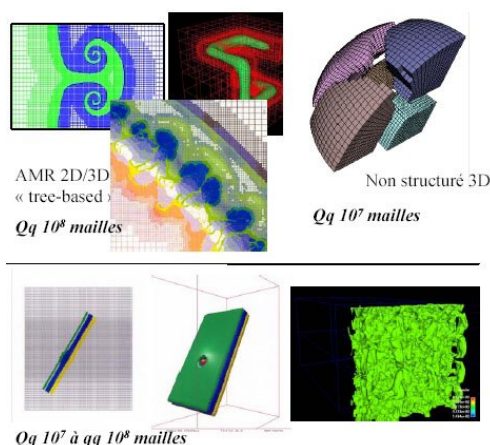


Analyse et Parallélisation efficace des méthodes de type A.M.R Adaptive Mesh Refinement



INRIA Futurs **RAPPORT DE STAGE** **Du 02 février 2004 au 18 juin 2004**

Mr Jean Roman (Tuteur ENSEIRB)		Professeur
Mr Olivier Coulaud et Mr Jean Roman		Directeur de Recherche, Professeur
Alaoui Mohamed	Informatique	Le 19 février 2004

Remerciements

Mes remerciements s'adressent à Monsieur Olivier Coulaud et Monsieur Jean Roman, pour leur encadrement, leur aide et leurs explications très précieuses.

Je tiens à remercier également ma famille et Melle von Ungern Sternberg pour leur soutien moral.

TABLE DES MATIERES

REMERCIEMENTS.....	2
TABLE DES MATIERES	3
INTRODUCTION	4
PARTIE N°1. LE RAFFINEMENT ADAPTATIF DE MAILLAGE	7
1.1 SCHEMA DE RESOLUTION.....	7
1.1.1 <i>Résolution sur un maillage donné, étape de calcul.....</i>	8
1.1.2 <i>Estimation de l'erreur.....</i>	9
1.1.3 <i>Les étapes de raffinement et/ou de regroupement.....</i>	9
1.2 ETAT DE L'ART SUR L' A.M.R. POUR LES MAILLAGES STRUCTUREES	9
1.2.1 <i>Contextes mathématiques.....</i>	10
1.2.2 <i>Les débuts du raffinement adaptatif de maillage.</i>	13
PARTIE N°2. LA STRATEGIE AMR SUR LES MAILLAGES NON STRUCTUREE.....	27
2.1 QUALITE DU MAILLAGE	27
2.1.1 <i>Conformité du maillage.....</i>	28
2.1.2 <i>Critères de qualité du maillage.....</i>	29
2.2 STRATEGIE DE RAFFINEMENT	30
2.2.1 <i>Cas des maillages triangulaires.....</i>	30
2.2.2 <i>Cas des maillages tétraédriques</i>	32
PARTIE N°3. SCHEMA IMPLICITE ET APPROCHE PARALLELE DE L'AMR.....	35
3.1 APPROCHE PARALLELE DE LA STRATEGIE AMR	35
3.1.1 <i>Partitionnement du maillage.....</i>	35
3.1.2 <i>Equilibrage ou Régulation de charge</i>	37
3.2 LE SCHEMA IMPLICITE	38
CONCLUSION	40
BIBLIOGRAPHIES.....	41

INTRODUCTION

Dans la continuité de mon option de troisième année, Parallélisme et Régulation de Charges, mon objectif concernant le projet de fin d'études était de travailler sur un sujet de recherche mettant l'informatique au service des sciences physiques et mathématiques. Le sujet proposé par l'unité de recherche Futurs de l'Institut National de Recherche en Informatique et en Automatique (INRIA Futurs) correspondait exactement à mes attentes.

Essayer de pousser les limites des codes de simulation numérique n'est pas chose aisée. Si les modèles mathématiques et physiques sont là pour décrire au mieux différents phénomènes, l'outil informatique présente des limites. Prenons par exemple le comportement d'un fluide situé initialement au-dessus d'un autre fluide moins dense. Les images suivantes visualisent ce comportement :

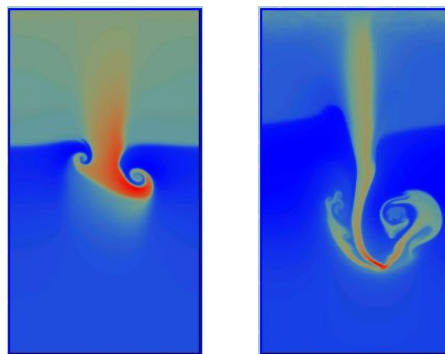


Figure 1 Comportement d'un fluide au-dessus d'un fluide moins dense

Pour pouvoir effectuer une simulation numérique fidèle à ce type de phénomène, il faudrait être capable de capturer les différents événements du comportement du système dans les différentes échelles de temps et d'espace, notamment dans les petites échelles. Par ailleurs, le fait que ces "petits phénomènes" ne soit pas localisable dans le temps et l'espace rend la tâche plus ardue.

Une idée serait de choisir pour l'ensemble du domaine de la simulation une forte résolution, c'est-à-dire une discrétisation spatio-temporelle assez fine permettant de capturer tous les phénomènes. On peut s'interroger sur la faisabilité d'une telle discrétisation. En effet, ce type de discrétisation n'est pas intéressant pour plusieurs raisons. Premièrement, une telle discrétisation ou un tel maillage serait si volumineux qu'il serait impossible de le stocker. Deuxièmement, la complexité calculatoire en terme de coût et de stabilité numérique dépasserait toutes les capacités des calculateurs existants. Troisièmement, certaines parties du maillage n'ont pas besoin d'une discrétisation spatio-temporelle aussi fine.

Il existe des techniques qui répondent dans une certaine mesure au besoin de réaliser des simulations numériques de grandes tailles, en vrai grandeur et de plus en plus précises. L'idée de ces techniques consiste à concentrer les ressources qui sont à notre disposition sur les parties du maillage qui en ont besoin. Les images qui suivent illustrent nos propos:

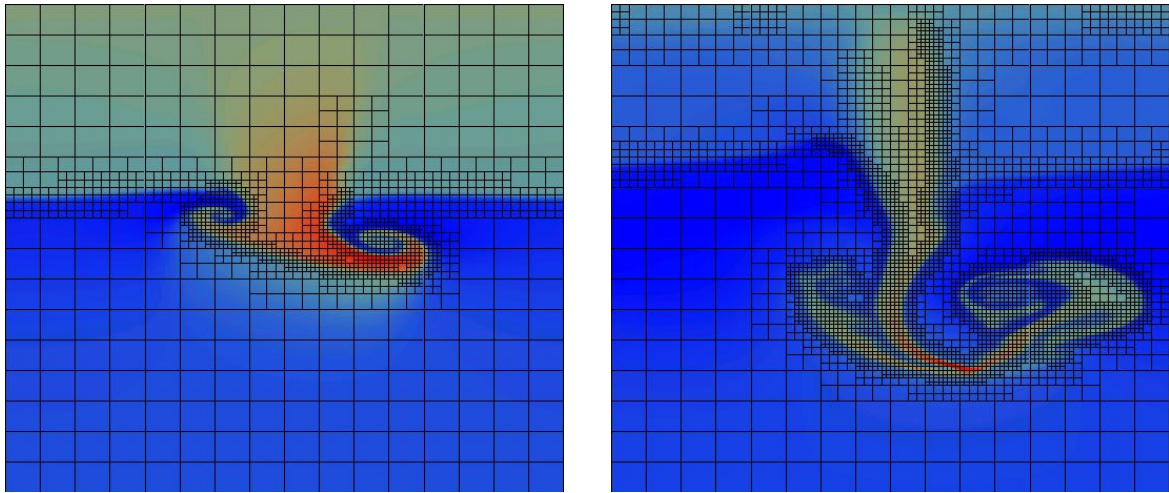


Figure 2 Illustration de l'Adaptation par Raffinement

En effet, on remarque que la résolution du maillage s'adapte au mouvement du fluide le moins dense, les régions intéressantes du maillage ont une résolution plus forte.

Ces techniques sont dites de Raffinement Adaptatif de Maillage (AMR : Adaptive Mesh Refinement). En d'autres termes, la discrétisation du problème traité n'est plus statique mais elle s'adapte à la solution du problème. L'intuition laisse penser que les techniques AMR sont capables de réduire les coûts informatiques des plus ambitieuses simulations numériques.

D'un autre côté, le développement du calcul massivement parallèle, promet un élargissement considérable de l'AMR à des domaines d'applications jusqu'ici étrangères à la simulation numérique. En effet, pouvoir traiter parallèlement différentes parties du maillages peut apporter un gain considérable en temps de calcul.

Cependant, la réalisation d'une architecture informatique parallèle mettant le Raffinement Adaptatif de maillage au service d'une large gamme d'applications n'est pas chose facile. En effet, chaque application possède ses propres spécificités qui font que les maillages utilisés, les modèles mathématiques adoptés et les schémas numériques mis en oeuvre sont tellement variés qu'il est difficile pour une architecture AMR d'avoir une certaine interopérabilité.

En particulier, la nature des maillages utilisés est une contrainte pour cette interopérabilité. Les maillages peuvent être de deux types : structurés ou non structurés. Selon la complexité géométrique du phénomène à simuler, l'un des deux types de maillage s'impose. Il est clair que les maillages non structurés sont plus aptes à épouser les géométries les plus complexes mais sont difficiles à manipuler dans le cas de l'AMR.

Par ailleurs, du point de vue parallèle, le caractère dynamique du maillage fait que la répartition des tâches sur les différentes ressources de calcul doit être dynamique aussi, ce qui pose des problèmes d'équilibrage de charge qui ne sont pas négligeables.

Dans ce mémoire, nous allons étudier l'AMR sous ses différentes facettes en faisant un état de l'art des différentes stratégies existantes. Nous commencerons dans un premier temps les concepts liés aux techniques AMR et nous étudierons en particulier les débuts de la stratégie AMR qui concerne les maillages structurés. La deuxième partie sera consacrée à l'AMR pour les maillages

non structurés et la troisième partie traitera des problèmes liés aux schémas implicites et à la parallélisation.

Dans la bibliographie citée en fin du mémoire, les références : [1], [2], [3], [5], [16], [17], [21], [24], [27], ont été étudiées minutieusement. Les autres références comprennent des articles étudiés brièvement et des articles intéressants pour ce mémoire.

PARTIE N°1. LE RAFFINEMENT ADAPTATIF DE MAILLAGE

1.1 Schéma de résolution

Dans cette section, nous présenterons d'une manière générale les étapes de la résolution d'un problème avec une technique A.M.R dans un cadre parallèle sur des maillage structuré ou non structuré.

Examinons le diagramme qui suit :

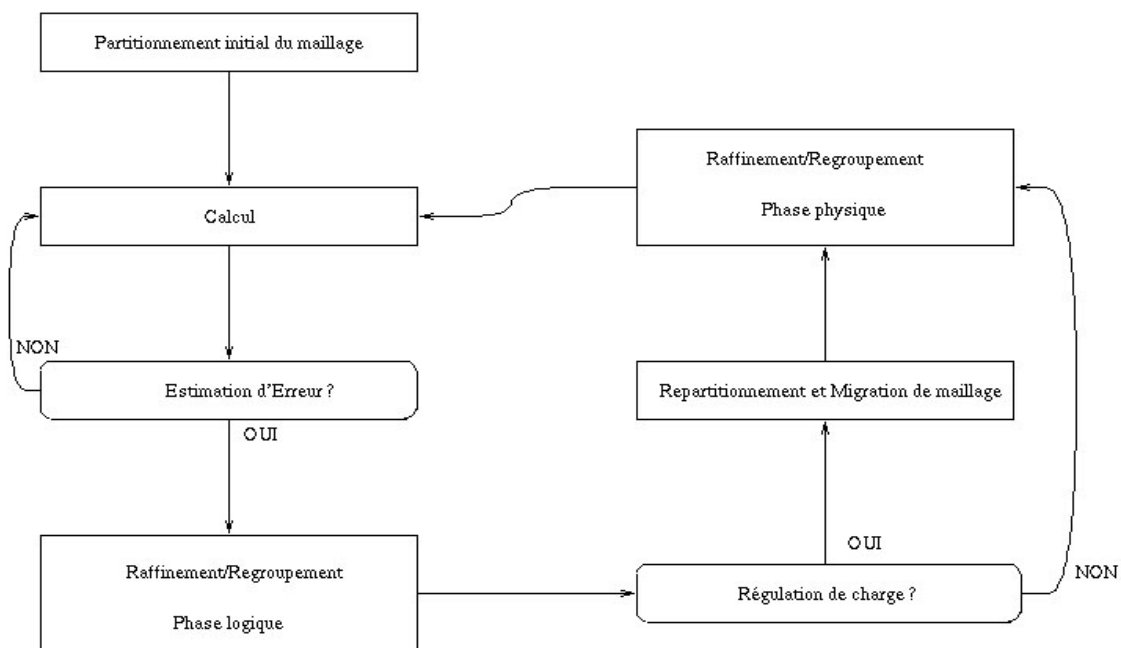


Figure 3 Diagramme AMR

Supposons que le maillage initial M_h existe déjà et que U_h^0 est l'approximation de la solution initiale.

- 1. Partitionnement initial du maillage :** M_h est distribuée d'une façon optimal sur toutes les ressources de calcul existantes. Chaque ressource traitera un maillage m_h . Le terme optimal prendra son sens plus tard dans la partie trois du mémoire.

