d'onde à 1 % | 137 |
| 5.15 | Interpolation directionnelle avec une seule direction : configuration. | 138 |
| 5.16 | Diffusions de champs diffractés (a, b et c) par un disque. Hormis la référence (a), les autres résultats sont obtenus avec des fonctions d'ordre 5 dans chaque direction. Comparaison de l'ajout d'une seule direction (c et e) au cas sans direction (b et d). | 140 |
| 5.17 | Diffraction autour d'un disque : configuration. | 141 |
| 5.18 | Subdivision du domaine et comparaison des ordres nécessaires. | 142 |
| 5.19 | Visualisation de champs diffractés. Comparaison de notre méthode avec le cas sans direction pour le même nombre d'éléments (a-b) et pour une précision similaire (a-c). | 142 |
| 5.20 | Diffraction autour d'un disque | 143 |
| 5.21 | Propagation d'une onde acoustique autour d'un disque pour un nombre d'onde élevé | 144 |
| 5.22 | Diffraction autour d'un objet <i>L-shape</i> | 145 |
| 5.23 | Subdivision du domaine et ordres nécessaires par direction pour la configuration en forme de L. | 145 |
| 5.24 | Visualisation du champ rayonné total (parties réelle (a) et absolue (b)) diffracté par un objet en forme de L pour une onde plane incidente donnée (angle 30 degrés). 146 | |
| 5.25 | Visualisation du champ rayonné total (parties réelle (a) et absolue (b)) diffracté par un objet en forme de L pour une seconde onde plane incidente (angle 120 degrés). | 147 |
| 5.26 | Subdivision du domaine et ordres nécessaires par direction pour la configuration en forme de L pour $k = 100$ | 147 |
| 5.27 | Visualisation du champ rayonné total (parties réelle (a) et absolue (b)) diffracté par un objet en forme de L pour une onde plane incidente à $k = 100$ | 148 |
| 5.28 | Solution calculée par la méthode des éléments finis de frontière. | 149 |
| 5.29 | Propagation d'une onde autour d'un rafale. Visualisation des champs rayonné dans des plans. Subdivision (en noir) des plans en question et maillages de représentation. | 150 |
| 5.30 | Visualisation des champs rayonné dans des plans. Propagation d'une onde autour d'un rafale. | 151 |

| | | |
|------|--|-----|
| 5.31 | Ordres suivant chaque direction pour le rayonnement 3D en dehors de l'obstacle diffractant | 151 |
| 5.32 | Visualisation du champ rayonné reconstruit. La subdivision du domaine est affichée en noir (308 éléments), le maillage de représentation se trouve en <i>wireframe</i> (7 985 584 tétraèdres). | 152 |
| 5.33 | Comparaison de la visualisation obtenue à 1 % avec la méthode adaptative (gauche) et sans direction (droite). | 152 |
| 5.34 | Subdivision du domaine en 844 éléments. | 153 |
| 5.35 | Ordres suivant chaque direction pour le rayonnement 3D autour de l'obstacle. | 153 |
| 5.36 | Coupe de la figure 5.35 permettant de voir les ordres près de l'obstacle. | 154 |
| 5.37 | Comparaison pour les mêmes degrés de liberté de la visualisation obtenue avec directions (gauche) et sans directions (droite). | 154 |
| 5.38 | Représentation des champs rayonnés autour d'un rafale avec nos méthodes de reconstruction et de visualisation. | 156 |
| A.1 | Exemples de maillages structuré (gauche) et non-structuré (droite). | 166 |
| A.2 | Exemples de maillages conforme (gauche) et non-conforme (droite). | 167 |
| B.1 | Localisation de points de Gauss, de Gauss-Lobatto et de Chebyshev sur le segment de référence. | 175 |
| B.2 | Transformation géométrique entre le segment unité et un segment quelconque. | 176 |
| B.3 | Transformation géométrique entre la cellule de référence $[-1,1]$ et un segment quelconque. | 177 |
| B.4 | Transformation géométrique entre le triangle unité et un triangle P^1 quelconque. | 178 |
| B.5 | Transformation géométrique entre le triangle de référence et un triangle P^2 quelconque. | 178 |
| B.6 | Transformation géométrique entre la cellule de référence et un quadrangle Q^1 quelconque. | 179 |
| B.7 | Transformation géométrique entre le carré unité et un quadrangle Q^2 quelconque. | 180 |
| B.8 | Numérotations des sommets pour l'hexaèdre et le tétraèdre. | 181 |
| B.9 | Fonctions de Raviart-Thomas d'ordre 0. | 183 |

