

École polytechnique de Louvain

Estimateur d'erreur efficace a posteriori pour la résolution d'écoulements granulaires immergés

Auteur: **Margaux BOXHO**

Promoteurs: **Vincent LEGAT, Jonathan LAMBRECHTS**

Lecteurs: **Matthieu CONSTANT, Jean-François REMACLE**

Année académique 2018–2019

Master [120] : ingénieur civil en mathématiques appliquées

Abstract

For finite element methods, the mesh is a fundamental tool that will influence the whole computation : the computational time, the accuracy of the results and, the stability of the model. Getting an accurate numerical solution in a short period is the main challenge of many finite element codes of this century. One of the way we could tackle the problem is to develop an error estimator to be further used in an adaptive strategy.

To respond to this challenge, we implemented an adaptive algorithm based on a gradient recovery estimator over the velocity and pressure field. This estimator is easy to implement in existing codes because it does not require knowledge about the problem. It was applied to immersed granular flows and, we obtained valuable results for the test case of a falling cloud of grains in a highly viscous fluid. The mesh was refined around the falling cloud and, the queue formed behind it was also correctly captured.

To further enhance the accuracy of the solution, we developed a local mesh adaptation algorithm. The local adaptation reduces the numerical diffusion introduced by the interpolation of the solution over the new mesh. This adaptive technique is inspired by the work of Bank and Sherman.

Abstract

Pour les méthodes en éléments finis, le maillage est un outil fondamental qui va conditionner tout le reste du calcul : à la fois le temps de calcul, la précision des résultats mais aussi la stabilité du modèle. Obtenir une solution numérique précise en un temps acceptable est un enjeu majeur de tous les codes en éléments finis du vingt et unième siècle. Une manière de s'attaquer au problème est de développer un estimateur d'erreur qui puisse ensuite être utilisé dans une stratégie adaptative.

En vue de répondre à cet enjeu, nous avons implémenté un algorithme d'adaptation de maillage basé sur un estimateur par reconstruction de gradient du champ de vitesse et de pression. Cet estimateur est facile à implémenter dans des codes existants car il ne requiert aucune connaissance du problème traité. Il est appliqué aux problèmes des écoulements granulaires immergés. Il offre de très bons résultats sur le cas test d'un nuage de grains chutant dans un fluide visqueux. Le maillage ainsi obtenu est raffiné autour du nuage mais également autour de la colonne de grains se formant à l'arrière de celui-ci.

Dans le but d'améliorer encore la précision de la solution, nous avons développé dans le code MigFlow un algorithme d'adaptation locale du maillage. Cette adaptation locale nous permet de réduire les phénomènes de diffusion liés à l'interpolation de la solution sur le nouveau maillage. Cette technique d'adaptation est en partie basée sur les travaux de Bank et Sherman.

Remerciements

C'est dans le département de mécanique et de mathématiques appliquées à l'université catholique de Louvain-la-Neuve, que j'ai réalisé mon mémoire au sein de l'équipe MigFlow.

Tout d'abord, je tiens à remercier Vincent Legat pour m'avoir offert l'opportunité de travailler dans son équipe durant une année sur le sujet des écoulements granulaires immergés. Tout au long de mes études, il s'est toujours montré disponible, à l'écoute et de bons conseils.

Je remercie également Jonathan Lambrechts pour les nombreuses heures de débogage et sa patience. De la même manière, je remercie Matthieu Constant pour son enthousiasme et sa disponibilité tout au cours de cette année.

Je réserve mes derniers mots pour ma famille, à ma maman qui a pris le temps de relire mon mémoire et pour ses encouragements, à mon papa pour son soutien et ses questions toujours très pertinentes et à ma mamie pour ses nombreuses bonnes pensées.

Dans le domaine des idées, tout dépend de l'enthousiasme.

Dans le monde réel, tout repose sur la persévérance.

Johann Wolfgang von Goethe

CHAPITRE 1

INTRODUCTION

Depuis l'avènement du calcul par ordinateur et principalement des simulations par ordinateur de phénomènes physiques réels et complexes, le traitement des erreurs numériques est devenu un sujet de grand intérêt. Pour les besoins du projet Manhattan dont le but était de modéliser le processus de détonation nucléaire, la première simulation numérique voit le jour pendant la Seconde Guerre Mondiale. Avec elle, les erreurs numériques apparaissent inévitablement. En effet, le modèle continu est discrétisé pour être ensuite traité par un ordinateur. Ce dernier n'est pas capable de capturer toute l'information contenue dans les modèles physiques caractérisés par des équations différentielles partielles ou encore par des équations intégrales. Pour nombre de ces simulations numériques, la qualité et la précision d'une solution sont essentielles et même cruciales afin d'interpréter correctement les résultats obtenus. Pour assurer un certain niveau de précision, la maîtrise de l'erreur numérique s'est avérée nécessaire. Comment est-il possible de mesurer ces erreurs numériques ? Pouvons-nous éventuellement les approximer ou tout du moins fournir une borne supérieure sur l'erreur commise ? Est-il possible de contrôler ou de minimiser ces erreurs ? Ce n'est qu'au cours des années 90, soit 50 ans plus tard, que de nombreuses avancées ont été réalisées dans le domaine. Ces questions ont pu être résolues sous la forme de méthodes et de théorèmes à propos des estimateurs d'erreur. Le succès de ces estimateurs a ouvert la porte à de nombreuses théories mathématiques concernant le contrôle de l'erreur et la mise en place d'algorithmes adaptatifs.

Il existe deux grandes classes d'estimateurs : les estimateurs *a priori* et les estimateurs *a posteriori*. L'estimateur d'erreur, dit *a priori*, fournit des informations à propos de la convergence et de la stabilité des simulations numériques mises en place pour un certain problème physique. Ce type d'estimateur ne sera pas traité dans ce mémoire car il n'est pas suffisamment précis. Il traduit uniquement le comportement asymptotique de l'erreur, c'est-à-dire un comportement plus global de celle-ci alors que nous recherchons un estimateur qui soit capable de juger de l'erreur localement.

Nous nous intéresserons donc aux estimateurs d'erreur *a posteriori*. Comparé à un estimateur *a priori* qui assure la convergence de l'erreur à condition d'une régularité

suffisante de la solution exacte, l'estimateur *a posteriori* ne demande lui pas plus de régularité de la part de la solution exacte que celle imposée par la formulation faible. Ces estimateurs ont la capacité de tirer parti de la solution numérique ainsi que des données du problème pour accéder à la précision. De plus, ils peuvent être utilisés comme une mesure indépendante de la qualité de la simulation. L'estimateur *a posteriori* idéal devrait avoir les cinq propriétés [Adjerid et Flaherty, 1999] suivantes,

1. être asymptotiquement correct ;
2. être simple à calculer en n'utilisant qu'une petite fraction du coût de calcul de la solution ;
3. être robuste en fournissant une estimation précise pour une vaste gamme de maillage et d'ordre de méthodes ;
4. fournir des bornes inférieures et supérieures relativement strictes de l'erreur exacte dans une norme choisie ;
5. fournir des indicateurs d'erreur locaux qui génèrent une estimation de l'erreur globale dans plusieurs normes.

Il s'est avéré par le biais de plusieurs recherches menées par différentes équipes, [Verfürth, 1996] par exemple, qu'aucun estimateur à ce jour n'est capable de répondre à ces cinq propriétés pour tout type de maillage, de géométrie et d'ordre de méthodes.

La classe des estimateurs d'erreur *a posteriori* peut elle-même être décomposée en deux grandes branches : les estimateurs explicites et les estimateurs implicites dans lesquels nous pouvons retrouver les méthodes des éléments résiduels et les méthodes des résidus équilibrés. Les estimateurs explicites appliqués à des problèmes unidimensionnels ont été développés par [Babuska et Rheinboldt, 1979]. Babuska et Rheinboldt sont les pionniers dans le domaine des estimateurs. Ils se sont d'abord intéressés aux problèmes elliptiques unidimensionnels [Babuska et Rheinboldt, 1981] pour lesquels ils ont créé des estimateurs explicites en éléments finis évalués en norme énergétique. Leurs travaux ont été étendus aux problèmes paraboliques par [Bieterman, 1982], [Bieterman et Babuska, 1982] ainsi qu'aux approximations bilinéaires pour des problèmes bidimensionnels par [Kelly et al., 1983]. Ces estimateurs explicites peuvent être directement et explicitement calculés à partir de la solution aux éléments finis. D'un autre côté, nous retrouvons les estimateurs implicites qui quant à eux nécessitent la résolution d'un problème local aux valeurs frontières approximant l'équation résiduelle de l'erreur elle-même. L'estimateur d'erreur qui en découle est simplement la norme de la solution du problème ainsi posé. Il s'agit d'une résolution locale c'est-à-dire sur un élément ou sur un ensemble formé par ses voisins, par exemple, afin de satisfaire la deuxième propriété des estimateurs *a posteriori*.

A ce stade, nous nous questionnons sur l'intérêt du développement des estimateurs implicites nécessitant la résolution d'un problème aux valeurs frontières alors que des estimateurs explicites existent d'ores et déjà. Premièrement, il faut savoir que les estimateurs explicites fournissent une indication locale du niveau d'erreur. L'expression de cet indicateur local contient des constantes génériques pour lesquelles nous pouvons obtenir une borne supérieure sur leurs valeurs mais elle est souvent

prise dans le pire des cas, ce qui rend l'estimateur très pessimiste. Deuxièmement, les estimateurs explicites sont calculés sur base de deux résidus, celui évalué à l'intérieur de l'élément et celui évalué sur la frontière de l'élément. Cependant, le poids relatif à attribuer à chacun de ces résidus est loin d'être évident. Cette étape est pourtant cruciale pour le développement de ce type d'estimateur. Afin de contrer ces deux aspects négatifs des estimateurs explicites, les mathématiciens ont construit les estimateurs implicites et plus particulièrement les méthodes des éléments résiduels. En résolvant un problème local aux valeurs frontières, les constantes génériques sont évincées et la balance entre les deux résidus est traitée par le processus de résolution lui-même. C'est à la conférence de 1984 à Lisbonne que les méthodes des éléments résiduels furent présentées pour la toute première fois. Ces méthodes sont applicables à de nombreux problèmes de mécanique et de physique comme décrits par [Demkowicz, et al., 1984]. Dans leur article, les estimateurs sont appliqués à des problèmes d'écoulement de fluides visqueux au travers de conduites aux frontières mobiles ou encore à des phénomènes transitionnels de conduction de chaleur dans des domaines variables temporellement. Le même processus de méthodes résiduelles fut suivi par [Bank et Weiser, 1985]. Ils ont appliqué leur technique à des problèmes elliptiques en deux dimensions. Dans leurs travaux, nous pouvons retrouver deux idées fondamentales de la théorie des estimateurs *a posteriori* : l'hypothèse de saturation et le critère d'équilibre des données aux frontières.

Il réside malgré tout quelques déficits dans la formulation basique du problème local comme décrit dans les méthodes des éléments résiduels. En effet, la construction des valeurs frontières est souvent *ad hoc*. Afin que le problème soit bien posé, il est requis de choisir les données frontières de telle façon qu'elles soient équilibrées avec le résidu calculé à l'intérieur de l'élément. C'est [Ainsworth, 1994] qui a généralisé et validé la méthode des résidus équilibrés et les développements mathématiques de [Bank et Weiser, 1985].

Grâce à ces estimateurs, il devient possible de produire des maillages adaptatifs permettant de contrôler et de minimiser l'erreur. Dans les années 80, les stratégies adaptatives se mettent peu à peu en place. Bien qu'étant une grossière approximation de l'erreur exacte, les estimateurs gérant cette adaptation sont simples à implémenter, fournissent de bons résultats et sont suffisants pour mener le processus d'adaptation. Dans ce cadre, nous mentionnons l'estimateur par interpolation de [Demkowicz, 1985]. Ces estimateurs sont capables de localiser les régions du domaine observant une discontinuité, des chocs ou encore des ondes de raréfaction. Ils ont montré plus d'une fois leur efficacité dans les logiciels de calcul de dynamique des fluides.

A côté des estimateurs *a priori* et *a posteriori*, il existe une troisième classe d'estimateur dit estimateur par reconstruction de gradient. La simplicité de cet estimateur en fait un de ses atouts majeurs. Il se base sur l'obtention d'une approximation continue du gradient par le biais d'un post-traitement du gradient de la solution en éléments finis. La norme de la différence entre l'approximation post-traitée et le gradient de la solution en éléments finis est utilisée comme estimateur. Cet estimateur est assez grossier et ne possède pas toutes les belles propriétés de l'estimateur *a posteriori* mais

il s'avère être étonnamment efficace. Nous pouvons également citer les travaux de [Zienkiewicz et Zhu, 1987] qui s'inscrivent dans la gamme des estimateurs par reconstruction de gradient. Par la suite, ils ont développé la méthode dite *superconvergent patch recovery*. Ils annoncent que l'implémentation de leur technique est simple et rentable. Ils l'ont appliquée à une large gamme d'éléments linéaires, quadratiques et cubiques pour des problèmes unis et bidimensionnels. Ils ont pu observer une convergence en $\mathcal{O}(h^4)$ pour des éléments triangulaires quadratiques.

Comment utiliser ces estimateurs pour des classes de problèmes plus générales ? C'est la question centrale des années 90 et c'est [Verfürth, 1989] qui ouvre la marche en développant un estimateur à deux bornes pour les équations de Stokes et plus généralement celles de Navier-Stokes.

Les estimateurs d'erreur *a posteriori* sont d'abord développés pour les *h*-versions des éléments finis. La *h*-version des éléments finis est la stratégie de discrétisation la plus communément utilisée. Dans cette version, le degré des polynômes des éléments est fixé et c'est le maillage qui est raffiné tel que le diamètre du plus grand élément approche zéro. A contrario, les *p*-versions des éléments finis fixent la taille du maillage et c'est le degré des polynômes associés aux éléments qui augmente tel que le polynôme de degré minimal tend vers l'infini. Bien que principalement appliquées aux *h*-versions, les méthodes des éléments résiduels peuvent très bien s'accorder avec la *p*-version des éléments finis mais aussi avec une version mixte. Une étude intensive de ces méthodes en *p*-version, *h*-version et *h-p* version a été effectuée par [Oden, 1989]. Les auteurs ont étendu leur approche aux méthodes des résidus équilibrés.

Comme nous pouvons le constater, les stratégies adaptatives sont nombreuses et très souvent implémentées dans des logiciels de résolutions numériques. Elles ont prouvé leur efficacité mais la théorie décrivant leurs avantages face aux méthodes non-adaptatives n'est pas complète. [Binev, 2004] a repris la stratégie adaptative de [Morin, Nochetto et Sieber, 2000] pour résoudre l'équation de Poisson en deux dimensions avec des éléments linéaires par morceaux en y rajoutant une méthode de déraffinement. Dans leur article, ils prouvent la convergence dans la norme énergétique et fournissent également le taux de convergence de leur méthode. Ils utilisent une méthode de raffinement locale de leur maillage appelée *newest vertex bisection* qui consiste à successivement découper les éléments triangulaires en deux sous-éléments. Il a été démontré que cette technique appliquée aux éléments triangulaires permet de ne pas dégénérer leur forme après une utilisation répétée. Cette propriété de non-dégénérescence est démontrée par l'existence d'une concordance parfaite sur le graphe dual du maillage admissible initial. Nous pouvons également citer les travaux de [Dörfler, 1996] quelques années auparavant appliqués à l'équation de Poisson. Sa méthode ne contenait pas d'algorithme de déraffinement. Pourtant, il a pu obtenir une décroissance à taux constant de l'erreur en norme énergétique jusqu'à ce que la borne prescrite sur l'erreur soit atteinte.

Ce mémoire s'inscrit dans le cadre de la théorie des estimateurs d'erreur afin de

développer un estimateur d'erreur efficace en éléments finis pour un problème physique bien défini : les écoulements granulaires immergés. Ce mémoire a deux grands objectifs, le développement d'un estimateur d'erreur peu coûteux en temps de calcul, robuste et précis. L'estimateur le plus simple à implémenter serait de calculer la solution sur un maillage plus fin et d'ensuite comparer les deux solutions afin de juger de la précision de la solution. Bien que très simple, nous remarquons tout de suite que cet estimateur est coûteux en temps de calcul. Le chapitre 2 s'intéressera au développement théorique de deux estimateurs appliqués à un problème modèle. Nous avons choisi de comparer un estimateur *a posteriori* implicite et un estimateur à reconstruction de gradient. D'un côté, avec l'estimateur implicite, nous cherchons à éviter les problèmes concernant les constantes génériques et la question du poids relatif de chacun des deux résidus des estimateurs explicites. D'un autre côté, avec l'estimateur par reconstruction de gradient, nous recherchons le côté pratique de l'implémentation bien qu'il possède moins de propriétés que l'estimateur *a posteriori*. Dans le chapitre 3, nous commencerons par présenter le modèle des écoulements granulaires immergés comme implémenté dans le code MigFlow^[1]. Il s'agit d'un logiciel open source dédié à la résolution numérique d'écoulements granulaires secs, immergés et composés de deux fluides. Il est développé à l'UCLouvain conjointement avec l'université de Montpellier. L'équipe LMGC90 de Montpellier^[2] est pionnière dans le domaine des écoulements granulaires. Nous utiliserons l'estimateur par reconstruction de gradient sur le champ de vitesse et le champ de pression de l'écoulement. La question des grandeurs caractéristiques de l'écoulement sera centrale dans ce chapitre. Leur définition nous permettra de rendre adimensionnel notre estimateur. Nous nous intéresserons à deux grands types d'écoulements granulaires largement étudiés dans la littérature : la chute d'un nuage de grains dans un liquide visqueux à bas nombre de Reynolds et la sédimentation de suspensions mono-dispersées en tubes droits ou inclinés. Pour chaque cas, nous étudierons l'évolution des données macroscopiques comme la position du centre de masse du nuage et la vitesse de chute de celui-ci. La vitesse de sédimentation et la position du front, interface entre les grains et le fluide, seront étudiées pour l'exemple de la suspension en récipients droits et inclinés.

Grâce à l'estimateur d'erreur, nous pourrions adopter une stratégie adaptative qui sera présentée dans les chapitres 2 et 3. Cette stratégie adaptative sera appliquée sur le h -version des éléments finis. C'est bien le maillage qui sera raffiné selon deux techniques bien distinctes. Dans le chapitre 3, le maillage sera généré par un logiciel de remaillage. Cette approche ne tient pas compte de la structure existante du maillage. Nous verrons que cette technique engendre de la diffusion et donc une perte d'incompressibilité de l'écoulement. Cette observation nous amène au deuxième objectif de ce mémoire : l'adaptation locale du maillage. Cette approche tiendra compte du fait que d'une itération à l'autre le maillage est sensiblement le même et que seuls quelques éléments doivent être raffinés. Elle nous permet de maintenir une grande partie du maillage et de réduire le phénomène de diffusion. Le chapitre 4 se concentrera sur le développement informatique lié à l'adaptation locale. Il s'agit

1. <https://www.migflow.be>

2. <http://www.lmgc.univ-montp2.fr>

de la partie la plus informatique du mémoire qui a pour but de manipuler plusieurs structures de données afin de construire le nouveau maillage, dont des structures arborescentes.

CHAPITRE 2

CONSTRUCTION THÉORIQUE DE DEUX ESTIMATEURS D'ERREUR

L'objectif de ce chapitre est de comparer l'estimateur d'erreur *a posteriori* implicite à l'estimateur d'erreur par reconstruction de gradient pour un problème modèle. Tout au long de ce chapitre, nous allons nous cantonner à un simple problème modèle aux valeurs frontières. Pour un domaine polygonal Ω du plan $\mathbb{R}^2$, nous notons Γ sa frontière qui peut être décomposée en deux parties couvrant entièrement la frontière et ne se chevauchant pas, comme illustré à la figure (Fig. 2.1),

1. Γ_D , une frontière de Dirichlet sur laquelle les déplacements sont imposés ;
2. Γ_N , une frontière de Neumann sur laquelle les contraintes normales sont imposées.

Le problème modèle traité au cours de ce chapitre est donné par une équation elliptique aux dérivées partielles s'écrivant dans sa formulation forte comme,

$$\nabla \cdot (a \nabla u) + f = 0 \quad \forall \mathbf{x} \in \Omega \quad (2.1)$$

$$u = g_0 \quad \forall \mathbf{x} \in \Gamma_D \quad (2.2)$$

$$\hat{n} \cdot (a \nabla u) = g_1 \quad \forall \mathbf{x} \in \Gamma_N. \quad (2.3)$$

Les conditions aux frontières sont aussi appelées dans la théorie des éléments finis, conditions frontières essentielles (Eq. 2.2) et conditions frontières naturelles (Eq. 2.3). Le terme source dans (Eq. 2.1) est donné par la fonction f sur Ω et peut être physiquement interprété comme la densité de surface des forces. En règle générale, la fonction g_0 prescrite sur Γ_D se doit d'être continue alors que la fonction g_1 prescrite sur Γ_N peut, quant à elle, être discontinue.

La méthode des éléments finis utilisée ici pour résoudre numériquement le problème modèle (Eq. 2.1) repose sur une partition du domaine Ω , appelé aussi maillage non structuré. Dans ce mémoire, nous nous sommes uniquement intéressés aux partitions en triangles. La partition en éléments triangulaires est dite admissible si l'intersection commune de deux triangles est soit un noeud commun, soit une arête commune. De plus, deux triangles ne peuvent pas se chevaucher. La figure (Fig. 2.2) présente deux

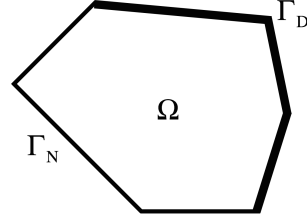


FIGURE 2.1 – Description du domaine muni de ses conditions frontières.

configurations non admissibles : (a) l'arête n'est pas commune aux deux triangles et (b) un noeud est flottant. Par ailleurs, l'ensemble formé par tous les éléments de la partition est noté $\mathcal{T}_h$ et la taille h de chacun de ces éléments est définie comme étant le diamètre du cercle circonscrit au triangle.

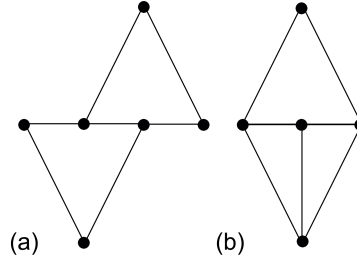


FIGURE 2.2 – Exemples de deux partitions non-admissibles car ces deux configurations présentent des noeuds flottants.

La théorie des estimateurs vise à construire un objet nous permettant d'approximer l'erreur exacte sur chaque élément de la partition. Cet estimateur d'erreur fournit sur chaque élément triangulaire K du maillage une valeur, notée κ . L'estimateur peut être construit soit sur un seul élément ou sur un sous-domaine. Dans ce cas-là, il faut considérer un ensemble d'éléments défini, par exemple, comme la réunion de ses voisins directs, noté $\tilde{K} = \text{int}\{\bar{J} : \bar{J} \cap \bar{K} \text{ est non-vide}\}$. Cependant, cet estimateur évalué sur un $\tilde{K}$ ne sera pas traité dans ce mémoire.

Dans ce chapitre, nous nous attelons au développement de deux estimateurs pour le problème modèle (Eq. 2.1) : l'estimateur d'erreur *a posteriori* résiduel en éléments finis et l'estimateur d'erreur par reconstruction de gradient. Ce chapitre se clôturera par une validation de ces deux estimateurs sur deux problèmes manufacturés.

2.1 Estimateur d'erreur *a posteriori* résiduel en éléments finis

La construction de cet estimateur est basée sur la théorie des éléments finis. Pour ce premier estimateur, nous nous sommes inspirés des travaux de [Bank et Weiser, 1985] et de [Verfürth, 1989]. Dans leurs articles, les auteurs ont décrit la méthode des éléments résiduels. L'idée de la méthode est d'augmenter l'espace de résolution pour construire une approximation d'ordre supérieur de la solution en éléments finis. Une

manière d'augmenter cet espace est de définir une ou plusieurs fonctions bulles. Bank et Weiser ont choisi d'utiliser uniquement des fonctions bulles définies sur les arêtes des éléments alors que Verfürth a, à la fois utilisé des fonctions bulles définies sur les arêtes mais aussi sur l'intérieur de l'élément. De notre côté, nous emploierons uniquement la fonction bulle intérieure pour dériver notre estimateur. De cette manière, nous ne prendrons pas en compte le résidu évalué sur la frontière de l'élément. Nous nous devons de commencer cette section par une définition formelle de la fonction bulle intérieure qui sera notre pilier pour la construction de l'estimateur.

A tout élément triangulaire du maillage non structuré, nous pouvons associer un élément de référence $\hat{K}$ défini sur ξ -domaine comme,

$$\hat{K} = \{(\xi, \eta) : 0 \leq \xi \leq 1, 0 \leq \eta \leq 1 - \xi\}.$$

Sur cet élément de référence, nous définissons une base de polynômes ϕ_i de degré p . Puisque nous choisissons d'associer à l'élément de référence trois noeuds Ξ_i distincts et non-alignés, nous obtenons trois polynômes à deux variables de degré au plus un valant l'unité sur le noeud auquel il est associé et zéro sur les deux autres. Ces polynômes appartenant à l'espace $\mathbb{P}_1$ sont appelés fonctions de forme locales. Elles sont schématisées à la figure (Fig. 2.3).

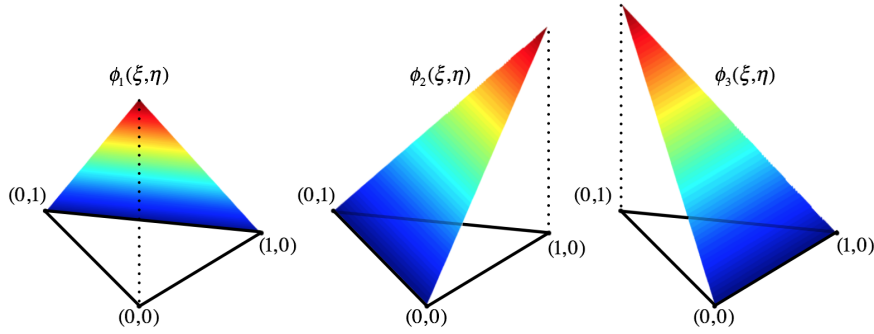


FIGURE 2.3 – Les trois fonctions de forme locales associées à l'élément de référence $\hat{K}$.

Un changement de variable $\mathbf{x} = \mathbf{x}(\xi)$ entre le domaine physique défini sur $\mathbf{x}$ et le domaine de calcul ξ est requis afin de passer de la représentation d'un quelconque élément K à l'élément de référence $\hat{K}$. Cette correspondance entre K et $\hat{K}$ est établie par,

$$\mathbf{x}(\xi) = \phi_1 \mathbf{X}_1^e + \phi_2 \mathbf{X}_2^e + \phi_3 \mathbf{X}_3^e. \quad (2.4)$$

C'est sur cet élément de référence, que nous pouvons définir rigoureusement la fonction bulle intérieure $\phi_+ : \hat{K} \mapsto \mathbb{R}$. Formellement, cette fonction est le polynôme de plus faible degré qui s'annule sur les frontières de l'élément de référence et qui est non nul à l'intérieur de $\hat{K}$. De par cette définition, la fonction bulle est simplement le produit des trois fonctions de forme locales.

2.1.1 Formulation faible du problème modèle

Nous devons tout d'abord définir les espaces contenant les fonctions u et $\hat{u}$ utilisées dans la formulation faible du problème modèle. Dans la théorie des éléments finis, la

fonction $\hat{u}$ est appelée fonction de test. Deux espaces seront principalement employés,

1. l'espace des fonctions carrées-intégrables :

$$L^2(\Omega) = \left\{ f : \Omega \mapsto \mathbb{R} \mid \int_{\Omega} |f|^2 < \infty \right\},$$

2. le second espace fait partie de la grande famille des espaces de Sobolev définis comme,

$$H^1(\Omega) = \left\{ u \in L^2(\Omega) \mid \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \in L^2(\Omega) \right\}.$$

Sur ce second espace, la norme induite est définie comme,

$$\|u\|_{H^1(\Omega)} = \left(\int_{\Omega} (\nabla u)^2 + \int_{\Omega} (u)^2 \right)^{1/2}. \quad (2.5)$$

Avant de construire l'estimateur résiduel en éléments finis, rappelons la forme variationnelle ou formulation faible du problème modèle (Eq. 2.1). Trouvé $u \in H^1(\Omega)$, telle que,

$$\underbrace{\langle (\nabla \hat{u}) \cdot (a \nabla u) \rangle}_{a(\hat{u}, u)} = \underbrace{\langle \hat{u} f \rangle + \langle \langle \hat{u} g_1 \rangle \rangle_N}_{b(\hat{u})} \quad \forall \hat{u} \in H_{\Gamma_D}^1(\Omega), \quad (2.6)$$

où $H_{\Gamma_D}^1(\Omega) = \{ \hat{u} \in H^1(\Omega) \mid \hat{u} = g_0, \text{ sur } \Gamma_D \}$. Pour que l'opérateur $\langle . \rangle$ soit valide dans la formulation (Eq. 2.6), la fonction f doit appartenir à l'espace $L^2(\Omega)$. Finalement, nous introduisons le sous-espace discret de résolution $\mathcal{U}^h = \{ u_h \in \mathcal{C}(\bar{\Omega}) \mid u_h|_K \in \mathbb{P}_1, \forall K \in \mathcal{T}_h \}$ comme l'espace des fonctions linéaires sur chaque élément et globalement continues. De cette façon, le problème d'approximation en éléments finis est de trouver $u^h \in \mathcal{U}^h$ telle que,

$$a(\hat{u}^h, u^h) = b(\hat{u}^h) \quad \forall \hat{u}^h \in \mathcal{U}^h.$$

L'erreur exacte d'approximation, notée $e = u - u^h$, appartient à l'espace de résolution $\mathcal{U}^h$ et satisfait la condition standard d'orthogonalité aussi appelée condition de Galerkin,

$$\langle r^h e \rangle = 0 \quad \text{où } r^h = \nabla \cdot (a \nabla u^h) + f \text{ et } \forall u^h \in \mathcal{U}^h. \quad (2.7)$$

Notre objectif est de quantifier cette erreur e sur chaque élément du maillage par le biais d'un estimateur d'erreur. La construction de cet estimateur est basée sur l'expression (Eq. 2.7) qui sera l'équation clé de nos développements. Nous la couplerons avec la définition de la fonction bulle intérieure afin d'obtenir une expression pour notre estimateur.