2. **Calcul** : Chaque ressource produit une partie u_h^1 de la solution sur son bout de maillage m_h au pas de temps suivant.
3. **Estimation d'erreur** : Si u_h^1 est de bonne qualité sur m_h alors on continue les calcul au pas de temps suivant. Le sens de l'expression "bonne qualité" sera explicitée ultérieurement. Pour l'instant nous retiendrons que la solution ne doit pas être trop grossière ni trop précise non plus.
4. **Raffinement/Regroupement (phase logique)** : Sinon on marque les éléments ou les nœuds de m_h qui fournissent une solution trop grossière et ceux qui fournissent une solution trop précise. Les parties marquées sont celles qui seront par la suite raffinées et/ou regroupées. On entend par regroupement, le contraire de raffinement. Cette notion est aussi utile que le raffinement. En effet, certaines zones du maillage peuvent avoir besoin d'une forte résolution uniquement pendant la durée de passage du phénomène que l'on capture.
5. **Régulation de charge** : On évalue le coût du nouveau maillage lié à chaque ressource. Ce coût est évaluable à partir du maillage présent et des zones marquées.
6. **Repartitionnement et migration de maille** : Si il y a un déséquilibre notable entre les ressources en terme de charge de calcul, on choisit de faire des migrations de mailles permettant un rééquilibrage des tâches des différentes ressources. Sinon on passe au raffinement/regroupement physique.
7. **Raffinement/Regroupement (phase physique)** : changement physique du maillage M_h induit par un raffinement et/ou un regroupement dans un sous maillage m_h . Cette phase fournit un maillage $M_{h'}$.
8. **Calcul** : On boucle sur le point 2. qui produit $u_{h'}^1$ qui subira les même traitement par la suite.

Dans les sections qui suivent nous décrirons brièvement les étapes inhérentes à l'AMR. Tout ce qui se rapporte au parallélisme sera traité plus tard.

1.1.1 Résolution sur un maillage donné, étape de calcul

Pour cette étape, il existe différents modèles mathématiques permettant de résoudre différents types de problèmes. Pour des problèmes où le domaine est maillé avec des rectangles ou bien des parallélépipèdes, on utilise souvent des méthodes du type différences finies. Plus généralement, on utilise la méthode des éléments finis pour les problèmes elliptiques ou paraboliques et la méthode des volumes finis pour les problèmes hyperboliques. Ces différentes notions sont passées en revue dans la section 1.2. Le choix des modèles mathématiques utilisés dépend forcément de l'application considérée. Par soucis de modularité, l'un des buts de cette étude est l'analyse des techniques A.M.R. en restant le plus général possible vis à vis des applications.

Par ailleurs, quelque soit les modèles mathématiques, il existe différents schémas numériques permettant la résolution des équations aux dérivées partielles régissant les phénomènes à simuler. Le schéma explicite permet de calculer explicitement une solution à l'instant $t+1$ connaissant la solution à l'instant t via un calcul matriciel directe. Cependant, la solution donnée par ce schéma numérique n'est stable que sous certaines contraintes de pas de temps. En général, il faut des petits pas de temps pour les fortes résolutions et donc beaucoup d'itérations dans le calcul. C'est justement ce point qui désavantage cette méthode surtout qu'elle est relativement facile à mettre en œuvre.

Dans le cas d'un schéma numérique implicite, le calcul donnant une solution au problème devient plus lourd car, pour calculer la solution au pas de temps suivant, une résolution d'un système linéaire s'impose. Ce schéma présente une forte propriété de stabilité qui permet de choisir

des pas de temps plus amples que ceux du schéma explicite et donc de réduire le nombre d'itérations. Cependant, résoudre un système linéaire de très grande taille à chaque pas de temps est quelque chose de très coûteux. De plus, ce schéma numérique ne se prête pas beaucoup à la parallélisation puisqu'il y a une grande dépendance entre les mailles. Par ailleurs, si le raffinement ou le regroupement du maillage intervient, le système linéaire à résoudre change de structure et complique donc la conception d'une gestion automatisée et parallèle des méthodes de type A.M.R. La parallélisation de la résolution de ces systèmes demande une attention particulière en analysant la structure de la matrice du système en fonction du raffinement ou regroupement du maillage.

La majorité des codes existants, utilisant l'AMR, n'utilise que des schémas explicite. Les problèmes soulevés par un schéma numérique implicite seront vues en dernière partie du mémoire.

1.1.2 Estimation de l'erreur

L'estimation de l'erreur dépend du problème traité et de l'approximation utilisée. Pour décider du raffinement ou du regroupement d'un sous ensemble du maillage, on a besoin, dans un premier temps, d'un indicateur d'erreur permettant d'étiqueter l'ensemble des noeuds concernés par un raffinement ou regroupement éventuel. Ce problème est non trivial étant donné qu'on ne connaît pas la solution exact du problème. Il y a cependant des standards permettant d'estimer ces erreurs (Voir [5] pour le détail de la méthode). Ce standard n'est valable que pour des solutions assez régulières dans l'espace. En générale, ces standards sont utilisés avec d'autres méthodes permettant de détecter les phénomènes induisant des discontinuités. Pour plus de précision, les références [7] [8] [9] [10] discutent des différentes techniques d'estimation. Cette partie relève du domaine de l'analyse numérique et ne fait pas partie intégrante de la stratégie AMR. De plus, selon la nature de l'application, certains estimateurs sont plus que d'autres valables. On présentera par la suite, une technique d'estimation qui est souvent utilisée.

1.1.3 Les étapes de raffinement et/ou de regroupement

Ces étapes orientent la stratégie selon la qualité de la solution obtenue. Une mauvaise qualité implique un raffinement des mailles défaillantes. Une trop bonne qualité implique éventuellement un regroupement de certaines mailles. Cependant, ce raffinement et/ou regroupement ne se fait pas d'une façon aléatoire. En effet, le maillage doit respecter certaines qualités liées au modèle mathématique adopté. Par exemple, la méthode des éléments finis impose un maillage conforme. De plus, les stratégies utilisées en structuré ou en non structuré sont différentes étant donnée la différences de nature des problèmes.

1.2 Etat de l'art sur l' A.M.R. pour les maillages structurées

Nous verrons dans un premier temps le contexte mathématique débouchant sur l'utilisation des méthodes de raffinement adaptatif de maillage. La deuxième partie sera consacrée à la première stratégie AMR développée par Marsha Berger basée sur une notion de grilles multiples indépendantes. Marsha Berger a commencé ses travaux avec Joseph Oliger à l'Université de Stanford [17] et les a poursuivis avec d'autres collaborateurs à l'Université de New York ainsi qu'à la NASA Ames Research Center avec notamment Jameson [18], [19], Collela [6], Bokhari [20], Bell, Saltzman et Welcome [21], Aftosmis et Melton [22] et enfin Rigoutos [23].

1.2.1 Contextes mathématiques

Les domaines d'applications de l'AMR sont divers. Tout phénomène régit par des équations aux dérivées partielles peut être traité par les techniques de type AMR. Dans ce qui suit, nous allons expliciter le formalisme mathématique existant autour des techniques AMR.

Commençons par présenter brièvement les différentes méthodes de discrétisation :

□ Méthode des différences finies : MDF

Notations : soit $x \in \Omega \subset \mathbb{R}^d$ $d=1,2,3$ variable de l'espace et $t \geq 0$ variable de temps. Soit $u = u(x, t)$ une fonction physique (champs scalaire ou vectoriel). On note x_j un point d'indice j et t^n un instant d'indice n . On note $u_j^n = u(x_j, t^n)$ l'approximation de la fonction physique en x_j et à l'instant t^n . Soit $h > 0$ un pas d'espace et $\tau > 0$ un pas de temps. On se place dans le cas où $d=1$, on a : $h = |x_{j+1} - x_j|$ et $\tau = t^{n+1} - t^n$. Considérons l'équation :

$$\partial_t u + \partial_{xx}^2 u = f(x, t), x \in \Omega \quad (1)$$

La discrétisation de la partie temporelle de (1) en différences finies donne :

$$\partial_t u = \frac{u_i^{n+1} - u_i^n}{\tau}$$

La discrétisation de la partie spatiale de (1) dépend du schéma de résolution adopté :

Le schéma explicite considère une discrétisation à l'instant n de la partie spatiale et du second membre :

$$\partial_{xx}^2 u = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h}$$

$$f(x, t) = f_i^n$$

et donc (1) devient :

$$\frac{u_i^{n+1} - u_i^n}{\tau} + \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h} = f_i^n \quad (2)$$

(2) permet une résolution directe à chaque pas de temps.

- Le schéma implicite considère une discrétisation à l'instant $n+1$ de la partie spatiale et du second membre :

$$\partial_{xx}^2 u = \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h}$$

$$f(x, t) = f_i^{n+1}$$

et donc (1) devient :

$$\frac{u_i^{n+1} - u_i^n}{\tau} + \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h} = f_i^{n+1} \quad (3)$$

(3) impose une résolution d'un système à chaque pas de temps.

Les différences finies s'appliquent la plupart du temps dans les cas des maillages fabriqués comme suit :

- Assemblage de sous domaines rectangulaires
- Rectangles déformés (coordonnées curvilignes)
- Assemblage de rectangles déformés

Dans le cas de maillages complexes, la méthode des différences finies est moins efficace que d'autres méthodes.

□ Méthode des éléments finis : MEF

Cette méthode consiste à calculer une solution approchée en calculant une interpolation polynomiale de la solution exacte sur chaque éléments du maillage. Cette méthode revient à se donner un triplet : (E, P, D) ayant pour sémantique :

- E : ensemble d'éléments géométriques (une, deux ou trois dimensions)
- P : espace des fonctions polynomiales
- D : ensemble des degrés de liberté définis sur E

Les degrés de liberté sont les inconnus à calculer pour l'élément considéré. Par exemple, si la solution du problème est un champs scalaire et que l'élément est un tétraèdre, on a 4 degrés de liberté. Prenons par exemple l'équation de convection diffusion. L'inconnue est un champ scalaire φ et on se donne un champs vectoriel u , un champs scalaire f et une constante ν :

$$\frac{\partial \varphi}{\partial t} + \text{div}(u\varphi) - \nu \Delta \varphi = f \quad \text{sur } \Omega \times]0, T[$$

$$\varphi(x, 0) = \varphi^0(x) \quad \text{sur } \Omega, \quad \varphi(x, t) = 0 \quad \text{sur } \partial\Omega \times]0, T[$$

La discrétisation de la partie temporelle se fera en différences finis et la partie spatiale sera évaluée suivant un schéma implicite, c'est à dire à l'instant $n+1$.

On cherche Φ^{n+1} solution de :

$$\frac{\Phi^{n+1} - \Phi^n}{\tau} + \text{div}(u^{n+1} \Phi^{n+1}) - \nu \Delta \Phi^{n+1} = f^{n+1} \quad \text{sur } \Omega$$

$$\Phi^{n+1} = 0 \quad \text{sur } \partial\Omega$$

où $\Phi^n \approx \varphi(., t^n)$ et les notations : $u^{n+1} = u(., t^{n+1})$ et $f^{n+1} = f(., t^{n+1})$

L'approximation en éléments finis consiste tout d'abord à passer à la forme intégrale du problème en respectant les formules de transformations d'intégrales:

Soit $\omega : \Omega \rightarrow \mathbb{R}$ une fonction respectant des conditions de régularité que l'on ne détaillera pas, elle est dite fonction test. On note :

$$a(\Phi^{n+1}, \omega) = \int_{\Omega} \Phi^{n+1} \omega dx - \tau \int_{\Omega} \Phi^{n+1} u^{n+1} \cdot \text{grad}(\omega) dx + \nu \tau \int_{\Omega} \text{grad}(\Phi^{n+1}) \cdot \text{grad}(\omega) dx$$

$$l(\omega) = \tau \int_{\Omega} f^{n+1} \omega dx + \int_{\Omega} \Phi^n \omega dx$$

Le problème à résoudre devient : trouver Φ^{n+1} à chaque pas de temps vérifiant :

$$\forall \omega, a(\Phi^{n+1}, \omega) = l(\omega)$$

La méthode consiste à résoudre cette équation sur chaque maille ou éléments du maillage éléments finis. On considère Φ^{n+1} restreint à chaque élément comme un champs local polynomial combinaison linéaire des valeurs aux noeuds c'est-à-dire des inconnues du problème. Un bon choix des fonctions ω permettra de poser le problème en terme de systèmes linéaires élémentaires qui peut être assemblés en un système linéaire global qu'il faudra résoudre à chaque pas de temps.

En d'autre terme, la méthode des éléments finis consiste à considérer un espace de projection polynomial pour approcher la solution exacte. Dans ce contexte, le raffinement/regroupement permet d'optimiser le choix de l'espace de projection. Dans la littérature, trois stratégies sont étudiées : le h-refinement [13], le p-refinement [14] et le r-refinement [15]. Le premier consiste à ajouter (raffinement) ou supprimer (regroupement) des polynômes de l'espace de projection selon l'indicateur d'erreur. Ceci se traduit par un raffinement /regroupement du maillage par endroit. Le second consiste à augmenter ou diminuer le degré de certains polynômes d'interpolation pour approcher au mieux la solution exacte. La dernière consiste à repositionner des noeuds du maillage afin d'avoir des polynômes d'interpolations optimaux sans ajout de nouveaux noeuds. Des stratégies hybrides peuvent être envisagées non sans, bien sûr, un coût d'implémentation important.