Bibliographie

- [Abramowitz and Stegun, 1964] Abramowitz, M. and Stegun, I. A. (1964). *Handbook of mathematical functions : with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation.
- [Alauzet and Frey, 2003] Alauzet, F. and Frey, P. (2003). Estimateur d’erreur géométrique et métriques anisotropes pour l’adaptation de maillage. partie i : aspects théoriques.
- [Arnold et al., 2002] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D. (2002). Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM journal on numerical analysis*, 39(5) :1749–1779.
- [Ayachit, 2015] Ayachit, U. (2015). The paraview guide : a parallel visualization application.
- [Bakry, 2016] Bakry, M. (2016). *Fiabilité et optimisation des calculs obtenus par des formules intégrales en propagation d’onde*. PhD thesis, Université Paris-Saclay.
- [Bebendorf et al., 2015] Bebendorf, M., Kuske, C., and Venn, R. (2015). Wideband nested cross approximation for helmholtz problems. *Numerische Mathematik*, 130(1) :1–34.
- [Bériot et al., 2016] Bériot, H., Prinn, A., and Gabard, G. (2016). Efficient implementation of high-order finite elements for helmholtz problems. *International Journal for Numerical Methods in Engineering*, 106(3) :213–240.
- [Bertsekas, 2014] Bertsekas, D. P. (2014). *Constrained optimization and Lagrange multiplier methods*. Academic press.
- [Börm, 2015] Börm, S. (2015). Directional h2-matrix compression for high-frequency problems. *arXiv preprint arXiv :1510.07087*.
- [Börm and Melenk, 2017] Börm, S. and Melenk, J. M. (2017). Approximation of the high-frequency helmholtz kernel by nested directional interpolation : error analysis. *Numerische Mathematik*, pages 1–34.
- [Bowyer, 1981] Bowyer, A. (1981). Computing Dirichlet tessellations. *The computer journal*, 24(2) :162–166.
- [Brandt, 1991] Brandt, A. (1991). Multilevel computations of integral transforms and particle interactions with oscillatory kernels. *Computer Physics Communications*, 65(1-3) :24–38.
- [Cheng et al., 1988] Cheng, J., Finnigan, P., Hathaway, A., Kela, A., and Schroeder, W. (1988). Quadtree/octree meshing with adaptive analysis. *Numerical grid generation in computational fluid mechanics’88*, pages 633–642.
- [Childs et al., 2012] Childs, H., Brugger, E., Whitlock, B., Meredith, J., Ahern, S., Pugmire, D., Biagas, K., Miller, M., Harrison, C., Weber, G. H., Krishnan, H., Fogal, T., Sanderson, A., Garth, C., Bethel, E. W., Camp, D., Rübel, O., Durant, M., Favre, J. M., and Navrátil, P. (2012). VisIt : An End-User Tool For Visualizing and Analyzing Very Large Data. In *High Performance Visualization—Enabling Extreme-Scale Scientific Insight*, pages 357–372.