2.1.2 Formulation de l'estimateur d'erreur résiduel

L'idée de base est de représenter la solution en éléments finis u^h avec un degré de liberté en plus. Cette nouvelle approximation est notée u_+^h et sur chaque élément K du maillage, cette approximation est donnée par la théorie des éléments finis comme,

$$u_+^h = \sum_{i=1}^{p+1} U_i \phi_i(\mathbf{x}) + U_+ \phi_+ = u^h + U_+ \phi_+,$$

où ϕ_+ est la fonction bulle définie précédemment. Le degré de liberté est fourni par cette fonction. Nous définissons donc l'espace de résolution augmenté par $\mathcal{U}_+^h = \text{span}\{\phi_+\} \cup \mathcal{U}^h$. L'erreur exacte e peut alors être approximée par $e^h = u_+^h - u^h$, satisfaisant la condition d'orthogonalité sur l'espace augmenté $\mathcal{U}_+^h$. En reprenant la condition d'orthogonalité (Eq. 2.7) appliquée à cet espace augmenté, nous obtenons une équation linéaire d'inconnue U_+ à résoudre sur chaque élément K du maillage,

$$\langle r_+^h \phi_+ \rangle = 0 \Rightarrow \left\langle \left(\nabla \cdot (a \nabla u_+^h) + f \right) \phi_+ \right\rangle = 0. \quad (2.8)$$

En développant cette équation, le problème se réduit à trouver l'expression de U_+ en utilisant l'approximation en éléments finis u^h ainsi que les données du problème modèle, les fonctions f , g_0 et g_1 . Étant donné la formulation du problème modèle et l'utilisation des fonctions de forme linéaires, il s'avère que plusieurs termes de l'équation peuvent être calculés avant même de calculer la solution u^h ce qui n'est pas nécessairement le cas pour tous les problèmes physiques. En appliquant l'intégration par parties et le théorème de la divergence sur l'équation (Eq. 2.8), nous obtenons une expression pour le terme U_+ ,

$$\begin{aligned} \langle \nabla \phi_+ (a \nabla u_+^h) \rangle &= \langle \nabla \phi_+ (a \nabla u_h) \rangle + a U_+ \langle (\nabla \phi_+) (\nabla \phi_+) \rangle, \\ &\quad \downarrow \text{La fonction } \phi_+ \text{ s'annule sur la frontière de l'élément.} \\ U_+ &= \frac{\langle f \phi_+ \rangle - a \langle \nabla \phi_+ (\nabla u^h) \rangle}{a \langle (\nabla \phi_+) (\nabla \phi_+) \rangle}. \end{aligned}$$

L'estimateur résiduel s'écrit donc sur chaque élément K du maillage comme la norme- $L_2(\Omega)$ de l'erreur approximée, $\kappa = \|e^h\|_{L_2(\Omega)}$ ou comme la norme- $H_1(\Omega)$, $\kappa = \|e^h\|_{H_1(\Omega)}$.

2.2 Estimateur d'erreur par reconstruction de gradient

Pour le même problème modèle, nous allons construire un estimateur par reconstruction de gradient. Dans la littérature, plusieurs approches ont été développées. Dans le domaine, nous pouvons citer les travaux de [Babuska et Rheinboldt, 1981]. De leur point de vue et dans le but d'être largement utilisable, l'estimateur se doit d'être disponible pour diverses normes physiquement importantes. Nous retrouvons aussi les travaux de [Babuska, et al., 1986] dans lesquels les auteurs ont développé un estimateur pour des éléments quadrilatères. Néanmoins, ces méthodes se basent toutes plus ou moins sur la même idée : post-traiter le gradient de la solution en éléments finis afin d'obtenir une approximation plus précise du gradient de la solution exacte. Cette méthode est très populaire parmi la communauté des ingénieurs et s'avère fournir de bons résultats. *A contrario* de l'estimateur *a posteriori* implicite développé à la section précédente, l'estimateur par reconstruction que nous allons construire n'utilise aucune donnée du problème originel.

Nous devons construire un opérateur de reconstruction de gradient G^h . Il se doit de fournir une approximation du gradient de la solution en éléments finis lorsque nous l'appliquons à u^h . Nous voulons que le calcul de cet opérateur soit local. Autrement dit, sur l'élément K , le calcul de l'opérateur ne peut profiter que des informations obtenues sur les voisins directs de l'élément K . En outre, cet opérateur doit répondre à une série de conditions suffisantes pour qu'appliqué à la solution u^h , il soit une bonne approximation du vrai gradient. Ces conditions [Ainsworth et Oden, 1997] sont les suivantes,

1. **Condition de consistance** : Si u appartient au sous-espace d'éléments finis d'ordre $p + 1$ alors,

$$G^h(u^h) = \nabla u.$$

2. **Condition de localisation** : Si $x^* \in K$ alors la valeur du gradient reconstruit, noté $G^h[v](x^*)$ ne dépend que de la valeur de ∇v échantillonné sur le voisinage $\tilde{K}$.

3. **Limite et condition de linéarité** : $G^h : \mathcal{U}^h \mapsto \mathcal{U}^h \times \mathcal{U}^h$ est un opérateur linéaire et il existe une constante C (indépendante de la taille h du maillage) telle que,

$$\|G^h[v]\|_{L_2(K)} \leq C|v|_{W^{1,\infty}} \quad \forall K \in \mathcal{P} \quad \forall v \in \mathcal{U}^h.$$

L'estimateur par reconstruction de gradient sur l'élément K du maillage est alors exprimé comme la norme- $L_2(\Omega)$ de l'erreur $\kappa = \|G^h(u^h) - \nabla u^h\|_{L_2(\Omega)}$.

Nous nous sommes basés sur les travaux de Rank [Rank et Zienkiewicz, 1987] pour construire un estimateur par reconstruction de gradient qu'il nomme dans son article *the jump indicator*. Puisque nous travaillons avec des fonctions de forme linéaires, le gradient de l'approximation par éléments finis est constant par morceaux et il est donc discontinu aux interfaces. L'idée principale de cette méthode est de lisser le gradient de la solution. Cette étape de lissage peut être approchée par différentes techniques comme la moyenne nodale ou l'interpolation locale. L'un des gros avantages de cette classe d'estimateur, est qu'il est assez aisé d'implémenter l'estimateur dans un code en éléments finis existant. Considérant toujours le problème modèle (Eq. 2.1) en deux dimensions avec le paramètre $a \in \mathbb{R}$, nous définissons la vitesse $\mathbf{v}$ de la solution exacte u comme $\mathbf{v} = (v_x, v_y)^T = -a\nabla u$. L'erreur exacte exprimée en norme énergétique sur l'élément K du maillage est exprimée comme,

$$\|e\|_E^2 = \int_K [\mathbf{v} - \nabla u^h]^T \nabla(u - u^h) d\Omega = \int_K [\mathbf{v} - \nabla u^h]^T \begin{bmatrix} 1/a & 0 \\ 0 & 1/a \end{bmatrix} [\mathbf{v} - \nabla u^h] d\Omega$$

Ne connaissant pas la solution exacte du problème et donc encore moins le gradient de celle-ci, l'objectif est de trouver une approximation de la vitesse $\mathbf{v}$ de la solution exacte. En utilisant une technique de lissage, nous allons remplacer la vitesse $\mathbf{v}$ par sa version lissée $\bar{\mathbf{v}}$ afin d'obtenir notre estimateur κ évalué sur l'élément K en norme énergétique,

$$\kappa^2 = \int_K [\bar{\mathbf{v}} - \nabla u^h]^T \begin{bmatrix} 1/a & 0 \\ 0 & 1/a \end{bmatrix} [\bar{\mathbf{v}} - \nabla u^h] d\Omega \quad (2.9)$$

Dans notre cas, la technique de lissage adoptée est l'interpolation des sauts de gradient évalués aux interfaces des éléments. Pour un problème bidimensionnel, nous évaluerons

les sauts de gradient au milieu de chaque arête et nous utiliserons ensuite une reconstruction linéaire du gradient sur chaque élément. Cette technique est différente de celle adoptée par [Babuska, et al., 1986]. Dans leur article, les auteurs évaluaient le gradient reconstitué sur chaque noeud. La valeur au noeud $\mathbf{x}$ était obtenue comme une somme pondérée des gradients échantillonnés au centre des éléments quadrilatères s'intersectant au noeud $\mathbf{x}$. Ce n'est qu'ensuite qu'ils interpolaient les valeurs nodales par les fonctions de forme locales afin d'obtenir leur approximation. L'avantage de cette technique est que l'approximation obtenue est continue sur le domaine alors que notre technique ne permet pas une telle propriété, l'approximation est continue uniquement sur l'élément.

2.3 Validation des estimateurs pour un problème unidimensionnel

Afin d'illustrer mais aussi dans le but de valider l'approche des deux estimateurs, nous réduisons notre problème modèle à un problème à une dimension pour lequel il est possible de calculer la solution exacte. Pour encore simplifier l'approche, nous considérons le paramètre a comme une constante. Le domaine de résolution est donné par un intervalle fermé $\Omega = [0, 1]$. Les conditions aux frontières sont des conditions de Dirichlet homogènes. Le problème unidimensionnel considéré est le suivant,

$$\begin{cases} -au'' = f & \forall x \in \Omega \\ u(0) = u(1) = 0 \end{cases} \quad (2.10)$$

Nous allons comparer les deux estimateurs, l'approche en éléments finis nécessitant l'augmentation de l'espace de résolution et l'approche liée à l'évaluation des sauts de gradient aux interfaces. Le problème (Eq. 2.10) est résolu par la méthode des éléments finis en utilisant des fonctions de forme linéaires. Dans le cas présent, seules deux fonctions de forme sont requises définissant l'espace de résolution par $\mathcal{U}^h = \text{span}\{\phi_1, \phi_2\}$, où $\phi_1 = \frac{1}{2}(1 - \xi)$ et $\phi_2 = \frac{1}{2}(1 + \xi)$. Le domaine de résolution, soit l'intervalle fermé, est subdivisé en une série de sous-intervalles de tailles différentes, $h_i = X_{i+1} - X_i$ for $i = 1, \dots, N$.

L'estimateur résiduel

En se basant sur les développements effectués dans la section 2.1, nous pouvons obtenir l'estimateur résiduel dans le cas unidimensionnel. L'estimateur local, sur chaque sous-intervalle du domaine Ω est donné par,

$$\kappa_i^2 = \|u_+^h - u^h\|_{H_{1,i}}^2 = \|U_+ \phi_+\|_{H_{1,i}}^2.$$

La valeur de la solution U_+ est fournie par l'expression (Eq. 2.8). Désormais, il ne reste qu'à évaluer l'erreur en norme H_1 sur chaque sous-intervalle. En utilisant l'isomorphisme entre l'intervalle parent et le i -ème intervalle, nous obtenons une

expression de l'estimateur en fonction uniquement de la taille du maillage h_i et de la valeur de U_+ ,

$$\begin{aligned}\kappa_i^2 &= \int_{X_i}^{X_{i+1}} \left[\frac{d}{dx} (U_+ \phi_+(x)) \right]^2 + (U_+ \phi_+)^2 dx \\ &= \frac{2U_+^2}{h_i} \int_{-1}^1 \left(\frac{d\phi_+}{d\xi} \right)^2 d\xi + \frac{h_i U_+^2}{2} \int_{-1}^1 \phi_+^2 dx \\ &= \frac{U_+^2}{3} \left(\frac{1}{h_i} + \frac{h_i}{10} \right).\end{aligned}$$

Notons que pour avoir une convergence de l'estimateur, autrement dit $\lim_{h \rightarrow 0} \kappa = 0$, il est nécessaire que la valeur U_+ tend plus rapidement vers zéro que la taille de l'intervalle. En observant de manière attentive l'expression de U_+ (Eq. 2.8), nous remarquons que le numérateur est proportionnel à h_i alors que le dénominateur est proportionnel à $\frac{1}{h_i}$. La valeur de U_+ est proportionnelle à h_i^2 et tend bien vers une valeur nulle pour une taille de maille h_i infinitésimale. Nous pouvons donc valider la convergence de l'estimateur résiduel mis en place pour l'équation de Poisson unidimensionnelle.

Dans le cas unidimensionnel, il devient plus évident de représenter la fonction bulle intérieure. Etant obtenue par le produit des deux fonctions de forme linéaires, elle a donc la forme d'une parabole inversée qui lui donne ce caractère arrondi typique d'une bulle. En vue de comprendre plus intuitivement le fonctionnement de la fonction bulle intérieure, nous résolvons d'abord le problème (Eq. 2.10) pour $f = \pi^2 \sin(\pi x)$ pour $N = 5$ intervalles de même taille puis pour $N = 6$ intervalles de tailles différentes. La figure (Fig. 2.4) présente les solutions obtenues u^h et la solution exacte du problème ainsi que les fonctions bulles sur chaque sous-intervalle. Concernant la subdivision en $N = 5$ intervalles, la solution u^h approxime mal la solution exacte sur le troisième intervalle du fait de la courbure de la solution. Notons que l'amplitude de la fonction bulle sur cet intervalle est nettement plus élevée que sur les quatre autres. Par cette représentation, nous mettons bien en évidence le lien entre l'amplitude de la fonction bulle et le niveau d'erreur commis. Lorsque nous découpons l'intervalle central en deux, nous observons une diminution de plus de la moitié de l'amplitude de la fonction bulle sur le troisième et quatrième intervalles par rapport à la précédente. Le cas unidimensionnel est un peu particulier. En effet, l'approximation en éléments finis interpole la solution exacte aux noeuds, [Tong, 1969]. C'est pour cette raison que l'amplitude des autres fonctions bulles n'est pas altérée par la subdivision de l'intervalle central. Cette remarque n'est plus valide en dimensions supérieures.

Estimateur par reconstruction de gradient

Cet estimateur est construit grâce à une interpolation de la vitesse de la solution u^h , notée $\bar{v}$, comme fourni par l'expression (Eq. 2.9) que nous écrivons dans le cas unidimensionnel comme, $\kappa_i^2 = \frac{1}{k} \int_i (\bar{v}(x) - (u^h)'(x))^2 dx$. Puisque l'approximation en éléments finis est du premier ordre, le gradient de celle-ci est constant par morceaux sur le domaine. Il y a donc une discontinuité à chaque interface. Nous devons donc évaluer le saut de gradient à gauche $aJ_l[(u^h)']$ et à droite $aJ_r[(u^h)']$ sur chaque noeud.

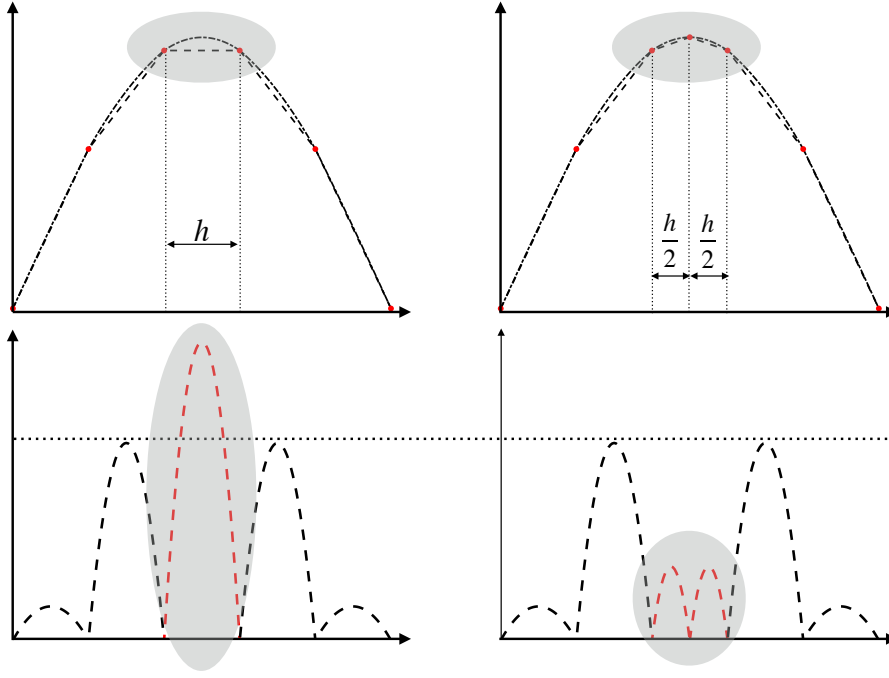


FIGURE 2.4 – Approximation en éléments finis u^h en pointillé, solution augmentée u_+^h en trait pointillé et fonctions bulles intérieures pour l'équation de Poisson unidimensionnelle. L'amplitude de la fonction bulle de l'intervalle central diminue lorsque cet intervalle est subdivisé. Grâce à cette subdivision, il en résulte que la solution u^h approxime mieux la courbure de la solution.

Ensuite, les deux valeurs nodales de chaque intervalle sont interpolées par les deux fonctions de forme ϕ_1 et ϕ_2 . L'interpolation linéaire s'écrit donc comme,

$$\bar{v}_i(x) = (u^h)'_i(x) - \frac{a}{2} J_l[(u^h)'_i] + a \left(J_l[(u^h)'_i] + J_r[(u^h)'_i] \right) \left(\frac{x - X_i}{2h_i} \right),$$

où, $J_l[(u^h)'_i] = (u^h)'_i - (u^h)'_{i-1}$ and $J_r[(u^h)'_i] = (u^h)'_{i+1} - (u^h)'_i$. Finalement, nous calculons la norme de l'erreur,

$$\begin{aligned} \kappa_i^2 &= \frac{a}{4} \int_{X_i}^{X_{i+1}} \left(J_l - (J_l + J_r) \frac{x - X_i}{h_i} \right)^2 dx = \frac{a}{4} \int_0^1 \left(J_l - (J_l + J_r) \xi \right)^2 d\xi \\ &= \frac{ah_i}{12} \left(J_l^2 - J_l J_r + J_r^2 \right). \end{aligned}$$

Cet estimateur dépend des sauts de gradient aux interfaces et est directement proportionnel à la taille du maillage. Autrement dit, cet estimateur converge lorsque la taille de la maille devient de plus en plus fine, cet estimateur est bien consistant.

Comparaison des deux estimateurs

Ayant implémenté ces deux estimateurs pour le problème modèle dans le cas unidimensionnel avec le paramètre a constant, égal à l'unité et avec la fonction $f = \pi^2 \sin(\pi x)$, nous pouvons observer sur la figure (Fig. 2.5), la solution exacte comparée à l'approximation en éléments finis et à l'approximation augmentée. Notons que la plus

grande erreur se concentre au centre du domaine. C'est à cet endroit que l'amplitude de l'estimateur devra être la plus élevée.

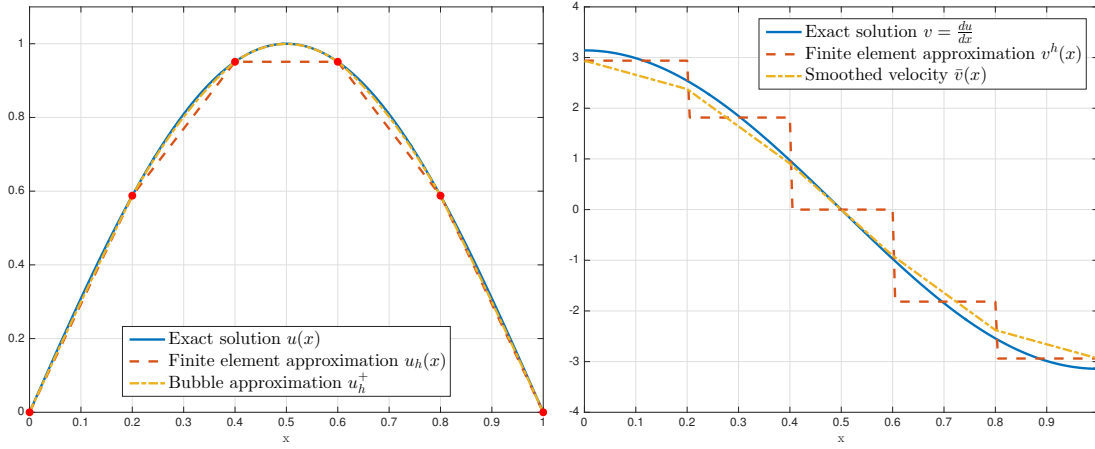


FIGURE 2.5 – Sur la figure de gauche, la solution exacte est comparée aux solutions u^h et u_+^h alors que sur la figure de droite, la vitesse de la solution exacte est comparée à la vitesse de l'approximation en éléments finis et à la vitesse reconstituée $\bar{v}$.

Dans chacun des cas, nous pouvons calculer l'erreur exacte puisque le problème sélectionné possède une unique solution analytique donnée étant $u(x) = \sin(\pi x)$. Nous pouvons donc calculer l'erreur exacte en norme $H_1(\Omega)$ et la confronter aux deux estimateurs. La figure (Fig. 2.6) affiche les deux estimateurs, chacun normalisé par leur valeur maximale atteinte sur le domaine mais aussi l'erreur exacte en norme H_1 , celle-ci est également normalisée par sa valeur maximale, pour différents nombres d'intervalles. Comme présager, les deux estimateurs indiquent un niveau élevé d'erreur au centre du domaine. L'estimateur résiduel a tendance à sous-estimer l'erreur exacte alors que l'estimateur par reconstruction de gradient surestime l'erreur plus on se rapproche des frontières. En effet, au niveau des frontières, nous ne pouvons fournir qu'une seule valeur de sauts, à gauche (droite) si la frontière se trouve à droite (gauche). Nous pourrions utiliser les conditions aux frontières dans le cas où la dérivée de la solution serait connue par le biais d'une condition de Neumann. Cependant, dans le présent exemple, nous ne connaissons pas la dérivée de la solution car ce sont des conditions de Dirichlet qui imposent la valeur de la solution aux frontières. La seule façon de réagir face à ce manque d'information est d'imposer un saut à zéro. En augmentant le nombre d'intervalles, les deux estimateurs se comportent de manière très similaire et indiquent bien là où l'erreur est importante.

2.4 Stratégie adaptative

Afin d'interpréter les résultats du problème manufacturé bidimensionnel, nous devons encore discuter de la stratégie adaptative à mettre en place. Cette stratégie a pour but d'établir le lien entre la taille locale h d'un élément K , la valeur de l'estimateur κ sur ce même élément et l'erreur prescrite e^* . L'erreur prescrite est fournie comme argument d'entrée de l'algorithme et pourra être définie par l'utilisateur. Nous faisons

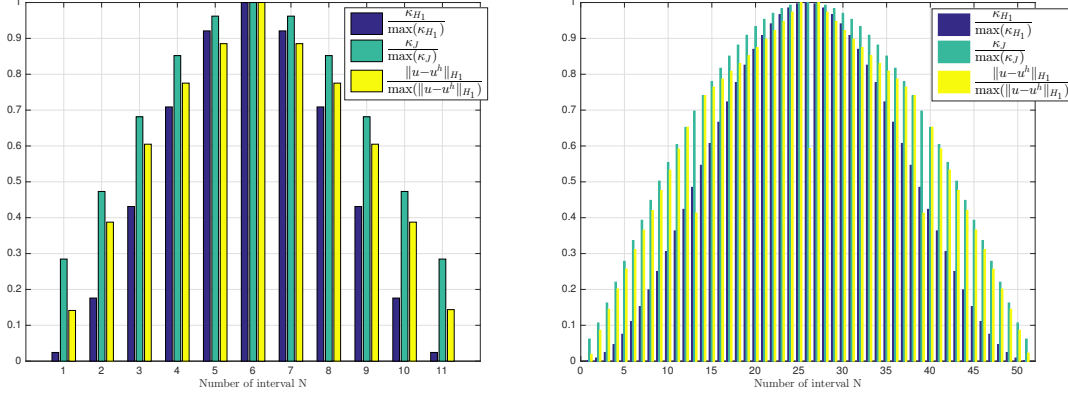


FIGURE 2.6 – Pour un nombre d’intervalles de $N = 11$ et $N = 51$, nous observons l’estimateur résiduel en norme H_1 en bleu, l’estimateur par reconstruction de gradient en turquoise et en jaune, l’erreur exacte en norme H_1 .

l’hypothèse que cette erreur est uniformément distribuée sur le maillage ce qui nous permet d’écrire une relation simple entre l’erreur prescrite globale et le niveau d’erreur autorisé localement : $e_K^* = e^*/\sqrt{N}$, où N est le nombre total de triangles dans le maillage. Grâce aux estimateurs précédemment construits et à un ratio entre l’erreur prescrite locale et l’estimateur, nous pouvons déterminer si l’élément K doit être raffiné ou non. Ce ratio évalué sur l’élément K ,

$$\zeta_K = \frac{\kappa_K}{e_K^*} > 1,$$

indique que cet élément doit être raffiné. Le même ratio fut utilisé par Zienkiewicz et Zhu pour leur stratégie de raffinement, [Zienkiewicz et Zhu, 1987]. Leur stratégie était fondée sur le développement d’un estimateur d’erreur par reconstruction de gradient appliqué à des problèmes élastiques linéaires. Pour un tel problème physique, leur estimateur était construit sur base de la contrainte d’élasticité. Afin d’obtenir des résultats acceptables pour la contrainte lissée, ils ont eu recours à une technique de projection nodale dans laquelle la contrainte était interpolée par les mêmes fonctions de forme que celles utilisées pour résoudre le déplacement.

Dans leur article, [Zienkiewicz et Zhu, 1987] ont assumé que la convergence de l’erreur était en $\mathcal{O}(h^p)$ valide pour la surface couverte par l’élément. La taille prédite h_K^* de l’élément est donnée par,

$$h_K^* = \frac{h_K}{(\zeta_K)^{1/p}},$$

où p est le degré de fonctions de forme locales. Il est à noter que cette convergence n’est plus valable proche des singularités. En utilisant cette dernière expression telle quelle, il est possible que la taille d’une maille soit drastiquement diminuée ce qui pourrait générer une multitude de nouveaux éléments. Afin d’éviter un excès d’éléments d’une itération à l’autre, nous décidons de diviser (multiplier) successivement les éléments par une constante α comme expliqué dans l’algorithme (Alg. 1). En plus de cette précaution, nous choisissons aussi de borner la taille des mailles par une valeur minimale et maximale. L’algorithme mis en place s’applique sur chaque élément du maillage existant.

Algorithm 1 *Stratégie d'adaptation du maillage*

```
1: procedure ADAPTIVE_STRATEGY( $h, h^*, \alpha, h_{\min}, h_{\max}$ )
2:    $h^* \leftarrow \frac{h}{(\zeta)^{1/2}}$ 
3:   if  $h^* \geq \alpha h$  then
4:      $h^* \leftarrow \alpha h$ 
5:   else if  $h^* \leq h/\alpha$  then
6:      $h^* \leftarrow h/\alpha$ 
7:   else
8:      $h^* \leftarrow h$ 
9:    $h^* \leftarrow \max(\min(h^*, h_{\max}), h_{\min})$ 
```

Seulement, cet algorithme ne permet pas de définir la taille des éléments afin d'avoir un nombre fixe d'éléments dans le nouveau maillage. Malgré les précautions prises, nous ne sommes pas protégés contre un accroissement drastique du nombre d'éléments. Il est toutefois possible en choisissant correctement le paramètre α et l'erreur prescrite e^* de diminuer le nombre total d'éléments dans le nouveau maillage.

Une fois la nouvelle carte de tailles obtenue, nous avons plusieurs possibilités pour adapter le maillage en conséquence. Une première technique est d'utiliser un logiciel de remaillage qui va sur base de la carte de tailles redéfinir un nouveau maillage complet en détruisant toute l'information de l'ancien. Une deuxième technique est de prendre en considération le maillage existant et de lui apporter une série de modifications afin d'obtenir le maillage raffiné. Nous avons utilisé la première technique pour l'exemple de la section 2.5 en utilisant le logiciel **GMSH**^[1]. La seconde approche sera quant à elle traitée plus amplement dans le chapitre 4.

2.5 Validation des estimateurs pour un problème bidimensionnel

Dès à présent, nous nous intéressons à la résolution d'un problème bidimensionnel manufacturé qui nous permettra de comparer nos deux estimateurs. Le caractère manufacturé nous permet comme dans l'exemple unidimensionnel, d'obtenir la solution du problème. Pour autant que la fonction f soit carrée intégrable au sens de l'intégrale de Lebesgue, la solution du problème modèle (Eq. 2.1) sera unique, continue, analytique et dérivable.

Nous nous attendons à ce que nos estimateurs soient capables de localiser l'erreur exacte. Par contre, nous ne recherchons pas tellement à obtenir un facteur d'efficacité $\kappa_K/\|u - u^h\|_K$ proche de l'unité.

Le domaine considéré est un carré $\Omega = [0, 1] \times [0, 1]$. Par simplicité, nous imposons le paramètre a constant et égal à l'unité. Nous voulons une solution $u(x, y)$ possédant un gradient le long de la diagonale $[0, 1] \rightarrow [1, 0]$. Sur cette diagonale, nous espérons

1. <http://www.geuz.org/gmsh>

que l'estimateur nous indique que le maillage soit raffiné. Une telle fonction est définie par,

$$u(x, y) = \frac{1}{2} \left(1 + \tanh [k(x + y)] \right).$$

En dérivant deux fois $u(x, y)$, nous obtenons l'expression de la fonction $f(x, y)$,

$$f(x, y) = 2k^2 \tanh [k(x + y)] \left(\operatorname{sech} [k(x + y)] \right)^2.$$

Le paramètre k permet de rendre la fonction plus ou moins raide sur la diagonale si sa valeur augmente ou diminue, respectivement.