La méthode des éléments finis couplée avec l'AMR est utilisée efficacement dans le domaine de la mécanique des milieux continus.

□ Méthode des volumes finis : MVF

Prenons par exemple l'équation de conservation suivante :

$$\frac{\partial U}{\partial t} + \text{div}(F(U)) = 0$$

On considère un maillage de volumes finis M et soit $K \in M$ un volume. On intègre l'équation sur un volume K en prenant comme fonction test la fonction unité, ce qui donne :

$$\frac{d}{dt} \int_K U(x, t) dx + \int_{\partial K} F(U) \cdot N d\sigma = 0$$

qui est le bilan conservatif sur K . On regarde en fait ce qui rentre et sort de la maille.

Le schéma numérique est le suivant , on pose :

$$U_k^n = \frac{1}{|K|} \int_K U(x, t^n) dx$$

Approximation de la forme faible :

Soit A_{kl} la frontière entre le volume K et ses volumes voisins :

$$\frac{U_k^{n+1} - U_k^n}{\tau} + \sum_{A_{kl} \in \partial K} \frac{|A_{kl}|}{|K|} \int_n^{n+1} \frac{1}{|A_{kl}|} \int_{A_{kl}} F(U(x, t)) \cdot N d\sigma = 0$$

ce qui donne pour un schéma de calcul implicite :

$$\frac{U_k^{n+1} - U_k^n}{\tau} + \sum_{A_{kl} \in \partial K} \frac{|A_{kl}|}{|K|} \Phi(U_k^{n+1}, U_l^{n+1}, N_{kl}) = 0 \quad (4)$$

avec :

$\Phi(U_k, U_l, N) = -\Phi(U_l, U_k, N)$: conservation.

$\Phi(U, U, N) = F(U) \cdot N$: consistance.

$\sum_{A_{kl} \in \partial K} \frac{|A_{kl}|}{|K|} N_{kl} = 0$: préservation des constante.

L'équation (4) est un système linéaire à résoudre à chaque pas de temps.

En adoptant des schémas de calcul implicite pour les trois méthodes cités (MDF, MEF, MVF), on aboutit dans tous les cas à un système linéaire de très grande taille à résoudre. Quand on décide de raffiner et/ou de regrouper, la matrice du système linéaire change de forme et rend l'automatisation des méthodes AMR compliquée.

Comparaison des méthodes :

	MDF	MEF	MVF
Formulation	forte	faible	faible
Rapidité des calculs	+	-	-
Géométries complexes	-	+	+
Raffinement/Regroupement	-	+	++
Raffinement hiérarchique	-	+	++
Rapidité de mise en oeuvre	++	-	-
Précision	+	++	-

L'analyse de ce tableau permet d'affirmer que dans le cas des géométries complexe, la MEF et La MVF sont les méthodes les plus utilisées dans le contexte AMR. La MEF est la plus précise mais elle est difficile à mettre en oeuvre et induit des calculs plus coûteux.

Comparaison selon le domaine applicatif :

	MDF	MEF	MVF
Mécanique des structures	+	++	-
Fluides incompressibles	+	++	+
Fluides compressibles	+	-	++
Elliptique	+	++	-
Hyperbolique	+	+-	++
Conservation	+	-	++
Expertise Utilisateur	-	++	+

La mécanique des milieux continus préfère la MEF étant donné la forme diversifiée des types d'éléments que l'on peut avoir. De plus, cette méthode est largement utilisée dans l'industrie et donc il existe une forte expertise dessus.

1.2.2 Les débuts du raffinement adaptatif de maillage.

La stratégie AMR de Marsha Berger concerne essentiellement les maillages structurés. Elle repose sur une notion multi grille dont chaque grille peut être traitée d'une façon indépendante. En d'autres termes, quelque soit la grille et son niveau de résolution, cette dernière contient toutes les informations nécessaires pour la poursuite des calculs. Par ailleurs, la totalité du maillage possède une structure hiérarchique de différents niveaux de résolution. Chaque niveau est constitué d'un ensemble de sous grilles disjointes couvrant certaines parties du maillage. La figure suivante représente la structure hiérarchique du maillage :

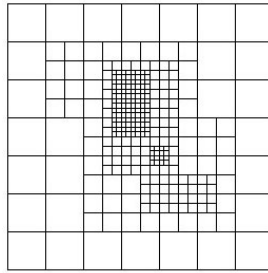


Figure 4 Maillage total

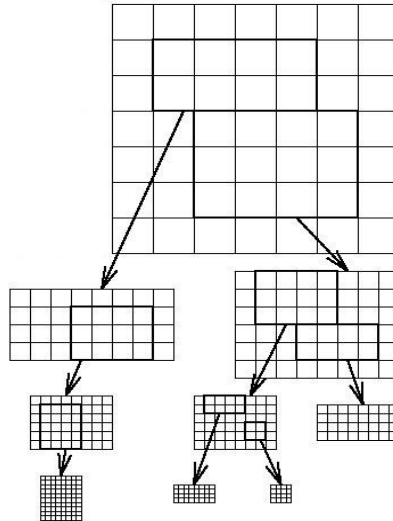


Figure 5 Hiérarchies de grille

Le premier niveau de résolution, dit aussi niveau racine, se compose d'un ensemble de grilles recouvrant l'ensemble des régions concernées par le calcul. Chaque grille contient :

- une zone de calcul,
- une zone frontière constituée de cellules fantômes regroupant les informations aux frontières.

Quant aux autres niveaux de résolution, chacun d'eux se compose d'un ensemble de grille dont chacune contient :

- une zone de calcul obtenue par raffinement du niveau directement moins fin,
- une zone tampon permettant une couverture totale du phénomène jusqu'au prochain remaillage
- une zone frontière composée de cellules fantômes pour assurer l'autonomie d'une grille.

La figure qui suit visualise les propos ci-dessous :

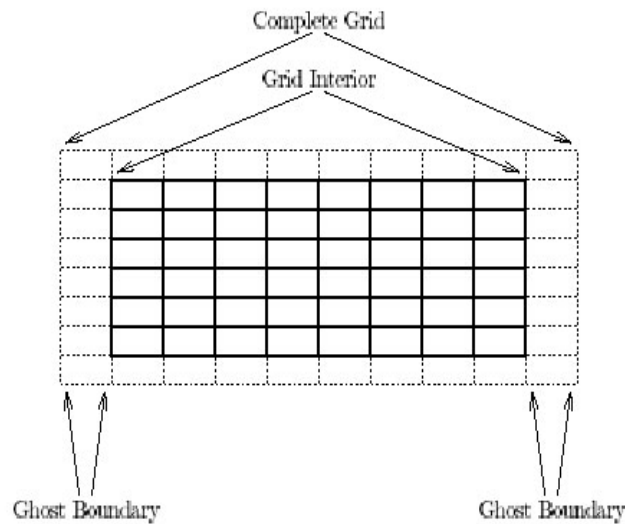


Figure 6 Structure d'une grille interne

1.2.2.1 Algorithme AMR de Berger

L'approche de Berger repose aussi bien sur un raffinement spatial que temporel. Soit h_l et τ_l , respectivement le pas d'espace et le pas de temps au niveaux de résolution l . Si le taux de raffinement entre le niveau de résolution l et $l+1$ est r , on a : $h_{l+1} = \frac{h_l}{r}$. Le choix de Berger est celui d'effectuer un raffinement temporel équivalent, c'est-à-dire : $\tau_{l+1} = \frac{\tau_l}{r}$. Ce qui signifie que la majorité des calculs sont concentrés sur les niveaux les plus fins.

La stratégie AMR exposée par Berger peut être décortiquée en 5 opérations :

1. la collecte des informations aux frontières des grilles. Ces informations peuvent provenir d'une part des conditions aux limites ou initiales notamment pour les grilles touchant le bord du domaine total et les grilles du niveau racine, d'autre part, des grilles parentes (grilles du niveau directement moins fin) ou enfin des grilles du même niveau dites aussi grilles frères.
2. l'appel au solveur numérique. En toute rigueur, le solveur numérique devrait être totalement indépendant de l'AMR. Dans l'approche de Berger, les spécifications du solveur ne sont pas abordées.
3. l'interfaçage entre les niveaux de raffinement basé sur la correction de flux. Cette phase concerne particulièrement les schémas de calcul conservatif.
4. la projection de solution qui permet l'amélioration des solutions du niveau de plus basse résolution.
5. le raffinement de maillage qui se fait par un placement des grilles avec une certaine résolution sur des parties choisies grâce à certains critères de sélection.

Avant de décrire chaque partie, voici une première version de l'algorithme de traitement développé par Berger:

TraitementAMR(Niveau)

Debut

Si "le raffinement est nécessaire" Alors Raffiner(Niveau)

CollectValeursFrontière(Niveau)

AppelSolveur(Niveau)

Si " Niveau \neq MaxDesNiveaux " Alors

Début

**Pour $r = 0$ à FacteurDeRaffinementTemporel - 1 Faire
TraitementAMR(Niveau + 1)**

End

IncrementationTemporelle(Niveau)

Si "Niveau différent de max des niveaux" Alors

Début

CorrectionDeFlux(Niveau, Niveau + 1)

Projection(Niveau, Niveau + 1)

End

End

❑ Collecte des informations aux frontières

Il existe trois sources à partir desquelles on peut extraire les informations pour initialiser les cellules fantômes des grilles d'un certain niveau :

- transfert des valeurs aux frontières vers un niveau plus fin à partir d'un niveau directement moins fin, c'est à dire on injecte des valeurs aux frontières du niveau l vers le niveau $l+1$. Cette opération s'appelle une injection et elle consiste en une interpolation.

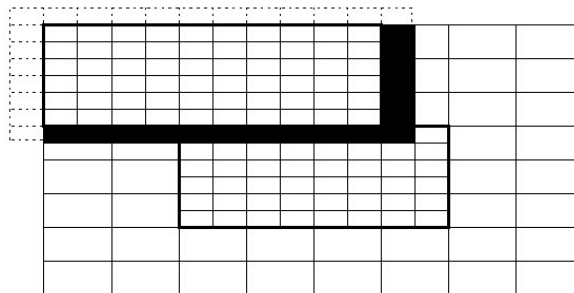


Figure 7 A partir d'une grille parente

- copies des valeurs aux frontières à partir d'un frère du même niveau. Dans ce contexte, un frère est une grille du même niveau et recouvrant les cellules fantômes de la grille à initialiser.

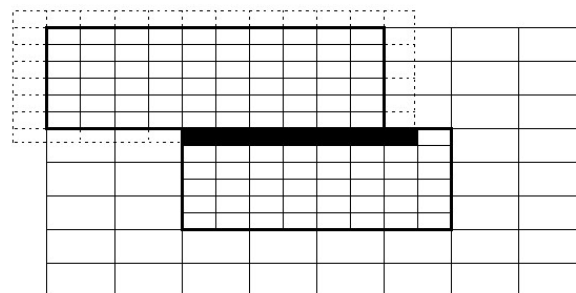
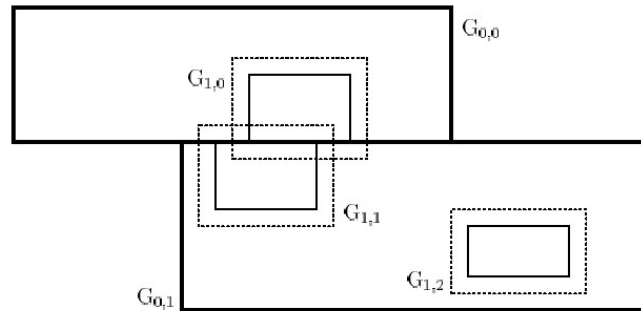


Figure 8 A partir d'une grille frère**Figure 9 Grilles frère**

$G_{1,0}$ et $G_{1,1}$ sont les uniques frères de la configuration de la figure 5.

- extrapolation à partir de l'extérieur du domaine de calcul.

**Figure 10 A partir de l'extérieur**

Le principe fondamental de cette collecte des informations aux frontières, notamment pour l'opération d'injection, est d'atteindre pour chaque grille un état de précision maximal. En d'autres termes, les grilles des niveaux $l > 0$ requièrent un coffrage maximal c'est à dire qu'elles doivent être entourées par des grilles appartenant au niveau directement moins fin. Cet état des choses assure une précision maximal car à chaque appel du solver sur une grille, les frontières de cette dernière sont aussi précises que le niveau directement moins fin c'est à dire le niveau le plus précis pour l'instant.

Idéalement, la copie à partir d'une grille frère serait la stratégie menant vers un maximum de précision. Cependant ce n'est pas toujours possible pour toutes les grilles car ceci supposerait un raffinement du maillage total, ce qui va à l'encontre de l'AMR.

Par ailleurs, pour les grilles du bord du domaine de calcul, les informations aux frontières sont issues de données construites non calculées. Ces données sont obtenues en général par extrapolation. Dans certain cas, on utilise un système de copie approprié basé sur des frontières périodiques, réflexives ou encore sur des procédures analytique.