- [Chung and Field, 2000] Chung, A. J. and Field, A. J. (2000). A simple recursive tessellator for adaptive surface triangulation. *Journal of graphics tools*, 5(3) :1–9.
- [Ciarlet, 1978] Ciarlet, P. G. (1978). *The finite element method for elliptic problems*, volume 4. Classics in applied mathematics.
- [Clausen and Žilinskas, 2002] Clausen, J. and Žilinskas, A. (2002). Subdivision, sampling, and initialization strategies for simplicial branch and bound in global optimization. *Computers & Mathematics with Applications*, 44(7) :943–955.
- [Cockburn et al., 2000a] Cockburn, B., Karniadakis, G. E., and Shu, C.-W. (2000a). The development of discontinuous galerkin methods. In *Discontinuous Galerkin Methods*, pages 3–50. Springer.
- [Cockburn et al., 2000b] Cockburn, B., Luskin, M., Shu, C.-W., and Süli, E. (2000b). Post-processing of galerkin methods for hyperbolic problems. In *Discontinuous Galerkin Methods*, pages 291–300. Springer.
- [Cohen and Pernet, 2016] Cohen, G. and Pernet, S. (2016). *Finite element and discontinuous Galerkin methods for transient wave equations*. Springer.
- [Colton and Kress, 2012] Colton, D. and Kress, R. (2012). *Inverse acoustic and electromagnetic scattering theory*, volume 93. Springer Science & Business Media.
- [Costabel, 1986] Costabel, M. (1986). *Principles of boundary element methods*. Techn. Hochsch., Fachbereich Mathematik.
- [Delaunay, 1934] Delaunay, B. (1934). Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800) :1–2.
- [Dobrzynski, 2005] Dobrzynski, C. (2005). *Adaptation de maillage anisotrope 3D et application à l'aéro-thermique des bâtiments*. PhD thesis, Université Pierre et Marie Curie-Paris VI.
- [Dudouit, 2014] Dudouit, Y. (2014). *Spatio-temporal refinement using a discontinuous Galerkin approach for elastodynamic in a high performance computing framework*. PhD thesis, Bordeaux.
- [D’Azevedo, 1991] D’Azevedo, E. F. (1991). Optimal triangular mesh generation by coordinate transformation. *SIAM Journal on scientific and statistical computing*, 12(4) :755–786.
- [Engquist and Ying, 2007] Engquist, B. and Ying, L. (2007). Fast directional multilevel algorithms for oscillatory kernels. *SIAM Journal on Scientific Computing*, 29(4) :1710–1737.
- [Ervin, 2012] Ervin, V. (2012). Computational bases for rtk and bdmk on triangles. *Computers & Mathematics with Applications*, 64(8) :2765–2774.
- [Forsythe and Wasow, 1960] Forsythe, G. and Wasow, W. R. (1960). *Finite-Difference Methods for Partial Differential Equations, Applied Mathematical Series*. Wiley, New York.
- [Frey and George, 2008] Frey, P. J. and George, P. L. (2008). *Mesh generation : application to finite elements*. Wiley Online Library.
- [George, 1971] George, J. A. (1971). Computer implementation of the finite element method. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE.
- [George and Borouchaki, 2012] George, P. and Borouchaki, H. (2012). Sur les éléments finis quadrilatéraux de degré 1 et 2 (version 2). *RR INRIA*, 7964.
- [George, 1991] George, P.-L. (1991). Génération de maillages par une méthode de type voronoï.
- [George, 1992] George, P. L. (1992). *Automatic mesh generation : applications to finite element methods*. John Wiley & Sons, Inc.

- [George et al., 1991] George, P. L., Hecht, F., and Saltel, É. (1991). Automatic mesh generator with specified boundary. *Computer methods in applied mechanics and engineering*, 92(3) :269–288.
- [George and Seveno, 1994] George, P. L. and Seveno, E. (1994). The advancing-front mesh generation method revisited. *International Journal for Numerical Methods in Engineering*, 37(21) :3605–3619.
- [Geuzaine and Remacle, 2008] Geuzaine, C. and Remacle, J. (2008). Gmsh reference manual : the documentation for gmsh, a finite element mesh generator with built-in pre-and post-processing facilities. URL <http://www.geuz.org/gmsh>.
- [Geuzaine and Remacle, 2009] Geuzaine, C. and Remacle, J.-F. (2009). Gmsh : A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11) :1309–1331.
- [Glassner, 1989] Glassner, A. S. (1989). *An introduction to ray tracing*. Elsevier.
- [Hughes et al., 2005] Hughes, T. J., Cottrell, J. A., and Bazilevs, Y. (2005). Isogeometric analysis : Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*, 194(39-41) :4135–4195.
- [Jones et al., 1993] Jones, D. R., Perttunen, C. D., and Stuckman, B. E. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of optimization Theory and Applications*, 79(1) :157–181.
- [Leone et al., 1998] Leone, A. O., Marzano, P., Gobetti, E., Scateni, R., and Pedinotti, S. (1998). Discontinuous finite element visualization. In *Proceedings 8th International Symposium on Flow Visualization*.
- [LeVeque, 2002] LeVeque, R. J. (2002). *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press.
- [Levoy, 1988] Levoy, M. (1988). Display of surfaces from volume data. *IEEE Computer graphics and Applications*, (3) :29–30.
- [Levoy, 1990] Levoy, M. (1990). Efficient ray tracing of volume data. *ACM Transactions on Graphics (TOG)*, 9(3) :245–261.
- [Lo, 1985] Lo, S. (1985). A new mesh generation scheme for arbitrary planar domains. *International Journal for Numerical Methods in Engineering*, 21(8) :1403–1426.
- [Lo, 1991a] Lo, S. (1991a). Volume discretization into tetrahedra—i. verification and orientation of boundary surfaces. *Computers & Structures*, 39(5) :493–500.
- [Lo, 1991b] Lo, S. H. (1991b). Volume discretization into tetrahedra—ii. 3d triangulation by advancing front approach. *Computers & Structures*, 39(5) :501–511.
- [Löhner, 1996] Löhner, R. (1996). Progress in grid generation via the advancing front technique. *Engineering with computers*, 12(3-4) :186–210.
- [Löhner, 1997] Löhner, R. (1997). Automatic unstructured grid generators. *Finite Elements in Analysis and Design*, 25(1-2) :111–134.
- [Löhner and Parikh, 1988] Löhner, R. and Parikh, P. (1988). Generation of three-dimensional unstructured grids by the advancing-front method. *International Journal for Numerical Methods in Fluids*, 8(10) :1135–1149.
- [Loseille and Feuillet, 2018] Loseille, A. and Feuillet, R. (2018). Vizir : High-order mesh and solution visualization using opengl 4.0 graphic pipeline. In *2018 AIAA Aerospace Sciences Meeting*, page 1174.