L'estimateur résiduel

La première étape est de calculer une expression pour U_+ . Pour une méthode $\mathbb{P}_1$, les fonctions de forme sont linéaires et l'expression de U_+ peut s'obtenir en calculant analytiquement chaque terme de l'équation (Eq. 2.8) :

1. le calcul du terme $\langle f \phi_+ \rangle$,
2. le calcul du terme $\langle \nabla \phi_+ (\nabla u^h) \rangle$,
3. et, finalement, le calcul du dénominateur $\langle (\nabla \phi_+) (\nabla \phi_+) \rangle$.

Ces étapes sont reprises en détail dans l'annexe A. Nous obtenons l'expression de l'estimateur local $\kappa_K = J_K \frac{|U_+|}{12\sqrt{35}}$ en norme L_2 , et où J_K est le Jacobien de la transformation sur l'élément K . L'estimateur étant directement proportionnel à l'aire des éléments du maillage, si le maillage devient de plus en plus fin, l'estimateur tend également vers zéro. Autrement dit, pour un maillage infinitésimal, l'erreur commise sera très faible.

Après avoir résolu l'équation de Poisson avec une méthode en éléments finis à fonctions de forme linéaires, nous pouvons calculer numériquement l'estimateur d'erreur résiduel sur chaque élément et ensuite appliquer l'algorithme adaptatif décrit à la section 2.4 afin d'obtenir le nouveau maillage par **GMSH**. Nous observons bien que l'erreur est la plus élevée dans la diagonale montrant bien que dans cette région, le nouveau maillage sera beaucoup plus fin. Par contre, le maillage sera bien plus grossier sur les régions plates de la solution $u(x, y)$ comme le montre la figure (Fig. 2.7) sur laquelle nous affichons l'estimateur sur la gauche, normalisé par sa valeur maximale prise sur le domaine et l'erreur exacte évaluée en norme L_2 , elle aussi normalisée par sa valeur maximale, sur le graphe de droite. Nous notons une certaine ressemblance entre les deux graphes montrant bien que l'estimateur localise l'erreur exacte comme attendu. En augmentant le nombre d'éléments dans le maillage jusqu'à 727 éléments (Fig. 2.8), nous observons le même comportement, une erreur élevée dans la diagonale et une erreur plus faible dans le coin haut droit et bas gauche.

Il est important de souligner que l'estimateur résiduel peut aussi être calculé en norme H_1 affiché à la figure (Fig. 2.9). L'estimateur nous indique toujours une erreur élevée dans la diagonale principale et la ressemblance entre l'estimateur et l'erreur exacte mesurée en norme H_1 est forte. De nouveau, en poussant la résolution à 964

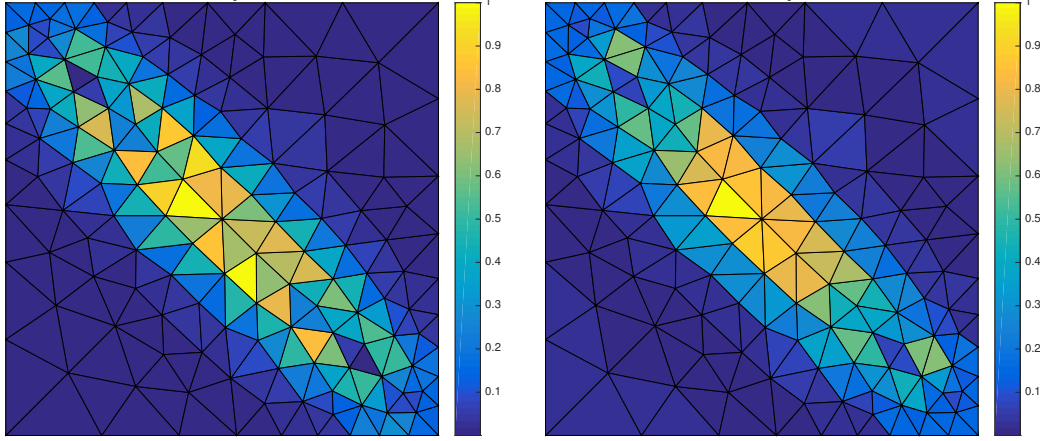


FIGURE 2.7 – Estimateur d’erreur en norme L_2 pour 252 éléments comparé à l’erreur exacte évaluée en norme L_2 à droite.

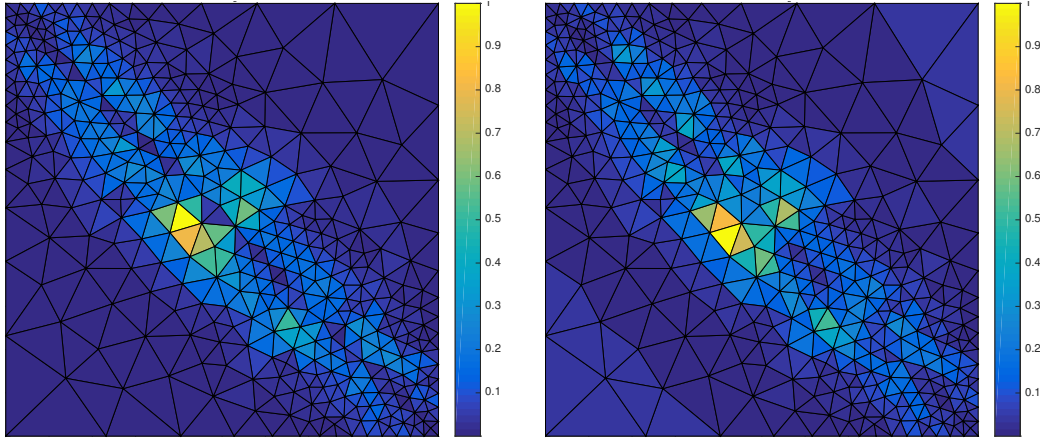


FIGURE 2.8 – Estimateur d’erreur en norme L_2 pour 727 éléments comparé à l’erreur exacte évaluée en norme L_2 à droite.

triangles, la ressemblance est d’autant plus frappante. Tout comme pour l’estimateur en norme L_2 , une ligne de triangles de faible erreur apparaît sur la diagonale. La solution du problème présente une portion linéaire aux alentours du point d’inflexion. Puisque les fonctions de forme sont linéaires, cette partie de la solution est bien représentée. Il y a donc une faible marge d’erreur entre la solution exacte et la solution approchée.

L’estimateur par reconstruction de gradient

A la différence des travaux réalisés par [Rank et Zienkiewicz, 1987], nous n’évaluons pas le saut de gradient aux noeuds des éléments mais sur le milieu des arêtes comme schématisé à la figure (Fig. 2.11). Ensuite, nous effectuons une interpolation linéaire locale de ces trois valeurs. L’intérêt de travailler sur les noeuds est qu’il est possible de réutiliser les fonctions de forme de la méthode des éléments finis utilisées pour reconstruire la solution u^h . Cependant, pour travailler sur les noeuds, il serait néces-

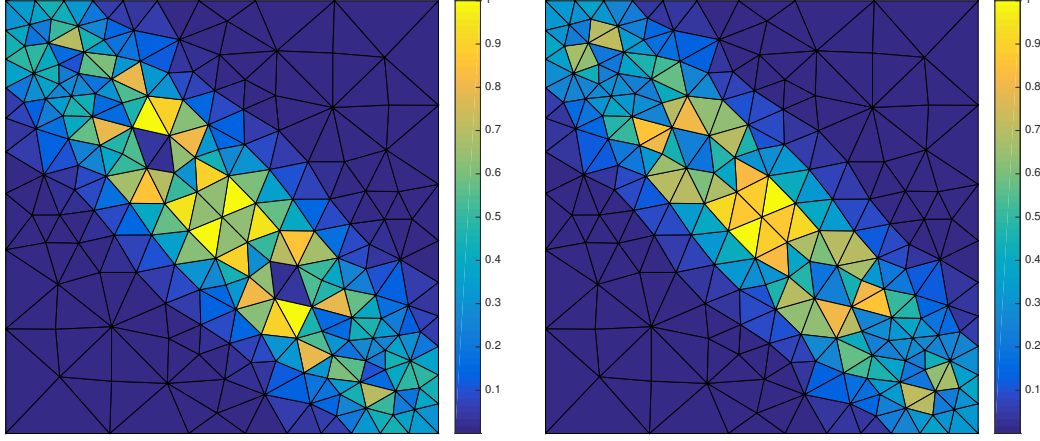


FIGURE 2.9 – Estimateur d’erreur en norme H_1 pour 324 éléments comparé à l’erreur exacte évaluée en norme H_1 à droite.

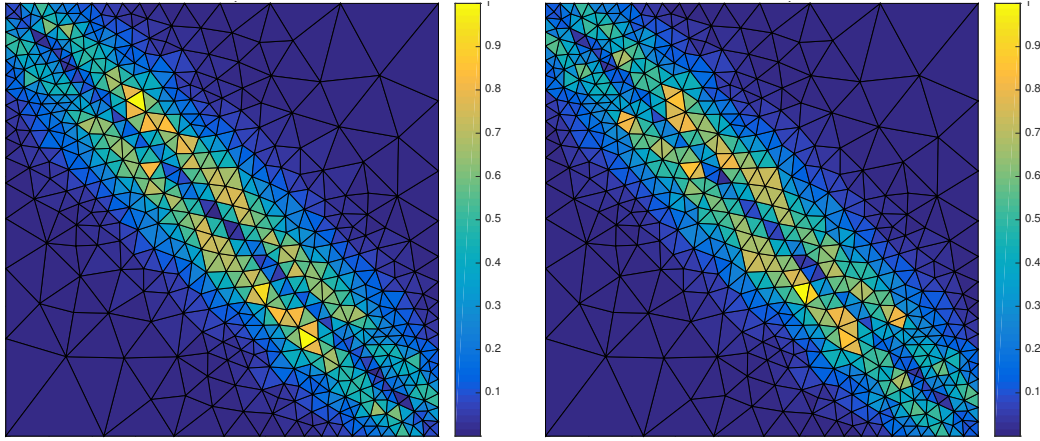


FIGURE 2.10 – Estimateur d’erreur en norme H_1 pour 964 éléments comparé à l’erreur exacte évaluée en norme H_1 à droite.

saire de moyenner la valeur de chaque milieu d’arêtes au noeud commun de celles-ci. Cette première étape serait déjà une forme de lissage du gradient et nous n’avons pas souhaité l’implémenter. Le second intérêt d’interpoler sur les noeuds et non au milieu des arêtes est que la fonction obtenue est globalement continue sur le domaine ce qui ne sera pas le cas avec une interpolation sur les arêtes. Néanmoins, le but étant d’évaluer localement l’erreur en intégrant sur l’élément le carré de la différence entre le gradient de la solution en éléments finis et le gradient reconstitué, la discontinuité à l’interface ne pose pas de problème.

Dans notre cas, nous devons utiliser trois autres fonctions d’interpolation. La fonction associée à une arête vaut l’unité au milieu de celle-ci et s’annule sur les deux autres. Ces trois fonctions sont définies sur le triangle de référence $\hat{K}$ de la manière suivante, $\psi_1 = 2\xi + 2\eta - 1$, $\psi_2 = 1 - 2\xi$ et $\psi_3 = 1 - 2\eta$.

L’estimateur est toujours donné par l’expression (Eq. 2.9). Cette expression se simplifie puisque le paramètre a a été choisi constant et égal à l’unité et nous

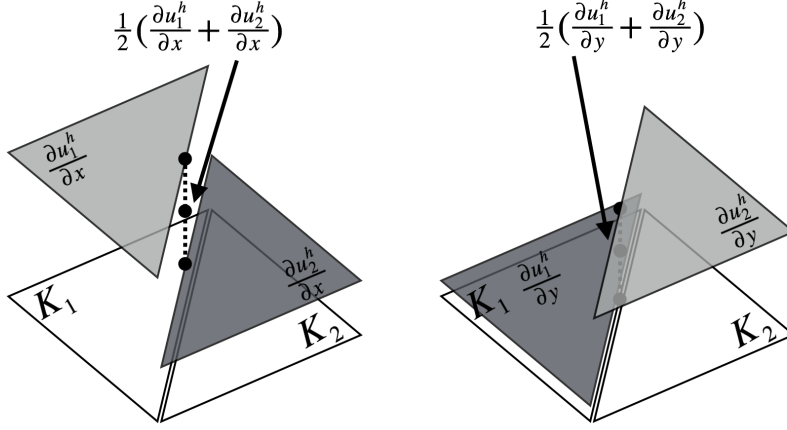


FIGURE 2.11 – Schéma de l'évaluation du saut de gradient à l'interface de deux éléments triangulaires voisins.

obtenons,

$$\kappa^2 = \int_K [\bar{\mathbf{v}} - \nabla u^h]^T [\bar{\mathbf{v}} - \nabla u^h] d\Omega = \int_K \left(\bar{\mathbf{v}}_x - \frac{\partial u^h}{\partial x} \right)^2 + \left(\bar{\mathbf{v}}_y - \frac{\partial u^h}{\partial y} \right)^2 d\Omega$$

où, $\bar{\mathbf{v}}$ est le gradient reconstruit grâce à l'évaluation des sauts dans la direction des x et des y . L'intégrale est ensuite évaluée numériquement par une méthode du troisième ordre : la quadrature de Gauss à quatre points. Puisque ∇u^h est d'ordre 0 et que l'interpolation est linéaire, l'intégrale numérique est exacte.

Comme observé dans le cas unidimensionnel, il y a un manque d'informations pour la reconstruction du gradient concernant les éléments se trouvant sur les frontières. En effet, tous ces éléments possèdent une arête pour laquelle le saut de gradient est évalué à zéro. Ce manque d'informations se répercute dans l'estimateur d'erreur qui annonce une erreur notable dans le coin supérieur gauche et inférieur droit de la figure (Fig. 2.12). De nouveau, nous affichons l'estimateur normalisé par sa valeur maximale sur le domaine et l'erreur exacte en norme H_1 est elle aussi normalisée par sa valeur maximale. De cette manière, nous pouvons mieux jauger de la ressemblance entre l'estimateur et l'erreur exacte. Dans le cas de l'estimateur par reconstruction de gradient, la ressemblance n'est pas aussi flagrante que pour les estimateurs résiduels surtout sur les frontières. Pour ce qui est du reste du domaine, l'estimateur indique bien les régions nécessitant d'être raffinées. Si nous poussons la résolution à 3195 éléments (Fig. 2.13), nous observons le même comportement : une surestimation nette de l'erreur sur les frontières.

2.6 Comparaison des deux estimateurs concernant le problème bidimensionnel

Dans les trois cas étudiés précédemment, c'est-à-dire, l'estimateur résiduel en norme L_2 , celui en norme H_1 et l'estimateur par reconstruction de gradient, ceux-ci indiquaient bien les régions du domaine qui devaient être raffinées. La particularité de

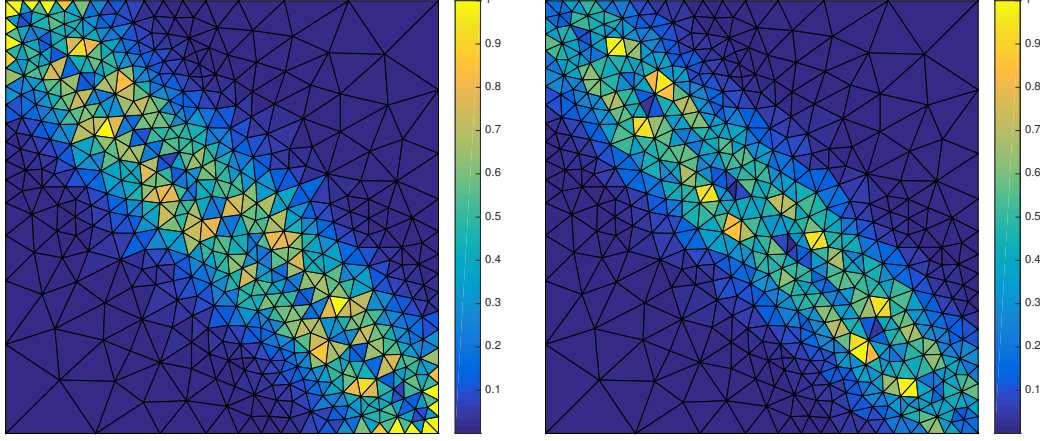


FIGURE 2.12 – Estimateur d’erreur par reconstruction de gradient pour 910 éléments comparé à l’erreur exacte évaluée en norme H_1 à droite.

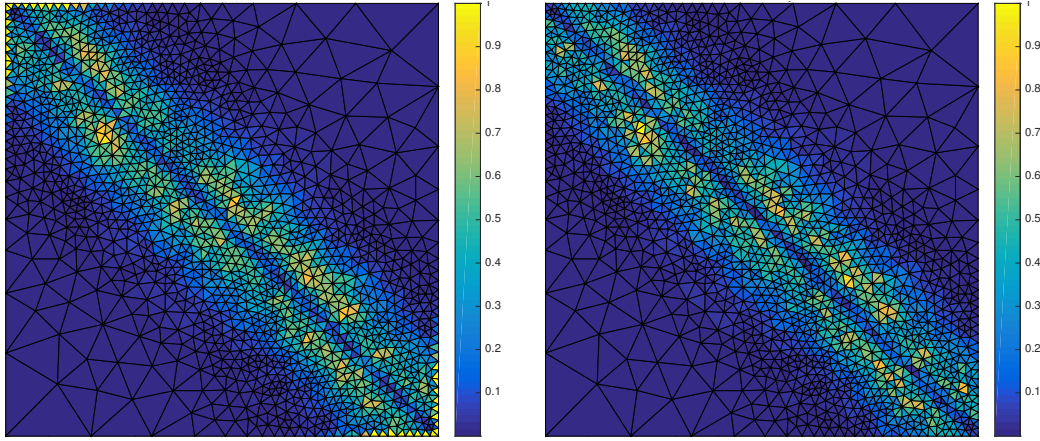


FIGURE 2.13 – Estimateur d’erreur par reconstruction de gradient pour 3195 éléments comparé à l’erreur exacte évaluée en norme H_1 à droite.

la fonction $\arctan$ est de posséder une zone linéaire autour de son point d’inflexion. Les trois estimateurs ont détecté cette zone en affichant une ligne de triangles à faible ratio $\frac{\kappa}{\max(\kappa)}$ sur les graphes (Fig. 2.8), (Fig. 2.10) et (Fig. 2.13). L’avantage de l’estimateur par reconstruction de gradient est sa facilité d’implémentation et ce dans tout code numérique. Il ne tient absolument pas compte des données du problème ni de la forme de celui-ci comparé à un estimateur résiduel qui demande la résolution d’un problème local nécessitant la connaissance des équations partielles différentielles décrivant le problème. Bien que plus compliqué à mettre en place dans un code d’éléments finis existant, il est plus précis sur les frontières du domaine que l’estimateur par reconstruction dans le cadre du problème modèle.

Il n’y a pas de différence marquée entre l’estimateur en norme L_2 et celle en norme H_1 . A priori, les deux normes sont équivalentes avec $H_1 \subset L_2$. En pratique, la norme L_2 est moins chère à évaluer et demande moins de régularité de la part de la solution que la norme H_1 .

Plusieurs pistes d'amélioration seraient à envisager pour l'estimateur par reconstruction de gradient. Premièrement, nous pourrions prendre en considération les conditions aux frontières de Neumann lorsqu'il y en a afin de remplir le manque d'informations concernant les éléments frontières. Deuxièmement, nous pourrions rassembler l'évaluation des sauts de gradient aux noeuds afin d'utiliser les fonctions de forme locales de la méthode des éléments finis et d'obtenir ainsi un gradient lissé globalement continu.

Au sujet de l'estimateur résiduel, nous pourrions ajouter la contribution du résidu frontière. Cet ajout aurait peut-être une incidence sur la précision de l'estimateur.

Au terme de ce chapitre, nous décidons de choisir l'estimateur par reconstruction de gradient bien que moins performant que l'estimateur résiduel, il sera plus simple à implémenter dans le code des écoulements granulaires immergés. La question la plus importante qui sera traitée dans le prochain chapitre sera celle de l'adimensionnalisation des champs de vitesse et de pression.

CHAPITRE 3

ESTIMATEUR D'ERREUR PAR RECONSTRUCTION DE GRADIENT POUR LES ÉCOULEMENTS GRANULAIRES IMMERGÉS

Ce chapitre est dédié au développement d'un estimateur d'erreur par reconstruction de gradient dans le cadre de la résolution d'écoulements granulaires immergés. Dans le chapitre 2, nous avons étudié deux estimateurs d'erreur que nous avons ensuite appliqués à un problème modèle unidimensionnel puis bidimensionnel. Cependant, l'objectif principal de ce mémoire est d'appliquer la théorie des estimateurs d'erreur au cas particulier des écoulements granulaires immergés. Avant de rentrer dans les détails de la construction de l'estimateur, nous examinerons, à la section 3.1, à la modélisation des écoulements granulaires telle qu'implémentée dans le code MigFlow. L'hypothèse fondamentale de cette modélisation est la séparation des échelles. La compréhension de cette hypothèse est nécessaire pour le développement de la stratégie adaptative. Dans cette modélisation, le fluide est résolu à une échelle bien plus grande que celle des grains. Les grains sont quant à eux traités indépendamment par un solveur de contact [Constant et al., 2018]. La force d'interaction entre les grains et le fluide sert d'équation de fermeture pour le modèle et établit ainsi le lien entre les grains et le fluide.

La section 3.2 est basée sur le développement de l'estimateur en vitesse/pression. La théorie sera établie pour des écoulements bidimensionnels mais son extension à des écoulements tridimensionnels n'est pas plus compliquée.

Nous nous pencherons, à la section 3.3, à la stratégie adaptative mise en place dans le cadre des écoulements granulaires immergés dans laquelle nous introduirons une stratégie d'adaptation en avance permettant de mieux capturer l'interface grains-fluide comme dans le cas de la chute d'un nuage de grains dans un fluide visqueux à bas nombre de Reynolds.

Finalement, nous nous intéresserons à la validation des estimateurs pour plusieurs cas tests : la chute d'un nuage de grains dans un liquide visqueux à bas nombre de

Reynolds et la sédimentation de suspensions mono et poly-dispersées en tubes droits ou inclinés. Chacun de ces cas tests a été étudié dans la littérature et possède ses caractéristiques macroscopiques propres comme le taux d’expansion du nuage de grains, la hauteur de la couche de sédiments, etc.

3.1 Modélisation d’un écoulement granulaire immergé

Où trouve-t-on les écoulements granulaires ? Regardez la nature, tournez-vous vers le monde de l’industrie (pharmaceutique, cosmétique et chimique), questionnez-vous sur les procédés utilisés dans le secteur agroalimentaire et intéressez-vous au monde du génie civil. En effet, les écoulements granulaires apparaissent dans de nombreux domaines. Les exemples des avalanches sous-marines, de l’érosion, du transport de sédiments, des glissements de terrain, de la dynamique des dunes, du système sanguin, des mélangeurs industriels n’en sont que quelques-uns parmi d’autres [Hungr, 2001], [Iverson, 2000] et [Masson et al., 2006]. L’intérêt pour leur étude à la fois expérimentale et numérique n’a fait que croître ces dernières décennies. Les phénomènes de fissuration, d’injection, d’infiltration influençant l’agencement des roches qui composent la croûte et le manteau terrestre, peuvent être modélisés par des écoulements granulaires. Ces phénomènes sont analysés en mécanique des sols et en géologie afin d’en étudier l’histoire.

Traditionnellement, les milieux granulaires étaient modélisés par les équations de la mécanique des milieux continus. Cependant, il n’était pas possible avec une telle théorie de représenter chaque grain et son interaction avec les autres grains et le fluide. Petit à petit, avec le développement de nouvelles techniques expérimentales et de méthodes numériques discrètes, une compréhension plus microscopique de ces écoulements fut possible. L’étude des interactions entre les grains et le fluide à l’échelle des grains devient abordable. Il est important de souligner que la viscosité du fluide tout comme son inertie ou encore sa faible compressibilité peuvent influencer la dynamique des grains.

Les deux milieux qui sont aujourd’hui les plus expérimentés en laboratoire et pour lesquels les modèles théoriques sont largement validés sont les suspensions diluées, milieux dominés par les forces hydrodynamiques et les milieux granulaires secs, dominés par les forces de contacts.

Milieu granulaire sec Un milieu granulaire sec ou divisé est composé de grains solides qui interagissent les uns avec les autres par des contacts unilatéraux. Un contact unilatéral est caractérisé par son impénétrabilité. Cette impénétrabilité signifie que les grains candidats pour un contact ne peuvent pas traverser la frontière de leurs homologues [Jean, 1999]. Les propriétés mécaniques qui découlent du caractère discret des milieux granulaires secs sont celles du solide et du liquide. Il s’agit à la fois d’un solide amorphe et d’un liquide non-newtonien. Les verres, les élastomères et les liquides peuvent être considérés comme des composés amorphes car leurs atomes ne

respectent aucun ordre particulier à moyenne et grande distance. Leurs atomes sont dits désordonnés *a contrario* des composés cristallins.

Milieu granulaire immergé Dans le cadre de ce mémoire, nous nous intéressons principalement aux écoulements granulaires immergés. Par immergé, nous entendons la nature biphasique de l'écoulement contenant d'une part une phase continue, le fluide et une phase discrète, les grains. Le comportement du mélange est régi par les interactions grains-grains et les interactions fluide-grains. L'enjeu principal est de comprendre l'effet des grains et de leurs collisions sur la micro-structure des suspensions. Inversement, nous pouvons aussi nous intéresser à l'effet des forces hydrodynamiques sur les assemblages. Ces forces hydrodynamiques sont soit de nature visqueuse soit de nature inertielle. Il en résulte deux régimes : le régime visqueux apparaissant lorsque la force de traînée est donnée par la force de Stokes et le régime inertiel qui lui apparaît lorsque la force de traînée est dominée par l'inertie du fluide.

Afin de prédire l'évolution d'un écoulement granulaire, différents modèles physiques et numériques ont été mis en place. Cependant, ces modèles ont leur limite et leur niveau de précision. Nous les classons selon l'échelle à laquelle l'écoulement est calculé.

- A grande échelle, le milieu est décrit par un modèle non-newtonien dans lequel les grains sont introduits de façon implicite [Rzadkiewicz et Zhu, 1996]. Une autre technique est de rajouter, toujours de manière implicite, les grains par l'intermédiaire d'une force de friction bien choisie [Gidaspow, 2012]. Cette force joue alors le rôle d'équation de fermeture dans les équations de la quantité de mouvement.
- A petite échelle, les deux phases sont considérées de façon distincte et les méthodes utilisant cette représentation sont appelées méthodes directes. Les mouvements du fluide sont capturés par une approche eulérienne alors que ceux des grains, sont quant à eux résolus de façon lagrangienne.

La trajectoire des grains dans un milieu granulaire sec est capturée de façon lagrangienne par les méthodes aux éléments discrets (Discret Elements Method ou DEM). Cette méthode ayant fait ses preuves [Radjaï, 2011] fut initialement introduite par Cundall et Strack en 1979. Pour être précis, il existe trois types de méthode aux éléments discrets.

1. La méthode des collisions [McNamara, 1992] résout les contacts de façon séquentielle mais elle ne peut s'attaquer à des milieux granulaires denses qui produisent un nombre considérable de chocs par pas de temps.
2. La méthode des sphères molles [Cundall et Strack, 1979] suppose des grains indéformables mais elle autorise cependant l'interpénétration de grains, celle-ci est ensuite utilisée pour la formulation des forces de contacts normales et tangentielles.
3. La méthode de dynamique des contacts [Jean, 1999] ou aussi appelée méthode des sphères rigides interdit l'interpénétration des grains. Les forces de contacts

sont alors obtenues par une résolution implicite des équations du mouvement des grains.

Puisque nous nous intéressons aux écoulements granulaires immergés, il faut également tenir compte du fluide et le résoudre autour des grains en mouvement. Soit nous considérons un maillage plus fin que le diamètre moyen des grains épousant leur forme et permettant de résoudre les équations de Navier-Stokes. Soit nous considérons des méthodes à maillages cartésiens fixes permettant de résoudre les équations de Navier-Stokes modifiées par la présence d'objets solides. Nous pouvons citer : les méthodes dites *Force Coupling Method* [Dance et Maxey, 2003], les méthodes dites *Distributed Lagrangien Multiplier* [Ardekani et Rangel, 2008], les méthodes dites *Tensorial Penalty* [Brändle de Motta, 2013] et les méthodes des frontières immergées [Peskin, 2000].

Le modèle développé dans le code MigFlow est celui d'un modèle hybride multi-échelles permettant de travailler sur deux échelles, une petite échelle pour la résolution des mouvements et des contacts des grains et une échelle plus grande pour résoudre le fluide. Cette méthode couple les méthodes aux éléments discrets avec les méthodes à plus grande échelle basées sur la mécanique des milieux continus.