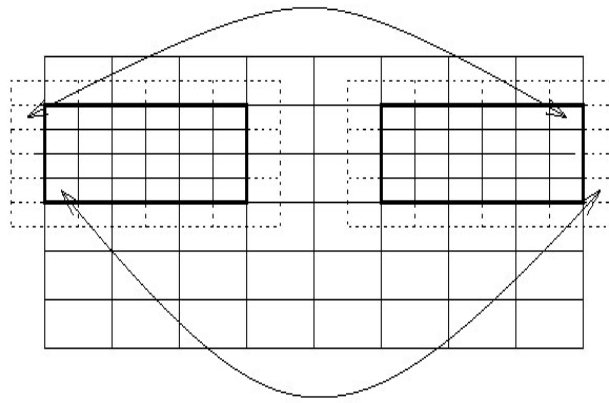


Figure 11 Périodicité

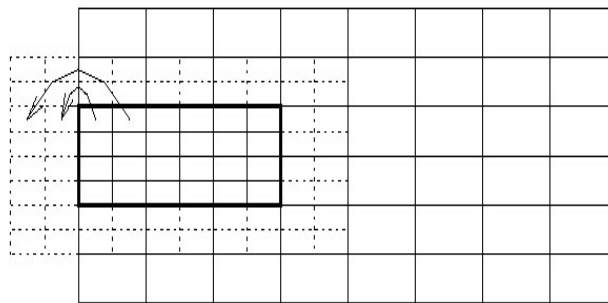


Figure 12 Réflexivité

□ L'appel au solveur

Dans l'approche de Berger, les mécanismes de l'AMR sont cachés dans une boîte noire indépendamment du solveur numérique. Le solveur attend de l'utilisateur des données en entrée afin de fournir des résultats en sortie et ceux indépendamment de la stratégie AMR. Le solveur fournit la solution au pas de temps suivant spécifié par l'utilisateur..

□ La correction de flux

La stratégie AMR de Berger permet la conservation des flux au niveau des interfaces entre une grille de niveau l et une grille de niveau $l+1$. La correction de flux permet cette conservation qui est rompue dès que le raffinement entre en jeu. Du point de vue pratique, le solveur reçoit non seulement les conditions aux limites, la description d'une grille (nombre de cellules sur les différents axes par exemple), les informations liées au pas de temps, mais aussi des tableaux permettant le stockage des flux le long de l'interface de chaque grille. Ces flux sont collectés après chaque pas de temps. Pour la grille la plus fine (celle du niveau $l+1$), les différents flux collectés sont sommés sur l'ensemble des pas de temps constituant le pas de temps de moindre résolution. La différence entre les flux du niveau $l+1$ et les flux niveau l est utilisée pour corriger la solution du niveau l .

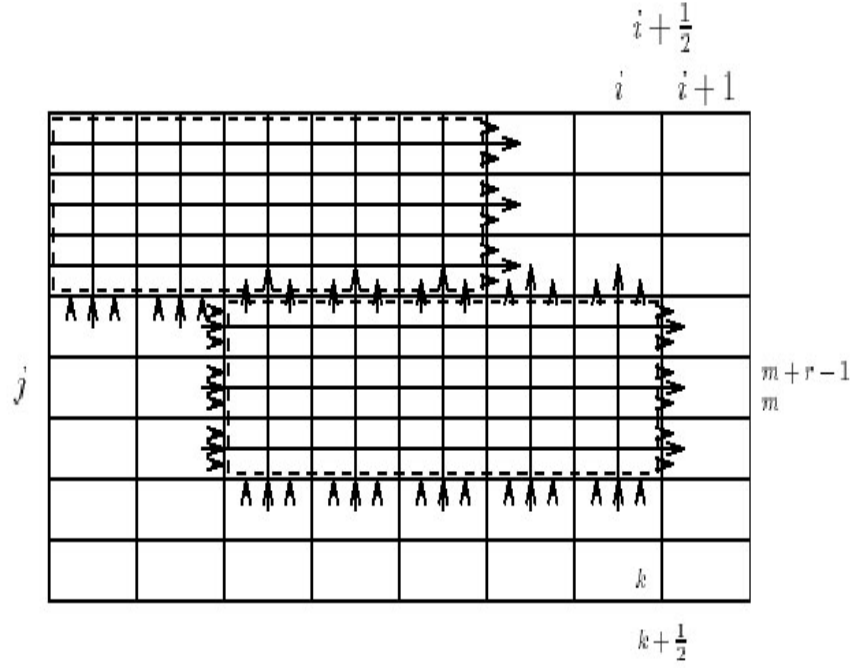


Figure 13 Correction de flux

En se basant sur la figure précédente, nous allons décrire les mécanismes de correction.

- Soit r le taux de raffinement.
- Soit $u_{i,j}^l$ la solution au centre de la cellule (i, j) du niveau l (niveau de basse résolution).
- Soit $f_{k+\frac{1}{2},m}^{l+1}$ le flux à l'interface entre (k, m) et $(k + 1, m)$ du niveau $l + 1$.
- Soit $f_{i+\frac{1}{2},j}^l$ le flux à l'interface des cellules basses résolutions (i, j) et $(i + 1, j)$.
- Soit $F_{k+\frac{1}{2},m}^{l+1}$ le flux total à l'interface entre (k, m) et $(k + 1, m)$ du niveau $l + 1$ calculé sur r pas de temps, correspondant à un pas de temps du niveau l .
- Soit, par exemple n_y^l le nombre de cellule du niveau l à l'interface verticale.

Initialement, après le traitement du niveau l , on stocke les $f_{i+\frac{1}{2},j}^l$ avec $j = 1 \dots n_y^l$.

Soit $F_{k+\frac{1}{2},m}^{l+1} \leftarrow 0, m = 1 \dots n_y^{l+1}$.

Après chaque pas de temps du niveau $l + 1$, on fait le cumul: $F_{k+\frac{1}{2},m}^{l+1} \leftarrow F_{k+\frac{1}{2},m}^{l+1} + f_{k+\frac{1}{2},m}^{l+1}, m = 1 \dots n_y^{l+1}$.

La figure suivante schématise les calculs de flux :

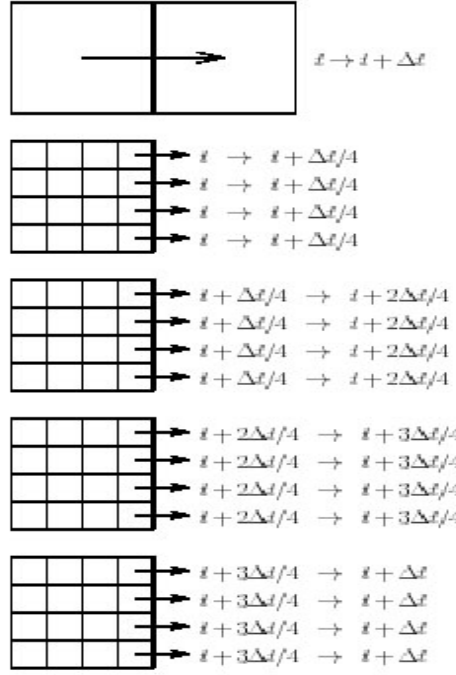


Figure 14 Correction de flux

Quand les r pas de temps du niveau $l+1$ sont terminés, c'est à dire il y a synchronisation entre le pas de temps de niveau l et celui de niveau $l+1$, on corrige de la manière suivante :

$$u_{i+1,j} \leftarrow u_{i,j} + (F_{k+\frac{1}{2},m}^{l+1} - f_{i+\frac{1}{2},j}^l), j = 1 \dots n_y^l.$$

Le procédé est le même pour les flux verticaux.

Une explication plus intuitive de la correction de flux serait que l'algorithme de Berger stocke la quantité de matière traversant une interface entre deux niveaux. Quand le pas de temps de niveau l est terminé, l'algorithme stocke le flux de matière à basse résolution. Quand les r pas de temps du niveau $l+1$ sont terminés, l'algorithme stocke le flux de matière à haute résolution. La différence entre les deux flux permet la correction de la solution du côté de la basse résolution.

Par ailleurs, si les deux flux sont égaux, la solution basse résolution reste inchangée. Cette configuration est quasiment improbable car cela voudrait dire que les deux niveaux ont la même résolution ce qui est en contradiction avec le raffinement qui a été fait.

Pour récapituler, le but de la correction de flux est d'ajuster la solution dans les cellules basse résolution situées aux interfaces. Ainsi, quand l'algorithme passera à la phase de projection, consistant en l'amélioration des solutions des cellules de basse résolution recouvertes par des cellules de plus haute résolution, la conservation des flux entre les deux niveaux de résolution différentes sera respectée.

□ La phase de projection

Cette phase permet la mise à jour du niveau l grâce aux calculs plus précis du niveau $l+1$. La projection peut se faire de plusieurs façons, parmi elles :

- la copie. Par exemple, la copie des vitesses aux nœuds,
- la sommation. Par exemple, la sommation de masse aux centres des cellules,
- la moyenne. Par exemple, la moyennes des densités aux centres des cellules.

Le schéma suivant permet la visualisation de la phase de projection :

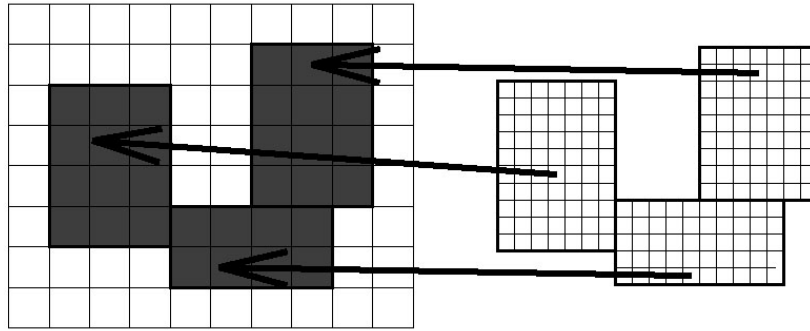


Figure 15 Phase de Projection

Notons bien que la projection se fait toujours après la phase de correction de flux. En effet, si on commence par la projection, qui fournit une solution plus précise, on risque d'annuler l'effet de cette projection si la correction de flux s'opère sur une cellule concernée par cette projection. Il reste maintenant à décrire la phase de raffinement de maillage développée par Berger. Cette phase est basée sur le placement dans certaines régions des grilles de plus grande résolution.

□ La phase de raffinement

Comme l'algorithme de traitement décrit précédemment, la stratégie de raffinement est récursive. En effet le raffinement du niveau l commence par le raffinement du niveau $l+1$ et ainsi de suite jusqu'au niveau le plus fin. Voici l'algorithme de raffinement proposé par Berger :

Raffiner(Niveau)

Debut

Si "Niveau=NiveauMaxPermis" Alors Retourner

Si "Niveau ≠ MaxDesNiveaux" Alors Raffiner(Niveau+1)

Selection(Niveau)

Expansion(Niveau)

Grouperment(Niveau)

Remaillage(Niveau+1)

Si "InitialPasDeTemps" et " Niveau ≠ NiveauMaxPermis" et "EstVideNiveauFin" Alors

Raffiner(Niveau)

End

Décrivons cet algorithme brièvement avant d'analyser les phases principale en détail. Soit l , le niveau traité par la procédure **Raffiner()**.

L'instruction :

Si "Niveau=NiveauMaxPermis" Alors Retourner

est le critère d'arrêt de la récursion. En effet, on ne peut permettre une profondeur de niveau quelconque. Si le niveau traité correspond au niveau maximal permis, on sort de la récursion.

L'instruction :

Si "Niveau ≠ MaxDesNiveaux" Alors Raffiner(Niveau+1)

permet à l'algorithme de raffinement d'effectuer un appel à lui même. A ce stade de l'algorithme, toutes les grilles du niveau $l+1$ n'ont pas encore été remaniées, cependant, celles des niveaux entre

$l+2$ et l_{\max} (si elles existent) ont été remaniées et présentent une hiérarchie de grille plus fines recouvrant toutes les régions intéressantes du niveau $l+1$. Le raffinement du niveau l impliquera le remaniement du niveau $l+1$ au moment du remaillage.

L'instruction :

Sélection(Niveau)

sera expliquée en détail dans la suite. Pour l'instant, il suffit de savoir que la sélection permet de produire les régions qui ont besoins d'être raffinées.

L'instruction :

Expansion(Niveau)

permet d'ajouter aux cellules sélectionnées, le nombre nécessaire de cellules pour remplir les zones tampons et les frontières des futures grilles du niveau $l+1$.

L'instruction :

Groupement(Niveau)

permet de grouper les cellules marquées dans un ensemble de régions rectangulaires. Chacune d'elles représente le sous domaine qui sera par la suite une grille. Notons que cette notion de groupement n'a rien avoir avec le regroupement cité tout au long du mémoire. La notion de regroupement est l'antonyme de raffinement. On l'utilise pour baisser la résolution dans les zones qui n'en ont plus besoin. L'algorithme de Berger ne prend pas en charge le groupement.

Nous verrons, par la suite, plus en détail comment se fait le groupement dans l'approche de Berger. Après le groupement, on remaille le niveau $l+1$:

Remaillage(Niveau+1)

Les grilles du niveau $l+1$ sont remplacées par les régions raffinées au niveau l et par les grilles créées du niveau $l+2$ au niveau l_{\max} . Nous verrons plus en détail, par la suite, cette phase de remaillage.