- [Maunoury et al., 2018a] Maunoury, M., Besse, C., Mouysset, V., and Pernet, S. (2018a). Accurate, automatic and compressed visualization of radiated helmholtz fields from boundary element solutions. In *International Conference on Computational Science*, pages 708–721. Springer.
- [Maunoury et al., 2018b] Maunoury, M., Besse, C., Mouysset, V., Pernet, S., and Haas, P.-A. (2018b). Well-suited and adaptive post-processing for the visualization of hp simulation results. *Journal of Computational Physics*, 375 :1179 – 1204.
- [Messner et al., 2012] Messner, M., Schanz, M., and Darve, E. (2012). Fast directional multilevel summation for oscillatory kernels based on chebyshev interpolation. *Journal of Computational Physics*, 231(4) :1175–1196.
- [Nédélec, 1980] Nédélec, J.-C. (1980). Mixed finite elements in \mathbb{R}^3 . *Numerische Mathematik*, 35(3) :315–341.
- [Nédélec, 2001] Nédélec, J.-C. (2001). *Acoustic and electromagnetic equations : integral representations for harmonic problems*, volume 144. Springer Science & Business Media.
- [Nelson et al., 2012] Nelson, B., Liu, E., Kirby, R. M., and Haines, R. (2012). ElVis : A system for the accurate and interactive visualization of high-order finite element solutions. *IEEE transactions on visualization and computer graphics*, 18(12) :2325–2334.
- [Nelson, 2012] Nelson, B. W. (2012). *Accurate and interactive visualization of high-order finite element fields*. The University of Utah.
- [Nikishkov, 2003] Nikishkov, G. (2003). Generating contours on fem/bem higher-order surfaces using java 3d textures. *Advances in Engineering Software*, 34(8) :469–476.
- [Nikishkov, 2011] Nikishkov, G. (2011). Visualization of results in higher-order finite elements.
- [Owen, 1998] Owen, S. J. (1998). A survey of unstructured mesh generation technology. In *IMR*, pages 239–267.
- [Peiro et al., 2015] Peiro, J., Moxey, D., Jordi, B., Sherwin, S., Nelson, B., Kirby, R., and Haines, R. (2015). High-order visualization with elvis. In *IDIHOM : Industrialization of High-Order Methods-A Top-Down Approach*, pages 521–534. Springer.
- [Quattrochi, 2006] Quattrochi, D. J. (2006). *Hypersonic heat transfer and anisotropic visualization with a higher order discontinuous galerkin finite element method*. PhD thesis, Massachusetts Institute of Technology.
- [Raviart and Thomas, 1977] Raviart, P.-A. and Thomas, J.-M. (1977). A mixed finite element method for 2-nd order elliptic problems. In *Mathematical aspects of finite element methods*, pages 292–315. Springer.
- [Ray et al., 1999] Ray, H., Pfister, H., Silver, D., and Cook, T. A. (1999). Ray casting architectures for volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(3) :210–223.
- [Rebay, 1993] Rebay, S. (1993). Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm. *Journal of computational physics*, 106(1) :125–138.
- [Reed and Hill, 1973] Reed, W. H. and Hill, T. (1973). Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Lab., N. Mex.(USA).
- [Remacle et al., 2007] Remacle, J.-F., Chevaugéon, N., Marchandise, E., and Geuzaine, C. (2007). Efficient visualization of high-order finite elements. *International Journal for Numerical Methods in Engineering*, 69(4) :750–771.