La méthode se base sur l'hypothèse de séparation des échelles. D'une part, l'écoulement du fluide entre les grains est modélisé par un mélange homogène à une plus grande échelle que les grains. Les équations de Navier-Stokes sont moyennées sur des sous-ensembles élémentaires du domaine qui sont constitués d'un volume particulaire ou solide Ω_p et d'un volume fluide Ω_f . Par l'hypothèse de séparation des échelles, il est important que le volume des grains soit négligeable par rapport au volume du sous-ensemble élémentaire. Sur un maillage en deux dimensions, ces sous-ensembles sont les éléments triangulaires ou quadrilatères et le diamètre moyen de grains se doit d'être beaucoup plus petit que la taille h des éléments. Dans la stratégie adaptative 3.3, nous devons stipuler la taille minimale octroyée aux éléments afin que la simulation ne perde pas tout son sens. Nous pouvons définir la fraction volumique du fluide dans un sous-ensemble élémentaire et cette fraction est appelée porosité et notée ϕ , à ne pas confondre avec les fonctions de forme locales.

$$\phi = \frac{\Omega_f}{\Omega}$$

Finalement, pour un fluide de densité ρ , les lois de conservations pour la phase fluide sont,

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \frac{\mathbf{u}\mathbf{u}}{\phi} \right) = \nabla \cdot [2\mu\phi \mathbf{d}(\mathbf{u}) - p\mathbf{I}] + \mathbf{f} + \phi\rho\mathbf{g} \quad (3.1)$$

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \mathbf{u} = 0, \quad (3.2)$$

où $\mathbf{g}$ est la gravité, μ est la viscosité du fluide, $\mathbf{d}(\mathbf{u})$ est le tenseur des taux des déformations, $\mathbf{u} = \phi\mathbf{w}$, $\mathbf{w}$ est la vitesse du fluide et $\mathbf{f}$ la force d'interaction entre les grains présents dans le volume représentatif et le fluide. Cette force n'est pas anodine

et se doit d'être modélisée afin de fermer le modèle. Cette force s'exerçant entre le grain i et le fluide à l'échelle des grains est modélisée par,

$$\mathbf{f}_i = -V_{p,i}\nabla\langle p\rangle_f - \mathbf{D}_i + \mathbf{g}V_{p,i}(\rho_{p,i} - \rho),$$

où $\mathbf{D}_i$ est la force de traînée, $V_{p,i}$ est le volume occupé par le grain i et $\rho_{p,i}$ est sa densité. La force de traînée est influencée par deux grands paramètres, la vitesse relative entre les grains et le fluide et la géométrie des grains. Nous utiliserons donc la modélisation de Di Felice et Rotondi [Di Felice et Rotondi, 2012] qui s'écrit comme,

$$\mathbf{D}_i = \underbrace{f(\phi)}_{=\phi^{-1.8}} C_d \left(\underbrace{Re_{p,i}}_{=2r\rho|\mathbf{v}_{rel}|\phi\mu^{-1}} \right) \frac{A_{p,i}\rho}{2} |\mathbf{v}_{rel}| \mathbf{v}_{rel},$$

où $\mathbf{v}_{rel} = \mathbf{v}_{p,i} - \langle \mathbf{v} \rangle_f$ est la vitesse relative entre le fluide et les grains, $A_{p,i}$ est l'aire du grain i projetée sur le plan perpendiculaire au mouvement du grain et $Re_{p,i}$ est le nombre de Reynolds du grain. Pour tenir compte de l'impact de la concentration en grains dans le fluide sur la force de traînée ressentie par un seul grain, nous utilisons une fonction notée $f(\phi)$ qui fut modélisée par Wen et Yu [Wen et Yu, 1966]. Il ne reste plus qu'à fournir une expression pour le coefficient de traînée. Plusieurs possibilités s'offrent à nous. Cependant, dans le code MigFlow, il a été décidé de choisir la modélisation proposée par Dallavalle et al [DallaValle, 1943],

$$C_d Re_{p,i} = \left(0.63 + \frac{4.8}{Re_{p,i}^{0.5}} \right)^2.$$

Ces équations concernent essentiellement le fluide. Nous devons aussi nous tourner vers la résolution des contacts à l'échelle des grains. Ces contacts sont résolus grâce à la méthode aux éléments discrets pour des sphères rigides qui porte, comme nous l'avons précédemment vu, le nom de méthode de dynamique des contacts. La loi de contact est donc inélastique. A chaque pas de temps, la vitesse et la position de chaque grain sont alors mises à jour par le biais de la seconde loi de Newton et ce sans tenir compte des éventuels contacts qui auraient pu survenir. Il est nécessaire de calculer la vitesse et la pression du fluide, puis de calculer la force d'interaction $\mathbf{f}$ (Eq. 3.1), et finalement d'interpoler la force depuis les noeuds du maillage sur les grains grâce aux fonctions de forme linéaires lagrangiennes associées à la matrice de masse. Avant de passer au pas de temps suivant, il est impératif de résoudre *a posteriori* tous les contacts négligés au cours du pas de temps grâce à une méthode itérative. En choisissant un pas de temps suffisamment petit, il n'est pas nécessaire de vérifier toutes les paires de grains, seuls les grains se trouvant dans un certain périmètre sont vérifiés.

3.2 Estimateur d'erreur par reconstruction de gradient

Dans cette section, l'estimateur par reconstruction de gradient en vitesse et pression est examiné. Nous allons, dans un premier temps, reprendre la méthodologie appliquée à la section 2.5 pour obtenir l'estimateur. Il faudra cependant considérer à la

fois une fonction scalaire, le champ de pression exprimé en pascals Pa et une fonction vectorielle, le champ de vitesse exprimé en mètres par seconde m/s . Il est primordial d'adimensionnaliser la pression et la vitesse grâce aux grandeurs caractéristiques de l'écoulement. Nous nous intéresserons, dans un second temps, à l'implémentation de cet estimateur dans le code MigFlow.

Pour construire l'estimateur, nous allons suivre les différentes étapes schématisées à la figure (Fig. 3.1) afin d'arriver à l'étape de la stratégie adaptative : l'obtention du nouveau maillage avec GMSH. La première étape, la résolution de l'écoulement, est réalisée par le logiciel MigFlow. L'évaluation des sauts de gradient s'effectue de la même manière qu'expliquée à la section 2.5. Le gradient de l'approximation en éléments finis se calcule à partir des trois fonctions de forme locales données par $\phi_1 = 1 - \xi - \eta$, $\phi_2 = \xi$ et $\phi_3 = \eta$ car les sauts de gradient sont évalués au milieu de chaque arête. En supposant que les valeurs nodales du champ de pression sur l'élément K sont données par P_1 , P_2 et P_3 , nous pouvons calculer l'expression du gradient du champ de pression sur chaque élément en tenant compte du changement de variable donné par l'expression (Eq. 2.4) comme,

$$\begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix} = \begin{bmatrix} P_2 - P_1 \\ P_3 - P_1 \end{bmatrix}^T \frac{1}{J} \begin{bmatrix} Y_3 - Y_1 & X_1 - X_3 \\ Y_1 - Y_2 & X_2 - X_1 \end{bmatrix},$$

où J est le Jacobien de la transformation et vaut, dans le cas présent, $(X_2 - X_1)(Y_3 - Y_1) - (X_3 - X_1)(Y_2 - Y_1)$. Le même genre de résultats est obtenu pour le gradient du champ de vitesse. La vitesse sera décomposée en sa composante verticale v et horizontale u , c'est-à-dire sa composante alignée avec l'axe des ordonnées et avec l'axe des abscisses du système de référence. C'est ensuite sur chacune de ces composantes que nous calculons le gradient. La partie la plus délicate de cette étape sera de trouver grâce aux structures de données du maillage les trois voisins de chaque élément puis ensuite de cibler l'arête commune de l'élément et de chacun de ses voisins.

L'adimensionnalisation du champ de vitesse va dépendre de l'écoulement considéré. En effet, chaque écoulement possède ses propres grandeurs caractéristiques comme la vitesse de sédimentation pour une suspension mono ou poly-dispersée dans un tube droit. Le choix de la vitesse caractéristique de l'écoulement U_0 sera plus amplement discuté dans la section 3.4. Quant au champ de pression, nous pourrions l'adimensionnaliser grâce à la pression dynamique de l'écoulement $\frac{1}{2}\rho U^2$, où ρ est la densité du fluide. Dans le cas d'un écoulement de Stokes, nous préférons la pression basée sur la viscosité et définie comme $P_0 = \mu/T_c$ où T_c est le temps caractéristique de l'écoulement. La cinquième étape de cet organigramme consiste à intégrer l'erreur commise sur le gradient du champ de vitesse $G^h(\mathbf{u}^h) - \nabla \mathbf{u}^h$ et sur le gradient du champ de pression $G^h(p^h) - \nabla p^h$, où $G^h(\cdot)$ est l'opérateur de reconstruction de gradient. L'estimateur est fourni par l'expression (Eq. 3.3),

$$\kappa^2 = \frac{\|G^h(\mathbf{u}^h) - \nabla \mathbf{u}^h\|^2}{U_0^2} + \frac{\|G^h(p^h) - \nabla p^h\|^2}{P_0^2}. \quad (3.3)$$

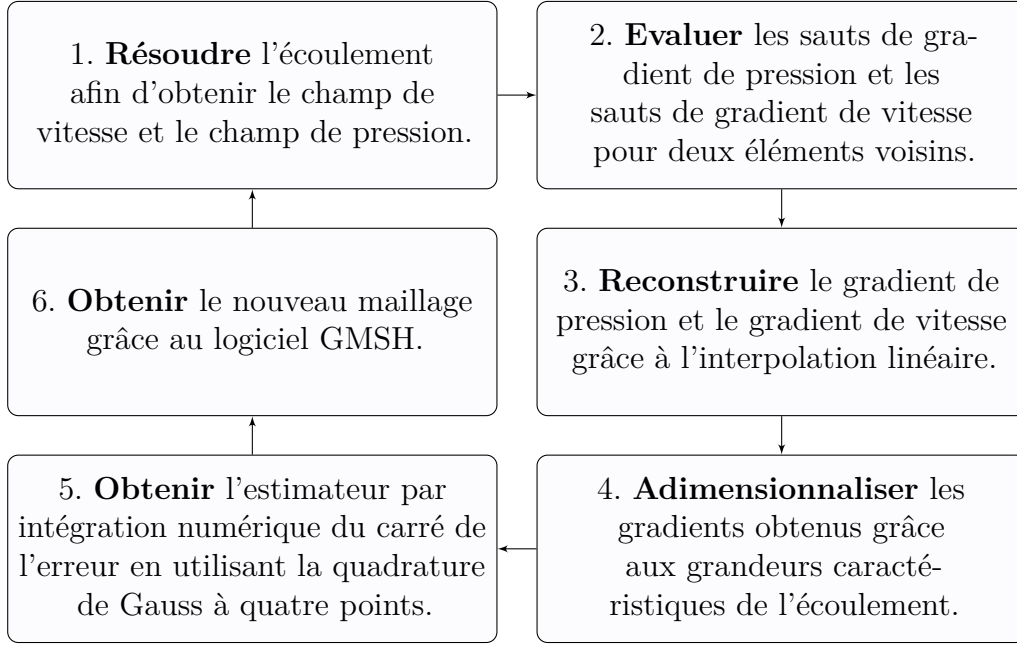


FIGURE 3.1 – Les différentes étapes à suivre pour l'obtention de l'estimateur par reconstruction du gradient de vitesse et de pression.

3.3 Stratégie adaptative

La stratégie adaptative servant à obtenir la nouvelle carte de tailles est similaire à la stratégie développée à la section 2.4. Pour chaque élément du maillage, nous devons toujours fournir à l'algorithme : son estimateur κ , l'erreur prescrite e^* , une valeur minimale $h_{\min}$ et maximale $h_{\max}$ pour l'élément. Comme émis précédemment lors de la modélisation de l'écoulement granulaire, la séparation des échelles est une hypothèse fondamentale. Le volume occupé par les grains dans le sous-ensemble élémentaire doit être négligeable. Cette hypothèse peut se traduire en termes de taille de maille : la taille h d'un élément ne peut être plus petite que le diamètre moyen des grains.

Le problème ne dépend plus uniquement de l'espace mais aussi du temps, c'est la grande différence avec le chapitre 2. Dans un premier temps, nous voulons adapter le maillage à chaque itération. Cependant, cette approche est bien trop coûteuse. Nous en concluons que l'opération d'adaptation doit s'effectuer à intervalle régulier Δt . Cependant au fur et à mesure que la simulation avance, le maillage est de moins en moins bon car il ne correspond plus à la dynamique actuelle de l'écoulement. Une façon de palier à ce problème est de résoudre l'écoulement sur un certain intervalle de temps Δt , d'adapter le maillage en conséquence et de revenir sur nos pas comme décrit à la figure (Fig. 3.2).

A chaque pas de temps, nous calculons l'estimateur sur chaque élément et nous appliquons la stratégie de raffinement sans raffiner le maillage. En somme, nous prenons le minimum de toutes les cartes de tailles générées de $n\Delta t$ à $(n+1)\Delta t$. A la fin de l'intervalle de temps, nous adaptons le maillage par le biais du logiciel **GMSH**

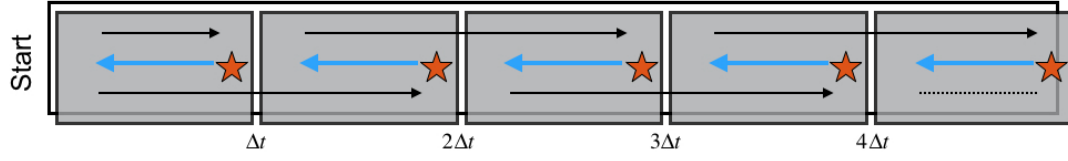


FIGURE 3.2 – Schéma itératif lié à la stratégie adaptative en avance où l'étoile représente l'adaptation du maillage et la flèche bleue le redémarrage Δt en arrière.

symbolisé par l'étoile orange dans la figure (Fig. 3.2). Avec ce nouveau maillage, nous revenons Δt temps en arrière et nous recommençons la simulation mais avec un maillage qui sera adapté à tous les pas de temps.

3.4 Validation de l'estimateur pour plusieurs cas tests

Cette section est dédiée à la validation de notre estimateur d'erreur pour plusieurs cas tests qui ont été intensivement étudiés dans la littérature. Nous nous intéresserons d'une part à la chute d'un nuage de grains dans un fluide visqueux à bas nombre de Reynolds, et d'autre part, à la sédimentation de suspensions mono et poly-dispersées en tubes droits ou inclinés. Le phénomène de sédimentation d'une suspension dans un tube incliné s'appelle l'effet Boycott, du nom du scientifique Arthur Boycott ayant observé ce phénomène pour la première fois en 1920. Il nota que la sédimentation de globules rouges dans un récipient incliné était plus rapide que dans un tube droit. Les grains s'accumulent rapidement contre la paroi inférieure du tube. Cette accélération du phénomène de sédimentation est due à l'apparition d'un flux de contre-écoulement le long de la paroi opposée (paroi supérieure du tube). Le fluide délesté de ses grains et donc plus léger continue son ascension le long de cette paroi et ne gêne dès lors plus la sédimentation.

Nous allons étudier ces deux écoulements granulaires en définissant leur vitesse U_0 et pression P_0 caractéristiques mais aussi les grandeurs macroscopiques les particularisant. L'idée est de comparer une simulation réalisée sur un maillage uniforme et une simulation réalisée avec la stratégie adaptative, adaptation globale du maillage. De cette manière, nous pourrions juger de la qualité de la simulation avec et sans l'adaptation du maillage.

3.4.1 Chute d'un nuage de grains dans un fluide visqueux

Quand les grains sont très petits ou bien que le fluide est très visqueux, l'interaction entre les grains est régie par les forces hydrodynamiques. Il en découle que les forces de surface comme les forces de Van der Waals ou encore le mouvement brownien sont négligeables. Les interactions hydrodynamiques engendrent un déplacement complexe et chaotique des grains bien que les équations de Stokes soient réversibles. L'évolution temporelle d'une sphère de grains tombant dans un fluide visqueux sous l'effet de la gravité est instable.

En effet, il a été observé que la sphère initiale évolue en une forme de tore qui s'éclate ensuite en gouttelettes secondaires se déformant à leur tour. Dans sa chute, le nuage perd quelques grains. Ces grains ont tendance à s'échapper en formant une traînée verticale à l'arrière du nuage suite aux fluctuations de vitesse à l'interface entre les grains et le fluide. Ces grains sont alors ralentis car ils ne profitent plus de l'entraînement du nuage. Cette perte de grains crée un déficit au centre du nuage contribuant à la formation du tore. Au cours du temps ce tore va encore s'étendre et nous pouvons montrer que le taux de fuite est influencé par le changement de forme du nuage au cours du temps.

Le présent système est instable comme mentionné par [Nitsche et Batchelor, 1997] car il est très sensible à une simple modification de la condition initiale ou même aux erreurs numériques qui peuvent perturber la position des grains. L'étude de la convergence de la trajectoire des grains n'a donc aucun sens et nous préférons porter notre attention sur la convergence de grandeurs macroscopiques de l'écoulement qui ont pu être mesurées expérimentalement et numériquement. Dans ce cadre, nous nous sommes intéressés au calcul de la vitesse de chute d'une sphère de grains et de la position de son centre de masse. Ce sont [Adachi, Kiriya, et Toshioka, 1978] qui ont calculé et mesuré cette vitesse de chute pour un nuage de grains initialement sphérique.

Pour cette étude, nous nous sommes inspirés des travaux de [Metzger, et al, 2006] qui ont étudié l'évolution temporelle à la fois numérique et expérimentale d'un nuage de grains tridimensionnel tombant dans un fluide visqueux. La seule différence avec nos simulations est que nous travaillons en deux dimensions alors que leurs expériences sont en trois dimensions. Néanmoins, une certaine comparaison est toujours possible. Comme le montre [Constant et al., 2017] dans ses travaux, la longueur de la traînée de grains est plus faible en deux dimensions qu'en trois, ce qui s'explique par le nombre plus faible de grains présents en deux dimensions. Constant a pu montrer que la simulation numérique en deux dimensions fournissait les mêmes résultats que [Machu et al., 2001].

La première étape est de fournir à l'estimateur la grandeur caractéristique de vitesse afin de le rendre sans dimension et donc de travailler avec un estimateur en vitesse/pression comme expliqué à la section 3.2. Nous choisissons d'adimensionnaliser les longueurs et les vitesses par le rayon initial du nuage R_0 et par la vitesse $U_0 = \frac{FN_0}{5\pi\mu R_0}$. La force F est alors donnée par $V_{p,i}(\rho_p - \rho)g$ où $V_{p,i}$ est le volume occupé par le grain i . Cette expression provient de la force d'interaction (Eq. 3.1) où initialement, la force de traînée est nulle et le champ de pression est supposé uniforme. Vu que le nombre de Reynolds est particulièrement faible, largement inférieur à 1, l'écoulement est un écoulement de Stokes. La pression caractéristique est donc liée aux effets visqueux et définie comme $P_0 = \frac{\mu}{T_c}$ où le temps caractéristique T_c est pris comme le ratio entre le rayon initial du nuage et la vitesse caractéristique.

La deuxième étape est de définir les grandeurs macroscopiques particularisant l'écou-

lement. Avant tout, nous devons formellement définir si le grain i appartient au nuage à un temps t^* . Les grains appartenant au nuage sont ceux qui ont une position verticale par rapport au centre de masse du nuage plus petite que R_0 . Il est à noter que la position du centre de masse va être modifiée car des grains quittent le nuage en formant une traînée et le nuage en lui-même est en mouvement. La position du centre de masse est définie par,

$$\mathbf{X}_c(t^*) = \frac{1}{M(t^*)} \sum_{i=1}^{N(t^*)} m_i \mathbf{x}_i(t^*),$$

où $M(t^*)$ est la masse totale du nuage au temps (t^*) , $N(t^*)$ est le nombre de grains effectivement présents dans le nuage, m_i et $\mathbf{x}_i$ sont la masse et la position du i ème grain appartenant au nuage au temps (t^*) , respectivement. En moyennant les vitesses individuelles de chacune des $N(t^*)$ grains, nous obtenons la vitesse du nuage $V(t^*)$.

Validation des grandeurs caractéristiques

Le fluide considéré pour cet écoulement est un mélange à 50% d'eau distillée et à 50% d'huile UCON qui a la propriété d'être une huile soluble. Quant aux grains, il s'agit des perles sphériques de verre.

Ces données ont été récupérées de l'article [Metzger, et al, 2006]. L'ensemble des données physiques et géométriques du problème est résumé au tableau (Tab. 3.1).

Afin de valider la vitesse caractéristique avec le champ de vitesse du fluide, nous avons calculé une solution de référence sur un maillage raffiné comptant environ 50700 éléments en prenant comme taille de maille dix fois le rayon des grains. Sur les 800 premières itérations, avec un pas de temps de $5 \cdot 10^{-3}$, nous observons que le maximum de vitesse se stabilise autour de $12 \cdot 10^{-4}$ m/s et le champ de pression duquel nous avons pris le soin de retirer la pression hydrostatique se stabilise autour de la valeur 0.27 Pa. Au niveau de la vitesse caractéristique, nous sommes un ordre de grandeur en dessous de la vitesse maximale et pour ce qui est de la pression caractéristique, l'ordre de grandeur est le même. Nous validons ces deux grandeurs caractéristiques.

Validation de la stratégie adaptative

L'objectif de la stratégie adaptative est de garder un temps de calcul modéré tout en ayant une solution précise comme celle obtenue sur un maillage uniformément raffiné. Pour le cas du nuage de grains chutant dans un fluide visqueux, nous avons calculé la solution sur trois maillages différents : un maillage raffiné et uniforme, un maillage grossier et un maillage adapté. Sur ces trois maillages, nommés A, B et C, nous allons comparer le temps de calcul pour compléter le même nombre d'itérations, la position finale du centre de masse et la vitesse de chute du nuage.

Nous remarquons qu'avec un maillage B le temps de calcul est bien moindre que celui pour le maillage A qui affiche un nombre colossal de mailles, comme indiqué dans le tableau (Tab. 3.2). Notons cependant que le temps de simulation pour un maillage C

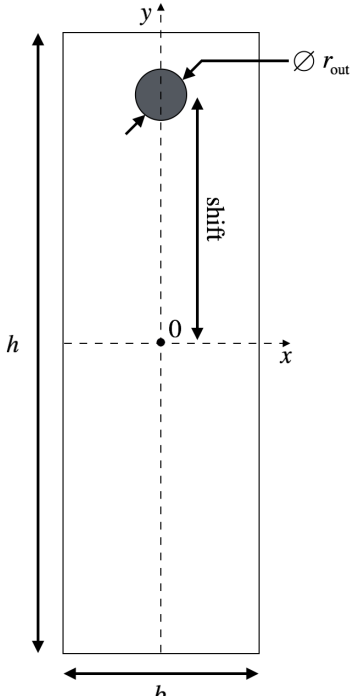
Géométrie du problème	Données
	Concernant le fluide, $\rho = 1030 \text{ [kg/m}^3\text{]}$ $\mu = 1.17 \text{ [Pa s]}$ Concernant les grains, $\rho_p = 2450 \text{ [kg/m}^3\text{]}$ $r_p = 154 \cdot 10^{-6} \text{ [m]}$ $\phi = 0.25 \text{ [-]}$ $N_0 = 101 \text{ [-]}$ Concernant la géométrie, $h = 0.6 \text{ [m]}$ $b = 0.1 \text{ [m]}$ $R_0 = 3.3 \cdot 10^{-3} \text{ [m]}$ La vitesse caractéristique vaut donc, $U_0 = 3.55 \cdot 10^{-4} \text{ [m/s]}$ La pression basée sur la viscosité vaut, $P_0 = 0.126 \text{ , [Pa]}$ Le nombre de Reynolds basé sur les grains vaut, $Re_p = 1.031 \cdot 10^{-3} \ll 1$

TABLE 3.1 – Données géométriques et physiques d’un nuage de grains chutant dans un fluide visqueux.

	$h_{\min} \text{ [m]}$	Nombre d’éléments	CPU	$Y_c(t^* = 20)$
Maillage fin (A)	$10r_p$	50700	96 min 36 s	-0.135
Maillage grossier (B)	$20r_p$	12872	14 min 22 s	-0.138
Maillage adapté (C)	$10r_p$	$3978 < n_{elt} < 5855$	20 min 7 s	-0.134

TABLE 3.2 – Le maillage adaptatif nous permet de réduire le temps de la simulation tout en ayant une précision de l’ordre de $\mathcal{O}(10^{-3})$.

est plus élevé que celui du maillage B bien qu’ayant un nombre d’éléments moindre. L’explication est simple. Dans la stratégie adaptative, nous effectuons un certain nombre d’itérations puis nous revenons en arrière, comme expliqué à la section 3.3. En réalité, nous effectuons le double d’itérations avec cette stratégie. Pour un $\Delta t = 5$, $t_{end} = 20$ et $dt = 5 \cdot 10^{-3}$, 4000 itérations sont effectuées pour les cas sans adaptation et 8000 pour le cas adapté.

Il reste encore à vérifier si le maillage C nous permet d’obtenir une solution qui se rapproche en terme de précision de la solution sur le maillage A en analysant

l'évolution de la vitesse de chute du nuage. Afin de valider ce critère, nous affichons l'évolution de la vitesse de chute du nuage V^* obtenue sur les trois maillages, (Fig. 3.3). Dans le maillage B, la vitesse de chute est plus élevée que les deux autres. Avec des éléments plus grands, le phénomène de sur-pression est plus faible et les grains chutent plus rapidement. Un peu avant $t^*/t_{end} = 0.7$, la courbe bleue intersecte la rouge et la noire indiquant que le nuage sur le maillage B chute dès lors moins vite. La vitesse de chute dans le maillage B présente aussi de plus larges oscillations indiquant que la physique est moins bien capturée que dans les deux autres cas. En ce qui concerne la comparaison entre la vitesse de chute dans le maillage A et le maillage C, nous observons la même amplitude des oscillations qui est bien plus petite que celle de la courbe bleue. Les deux courbes sont très proches l'une de l'autre et s'intersectent un peu avant $t^*/t_{end} = 1$.

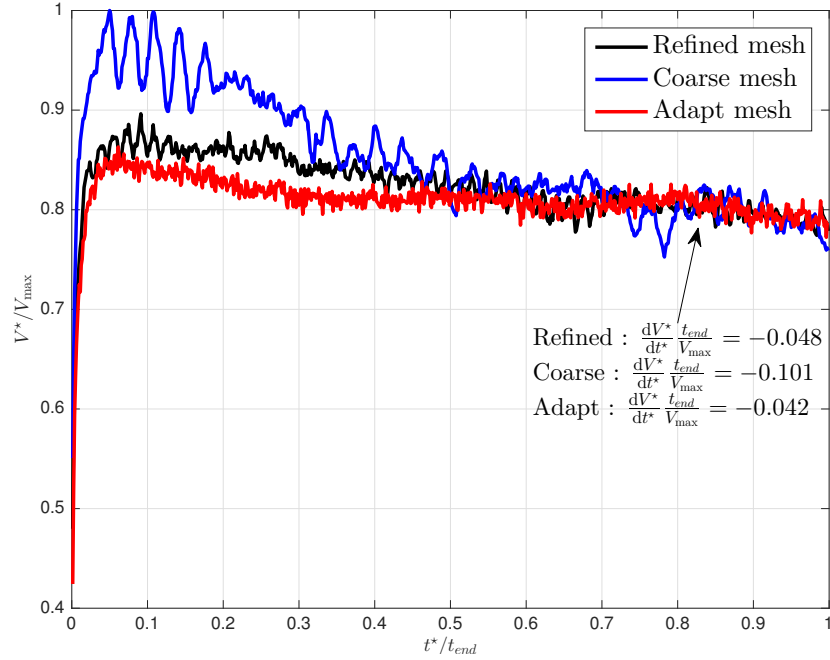


FIGURE 3.3 – Evolution de la vitesse de chute observée sur les trois maillages avec une convergence de la vitesse de chute pour le maillage adapté et le maillage raffiné.

Afin d'expliquer la raison de l'intersection entre les courbes, regardons l'évolution de la chute du nuage en affichant le nuage au complet avec son centre de gravité pour 5 temps distincts : 0, 5, 10, 15 et 20, voir la figure (Fig. 3.4). Nous remarquons que le nuage calculé sur le maillage B subit une plus forte expansion que les deux autres nuages. Lorsque les grains sont plus écartés, c'est-à-dire que sur un volume donné la porosité diminue, le nuage est freiné. Les grains ne s'entraînent plus autant dans leur chute et la vitesse de chute du nuage diminue. Il y a plus de grains dans les deux autres nuages, ils sont plus compacts ce qui explique que leur vitesse de chute reste plus élevée. Sur ces observations, nous pouvons conclure que la stratégie adaptative est plus rapide en temps de calcul et donne d'aussi bons résultats que la solution sur maillage raffiné.

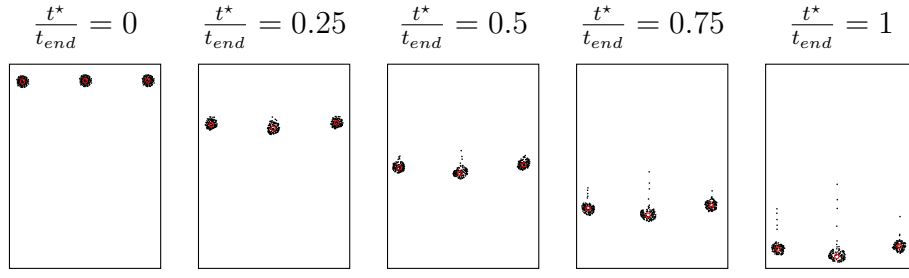


FIGURE 3.4 – Evolution de la chute des nuages pour les trois simulations, dans chaque boîte, sur la gauche, il s'agit du nuage évoluant dans le maillage A, au milieu, du nuage évoluant sur le maillage B et à droite, du nuage évoluant sur le maillage C.

Evolution du maillage pour l'estimateur en vitesse/pression

Un élément assez caractéristique du nuage de grains est cette traînée qu'il laisse derrière lui et qui s'allonge de plus en plus car les grains qui s'y trouvent sont ralentis par la viscosité du fluide et ne sont plus entraînés par le nuage. Afin de valider une fois de plus notre estimateur, nous voudrions observer si le maillage se raffine en la présence de la colonne de grains. Pour cette simulation, nous avons été jusqu'à $t_{end} = 100$ ce qui nous a permis de voir une belle et grande colonne de grains à vitesse réduite apparaissant petit à petit derrière le nuage.