La dernière instruction :

Si "InitialPasDeTemps" et "Niveau \neq NiveauMaxPermis" et "EstVideNiveauFin"

Alors

Raffiner(Niveau)

est utilisée pour l'initialisation de la hiérarchie. En effet, l'algorithme de raffinement est exécuté à chaque niveau existant. Initialement, il n'existe pas de grilles et cette dernière instruction permettra la créations de grilles.

Décrivons maintenant, la sélection, le groupement et enfin le remaillage :

□ La Sélection

Le problème de la sélection est spécifique à la nature de l'application utilisant l'AMR. Une variété de critères de sélections peut être utilisées.

Cette partie du exposé rejoint le paragraphe 2.1.2 des indicateurs d'erreur.

Nous allons présenter un critère de sélection recommandé par Marsha Berger. C'est l'estimation d'erreur par la troncation de Richardson.

Pour estimer cette erreur de troncation, on se pose la question suivante :

En continuant le calcul avec la résolution actuelle, de combien varie la solution du problème par rapport à la solution calculée avec une plus grande résolution?

Si pour une cellule, la variation est significative, alors le raffinement de cette cellule s'impose. Le terme "significative" prend son sens dès que l'on définit un seuil qui permettra de décider.

Le problème qui se pose est celui de la transcription analytique de la question précédente.

L'idée est de comparer la solution obtenue sur une grille de pas spatial h sur deux pas de temps d'amplitude Δt avec une solution obtenue sur une grille de pas spatial $2h$ sur un seul pas de temps $2\Delta t$.

En d'autre terme :

- Soit $u(x, t)$, le champs solution à l'instant t en x .
- Soit Q_h , l'opérateur du schéma numérique.
- Soit q , l'ordre de précision de l'opérateur.

On a : $u(x, t + \Delta t) \cong Q_h u(x, t)$.

La solution obtenue sur une grille de pas spatial h sur deux pas de temps d'amplitude Δt est évaluée par :

$$Q_h^2 u(x, t)$$

La solution obtenue sur une grille de pas spatial $2h$ sur un seul pas de temps d'amplitude $2\Delta t$ est évaluée par:

$$Q_{2h} u(x, t)$$

En tenant compte des ordres de précision, l'erreur de troncation ε_{tronc} est évalué de la manière suivante :

$$\frac{Q_h^2 u(x, t) - Q_{2h} u(x, t)}{2^{q+1} - 2} = \varepsilon_{tronc} + O(h^{q+1})$$

A première vue, cet estimateur d'erreur est idéal pour l'AMR, puisqu'il utilise le même solveur qui nous permet de calculer les solutions. Cependant, si le terme $Q_h^2 u(x, t)$ est facile à calculer en faisant deux appels au solveur, le terme $Q_{2h} u(x, t)$ est plus délicat à calculer. En effet, on doit faire un appel au solveur sur une grille de pas spatial $2h$ et cette grille n'existe pas, ou du moins, elle n'est pas explicitement définie.

Une solution au problème serait de faire une copie de la grille de pas spatial h en gardant uniquement les cellules de pas $2h$, on pourra appeler ensuite le solveur sur cette grille sans aucun problème.

Par ailleurs, on a besoin des zones tampons et frontières de cette grille. Cette contrainte est plus problématique. Pour obtenir ces valeurs aux frontières, Berger propose deux approches : la première consiste en une expansion des valeurs intérieurs de la grille et la deuxième collecte ces valeurs à partir de grilles parentes à définir.

Ainsi le calcul de $Q_{2h} u(x, t)$ ne se fait non sans difficulté. Quelques implémentations de la stratégie AMR de Berger ajustent le code du solveur numérique pour simplifier le calcul de $Q_{2h} u(x, t)$.

Cependant cette intrusion de l'AMR dans le code du solveur altère l'un des avantages essentiel de la stratégie de Berger, c'est à dire la possibilité d'utiliser facilement la stratégie AMR à des nouvelles applications en incorporant les solveurs de l'application avec un minimum de recodage.

□ Le Groupement

L'algorithme de groupement de Berger et Rigoutsos [23] est basé sur une méthode utilisée en reconnaissance de forme. Les étapes de l'algorithme sont les suivantes:

1. Placer une grille minimale (2D ou 3D) contenant toutes les cellules marquées (Figure 12a).
2. Calculer un entier relatif S_i , dit signature, comptant les cellules marquées sur la $i^{\text{ème}}$ surface de la grille, c'est à dire la $i^{\text{ème}}$ rangée en 2D ou le $i^{\text{ème}}$ plan en 3D. Par exemple en 3D : $S_i = \sum_j \sum_k f_{ijk}$, avec $f_{ijk} = 1$ si la cellule ijk est marquée et 0 sinon.
3. Les S_i nulles correspondent aux entrées où il n'y a pas de cellules marquées et sont donc candidat potentiel pour une coupe de la grille. L'index i choisi pour la coupe est celui qui se trouve le plus au centre de la grille (Figure 12b).
4. Si toutes les signatures S_i sont non nulles, on calcul le Laplacien Δ_i des signatures de la manière suivante : $\Delta_i = S_{i+1} - 2S_i + S_{i-1}$. Le changement de signe de Δ nous informe sur les entrées potentielles pour effectuer la coupe. On choisit l'indice i tel que la magnitude $Z_{i+\frac{1}{2}} = |\Delta_{i+1} - \Delta_i|$ soit maximale (Figure 12c-d).

La figure suivante schématise les étapes précédentes dans le cas des grilles 2D :

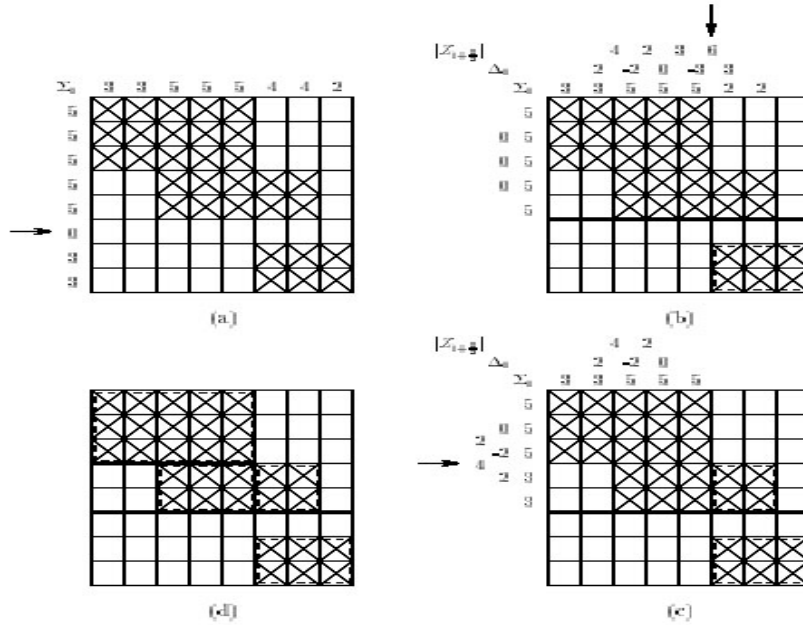


Figure 16 Exemple de groupement avec signature

L'algorithme de groupement est récursif et effectue un appel à lui même sur chaque sous domaine. La récursivité se poursuit jusqu'à la satisfaction d'un critère d'arrêt qui pourrait être un certain nombre de cellules par grille, fixée par rapport au nombre total des cellules.

L'algorithme de groupement opère sur tout l'espace de calcul et se fait en un temps en $O(n \log n)$ par rapport au nombre de cellules.

Nous allons passer maintenant au remaillage.

□ Le remaillage

L'algorithme de remaillage crée des grilles plus fines à partir des régions construites par le groupement. Les valeurs à l'intérieur du domaine de calcul de ces nouvelles grilles sont au pire des cas, un niveau de résolution moins précises que les anciennes grilles du même niveau, car la plupart de ces valeurs sont copiées à partir de ces anciennes grilles. Les valeurs des régions des nouvelles grilles qui ne sont pas recouvertes par d'anciennes grilles proviennent de grilles parentes.

Le schéma suivant illustre nos propos :

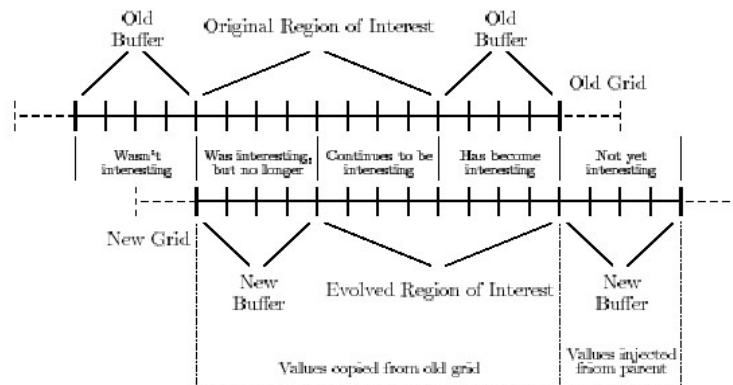


Figure 17 Copie des anciennes grilles vers les nouvelles grilles

1.2.2.2 Evolution de la stratégie de Berger

La première implémentation de la stratégie AMR par Berger avait quelques propriétés qui ont été délaissées par la suite. Nous allons en citer quelques unes.

- Le type de maillage sur lequel opérait la stratégie de Berger était isotropique. Des mécanismes capables d'effectuer un adressage entre des grilles rectangulaires et des grilles curvilignes ont permis de remédier à cela.
- Dans la stratégie originale, les grilles d'un même niveau de résolution pouvaient s'entrelacer. Ce chevauchement entraînait une redondance dans les calculs et une consommation considérable de la mémoire. Un ajustement dans le code AMR permet d'interdire cet entrelacement.
- A l'origine, les valeurs étaient localisées aux nœuds des grilles. Avec l'introduction du schéma conservatif et un nouveau algorithme de groupement (celui basé sur les signatures), la localisation au centre des cellules s'est imposée. Cette décision n'est pas en contradiction avec beaucoup de champs d'application de l'AMR de Berger puisqu'il existe beaucoup de phénomènes utilisant des solveurs avec des valeurs centrées. Cependant, cette approche reste restrictive.

Pour conclure, nous pouvons retenir que les avantages de la stratégie AMR de Berger résident dans le fait qu'elle permet de traiter des phénomènes utilisant différentes physiques sur des échelles multiples sans raffiner la totalité du maillage. Cependant, la puissance de cette stratégie est

diminuée dès qu'on examine la complexité des relations existantes entre les données et donc la difficulté d'implémenter une architecture de structure de données facile d'utilisation pour les concernés par la simulation et surtout applicable à une large gamme d'applications.

Cette présentation de la stratégie de Berger est essentiellement basée sur la thèse de H.J. Neeman [24] où il a exposé un système de gestion hiérarchique de la stratégie AMR de Berger. Ce système permet :

- Une gestion de données d'une façon autonome.
- L'expression des relations les plus complexes entre les données.
- L'utilisation de la stratégie AMR pour de nouvelles applications.

L'article [24] fournit une implémentation détaillée de ce système.

PARTIE N°2. LA STRATEGIE AMR SUR LES MAILLAGES NON STRUCTUREE

Le maillage des domaines de calcul doit être optimal pour une simulation fidèle à la réalité. De plus, certains comportements à simuler font varier la géométrie du problème en la rendant de plus en plus compliquée. Les grilles non structurées offrent un potentiel important pour une discrétisation plus précise du domaine. Cependant, les solveurs basés sur les maillages non structurés requièrent plus de ressources de calcul que leurs voisins basés sur les maillages structurés. De ce fait, l'application de maillages non structurés à des grands problèmes en trois dimensions reste limité.

En outre, la méthode mathématique derrière les maillages non structurés est souvent la méthode des éléments finis. Cette méthode a besoin d'une bonne qualité de maillage pour qu'elle soit précise. Par ailleurs, dans un contexte de raffinement adaptatif de maillage, la qualité de ce dernier risque de se détériorer au fur et à mesure de la stratégie d'adaptation. Il est donc important d'avoir des garanties sur l'évolution du maillage.

En général, quand on parle de maillage non-structuré, on est en présence de mailles triangulaires en 2D et tétraédriques en 3D.

La figure suivante donne un exemple de maillage non structuré :

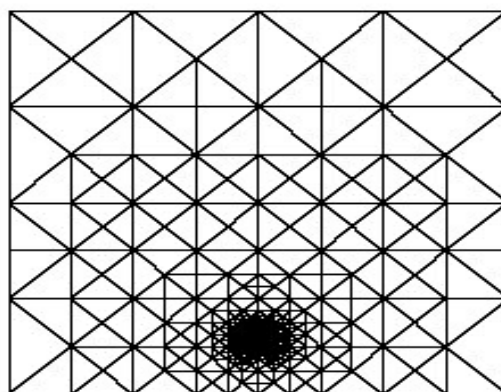


Figure 18 Exemple de maillage non structuré

2.1 Qualité du maillage

Les articles [2] et [3] illustrent la technique A.M.R dans le cadre de maillage tétraédrique.

Les maillages utilisés pour une résolution de type AMR doivent satisfaire certaines conditions imposées par le modèle mathématique adopté. Le modèle des éléments finis, par exemple, demande un maillage dit conforme, c'est à dire pour un maillage 3D tétraédrique, l'intersection entre deux tétraèdres est soit un triangle, soit un segment connectant deux noeuds, soit un seul nœud ou bien l'ensemble vide. Par ailleurs, l'approximation de la solution exacte fournis par la méthode des éléments finis se dégrade dès que les tailles de deux éléments voisins diffèrent énormément. Ainsi la stratégie adoptée pour le raffinement/regroupement doit sauvegarder la qualité du maillage.