- [Remacle et al., 2010] Remacle, J.-F., Geuzaine, C., Compère, G., and Helenbrook, B. (2010). Adaptive mesh generation and visualization. *Encyclopedia of Aerospace Engineering*.
- [Richtmyer and Morton, 1994] Richtmyer, R. D. and Morton, K. W. (1994). Difference methods for initial-value problems. *Malabar, Fla. : Krieger Publishing Co., c1994, 2nd ed.*
- [Ruprecht and Müller, 1998] Ruprecht, D. and Müller, H. (1998). A scheme for edge-based adaptive tetrahedron subdivision. In *Mathematical Visualization*, pages 61–70. Springer.
- [Schöberl, 1997] Schöberl, J. (1997). Netgen an advancing front 2d/3d-mesh generator based on abstract rules. *Computing and visualization in science*, 1(1) :41–52.
- [Schroeder et al., 2006] Schroeder, W. J., Bertel, F., Malaterre, M., Thompson, D., Pebay, P. P., O’Bara, R., and Tendulkar, S. (2006). Methods and framework for visualizing higher-order finite elements. *IEEE Transactions on Visualization and Computer Graphics*, 12(4) :446–460.
- [Schroeder et al., 2005] Schroeder, W. J., Bertel, F., Malaterre, M., Thompson, D., Pebay, P. P., O’Barall, R., and Tendulkar, S. (2005). Framework for visualizing higher-order basis functions. In *Visualization, 2005. VIS 05. IEEE*, pages 43–50. IEEE.
- [Schroeder et al., 2004] Schroeder, W. J., Lorensen, B., and Martin, K. (2004). *The visualization toolkit : an object-oriented approach to 3D graphics*. Kitware.
- [Smith, 1985] Smith, G. D. (1985). *Numerical solution of partial differential equations : finite difference methods*. Oxford university press.
- [Tzoulis and Eibert, 2006] Tzoulis, A. and Eibert, T. (2006). Fast computation of electromagnetic near-fields with the multilevel fast multipole method combining near-field and far-field translations. *Advances in Radio Science : ARS*, 4 :111.
- [Üffinger et al., 2010] Üffinger, M., Frey, S., and Ertl, T. (2010). Interactive high-quality visualization of higher-order finite elements. In *Computer Graphics Forum*, volume 29, pages 337–346. Wiley Online Library.
- [Walfisch, 2007] Walfisch, D. (2007). *Visualization for high-order discontinuous Galerkin CFD results*. PhD thesis, Massachusetts Institute of Technology.
- [Watson, 1981] Watson, D. F. (1981). Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The computer journal*, 24(2) :167–172.
- [Weiler et al., 2003] Weiler, M., Kraus, M., Merz, M., and Ertl, T. (2003). Hardware-based ray casting for tetrahedral meshes. In *Visualization, 2003. VIS 2003. IEEE*, pages 333–340. IEEE.
- [Wolff, 2011] Wolff, D. (2011). *OpenGL 4.0 shading language cookbook*. Packt Publishing Ltd.
- [Xu et al., 2017] Xu, L., Ren, X., Xu, X., Li, H., Tang, Y., and Feng, Y. (2017). An adaptive visualization tool for high order discontinuous galerkin method with quadratic elements. In *Computer and Information Technology (CIT), 2017 IEEE International Conference on*, pages 176–183. IEEE.
- [Yerry and Shephard, 1984] Yerry, M. A. and Shephard, M. S. (1984). Automatic three-dimensional mesh generation by the modified-octree technique. *International Journal for Numerical Methods in Engineering*, 20(11) :1965–1990.
- [Ypma, 1995] Ypma, T. J. (1995). Historical development of the newton–raphson method. *SIAM review*, 37(4) :531–551.
- [Zhou and Garland, 2006] Zhou, Y. and Garland, M. (2006). Interactive point-based rendering of higher-order tetrahedral data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5) :1229–1236.