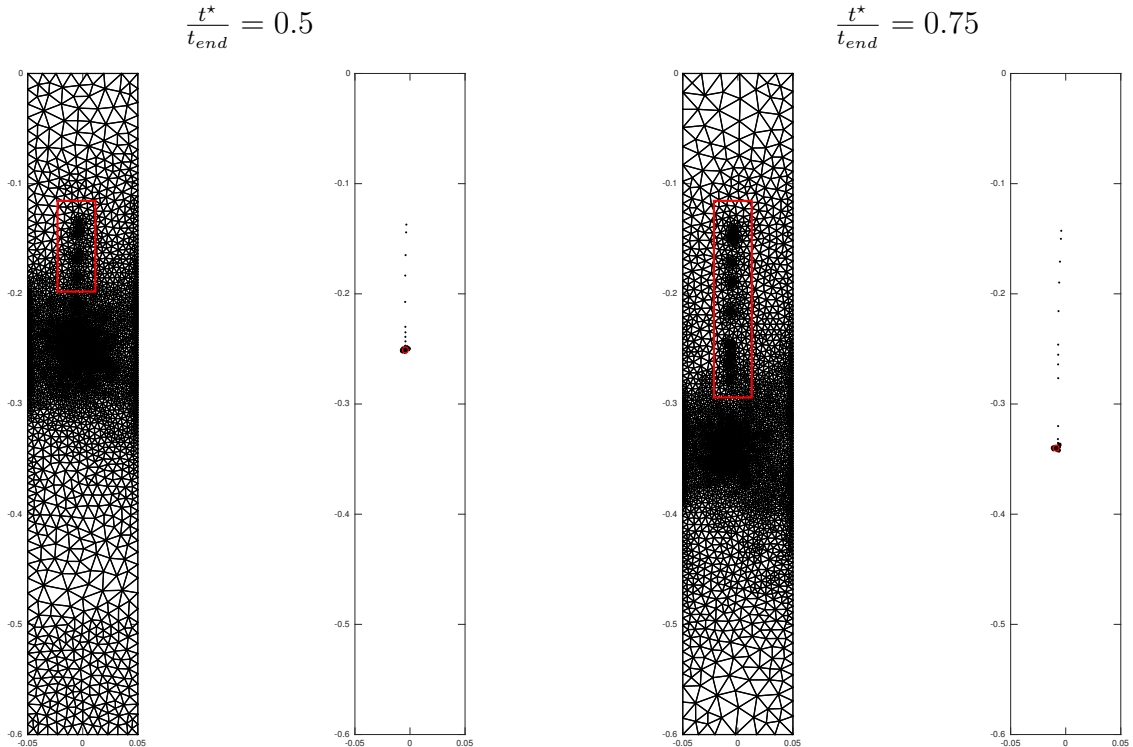


FIGURE 3.5 – Observation du maillage adapté à deux temps distincts comparé à la position du nuage de grains, notons que le maillage raffine bien la colonne de grains laissée par le nuage en chutant.

En utilisant l'estimateur en vitesse/pression, le maillage se raffine correctement le long de la traînée de grains comme indiqué dans la figure (Fig. 3.5). Le raffinement est tel qu'il en devient presque possible de compter le nombre de grains présents dans la colonne juste en observant le maillage. Notons également que le maillage est bien déraffiné à l'avant du nuage et à l'arrière montrant que le maillage est bien ciblé autour du mouvement des grains. Etant donné que le fluide est très visqueux et que les grains sont assez petits, ceux-ci vont générer un champ de pression et de vitesse bien localisés comme indiqué dans les figures (Fig. 3.6) et (Fig. 3.7), respectivement.

Nous pouvons nous demander si l'estimateur basé uniquement sur la pression ou l'estimateur basé uniquement sur la vitesse est capable de capturer la colonne de grains qui se forme à l'arrière du nuage. D'un côté, nous avons adapté le maillage uniquement sur le champ de pression (Fig. 3.6). Le champ de pression entoure principalement le nuage de grains tombant. Cependant, nous remarquons que le champ de pression possède de légères variations le long de la colonne. Ces variations sont capturées par l'estimateur se traduisant en un raffinement local de la traînée de grains comme indiqué à la figure (Fig. 3.8) sur le graphe de gauche.

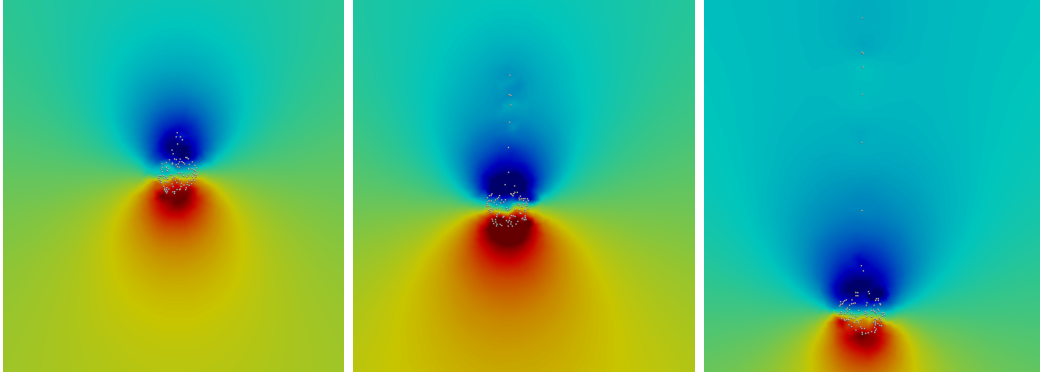


FIGURE 3.6 – Observation du champ de pression aux temps $\frac{t^*}{t_{end}} = 0.025$, $\frac{t^*}{t_{end}} = 0.05$ et $\frac{t^*}{t_{end}} = 0.075$.

D'un autre côté, nous avons adapté le maillage sur le champ de vitesse. Ce champ de vitesse est localisé autour des grains mais ne suit pas la colonne de grains, (Fig. 3.7). Nous ne le voyons pas sur ces graphes mais le champ de vitesse part du centre du nuage et rejoint les bords en dessinant une forme d'ellipse. Le champ de vitesse ne permet pas de capturer les grains un à un dans la colonne. Cependant, l'estimateur raffine sur une bien plus grande partie du domaine en englobant toute la colonne de grains comme indiqué à la figure (Fig. 3.8). Nous avons noté précédemment que les grains qui sont éjectés du nuage commencent à ralentir peu à peu car ils ne sont plus entraînés par les autres grains. Dans cette configuration, l'estimateur en vitesse sera incapable de capturer ces grains qui ont une vitesse pratiquement nulle. C'est pour cette raison que l'estimateur doit aussi tenir compte de la pression pour fournir une adaptation valable dans toutes les configurations.

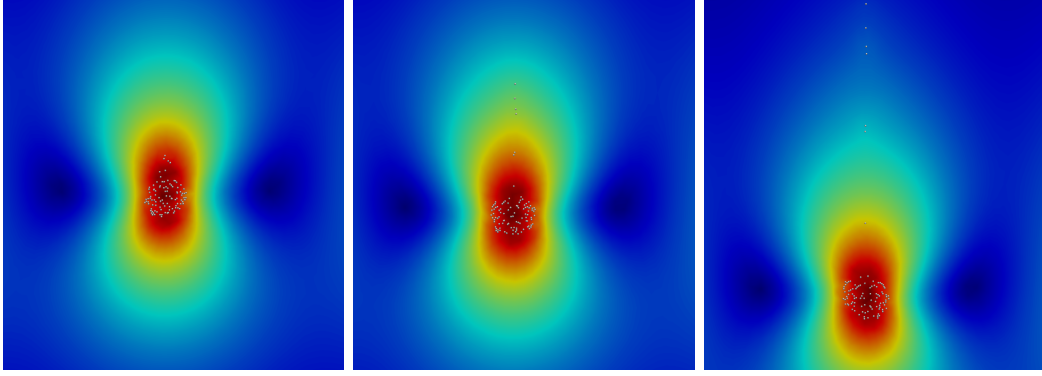


FIGURE 3.7 – Observation de la norme du champ de vitesse aux temps $\frac{t^*}{t_{end}} = 0.025$, $t^* = 0.05$ et $t^* = 0.075$.

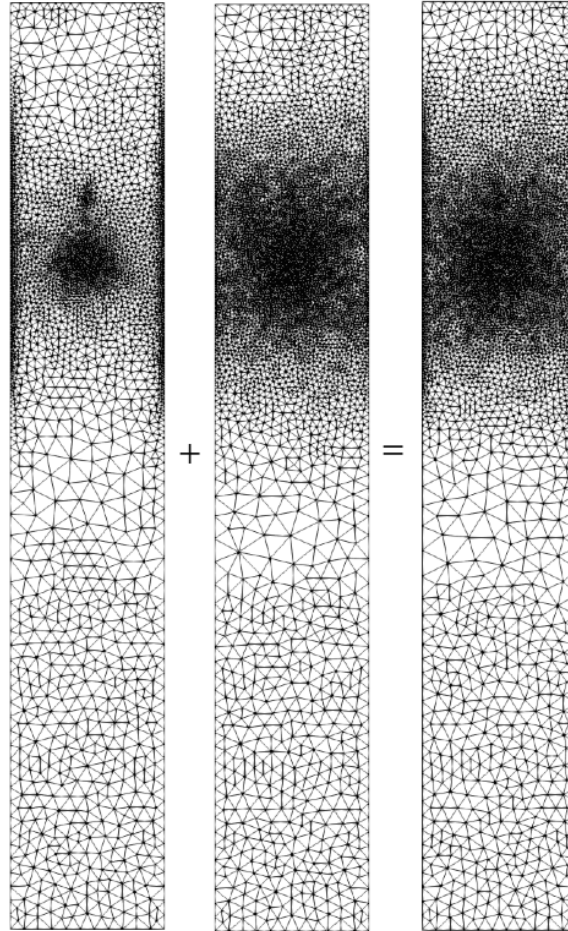


FIGURE 3.8 – La combinaison d'un maillage adapté en pression (à gauche) avec un maillage en vitesse (au centre) fournit un maillage adapté en vitesse et pression (à droite) au temps $\frac{t^*}{t_{end}} = 0.075$.

3.4.2 Sédimentation de suspensions mono-dispersées

La sédimentation est le processus par lequel les grains vont se séparer progressivement du fluide sous l'action de la gravité. Cette technique est depuis longtemps utilisée

dans l'industrie pétrolière, pharmaceutique, d'exploitations minière et chimique pour séparer les grains du fluide et ainsi nettoyer le fluide. La sédimentation peut aussi être employée comme moyen de séparation des grains ayant des vitesses de sédimentation différentes.

Nous pouvons analyser deux types de sédimentation : la sédimentation de suspensions mono-dispersées, contenant un seul type de grains et la sédimentation de suspensions poly-dispersées pour laquelle les grains peuvent avoir différentes tailles et densités. Nous allons uniquement considérer une suspension de grains rigides ayant les mêmes dimensions et les mêmes densités. Ces grains sédimentent dans un fluide newtonien à très faible nombre de Reynolds particulaire. Nous nous sommes inspirés de l'article de [David, 1985] qui suppose que les forces de Van der Waals et le mouvement brownien sont suffisamment faibles pour que cette suspension ne s'agglomère pas. Dans le cas où la suspension n'est pas infiniment diluée, la vitesse moyenne de sédimentation des grains est plus lente que la vitesse de Stoke donnée pour une sphère de rayon a et de densité ρ_p par,

$$U = \frac{2a^2(\rho_p - \rho)g}{9\mu},$$

où μ est la viscosité de fluide et g la constante gravitationnelle. Pour une suspension mono-dispersée, la vitesse moyenne de sédimentation est donnée par,

$$\bar{U} = Uf(\phi) \quad \text{avec} \quad f(\phi) = \frac{(1 - \phi)^2}{(1 + \phi^{1/3}) \exp(5\phi/3(1 - \phi))}, \quad (3.4)$$

où $f(\phi)$ est la fonction de sédimentation entravée, *hindered settling function*. Grâce à cette expression, nous pourrions utiliser la vitesse initiale de sédimentation comme grandeur caractéristique pour adimensionnaliser les gradients de pression et de vitesse dans le calcul de l'estimateur.

Les grandeurs macroscopiques de l'écoulement étudié sont la position de l'interface fluide/grains au cours du temps et la vitesse moyenne de la suspension. L'interface, aussi appelée front du dépôt, est définie arbitrairement comme la moyenne des positions des 120 premiers grains. Par *premiers*, nous entendons les grains ayant l'ordonnée la plus grande en valeur absolue étant donné la définition du domaine. Quant à la vitesse de sédimentation, il s'agit de la moyenne de la norme de vitesse de chaque grain composant l'écoulement. Nous pourrions comparer les résultats de différentes simulations utilisant l'adaptation avec une simulation réalisée sur un maillage fixe et uniforme.

Validation des grandeurs caractéristiques

Afin de trouver la vitesse et la pression caractéristiques, nous avons calculé une solution de référence sur un maillage contenant 9600 éléments et dont la taille de la maille est de 10^{-3} . Les caractéristiques géométriques du domaine et physiques de l'écoulement sont résumées au tableau (Tab. 3.3). Les grains sont placés dans le rectangle grisé de telle sorte que la porosité qui en résulte soit de 0.25. La simulation a été effectuée sur un temps total de $t_{end} = 10$ et un pas de temps fixé à $5 \cdot 10^{-3}$.

Sur base de cette simulation, nous avons pu extraire le maximum de la norme de la vitesse et le maximum de pression pour chaque pas de temps, voir figure (Fig. 3.9). Notons que la vitesse caractéristique fournie par l'expression (Eq. 3.4) possède le même ordre de grandeur que les vitesses maximales extraites. Quant à la pression caractéristique basée cette fois sur la pression dynamique, elle possède également le même ordre de grandeur. Nous en concluons que les grandeurs caractéristiques précédemment définies sont cohérentes avec les grandeurs physiques obtenues par la simulation et que nous pouvons les utiliser pour adimensionnaliser notre estimateur en vitesse/pression.

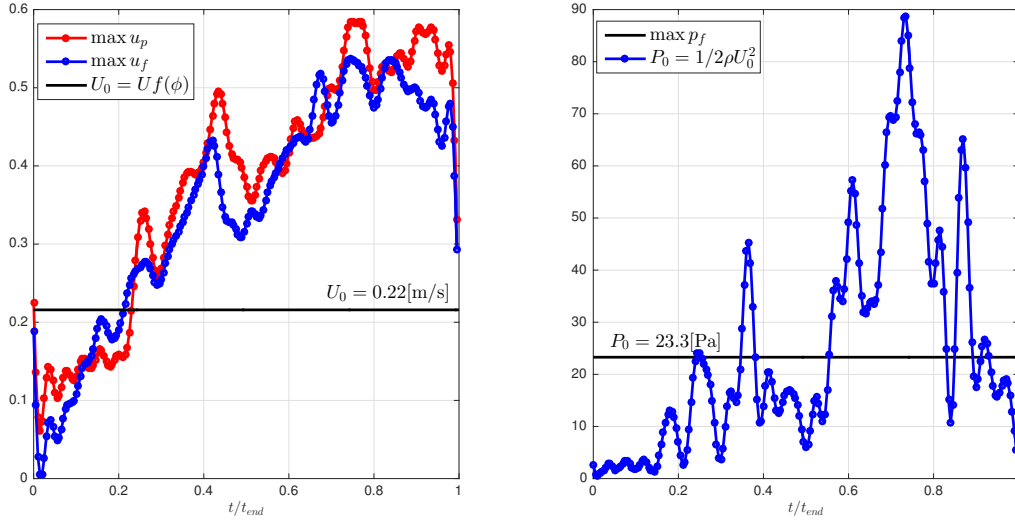


FIGURE 3.9 – La vitesse caractéristique U_0 et la pression caractéristique P_0 ont les mêmes ordres de grandeurs que le maximum de la norme de la vitesse et le maximum de la pression évalués à chaque pas de temps, respectivement.

Validation de la stratégie adaptative

Tout comme dans la section précédente, nous avons comparé trois types de maillage, un maillage raffiné, un maillage grossier et un maillage adapté. Pour les trois simulations, le pas de temps était de $5 \cdot 10^{-3}$ avec un temps final de $t_{end} = 5$. Pour la partie adaptative, nous avons utilisé une erreur de 10^{-7} et un $\Delta t = 10$, utilisé pour remailler en avance. Dans les trois situations, nous avons relevé la position du front du dépôt. Ce front est calculé comme la moyenne des positions verticales des grains qui forment l'interface avec le fluide. La position du front après le temps de simulation pour les trois maillages est fournie à la figure (Fig. 3.10).

Dans le cas de la sédimentation, le nombre d'éléments dans le maillage est assez élevé. Il reste bien moindre que dans le cas d'un maillage raffiné mais la stratégie adaptative génère le double d'itération. C'est la raison pour laquelle le temps CPU n'est pas nettement plus faible comme nous avons pu le constater dans le cas test du nuage de grains.

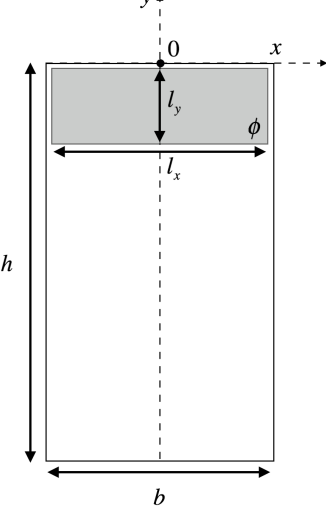
Géométrie du problème	Données
	Concernant le fluide, $\rho = 1000 \text{ [kg/m}^3\text{]}$ $\mu = 10^{-3} \text{ [Pa. s]}$ Concernant les grains, $\rho_p = 1500 \text{ [kg/m}^3\text{]}$ $r_p = 1 \cdot 10^{-3} \text{ [m]}$ $\phi = 0.25 \text{ [-]}$ $N_0 = 3024 \text{ [-]}$ Concernant la géométrie, $h = 1.2 \text{ [m]}$ $b = l_x = 0.4 \text{ [m]}$ $l_y = 0.1 \text{ [m]}$ La vitesse caractéristique vaut donc, $V_0 = 0.22 \text{ [m/s]}$ La pression dynamique vaut, $p_0 = 23.3 \text{ [Pa]}$ Le nombre de Reynolds basé sur les grains vaut, $Re_p = 215$

TABLE 3.3 – Données géométriques et physiques pour la sédimentation en tube droit.

	$h_{\min} \text{ [m]}$	Nombre d'éléments	CPU	$Y_f(t = t_{end})$
Maillage fin (A)	$5r_p$	38400	17 min 6 s	-0.638
Maillage grossier (B)	$20r_p$	12872	1 min 1 s	-0.722
Maillage adapté (C)	$5r_p$	$4350 < n_{elt} < 20734$	13 min 58 s	-0.634

TABLE 3.4 – Le maillage adaptatif nous permet de réduire le temps de la simulation tout en ayant une précision assez proche du maillage raffiné avec une erreur de l'ordre de $\mathcal{O}(10^{-3})$ sur la position du front.

Evolution du maillage pour l'estimateur en vitesse/pression

Il serait également intéressant d'afficher le maillage par rapport à la position de la suspension dans le tube droit. Le maillage est obtenu grâce à la stratégie adaptative pour laquelle l'erreur prescrite est de 10^{-7} et l'intervalle adaptatif est de $\Delta t = 10$. Le temps final de la simulation fut fixé à $t_{end} = 5$. Au démarrage, le nombre d'éléments est encore suffisamment restreint que pour que la stratégie adaptative soit efficace. Au fur et à mesure, le champ de vitesse devient de plus en plus intense dans la partie haute du domaine.

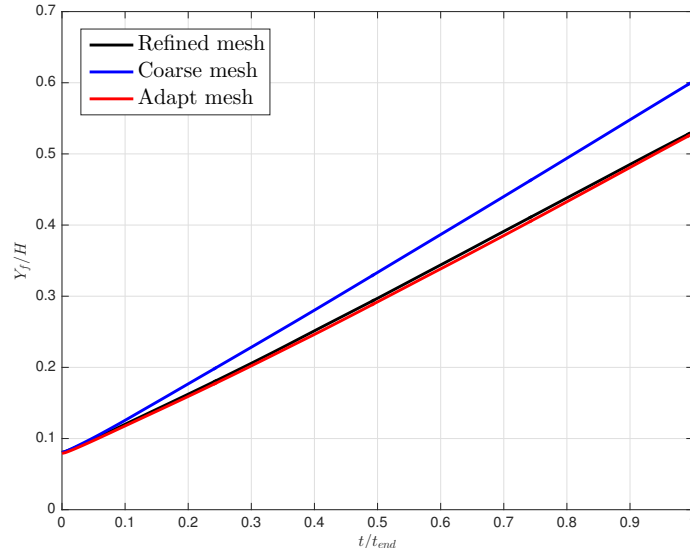


FIGURE 3.10 – Observation de l'évolution de la position verticale du front, interface entre les grains et le fluide pour les trois maillages, notons que la position du front dans le cas adapté et dans le cas raffiné sont très proches. L'erreur entre les deux courbes est de l'ordre de $\mathcal{O}(10^{-3})$ alors que l'erreur maximale entre la position du front dans le cas du maillage grossier est de l'ordre de $\mathcal{O}(10^{-2})$. L'adaptation locale nous fait gagner un ordre de grandeur.

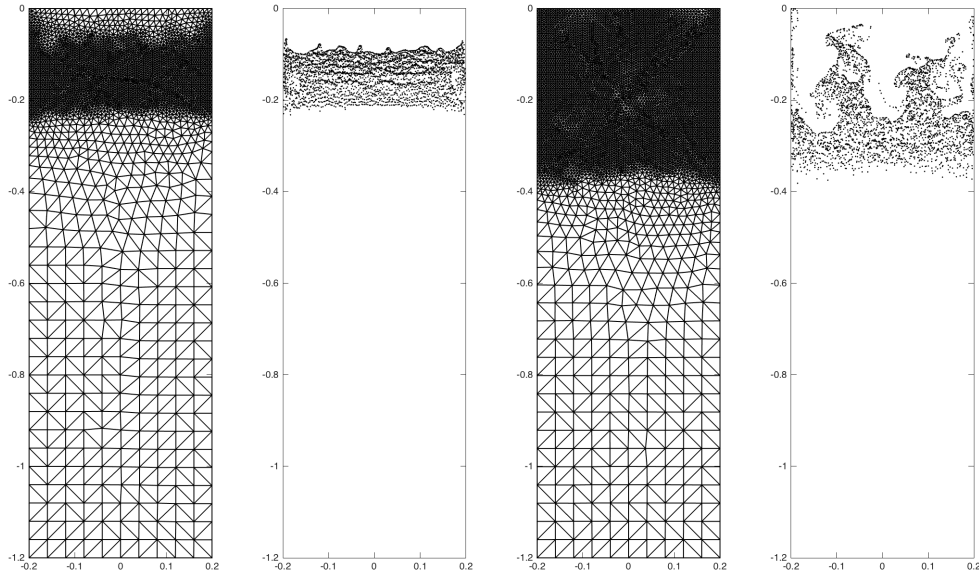


FIGURE 3.11 – Observation du maillage adapté versus la position de la suspension au temps $\frac{t}{t_{end}} = 0.25$ et $\frac{t}{t_{end}} = 0.5$.

Les grains situés dans cette partie du domaine sont emportés en formant de jolies volutes, comme illustré à la figure (Fig. 3.11). Dans la partie basse du domaine, il ne se passe encore rien, le fluide est encore au repos. A la figure (Fig. 3.12), le champ de vitesse et de pression sont affichés côte à côte avec le maillage. L'interface

fluide/grains est bien capturée par l'estimateur comme indiqué par les rectangles rouges. Dans ce rectangle du côté du maillage, nous remarquons quelques variations au niveau des tailles des éléments. Ces variations sont le reflet des légères fluctuations du champ de vitesse formant des filaments verticaux dans l'image de gauche.

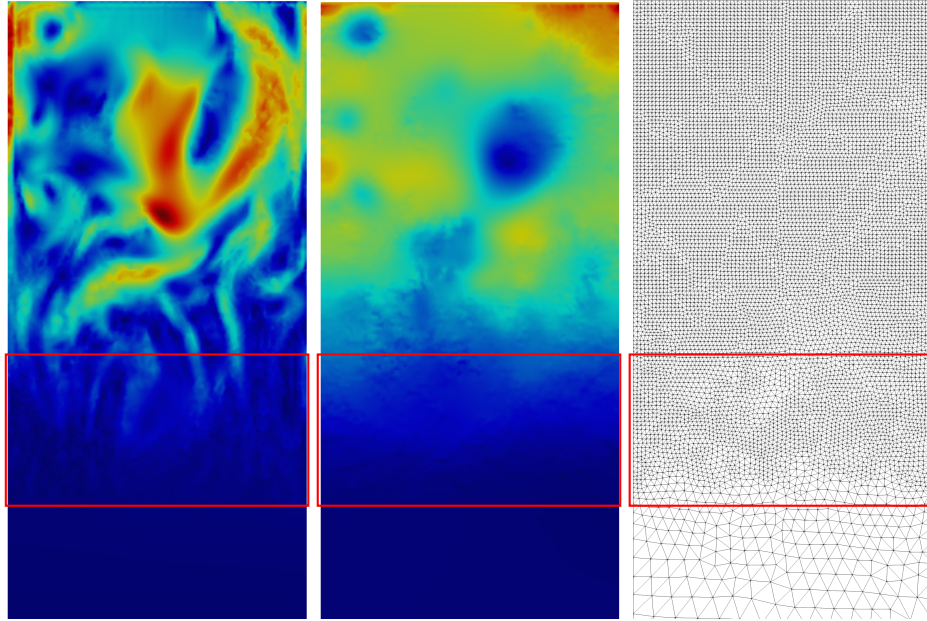


FIGURE 3.12 – Observation du champ de vitesse (à gauche), du champ de pression (au milieu) et du maillage au temps $\frac{t}{t_{end}} = 1$

Cependant, une fois que les grains ont touché le fond du récipient, l'entière du domaine observe des variations du champ de vitesse et de pression. L'estimateur raffinant sur ces deux champs impose donc une taille minimale des éléments sur la quasi-totalité du domaine. Le nombre d'éléments monte en flèche et ralentit le temps de simulation. Bien que la quasi totalité des grains soit désormais dans la couche de sédiments au fond du récipient, il y a toujours une petite fraction de grains présente dans le haut du domaine. Il faut attendre une longue période avant que les champs de pression et de vitesse décroissent en intensité, que les grains finissent par retomber et que l'estimateur commence à déraffiner le maillage.

3.4.3 L'effet Boycott : sédimentation de suspensions en tube incliné

De prime abord nous pouvons expliquer pourquoi la sédimentation dans un tube incliné est plus rapide que dans un tube droit. Dans les deux cas, schématisés à la figure (Fig. 3.13), incliné et droit, les grains vont sédimenter verticalement jusqu'à ce qu'ils atteignent la paroi inférieure ou le fond du tube, respectivement. Pour le tube droit, un grain va voyager sur une distance de l'ordre de $\mathcal{O}(h(t))$ avant de pouvoir être récolté alors qu'un grain en tube incliné ne parcourt qu'une distance de l'ordre de $\mathcal{O}(b)$. L'amélioration est de l'ordre de $\mathcal{O}(h(t)/b)$. Cette prédiction est tout à fait correcte bien que le processus de sédimentation en tube incliné soit plus complexe que cela comme nous l'explique l'article de [Acrivos et Herbolzheimer, 1979]. En effet, les interactions entre les grains dépendent de la concentration locale mais aussi du cisaillement.

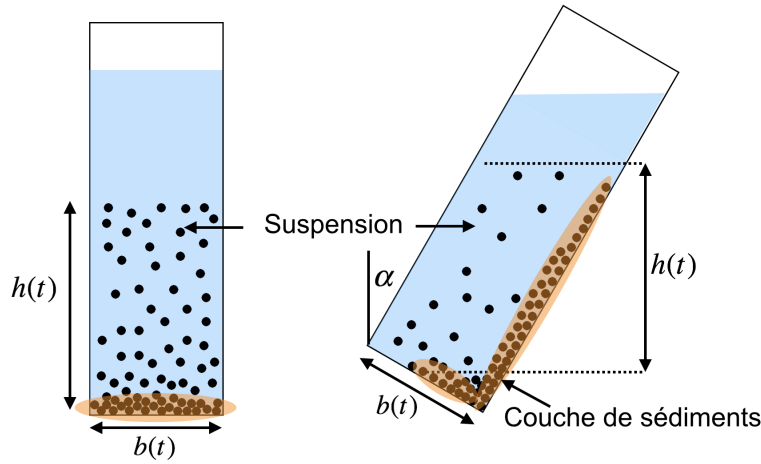


FIGURE 3.13 – Comparaison entre la sédimentation en tube droit et incliné.

Ce processus de sédimentation possède des caractéristiques qui lui sont propres dont l'angle d'inclinaison de la couche de sédiments et la vitesse de sédimentation. Les travaux [Palma et al, 2016] ont montré que l'angle d'inclinaison final de la couche de sédiments dépend fortement de la concentration initiale de grains et de l'inclinaison du tube mais pas de la viscosité du fluide. Nous devons d'abord définir la vitesse U_0 et la pression P_0 caractéristiques de l'écoulement et ensuite déterminer la position de l'interface fluide/grains et la vitesse de sédimentation pour différentes inclinaisons.

Obtention des grandeurs caractéristiques de l'écoulement

Afin de déterminer les grandeurs caractéristiques qui vont adimensionnaliser notre estimateur, nous avons réalisé trois simulations de référence sur un maillage fin comportant 9600 éléments pour une taille minimale de la maille de 10 fois le rayon des grains. Pour chacune de ces expériences, nous avons fixé trois inclinaisons différentes : 0° , 20° et 30° . Les caractéristiques physiques des grains et la géométrie du domaine sont résumées dans le tableau (Tab. 3.5). Les grains sont placés de manière aléatoire

dans le rectangle grisé de façon à retrouver une porosité assez faible de 0.08. Nous avons également relevé la vitesse moyenne du fluide et sa pression moyenne. Celles-ci sont reprises à la figure (Fig. 3.14).