Dans le cas 2D et pour un maillage éléments finis triangulaire, il est important que le raffinement/regroupement ne génère pas des angles trop petits ou trop grands, Sinon le maillage devient de mauvaise qualité. En effet, la solution approchée se dégrade pour des angles trop grand et le conditionnement des matrices élémentaires croît en $O(1/\theta \min)$. Dans le cas 3D, un critère sur les angles des faces du tétraèdre ne suffit pas. Trois critères ont été étudiés. La mesure de l'angle solide minimal de la tétraèdralisation, la mesure du rapport volumique entre la sphère inscrite dans le tétraèdre et la sphère circonscrite et enfin la mesure du rapport principal basé sur les valeurs propres de l'application transformant le tétraèdre en un tétraèdre régulier. Joe et Liu montre dans [11] que les trois critères sont équivalents dans une certaine mesure. Par ailleurs une stratégie de raffinement et de regroupement est donnée dans [12]. Celle-ci repose sur une bisection suivant l'arrête de plus grande taille. Cette stratégie de raffinement permet d'avoir des tétraèdres avec des angles solides bornés indépendamment du niveau de raffinement ce qui sauvegarde la qualité du maillage.

2.1.1 Conformité du maillage

Illustrons la conformité d'un maillage dans le cas d'un maillage triangulaire 2D. Un maillage triangulaire est dit conforme si l'intersection entre deux triangle quelconque est soit un segment ayant pour extrémité 2 nœuds, soit un nœud, soit l'ensemble vide. La figure qui suit montre la différence entre un maillage conforme et un maillage non conforme :

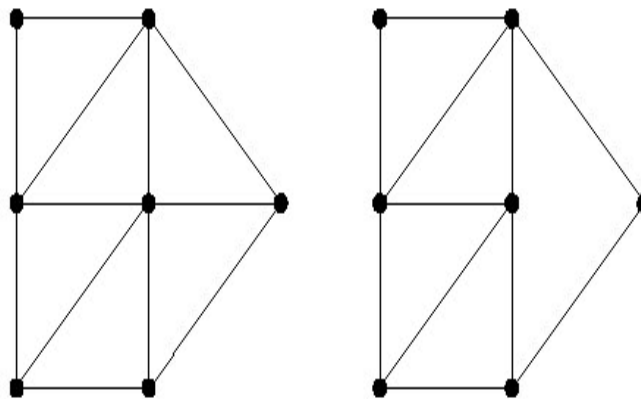


Figure 19 Maillage conforme (gauche) et Maillage non conforme (droite)

La stratégie d'adaptation de maillage doit tenir compte de cette propriété du maillage. Ceci se traduit par le fait que certaines mailles doivent être raffinées même si elles ne sont pas marquées pour être raffinées. Ainsi, la sélection des zones à raffinées se fera de telle façon que le maillage reste conforme.

Une manière de faire les choses, serai de raffiner en mettant un nœud au centre des éléments marqués. La figure suivante illustre cette stratégie :

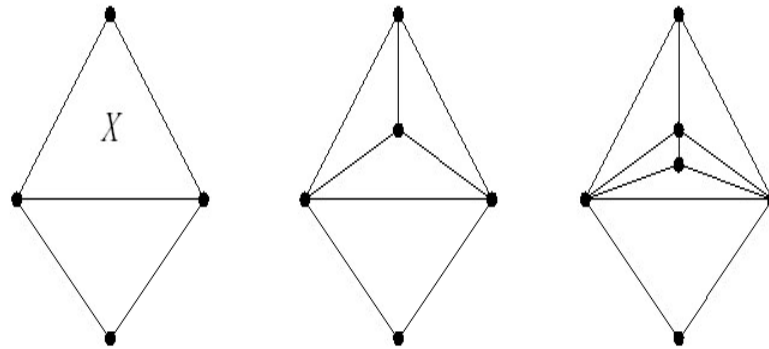


Figure 20 Raffinement par ajout d'un point central

L'avantage de cette stratégie de raffinement réside dans le fait qu'on ne remet en cause à aucun moment la conformité du maillage. Cependant, l'accumulation de raffinement produit des éléments de très mauvaises qualités tendant vers des triangles dégénérés nuisant à la précision de la méthode numérique. Nous verrons par la suite, les différentes stratégies garantissant une bonne qualité du maillage.

2.1.2 Critères de qualité du maillage

Dans le cas des maillages triangulaires (2D), il est important de générer des angles ni trop grands, ni trop petits. Dans le cas contraire, l'approximation de la solution est de très mauvaise qualité. D'un autre côté, le maillage doit être le plus régulier possible vis à vis de la taille des éléments, c'est à dire deux éléments voisins doivent rester de taille comparable. Si deux éléments voisins diffèrent de beaucoup, l'erreur locale commise par l'approximation de la solution est significative. Il faudrait donc attacher une grande précaution à la stratégie de raffinement. On verra dans la partie 2.2 qu'il existe des méthodes garantissant des angles bornés et un maillage assez régulier.

Le schéma suivant illustre une certaine régularité du maillage, c'est à dire qu'entre une région raffiner et une région qui ne l'est pas, les éléments doivent rester comparable :

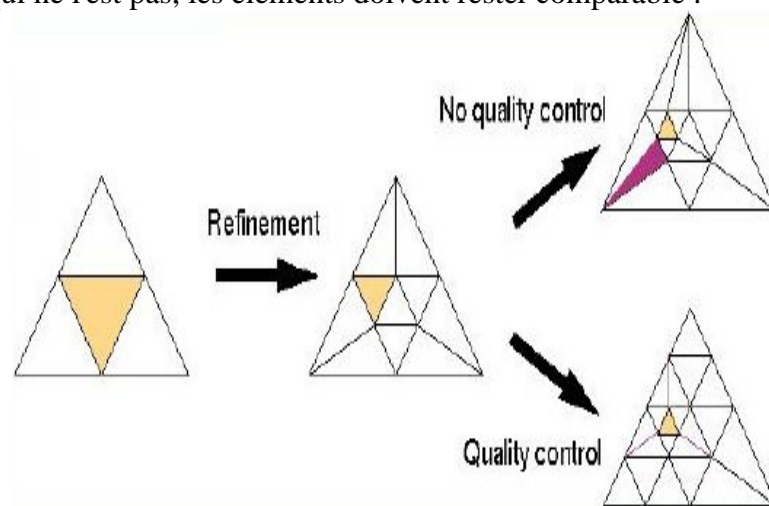


Figure 21 Qualité du maillage

Dans le cas des maillages tétraédriques (3D), la propriété angulaire citée dans le cas 2D, c'est à dire celle concernant les faces d'un tétraèdre, n'est pas une bonne mesure pour la qualité du maillage. En effet, considérons le tétraèdre défini comme suit :

$$v_0 = (0,0,0)$$

$$v_1 = (0,1,0)$$

$$v_2 = (0.5,0.5,\varepsilon)$$

$$v_3 = (0.5,-0.5,2\varepsilon)$$

Les angles des faces de ce tétraèdre sont de bonne qualité, cependant, dès que $\varepsilon \rightarrow 0$ les angles solides internes du tétraèdre et son volume tendent vers zéro. Nous verrons dans 2.2 une stratégie de raffinement sauvegardant une bonne qualité des tétraèdres.

2.2 Stratégie de raffinement

D'après ce qui précède, l'obtention d'une solution approchée correcte dépendra de la capacité du raffinement adaptatif de maillage de sauvegarder une bonne qualité de maillage.

2.2.1 Cas des maillages triangulaires

Nous avons vu que le raffinement par ajout d'un nœuds au centre des éléments ne garantit pas un maillage de bonne qualité.

La méthode présentée dans la littérature permettant une sauvegarde de la qualité du maillage repose sur une bisection des éléments. Comme illustré dans la figure suivante, la bisection permet de diviser un triangle marqué en deux triangles :

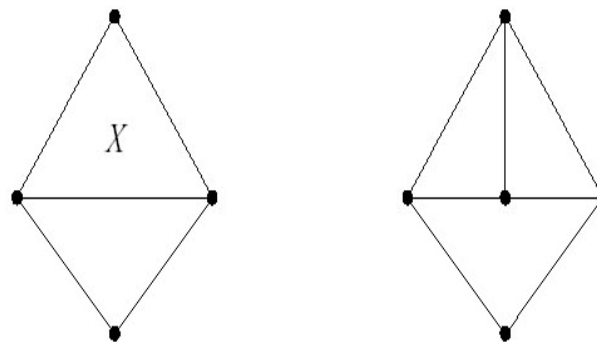


Figure 22 Bisection

Cette approche, comme le montre le dessin de droite, conduit vers des maillages non conforme. On peut résoudre ce problème en étendant le raffinement aux triangles non conformes. Par ailleurs, si la bisection se fait seulement selon le côté opposé de plus grande taille, on peut borner le maximum et le minimum des angles créés pendant la bisection et ce indépendamment du nombre de bisection. En effet, si un triangle et ses descendants sont successivement bisectionnés selon le côté opposé de plus grande taille, le plus petit angle résultant est borné au pire, par la moitié du plus petit angle du triangle racine. Un simple corollaire permet d'affirmer que le plus grand angle

généralisé reste borné aussi. Pour plus de détails sur cette partie, qui relève de la géométrie, voir l'article [25].

Une autre approche consiste en un raffinement régulier des triangles. Un triangle est partagé en quatre triangles de la manière suivante :

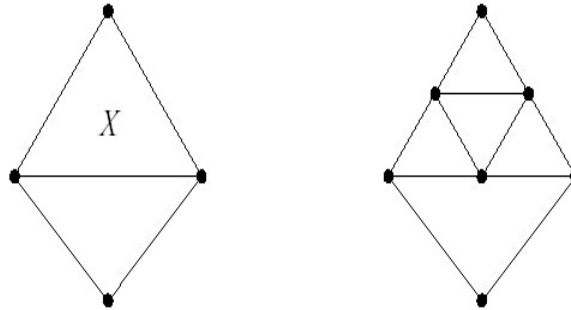


Figure 23 Raffinement régulier

Cette stratégie permet de garder les bonnes propriétés liées aux angles des triangles. Cependant, elle conduit à un maillage non conforme. On peut obtenir un maillage conforme en réalisant une bisection temporaire de l'arrête non conforme. Les arrêtes ainsi créées sont supprimées au niveau de raffinement suivant et la procédure se poursuit en raffinant d'une façon régulière les triangles ayant au moins deux arrêtes non conforme.

2.2.1.1 Algorithmes de bisection

Rivara, dans [26], a décrit un algorithme de raffinement de maillage basé sur la bisection. Initialement, on est en présence d'un maillage avec des éléments marqués pour être raffinés. Dans la phase de raffinement, dès qu'un triangle devient non conforme, il est marqué à son tour pour être raffiné. L'algorithme continue jusqu'à la construction d'un maillage conforme. Voici une version simple de cet algorithme :

Soit T_0 l'ensemble des triangles marqués

$i = 0$

Tant que (T_i) *n'est pas vide* **Faire**

BisectionGrandeArrêtes(T_i);

Soit T_i l'ensemble des triangles non conforme

$i = i + 1$

Fin Faire

Illustration :

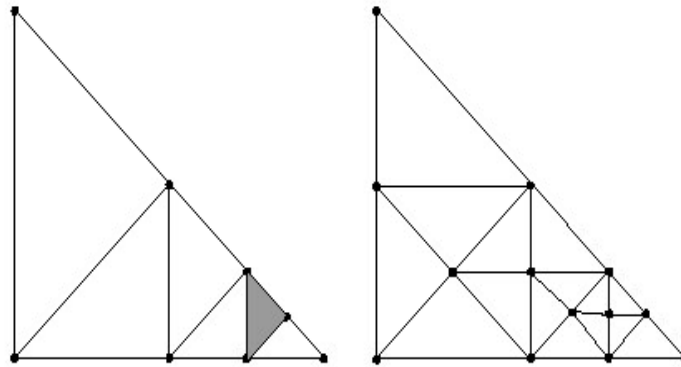


Figure 24 Algorithme de Rivara

Rivara à montré que cette algorithme termine, pourtant, il n'existe pas de borne acceptable en pratique pour le nombre de tour de la boucle "Tant que". Une solution consiste à raffiner les triangles non conformes selon l'arrête non conforme. Cette stratégie permet de réduire le nombre de tour.

2.2.1.2 Algorithmes de raffinement régulier

Dans cette approche, les triangles initialement marqués sont raffinés en quatre triangles comme indiqué sur la figure 22. Quant à la propagation du raffinement pour garder l'aspect conforme du maillage, les triangles avec au minimum deux arrêtes non conformes sont raffinés régulièrement, c'est à dire en quatre triangles, par contre, ceux avec une seul arrête non conforme sont raffinés avec une bisection suivant cette arrête. Au niveau de raffinement suivant, on fusionne les triangles bisectionnés avant de relancer le raffinement.

2.2.2 Cas des maillages tétraédriques

Pour bisectionner un tétraèdre défini par les sommets : (v_0, v_1, v_2, v_3) suivant l'arrête (v_0, v_1) , on commence par créer un nouveau point v_{01} milieu de (v_0, v_1) . On effectue ensuite une coupe du tétraèdre à l'aide du plan défini par les points : (v_{01}, v_2, v_3) comme le montre la figure suivante :

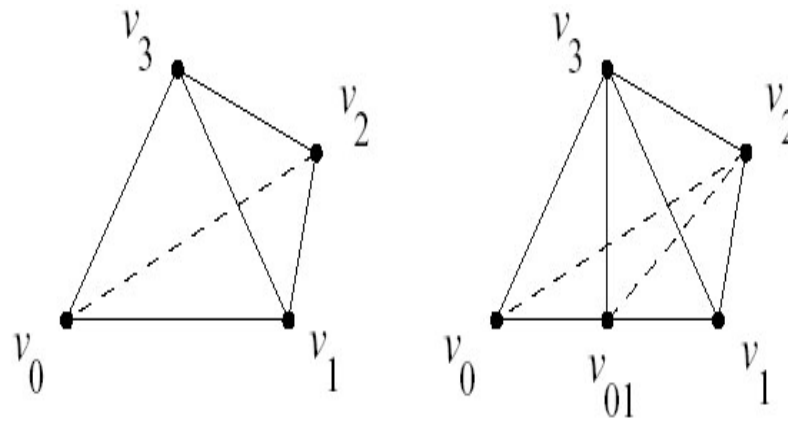


Figure 25 Bisection d'un tétraèdre

Pour que la procédure de raffinement soit correcte, il faudrait que les tétraèdres voisins se mettent d'accord sur la façon de disséquer la face qui les sépare. Sinon un maillage non conforme apparaîtra.

Le principal problème de cette bisection est la dégradation de la qualité du maillage. En effet, avec ce type de procédé, l'algorithme n'a aucun contrôle sur la dégénérescence des angles solides des tétraèdres. D'autres stratégies ont été adoptées pour raffiner des tétraèdres, notamment des stratégies issues du raffinement régulier en (2D).

Voici différents raffinement possibles pour un tétraèdre :

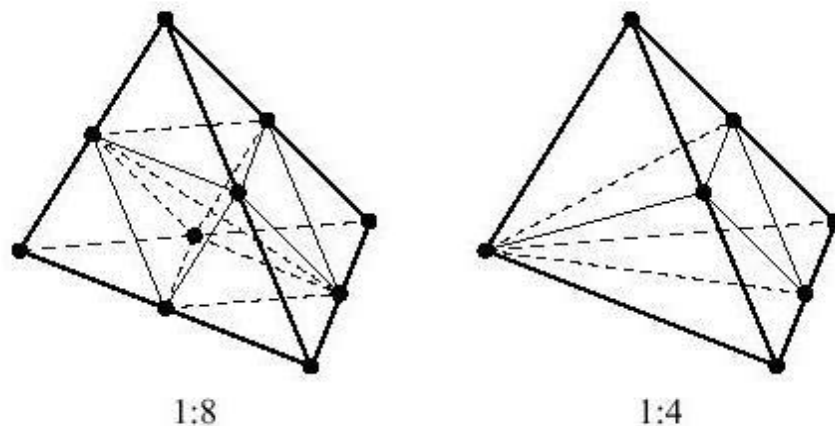


Figure 26 Raffinement régulier (gauche) et Raffinement à une face régulier

Dans cette partie consacrée aux maillages non structurés, nous avons vu des stratégies de raffinement conduisant à des maillages d'une bonne qualité dans une certaine mesure.

En deux dimensions, ces stratégies assurent une qualité du maillage presque aussi bonne que le maillage initial. Cependant, en trois dimensions, la théorie n'assure pas encore des bornes assez bonnes pour conduire à des maillages de bonne qualité. Beaucoup de problèmes restent encore ouverts. D'un autre côté, de bons résultats ont été obtenus dans le cas 3D avec les stratégies citées dans notre mémoire.

En conclusion de cette partie, la faisabilité de la stratégie AMR avec des grilles non structurées offre de grandes perspectives pour des simulations à géométrie de plus en plus

complexe. Par ailleurs, pour des phénomènes dont la solution varie rapidement, les grilles non structurées sont adaptés puisqu'elles permettent d'approcher la solution avec un minimum de nœuds.

PARTIE N°3. SCHEMA IMPLICITE ET APPROCHE PARALLELE DE L'AMR

Les différents codes existant et utilisant le raffinement adaptatif de maillage suivent des schémas numériques explicites. La stabilité de l'approche explicite requière des pas de temps imposés par la résolution spatiale du maillage. En effet, le rapport temps/espace influe directement sur la convergence du schéma numérique explicite. Ainsi, pour les zones à forte résolution, les pas de temps ont une petite amplitude, ce qui multiplie le nombre d'itérations à effectuer dans le temps. L'avantage du schéma explicite réside sur la simplicité relative du calcul fait à chaque pas de temps. En effet, à chaque pas de temps on est en présence d'un calcul matriciel direct qui se prête en plus au traitement parallèle. Pour le schéma implicite, les choses ne se passent pas de la même manière.

Dans cette partie, nous allons étudier les problématiques soulevées par la parallélisation du raffinement adaptatif de maillage et nous allons ensuite discuter du schéma implicite dans le contexte AMR.

3.1 Approche parallèle de la stratégie AMR

Le développement du parallélisme et du calcul massivement parallèle des ces dernières années, ouvre une perspective d'avenir très prometteuse pour la simulation numérique. En effet, des applications trop coûteuse du point de vu ressources informatiques peuvent ne plus l'être grâce au parallélisme. En particulier, le raffinement adaptatif de maillage peut tirer profit du calcul parallèle haute performance. Cependant, une stratégie AMR dans un cadre parallèle ne se fait pas non sans difficulté. Deux principales questions se posent : Comment partitionner le maillage d'une façon optimale ? Cette partition suffit-elle tout au long des calculs ?

3.1.1 Partitionnement du maillage

La capacité à identifier des dépendances de tâches ou de données entre les éléments du maillage est la principale condition pour la réussite d'un partitionnement. En effet, le but ultime est celui d'avoir une partition qui limite la quantité de communication entre ses différentes parties. D'un autre côté, pour occuper d'une façon équitable toutes les ressources de calcul, chaque partition doit avoir, à peu de choses près, la même taille car cette dernière est directement liée à la charge d'un processeur.

Du point de vue modélisation, ce type de problème est assimilé à un partitionnement de graphe où chaque sommet correspond à une unité de données/tâches et chaque arrête correspond aux

dépendances entre données ou concurrence entre tâches. Par ailleurs, pour une modélisation plus précise, les arrêtes et les sommets du graphe sont pondérés. L'ensemble des poids des arrêtes correspond au coût des communications et l'ensemble des poids des sommets correspond à la charge de calcul total.

Dans ce contexte, le partitionnement revient à construire des sous graphes connexes de poids à peu près égale en minimisant le poids total des arrêtes connectant ces sous graphes.

Soit $G = (S, A, P_s, P_A)$ le graphe modélisant le problème avec :

- S l'ensemble des sommets.
- A l'ensemble des arrêtes
- P_s la fonction poids des sommets
- P_A la fonction poids des arrêtes

La phase initiale du partitionnement consiste à trouver un ensemble d'arrêtes déconnectant le graphe G . Cette ensemble est une coupe $C(S_1, S_2)$ où S_1 et S_2 sont les ensembles de sommets de chaque partition. Deux contraintes sont à respecter :

- La coupe $C(S_1, S_2)$ doit être de poids minimal.
- S_1 et S_2 doivent être de poids comparables.

Deux approches sont possibles pour réaliser se partitionnement initiale :

- La première consiste à se fixer comme contrainte l'équilibre entre les poids de S_1 et S_2 , puis essayer de trouver une coupe minimale.
- La deuxième consiste à se fixer une coupe minimale et puis tenter d'équilibrer le poids des deux parties S_1 et S_2 .

La figure qui suit illustre la qualité de la coupe :

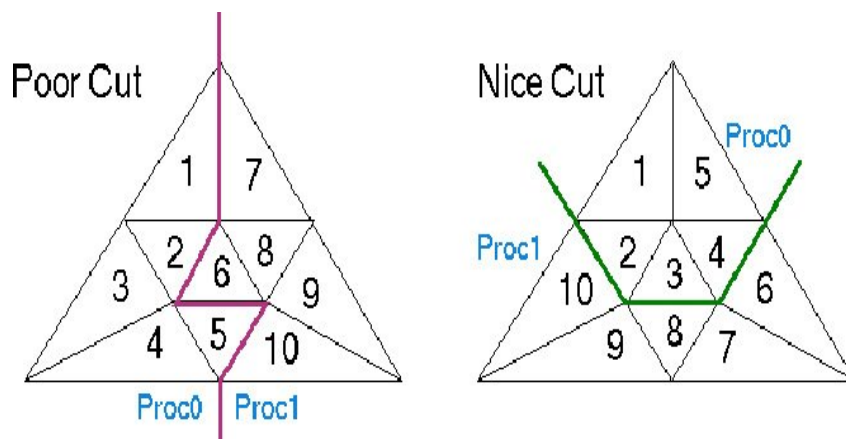


Figure 27 Mauvaise partition (gauche), Bonne partition (droite)

Le partitionnement se poursuit ensuite en appliquant le même traitement sur les sous graphes obtenus. Le processus se poursuit jusqu'à l'obtention de partitions de taille raisonnable pouvant être mappées sur les différentes ressources de calculs.

Différents algorithmes de partitionnement sont décrit dans le papier [27].

Ce partitionnement permet d'avoir un ensemble de sous maillages. Chaque maillage est livré à chaque ressource de calcul pour y être traité. Au cours des différents traitements en parallèle, la stratégie AMR appliquée à chaque sous maillage conduit à un déséquilibre de charge entre les différents processeurs. Ainsi, ce partitionnement initial est remis en cause pratiquement à chaque raffinement ce qui implique un nouveau repartitionnement.

3.1.2 Equilibrage ou Régulation de charge

Avant de décrire le processus d'équilibrage de charge, rappelons-nous brièvement le diagramme AMR :

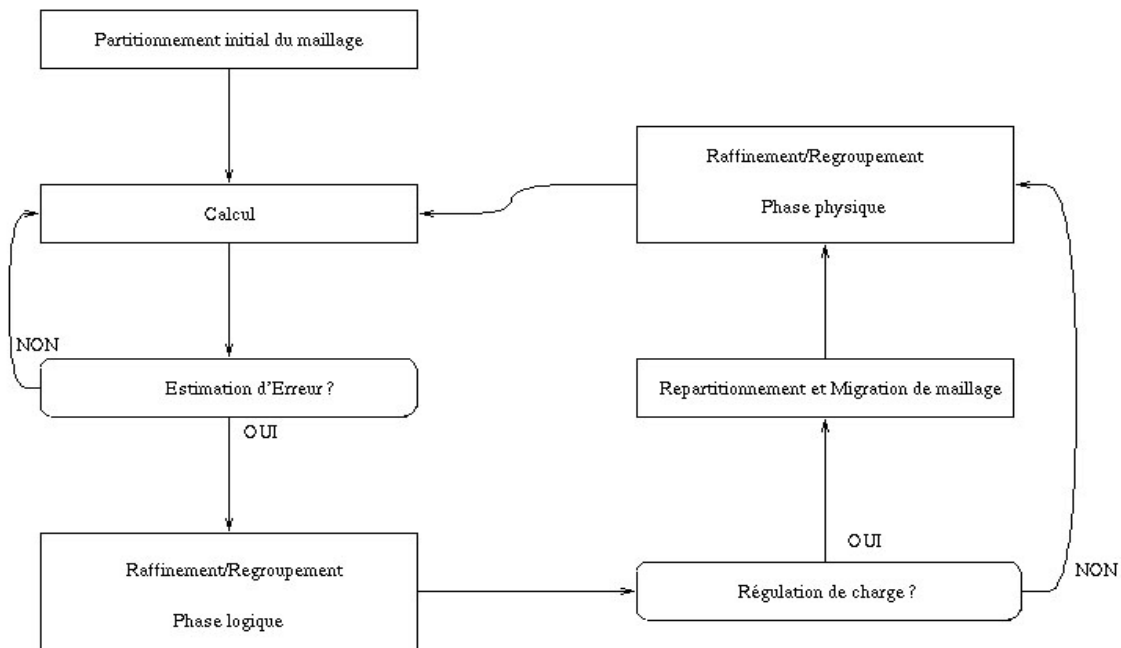


Figure 28 Diagramme AMR

Plaçons-nous au moment où la phase logique du Raffinement/Regroupement vient d'être effectuée. Ainsi, certaines parties du maillage distribué sur les processeurs sont marquées pour être raffinées ou regroupées.

Une remarque s'impose lors de la lecture du diagramme précédant. La phase de régulation de charge se trouve entre la phase logique et la phase physique du Raffinement/Regroupement. Ceci est dû au fait que le coût du Raffinement/Regroupement est inclu dans la charge d'un processeur. Ainsi, la charge d'un processeur consiste en le coût en temps de résoudre la partie du système linéaire dont il a la charge et le coût temporelle du Raffinement/Regroupement.

L'idée de base de la régulation de charge est comme suit. Avant de générer physiquement le maillage M_1 à partir du maillage M_0 marqué par la phase logique, on estime la taille et la distribution de M_1 . En se basant sur cette estimation, on peut calculer une partition de bonne qualité de M_1 avant même que ce maillage soit généré. Notons, que cette partition ne peut se faire réellement que sur le maillage M_0 . En réalité, la partition calculée à partir des estimations est projetée sur le maillage M_0 où le repartitionnement peut se faire réellement. Ainsi, chaque sous maillage de M_0 sur chaque processeur est repartitionné mais cette fois selon des critères d'équilibrage et de minimisation de coût de communication liée à M_1 . Avant de raffiner et/ou de regrouper physiquement, l'algorithme effectue des migrations de certaines mailles en tenant compte

de l'optimisation du coût de communication lié au mouvement de maille et à la résolution du système numérique sur M_1 .

Voici un schéma visualisant la régulation de charge :

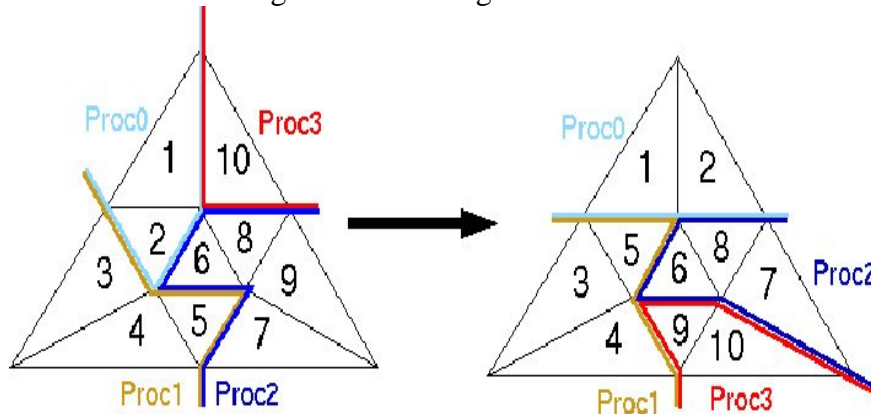


Figure 29 Régulation de charge

Notons par ailleurs, que la régulation de charge ne se fait pas à chaque intervention du raffinement/regroupement, sinon son coût couvrirait celui du calcul global. En pratique, on décide de réguler la charge périodiquement après une estimation de la taille des zones du maillages touchées par le raffinement/regroupement.

Pour plus de précision sur la régulation de charge, le chapitre 4 du document [27] donne les algorithmes de régulation de charge utilisés dans le contexte AMR.

Ces différentes méthodes de partitionnement et repartitionnement sont utilisées avec des résultats encourageant en mécanique des fluides, en dynamique rapide ou d'autres domaines d'application demandant une résolution d'équations aux dérivées partielles.

Les différentes plateformes que j'ai pu examiner pendant mon étude utilisent ces différentes méthodes dans un cadre AMR avec des schéma de calcul explicite. Le logiciel PYRAMID, par exemple, développé par le Jet Propulsion Laboratory de l'Institut de Technologie de Californie, est l'une des plateformes les plus performante faisant de l'AMR sur des maillages complexes 3D de grande taille dans un cadre parallèle. Mais aucun travail n'a été publié concernant les schémas numériques implicites et le raffinement adaptatif de maillage. Dans ce qui suit, nous essayerons d'expliciter les différentes difficultés liées au schéma implicite.

3.2 Le schéma implicite

Comme on l'a précisé ultérieurement, le schéma de calcul explicite est facile à mettre en œuvre. Il se prête à la parallélisation puisqu'il repose sur un calcul matriciel directe, mais nécessite des pas de temps faibles pour une bonne stabilité.

Le schéma implicite est une alternative au schéma explicite. Rappelons brièvement le principe de ce schéma, soit l'équation suivante à résoudre :

$$\partial_t u + \partial_{xx}^2 u = 0, x \in \Omega$$

On évalue la dérivée spatiale à l'instant $n + 1$, on obtient :

$$\frac{u_i^{n+1} - u_i^n}{\tau} + \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h} = 0$$

ou bien

$$U^n = A(U^{n+1})$$

A est dite matrice d'assemblage.

Ainsi, le calcul repose sur une inversion matricielle à chaque itération en temps. Cependant, le problème de stabilité du schéma explicite n'existe plus et donc les pas de temps peuvent être choisis de telle façon à limiter le nombre d'itérations dans le temps.

A chaque pas de temps, on a un système, creux en générale, de très grande taille à résoudre. Il existe des méthodes directes en algèbre linéaire creuse qui permettent de résoudre ce type de système dans un cadre parallèle. Ces méthodes font intervenir des algorithmes coûteux. D'un autre côté, si la matrice A ne change pas de forme durant toutes les itérations en temps, il suffirait de factoriser A (par exemple une factorisation LU) une fois pour toute et ensuite résoudre le système à chaque pas de temps. Mais il n'en est rien, puisque dès que le raffinement et/ou le regroupement intervient, la matrice A change de forme. En réalité, certains blocs de la matrice apparaissent (raffinement du maillage) et d'autres blocs disparaissent (regroupement du maillage). Ainsi à chaque pas de temps il faudrait refaire toute une factorisation de la matrice A . Ceci n'est pas envisageable étant donné les coûts de cette opération.

Une manière de faire les choses serait d'essayer d'identifier les blocs de A qui sont invariants par rapport au raffinement et/ou regroupement. Mais cela ne suffit pas puisqu'il existe un fort couplage entre les équations du système linéaire (caractéristique des schéma implicites) et donc il faudrait identifier aussi la relation, modulo la factorisation de A , qui existe entre les blocs invariants et les blocs non invariants c'est à dire ceux qui apparaissent et ceux qui disparaissent aux cours des itérations dans le temps. Cette identification permettra de garder à chaque fois les blocs de la factorisation de A , qui ne sont en aucun cas concernés par le raffinement et/ou le regroupement.

Si la deuxième identification est possible, c'est à dire on pourrait expliciter les dépendances (lié à la factorisation de A) existantes entre les différents blocs de A durant toute la simulation, il est à mon sens très difficile d'identifier les blocs de A qui sont invariant, car ceci suppose la connaissance préalable des différentes zones critiques (à raffiner et/ou à regrouper), or la localisation des zones critiques dépend de la physique du problème et est dans la majorité des cas imprévisible.

CONCLUSION

Les techniques du type raffinement adaptatif de maillage (AMR) sont utilisées dans des domaines multiples, comme la mécanique des milieux continus, l'électromagnétisme, l'astrophysique et encore bien d'autres. Elles consistent en l'augmentation de la résolution du maillage là où les mailles sont défailantes et en la diminution de cette même résolution quand les mailles sont trop précises. Elle permet donc d'éviter au code de simulation d'occuper sans intérêt les ressources informatiques. La stratégie du raffinement adaptatif de maillage a fait ses débuts dans les années 80 grâce à Marcha Berger qui a développé un algorithme AMR basé sur une hiérarchie de grilles structurées de plus en plus fines. L'utilisation par la suite de grilles non structurées a ouvert l'AMR à d'autres domaines d'applications. En effet, ce type de grilles est adapté aux géométries à frontières complexes et aux phénomènes décrits par des champs rapides. Cependant, les propriétés des maillages 3D non structurés ne sont pas encore totalement étudiées et donc dans ce contexte l'AMR connaît des limites. D'un autre côté, le développement ces dernières années du calcul parallèle pousse à penser que l'AMR peut atteindre d'autres champs d'application encore plus compliqués. En effet, il existe déjà des architectures parallèles permettant de mettre le raffinement adaptatif de maillage au service de certaines applications. On peut citer CHOMBO du NERSC de Berkeley Lab, qui fait de l'AMR en maillage structuré dans un cadre parallèle et PYRAMID, développé par le Jet Propulsion Laboratory de l'Institut de Technologie de Californie qui fait de l'AMR en maillage non structuré dans un cadre parallèle aussi.

Cependant, ces différents logiciels visent une certaine catégorie d'applications et ne peuvent avoir la prétention de couvrir une large gamme d'applications. Les recherches actuelles sont focalisées sur les maillages non structurés en trois dimensions qui présentent beaucoup de difficulté du point de vue de la qualité des maillages obtenus. Cependant, tous les travaux disponibles sur les codes AMR suivent des schémas de calcul explicite présentant l'inconvénient de la petitesse du pas de temps. Un schéma plus stable peut être envisagé : le schéma implicite. Des efforts doivent être encore faits pour étudier l'AMR dans un cadre parallèle et implicite.

BIBLIOGRAPHIES

- [1] Manish Parashar and JamesC. Browne, “Distributed Dynamic Data-Structures for Parallel Adaptive Mesh-Refinement,” Departement of Computing Sciences, University of Texas at Austin (To be presented at HiPC, December, 1995).
- [2] Mark T. Jones and Paul E. Plasmann, “Adaptive Rafinement of Unstructured Finite-Element Meshes,”.
- [3] Mark T. Jones : Computer Science Department, University of Tennessee, Knoxville, TN 37996 and Paul E. Plasmann : Mathematics & Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, “Parallel Algorithms for the Adaptive Refinement and Partitioning of Unstructured Meshes”.
- [4] Johan Steensland, 10th March 1998, “Adaptive Mesh Refinement on Structured Grids”.
- [5] C. Özturan, H.L. deCougny, M.S. Shephard, J.E. Flaherty, Scientific Computing Research Centre, Rensselaer Polytechnic Institute, Troy, NY 12181, “Parallel Adaptive Mesh Refinement and Redistribution on Distriduted Memory Computers”.
- [6] M. J. Berger and P. Colella. “Local adaptive mesh refinement for shock hydrodynamics”. Journal of Computational Physics, 82, 1989.
- [7] M. B. Giles. “On adjoint equations for error analysis and optimal grid adaptation in CFD”. Technical report, Numerical Analysis Group at Oxford University Computing Laboratory, 1997.
- [8] Robert D. Skeel. “Thirteen way to estimate global error”. Numer. Math., 48, 1986.
- [9] Wayne H. Enright and Philipe W. Sharp Desmond J. Higham, Brynjulf Owren. “A survey of the explicit rung-kutta methode, 1995”.
- [10] Kenneth Eriksson, Don Estep, Peter Hansbo, and Claes Johnson. “Introduction to adaptive methods for differential equations. Acta Numerica, 1995.
- [11] A. Liu and B. Joe, “Relationship between tetrahedron shape measures”, BIT, 34 (1994), pp. 268-287.
- [12] A. Liu and B. Joe, “On the shape of tetrahedral from bisection”, Mathematics of Computation, 63, (1994), pp. 141-154.
- [13] Burnett D. S. “Finite Element Analysis, from Concepts to Applications” (Reading, MA:Addison-Wesley)
- [14] Zienkiewicz O. C. and Craig A. 1996 Adaptive refinement,error estimates, multigrid solution and hierarchic finite element methode concept, “Accuracy Estimates and adaptive Refinements in Finite Elements Computations ”(London:Wiley)pp 25-59

- [15] Shephard M. S. 1985 “Automatic and adaptive mesh generation” IEEE Trans. Magn. 21 2484-89
- [16] Daniel B. Vollmer, “Adaptive Mesh Refinement using Subdivision of Unstructured Elements for Conservation Laws”, DMS, University of Reading
- [17] Marsha J.Berger and Joseph Oliger. Adaptive Mesh refinement for Hyperbolic Partial Differential Equations. Journal of Computational Physics, 53:484-512,1984.
- [18] Marsha J.Berger and Antony Jameson. Automatic Adaptive Multigrid Method for the Euler Equations. In Soubbaramayer and J.P. Boujot, editors, Lecture Notes in Physics 218 (Ninth International Conference on Numerical Methods in Fluid Dynamics), page 92-97, New York, 1985 Springer-Verlag.
- [19] Marsha J.Berger and Antony Jameson. Automatic Adaptive Grid Refinement for the Euler Equations. AIAA Journal, 23(4):561-568, Avril 1985.
- [20] Marsha J.Berger and Shahid H. Bokhari. A Partitioning Strategy for Nonuniform Problems on Multiprocessors. IEEE Transactions on Computers, C-36(5):570-580, May 1987.
- [21] John Bell, Marsha Berger, Jeff Saltzman and Mike Welcome. Three Dimensional Adaptive Mesh Refinement for Hyperbolic Conservation Laws. Technical report, Lawrence Livermore National Laboratory, 1991.
- [22] M. J. Aftosmis, J. E. Melton and M. J. Berger. Adaptive and Surface Modeling for Cartesian Mesh Methods. In Proceedings of the 12th AIAA Computational Fluid Dynamics Conference, pages 881-891. AIAA, June 1995. AIAA-95-1725-CP.
- [23] Marsha J.Berger and Isidore Rigoutsos. An Algorithm for Point Clustering and Grid Generation. Technical Report 501, Courante Institute of Mathematical Sciences, New York University, April 1990.
- [24] H.J.Neeman. Autonomous Hierarchical Adaptive Mesh Refinement For Multiscale Simulations. Urbana, Illinois, 1996.
- [25] I.G. Rosenberg and F. Stenger, A lower bound on the angles of triangles constructed by dissecting the longest side, Mathematics of Computation, 29 (1975, pp. 390-395)
- [26] M.C. Rivara, Mesh refinement processes based on the generalized bisection of simplexes, SIAM Journal of Numerical Analysis, 21 (1984), pp 604-613
- [27] Xiang Li, Dynamic Load Balancing For Parallel Adaptive Mesh refinement, B.Eng., Tsinghua University, Beijing, Urbana Illinois, 1995, pp 28-34