- [Zhou et al., 2004] Zhou, Y., Garland, M., and Haber, R. (2004). Pixel-exact rendering of spacetime finite element solutions. In *Proceedings of the Conference on Visualization'04*, pages 425–432. IEEE Computer Society.
- [Zienkiewicz and Phillips, 1971] Zienkiewicz, O. and Phillips, D. (1971). An automatic mesh generation scheme for plane and curved surfaces by ‘isoparametric’ co-ordinates. *International Journal for Numerical Methods in Engineering*, 3(4) :519–528.
- [Zienkiewicz and Taylor, 1977] Zienkiewicz, O. C. and Taylor, R. L. (1977). *The finite element method*, volume 3. McGraw-hill London.
- [Žilinskas and Žilinskas, 2002] Žilinskas, A. and Žilinskas, J. (2002). Global optimization based on a statistical model and simplicial partitioning. *Computers & Mathematics with Applications*, 44(7) :957–967.

Résumé

Bien que les méthodes d'ordre élevé permettent de réaliser des simulations très précises et peu coûteuses, il y a un manque d'outils pour analyser et exploiter les résultats obtenus par ces nouveaux schémas. L'objectif de cette thèse est de mettre en place un cadre et des algorithmes efficaces pour visualiser des solutions calculées par des méthodes d'ordre élevé.

Notre approche est basée sur la construction d'une approximation affine optimisée de la solution numérique qui peut être post-traitée dans un logiciel de visualisation standard. Un maillage de représentation est créé via un indicateur d'erreurs *a posteriori* qui contrôle l'erreur de visualisation entre la solution numérique et sa représentation ponctuellement. Une stratégie est établie afin d'assurer que les (dis)continuités soient bien rendues. Un travail particulier est développé pour traiter les éléments d'ordre élevé (éléments courbes) et utilise notamment des indicateurs d'erreurs *a posteriori* spécifiques. Des exemples numériques montrent le potentiel de la méthode de visualisation.

Dans une seconde partie, nous nous intéressons au calcul et à la reconstruction de champs rayonnés pour des problèmes d'ondes en régime harmonique. Nous proposons une méthodologie pour générer une reconstruction précise de champs rayonnés tout en limitant le nombre d'informations nécessaires (i.e. en compressant les données). Pour ce faire, nous nous appuyons sur des fonctions de base composées de polynômes d'ordre élevé et d'ondes planes, ainsi que sur un développement du noyau de la formule intégrale servant au rayonnement. La méthode de visualisation permet alors de représenter fidèlement (décompression) les cartographies obtenues.

Mots clés : Visualisation, post-traitement, solutions numériques d'ordre élevé, champs rayonnés, reconstruction d'ordre élevé.

Abstract

While high order methods allow to perform very accurate simulations with low costs, there is a lack of tools to analyze and exploit results obtained by these new schemes. The objective of this thesis is to design a framework and efficient algorithms to visualize solutions computed by high order methods.

Our approach is based on the construction of an optimized affine approximation of the numerical solution which can be handled by any standard visualization software. A representation mesh is created via an *a posteriori* estimate which control visualization error between the numerical solution and its representation, and is performed pointwise. A strategy is established to ensure that (dis)continuities are well-rendered. A special work is done to treat high order elements (curved elements) and in particular use specific *a posteriori* estimates. Several numerical examples demonstrate the potential of the visualization method.

In a second part, we examine the computation and reconstruction of radiated fields for wave problems in harmonic regime. We propose a methodology to generate an accurate reconstruction of radiated fields while limiting the information needed (i.e. compressing the data). For this purpose, we rely on basis functions composed of high order polynomials and plane waves, as well as a development of the kernel used for the integral representation. The visualization method allows to faithfully represent (decompression process) the cartographies obtained.

Key words : Visualization, post-processing, high order numerical solutions, radiated fields, high order reconstruction.