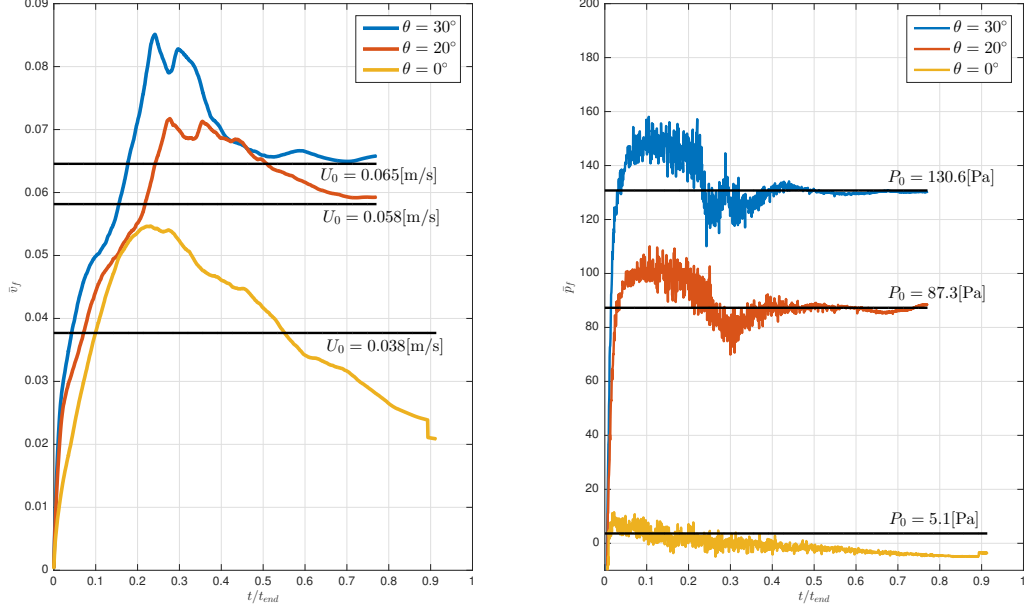


FIGURE 3.14 – Vitesse et pression moyennes pour les trois expériences de sédimentation en tube incliné.

Il s'avère que la vitesse moyenne du fluide est plus élevée dans le tube incliné à 30° que dans celui à 20° et encore plus que dans le tube droit. A noter que l'angle d'inclinaison ne doit pas être trop fort sinon les grains ne vont plus pouvoir glisser sur la paroi inférieure ayant une énergie cinétique insuffisante avec une faible pente. Autrement dit, l'effet Boycott est intéressant dans une certaine gamme d'angle d'inclinaison.

Validation de la stratégie adaptative

Pour cette validation, les simulations vont être réalisées sur trois maillages différents, un raffiné, un grossier et un adapté. Nous allons cibler le cas du tube incliné à 30° et nous comparerons les résultats des trois simulations pour la position du front de grains et la vitesse moyenne des grains. L'ensemble des données des maillages utilisés est fourni dans le tableau (Tab. 3.6). Néanmoins, nous notons que la stratégie adaptative n'est pas efficace dans ce cas-ci. Le nombre d'éléments dans le maillage C est bien moindre que dans le maillage A mais le fait d'adapter en avance et donc de faire deux fois plus d'itérations génère un temps de calcul bien plus long. Une manière de pallier à ce problème serait de réduire le nombre d'itérations à faire en plus, en ne revenant que partiellement en arrière dans la stratégie adaptative.

Puisque le maillage B contient moins d'éléments, chaque élément est localement plus grand et le phénomène de sur-pression appliqué par le fluide sur chaque grain est plus faible. Les grains observent moins de résistance de la part du fluide et ceux-ci

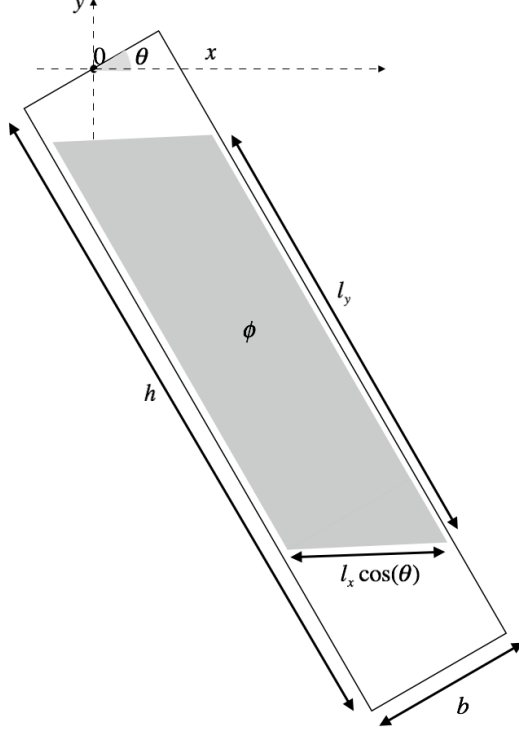
Géométrie du problème	Données
	Concernant le fluide, $\rho = 1000 \text{ [kg/m}^3\text{]}$ $\mu = 10^{-3} \text{ [Pa. s]}$ Concernant les grains, $\rho_p = 2200 \text{ [kg/m}^3\text{]}$ $r_p = 1 \cdot 10^{-3} \text{ [m]}$ $\phi = 0.08 \text{ [-]}$ $N_0 = \sim 1120 \text{ [-]}$ Concernant la géométrie, $h = 1.2 \text{ [m]}$ $b = l_x = 0.1 \text{ [m]}$ $l_y = 0.8 \text{ [m]}$ $\theta = 30^\circ$

TABLE 3.5 – Données géométriques et physiques pour la sédimentation en tube incliné.

	$h_{\min} \text{ [m]}$	Nombre d'éléments	CPU	Position du front
Maillage fin (A)	$5r_p$	9600	10 min 23 s	voir figure (3.15)
Maillage grossier (B)	$20r_p$	600	1 min 23 s	voir figure (3.15)
Maillage adapté (C)	$5r_p$	$1145 < n_{elt} < 8669$	26 min 0 s	voir figure (3.15)

TABLE 3.6 – La stratégie adaptative ne nous permet pas de réduire le temps de la simulation mais la précision de la simulation est très similaire à celle sur le maillage raffiné.

chutent plus rapidement. Pour le maillage A et C, les courbes sont très similaires. L'adaptation locale permet bien de cibler les zones du domaine qui rencontrent les grains et raffine en conséquence afin que le phénomène de sur-pression soit similaire à celui ressenti par les grains dans le maillage A.

Evolution des grandeurs caractéristiques

Nous avons étendu notre étude des écoulements en tubes droits et inclinés en comparant les résultats du maillage adapté avec ceux du maillage raffiné pour un temps final de $t_{end} = 7$. Nous avons gardé les mêmes configurations que dans les trois expériences précédentes. L'idée est de comparer l'évolution de la position du front et de la vitesse moyenne des grains calculés sur le maillage raffiné et sur le maillage adapté. Les résultats sont présentés à la figure (Fig. 3.16). Sur le graphe de gauche, nous observons la position du front et il est clair que ce front est plus rapidement dans le fond du tube lorsque celui-ci est incliné. Sur le graphe de droite,

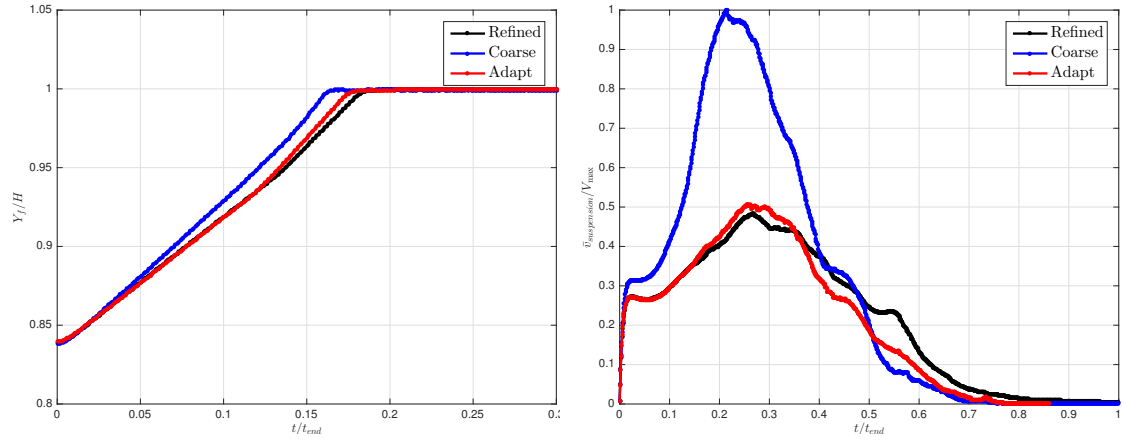


FIGURE 3.15 – Evolution temporelle de la position du front et de la vitesse moyenne des grains pour les trois maillages. Le maillage raffiné et celui adapté fournissent des résultats très semblables.

nous observons la vitesse moyenne des grains dans les trois cas. Les deux pics de vitesse observés pour les cas inclinés expliquent bien le concept de l'effet Boycott. Lorsque le tube est incliné, nous avons vu qu'un contre-flux apparaissait sur la paroi supérieure ne gênant plus les grains qui eux glissent le long de la paroi inférieure. De cette manière, les grains atteignent une vitesse bien plus élevée.

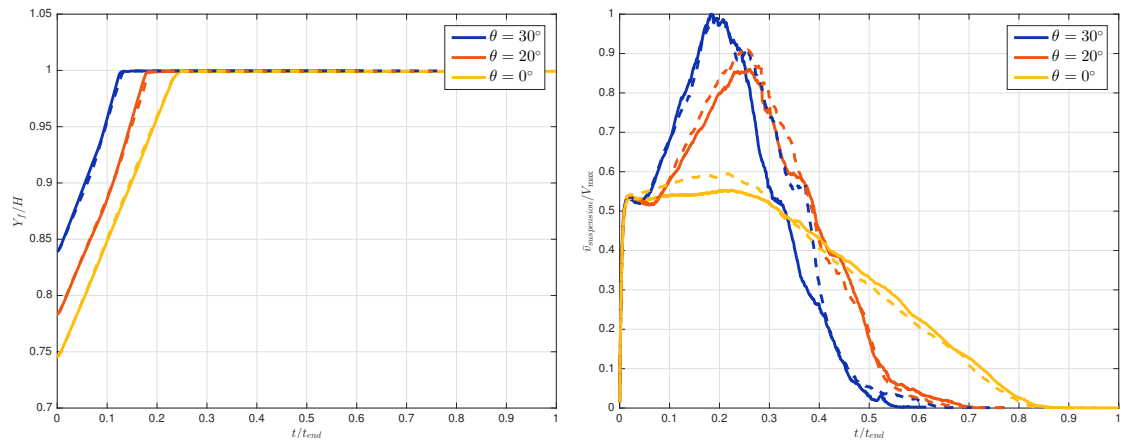


FIGURE 3.16 – Les courbes en trait plein correspondent aux simulations réalisées sur un maillage adapté. Les courbes du maillage adapté et fixe se suivent de très près. Basé sur ces données macroscopiques, le maillage adapté représente la même physique que le maillage fixe et ne déforme que très peu la solution.

Dans le cas du tube droit, les grains qui sédimentent sont gênés par un flux qui remonte au travers de la sédimentation. Ce phénomène explique pourquoi la courbe jaune atteint plus rapidement sa vitesse maximale car les grains sont plus vite freinés par le flux montant. Nous pouvons visualiser l'écoulement des grains dans les trois cas en vis-à-vis des trajectoires de l'écoulement instationnaire, figure (Fig. 3.17). Ces trajectoires suivent le mouvement d'une particule de fluide au cours du temps. Elles ont été tracées avec le logiciel *Paraview*. La ligne source est prise depuis le coin inférieur gauche au coin supérieur droit. Le logiciel affiche ensuite toutes les

trajectoires des points se situant sur cette ligne source. Dans le cas des tubes inclinés, nous observons des trajectoires bien plus nettes parallèles à la paroi supérieure du tube. Les grains descendent le long de la paroi par paquet ce qui génère un accroissement local de la vitesse du fluide. Les trajectoires des particules fluides viennent s'enrouler autour de ces paquets de grains. Dans le cas du tube droit, les trajectoires des particules fluides se faufilent entre les grains chutant, les ralentissant.

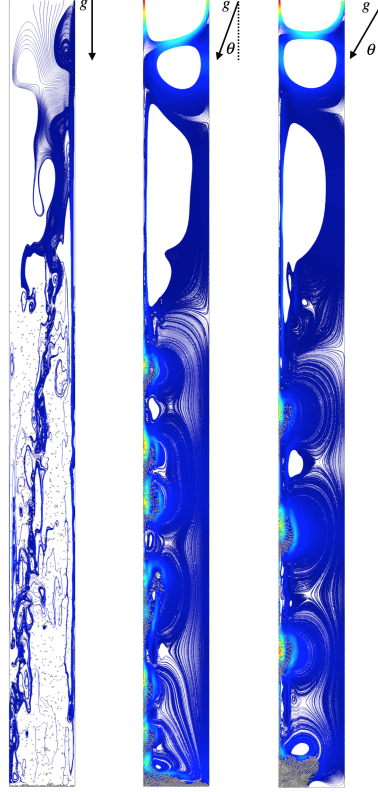


FIGURE 3.17 – Pour le temps $\frac{t}{t_{end}} = 0.57$, comparaison de la position des grains et des trajectoires des particules fluides de l'écoulement pour les trois tubes.

Evolution du maillage pour l'estimateur en vitesse/pression

Lorsque les grains glissent le long de la paroi inférieure, le fluide est mis en mouvement. C'est sur cette bande que des variations de vitesse et de pression vont se faire ressentir. Puisque l'estimateur se base expressément sur les gradients de vitesse et de pression, nous devrions voir le maillage se raffiner sur cette bande. La figure (Fig. 3.18) affiche le champ de pression (à droite), le champ de vitesse (au milieu) et le maillage adapté. Le comportement de l'estimateur est bien celui escompté car il raffine là où les variations se font ressentir.

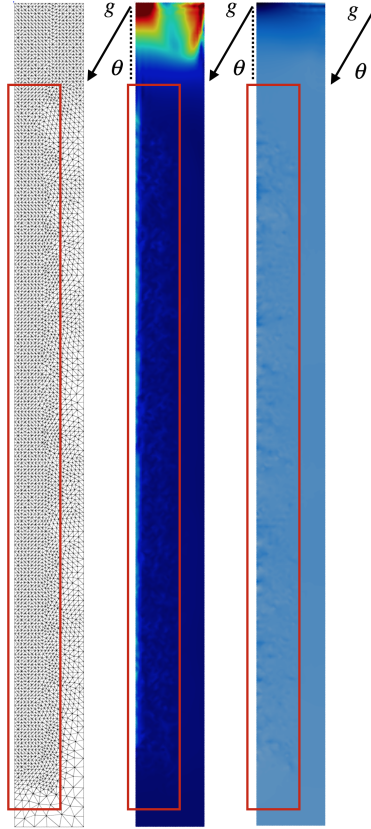


FIGURE 3.18 – Le maillage est bien raffiné aux endroits du domaine qui observe une variation du champ de vitesse et de pression de l'écoulement.

3.5 Conclusion sur la stratégie adaptative

Ces trois cas tests ne représentent qu'un petit panel d'exemples d'écoulements granulaires néanmoins ils nous ont permis de valider notre estimateur en vitesse/pression. Dans chacun des cas, l'estimateur raffine notre maillage de telle sorte que l'écoulement qui en résulte soit tout aussi précis que l'écoulement obtenu sur un maillage uniformément raffiné. Dans le cas du nuage de grains, nous avons validé la précision sur la position du centre de masse. De plus, ce cas test fut vraiment convainquant en termes de temps de calcul, le temps mis pour résoudre l'écoulement sur le maillage adapté était bien moindre. Dans le cas de la sédimentation en tube droit, la précision fut cette fois mesurée grâce à la position du front. Dans ce cas-là étant donné la grandeur du domaine, la stratégie adaptative avait son sens et le gain de temps entre la simulation sur le maillage raffiné et sur le maillage adapté était notable. Notre déception concernant notre stratégie adaptative est liée au dernier cas test, bien que la précision sur la position du front et la vitesse moyenne des grains soient extrêmement proches de la solution sur le maillage fin, le temps de calcul est bien trop long et ce à cause de la stratégie adaptative *en avance* qui requiert le double d'itérations.

Nous avons quelques pistes d'améliorations possibles. Premièrement, nous aimerions limiter l'ajout d'itérations concernant la stratégie adaptative. L'adaptation

du maillage à chaque itération s'avère beaucoup trop chère c'est pour cette raison que nous avons décidé de n'adapter que toutes les X itérations. Cependant, cette approche nous amenait un autre problème, le maillage risquait d'être toujours en retard par rapport à l'interface fluide/grains. Le front étant moins bien capturé, des grains se retrouvaient dans des éléments grossiers dans lesquels le phénomène de sur-pression serait limité entraînant une accélération de la chute des grains. Sur base de cette observation, nous avons donc choisi d'implémenter une stratégie adaptative *en avance* qui prenait les informations sur X itérations pour construire le nouveau maillage et ensuite revenait dans *le passé* pour repartir avec ce nouveau maillage. Une manière de limiter le problème du surplus de temps de calcul serait de ne revenir que partiellement dans le passé. Cette nouvelle stratégie est schématisée à la figure (Fig. 3.19) où les lignes en pointillés rouges représentent l'arrêt, nous ne nous déplacerions plus que de $\Delta t/2$ en arrière, par exemple.

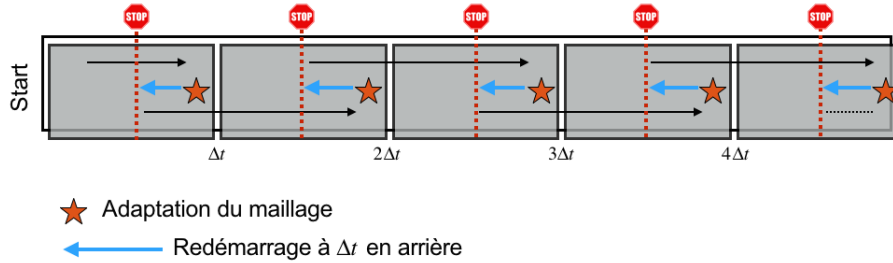


FIGURE 3.19 – Nouveau schéma adaptatif n'utilisant plus que la moitié d'itérations supplémentaires.

Deuxièmement, il serait intéressant de trouver une stratégie qui permettrait de limiter le nombre d'éléments dans le nouveau maillage sans palier à la précision de la solution. Troisièmement, nous avons dû pour chaque cas test déterminer la vitesse et la pression caractéristiques de l'écoulement de manière externe. Il serait possible de calculer cette vitesse et cette pression ou tout au plus un ordre de grandeur de manière automatique. L'idée serait d'utiliser les premières itérations de la simulation pour calculer la moyenne du champ de vitesse et de pression et d'ensuite démarrer la stratégie adaptative avec ces valeurs. De cette manière l'utilisateur pourrait profiter d'une adaptation sans devoir se renseigner en amont sur le type d'écoulement qu'il veut simuler.

CHAPITRE 4

ADAPTATION LOCALE DU MAILLAGE

Dans ce dernier chapitre, nous allons aborder la question de l'adaptation locale. Précédemment, nous utilisons le logiciel **GMSH** pour obtenir le nouveau maillage à partir de la carte de tailles fournie par l'estimateur. Une autre technique est cependant possible : l'adaptation locale. Cette technique revient à uniquement raffiner les éléments ciblés par l'estimateur sans modifier complètement le maillage. Dans cette approche, nous tenons compte du maillage existant, celui-ci n'est plus effacé au profit d'un tout nouveau maillage. Avec cette approche, nous cherchons à minimiser la diffusion produite par l'interpolation de la solution existante sur le nouveau maillage. Cette étape sera toujours nécessaire mais elle sera plus localisée. Il est important de souligner que l'interpolation même de haut ordre génère de la diffusion numérique et engendre donc une perte d'incompressibilité de l'écoulement.

L'opérateur d'interpolation est un opérateur de diffusion d'ordre $\mathcal{O}(h^2)$. En effet, en passant d'un maillage à un nouveau, les noeuds de l'un ne se retrouvent pas nécessairement aux mêmes endroits que les noeuds de l'autre, pour chaque noeud du nouveau maillage, il est alors nécessaire de trouver dans quel élément de l'ancien maillage il se situe. Une fois localisé, nous effectuons une interpolation de la solution obtenue sur les trois noeuds de l'élément et nous évaluons cette interpolation au nouveau noeud. Cette opération engendre une perte d'information. Cette perte d'information se traduit comme une diffusion numérique. Ce n'est pas le problème physique qui introduit de la diffusion mais le code numérique lui-même. Nous pouvons illustrer ce phénomène de diffusion de manière très simple. Nous nous rapportons à un problème en une dimension afin de rendre l'exposé plus facile à comprendre mais il peut s'appliquer sans problème aux dimensions supérieures. Supposons que nous ayons deux maillages m_1 et m_2 dont les éléments sont du même ordre mais les noeuds de ses deux maillages ne se trouvent pas aux mêmes endroits. Supposons également que nous ayons obtenu une solution f_1 sur le maillage m_1 et que nous voulons obtenir sa valeur sur le maillage m_2 . Il s'avère nécessaire d'effectuer une interpolation de cette solution sur le nouveau maillage. Cette interpolation est une interpolation linéaire basée sur les fonctions de forme ϕ_1 et ϕ_2 présentées à la section 2.1 du deuxième chapitre. La fonction considérée est une gaussienne de moyenne μ

nulle et de variance $\sigma = 0.1$. L'idée est d'appliquer ce processus plusieurs fois. La solution évaluée sur le maillage m_1 est interpolée sur le maillage m_2 puis la solution ainsi obtenue sur m_2 est de nouveau interpolée mais sur le maillage m_1 et ainsi de suite. A la figure (Fig. 4.1), nous observons bien que la gaussienne diffuse au fur et à mesure des itérations. Après seulement 90 itérations, la solution a perdu 30% de sa valeur maximale. Avec cet exemple, nous motivons notre envie d'adapter localement le maillage afin de limiter ce phénomène de diffusion numérique.

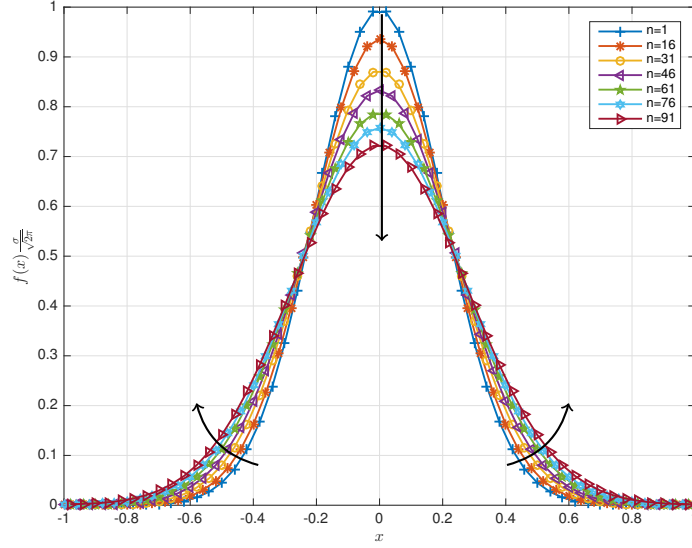


FIGURE 4.1 – La Gaussienne diffuse peu à peu dans le domaine et perd 30% de sa valeur après 90 itérations du processus d'interpolation d'un maillage à l'autre.

Dans la littérature, deux grandes méthodes de raffinement local sont largement documentées : la méthode de *quadrisections* introduite pour la première fois par Bank et Sherman [Bank et Sherman, 1981] et la méthode dite *newest vertex bisection*, [Mitchell, 2016].

La première méthode fut largement utilisée dans les codes d'éléments finis jusque dans les années 1980. Cette méthode consiste à subdiviser l'élément en quatre sous-éléments, cette étape est appelée *red refinement* alors que l'étape de *green refinement* est quant à elle, utilisée pour recouvrer la compatibilité et la stabilité du maillage, comme illustré à la figure (Fig. 4.2). Leur algorithme est motivé par deux contraintes.

1. La taille du plus petit angle interne de tout triangle est délimité loin de zéro.
2. La transition entre les différentes tailles d'éléments se doit d'être graduelle.

Lorsqu'un triangle est subdivisé en quatre, les triangles résultant de cette subdivision possèdent les mêmes angles que le triangle parent, nous parlons alors de raffinement régulier. Il est à noter que la division en deux sous-triangles transgresse la première contrainte. C'est d'ailleurs pour cette raison que les auteurs n'utilisent cette découpe que pour rendre le maillage conforme et qu'il est primordial de retirer cette subdivision avant toute nouvelle adaptation.

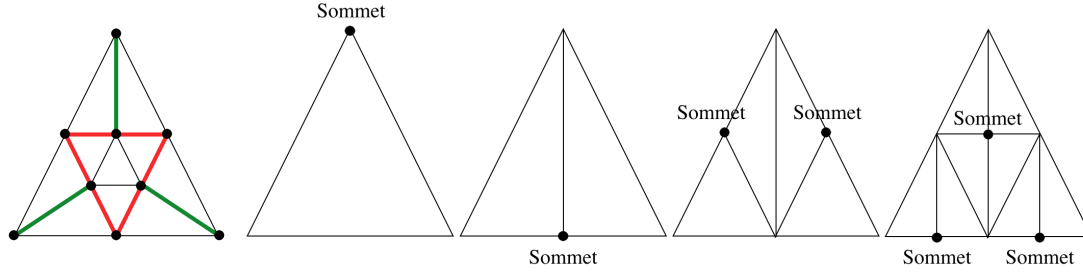


FIGURE 4.2 – Sur l’image de gauche, nous observons la méthode de Bank et Sherman où le triangle interne est découpé en quatre (red refinement) et ensuite le raccordement (green refinement) est opéré avec ses voisins afin de supprimer les noeuds flottants. Sur l’image de droite, nous observons la propagation du sommet déterminant l’arête à découper pour la newest vertex bisection méthode.

La question posée par le professeur Bob Skeel est la suivante : Est-il possible de trouver un algorithme de raffinement ne devant pas passer par ce processus compliqué d’empiècement et de désempiècement ? La réponse à cette question fut la naissance d’un nouvel algorithme de raffinement local appelé *newest vertex bisection*. Cette nouvelle approche est plus simple et permet de constamment garder la compatibilité et la stabilité du maillage. Dans cette méthode, un triangle est raffiné en connectant un sommet à son arête opposée, dite arête de raffinement. L’initialisation de ce sommet pour chaque élément n’est pas arbitraire. Cette initialisation permet à la méthode ne pas dégénérer les triangles après une utilisation répétée. Cette initialisation est basée sur la théorie des graphes et plus précisément sur l’existence d’une couverture parfaite dans un graphe cubique, démontrée par [Petersen, 1891]. Nous pouvons montrer que le graphe dual d’une triangulation, maillage formé à partir de triangles, est un graphe cubique, chaque noeud du graphe possède exactement trois voisins. Par la suite, le sommet opposé à l’arête de raffinement devient simplement le noeud milieu de l’arête découpée, comme illustré dans la figure (Fig. 4.2). Dans le cas où un ensemble d’éléments doit être raffiné, il en résulte un certain nombre de noeuds flottants qui doivent être eux-mêmes connectés à un autre noeud afin de rendre le maillage conforme. Ce processus de gestion des noeuds flottants génère une séquence de triangles appelée chaîne de compatibilité et se termine dans deux cas :

1. deux triangles partageant une arête de raffinement commune sont raffinés en même temps,
2. un triangle ayant une arête de raffinement sur une frontière est raffiné.

La méthode d’adaptation locale développée pour le code MigFlow est plus proche de la méthode de Bank et Sherman. Nous décidons de découper les triangles devant être raffinés en quatre afin d’obtenir un raffinement régulier mais nous acceptons les découpes en deux et en trois. Ces deux types de raffinement nous permettent de gérer les noeuds flottants et ils devront eux aussi être retirés au cours d’une nouvelle adaptation toujours dans le but de ne pas dégénérer les éléments lors d’une utilisation répétée. En ce qui concerne la gestion des noeuds flottants, trois configurations sont à prendre en compte : un seul voisin est raffiné, deux voisins sont raffinés et trois voisins sont raffinés. Ces différentes configurations sont résumées dans le tableau (Tab. 4.1)

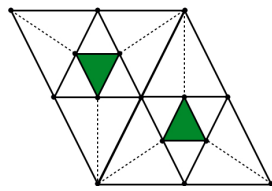
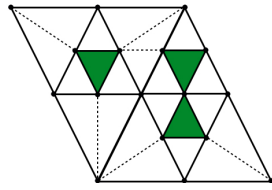
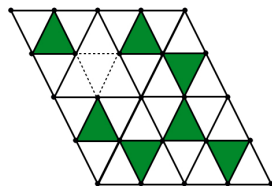
Case 1		Dans cette figure, seuls deux éléments sont raffinés et ils ne sont pas voisins. Afin de rendre le maillage conforme, une arête est créée en reliant le nouveau noeud au noeud faisant face à l'arête subdivisée dans le triangle voisin.
Case 2		Un des triangles possède deux voisins ayant été raffinés. Le seul moyen d'obtenir un maillage conforme est de créer trois sous-triangles. Le choix de la meilleure configuration est basée sur la maximisation du ratio entre le rayon du cercle inscrit et le rayon du cercle circonscrit au triangle. Ce ratio s'avère être maximal pour un triangle équilatéral. La maximisation de ce critère nous assure de garder le maillage le plus isotrope possible.
Case 3		Dans ce troisième cas, un triangle non raffiné possède trois voisins qui le sont. Ce triangle sera lui aussi raffiné de la même façon en le découpant en quatre sous-triangles. La seule différence est qu'il ne faut pas recréer de noeuds, ceux-ci existent déjà dans la base de données.

TABLE 4.1 – Les différentes configurations liées à la stratégie de raffinement

4.1 Présentation de l'algorithme de raffinement local

Dans cette section, nous présentons les différentes étapes menant à la construction du maillage adapté localement grâce à une succession de découpages. Nous nous sommes basés sur une structure arborescente qui nous permet au fil des itérations de maintenir l'historique du raffinement de chaque élément. Cette structure arborescente en langage **C** sera un tableau dynamique d'arbres. Chaque entrée de ce tableau caractérise un élément du maillage initial. Toutes les ramifications de l'arbre symbolisent les découpes de l'élément en 2, 3 ou 4 enfants. L'arbre est une structure qui possède un certain nombre de champs comme décrits à la figure (Fig. 4.3). Chaque élément est un **Tree_cell** et est donc représenté par un index, un ensemble de trois noeuds, un ensemble de trois voisins, un ensemble d'index physiques décrivant la nature de la frontière, un ensemble d'index pour les arêtes sur les frontières, un nombre d'enfants, un pointeur vers son parent afin de pouvoir remonter dans l'arbre et un pointeur vers un tableau dynamique d'enfants permettant de descendre dans l'arbre. En tout premier lieu, nous devons créer l'arbre au tout début de la simulation sur base du maillage initial. Lors de cette étape, nous devons remplir les champs dédiés

aux noeuds, aux voisins et aux index physiques des frontières pour chaque élément. Dans le cas où un élément se situe sur une frontière, l'index du voisin est mis à -1 et l'entrée correspondante de la liste des index physiques est mise à une valeur différente de -1 . De cette manière, nous connaissons sur quelle frontière physique du domaine se situe l'élément. Cette information est cruciale pour la reconstruction du maillage et pour la reconstruction des frontières. Dans cette étape d'initialisation, nous plaçons les pointeurs du parent et des enfants à NULL.

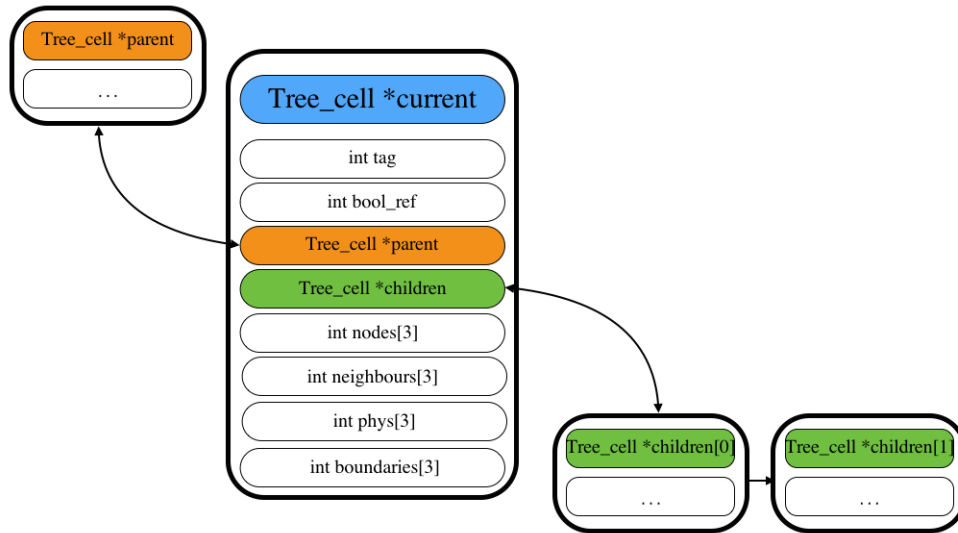
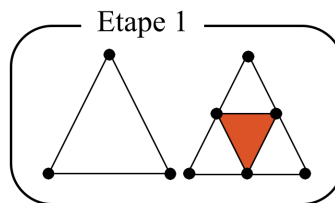


FIGURE 4.3 – Représentation informatique de l'arbre avec son pointeur vers le parent et vers les potentiels enfants de l'élément courant.

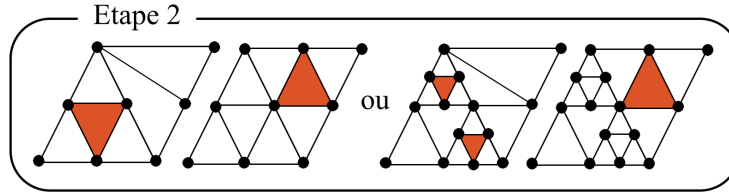
Lorsque le maillage doit être adapté, il est nécessaire de respecter un ordre bien déterminé pour le raffinement. Ce raffinement s'effectue en trois étapes bien distinctes.

1. **Etape 1 :** nous nous intéressons aux raffinements réguliers représentés par une découpe en 4 sous-éléments. Nous ajoutons un tableau de 4 enfants à l'élément. Chacun de ces sous-éléments ou enfants est une structure **Tree_cell** qui reçoit un index et une liste de noeuds le décrivant. Les pointeur des enfants vers l'élément parent est également mis à jour.

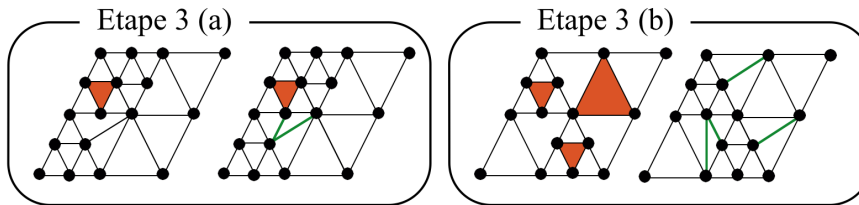


2. **Etape 2 :** les éléments précédemment découpés en 2 ou 3 sous-éléments sont à leur tour découpés en 4 sous-éléments si l'estimateur les a pointés comme élément à raffiner ou s'ils possèdent au moins un voisin qui a été raffiné à la première étape, comme le schématise la figure ci-dessous. Nous commençons par supprimer les 2 ou 3 enfants créés précédemment et ensuite, nous créons

les 4 nouveaux enfants qui reçoivent chacun un index et une liste de noeuds les décrivant dans le domaine. Les pointeurs des enfants vers le parent sont également mis à jour.



3. **Etape 3 :** cette troisième étape complète le raffinement en gérant les noeuds flottants. Si l'élément qui est touché par un noeud flottant, a été précédemment découpé en 2, comme c'est le cas sur la figure de gauche, nous décidons de découper cet élément en 3 en lui ajoutant un enfant. Si l'élément avait été touché par deux noeuds flottants, nous l'aurions découpé en 4, ce qui reviendrait à une découpe régulière. Cette première sous-étape (a) sert à ne pas répandre de nouveaux noeuds flottants comme nous pouvons le faire à l'étape 2. En seconde sous-étape (b), nous complétons le raffinement par les découpes en 2, en 3 ou en 4 pour des éléments qui n'ont pas encore été découpés et qui ne possèdent donc pas d'enfant comme schématisé à la figure de droite. L'élément se voit construire 2 ou 3 ou 4 enfants qui sont initialisés de la même manière que dans l'étape 1 et 2.



En pratique ... petit exemple

Mettons en pratique sur un petit exemple décrit à la figure (Fig. 4.4), ce qui vient d'être expliqué. Nous ne considérons pas encore le traitement des arêtes frontières, nous nous concentrons uniquement sur le jeu de l'ajout et du retrait de certaines structures en suivant les trois étapes décrites ci-dessus. Avant même de commencer les itérations, le tableau d'arbres doit être initialisé depuis le maillage initial. En l'occurrence, les quatre éléments, en bleu, initialement présents seront les racines des quatre arbres qui vont être créés par la suite.

Supposons que notre estimateur nous indique de raffiner l'élément 1 et 4 lors de la première étape d'adaptation. Ces deux éléments vont être traités comme à l'étape 1. Ils sont découpés selon un raffinement régulier et quatre enfants sont ainsi créés. Désormais, le maillage n'est plus conforme car 3 noeuds flottants sont dès lors présents. Ces noeuds ne se trouvent pas sur les arêtes d'éléments possédant 2 ou 3 enfants. L'étape 2 n'est donc pas nécessaire et nous passons directement à la troisième étape.

A cette étape, nous observons donc la création de 2 enfants pour l'élément 2 et de 3 enfants pour l'élément 3.

Lors d'une deuxième adaptation, l'estimateur nous indique de raffiner l'élément 11 qui provient d'une découpe en 3 de l'élément 3. Etant donné qu'il s'agit d'un des trois enfants du troisième élément racine, nous passons par la deuxième étape pour raffiner régulièrement cet élément. Les trois enfants sont supprimés et quatre nouveaux sont créés et initialisés. Il s'avère que le maillage ne possède pas de noeuds flottants et nous ne devons pas passer par la troisième étape.

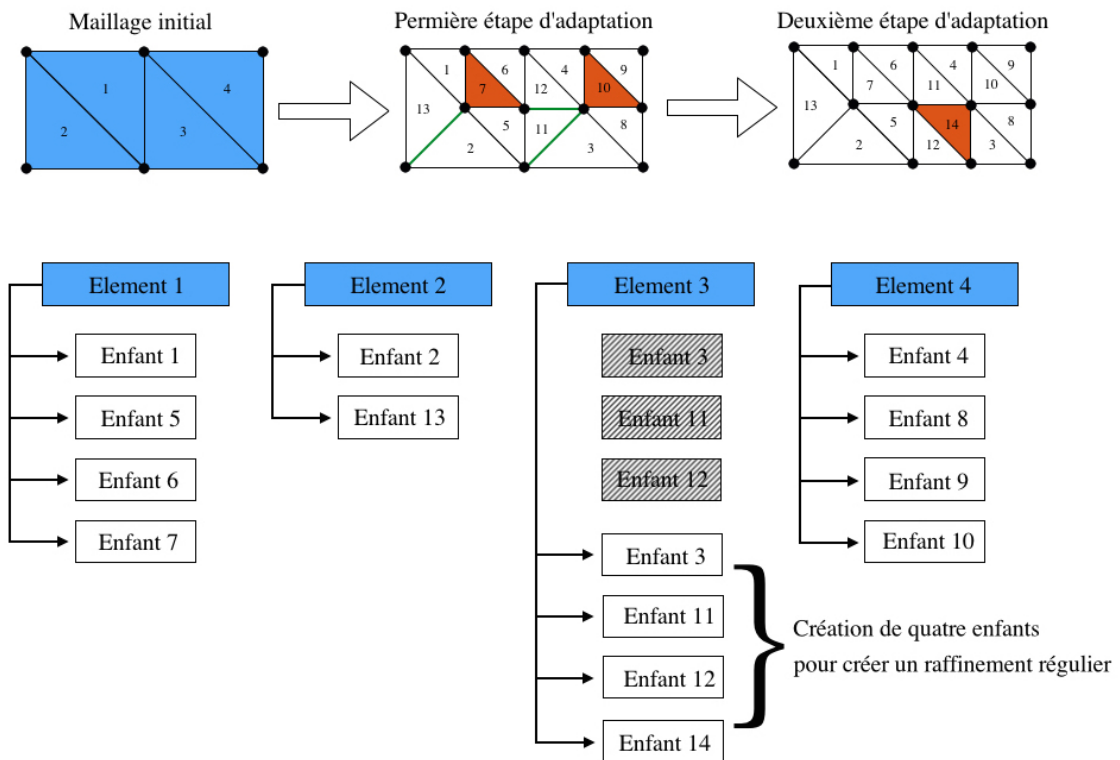


FIGURE 4.4 – Exemple d'un maillage contenant initialement 4 éléments qui est adapté deux fois. La correspondance avec les structures arborescentes y est également indiquée.

La gestion des arêtes frontières

Pour la gestion des frontières, nous avons décidé de travailler avec le champ **boundaries** de la structure **Tree_cell**. A chaque nouvel appel de l'adaptation, nous commençons par réinitialiser tous les éléments du maillage. L'ensemble de ces éléments sont les feuilles des différentes arbres créés depuis l'initialisation. En effet, à l'étape précédente, ces éléments ne contenaient pas l'information à propos des voisins ou bien cette information était faussée par l'ajout de nouveaux éléments générés par les découpages. Cette première étape, nous permet donc de repartir avec des bonnes bases concernant les voisins, les arêtes frontières ainsi que leurs tags physiques. Pour tous les éléments du maillage, si un indice du tableau des voisins vaut -1 , cela signifie qu'il y a une arête sur une des frontières du domaine et `int boundaries[3]` doit être

mis à jour en conséquence. Le positionnement du tag frontière dans ce tableau est lié à la position de l'arête. S'il s'agit de l'arête e_1 , comme schématisé à la figure (Fig. 4.5), la deuxième entrée du tableau est remplie. Ce schéma représente bien comment chaque élément est représenté dans la structure `Tree_cell`. L'arête 0 est caractérisée par le noeud 0 et le noeud 1 et ainsi de suite.

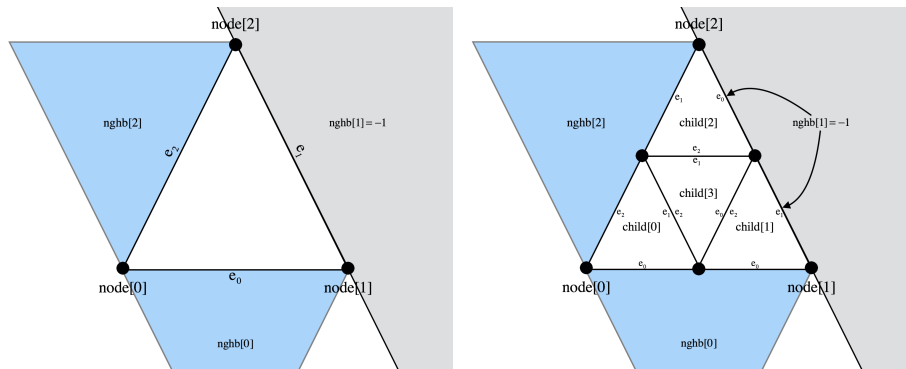


FIGURE 4.5 – Représentation d'un élément triangulaire avec ses trois noeuds, ses trois arêtes et ses trois voisins, à gauche, et ce même élément est raffiné en 4 sous-éléments, à droite. L'arête 1 est une arête frontière et s'avère être subdivisée lors du processus d'adaptation.

Dès qu'un élément est découpé ou transformé en 4 sous-éléments, de nouveaux noeuds sont créés au centre de chaque arête. Ce sont les seuls et uniques cas où une arête frontière peut être subdivisée. Dans ces deux configurations, nous prenons le soin de vérifier si l'arête se trouve bien sur une frontière si c'est le cas, nous renseignons aux enfants le tag de la nouvelle arête. Pour illustrer notre explication, considérons le cas schématisé à la figure (Fig. 4.5). L'élément se situe sur une frontière et l'arête 1 est subdivisée. L'information de cette subdivision est transmise à l'enfant 1 sur son arête 1 et à l'enfant 2 sur son arête 0. Autrement dit, le tag de l'arête frontière est inséré à l'entrée 1 du tableau **boundaries** pour l'enfant 1 et un nouveau tag est ajouté à l'entrée 0 du tableau pour l'enfant 2. Par la même occasion, nous incrémentons de 1 le compteur des arêtes frontières. Pour ce qui est des découpages en 2, en 3 ou en 4 de l'étape 3(b), ils ne vont jamais générer une nouvelle frontière et l'information contenue au niveau des parents est alors suffisante pour pouvoir reconstituer toutes les frontières du domaine.

4.2 Validation

Cette validation est effectuée sur une nouvelle version du code MigFlow ce qui explique les différences dans les valeurs de la vitesse caractéristique et de la pression caractéristique. Nous avons dû recalibrer ces valeurs à partir de la solution de référence. Nous avons ainsi obtenu 0.015 m/s comme nouvelle vitesse caractéristique et 4.32 Pa comme nouvelle pression caractéristique (Fig. 4.6).

Pour cette validation, nous nous sommes de nouveau intéressés au cas du nuage de grains chutant dans un fluide visqueux. Nous avons en effet noté que l'adaptation locale fonctionnait très bien sur ce cas test à la fois en terme de rapidité et de précision

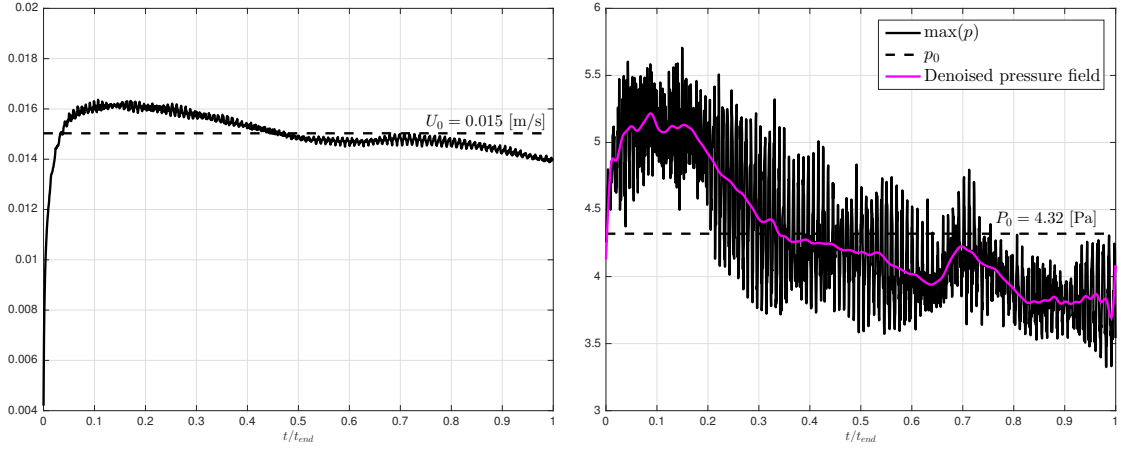


FIGURE 4.6 – Evolution du maximum de la vitesse du fluide et de la pression du fluide des-quelles la vitesse et la pression caractéristiques sont extraites.

de calcul. La configuration est la même que celle décrite au tableau (Tab.3.1). La position du centre de masse pour le maillage uniforme et pour le maillage adapté localement sera extraite des deux simulations ainsi que la vitesse du nuage. Le temps final de la simulation fut fixé à $t_{end} = 10$ pour un pas de temps fixé à $5 \cdot 10^{-3}$, ce qui fait un total de 2000 itérations pour le cas à maillage fixe et 4000 itérations pour le cas à maillage adapté avec la stratégie adaptative en avance. De plus, pour la stratégie adaptative, nous avons utilisé une erreur prescrite de 10^{-6} et un intervalle d'adaptation de $\Delta t = 5$. Nous comparons par la même occasion les résultats que nous aurions obtenus sur un maillage adapté globalement. De cette manière, nous pourrions juger de l'ampleur du phénomène de diffusion.

Le tableau (Tab. 4.2) affiche les temps d'exécution de deux approches ainsi que la position finale du centre de masse. Encore une fois, nous notons que la stratégie adaptative est plus rapide malgré que le double d'itérations soit effectué. La précision des deux solutions est très similaire, l'erreur sur la position est de l'ordre de $\mathcal{O}(10^{-3})$.

	$h_{\min}$ [m]	Nombre d'éléments	CPU	$Y_c(t^* = 10)$
Maillage fin (A)	$10r_p$	58580	48 min 11 s	-0.207
Maillage adapté localement (B)	$10r_p$	$5792 < n_{elt} < 13084$	19 min 53 s	-0.206
Maillage adapté globalement (C)	$10r_p$	$4055 < n_{elt} < 6153$	9 min 14	-0.205

TABLE 4.2 – Le temps de simulation sur maillage adapté est réduit sans perdre en précision par rapport au maillage raffiné. L'erreur entre les positions des centres de masse pour les trois maillages est du même ordre en $\mathcal{O}(10^{-3})$. Le temps d'adaptation locale est cependant deux fois plus élevé que le cas global.

Nous observons également à quel point les deux courbes de vitesse du nuage, entre le maillage A et B, sont proches l'une de l'autre appuyant encore plus l'observation de la précision de la solution obtenue par l'adaptation locale. Les pentes de ces deux courbes convergent vers la même valeur de -0.2 comme indiqué à la figure (Fig.

4.7). Nous constatons que la courbe bleue, représentant la vitesse sur le maillage C, est également très proche des deux autres courbes mais elle oscille plus fortement, comme observé dans le zoom à la figure (Fig. 4.7). Ces oscillations sont le reflet du phénomène de diffusion introduit par l'adaptation globale que nous cherchions à limiter avec l'adaptation locale. Comme nous ne déraffinons pas le maillage, aucune diffusion n'est introduite avec l'adaptation locale. Prenons l'exemple simplifié d'un élément ayant été raffiné de la figure (Fig. 4.8).

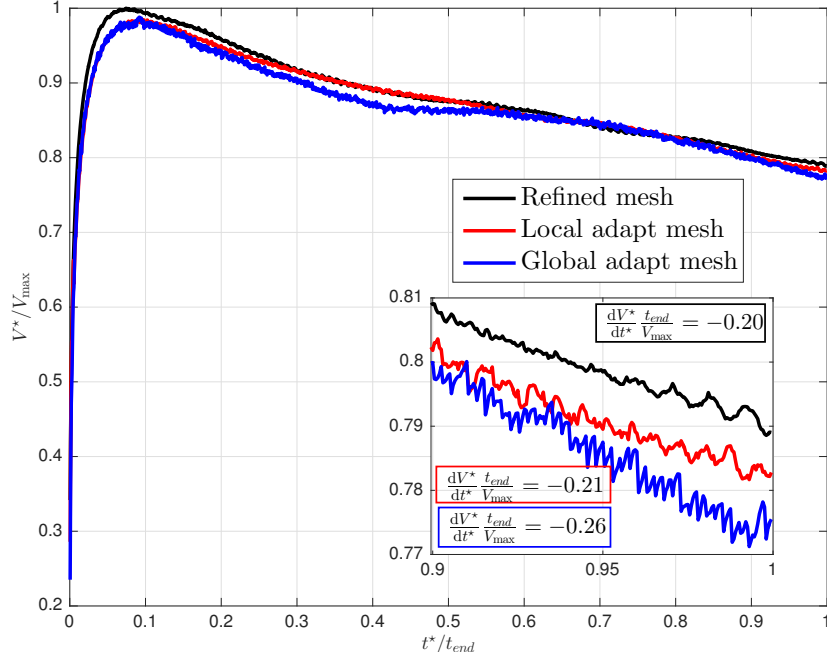


FIGURE 4.7 – Evolution de la vitesse du nuage pour les simulations avec maillage fixe et maillage adapté localement. Les deux courbes sont très proches l'une de l'autre et convergent vers les mêmes valeurs.

Lorsque l'élément est découpé en quatre, trois noeuds sont ajoutés et l'évaluation de la solution sur ces trois noeuds ne provoquent aucune diffusion numérique. La solution de l'ancien maillage n'est pas modifiée sur le nouveau.

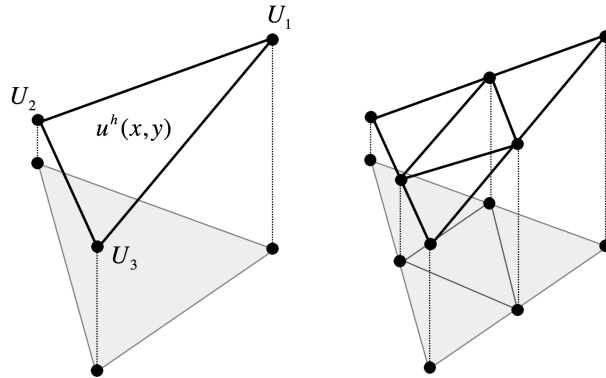


FIGURE 4.8 – Interpolation de la solution u^h sur le nouveau maillage par le biais des fonctions de forme locales.

Nous désirons également observer la position du maillage adapté par rapport au position des grains dans le domaine, figure (Fig. 4.9). Grâce à la stratégie adaptative en avance, l'interface grains/fluide est bien capturée et les grains restent dans la zone la plus raffinée. Le déraffinement n'a pas été implémenté impliquant qu'une zone non négligeable du domaine reste raffinée. Elle a malgré tout un intérêt pour la colonne de grains qui est de fait bien capturée.

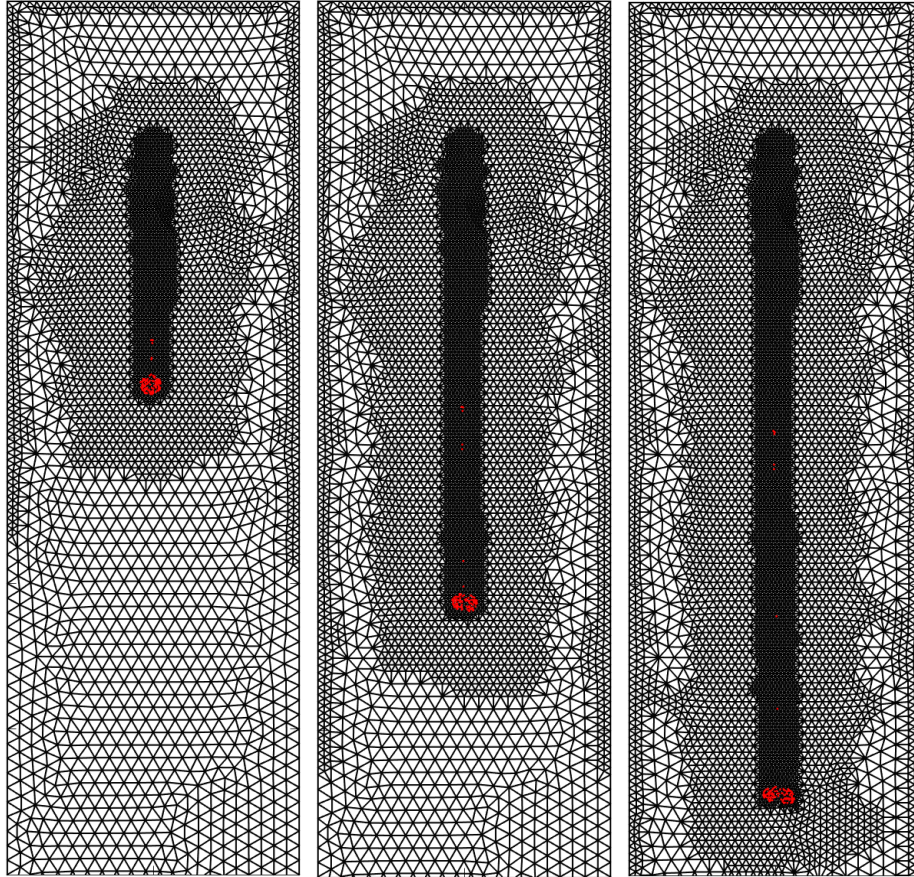


FIGURE 4.9 – Evolution de la position du nuage de grains via à vis de la position du maillage adapté localement pour les temps $t^* = 5$, $t^* = 10$ et $t^* = 15$.

4.3 Conclusion concernant l'adaptation locale

L'adaptation locale sans déraffinement n'introduit aucune erreur de diffusion numérique, la solution se comporte donc comme sur un maillage uniformément raffiné. La solution ainsi obtenue sur le maillage adapté localement est toute aussi bonne que celle obtenue sur le maillage raffiné en termes de position du centre de masse et de la vitesse de chute du nuage. Le temps CPU est nettement inférieur à celui sur le maillage raffiné. Néanmoins, il est deux fois plus grand que le temps CPU mesuré avec l'adaptation globale. Il sera donc nécessaire d'optimiser les fonctions gérant l'adaptation locale et peut-être aussi minimiser le coût de stockage des informations liées aux structures arborescentes.

A ce stade, il ne reste que le déraffinement à implémenter afin d'obtenir une complète adaptation locale. L'étape de déraffinement est liée à la suppression d'une série d'éléments et de noeuds contenus dans le maillage. Une fois tous les noeuds retirés, il faudra renuméroter tous les noeuds restant dans le maillage afin de garder une continuité dans les index. Il sera également nécessaire de garder une transition graduelle entre les éléments devant être raffinés et ceux devant être déraffinés. La gestion des pointeurs et la libération de la mémoire devra être effectué avec soin surtout lors de la suppression d'enfants.

Ce travail avait pour but dans un premier temps de développer un estimateur couplé à une stratégie adaptative qui puisse être implémentée dans un code existant, le code MigFlow. Dans un deuxième temps, nous nous sommes intéressés à l'adaptation locale ayant comme objectif de ne raffiner que localement le maillage par des subdivisions successives ne dégénérant pas les éléments.

L'état de l'art sur les estimateurs d'erreur nous a permis de sélectionner deux estimateurs pour une étude théorique basée sur un problème modèle : l'estimateur *a posteriori* résiduel et l'estimateur par reconstruction de gradient. A la suite de cette étude développée dans le chapitre 2, nous avons choisi le deuxième estimateur étant le plus simple à implémenter dans un code existant. Il fournissait également des résultats tout aussi concluants que l'estimateur résiduel dans les deux problèmes modèles étudiés.

Dans le chapitre 3, nous nous sommes tournés vers les écoulements granulaires. La stratégie adaptative fut validée sur trois cas tests : le nuage de grains chutant dans un fluide visqueux et une sédimentation d'une suspension mono-dispersée en tubes droits et inclinés. Cette stratégie utilise l'estimateur par reconstruction de gradient pour ensuite déterminer la taille de chaque élément dans le nouveau maillage. La validation fut concluante sur le cas du nuage de grains où le maillage avançait au même rythme que le nuage tombant tout en raffinant la colonne de grains se formant à l'arrière du nuage. Nous avons également souligné qu'un estimateur en pression était suffisant pour capturer cette colonne mais que l'estimateur en vitesse n'était au contraire pas capable de repérer les variations de vitesses dans la colonne de grains. Dans le cas de la sédimentation en tube droit, la stratégie adaptative a moins fait ses preuves. En effet, le temps de calcul était tout aussi long que pour un maillage raffiné. Bien qu'une grande partie des grains tombait rapidement dans le fond du récipient, une plus petite fraction était transportée au travers du domaine par le champ de vitesse du fluide encore bien présent, champ de vitesse généré par les grains chutant. Autrement dit, le champ de vitesse observait des variations dans la quasi-totalité du domaine. Ces variations sont repérées par l'estimateur qui raffine

donc le maillage dans son entièreté. Lors de l'étude de l'effet Boycott, sédimentation en tube incliné, la sédimentation est améliorée. Dans cette configuration, les grains se retrouvaient collés à la paroi inférieure et glissaient jusque dans le fond du tube pour y former la couche de sédiments. L'estimateur a donc raffiné principalement le long de cette paroi et l'autre moitié du tube qui n'observait que de très faibles variations de vitesses et de pression n'est, quant à elle, pratiquement pas raffinée.

Le chapitre 4 fut consacré à l'adaptation locale. Précédemment, la reconstruction du maillage était effectuée par le logiciel GMSH, nous parlions alors d'adaptation globale du maillage. Le maillage était fréquemment modifié au profit d'un tout autre maillage pour lequel la position des éléments était différente. Cette modification topologique du maillage entraînait de la diffusion numérique lors de l'interpolation de la solution sur le nouveau maillage. Afin de minimiser cet effet, nous avons pris le parti d'adapter localement le maillage. Dans la littérature, nous avons observé deux techniques : *the newest vertex bisection* et la méthode de quadrisection de Bank et Sherman. Nous nous sommes penchés sur cette deuxième technique qui octroyait un découpage en 2 des éléments permettant la gestion des noeuds flottants. A la différence de leurs travaux, nous avons également accordé des découpes en 3 qui devaient tout comme les découpes en 2 être supprimées afin de ne pas dégénérer le maillage après une utilisation répétée. Seul le raffinement a été implémenté et donc aucune erreur de diffusion numérique ne fut introduite. Nous avons pu observer la solution sur le maillage adapté localement et globalement et plus précisément l'évolution de la vitesse de chute du nuage. La courbe de vitesse obtenue sur le maillage adapté globalement présentait des oscillations plus fortes mettant en évidence les erreurs de diffusion introduites par la modification topologique du maillage.

A ce stade, quelques points peuvent encore être améliorés. Premièrement, concernant la stratégie adaptative, il serait nécessaire de fournir une borne sur le nombre d'éléments et non pas une erreur. De cette façon, nous pourrions imposer le nombre maximal d'éléments à ne pas dépasser au cours d'une simulation. Actuellement, il est encore possible que le nombre d'éléments monte en flèche au cours d'une simulation malgré le fait qu'une erreur prescrite soit fournie ainsi que la taille minimale des éléments. Nous voudrions surtout éviter que le nombre d'éléments ne tende pas vers le nombre d'éléments du maillage uniformément raffiné. Deuxièmement, concernant la vitesse et la pression caractéristiques, il serait intéressant de les extraire directement de quelques itérations au début de la simulation. Nous pourrions moyenner le champ de pression et la norme du champ de vitesse au cours de ces quelques itérations. Les valeurs ainsi obtenues seraient utilisées par le code comme pression et vitesse caractéristiques, respectivement. Troisièmement, concernant l'adaptation locale, le déraffinement doit encore être implémenté. Il s'agit là d'une partie assez technique de l'adaptation locale qui consiste à supprimer des éléments et des noeuds du maillage. Il est à noter que le déraffinement va, contrairement au raffinement, générer de la diffusion numérique lors de l'interpolation car de l'information est perdue d'un maillage à l'autre. Une mesure de cette diffusion numérique devra être effectuée afin de valider l'approche de l'adaptation locale.

Enfin, je terminerais cette conclusion par les compétences techniques et transversales que j'ai pu travailler dans ce mémoire. En effet, ce dernier est un travail de longue haleine qui se déroule sur une année complète. Il demande une certaine organisation, de la persévérance et de la rigueur. Chaque semaine, une réunion était organisée durant laquelle je pouvais communiquer mes idées, les défendre mais aussi recevoir des conseils pour continuer à avancer dans mon travail. Pouvoir communiquer oralement et par écrit ses résultats de façon claire et efficace est une des compétences transversales qui est énormément travaillée dans le cursus des ingénieurs civils à l'école polytechnique de Louvain-la-Neuve.

Le mémoire sert aussi à développer des compétences techniques. J'ai pu d'une part approfondir mes connaissances concernant les méthodes en éléments finis dont la théorie m'a servi à créer un estimateur résiduel. D'autre part, j'ai pu améliorer mes compétences dans le langage C en manipulant des structures telles que des tableaux d'arbres.

Le développement de ces compétences me permet aujourd'hui de fournir à l'équipe MigFlow un estimateur efficace par reconstruction de gradient et une implémentation de l'adaptation locale.

- [Acrivos et Herbolzheimer, 1979] A. Acrivos and E. Herbolzheimer, *Enhanced sedimentation in settling tanks with inclined walls*, Journal of Fluid Mechanics, Vol. 92, part 3, pp. 435-457, 1979.
- [Adachi, Kiriyaama, et Toshioka, 1978] K. Adachi, S. Kiriyaama and N. Yoshioka, *The behavior of a swarm of particles moving in a viscous fluid*, Chemical Engineering Science, Vol. 33, pp. 115-121, 1978.
- [Adjerid et Flaherty, 1999] S. Adjerid and J. E. Flaherty, *A posteriori error estimator for the finite element method-of-lines solution of parabolic problems*, Mathematical Models and Methods in Applied Sciences, Vol. 09, No. 02, pp. 261-286, 1999.
- [Ainsworth, 1994] M. Ainsworth, *The performance of Bank-Weiser's error estimator for quadrilateral finite elements*, Numerical Methods PDE, Vol. 10, pp. 609-623, 1994.
- [Ainsworth et Oden, 1997] M. Ainsworth and J.T. Oden, *A posteriori error estimation in finite element analysis*, Computer methods in applied mechanics and engineering, Vol. 142, pp. 1-88, 1997.
- [Ardekani et Rangel, 2008] A. M. Ardekani and R. H. Rangel, *Numerical investigation of particle-particle and particle-wall collisions in a viscous fluid*, Journal of Fluid Mechanics, Vol. 596, pp. 437-466, 2008.
- [Babuska et Rheinboldt, 1979] I. Babuska and W.E. Rheinboldt, *Analysis of optimal finite element meshes in $\mathbb{R}^1$* , Mathematics of Computation, Vol. 33, pp. 435-463, 1979.
- [Babuska et Rheinboldt, 1981] I. Babuska and W.C. Rheinboldt, *A posteriori error analysis of finite element solutions for one dimensional problems*, SIAM Journal on Numerical Analysis, Vol. 18, pp. 565-589, 1981.
- [Babuska, 1994] I. Babuska, T. Strouboulis, C.S. Upadhyay and S.K. Gangaraj, *A model study of the quality of a posteriori estimators for linear elliptic problems error estimation in the interior of patchwise uniform grids of triangles*, Computer Methods in Applied Mechanics and Engineering, Vol. 114, pp. 307-378, 1994.
- [Babuska et al., 1994] I. Babuska, T. Strouboulis, C.S. Upadhyay, S.K. Gangaraj and K. Copps, *Validation of a posteriori error estimators by a numerical approach*,

- International Journal for Numerical Methods in Engineering, Vol. 37, pp. 1073-1123, 1994.
- [Babuska, et al., 1986] I. Babuska, O.C. Zienkiewicz, J. Gago and E.A. Oliveira, *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, Wiley, 1986.
- [Bank et Sherman, 1981] R.E. Bank and A.H. Sherman, *An adaptive multilevel method for elliptic boundary value problems*, Computing, Vol. 26, pp. 91-105, 1981.
- [Bank et Weiser, 1985] R.E. Bank and A. Weiser, *Some a posteriori error estimators for elliptic partial differential equations*, Mathematics of Computation, Vol. 44, pp. 283-301, 1985.
- [Bieterman, 1982] M. Bieterman and I. Babuska, *The finite element method for parabolic equations. Part I : A posteriori error estimation*, Numerische Mathematik, Vol. 40, pp. 339-371, 1982.
- [Bieterman et Babuska, 1982] M. Bieterman and I. Babuska, *The finite element method for parabolic equations. Part II : A posteriori error estimation*, Numerische Mathematik, Vol. 40, pp. 373-406, 1982.
- [Binev, 2004] P. Binev, W. Dahmen, R. DeVore, *Adaptive Finite Element Methods with convergence rates*, Springer, Numerische Mathematik, Vol. 97, 2004.
- [Brändle de Motta, 2013] J. C. Brändle de Motta, W-P. Breugem, B. Gazanion, J-L. Estivalezes, S. Vincent, and E. Climent, *Numerical modelling of finite-size particle collisions in a viscous fluid*, Physics of Fluids, Vol. 25, 083302, 2013.
- [Constant et al., 2017] M. Constant, F. Dubois, J. Lambrechts and V. Legat, *An hybrid multiscale model for immersed granular flows*, EPJ Web of Conferences 140, 09021, 2017.
- [Constant et al., 2018] M. Constant, F. Dubois, J. Lambrechts and V. Legat, *Implementation of an unresolved stabilised FEM-DEM model to solve immersed granular flows*, Computational Particle Mechanics, Vol. 6, pp. 213-226, 2018.
- [Cundall et Strack, 1979] P. A. Cundall, O. D. L. Strack, *A discrete numerical model for granular assemblies*, Géotechnique, Vol. 29(1), pp. 47-65, 1979.
- [DallaValle, 1943] J. M. DallaValle, A. Klemin, *Micromeritics : the technology of the particles*, Pitman Publishing Corporation, 1943.
- [Dance et Maxey, 2003] S. L. Dance and M. R. Maxey, *Incorporation of lubrication effects into the force-coupling method for particulate two-phase flow*, Journal of computational Physics, Vol. 189, pp. 212-238, 2003.
- [David, 1985] R.H. David, *Sedimentation of noncolloidal particles at low Reynolds number*, Annual Review of Fluid Mechanics, Vol. 17, pp. 91-118, 1985.
- [Demkowicz, et al., 1984] L. Demkowicz, J.T. Oden and T. Strouboulis *Adaptive finite elements for flow problems with moving boundaries. Part I : Variational principles and a posteriori error estimates*, Computer Methods in Applied Mechanics and Engineering, Vol. 46, pp. 217-251, 1984.
- [Demkowicz, 1985] L. Demkowicz, Ph. Devloo and J.T. Oden, *On an h-type mesh refinement strategy based on minimization of interpolation errors*, Computer Methods in Applied Mechanics and Engineering, Vol. 53, pp. 67-89, 1985.

- [Dörfler, 1996] W. Dörfler, *A convergent adaptive algorithm for Poisson's equation*, SIAM Journal on Numerical Analysis, Vol. 33, pp. 1106–1124, 1996.
- [Di Felice et Rotondi, 2012] R. Di Felice, M. Rotondi, *Fluid-particle Drag Force in Binary-solid Suspensions*, International Journal of Chemical Reactor Engineering, Vol. 10, 2012.
- [Gidaspow, 2012] D. Gidaspow, *Multiphase Flow and Fluidization : Continuum and Kinetic Theory Descriptions*, Academic Press, 2012.
- [Guerra et Becker, 1978] F.M. Guerra and E.B. Becker, *Finite element analysis for the adaptive method of rezoning*, The University of Texas at Austin, TICOM Rept., pp. 78-13, 1978.
- [Hung, 2001] O. Hung, S. G. Evans, M. J. Bovis, and J. N. Hutchinson, *A review of the classification of landslides of the flow type*, Environmental and Engineering Geoscience, Vol. 7(3), pp. 221–238, 2001.
- [Iverson, 2000] R.M. Iverson, *Landslide triggering by rain infiltration*, Water Resources Research, Vol. 36(7), pp. 1897–1910, 2000.
- [Jean, 1999] M. Jean, *The non-smooth contact dynamics method*, Computer Methods in Applied Mechanics and Engineering, Vol. 177, pp. 235–257, 1999.
- [Kelly et al., 1983] D.W. Kelly, J.R. Gago, O.C. Zienkiewicz and I. Babuska, *A posteriori error analysis and adaptive processes in the finite element method : Part 1-Error analysis*, International Journal for Numerical Method in Engineering, Vol. 19, pp. 1593-1619, 1983.
- [Machu et al., 2001] G. Machu, W. Meile, L.C. Nitsche and U. Schaflinger, *Coalescence, torus formation and break-up of sedimenting clouds : experiments and computer simulations*, Journal of Fluid Mechanics, Vol. 447, pp. 299–336, 2001.
- [Masson et al., 2006] D. G. Masson, C. B. Harbitz, R. B. Wynn, G. Pedersen, and F. Lovholt, *Submarine landslides : processes, triggers and hazard prediction*, Philosophical Transactions of the Royal Society A, Vol. 364(1845), pp. 2009–2039, 2006.
- [Metzger, et et al, 2006] B. Metzger, M. Nicolas and E. Guazzelli, *Falling clouds of particles in viscous fluids*, Journal of Fluid Mechanics, Vol. 580, pp. 283-301, 2006.
- [Mitchell, 2016] W.F. Mitchell, *30 Years of Newest Vertex Bisection*, Journal of Numerical Analysis, Industrial and Applied Mathematics, 2016.
- [Morin, Nochetto et Sieber, 2000] P. Morin, R. Nochetto, K. Siebert, *Data Oscillation and Convergence of Adaptive FEM*, SIAM Journal on Numerical Analysis, Vol. 38, pp. 466-488, 2000.
- [McNamara, 1992] S. McNamara and W. R. Young, *Inelastic collapse and clumping in a one-dimensional granular medium*, Physics of Fluids, Vol. 4, pp. 496–504, 1992.
- [Nitsche et Batchelor, 1997] J. M. Nitsche and G. K. Batchelor *Break-up of a falling cloud containing dispersed particles*, Journal of Fluid Mechanics, Vol. 340, pp. 161–175, 1997.

- [Oden, 1989] J.T. Oden, L. Demkowicz, W. Rachowicz and T.A. Westermann, *Toward a universal h-p adaptive finite element strategy. Part2 : A posteriori error estimation*, Computer Methods in Applied Mechanics and Engineering, Vol. 77, pp. 113-180, 1989.
- [Palma et al, 2016] S. Palma, C. F. Ihle A. Tamburrino and S. B. Dalziel, *Particle organization after viscous sedimentation in tilted containers*, Physics of Fluids, Vol. 28, 073304, 2016.
- [Peskin, 2000] C. S. Peskin, *The immersed boundary method*, Courant Institut of Mathematic Sciences New York University, 2000.
- [Petersen, 1891] J. Petersen, *Die Theorie der regulären Graphen*, Acta Math, Vol.15, 193-220, 1891.
- [Radjaï, 2011] F. Radjaï, F. Dubois, et al. *Discrete-element modeling of granular materials*, 2011.
- [Rank et Zienkiewicz, 1987] E. Rank and O.C. Zienkiewicz, *A simple error estimator in the finite element method*, Communication in Applied Numerical Methods, Vol. 3, pp. 243-249, 1987.
- [Rzadkiewicz et Zhu, 1996] S. A. Rzadkiewicz, C. Mariotti, P. Heinrich, *Modelling of submarines landslides and generated water waves* Physics and Chemistry of the Earth, Vol. 21(1-2), pp. 1-12, 1996.
- [Tong, 1969] P. Tong, *Exact Solution of Certain Problems by Finite-Element Method*, AIAA Journal, Vol. 7, pp. 178-180, 1969.
- [Verfürth, 1989] R. Verfürth, *A posteriori error estimators for the Stokes equations*, Springer, Numerische Mathematik, Vol. 55, pp. 309-325, 1989.
- [Verfürth, 1996] R. Verfürth, *A Review of a Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Teubner-Wiley, 1996.
- [Wen et Yu, 1966] C Y Wen, H Y Yu, *Chemical Engineering Progress Symposium Series*, Mechanics of fluidization, Vol. 62, pp. 100-111, 1966.
- [Zienkiewicz et Zhu, 1987] O.C. Zienkiewicz and J.Z. Zhu, *A simple error estimator and adaptive procedure for practical engineering analysis*, International Journal for Numerical Methods in Engineering, Vol. 24, pp. 337-357, 1987.

Cette annexe est dédiée au calcul complet de l'estimateur résiduel pour le problème modèle (Eq.2.1) en deux dimensions. Nous décrivons étape par étape les intégrales que nous devons calculer afin d'obtenir la valeur de U_+ suivant l'expression (Eq.2.8). Nous nous concentrons sur les développements liés aux termes : $\langle f\phi_+ \rangle$, $\langle \nabla\phi_+(\nabla u^h) \rangle$ et $\langle (\nabla\phi_+)(\nabla\phi_+) \rangle$.

Calcul du terme $\langle f\phi_+ \rangle$ La fonction f est connue *a priori* et il peut s'agir de n'importe quelle fonction carrée intégrable au sens de l'intégrale de Lebesgue. Cette intégrale devra dans tous les cas être calculée numériquement en utilisant une quadrature de Gauss du troisième ordre défini par quatre points sur le triangle parent. Il est en effet nécessaire de stipuler que toutes les intégrales devront être évaluées sur l'élément parent par le biais de l'isomorphisme.

Calcul du terme $\langle \nabla\phi_+(\nabla u^h) \rangle$ L'approximation en éléments finis, notée u^h est linéaire par morceaux et donc son gradient est une constante. Ce terme n'est rien d'autre que l'intégrale du gradient de la fonction bulle intérieure. Nous obtenons donc deux valeurs notées M et N étant l'intégrale de la dérivée selon x de la fonction bulle et de la dérivée selon y de la fonction bulle, respectivement.

$$M = \int_0^1 \int_0^{1-\xi} \frac{\partial\phi_+}{\partial x}(\xi, \eta) d\eta d\xi = \int_0^1 \int_0^{1-\xi} (1 - 2\xi - \eta) \eta d\eta d\xi$$

$$N = \int_0^1 \int_0^{1-\xi} \frac{\partial\phi_+}{\partial y}(\xi, \eta) d\eta d\xi = \int_0^1 \int_0^{1-\xi} (1 - \xi - 2\eta) \xi d\eta d\xi$$

Il s'avère que ces deux intégrales sont exactement égales à zéro. Il semble donc que pour le problème modèle considéré, la valeur de U_+ ne dépend pas de la solution u^h . Nous pouvons donc retirer ce terme de l'expression de U_+ ce qui nous amène finalement au calcul du dénominateur.

Calcul du dénominateur $\langle (\nabla\phi_+)(\nabla\phi_+) \rangle$ Il est important d'exprimer cette intégrale sur l'élément parent en tenant bien compte du jacobien de la transformation.

L'élément quelconque est noté K et l'élément parent est, quant à lui, noté $\hat{K}$. Nous obtenons donc,

$$\begin{aligned}\langle \nabla \phi_+ \nabla \phi_+ \rangle &= \int_K \nabla \phi_+ \nabla \phi_+ d\Omega \\ &= \int_K \frac{\partial \phi_+}{\partial x} \frac{\partial \phi_+}{\partial x} + \frac{\partial \phi_+}{\partial y} \frac{\partial \phi_+}{\partial y} d\Omega \\ &= \int_{\hat{K}} \left(\frac{\partial \phi_+}{\partial x}(\xi, \eta) \frac{\partial \phi_+}{\partial x}(\xi, \eta) + \frac{\partial \phi_+}{\partial y}(\xi, \eta) \frac{\partial \phi_+}{\partial y}(\xi, \eta) \right) J_e(\xi, \eta) d\hat{\Omega}.\end{aligned}$$

Cette intégrale ne contient que des polynômes du quatrième ordre ce qui rend le calcul de l'intégrale assez simple et il n'est donc pas nécessaire d'effectuer une intégration numérique. Nous devons dans un premier temps évaluer le gradient de la fonction bulle puis dans un second temps évaluer l'inverse du gradient de la transformation isomorphe $\frac{\partial \mathbf{x}}{\partial \xi}(\xi)$. Cette transformation est définie comme,

$$\mathbf{x}(\xi) = \sum_{i=1}^3 \mathbf{X}_i \phi_i(\xi),$$

où ϕ_i sont les fonctions de formes. Il est important de souligner que l'inverse du gradient de cette transformation existe.

Sachant que la fonction bulle est définie sur l'espace ξ et que nous voulons évaluer son gradient selon l'espace $\mathbf{x}$, nous devons utiliser la définition de la transformation entre l'espace de calcul et l'espace physique pour obtenir,

$$\frac{\partial \phi_+}{\partial \mathbf{x}} = \frac{\partial \phi_+}{\partial \xi} \frac{\partial \xi}{\partial \mathbf{x}} = \frac{\partial \phi_+}{\partial \xi} \left(\frac{\partial \mathbf{x}}{\partial \xi} \right)^{-1}$$

Cette dernière expression requiert la définition de l'inverse du gradient de la transformation qui est donnée par,

$$\left(\frac{\partial \mathbf{x}}{\partial \xi}(\xi) \right)^{-1} = \frac{1}{J_e} \begin{bmatrix} (Y_3 - Y_1) & -(X_3 - X_1) \\ -(Y_2 - Y_1) & (X_2 - X_1) \end{bmatrix}$$

où, le jacobien J_e vaut $(X_2 - X_1)(Y_3 - Y_1) - (X_3 - X_1)(Y_2 - Y_1)$.

En développant l'intégrale de surface sur l'élément de référence, nous observons que seules deux intégrales doivent être évaluées par symétrie. Nous obtenons que l'intégrale de $\phi_x^+ \phi_x^+$ et celle de $\phi_y^+ \phi_y^+$ sur $\hat{K}$ sont égales et valent 1/180 alors que l'intégrale de $\phi_x^+ \phi_y^+$ est deux fois plus petite et vaut 1/360. En combinant ces différents résultats, nous pouvons écrire une expression analytique de l'intégrale du dénominateur comme

$$\begin{aligned}\langle \nabla \phi_+ \nabla \phi_+ \rangle &= \frac{1}{J_e} \left((Y_3 - Y_1)^2 I_1 - (Y_3 - Y_1)(Y_2 - Y_1) I_2 + (Y_2 - Y_1)^2 I_3 \right) \\ &\quad + \left((X_3 - X_1)^2 I_1 - (X_3 - X_1)(X_2 - X_1) I_2 + (X_2 - X_1)^2 I_3 \right),\end{aligned}$$

où (X_1, Y_1) , (X_2, Y_2) et (X_3, Y_3) sont les trois coordonnées des noeuds d'un élément triangulaire K . Finalement, l'estimateur est évalué localement, c'est-à-dire sur chaque élément du maillage. Il est donc nécessaire de calculer la norme L_2 de l'erreur approximée e^h valant,

$$\kappa^2 = \|e^h\|_{L_2(K)}^2 = U_+^2 \int_K (\phi_+)^2 d\Omega = U_+^2 \int_{\hat{K}} (\phi_+)^2 d\hat{\Omega} \quad (5.1)$$

Nous rappelons que l'estimateur résiduel développé ici est un estimateur local, il ne prend pas en compte l'influence des voisins. Il peut être montré que pour un problème elliptique, la diffusion supprime complètement la propagation des erreurs. Cependant, pour les équations de Navier-Stokes qui contiennent un terme de convection, les erreurs commises en un endroit du maillage peuvent être convectées sur une certaine distance. C'est pour cette raison que dans la littérature, nous trouvons des estimateurs évalués sur des *patch*. La norme L_2 de la fonction bulle est aisé à calculer sur base de tous les résultats précédents et nous obtenons donc l'expression finale de notre estimateur dans le cas bidimensionnel,

$$\kappa = J_e \frac{|U_+|}{12\sqrt{35}}$$

Cette annexe est dédiée au développement d'un estimateur résiduel basé sur le champs de vitesse uniquement. Pour ce faire, nous utilisons les équations d'un écoulement granulaire immergé tel que discutées dans le chapitre 3. En utilisant l'équation (3.1), nous pouvons construire un estimateur *a posteriori* résiduel pour le champ de vitesse $\mathbf{u}$. Nous allons de nouveau utiliser la théorie développée autour de la fonction bulle intérieure. Avec cette théorie, une nouvelle solution, notée $\mathbf{u}_+^h$ peut être obtenue en résolvant localement un problème aux éléments finis utilisant la condition de Galerkin. Cette condition permet d'obtenir un système dont l'inconnue est un vecteur $\mathbf{U}_+$ de $\mathbb{R}^2$ qui dépend du temps mais pas de l'espace $\frac{\partial \mathbf{U}_+}{\partial t} \neq \mathbf{0}$ et $\frac{\partial \mathbf{U}_+}{\partial x} = \frac{\partial \mathbf{U}_+}{\partial y} = 0$. La condition d'orthogonalité s'écrit donc comme,

$$\langle \mathbf{r}_h^+ \phi_+ \rangle = \mathbf{0}$$

dans laquelle le résidu de l'équation est défini comme,

$$\begin{aligned} \mathbf{r}_h^+ &= \rho \left(\frac{\partial \mathbf{u}_h^+}{\partial t} + \nabla \cdot \frac{\mathbf{u}_h^+ \mathbf{u}_h^+}{\phi} \right) - \nabla \cdot [2\mu\phi \mathbf{d}(\mathbf{u}_h^+) - p_h \mathbf{I}] + \mathbf{f} + \phi \rho \mathbf{g} \\ &= \mathbf{r}_h + \rho \left(\frac{\partial \mathbf{U}_+ \phi_+}{\partial t} + \nabla \cdot \frac{\mathbf{u}_h (\mathbf{U}_+ \phi_+)}{\phi} + \nabla \cdot \frac{(\mathbf{U}_+ \phi_+) \mathbf{u}_h}{\phi} + \nabla \cdot \frac{(\mathbf{U}_+ \phi_+) (\mathbf{U}_+ \phi_+)}{\phi} \right) \\ &\quad - \nabla \cdot [2\mu\phi \mathbf{d}(\mathbf{U}_+ \phi_+)] \end{aligned}$$

Nous supposons que le vecteur $\mathbf{U}_+$ est une faible perturbation de l'approximation en éléments finis $\mathbf{u}_h$. De par cette hypothèse, nous pouvons linéariser le résidu en négligeant le terme quadratique. En appliquant le théorème de la divergence et le fait que la fonction bulle ϕ_+ s'annule sur les frontières de l'élément, nous obtenons le système suivant à résoudre pour le vecteur $\mathbf{U}_+$,

$$\begin{aligned} \rho \left\langle \frac{d\mathbf{U}_+}{dt} \phi_+ \phi_+ \right\rangle - \rho \left\langle \frac{\phi_+}{\phi} \mathbf{u}_h [\mathbf{U}_+ \cdot \nabla \phi_+] \right\rangle \\ - \rho \left\langle \frac{\phi_+}{\phi} [\mathbf{u}_h \cdot \nabla \phi_+] \right\rangle \mathbf{U}_+ \\ + 2\mu \left\langle \phi \nabla \left(\frac{\phi_+}{\phi} \right) \cdot \nabla \phi_+ \right\rangle \mathbf{U}_+ = -\langle \mathbf{r}_h \phi_+ \rangle. \end{aligned}$$

Nous décidons de discrétiser la dérivée temporelle grâce à un schéma d'Euler explicite au premier ordre en utilisant les termes $\mathbf{U}_+^n$ and $\mathbf{U}_+^{n+1}$. Cependant, le terme en n n'est pas connu, du moins lors de la première itération. Nous pouvons supposer que sa valeur est nulle au démarrage. Cependant, en avançant dans la simulation, la connexion entre maillage actuel et le maillage précédent est perdue car le maillage raffiné est complètement modifié et ne contient pas les mêmes éléments. Il faudrait garder une structure qui permettrait de garder en mémoire ce lien. De plus, nous ne raffinons pas à chaque itération ce qui entraîne un décalage entre les solutions obtenues. Il n'est pas nécessaire de garder en mémoire une information d'un temps passé. En conclusion, le terme en n est mis à zéro.

L'approximation en éléments finis peut se réécrire sous la forme $\mathbf{u}_h = u_h \hat{\mathbf{e}}_x + v_h \hat{\mathbf{e}}_y$ avec deux composantes réelles qui dépendent du temps et de l'espace. Nous pouvons ainsi obtenir un système linéaire explicite pour le vecteur $\mathbf{U}_+$.

$$\begin{bmatrix} G(\phi, \mathbf{u}_h) - F(\hat{\mathbf{e}}_x, \hat{\mathbf{e}}_x) & -F(\hat{\mathbf{e}}_x, \hat{\mathbf{e}}_y) \\ -F(\hat{\mathbf{e}}_y, \hat{\mathbf{e}}_x) & G(\phi, \mathbf{u}_h) - F(\hat{\mathbf{e}}_y, \hat{\mathbf{e}}_y) \end{bmatrix} \mathbf{U}_+ = -\langle \mathbf{r}_h \phi_+ \rangle \quad (5.2)$$

où les fonctions G et F sont définies de la façon suivante,

$$\begin{aligned} G(\phi, \mathbf{u}_h) &= \rho \left\langle \frac{\phi_+ \phi_+}{\Delta t} \right\rangle - \rho \left\langle \frac{\phi_+}{\phi} \mathbf{u}_h \cdot \nabla \phi_+ \right\rangle + 2\mu \left\langle \phi \nabla \left(\frac{\phi_+}{\phi} \right) \cdot \nabla \phi_+ \right\rangle \\ F(\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2) &= \rho \left\langle \frac{\phi_+}{\phi} (\mathbf{u}_h \cdot \hat{\mathbf{e}}_1) \nabla \phi_+ \cdot \hat{\mathbf{e}}_2 \right\rangle \end{aligned}$$

Il ne reste plus qu'à résoudre le système en créant la matrice sur chaque élément en évaluant les fonctions G et F . Une fois l'expression de U_x^+ et U_y^+ obtenue, nous devons encore exprimer l'erreur dans la norme L_2 ou dans la norme H_1 . L'erreur d'approximation $\mathbf{e}^h = \mathbf{u}_+^h - \mathbf{u}^h$ est désormais un vecteur étant donné que le champ de vitesse est un champ vectoriel. Si l'estimateur est évaluée en norme L_2 , nous obtenons,

$$\begin{aligned} \kappa^2 &= \|\mathbf{e}^h\|_{L_2}^2 = \int_K [\mathbf{u}_+^h - \mathbf{u}^h]^T [\mathbf{u}_+^h - \mathbf{u}^h] d\Omega \\ &= \int_K (U_x^+ \phi_+)^2 + (U_y^+ \phi_+)^2 d\Omega \\ &= \left[(U_x^+)^2 + (U_y^+)^2 \right] \|\phi_+\|_{L_2}^2. \end{aligned}$$

Dans le cas où l'erreur est évalué en norme H_1 , l'estimateur s'écrit comme,

$$\kappa^2 = \left[(U_x^+)^2 + (U_y^+)^2 \right] \left[\|\phi_+\|_{L_2}^2 + \|\nabla \phi_+\|_{L_2}^2 \right]. \quad (5.3)$$

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl