



**HAL**  
open science

# Methode multigrilles parallèle pour les simulations 3D de mise en forme de matériaux

Frédéric Vi

► **To cite this version:**

Frédéric Vi. Methode multigrilles parallèle pour les simulations 3D de mise en forme de matériaux. Matériaux. Université Paris sciences et lettres, 2017. Français. NNT : 2017PSLEM009 . tel-01777424

**HAL Id: tel-01777424**

**<https://pastel.hal.science/tel-01777424>**

Submitted on 24 Apr 2018

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

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

# THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres  
PSL Research University

Préparée à MINES ParisTech

Méthode multigrilles parallèle pour les simulations 3D de mise en  
forme de matériaux

**Ecole doctorale n°364**

SCIENCES FONDAMENTALES ET APPLIQUÉES

**Spécialité** MÉCANIQUE NUMÉRIQUE ET MATÉRIAUX

**Soutenu par Frédéric VI  
le 16 juin 2017**

Dirigée par **Lionel Fourment**  
et **Katia Mocellin**



## COMPOSITION DU JURY :

M. Hugues Digonnet  
École Centrale de Nantes  
Examineur

M. Etienne Perchat  
Transvalor  
Examineur

M. Frédéric Lebon  
Aix-Marseille Université  
Rapporteur

M. Yvan Notay  
Université Libre de Bruxelles  
Rapporteur

M. François-Xavier Roux  
ONERA  
Président du jury



# Remerciements

Pour commencer, je tiens à remercier Lionel Fourment, mon directeur de thèse, pour son encadrement, ses conseils et l'expertise dont il m'a fait bénéficier et Katia Mocellin, pour avoir accepté de reprendre puis de participer à l'encadrement et sans qui cette thèse n'aurait pu aboutir. Son expérience, sa disponibilité et son soutien ont permis la réussite de ce projet. Merci pour vos nombreuses relectures, votre patience et votre rigueur ayant permis la finalisation de ce manuscrit. Je n'oublie bien évidemment pas Hugues Digonnet, pour sa présence au démarrage de cette thèse et qui m'a permis de m'y lancer de la meilleure des façons.

Cette thèse CIFRE a pu se réaliser grâce à Transvalor que je remercie, et particulièrement Etienne Perchat, pour la confiance qu'il m'a accordée, ses conseils et son aide précieuse.

J'exprime ensuite ma gratitude à la direction du CEMEF pour m'avoir permis d'effectuer ce travail au sein de ce prestigieux laboratoire et d'y intégrer le pôle CMP.

Je remercie également vivement François-Xavier Roux, Frédéric Lebon et Yvan Notay pour leur présence dans le jury de cette thèse, le temps et l'intérêt qu'ils ont accordé à mon travail et à son évaluation.

Enfin, je remercie Altair, en particulier Rémy Perrin-Bit, à la fois pour m'avoir donné la chance d'intégrer son équipe et de participer à un projet en continuité avec mon travail de thèse, mais aussi pour m'avoir permis de terminer ce manuscrit et de préparer cette soutenance dans les meilleures conditions. Je n'oublie pas le soutien et les encouragements des équipes d'Altair Meylan et Sophia-Antipolis et leur exprime également ma reconnaissance.

Je termine par remercier ma famille, mes amis et mes collègues pour leur soutien et les moments que nous avons pu partager au cours de ces 3 années et qui ont, à leur manière, contribué à la réussite de cette thèse.



There is no limits to our ambitions  
except for the ones we build for our-  
selves and the one people will build  
for us. Tout est possible à qui rêve,  
ose, travaille et n'abandonne jamais.

---

Xavier Dolan



# Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 Méthodes Multigrilles</b>	<b>5</b>
1.1 Présentation	7
1.1.1 Principe	7
1.1.2 Méthode bigrille	8
1.1.3 Méthodes multigrilles	9
1.2 Multigrilles Algébriques (AMG)	11
1.3 Multigrilles Géométriques (GMG)	14
1.3.1 Grilles emboîtées par élément	14
1.3.2 Grilles emboîtées par noeud	15
1.3.3 Grilles indépendantes	17
1.4 Bilan et approche retenue	19
<b>2 FORGE®</b>	<b>21</b>
2.1 Modélisation du problème mécanique	23
2.1.1 Equations	23
2.1.2 Eléments Finis Mixtes	26
2.1.3 Discrétisation	27
2.2 Résolution numérique du problème mécanique	31
2.2.1 Système à résoudre	31
2.2.2 Méthode de résolution	33
2.3 Génération de maillage et remaillage	35
2.3.1 Mailleur topologique MTC	35
2.4 Calcul Parallèle	38

2.4.1	Stratégie SPMD . . . . .	38
2.4.2	Partitionnement et remaillage . . . . .	39
2.5	Simulations de référence . . . . .	41
2.5.1	Ecrasement entre tas plats . . . . .	41
2.5.2	Triaxe frottant avec repli . . . . .	42
2.5.3	Laminage retour . . . . .	43
2.5.4	Martelage rotatif . . . . .	45
<b>3</b>	<b>Méthode Multigrille pour FORGE®</b>	<b>47</b>
3.1	Méthode multigrille hybride . . . . .	49
3.1.1	Construction parallèle des maillages grossiers . . . . .	49
3.1.2	Construction parallèle des opérateurs de transfert . . . . .	51
3.1.3	Éléments complémentaires . . . . .	54
3.2	Réglage du préconditionneur Multigrille . . . . .	54
3.2.1	Lisseur . . . . .	55
3.2.2	Facteur de déraffinement . . . . .	59
3.2.3	Grille grossière . . . . .	60
3.3	Extensions de la méthode multigrille . . . . .	64
3.3.1	GMRES + Multigrille . . . . .	64
3.3.2	Solveur Multigrille . . . . .	67
3.4	Conclusion . . . . .	70
<b>4</b>	<b>Etude de performances et applications</b>	<b>73</b>
4.1	Etude de performances . . . . .	75
4.1.1	Complexité asymptotique . . . . .	75
4.1.2	Calcul parallèle . . . . .	80
4.2	Applications . . . . .	84
4.2.1	Forgeage du Triaxe . . . . .	84
4.2.2	Laminage retour . . . . .	92
4.2.3	Martelage rotatif . . . . .	97
4.3	Conclusion . . . . .	103
	<b>Conclusions et Perspectives</b>	<b>107</b>
	<b>Bibliographie</b>	<b>111</b>

# Introduction générale

La réduction des temps de calcul est un problème récurrent en simulation numérique qui peut être considéré de multiples façons : par le développement de méthodes numériques efficaces, l'utilisation de formulations adaptées aux problèmes à résoudre ou encore la réduction de modèle, avec, en plus, une utilisation optimale des ressources de calcul parallèle. Cependant, dans le cadre d'une formulation implicite et d'applications industrielles conduisant à des problèmes de tailles conséquentes, 70 à 80% du temps de calcul est consacré à la résolution de systèmes linéaires. Il représente ainsi le principal axe de recherche pour la réduction des temps de calcul. L'utilisation de méthodes itératives, combinée à la distribution des tâches par le calcul parallèle, a déjà permis de réduire les temps de calcul. Toutefois malgré des ressources de calcul toujours plus grandes, il y a toujours un nombre de processeurs optimal au-delà duquel les performances diminuent. De plus, dans une approche par partitionnement de domaine, le calcul parallèle n'a aucune influence sur le nombre d'itérations du solveur linéaire qui augmente avec  $N$  le nombre de degrés de liberté du problème. L'augmentation du nombre de processeurs ne suffit donc pas pour réduire les temps de calculs de problèmes de plus en plus grands. Une amélioration du solveur linéaire est donc indispensable. Les solveurs itératifs classiquement utilisés présentent un coût asymptotique en  $O(N^\alpha)$ , avec  $\alpha > 1$ . Ils conduisent à l'explosion des temps de calcul avec la taille du problème telle que nous venons de l'expliquer et rend ainsi inenvisageable la réalisation de certaines simulations. La recherche de méthodes de résolution plus efficaces nous amène à nous intéresser aux méthodes multigrilles. En effet, dans certaines configurations, cette méthode présente un comportement asymptotique linéaire en  $O(N)$  et apparaît plus adaptée à la résolution de gros systèmes. Nous étudions donc dans cette thèse l'utilisation de méthodes multigrilles dans le cadre des problèmes de mise en forme de matériaux.

La mise en forme de pièces forgées concerne en effet de multiples secteurs industriels nécessitant des pièces très résistantes aux chocs et en fatigue : transport, aérospatial, énergie,

etc. Face à la complexité des pièces à fabriquer, l'utilisation de la simulation numérique est aujourd'hui indispensable. En effet, elle permet une analyse préliminaire des différents aspects de la production de pièces forgées (choix et optimisation du procédé de fabrication, de sa température, des matériaux, etc.), réduisant ainsi les temps de développement de nouveaux produits. Pour répondre à ces besoins, la société Transvalor, en collaboration avec le Centre de Mise en Forme des Matériaux (CEMEF), développe le logiciel FORGE® NxT, dédié à la simulation de procédés de mise en forme à chaud et à froid. Devenu une référence dans ce domaine, il permet de simuler différents procédés de la mise en forme de métaux à l'état massif. Alors que la plupart de ces simulations peuvent être réalisées dans les délais compatibles avec leur utilisation en bureau d'études, d'autres peuvent demander des semaines voire des mois de calculs, malgré l'utilisation du calcul parallèle (Péremé *et al.*, 2011). L'utilisateur est alors contraint de limiter la taille des calculs, par l'utilisation de maillages insuffisamment raffinés ou d'hypothèses simplifiant le problème. Ces limitations empêchent donc encore aujourd'hui les entreprises d'étudier certains des procédés les plus complexes de manière suffisamment précise et en adéquation avec la réalité. C'est dans ce contexte que se place cette thèse : réduire les temps de calculs en améliorant l'efficacité des méthodes de résolution.

Considérant l'importance de la part de la résolution des systèmes linéaires sur le temps de calcul, leur accélération est l'objectif principal de toute réduction des coûts. Avec une complexité asymptotique de l'ordre de  $N^{\frac{3}{2}}$  (Coupez et Marie, 1997), l'utilisation d'un solveur itératif pour la résolution de problèmes de grande taille conduit rapidement à une saturation des moyens de calcul. En trois dimensions, la recherche de résultats plus précis en divisant la taille de maille par 2 revient à multiplier le nombre de degrés de liberté par 8 et le temps de calcul associé à la résolution de systèmes par 22. La méthode multigrille présente un meilleur comportement asymptotique, idéalement en  $O(N)$  mais plus généralement en  $O(N \log N)$  selon sa configuration. Elle devrait permettre de réduire les temps de calcul lorsque les problèmes traités atteignent des grands nombres de degrés de liberté, ce qui est le contexte de notre étude (figure 1).

Initiée par la société Transvalor, cette thèse a pour objet le développement d'une méthode multigrille dans ce contexte industriel sur des problèmes de mise en forme de pièces forgées. Afin d'être un réel atout pour la réduction des temps de calcul, cette méthode devra être compatible avec le calcul parallèle dont bénéficie déjà le logiciel FORGE® NxT. L'objectif de ce travail de recherche est ainsi de proposer une alternative efficace au solveur linéaire actuel, en permettant l'accélération de la résolution complète de simulations de différents procédés de mise en forme. Afin d'y parvenir, la méthode multigrille développée devra être compatible avec les multiples particularités des problèmes à traiter, telles que les grandes déformations impliquant de fréquents remaillages et les spécificités physiques de l'écoulement (incompres-

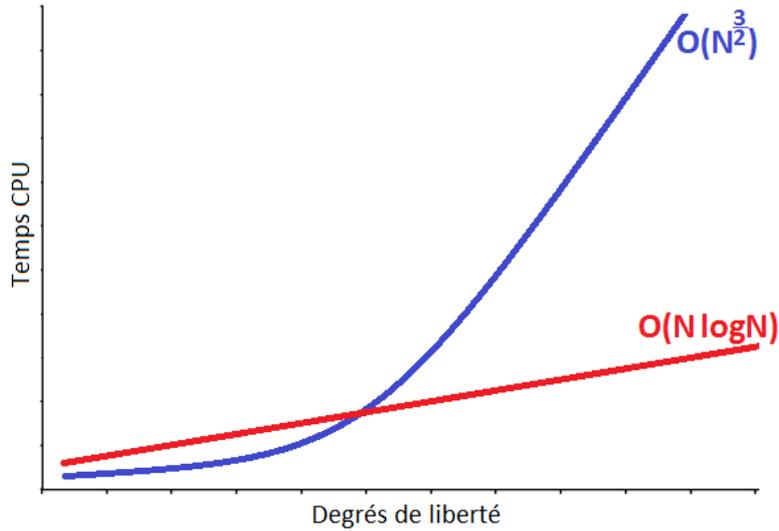


FIGURE 1 – Evolution du temps CPU en fonction du nombre de degrés de liberté pour des méthodes de complexité asymptotique en  $O(N^{\frac{3}{2}})$  et en  $O(N \log N)$

sibilité, contact,...) donnant lieu à des matrices indéfinies et imparfaitement conditionnées.

Dans un premier temps, nous présentons dans le chapitre 1 un état de l'art qui a orienté nos choix pour le développement d'une méthode multigrille particulière et adaptée à la résolution des problèmes de mise en forme. Le chapitre 2 se concentre sur les équations du problème thermo-mécanique de mise en forme résolues par le code FORGE® NxT, les méthodes de résolution retenues et les autres spécificités nécessaires à la résolution des problèmes de grandes transformations. Dans le troisième chapitre, la recherche des paramètres permettant d'obtenir une méthode multigrille efficace pour la résolution de ces systèmes linéaires est présentée. Enfin, le chapitre 4 expose les résultats relatifs au comportement et aux performances de la méthode pour la résolution de procédés de forgeage.



# 1 | Méthodes Multigrilles

A single idea from the human mind can build cities. An idea can transform the world and rewrite all the rules.

---

Dominic Cobb

Nous présentons dans ce premier chapitre le principe des méthodes multigrilles à la base de la méthode développée dans la suite de la thèse, afin de réduire les temps de calcul des simulations de mise en forme et d'obtenir une convergence asymptotique plus favorable à la résolution de grands systèmes linéaires.

Nous effectuons ici une description des méthodes multigrilles de la littérature et des différentes possibilités de leur mise en oeuvre. Après une brève présentation générale, nous discuterons des multigrilles algébriques et géométriques avant d'introduire l'approche plus spécialement retenue pour cette thèse que nous définissons comme hybride, algébro-géométrique.

## Sommaire

---

<b>1.1</b>	<b>Présentation</b>	<b>7</b>
1.1.1	Principe	7
1.1.2	Méthode bigrille	8
1.1.3	Méthodes multigrilles	9
<b>1.2</b>	<b>Multigrilles Algébriques (AMG)</b>	<b>11</b>
<b>1.3</b>	<b>Multigrilles Géométriques (GMG)</b>	<b>14</b>
1.3.1	Grilles emboîtées par élément	14
1.3.2	Grilles emboîtées par noeud	15
1.3.3	Grilles indépendantes	17
<b>1.4</b>	<b>Bilan et approche retenue</b>	<b>19</b>

---

## 1.1 Présentation

Les méthodes multigrilles forment une famille d’algorithmes itératifs pour la résolution de grands systèmes d’équations linéaires. Devenues très populaires en raison d’une complexité en  $O(N)$  dans leur meilleure configuration, elles s’avèrent donc une excellente alternative aux solveurs itératifs plus classiques, d’une complexité en  $O(N^\alpha)$  avec  $\alpha > 1$ , dans les cas où elles peuvent être mises en oeuvre (Yavneh, 2006). Introduites dans les années 1960 par Fedorenko (Fedorenko, 1961) pour la résolution des équations de Poisson, elles ont ensuite été beaucoup utilisées en mécanique des fluides (par (Ghia *et al.*, 1982) ou (Vanka, 1986) par exemple) en raison de la facilité de leur application aux discrétisations par différences finies ou par volumes finis (Wesseling, 1992), avant d’être utilisées pour d’autres types de problèmes.

Les méthodes multigrilles ont aussi été étudiées pour la mécanique du solide, par exemple par Parsons et Hall ((Parsons et Hall, 1990a) et (Parsons et Hall, 1990b)), où des accélérations d’un facteur proche de 10 par rapport à un solveur direct sont observées pour la résolution de problèmes simples comportant 10 000 degrés de liberté. Des problèmes plus complexes ont ensuite été abordés, par exemple par (Lee, 1998) et (Mocellin, 1999) où un facteur d’accélération de 10 est cette fois obtenu par rapport à un solveur itératif sur un cas de forgeage comportant 260 000 degrés de liberté. En près de 10 ans, l’évolution des ressources de calcul, les optimisations des solveurs et des processeurs expliquent que la résolution d’un problème 26 fois plus gros soit nécessaire pour obtenir le même facteur d’accélération. Aujourd’hui, il est désormais difficile d’atteindre de telles accélérations, mais l’approche multigrille reste une alternative intéressante aux solveurs itératifs classiques en étant plus performante sur les problèmes de très grande taille.

### 1.1.1 Principe

Les méthodes itératives classiques (telles que Gauss-Seidel ou Jacobi) sont très efficaces durant les premières itérations pendant lesquelles elles réduisent rapidement les valeurs du résidu. Mais ensuite, la convergence ralentit dès lors qu’il s’agit de réduire les basses fréquences, qui pourraient être plus rapidement évaluées sur une grille grossière. Cette observation a mené au développement des méthodes multigrilles, qui reposent sur l’utilisation de plusieurs niveaux de calcul. Le principe est d’effectuer des itérations d’une méthode itérative ayant des bonnes propriétés de lissage des hautes fréquences sur le niveau de calcul le plus fin, afin d’évaluer ces hautes fréquences puis de restreindre le problème sur un ou plusieurs niveaux plus grossiers, comportant moins de degrés de liberté, et permettant de calculer les basses fréquences beaucoup plus efficacement.

Selon la méthode employée pour la construction des différents niveaux de calcul, on distingue les approches algébriques et géométriques, qui seront décrites dans la suite du chapitre. Présentons tout d'abord plus en détail l'algorithme général de la méthode en commençant par sa version la plus simple, la méthode bigrille.

### 1.1.2 Méthode bigrille

La méthode bigrille n'utilise que deux niveaux de calcul, c'est-à-dire qu'elle n'a qu'une seule grille grossière en complément de la grille de départ. Nous utilisons l'indice  $h$  pour désigner le niveau fin et  $H$  pour le niveau grossier.

Considérons le système linéaire à résoudre :

$$A_h X_h = b_h \quad (1.1)$$

Les différentes étapes d'un cycle bigrille sont les suivantes :

- **Pré-lissage** : on effectue  $\nu$  itérations (appelées lissages) d'une méthode itérative (appelé lisseur)  $L$  afin d'obtenir une première approximation  $\tilde{X}_h$  de la solution

$$\tilde{X}_h = L(A_h, b_h, X_h, \nu) \quad (1.2)$$

- **Calcul du résidu** :

$$r_h = b_h - A_h \tilde{X}_h \quad (1.3)$$

- **Restriction du résidu** : le résidu est projeté sur la grille grossière grâce à l'opérateur de restriction  $R$

$$r_H = R r_h \quad (1.4)$$

- **Calcul de la correction** : la correction sur la grille grossière  $\delta X_H$  est calculée en résolvant le système linéaire suivant :

$$A_H \delta X_H = r_H \quad (1.5)$$

grâce à une méthode résolution  $\Phi$ , où  $A_H$  est la matrice du système linéaire projetée sur la grille grossière :

$$\delta X_H = \Phi(A_H, r_H) \quad (1.6)$$

- **Prolongation de la correction** : la correction  $\delta X_H$  est interpolée sur la grille fine grâce à l'opérateur de prolongement  $P$

$$\delta X_h = P \delta X_H \quad (1.7)$$

- **Application de la correction** : la solution approchée  $\tilde{X}_h$  est actualisée en lui ajoutant la correction venant de la grille grossière

$$\tilde{X}_h \leftarrow \tilde{X}_h + \delta X_h \quad (1.8)$$

- **Post-lissage** : on effectue  $\nu'$  lissages d'un lisseur  $L'$  pour éliminer les hautes fréquences de la correction. Généralement,  $L = L'$

$$\tilde{\tilde{X}}_h = L'(A_h, b_h, \tilde{X}_h, \nu') \quad (1.9)$$

Ce cycle est ensuite répété jusqu'à obtenir la précision recherchée.

### 1.1.3 Méthodes multigrilles

La généralisation de la méthode bigrille mène aux méthodes multigrilles, dans lesquelles on considère davantage de niveaux de calcul. La correction est alors uniquement calculée sur le niveau le plus grossier, tandis que les étapes de pré-lissages et post-lissages sont effectuées sur le niveau fin et sur les niveaux intermédiaires. L'ordre dans lequel les différentes grilles sont parcourues décrit le cycle multigrille, dont certains sont schématisés sur la figure 1.1.

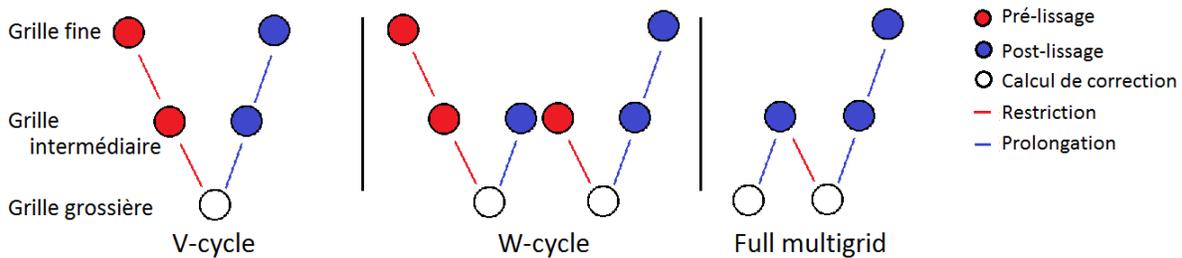


FIGURE 1.1 – Schémas de V-cycle, W-cycle et Full Multigrid

Le V-cycle consiste à remplacer l'opérateur  $\Phi$  par un autre cycle bigrille au fur et à mesure de l'ajout d'un niveau plus grossier. Le passage d'une méthode bi-grille à un V-cycle de 3 grilles est schématisé sur la figure 1.2. Ce cycle, le plus couramment utilisé dans la littérature, sera retenu dans cette thèse.

Le W-cycle consiste à effectuer plusieurs opérations de la méthode bigrille sur les grilles les plus grossières avant de prolonger la correction sur la grille fine. En multipliant les opérations sur les grilles grossières, normalement peu coûteuses, une meilleure correction est calculée et permet d'améliorer légèrement la convergence (Strang, 2007). Peu utilisé dans la littérature, car moins naturel que le V-cycle lors du passage de 2 à plus de grilles, le W-cycle est numériquement plus coûteux et il faut compter sur une réduction significative du nombre de cycles

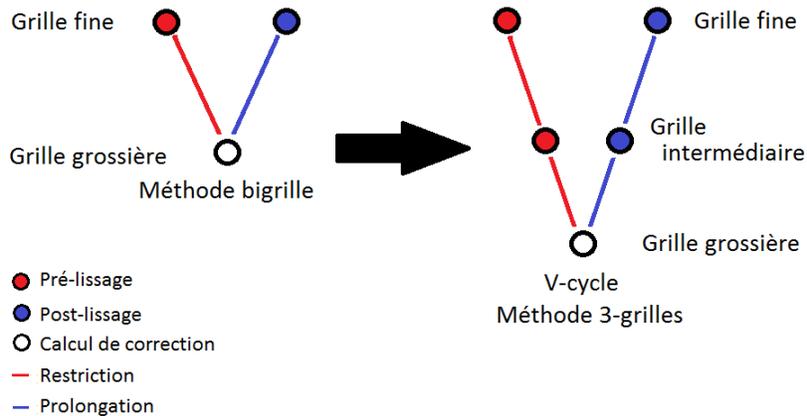


FIGURE 1.2 – Passage d’une méthode 2 grilles vers un V-cycle de 3 grilles

pour qu’il devienne justifié. Dans (Lebon *et al.*, 2007) par exemple, les auteurs obtiennent une accélération d’un facteur 8 par rapport à un solveur direct sur la résolution d’un problème de contact en 2D considérant 13 000 noeuds en utilisant un W-cycle, mais précisent que ces résultats ne sont que légèrement meilleurs qu’avec un V-cycle.

La méthode Full Multigrid, consiste à initialiser le cycle multigrille sur la grille la plus grossière et à parcourir les niveaux plus grossiers avant d’atteindre la grille la plus fine. Ce cycle présente un meilleur comportement asymptotique théorique en  $O(N)$  plutôt qu’en  $O(N \log N)$  pour les autres. Dans (Strang, 2007), l’auteur met mathématiquement en évidence cette propriété, mais également la difficulté à numériquement reproduire ce comportement optimal. En pratique, l’utilisation des Full Multigrid nécessite un assemblage du système à résoudre sur la grille grossière, ce qui, comme nous le verrons dans la suite, n’est pas envisageable pour les problèmes traités dans cette thèse.

Une fois le type de cycle choisi, la méthode multigrille est définie à partir des paramètres de résolution suivants :

- Les lisseurs  $L$  et  $L'$
- Les nombres d’itérations de lissages  $\nu$  et  $\nu'$
- La méthode de résolution  $\Phi$  (et ses éventuels paramètres) pour la résolution du système de la grille la plus grossière

Ces paramètres seront discutés dans la suite de ce manuscrit, dans le chapitre 3.

L’utilisation d’une méthode multigrille repose également sur la construction des éléments suivants :

- L'opérateur de restriction  $R$
- L'opérateur de prolongement  $P$
- Les matrices des grilles grossières

Leur mode de construction détermine deux grandes familles de multigrilles, les multigrilles algébriques et les multigrilles géométriques.

## 1.2 Multigrilles Algébriques (AMG)

Les multigrilles algébriques sont décrites en suivant Saad ([Saad, 2003](#)), c'est-à-dire à partir du mode de construction des matrices grossières et des opérateurs de transfert. Dans l'approche algébrique, ces constructions sont uniquement basées sur la matrice du système à résoudre. Elles sont donc totalement indépendantes du maillage.

Nous décrivons brièvement cette construction à partir de l'équation [1.1](#), avec  $N$  le nombre de degrés de liberté du système. Notons  $a_{ij}$  le coefficient  $(i, j)$  de la matrice  $A_h$  et  $V_i$  l'ensemble des voisins du degré de liberté  $i$  défini par :

$$\forall i \in \{1, \dots, N\}, \quad V_i = \{j \in \{1, \dots, N\} \mid \text{tel que } a_{ij} \neq 0\} \quad (1.10)$$

On définit ensuite  $S_i$  l'ensemble des points auxquels le degré de liberté  $i$  est fortement connecté, c'est-à-dire dont les coefficients de la matrice sont particulièrement forts par rapport aux autres coefficients de la ligne correspondante :

$$\forall i \in V_i, \quad S_i = \{j \in \{1, \dots, N\} \mid \text{tel que } |a_{ij}| \geq \alpha \max_{k \neq i} |a_{ik}|\}, \quad 0 < \alpha \leq 1 \quad (1.11)$$

On construit ensuite, de manière itérative, grâce à ces deux ensembles un ensemble  $C$  contenant les degrés de liberté fortement couplés et qui formeront l'ensemble des degrés de liberté du niveau grossier. Cette stratégie est détaillée dans par Ruge et Stüben ([Ruge et Stüben, 1987](#)) et un exemple de construction de cet ensemble est décrit dans ([Iwamura et al., 2003](#)). On définit ensuite l'opérateur de prolongement  $P$  reliant tout champ  $x_H$  de la grille grossière au champ  $x_h$  de la grille fine correspondant :

$$\forall i \in \{1, \dots, N\}, \quad (x_h)_i = (Px_H)_i = \begin{cases} (x_H)_i & \text{si } i \in C \\ \sum_{j \in S_i} \omega_{ij} (x_H)_j & \text{si } i \notin C \end{cases} \quad (1.12)$$

avec  $\omega_{ij}$  des coefficients à déterminer, dépendant de l'ensemble  $C$ .

Une fois l'opérateur de prolongement construit, la relation de Ritz-Galerkin définit l'opérateur de restriction  $R$  comme sa transposée :

$$R = P^T \quad (1.13)$$

Ces deux opérateurs permettent ensuite de construire la matrice du système de la grille grossière grâce à la relation de Galerkin :

$$A_H = RA_hP \quad (1.14)$$

Les multigrilles algébriques, par leur construction, présentent l'avantage de ne pas dépendre de la complexité géométrique du maillage. Elles sont donc plus faciles à mettre en place mais sont, en général, moins performantes que les méthodes géométriques correctement mises en oeuvre. Dans (Chow *et al.*, 2006), cette différence d'efficacité est mise en évidence et s'explique par la largeur de bande des matrices grossières. En effet, même si cette largeur de bande est généralement faible sur la matrice fine initiale, la construction des matrices grossières l'augmente progressivement, jusqu'à s'approcher de plus en plus d'une matrice pleine et donc très coûteuse à résoudre.

Dans le cadre plus spécifique des problèmes traités dans cette thèse, la structure de la matrice à résoudre rend difficile l'utilisation des méthodes multigrilles algébriques. En effet, comme nous le verrons dans le chapitre 2, cette matrice représentant un problème en vitesse/pression dispose d'une structure par bloc, dont les valeurs peuvent être d'ordre de grandeur différentes, notamment pour le bloc pression, mais surtout avec des termes de pénalisation provenant du contact. La bonne utilisation des algorithmes classiques de construction des grilles grossières pourrait donc être perturbée et empêcher une utilisation optimale des multigrilles algébriques. Dans (Notay, 2016), l'auteur s'intéresse à ces problèmes en proposant une stratégie consistant à transformer le système linéaire à résoudre afin de le rendre plus compatible avec les multigrilles algébriques.

L'équation 1.1 s'écrit après discrétisation d'un problème de Stokes :

$$A \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} b_u \\ b_p \end{pmatrix} \Leftrightarrow \begin{pmatrix} V & B \\ B^T & -P \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} b_u \\ b_p \end{pmatrix} \quad (1.15)$$

avec  $u$  et  $p$  respectivement les champs de vitesse et pression. La transformation du système consiste à utiliser une matrice triangulaire inférieure et une matrice triangulaire supérieure pour effectuer un "préconditionnement" avant application d'une méthode multigrille algébrique. Il définit alors  $\tilde{L}$  et  $\tilde{U}$  :

$$\tilde{L} = \begin{pmatrix} I_1 & 0 \\ \alpha BD^{-1} & -I_2 \end{pmatrix}, \quad \tilde{U} = \begin{pmatrix} I_1 & -\alpha D^{-1} B^T \\ 0 & -I_2 \end{pmatrix} \quad (1.16)$$

avec  $D = \text{diag}(A)$ ,  $\alpha = (\|D^{-1}A\|_\infty)^{-1}$  et  $I_1$  et  $I_2$  les matrices identité de tailles consistantes avec la taille initiale.

Il introduit ensuite le changement de variable suivant :

$$\begin{pmatrix} u \\ p \end{pmatrix} = \tilde{U} \begin{pmatrix} \tilde{u} \\ \tilde{p} \end{pmatrix} \quad (1.17)$$

Il peut finalement réécrire l'équation 1.15, en multipliant les deux termes par  $\tilde{L}$  :

$$\tilde{A} \begin{pmatrix} \tilde{u} \\ \tilde{p} \end{pmatrix} = \tilde{L} \begin{pmatrix} b_u \\ b_p \end{pmatrix} \quad (1.18)$$

avec  $\tilde{A} = \tilde{L}A\tilde{U}$ .

Notay montre alors que la forme de l'équation 1.18 est beaucoup plus adaptée pour une utilisation des multigrilles algébriques et pourrait permettre une utilisation future de cette stratégie pour la résolution de problèmes de mise en forme.

Le principal avantage des multigrilles algébriques réside dans sa simplicité d'utilisation, qui permet une utilisation en "boîte noire", n'ayant besoin que de la matrice. Cette approche est considérée comme robuste et peut s'appliquer à une grande variété de problèmes, sans aucune dépendance à la complexité géométrique. Cependant, cette simplicité et cette robustesse ont un coût : dans (Stüben, 2001), Stüben met en avant la complexité de la construction des multigrilles algébriques qui les rend moins efficaces que des multigrilles géométriques correctement mises en oeuvre. Pour lui, cette méthode ne peut pas être compétitive face aux multigrilles géométriques mais représente une alternative aux méthodes numériques standards, dans les cas où la complexité du problème rend impossible l'utilisation de l'approche géométrique. Par exemple, dans (Watanabe *et al.*, 2005), Watanabe compare les performances des méthodes du Gradient conjugué, des multigrilles algébriques et des multigrilles géométriques pour la résolution d'un problème magnétostatique, sur différentes tailles de problème. La figure 1.3 présentant les temps de calculs en fonction de la taille du problème permet d'illustrer les différents arguments donnés jusqu'à présent.

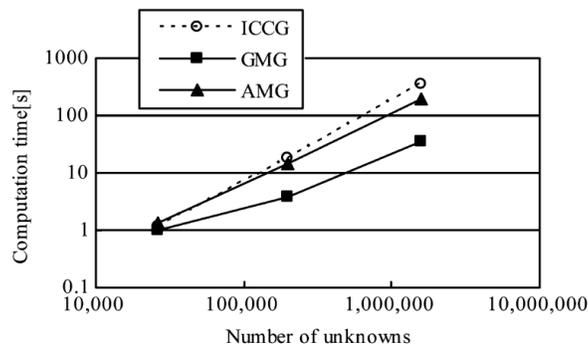


FIGURE 1.3 – Temps de calcul des méthodes du gradient conjugué (ICCG), des multigrilles algébriques (AMG) et des multigrilles géométriques (GMG) en fonction de la taille d'un problème magnétostatique (Watanabe *et al.*, 2005)

Sur un problème de taille relativement petite (environ 40 000 degrés de liberté), les 3 méthodes présentent des temps de calcul similaires et les multigrilles deviennent intéressantes sur un plus grand nombre d'inconnues. A 200 000 et à 1,5 million de degrés de liberté, les multigrilles algébriques et géométriques sont toutes les deux plus rapides que le gradient conjugué et les courbes suggèrent que le gain apporté par les multigrilles continuera à grandir avec le nombre d'inconnues. Cependant, les temps de calcul de l'approche algébrique restent proches de ceux du gradient conjugué, tandis que l'approche géométrique se montre clairement plus efficace, allant jusqu'à être près de 4 fois plus rapide. Cela coïncide donc avec les conclusions de Stüben et nous amène à nous orienter vers les multigrilles géométriques.

## 1.3 Multigrilles Géométriques (GMG)

Dans l'approche des multigrilles géométriques, les grilles grossières correspondent à des maillages construits sur la géométrie du domaine. Leur principale difficulté réside dans la construction de ces maillages, surtout lorsqu'ils sont non structurés et que le domaine est complexe. Différentes méthodes de construction peuvent être citées :

### 1.3.1 Grilles emboîtées par élément

Cette méthode repose sur le raffinement d'un maillage grossier qui est donc pris comme point de départ. Récemment utilisé par (Kimmritz et Richter, 2011) en mécanique des fluides, cette approche a également été utilisée dans les travaux de (Mocellin, 1999) pour résoudre des problèmes de mise en forme.

Pour construire les différents niveaux, le maillage de départ est donc le maillage le plus grossier. Pour un cas 3D et un maillage tétraédrique, la figure 1.4 montre la stratégie de raffinement de tétraèdres, consistant à diviser chacun d'eux en 8 sous-tétraèdres qui composeront un niveau plus fin de calcul pour les multigrilles.

Cette approche d'emboîtement par élément a l'avantage d'engendrer des opérateurs de transfert très simples. En effet, les éléments de chacun des niveaux appartiennent par construction aux niveaux plus fins. Les opérateurs de transfert sont donc très simples à construire, précis et ne sont pas coûteux à utiliser.

Cependant, plusieurs inconvénients sont à relever :

- Cette stratégie de raffinement des tétraèdres conduit nécessairement à la réduction de qualité des éléments. La figure 1.4 montre d'ailleurs un élément étiré en son centre.
- La nécessité de démarrer la génération des grilles à partir d'un maillage grossier est incompatible avec une utilisation industrielle, puisqu'il est alors difficile de traiter des

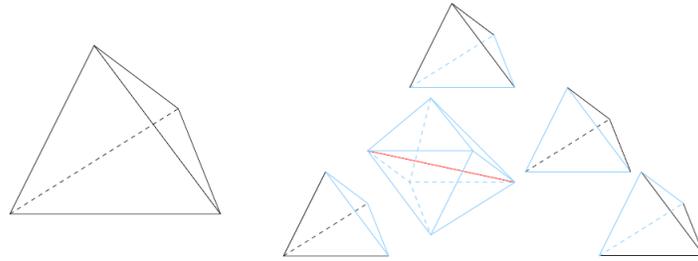


FIGURE 1.4 – Subdivision d’un tétraèdre pour le raffinement de maillage (Mocellin, 1999)

géométries complexes qui sont généralement définies (en mise en forme) par le maillage le plus fin.

- Dans le cas de géométries complexes, la description géométrique de la frontière n’est en générale pas correcte. Bien qu’une méthode de projection des noeuds de surface soit proposée par (Mocellin, 1999) pour pallier à la différence de volume provenant de la méthode de raffinement, cela entraîne de nouvelles dégradations de la qualité du maillage.

### 1.3.2 Grilles emboîtées par noeud

Pour pallier aux limitations provenant de la méthode de déaffinement de maillage emboité par élément, (Rey, 2007) propose, à partir des travaux de (Carte *et al.*, 2000) et (Coupez, 2000), de générer des grilles grossières emboîtées par noeuds. Pour cela, à partir d’un maillage fin tridimensionnel initial  $M_h$ , défini par son ensemble de noeuds  $I_h$  et de ses tétraèdres  $T_h$ , il utilise l’algorithme de déaffinement de maillage décrit sur l’algorithme 1 afin de générer un maillage grossier  $M_H$  respectant la géométrie initiale et qui a une taille de maille  $C$  fois plus grande. Cela revient, pour chaque noeud  $i$  de  $I_h$ , à calculer la longueur moyenne  $l_h(i)$  des arêtes qui lui sont connectées et à chercher à obtenir sur le maillage  $M_H$ , pour tout  $i$  de  $I_H$ ,  $l_H(i) \approx Cl_h(i)$ .

On définit  $T_H(i)$  l’ensemble des tétraèdres contenant le noeud  $i$ ,  $\varepsilon_H(i)$  l’ensemble des noeuds de  $T_H(i)$  ( $i$  exclu) et  $F_H(i)$  l’ensemble des faces extérieures de  $T_H(i)$  ne contenant pas  $i$ .

1. Initialisation :
  - Pour chaque noeud  $i \in I_h$ , calcul de  $l_c(i) = Cl_h(i)$
  - $M_H = M_h$ ;  $T_H = T_h$ ;  $I_H = I_h$
2. Pour chaque noeud  $i$  de  $I_H$ , calcul de  $l_H(i)$
3. si  $l_H(i) \geq l_c(i)$  alors
  - $i \rightarrow i + 1$  et aller à l'étape 3
 sinon
  - Remailler le volume défini par  $F_H(i)$  en utilisant seulement les noeuds de  $\varepsilon_H(i)$ . Soit  $T_{new}(i)$  l'ensemble des tétraèdres de cette nouvelle triangulation (qui ne contient pas  $i$ )
  - si  $T_{new}(i)$  est acceptable alors
    - $I_H = I_H - \{i\}$
    - $T_H = T_H - T_H(i) + T_{new}(i)$
    - $i \rightarrow i + 1$  et aller à l'étape 3
  - sinon
    - $i \rightarrow i + 1$  et aller à l'étape 3

**Algorithme 1** : Algorithme de déraffinement de maillage, générant un maillage emboîté par noeud (Rey, 2007)

A la fin de cette procédure, la nouvelle triangulation obtenue est composée de noeuds appartenant au maillage fin et dont la longueur moyenne des arêtes est approximativement  $l_H(i)$ . Ces noeuds formeront le maillage grossier utilisé par la méthode multigrille. Un exemple d'application de la génération de maillages emboîtés par élément sur un cas de forgeage en 3D est donné sur la figure 1.5.

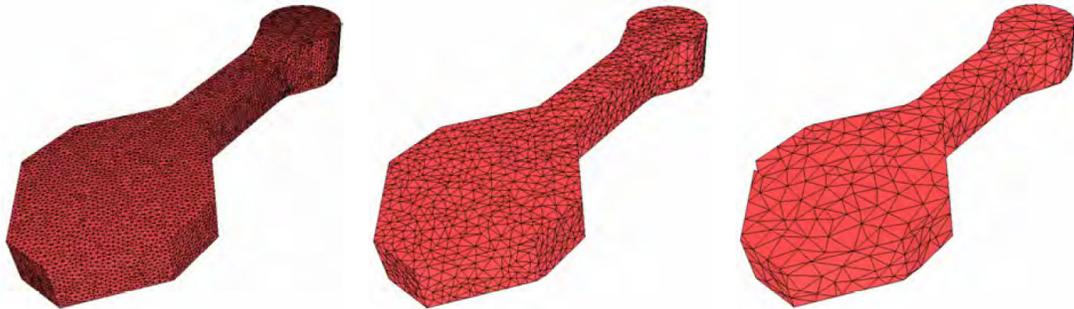


FIGURE 1.5 – Différents niveaux de maillages emboîtés par noeuds d'une préforme pour la simulation du forgeage d'une bielle en début de procédé (Rey, 2007)

Cette approche permet de partir de la grille fine et de procéder par déraffinement tout

en contrôlant relativement bien la qualité des éléments construits. D'une manière similaire à l'approche emboîtée par élément, le grand avantage de cette méthode vient de la simplicité des opérateurs de transfert et de leur précision.

Cependant, la contrainte de déraffiner en conservant les noeuds du maillage fin ne permet pas toujours de générer un maillage de qualité, notamment sur des géométries complexes. Des difficultés à respecter la géométrie initiale ou au contraire l'impossibilité de déraffiner suffisamment ont notamment été reconnus et observés comme limitant.

### 1.3.3 Grilles indépendantes

Afin de mailler avec précision la géométrie du domaine à tous les niveaux, l'utilisation de maillages indépendants s'impose, même si elle nécessite la construction d'opérateurs plus complexes. Dans ([Feng \*et al.\*, 1997](#)), les auteurs développent une méthode multigrille basée sur des maillages non structurés et indépendants pour la résolution de problèmes linéaires et non linéaires de la mécanique du solide. Celle-ci fournit des résultats très encourageants sur diverses applications en 2D et 3D. Par exemple, avec les maillages indépendants présentés sur la figure 1.6, la résolution du problème élastique comportant 81 500 degrés de liberté par la méthode multigrille permet une accélération d'un facteur 5,5 par rapport au solveur itératif par gradient conjugué.

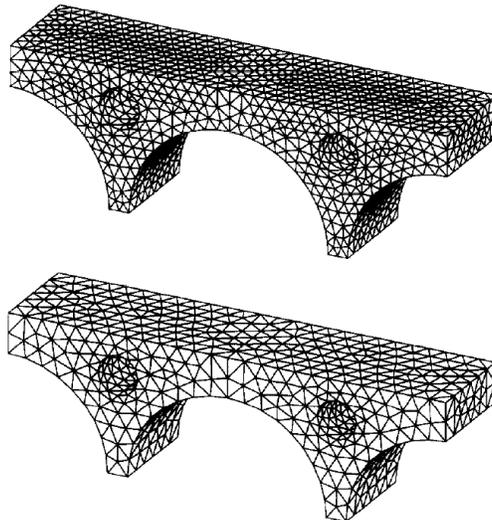


FIGURE 1.6 – Exemple de maillages indépendants générés dans ([Feng \*et al.\*, 1997](#))

(Adams, 2002) met toutefois en évidence les inconvénients des multigrilles basées sur des maillages indépendants :

- Même si les cycles de résolution apparaissent comme très efficaces étant donnée la bonne représentation géométrique des différents niveaux, cette méthode fait face à d'importants coûts de construction. En effet, l'utilisation de maillages indépendants complexifie la construction des opérateurs de transfert, généralement basés sur l'interpolation des noeuds du maillage fin sur les éléments du maillage grossier pour l'opérateur de restriction, puis des noeuds grossiers sur le maillage fin pour l'opérateur de prolongement (figure 1.7). Pour les problèmes non linéaires ou incrémentaux, cet inconvénient est néanmoins atténué puisque ces opérateurs servent à la résolution de nombreux systèmes linéaires.
- Des problèmes de robustesse peuvent également être induits par l'algorithme de déraffinement de maillage. Dans l'exemple utilisé par (Adams, 2002), l'algorithme de Delaunay permettant de générer les niveaux grossiers engendre des maillages trop grossiers qui ne permettent pas la convergence de l'algorithme multigrille. Ce problème met en évidence la nécessité d'un contrôle sur la conservation du volume du domaine ou sur un nombre de noeuds minimal à respecter.

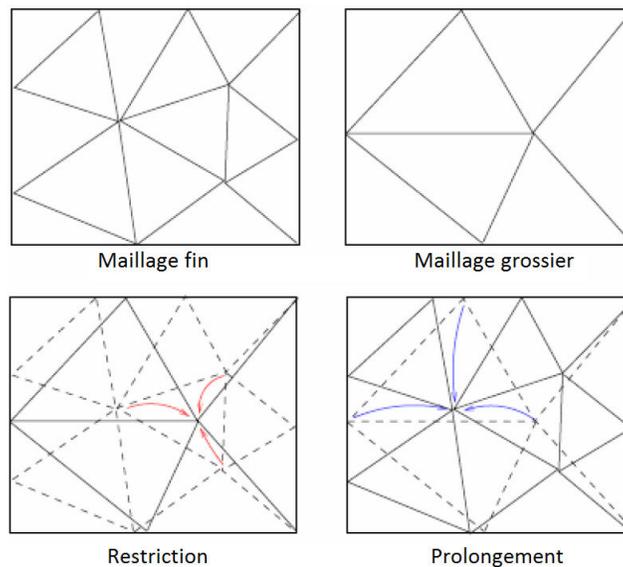


FIGURE 1.7 – Construction des opérateurs de transfert pour des maillages indépendants

## 1.4 Bilan et approche retenue

Nous avons vu dans cette partie les caractéristiques des deux grandes familles de multigrilles : d’une part, les multigrilles algébriques, dont la caractéristique est de proposer une approche totalement indépendante du maillage et ne se reposant que sur la matrice de calcul. Cela donne une méthode facile d’utilisation pouvant s’appliquer à de nombreux problèmes avec une certaine robustesse. Cependant, pour des problèmes où des multigrilles géométriques peuvent être utilisées, l’approche algébrique s’avère moins efficace.

Les multigrilles géométriques sont quant à elles plus difficiles à mettre en oeuvre, notamment à cause de la difficulté à générer des maillages grossiers, et différentes approches ont été décrites à cette fin. Cette thèse visant des géométries complexes définies sur la grille fine, la méthode de génération par emboîtement d’éléments à partir de la grille grossière n’est donc pas envisageable ici. L’approche emboîté par nœuds de (Rey, 2007) présente elle aussi des difficultés à être appliquée à des telles géométries, cette contrainte sur les nœuds étant trop forte.

L’objectif de cette thèse étant de développer une méthode multigrille efficace, en séquentiel et en parallèle, pour la résolution de problèmes de mécanique du solide et adaptée à des géométries complexes et évolutives, nous nous orientons vers une méthode multigrille hybride, semblable à celle des travaux de (Feng *et al.*, 1997) que nous enrichissons de nouveautés algorithmiques permettant de bénéficier au mieux du calcul parallèle, devenu indispensable aujourd’hui. Dans ce domaine, les récents travaux de (Ramadan, 2010) nous apportent une méthode de déraffinement parallèle de maillage et de construction des opérateurs de transfert correspondants parfaitement adaptée à la génération de maillages grossiers. Associée à cette approche géométrique de construction des niveaux grossiers, la méthode algébrique sera utilisée pour construire les systèmes linéaires grossiers afin de conserver au mieux les informations contenues dans la matrice assemblée sur la grille fine. Enfin, l’utilisation de bibliothèques de calcul parallélisées permettent aujourd’hui de bénéficier aisément et efficacement de performances parallèles, y compris pour les méthodes directes de résolution (Amestoy *et al.*, 2001), (Amestoy *et al.*, 2006).

Avant de détailler cette méthode multigrille hybride, nous nous intéressons dans le chapitre suivant à la présentation des équations à résoudre en mise en forme des matériaux et aux spécificités du logiciel FORGE® NxT, dans lequel les développements sont intégrés.



## 2 | FORGE®

If you can't explain it to a six year old, you don't understand it yourself

---

Albert Einstein

Nous proposons dans cette partie une brève description des équations du problème mécanique retenues pour modéliser la mise en forme de métaux, la formulation éléments finis utilisée, les méthodes de résolution sélectionnées et les autres spécificités du logiciel FORGE® NxT pour prendre en compte le remaillage et autres particularités des grandes déformations. La partie thermique n'est pas abordée. Le lecteur intéressé pourra se reporter à ([Kpodzo, 2014](#)) et ([Ramadan, 2010](#)) pour la modélisation et la résolution du problème thermique ainsi que du couplage thermo-mécanique pendant la mise en forme des matériaux.

Le matériau est considéré comme continu, homogène et mis en forme à chaud, les déformations élastiques peuvent ainsi être négligées et la matière est considérée comme incompressible. Nous considérons également le matériau comme étant rigide viscoplastique, qui est le plus caractéristique et qui représente parfaitement les difficultés rencontrées lors de la résolution des principaux problèmes de mise en forme.

Dans un second temps, nous présentons la stratégie adoptée pour le calcul parallèle et plus particulièrement la gestion du remaillage en parallèle. Nous terminons finalement le chapitre par une présentation des simulations de mise en forme qui seront utilisées dans les prochains chapitres pour mettre en place et évaluer la méthode multigrille.

---

## Sommaire

---

<b>2.1</b>	<b>Modélisation du problème mécanique</b>	<b>23</b>
2.1.1	Equations	23
2.1.2	Eléments Finis Mixtes	26
2.1.3	Discrétisation	27
<b>2.2</b>	<b>Résolution numérique du problème mécanique</b>	<b>31</b>
2.2.1	Système à résoudre	31
2.2.2	Méthode de résolution	33
<b>2.3</b>	<b>Génération de maillage et remaillage</b>	<b>35</b>
2.3.1	Mailleur topologique MTC	35
<b>2.4</b>	<b>Calcul Parallèle</b>	<b>38</b>
2.4.1	Stratégie SPMD	38
2.4.2	Partitionnement et remaillage	39
<b>2.5</b>	<b>Simulations de référence</b>	<b>41</b>
2.5.1	Ecrasement entre tas plats	41
2.5.2	Triaxe frottant avec repli	42
2.5.3	Laminage retour	43
2.5.4	Martelage rotatif	45

---

## 2.1 Modélisation du problème mécanique

La pièce à forger occupe un domaine  $\Omega$  de frontière  $\partial\Omega$ . Nous cherchons à exprimer le problème mécanique en fonction du champ de vitesse  $v$  et de pression  $p$  en tout point de  $\Omega$ . On exprime ici les équations qui régissent les déformations de ce domaine sous l'effet des sollicitations mécaniques imposées par les outils pendant le forgeage.

### 2.1.1 Equations

#### Equations générales

L'équilibre mécanique du domaine  $\Omega$  à chaque instant  $t$  est donné par l'équation suivante :

$$\operatorname{div}(\sigma) + \rho g = \rho \gamma \quad (2.1)$$

$\sigma$  représentant le tenseur des contraintes de Cauchy,  $\rho$  la masse volumique du matériau,  $g$  la pesanteur et  $\gamma$  le vecteur accélération.

Pour un grand nombre de procédés de mise en forme, les forces de gravité et d'inertie sont considérées comme négligeables. L'équation d'équilibre mécanique se résume donc à la forme suivante :

$$\operatorname{div}(\sigma) = 0 \quad (2.2)$$

L'équation de conservation de la masse du matériau s'écrit sous la forme :

$$\frac{d\rho}{dt} + \operatorname{div}(\rho v) = 0, \quad \text{sur } \Omega \quad (2.3)$$

La dérivée de la masse volumique par rapport au temps est considérée comme nulle, en faisant l'hypothèse d'incompressibilité et donc en négligeant les déformations élastiques devant les déformations plastiques. Nous pouvons réécrire 2.3 pour un matériau homogène :

$$\operatorname{div}(v) = 0 \quad (2.4)$$

On considérera le système à résoudre reprenant les équations 2.4 et 2.2 :

$$\begin{cases} \operatorname{div}(v) = 0 \\ \operatorname{div}(\sigma) = 0 \end{cases} \quad (2.5)$$

### Conditions aux limites

Dans la plupart des cas de forgeage, la frontière  $\partial\Omega$  du lopin peut se décomposer en deux parties distinctes :

$$\partial\Omega = \partial\Omega_l \cup \partial\Omega_c$$

- $\partial\Omega_l$  représente la frontière libre. La pièce ne subit ici aucune contrainte :

$$\sigma n = 0 \tag{2.6}$$

avec  $n$  la normale sortante au domaine.

- $\partial\Omega_c$  représente la frontière en contact avec les outils. Deux types de conditions aux limites sont imposées sur cette surface :

- Une condition de non pénétration des nœuds dans l'outil, décrite généralement par les équations de Signorini

$$\begin{cases} (u - u^{outil}) \cdot n \leq 0 \\ \sigma_n \leq 0 \\ [(u - u^{outil}) \cdot n] \sigma_n = 0 \end{cases} \tag{2.7}$$

avec  $u^{outil}$  le déplacement de l'outil,  $u$  celui de la pièce et  $\sigma_n = (\sigma \cdot n) \cdot n$  la pression de contact.

- Une condition de frottement entre la pièce et les outils, permettant d'exprimer le taux de scission de frottement  $\tau$

$$\tau = \sigma \cdot n - (\sigma_n) \cdot n \tag{2.8}$$

La loi associée à cette condition sera introduite dans la sous-section 2.1.1.

### Loi de comportement

Le comportement du matériau lors d'un procédé de forgeage peut suivre différentes lois de comportement, par exemple élasto-plastique ou élasto-viscoplastique. Nous nous plaçons ici dans des conditions de mise en forme à chaud ; la partie élastique de la déformation est négligée et une loi viscoplastique est généralement utilisée. Celle de Norton-Hoff, reliant le déviateur des contraintes  $s$  au tenseur des vitesses de déformation  $\dot{\varepsilon}(v)$ , sera considérée :

$$s = 2K(\sqrt{3}\dot{\varepsilon}(v))^{m-1}\dot{\varepsilon}(v) \tag{2.9}$$

Dans cette expression non linéaire,  $K$  représente la consistance du matériau, qui peut être thermodépendante et écrouissable,  $m$  la sensibilité à la vitesse de déformation ( $0 < m \leq 1$ ,

$m = 1$  permettant d'obtenir un comportement Newtonien) et  $\dot{\varepsilon}$  le tenseur des vitesses de déformation, défini sous l'hypothèse des petites déformations par :

$$\dot{\varepsilon}(v) = \frac{1}{2}(\text{grad}(v) + \text{grad}(v)^T) \quad (2.10)$$

La vitesse de déformation généralisée  $\dot{\varepsilon}$  s'écrit alors :

$$\dot{\varepsilon}(v) = \sqrt{\frac{2}{3}\dot{\varepsilon}(v) : \dot{\varepsilon}(v)} \quad (2.11)$$

Le déviateur des contraintes  $s$  est défini par :

$$s = \sigma + pI \quad (2.12)$$

avec  $I$  le tenseur identité et  $p$  la pression hydrostatique définie par :

$$p = -\frac{1}{3}\text{Tr}(\sigma) \quad (2.13)$$

L'équation 2.2 se réécrit :

$$\text{div}(s) - \text{grad}(p) = 0 \quad (2.14)$$

### Loi de frottement

L'action du frottement à l'interface de contact entre le lopin et les outils suit ici une loi de Coulomb limitée par la loi de Tresca. Dans la loi de Tresca, la scission de frottement  $\tau$  s'exprime en fonction de la contrainte d'écoulement  $\sigma_0$  :

$$\tau = -\bar{m} \frac{\sigma_0}{\sqrt{3}} \frac{\Delta v_g}{\|\Delta v_g\|} \quad (2.15)$$

$\bar{m}$  ( $0 \leq \bar{m} \leq 1$ ) correspond au coefficient de Tresca et  $\Delta v_g$  à la vitesse de glissement relative de l'outil par rapport à la pièce :

$$\Delta v_g = (v - v^{outil}) - (v - v^{outil}) \cdot n$$

avec  $v^{outil}$  la vitesse de l'outil.

La loi de Coulomb relie la scission de frottement  $\tau$  à la pression de contact  $\sigma_n$  :

$$\tau = -\mu \sigma_n \frac{\Delta v_g}{\|\Delta v_g\|} \quad (2.16)$$

$\mu$  étant le coefficient de Coulomb.

En combinant les équations 2.15 et 2.16, la loi de Coulomb limitée par la loi de Tresca utilisée ici s'écrit :

$$\tau = \begin{cases} -\mu\sigma_n \frac{\Delta v_g}{\|\Delta v_g\|} & \text{si } |\mu\sigma_n| < \bar{m} \frac{\sigma_0}{\sqrt{3}} \\ -\bar{m} \frac{\sigma_0}{\sqrt{3}} \frac{\Delta v_g}{\|\Delta v_g\|} & \text{si } |\mu\sigma_n| \geq \bar{m} \frac{\sigma_0}{\sqrt{3}} \end{cases} \quad (2.17)$$

## 2.1.2 Eléments Finis Mixtes

### Formulation du problème

En reprenant le système obtenu à l'équation 2.5 et les conditions aux limites données à l'équation 2.7 associées au frottement de l'équation 2.17, la formulation forte de notre problème donne le système suivant :

$$\left\{ \begin{array}{ll} \operatorname{div}(v) = 0 & \text{sur } \Omega \\ \operatorname{div}(\sigma) = 0 & \text{sur } \Omega \\ \tau = \begin{cases} -\mu\sigma_n \frac{\Delta v_g}{\|\Delta v_g\|} & \text{si } |\mu\sigma_n| < \bar{m} \frac{\sigma_0}{\sqrt{3}} \\ -\bar{m} \frac{\sigma_0}{\sqrt{3}} \frac{\Delta v_g}{\|\Delta v_g\|} & \text{si } |\mu\sigma_n| \geq \bar{m} \frac{\sigma_0}{\sqrt{3}} \end{cases} & \text{sur } \partial\Omega_c \\ (u - u^{outil}) \cdot n \leq 0 & \text{sur } \partial\Omega_c \\ \sigma_n \leq 0 & \text{sur } \partial\Omega_c \\ [(u - u^{outil}) \cdot n] \sigma_n = 0 & \text{sur } \partial\Omega_c \\ \sigma n = 0 & \text{sur } \partial\Omega_l \end{array} \right. \quad (2.18)$$

### Formulation faible

En décomposant  $\sigma$  en sa partie déviatorique et sa partie sphérique puis en multipliant les équations du 2.18 par des fonctions tests et en intégrant sur le domaine, nous obtenons la formulation faible mixte en vitesse pression  $(v, p)$  :

$$\begin{cases} \forall v^* \in V_0^{ca} & \int_{\Omega} s : \dot{\varepsilon}(v^*) d\omega - \int_{\Omega} p \operatorname{div}(v^*) d\omega - \int_{\partial\Omega_c} \tau \cdot v^* ds = 0 \\ \forall p^* \in P & \int_{\Omega} p^* \operatorname{div}(v) d\omega = 0 \end{cases} \quad (2.19)$$

avec  $(v, p) \in (V^{ca}, P)$  et  $(v^*, p^*) \in (V_0^{ca}, P)$ , ces espaces étant définis par :

$$\begin{aligned} V^{ca} &= \{v \in (H^1(\Omega))^3 \text{ telle que } (u - u^{outil}) \cdot n \leq 0 \text{ sur } \partial\Omega_c\} \\ V_0^{ca} &= \{v \in (H^1(\Omega))^3 \text{ telle que } (u \cdot n) \leq 0 \text{ sur } \partial\Omega_c\} \\ P &= L^2(\Omega) \end{aligned} \quad (2.20)$$

### 2.1.3 Discrétisation

#### Discrétisation spatiale

Pour la discrétisation du problème, la méthode des éléments finis est utilisée. Les simulations de mise en forme traitant des grandes déformations, des remaillages fréquents sont nécessaires pour mener la simulation à son terme. L'élément tétraédrique est donc choisi pour sa flexibilité vis à vis du maillage et du remaillage des géométries complexes (Coupez, 1994). Introduit par (Arnold *et al.*, 1984), l'élément P1+/P1 est utilisé ici pour sa compatibilité avec la formulation mixte et parce qu'il permet de vérifier la condition de Brezzi-Babuska (Babuška, 1973), assurant à notre problème discret l'existence et l'unicité de la solution  $(v_h, p_h) \in (V_h \times P_h)$ .

Le champ de vitesse est ainsi approximé par une interpolation linéaire à partir des sommets du tétraèdre et enrichi par une fonction bulle linéaire par morceaux sur les quatre sous-tétraèdres formés par les sommets et le centre de gravité du tétraèdre (figure 2.1a). La pression est interpolée linéairement (figure 2.1b).



FIGURE 2.1 – Elément tétraédrique P1+/P1

En considérant un domaine discret  $\Omega_h$  formé de tétraèdres  $\omega_h$  formant un maillage  $T_h$  et

$\partial\Omega_h$  sa surface, on peut écrire les espaces d'approximations :

$$\left\{ \begin{array}{l} P_h = \{p_h \in C^0(\Omega_h) \quad \text{tel que } \forall \omega_h \in T_h, p_h/\omega_h \in P^1(\omega_h)\} \\ V_h = L_h \oplus B_h \\ L_h = \{v_h \in (C^0(\Omega_h))^3 \quad \text{tel que } \forall \omega_h \in T_h, v_h/\omega_h \in (P^1(\omega_h))^3\} \\ B_h = \{b_h \in (C^0(\Omega_h))^3 \quad \text{tel que } \forall \omega_h \in T_h, b_h/\omega_h^k \in (P^1(\omega_h^k))^3 \text{ et } b_h = 0 \text{ sur } \partial\Omega_h\} \end{array} \right. \quad (2.21)$$

Les  $\omega_h^k$ , avec  $k = 1, \dots, 4$  sont les sous-tétraèdres du tétraèdre  $w_h$ .

Les champs de vitesse et de pression discrets sont donnés par :

$$\left\{ \begin{array}{l} v_h = v_h^l + v_h^b = \sum_{k=1}^{Nbn} V_k^l N_k^l + \sum_{j=1}^{Nbe} V_j^b N_j^b \\ p_h = \sum_{k=1}^{Nbn} P_k N_k^l \end{array} \right. \quad (2.22)$$

avec  $N_k^l$  et  $N_j^b$  les fonctions de base linéaires et bulle associées respectivement au nœud  $k$  et à l'élément  $j$ ,  $Nbn$  le nombre de nœuds et  $Nbe$  le nombre d'éléments du maillage.

Nous obtenons ainsi la formulation discrétisée du problème en remplaçant dans l'équation 2.19 les fonctions tests par les fonctions de base et  $(v, p)$  par  $(v_h, p_h)$  :

$$\left\{ \begin{array}{l} \forall k \in \{1, \dots, Nbn\} \quad ; \quad \forall i \in \{1, \dots, 3\} : \\ \int_{\Omega_h} 2K(\sqrt{3}\dot{\varepsilon}(v_h^l))^{m-1} \dot{\varepsilon}(v_h^l + v_h^b) : \dot{\varepsilon}(N_k^l e_i) d\Omega_h \\ \quad - \int_{\Omega_h} p_h \operatorname{div}(N_k^l e_i) d\Omega_h + \int_{\partial\Omega_{ch}} \mu \sigma_n \frac{\Delta v_{gh} \cdot e_i}{\|\Delta v_{gh}\|} N_k^l ds = 0 \\ \forall j \in \{1, \dots, Nbe\} \quad ; \quad \forall i \in \{1, \dots, 3\} : \\ \int_{\Omega_h^e} 2K(\sqrt{3}\dot{\varepsilon}(v_h^l))^{m-1} \dot{\varepsilon}(v_h^l + v_h^b) : \dot{\varepsilon}(N_j^b e_i) d\Omega_h - \int_{\Omega_h} p_h \operatorname{div}(N_j^b e_i) d\Omega_h = 0 \\ \forall k \in \{1, \dots, Nbn\} : \\ \int_{\Omega_h} N_k^l \operatorname{div}(v_h^l + v_h^b) d\Omega_h = 0 \end{array} \right. \quad (2.23)$$

avec  $e_i$  le vecteur unitaire dans la direction  $i$  de l'espace.

### Discrétisation temporelle

Soit  $t^0$  et  $t^{fin}$  le temps initial et final de la simulation. L'intervalle  $[t^0, t^{fin}]$  représente ainsi le temps total de la simulation et est divisé en  $N$  incréments de temps définis par des pas de

temps  $\Delta t^i$  :

$$[t^0, t^{fin}] = \bigcup_{i=0, N-1} [t^i, t^i + \Delta t^i] \quad (2.24)$$

A l'instant  $t$ , la configuration  $\Omega^t$  connue permet de calculer la solution  $(v^t, p^t)$  vérifiant l'équilibre à cet instant. L'évolution du domaine suit une formulation de type lagrangien réactualisé. La configuration à l'instant  $t + \Delta t$  est ainsi calculée à partir de la configuration  $\Omega^t$ .

Considérant  $x(t)$  comme la position à l'instant  $t$ , l'équation d'évolution du domaine est donnée par :

$$\frac{dx(t)}{dt} = v \quad (2.25)$$

Le schéma utilisé pour intégrer cette équation est un schéma d'Euler implicite :

$$\forall t, \quad x_h^{t^i + \Delta t^i} = x_h^{t^i} + v_h^{t^i + \Delta t^i} \Delta t^i \quad (2.26)$$

Notons que pour les procédés engendrant des variations de volume importantes, des schémas d'intégration d'ordre supérieur sont utilisés ([Traoré, 2001](#)).

### Gestion du contact

La gestion du contact est présentée en détail dans ([Mocellin, 1999](#)). Ici, une brève description est faite où nous considérons les outils comme non déformables.

La distance signée  $\delta$  d'un point  $M$  du domaine  $\Omega$  à l'outil, décrite sur la figure 2.2, est définie par :

- $\delta(M) > 0$  si le point  $M$  est à l'extérieur de l'outil
- $\delta(M) = 0$  si le point  $M$  est sur la surface de l'outil
- $\delta(M) < 0$  si le point  $M$  a pénétré l'outil

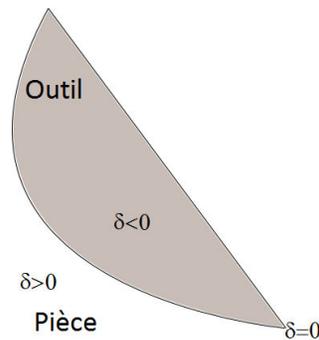


FIGURE 2.2 – Définition de la distance signée  $\delta$

Les conditions de contact sont imposées aux nœuds de la surface, utilisant la formulation décrite par (Fourment *et al.*, 1999). La condition de non pénétration entre la pièce et l'outil s'écrit alors :

$$\forall M_k \in \partial\Omega_h, \quad \delta(M_k) \geq 0 \quad (2.27)$$

On impose la condition de non pénétration du nœud dans l'outil à la fin de chaque incrément de temps :

$$\forall M_k \in \partial\Omega_h, \quad 0 \leq \delta(M_k^{t+\Delta t}) \quad (2.28)$$

Un développement limité d'ordre 1 en temps et l'approximation de la surface de l'outil par son plan tangent permet d'exprimer cette distance en fonction de la distance du nœud à la surface en début d'incrément et de sa vitesse au cours de l'incrément :

$$\forall M_k \in \partial\Omega_h, \quad \delta(M_k^{t+\Delta t}) \approx \delta(M_k^t) + (v^{outil} - v_k) \cdot n_k^t \Delta t \quad (2.29)$$

avec  $n_k^t$  la normale sortante à l'outil au nœud  $k$  à l'instant  $t$ . La condition de contact peut donc s'exprimer ainsi en chaque nœud du maillage :

$$\forall M_k \in \partial\Omega_h, \quad h(v_k) = (v_k - v^{outil}) \cdot n_k - \frac{\delta(M_k)}{\Delta t} \leq 0 \quad (2.30)$$

En introduisant le multiplicateur de Lagrange  $\lambda_k \leq 0$  associé à l'inégalité de contact 2.30 et homogène à la contrainte normale de contact, la première équation du problème discrétisé 2.23 se réécrit :

$$\left\{ \begin{array}{l} \forall k \in \{1, \dots, Nbn\} \quad ; \quad \forall i \in \{1, \dots, 3\} : \\ \int_{\Omega_h} 2K(\sqrt{3}\dot{\varepsilon}(v_h^l))^{m-1} \dot{\varepsilon}(v_h^l + v_h^b) : \dot{\varepsilon}(N_k^l e_i) d\Omega_h - \int_{\Omega_h} p_h \operatorname{div}(N_k^l e_i) d\Omega_h \\ \quad + \int_{\partial\Omega_{ch}} \mu \sigma_n \frac{\Delta v_{gh} \cdot e_i}{\|\Delta v_{gh}\|} N_k^l ds - \lambda_k (n_k \cdot e_i) S_k \mathbf{1}_{\partial\Omega_{ch}}(X_k) = 0 \\ \forall M_k \in \partial\Omega_{ch} : \\ h(v_k) \leq 0 \end{array} \right. \quad (2.31)$$

avec  $\partial\Omega_{ch}$  la frontière discrétisée en contact avec les outils et  $\mathbf{1}_{\partial\Omega_{ch}}$  la fonction indicatrice :

$$\forall X_k \in \partial\Omega_{ch}, \quad \mathbf{1}_{\partial\Omega_{ch}}(X_k) = \begin{cases} 1 & \text{si } X_k \text{ est un nœud de } \partial\Omega_h \\ 0 & \text{sinon} \end{cases} \quad (2.32)$$

et  $S_k$  la surface associée à un nœud  $k$ , permettant d'affecter à ce nœud un poids proportionnel à la surface des faces frontières adjacentes :

$$S_k = \int_{\partial\Omega_{ch}} N_k dS \quad (2.33)$$

L'utilisation d'une méthode de pénalisation, décrite par (Chenot *et al.*, 2002), permet de gérer l'imposition de l'inéquation de contact (dernière équation de 2.31) et fait disparaître le multiplicateur de Lagrange.

On introduit la fonction partie positive :

$$[x]^+ = \max(0, x) \quad (2.34)$$

et on peut finalement écrire notre problème discretisé :

$$\left\{ \begin{array}{l} \forall k \in \{1, \dots, Nbn\} \quad ; \quad \forall i \in \{1, \dots, 3\} : \\ \int_{\Omega_h} 2K(\sqrt{3}\dot{\varepsilon}(v_h^l))^{m-1} \dot{\varepsilon}(v_h^l + v_h^b) : \dot{\varepsilon}(N_k^l e_i) d\Omega_h - \int_{\Omega_h} p_h \operatorname{div}(N_k^l e_i) d\Omega_h \\ \quad + \int_{\partial\Omega_{ch}} \mu \sigma_n \frac{\Delta v_{gh} e_i}{\|\Delta v_{gh}\|} N_k^l ds + \rho_c \mathbf{1}_{\partial\Omega_{ch}}(X_k) [h(v_k)]^+ S_k n_k \cdot e_i = 0 \\ \forall j \in \{1, \dots, Nbe\} \quad ; \quad \forall i \in \{1, \dots, 3\} : \\ \int_{\Omega_h} 2K(\sqrt{3}\dot{\varepsilon}(v_h^l))^{m-1} \dot{\varepsilon}(v_h^l + v_h^b) : \dot{\varepsilon}(N_j^b e_i) d\Omega_h - \int_{\Omega_h} p_h \operatorname{div}(N_j^b e_i) d\Omega_h = 0 \\ \forall k \in \{1, \dots, Nbn\} : \\ \int_{\Omega_h} N_k^l \operatorname{div}(v_h^l + v_h^b) d\Omega_h = 0 \end{array} \right. \quad (2.35)$$

avec  $\rho_c$  le coefficient de pénalisation.

## 2.2 Résolution numérique du problème mécanique

### 2.2.1 Système à résoudre

Les équations 2.35 du problème discret forment un système d'équations non linéaires :

$$R(v^l, v^b, p) = \begin{cases} R_l(v^l, v^b, p) = 0 \\ R_b(v^l, v^b, p) = 0 \\ R_p(v^l, v^b) = 0 \end{cases} \quad (2.36)$$

La méthode de Newton-Raphson permet de résoudre ce système : partant d'une approximation initiale  $(v_0^l, v_0^b, p_0)$  de la solution, les corrections successives  $(\Delta v^l, \Delta v^b, \Delta p)$  sont calculées de façon à itérativement s'approcher de l'annulation du résidu  $R$ .

L'expression du résidu à l'itération  $k$  de Newton-Raphson s'écrivant  $R(v_k^l, v_k^b, p_k)$ , on cherche à obtenir à l'itération  $k+1$  :

$$R(v_k^l + \Delta v^l, v_k^b + \Delta v^b, p_k + \Delta p) = 0 \quad (2.37)$$

Un développement de Taylor permet d'écrire :

$$R(v_k^l + \Delta v^l, v_k^b + \Delta v^b, p_k + \Delta p) \approx R(v_k^l, v_k^b, p_k) + \frac{\partial R}{\partial v^l} \Delta v^l + \frac{\partial R}{\partial v^b} \Delta v^b + \frac{\partial R}{\partial p} \Delta p \quad (2.38)$$

Le système linéaire à résoudre s'écrit alors :

$$\begin{pmatrix} \frac{\partial R_l}{\partial v^l} & \frac{\partial R_l}{\partial v^b} & \frac{\partial R_l}{\partial p} \\ \frac{\partial R_b}{\partial v^l} & \frac{\partial R_b}{\partial v^b} & \frac{\partial R_b}{\partial p} \\ \frac{\partial R_p}{\partial v^l} & \frac{\partial R_p}{\partial v^b} & \frac{\partial R_p}{\partial p} \end{pmatrix} \begin{pmatrix} \Delta v^l \\ \Delta v^b \\ \Delta p \end{pmatrix} = - \begin{pmatrix} R_l(v^l, v^b, p) \\ R_b(v^l, v^b, p) \\ R_p(v^l, v^b, p) \end{pmatrix} \quad (2.39)$$

$R_p$  ne dépendant pas de  $p$ , le terme  $\frac{\partial R_p}{\partial p}$  est nul. D'autre part, les propriétés de la bulle décrites par exemple dans (Perchat, 2000) permettent de montrer que  $\frac{\partial R^l}{\partial v^b} = \frac{\partial R^b}{\partial v^l} = 0$  dans le cas linéaire et d'étendre cette propriété au cas non linéaire. Le système linéaire devient alors :

$$\begin{pmatrix} \frac{\partial R_l}{\partial v^l} & 0 & \frac{\partial R_l}{\partial p} \\ 0 & \frac{\partial R_b}{\partial v^b} & \frac{\partial R_b}{\partial p} \\ \frac{\partial R_p}{\partial v^l} & \frac{\partial R_p}{\partial v^b} & 0 \end{pmatrix} \begin{pmatrix} \Delta v^l \\ \Delta v^b \\ \Delta p \end{pmatrix} = - \begin{pmatrix} R_l(v^l, v^b, p) \\ R_b(v^l, v^b, p) \\ R_p(v^l, v^b, p) \end{pmatrix} \quad (2.40)$$

Une technique de condensation permet ensuite d'éliminer les termes bulle et de réduire le système à l'expression suivante :

$$\begin{pmatrix} H^{vv} & H^{vp} \\ H^{pv} & -\beta H^{pp} \end{pmatrix} \begin{pmatrix} \Delta v \\ \Delta p \end{pmatrix} = - \begin{pmatrix} R_v^* \\ R_p^* \end{pmatrix} \quad (2.41)$$

avec  $\beta$  un terme de stabilisation et les autres termes suivants :

$$\left\{ \begin{array}{l} H^{vv} = \frac{\partial R_l}{\partial v^l} \\ H^{vp} = \frac{\partial R_l}{\partial p} \\ H^{pv} = \frac{\partial R_p}{\partial v^l} \\ H^{pp} = \frac{\partial R_p}{\partial v^b} \left( \frac{\partial R_b}{\partial v^b} \right)^{-1} \frac{\partial R_b}{\partial p} \\ R_v^* = R_l \\ R_p^* = R_p - \frac{\partial R_p}{\partial v^b} \left( \frac{\partial R_b}{\partial v^b} \right)^{-1} R_b \\ \Delta v = \Delta v^l \end{array} \right. \quad (2.42)$$

Ce système présente ainsi 4 inconnues par nœud du maillage éléments finis, 3 composantes pour la vitesse et 1 composante pour la pression.

L'élimination de la bulle permet d'augmenter la stabilité du système, grâce à la suppression des termes nuls présents initialement sur la diagonale et menant à un meilleur conditionnement de la matrice.

Remarquons que la matrice  $H^{pp}$  est symétrique définie positive : cela implique que la matrice du système 2.41 est symétrique et indéfinie, ce qui impose le choix d'une méthode de résolution adaptée.

### 2.2.2 Méthode de résolution

Pour nos cas d'étude, le système à résoudre à chaque itération de Newton-Raphson est celui donné en 2.41, notons le  $Ax = b$ . Considérons  $J$ , la fonctionnelle définie par :

$$J(x) = \frac{1}{2}(Ax, x) - (b, x) \quad (2.43)$$

Cette fonctionnelle n'étant pas strictement convexe, l'existence et l'unicité de la solution du problème de minimisation ne sont pas assurées. La méthode de résidu conjugué classiquement utilisée consiste à minimiser la norme du résidu  $r = b - Ax$ . On considère alors  $E$ , la fonctionnelle définie par :

$$\begin{aligned} E(r) &= (r, r) \\ &= (b - Ax, b - Ax) \\ &= (A^2x, x) - 2(Ab, x) + (b, b) \end{aligned} \quad (2.44)$$

Cela revient à résoudre  $A^2x = Ab$ , avec  $A^2$  une matrice symétrique définie positive, assurant l'existence et unicité de la solution du système à résoudre.

Le résidu conjugué préconditionné (PCR) consiste à introduire une matrice de préconditionnement  $M$ , transformant la fonctionnelle à résoudre 2.44 en :

$$E_M(r) = (M^{-1}r, r) \quad (2.45)$$

L'ajout de ce préconditionneur  $M$  doit permettre à la matrice  $M^{-1}A$  d'avoir un conditionnement proche de 1 afin d'accélérer la convergence. L'algorithme 2 présente l'algorithme du résidu conjugué préconditionné.

**Données** :  $Ax = b$  à résoudre ;  $M$  un préconditionneur

**Entrées** :  $x_0$  solution initiale

$r_0 = M^{-1}(b - Ax_0)$ ;

$p_0 = r_0$ ;

**tant que** *Non convergence* **faire**

$$\alpha_k = \frac{r_k^T Ar_k}{(Ap_k)^T M^{-1} Ap_k};$$

$$x_{k+1} = x_k + \alpha_k p_k ;$$

$$r_{k+1} = r_k - \alpha_k M^{-1} Ap_k;$$

$$\beta_k = \frac{r_{k+1}^T Ar_{k+1}}{r_k^T Ar_k};$$

$$p_{k+1} = r_{k+1} + \beta_k p_k ;$$

$$Ap_{k+1} = Ar_{k+1} + \beta_k Ap_k ;$$

$$k = k + 1$$

**fin**

**Algorithme 2** : Méthode du résidu conjugué préconditionné

Une étude détaillée des performances des différents types de préconditionneurs peut être trouvée dans (Marie, 1997), (Mocellin *et al.*, 2001) et (Perchat, 2000). Nous nous intéressons ici au préconditionneur utilisé comme référence et étant le plus performant dans le logiciel FORGE® : une factorisation incomplète de type *ILU*.

Considérons la décomposition *LU* d'une matrice  $A$ , avec  $L$  une matrice triangulaire inférieure et  $U$  une matrice triangulaire supérieure :

$$A = LU \tag{2.46}$$

La factorisation *ILU* à 0 niveau de remplissage (notée *ILU(0)*) consiste en le calcul d'une décomposition *LU* approchée de  $A$  telle que :

$$\left\{ \begin{array}{l} A \approx \tilde{L}\tilde{U} \\ \tilde{L}_{ij} = \begin{cases} 0 & \text{si } A_{ij} = 0 \\ \tilde{l}_{ij} & \text{sinon} \end{cases} \\ \tilde{U}_{ij} = \begin{cases} 0 & \text{si } A_{ij} = 0 \\ \tilde{u}_{ij} & \text{sinon} \end{cases} \end{array} \right. \tag{2.47}$$

Cette méthode permet de garder la structure creuse de la matrice initiale  $A$  et d'obtenir une factorisation plus rapidement qu'une décomposition *LU* complète. Les factorisations à  $k$

niveaux de remplissage (notées  $ILLU(k)$ ) ajoutent  $k$  voisins aux termes non-nuls de  $\tilde{L}$  et  $\tilde{U}$ , augmentant le niveau de remplissage et ainsi l'efficacité du préconditionneur, mais également son coût de construction et d'utilisation. On remarque qu'une factorisation  $ILLU(\infty)$  correspond à la décomposition  $LU$  initiale.

Le préconditionneur classiquement utilisé et considéré comme le plus performant pour les applications de mise en forme traitées par FORGE® NxT est  $ILLU(1)$ .

Le solveur linéaire utilisé est donc un résidu conjugué préconditionné par une factorisation  $ILLU(1)$  (noté dans la suite PCR+ILU1). L'étude des temps de calcul d'une simulation représentative d'un problème de mise en forme permet d'évaluer le temps dédié aux résolutions des systèmes linéaires comme représentant 70 à 80 % du temps global. S'intéresser à des méthodes plus rapides pour résoudre ces systèmes aura donc un impact important sur la réduction des temps de calcul, et donc sur l'accélération des simulations. Les méthodes multigrilles présentées dans le chapitre précédent peuvent répondre à ces attentes. Elles doivent cependant être correctement paramétrées pour une bonne compatibilité avec les spécificités des équations que nous traitons ici. Il faudra en particulier s'assurer de la compatibilité avec la méthode de parallélisation choisie pour le logiciel FORGE® NxT, décrite dans la suite, section 2.4. Il faudra également que le solveur s'adapte avec la procédure de remaillage nécessaire à la prise en compte des grandes déformations du domaine, dont nous décrivons la stratégie.

## 2.3 Génération de maillage et remaillage

Nous pouvons distinguer quatre principales techniques pour la génération de maillage :

- La méthode octree (Yerry *et al.*, 1983)
- La méthode frontale (Rassineux, 1995)
- La méthode Delaunay (Delaunay, 1934)
- La méthode topologique (Coupez, 1991)

Des détails sur les trois premières méthodes peuvent être trouvés dans (Frey et George, 1999) et (George, 2001). Nous nous intéressons ici à la méthode basée sur la topologie de maillage, développée par (Coupez, 2000) et utilisée dans le logiciel FORGE® NxT.

### 2.3.1 Maillageur topologique MTC

La formulation utilisée dans FORGE® NxT est basée sur une approche lagrangienne, le maillage suivant la déformation de la matière. Cependant, la pièce subissant des grandes déformations lors des procédés de mises en forme, les éléments finissent par être de moins

bonne qualité et par ne plus correctement représenter la géométrie du domaine. Une étape de remaillage s'effectue alors, permettant la génération d'un nouveau maillage adapté au domaine déformé et la continuité du calcul. Elle s'appuie sur un processus itératif d'optimisations locales.

Afin de présenter l'algorithme du mailleur topologique, nous devons introduire les notions de topologie et de critère de volume minimal.

## Topologie

Un maillage est composé d'un ensemble d'éléments. Chaque élément est lui-même défini par des nœuds numérotés et formant les sommets. La topologie du maillage correspond au tableau de connexions des éléments (qui associe chaque élément aux numéros des nœuds constituant ses sommets).

Soit  $N$  l'ensemble fini des nœuds d'un domaine  $\Omega$ , et  $T$  l'ensemble des éléments dont les sommets appartiennent à  $N$ .  $F$  est l'ensemble des faces de ces éléments, tel que :

$$F(T) = \bigcup_{e \in T} \partial\Omega_e \quad (2.48)$$

avec  $\partial\Omega_e$  la frontière du tétraèdre  $\Omega_e$  composée de ses quatre faces.

$T$  est une topologie de maillage de  $\Omega$  si et seulement si :

$$\forall f \in F(T), \quad 1 \leq \#\{e \in T \text{ tel que } f \in \partial\Omega_e\} \leq 2 \quad (2.49)$$

avec  $\#\{\cdot\}$  le nombre d'éléments de l'ensemble  $\{\cdot\}$ . Cela revient à dire que  $T$  est une topologie de maillage si chaque face de  $F$  appartient à un ou deux tétraèdres de  $T$  mais pas plus.

## Critère de volume minimal sur le maillage

Considérons  $T$  une topologie de maillage sur un ensemble de nœuds  $N$ .  $(T, N)$  est un maillage du domaine  $\Omega$  si et seulement si :

$$\sum_{e \in T} |\Omega_e| = |\Omega| \quad (2.50)$$

avec  $|\Omega_e|$  et  $|\Omega|$  respectivement les volumes de l'élément  $e$  et du domaine  $\Omega$ .

## Génération du maillage

La construction du maillage par le mailleur topologique s'effectue en 2 étapes :

1. Construction d'une topologie initiale à partir d'une discrétisation de la frontière du domaine. Cela consiste à connecter par un opérateur d'étoilement un nœud à tous les autres sauf ceux appartenant à la même face que ce dernier. Le résultat n'est pas un maillage (éléments superposés, traversant la frontière,...) mais constitue une topologie de départ
2. Amélioration progressive par balayage des nœuds et des arêtes. Pour chaque nœud et chaque arête :
  - Suppression de la topologie locale étoilée par le nœud ou l'arête (i.e l'ensemble des éléments contenant le nœud ou l'arête)
  - Remplacement de cette sous-topologie par une autre topologie obtenue par l'opérateur d'étoilement

Parmi les topologies obtenues, nous retenons celles qui minimisent le volume total  $\sum_{e \in T} |\Omega_e|$ . Plusieurs topologies peuvent avoir le même volume, nous utilisons donc un critère de qualité basé sur la géométrie pour retenir la plus satisfaisante :

$$C(e, h_e) = C_0 \frac{|\Omega_e|}{h_e^3} \quad (2.51)$$

avec  $h_e$  la longueur moyenne des arêtes du tétraèdre  $e$ ,  $C_0$  une constante permettant de normaliser le critère de forme  $C$  à 1 pour un tétraèdre équilatéral.

La valeur de  $C(e, h_e)$  est maximale pour un tétraèdre régulier et vaut 0 pour un tétraèdre plat. L'amélioration topologique passe donc par une maximisation de  $C(e, h_e)$  sous la contrainte de minimisation de volume.

### Remaillage de surface et couplage avec le volume

Un des avantages du remaillage topologique est qu'il permet de remailler simultanément sur le volume et la surface du domaine  $\Omega$  tout en respectant la géométrie. L'idée consiste à créer, dans un premier temps, une couche d'éléments virtuels connectés aux sommets du maillage surfacique et extérieurs au domaine. Pour cela, une duplication de chaque nœud frontière est réalisée afin de le relier à son double, en formant des tétraèdres virtuels qui ne contribuent pas au calcul du volume (ce qui permet de conserver le volume du domaine). La construction de cette couche d'éléments virtuels transforme les nœuds frontières en nœuds volumiques. Cela permet de déplacer ou supprimer des nœuds frontières afin d'introduire d'autres nœuds. On applique alors les opérations topologiques de volume résumées ci-dessus pour modifier le maillage surfacique (Coupez, 1991).

## 2.4 Calcul Parallèle

La taille des simulations de mise en forme en trois dimensions nécessite l'utilisation du calcul parallèle. En associant une méthode de partitionnement de domaine au solveur itératif, la parallélisation se montre efficace pour des problèmes de grande taille.

### 2.4.1 Stratégie SPMD

Le parallélisme adopté dans le logiciel FORGE® NxT suit une approche SPMD (Single Program Multiple Data) basée sur une partition du maillage. Dans (Karaseva, 2005), l'auteur montre en effet que cette approche se révèle plus efficace que la décomposition de domaine avec une méthode de type FETI, décrite par exemple par Farhat et Roux (Farhat *et al.*, 1994).

Le maillage éléments finis du problème discret est donc ici partitionné en plusieurs sous-domaines de dimensions équivalentes. Afin de gagner en efficacité, ce partitionnement est généré de façon à minimiser la taille des interfaces entre les sous-domaines ce qui permet de limiter les communications. Le code de calcul s'exécute ensuite sur chaque processeur avec les conditions limites du problème global, tout comme les conditions de contact et de frottement. La surface de contact étant une inconnue du problème, les paramètres de pilotage et la structure des outils sont dupliqués sur chaque processeur et l'analyse de contact se fait localement sur tous les processeurs.

Considérons une décomposition en  $N_{proc}$  d'un domaine  $\Omega$  de maillage  $M$  :

$$\Omega = \bigcup_{i=1}^{N_{proc}} \Omega_i$$

Chaque sous-domaine  $\Omega_i$  possède sa partition de maillage  $M_i$  telle que :

$$\begin{cases} M = \bigcup_{i=1}^{N_{proc}} M_i \\ \gamma_{ij} = M_i \cap M_j \quad \forall i \neq j \end{cases} \quad (2.52)$$

$\gamma_{ij}$  étant l'interface entre les sous domaines  $\Omega_i$  et  $\Omega_j$ . Chaque élément du maillage appartient à une seule partition.

Chaque processeur  $i$  ne connaît que sa partition de maillage et assemble une matrice

locale  $A_i$ . La matrice globale  $A$  correspondant au problème sur le domaine  $\Omega$  s'écrit alors :

$$A = \sum_{i=1}^{N_{proc}} \tilde{A}_i$$

avec  $\tilde{A}_i$  correspondant à la matrice  $A_i$  renumérotée et complétée par des zéros pour être de même dimension que  $A$ .

Conservant la même notation, et notant toujours  $Ax = b$  le problème global à résoudre, chaque processeur aura à résoudre  $A_i x_i = b_i$ . Cela revient, détaillé par (Marie, 1997), à des produits matrices-vecteur réalisés de manière parallèle sur chacun des processeur puis à un processus d'actualisation par somme des valeurs aux interfaces. Pour un processeur  $i$ , et pour tout  $j$  tel que  $\gamma_{ij} \neq \emptyset$ , les valeurs calculées aux interfaces sont envoyées au processeur  $j$  et les sommes calculées prennent en compte les valeurs locales et les valeurs reçues par les autres sous-domaines. Ainsi, la minimisation de la taille de l'intersection entre les sous-domaines est indispensable à l'efficacité du solveur parallèle.

L'utilisation de la librairie PETSc permet la prise en charge optimisée et en parallèle de nombreuses fonctions scientifiques, telles que les multiplications de matrices, de vecteurs ou de matrice-vecteur. Elle dispose également de multiples méthodes de résolution, dont la méthode du résidu conjugué, permettant d'avoir un code de calcul efficace en parallèle ((Balay *et al.*, 2015a), (Balay *et al.*, 2015b), (Balay *et al.*, 1997)).

## 2.4.2 Partitionnement et remaillage

Le partitionnement du maillage constitue une étape importante pour la parallélisation, la performance du calcul dépend de la taille des interfaces entre les sous-domaines et du bon équilibre des charges. Le partitionneur s'appuie sur la topologie du maillage et travaille en deux étapes, décrites succinctement ici et en détail par (Marie, 1997) et (Digonnet, 2001) :

- Un algorithme dit "glouton" commence par déterminer  $N_{proc}$  éléments de maillage appelés "germes" aussi distants que possible les uns des autres, avec  $N_{proc}$  le nombre de processeurs du calcul. Chacun de ces germes constituera le centre d'un sous-domaine de la partition, autour duquel la partition sera construite en y ajoutant les éléments adjacents.
- Une amélioration du partitionnement obtenue avec cet algorithme glouton s'effectue par minimisation du nombre de nœuds aux interfaces. Pour cela, à chaque élément on attribue un critère prenant en compte le nombre total d'éléments par processeur et le nombre de communications avec des éléments d'autres processeurs. En résulte une réattribution ou non de l'élément à un autre processeur.

Le remaillage parallèle repose sur le principe du remaillage séquentiel vu dans la section précédente 2.3 sous la contrainte du blocage des nœuds des interfaces entre les sous-domaines. L'algorithme est le suivant :

Tant que la totalité du domaine n'a pas été remaillée :

- La procédure de maillage séquentiel est effectuée en parallèle sur chaque processeur, sans modifier les interfaces. Elle ne nécessite aucune communication entre les processeurs.
- La procédure de repartitionnement est effectuée, rééquilibrant les tailles des sous-domaines et déplaçant les interfaces afin que celles-ci puissent être traitées lors du prochain remaillage.

Un exemple, sur un domaine 2D, est donné sur la figure 2.3.

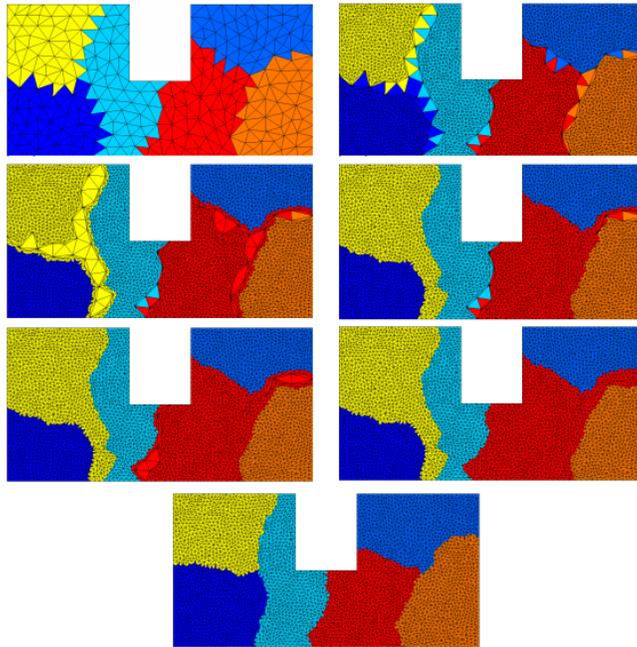


FIGURE 2.3 – Remaillage en parallèle sur 6 processeurs d'un domaine 2D (Digonnet *et al.*, 2007)

Nous verrons dans le chapitre suivant que cette stratégie permet également de construire les maillages grossiers de la méthode multigrille hybride développée, dans la section 3.1.1.

Après cette présentation du problème mécanique permettant de modéliser la mise en forme des matériaux et des stratégies adoptées par le logiciel FORGE® NxT, nous terminons ce

chapitre par une présentation des simulations que nous utiliserons afin de mettre en place et d'évaluer notre méthode multigrille.

## 2.5 Simulations de référence

### 2.5.1 Ecrasement entre tas plats

Ce premier problème est simple mais représentatif : l'écrasement d'un cylindre entre deux tas plats (figure 2.4). Le matériau suit un comportement Newtonien, avec  $m = 1$  et  $K = 1809$  MPa · s dans l'équation de la loi de Norton-Hoff 2.9. Le contact entre le lopin et l'outil est parfaitement collant, ainsi  $v_k = v^{outil}$  dans l'équation du contact 2.30. Le cylindre a une hauteur initiale de 90 mm et un diamètre de 21,9 mm. L'outil inférieur est fixe et l'outil supérieur a une vitesse de  $615 \text{ mm}\cdot\text{s}^{-1}$ . La hauteur finale est de 10 mm. Enfin, la symétrie de la simulation nous permet de considérer un sixième du cylindre pour les calculs. Pour mener cette simulation à son terme, 1 020 incréments de calcul sont nécessaires, avec un pas de temps d'environ  $10^{-4}$ s, pour une durée de procédé de 0,13s.

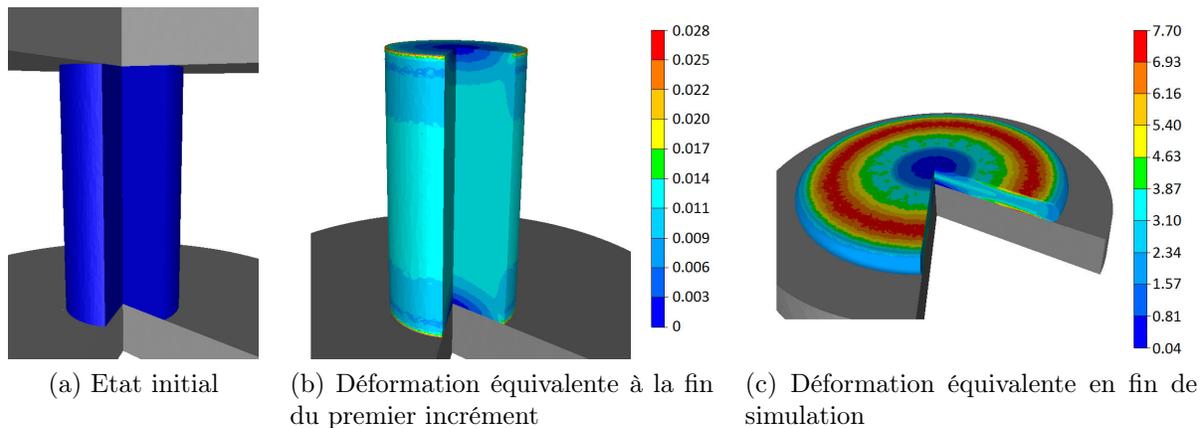


FIGURE 2.4 – Ecrasement entre tas plat

Ce premier cas test conduit à la résolution de problèmes linéaires simples. Il présente de plus l'avantage de pouvoir comporter un très grand nombre de nœuds, tout en ayant des temps de calculs raisonnables, ce qui permet de multiplier les expérimentations. Ainsi, nous l'utiliserons tout d'abord dans le chapitre 3 pour déterminer les paramètres optimaux de la méthode multigrille. Ensuite, nous y ferons également appel dans le chapitre 4 lorsque

nous aurons besoin d'un grand nombre de degrés de liberté afin de tester la convergence asymptotique et l'efficacité parallèle de la méthode.

### 2.5.2 Triaxe frottant avec repli

Le deuxième problème est plus complexe. Il présente le forgeage d'un triaxe à partir d'un lopin de forme quasi cylindrique (figures 2.5 et 2.6). Nous nous situons dans les conditions de forgeage à chaud, avec  $m = 0,14$  et  $K = 1809 \text{ MPa} \cdot \text{s}^m$  dans l'équation 2.9. La loi de Coulomb limitée par Tresca décrite en 2.17 est utilisée avec  $\mu = 0,15$  et  $\bar{m} = 0,3$  pour modéliser le frottement entre la pièce et les outils. Le cylindre a une hauteur initiale de 70 mm et un diamètre de 22,8 mm. L'outil inférieur est fixe et l'outil supérieur obéit à un pilotage par presse mécanique. Il se déplace à une vitesse moyenne de  $564 \text{ mm} \cdot \text{s}^{-1}$ . La fin du procédé correspond à une course de l'outil supérieur de 62 mm et est atteinte après environ 600 incréments, avec un pas de temps d'environ  $10^{-4} \text{ s}$  pour une durée de procédé de 0,11s. Tout comme pour l'écrasement entre tas plats, la symétrie nous permet de considérer seulement un sixième de la pièce lors du calcul.

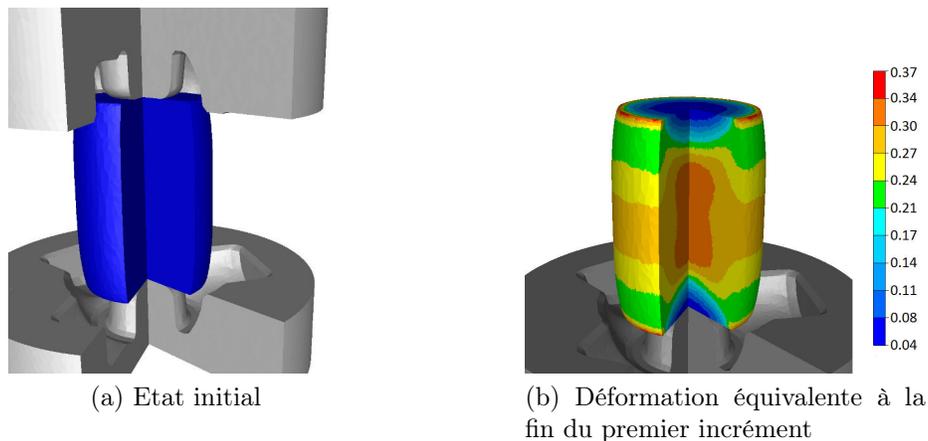


FIGURE 2.5 – Forgeage d'un triaxe - Début de simulation

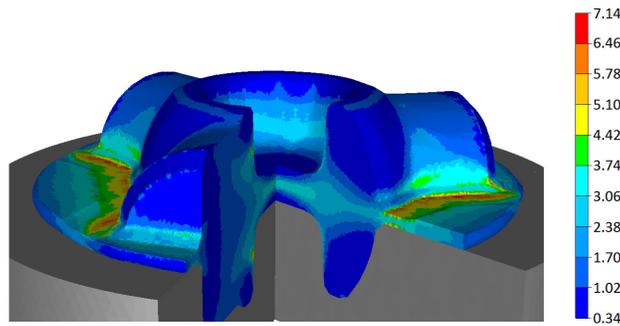


FIGURE 2.6 – Forgeage d’un triaxe - Déformation équivalente en fin de simulation

Ce deuxième cas test sera également utilisé dans le chapitre 3 comme étant plus représentatif d’un problème de forgeage. Nous considérerons alors un incrément de temps en particulier, en milieu de simulation, après une réduction de hauteur de 40 mm (figure 2.7). La pièce a alors une importante surface de contact avec l’outil complexe et le problème non-linéaire résultant est davantage représentatif de ceux pouvant être rencontrés lors de procédés de forgeage classiques. Dans le chapitre 4, ce cas d’application permettra également de mettre en évidence les performances de la méthode multigrille sur des simulations présentant des replis et un écoulement très contraint.

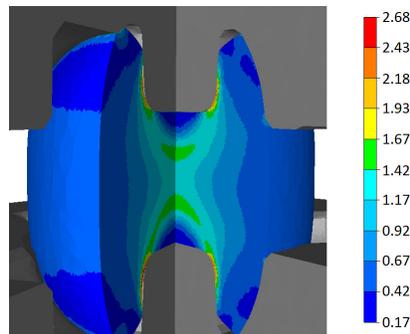


FIGURE 2.7 – Déformation équivalente du triaxe en milieu de simulation après une course de 40 mm de l’outil

### 2.5.3 Laminage retour

La simulation du laminage retour constitue notre troisième problème de mise en forme (figure 2.8), sur lequel nous évaluerons les performances de la méthode multigrille (dans le chapitre 4). Elle est composée de deux passes qui mettent en forme une géométrie initiale

parallélépipédique de dimension 57 mm x 57 mm x 108 mm. Nous nous plaçons une nouvelle fois dans des conditions de forgeage à chaud avec  $m = 0,15$  et  $K = 1576 \text{ MPa} \cdot \text{s}^m$  dans l'équation 2.9. La loi de Coulomb limité par Tresca (equation 2.17) modélise le frottement entre les outils et la pièce, avec  $\mu = 0,4$  et  $\bar{m} = 0,8$ . Les outils ont une vitesse de rotation de 600 tpm. La totalité du procédé est simulée après environ 130 incréments, avec un pas de temps d'environ  $10^{-3}$ s pour une durée de procédé de 0,39s.

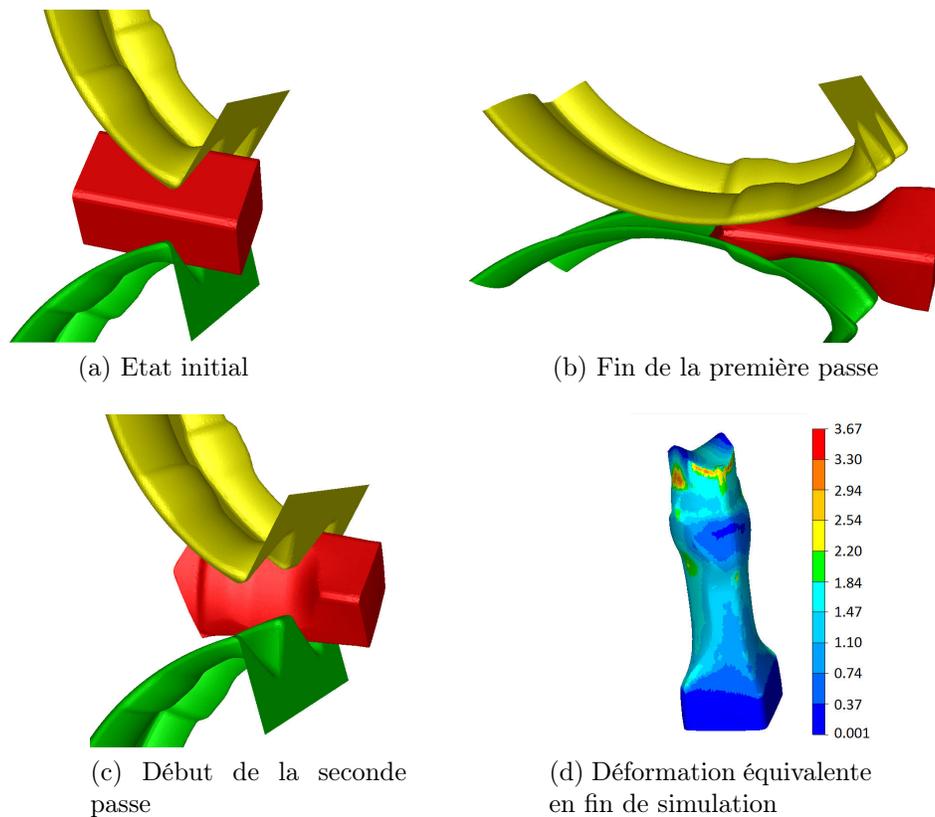


FIGURE 2.8 – Simulation de laminage retour

Ce cas d'application permettra de tester les performances de la méthode multigrille sur la simulation d'un procédé bien différent des deux précédents et représentatif du forgeage, avec un nombre de nœuds relativement faible. Cela permettra d'évaluer l'efficacité de la méthode pour des simulations de petites tailles mais avec des écoulements complexes et peu contraints.

### 2.5.4 Martelage rotatif

Le dernier problème retenu est la simulation d'un procédé de martelage rotatif (figure 2.9). Le lopin initial est un tube cylindrique de 140 mm de longueur et 63 mm de diamètre extérieur et 40 mm de diamètre intérieur. Nous nous situons dans les conditions de forgeage à chaud, avec  $m = 0,15$  et  $K = 1324 \text{ MPa} \cdot \text{s}^m$  dans l'équation 2.9. La loi de Coulomb limitée par Tresca décrite en 2.17 est utilisée avec  $\mu = 0,15$  et  $\bar{m} = 0,3$  pour modéliser le frottement entre la pièce et le mandrin cannelé. Ce procédé complexe consiste en l'utilisation de multiples marteaux qui écrasent la pièce positionnée sur un mandrin. La pièce subit de plus une série de rotations qui alternent avec les nombreux coups de ces marteaux. La simulation de la totalité du procédé requiert 1 200 incréments, avec un pas de temps d'environ  $5 \cdot 10^{-3} \text{ s}$  pour une durée de procédé de 2,93s.

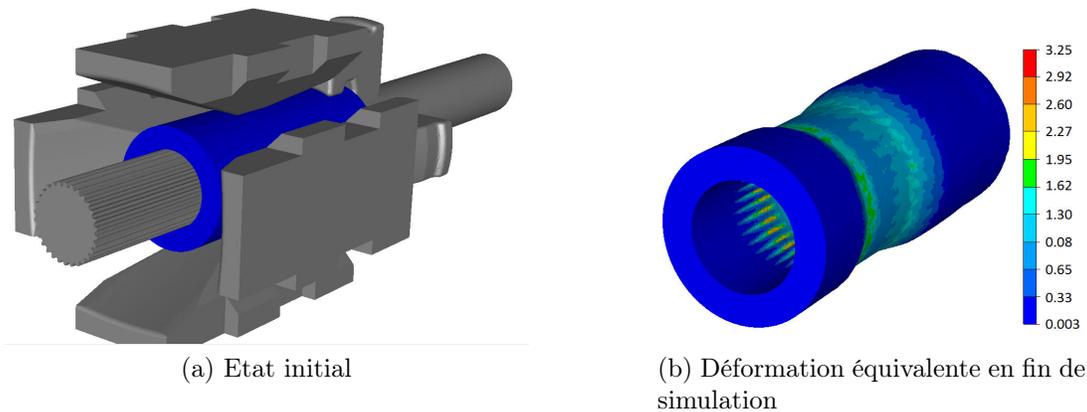


FIGURE 2.9 – Simulation de martelage rotatif

Ce dernier cas d'application présente un très grand nombre de nœuds et correspond à un procédé de forgeage incrémental dont la simulation est aujourd'hui très longue, à la fois du fait de la taille du problème et du nombre de pas de temps. L'association d'accélération parallèle et multigrille est donc espérée sur ce problème.

Après ces deux premiers chapitres consacrés à un état de l'art et à l'exposé du problème mécanique, le chapitre suivant traitera de la méthode multigrille développée pendant cette thèse. On s'appliquera à décrire les différents éléments de la méthode, la mise au points des multiples paramètres ainsi que les premiers résultats.



# 3 | Méthode Multigrille pour FORGE®

I know it's not ordinary. But who  
ever loved ordinary?

---

Joan Clarke

Nous présentons dans cette partie la stratégie multigrille hybride développée durant cette thèse. Cette méthode a pour objectif de développer une méthode de résolution permettant de réduire les temps de calcul des simulations de procédés de mise en forme présentant les spécificités décrites dans le chapitre précédent. Combinant des aspects des multigrilles géométriques et algébriques vus dans le chapitre 1, elle est basée sur une génération de maillages par déraffinement non emboîté et sur l'utilisation de la relation de Galerkin. Après une présentation des méthodes de construction des éléments de l'approche choisie, nous discuterons des réglages effectués sur le préconditionneur, concernant le lisseur et le nombre de lissages, le facteur de déraffinement et une étude sur la taille et la qualité de la grille grossière visant à optimiser la méthode. Enfin, nous discuterons des performances du préconditionneur multigrille développé, associé à une autre méthode de résolution, ainsi que de l'alternative d'un solveur multigrille.

## Sommaire

---

<b>3.1</b>	<b>Méthode multigrille hybride</b> . . . . .	<b>49</b>
3.1.1	Construction parallèle des maillages grossiers . . . . .	49
3.1.2	Construction parallèle des opérateurs de transfert . . . . .	51
3.1.3	Éléments complémentaires . . . . .	54
<b>3.2</b>	<b>Réglage du préconditionneur Multigrille</b> . . . . .	<b>54</b>
3.2.1	Lisseur . . . . .	55
3.2.2	Facteur de déaffinement . . . . .	59
3.2.3	Grille grossière . . . . .	60
<b>3.3</b>	<b>Extensions de la méthode multigrille</b> . . . . .	<b>64</b>
3.3.1	GMRES + Multigrille . . . . .	64
3.3.2	Solveur Multigrille . . . . .	67
<b>3.4</b>	<b>Conclusion</b> . . . . .	<b>70</b>

---

## 3.1 Méthode multigrille hybride

L’approche retenue vise à combiner les avantages des multigrilles algébriques et géométriques et constitue ainsi une stratégie multigrille hybride. D’une part, ayant à notre disposition un mailleur robuste et efficace présenté dans le chapitre précédent, section 2.3, nous le mettons à profit afin de générer différents niveaux de maillages grossiers par déraffinement, suivant en cela l’approche géométrique des multigrilles. D’autre part, le calcul des basses fréquences de la solution sur les grilles grossières nécessite de conserver la condition d’incompressibilité et les conditions aux limites lors des opérations de projection, ce qui nécessite une méthode de type Galerkin (equation 1.14), caractéristique des multigrilles algébriques.

### 3.1.1 Construction parallèle des maillages grossiers

Pour construire les niveaux de calcul grossier, on se base sur la génération de maillages construits par déraffinement à partir du maillage fin. Comme vu dans le chapitre 1, section 1.3, la génération de maillages emboîtés (Rey, 2007) ne permet pas de traiter des géométries complexes. Nous utilisons donc ici une approche non emboîtée.

Lors d’un calcul parallèle, le maillage initial, fournissant la géométrie du domaine, est partitionné en plusieurs sous-domaines répartis sur les différents processeurs. La génération des maillages grossiers doit donc se faire également de manière parallèle, en s’inspirant de l’approche décrite dans la section 2.3.1, sur la base des travaux de (Gruau et Coupez, 2005) et de (Coupez *et al.*, 2000). A partir du maillage initial, un maillage plus grossier est généré de manière itérative en suivant les étapes présentée sur la figure 3.1, combinant remaillage local et repartitionnement de maillage. On illustre cette stratégie sur un maillage 2D mais la méthode est naturellement compatible avec un problème 3D. Le choix de la taille de maille du maillage grossier sera discuté dans la section 3.2.2.

Sur ce domaine 2D (figure 3.1.1), les partitions du maillage initial subissent tout d’abord un déraffinement parallèle en bloquant les interfaces entre les sous-domaines (figure 3.1.2). Un repartitionnement parallèle du domaine (figure 3.1.3) permet ensuite le déplacement de ces interfaces, de telle sorte qu’elles puissent être remaillées durant l’étape de déraffinement suivante (figure 3.1.4). Ce processus est répété jusqu’à obtenir le déraffinement complet de la géométrie initiale. Le nombre d’opérations à effectuer décroît rapidement avec les itérations de remaillage/repartitionnement. La diminution du nombre d’éléments à traiter permet de générer ces maillages grossiers en un temps raisonnable, comme nous le verrons plus en détail dans les applications, section 4.2. L’algorithme se termine par une dernière étape d’équilibrage des charges, en déplaçant une nouvelle fois les interfaces entre les sous-domaines (figure 3.1.7).

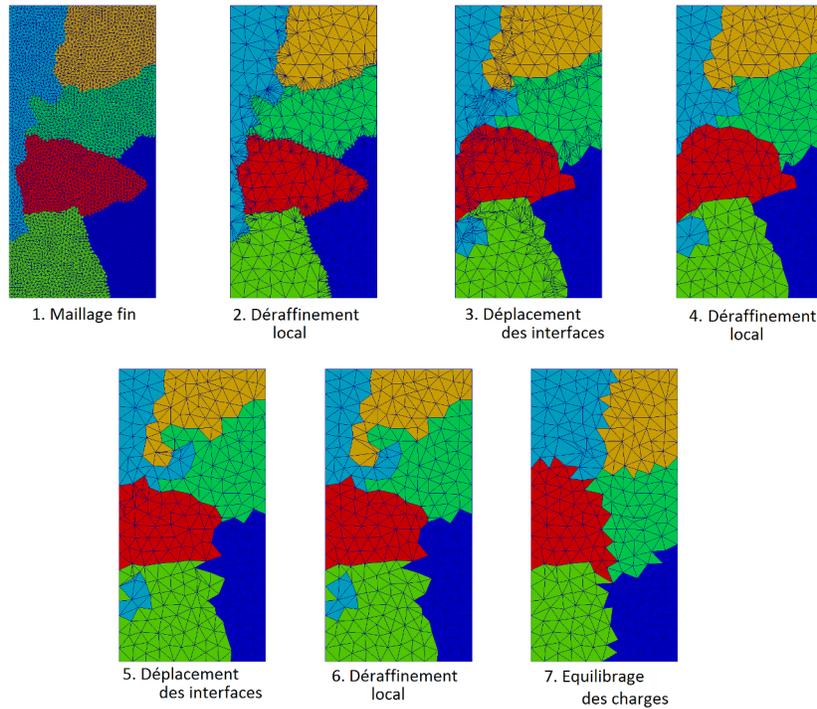


FIGURE 3.1 – Construction itérative parallèle d'un maillage grossier par déraffinement sur 6 coeurs

Pour les géométries complexes rencontrées en mise en forme, l'algorithme de déraffinement doit préserver la géométrie du domaine initial, en respectant une condition sur la différence de volume entre le maillage fin et le maillage grossier : soit  $M_f$  le maillage initial composé d'éléments  $Ef_i$ ,  $i$  allant de 1 au nombre d'éléments du maillage fin,  $nbe_f$ . Notons  $M_g$  le maillage grossier composé d'éléments  $Eg_j$ ,  $j$  allant de 1 au nombre d'éléments du maillage grossier,  $nbe_g$ . Le maillage grossier doit vérifier :

$$\frac{1}{\alpha} > \frac{\sum_{i=1}^{nbe_f} V(Ef_i)}{\sum_{j=1}^{nbe_g} V(Eg_j)} > \alpha \quad (3.1)$$

avec  $V(E)$  le volume de l'élément  $E$ .  $\alpha$  représente ainsi la "variation de volume" autorisée lors de la construction du maillage grossier. Lors d'un remaillage classique, ce paramètre est fixé à 0.99 (1% d'erreur). Dans le cadre des multigrilles, les maillages grossiers ne servant

qu'à calculer une correction basses fréquences, ce critère peut être relâché. Nos expériences numériques montrent que  $\alpha = 0.9$  (10% d'erreur) offre le meilleur compromis pour générer des maillages grossiers qui conservent la géométrie initiale avec un nombre de nœuds aussi faible que nécessaire pour les performances des multigrilles, comme nous le verrons dans la suite. Par exemple, sur la figure 3.2, où la méthode de déraffinement est appliquée à la fin du forgeage d'un triaxe, le déraffinement permet de passer d'un maillage de plus de 266 000 nœuds à un maillage n'en comportant plus que 2 000.

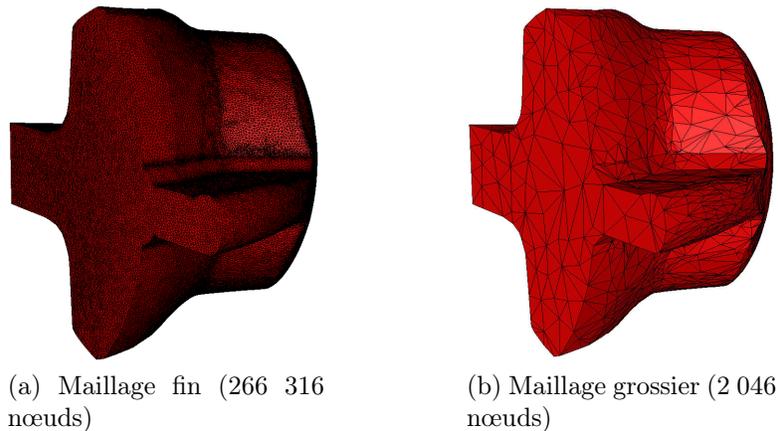


FIGURE 3.2 – Maillage initial (gauche) et maillage grossier (droite) obtenu par déraffinement. Cas du triaxe en fin de forgeage

Une fois les différents niveaux de maillages générés, la définition des opérateurs de transfert est nécessaire afin d'envoyer les informations d'une grille à une autre.

### 3.1.2 Construction parallèle des opérateurs de transfert

Les opérateurs de transfert permettent d'échanger les données d'un maillage à un autre. Entre deux niveaux, nous pouvons distinguer l'opérateur de restriction  $R$ , permettant de passer du plus fin au plus grossier, et l'opérateur d'interpolation  $P$ , permettant le passage du plus grossier vers le plus fin. Nous décrivons ici la construction de l'opérateur d'interpolation. Il est défini comme étant la matrice d'interpolation des coordonnées des nœuds du maillage fin dans le maillage grossier.

$P$  est donc une matrice dont les composantes sont les coordonnées barycentriques de la projection des nœuds fins sur les éléments du maillages grossiers. Ainsi, elle est de taille  $n_h \times n_H$ , avec  $n_h$  et  $n_H$  respectivement le nombre de nœuds sur le maillage fin et sur le maillage

grossier, chaque ligne de  $P$  contenant les coordonnées barycentriques du nœud fin (correspondant à cette ligne) projeté sur l'élément grossier contenant ce nœud. Un exemple 1D est donné sur la figure 3.3.

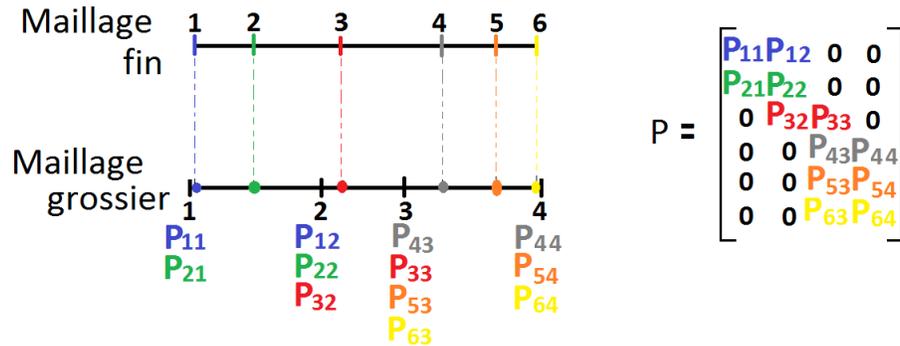


FIGURE 3.3 – Construction de la matrice d'interpolation

Sur certaines géométries complexes, les frontières des maillages fin et grossier peuvent ne pas coïncider parfaitement, et aucun élément grossier contenant le nœud fin n'est alors trouvé (figure 3.4). La stratégie adoptée est de projeter ce nœud sur le domaine d'arrivée (le maillage grossier) et donc sur l'élément le plus proche, et de calculer les coordonnées barycentriques du projeté sur cet élément. Cette approche permet de garder des valeurs positives dans la matrice d'interpolation, contrairement à une approche par extrapolation (c'est à dire à l'extérieur de l'élément le plus proche) conduisant à des valeurs hors des bornes de l'interpolation.

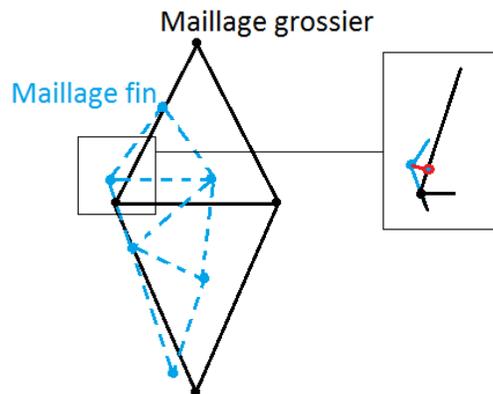


FIGURE 3.4 – Projection d'un nœud si aucun élément grossier contenant n'est trouvé

Lors d'un calcul en parallèle, cette reprojection est précédée par la recherche d'un "élément contenant" sur l'ensemble des processeurs. Ainsi, si la projection du nœud fin n'appartient pas au même processeur, ses coordonnées sont envoyées aux autres processeurs pour une recherche sur leurs sous-domaines associés. Si aucun élément contenant n'est trouvé, l'élément le plus proche est retenu. Afin de réduire les communications, les différents niveaux de grilles doivent avoir des partitionnements aussi semblables que possible. Des sous-domaines parfaitement coïncidants sont toutefois difficiles à obtenir, comme le montre la figure 3.5, mais nous verrons dans la section 4.2 que cela ne nuit pas significativement aux performances de la méthode.

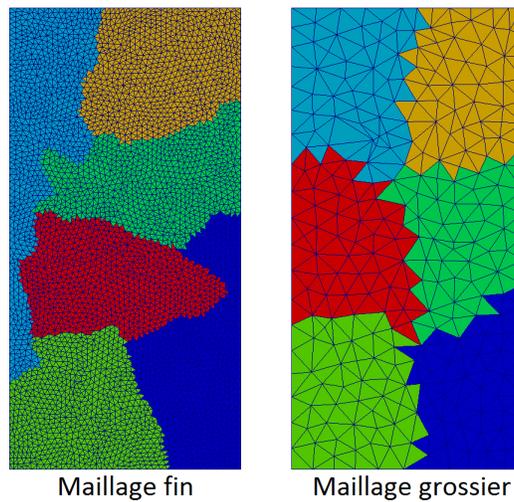


FIGURE 3.5 – Partitionnement de maillages fin (gauche) et grossier (droite)

Contrairement à l'approche purement géométrique, avec la stratégie hybride la construction du système grossier ne repose pas sur l'assemblage de la matrice de raideur sur la grille grossière. En effet, comme vu dans le chapitre précédent, les matrices issues des problèmes de mise en forme présentent des caractéristiques particulières en raison de la formulation mixte stabilisée et de la gestion du contact par pénalisation. Pour assurer l'efficacité de la méthode multigrille, on utilise la relation de Galerkin, généralement associée aux méthodes algébriques. Les matrices grossières  $A_H$  sont construites à partir de la matrice fine  $A_h$  suivant :

$$A_H = RA_hP \quad (3.2)$$

Ces produits de matrices sont effectués à l'aide des multiplications matrice-matrice de PETSc pour un calcul en parallèle efficace. Enfin, et afin de conserver la symétrie de la matrice

initiale, nous définissons l'opérateur de restriction  $R$  comme étant la transposée de l'opérateur d'interpolation  $P$  :

$$R = P^T \quad (3.3)$$

La construction des différents niveaux de maillages et des opérateurs de transfert ayant été décrite, définissons maintenant les derniers composants de la méthode multigrille hybride avant de passer au réglage des paramètres.

### 3.1.3 Éléments complémentaires

Bien que les méthodes multigrilles aient initialement été développées pour être des solveurs, leur utilisation en tant que préconditionneur est plus fréquente dans les travaux récents. Nous retenons, nous aussi, cette approche. Nous verrons dans la section 3.3 que les deux approches (solveur et préconditionneur) se valent et que l'utilisation en préconditionnement est un peu plus efficace. Nous utilisons la librairie PETSc et son préconditionneur nommé PCMG comme support de notre implémentation.

Pour une utilisation en tant que solveur comme en tant que préconditionneur, la méthode de résolution du système sur la grille grossière peut être soit itérative, soit directe. Dans la mesure où l'algorithme de génération des grilles grossières nous permet d'obtenir des maillages avec un nombre de nœuds raisonnablement faible, la méthode directe s'avère plus efficace et nécessite moins de paramètres. Nous utilisons donc une résolution directe de type LU, qui doit être compatible avec le calcul parallèle. La librairie MUMPS ((Amestoy *et al.*, 2001) et (Amestoy *et al.*, 2006)) remplit ces exigences et sera donc retenue.

L'utilisation de la méthode multigrille en tant que préconditionneur repose sur le choix d'une méthode de résolution. Le solveur de référence étant la méthode du résidu conjugué préconditionné, nous utiliserons cette méthode, avant de discuter, en fin de chapitre, de la possibilité d'utiliser une autre méthode de résolution.

## 3.2 Réglage du préconditionneur Multigrille

Dans cette section, nous utilisons les exemples d'application décrits dans la section 2.5 afin de déterminer numériquement les derniers paramètres de la méthode multigrille, à savoir :

- Le lisseur à utiliser sur les grilles fine et intermédiaire
- Le nombre d'itérations de pré-lissage et post-lissage
- Le facteur de déraffinement de maillage entre les niveaux
- La taille optimale et la qualité minimale du maillage pour les grilles grossières

### 3.2.1 Lisseur

Dans la littérature, le lisseur d'une méthode multigrille est le plus souvent une méthode itérative simple présentant de bonnes propriétés de réduction des hautes fréquences. De nombreux auteurs utilisent la méthode de Jacobi ou de Gauss-Seidel ((Adams et Taylor, 2000), (Gmeiner *et al.*, 2015)). Dans (Adams *et al.*, 2003), il est montré que des lisseurs plus complexes peuvent toutefois être plus efficaces, en fonction du type de problème à résoudre. Nous avons ainsi testé différentes méthodes itératives disponibles dans la librairie PETSc.

Pour les problèmes de mise en forme, suivant (Rey, 2007), la méthode de Richardson préconditionnée (PR) se montre la plus adaptée :

$$Ax = b \quad \left| \begin{array}{l} x_0 \leftarrow \text{solution initiale} \\ \forall k > 0, \quad x_{k+1} = x_k + \omega M^{-1}(b - Ax_k) \end{array} \right. \quad (3.4)$$

avec  $M$  un préconditionneur et  $\omega$  un paramètre de relaxation permettant d'assurer la convergence. Plusieurs tests numériques ont montré qu'une valeur  $\omega = \frac{2}{3}$  était nécessaire et suffisante pour assurer convergence et efficacité.

Cette méthode permet de construire trois lisseurs différents selon le type de matrice  $M$  retenue :

- Richardson préconditionné par Jacobi (PR + Jac) :

$$M = D, \text{ avec } D \text{ la matrice bloc diagonale de } A \quad (3.5)$$

- Richardson préconditionné par SSOR (PR + SSOR) :

$$\begin{aligned} M &= (D + L)D^{-1}(D + L)^T \\ \text{avec } D &\text{ la diagonale et } L \text{ la partie triangulaire inférieure de } A \\ A &= D + L + L^T \end{aligned} \quad (3.6)$$

- Richardson préconditionné par ILU0 (PR + ILU0) : voir équation 2.47

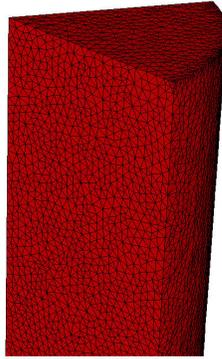
Notons que la méthode de Richardson préconditionnée par ILU1 a également fait l'objet de tests mais que ceux-ci ont montré qu'elle était peu efficace. Son coût supplémentaire tant lors de sa construction que de son utilisation n'est pas compensé par la réduction du nombre d'itérations qui en résulte.

Pour évaluer les performances de ces trois lisseurs, on considère une méthode bigrille idéale. Elle se traduit par l'utilisation de seulement deux niveaux de grilles et d'une résolution directe sur la grille grossière. Le temps CPU passé sur la grille grossière pour la

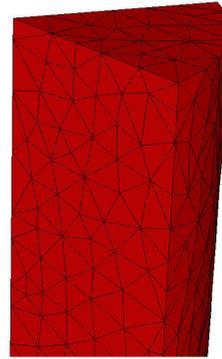
factorisation et la résolution de son système linéaire est constant, tout comme le temps passé sur le transport entre les niveaux. Ainsi, la variation du temps total de résolution ne dépend que du lisseur, du coût de chaque itération de lissage ainsi que nombre d'itérations de pré et post lissage. On peut ainsi évaluer l'influence des lisseurs indépendamment des autres optimisations à venir.

### Ecrasement entre tas plat

Nous utilisons tout d'abord le cas d'écrasement entre tas plats présenté dans la section 2.5.1. La figure 3.6a montre le maillage fin de 22 000 nœuds considéré, à partir duquel le maillage grossier de 900 nœuds a été construit (figure 3.6b).



(a) Maillage fin - 22 059 nœuds



(b) Maillage grossier - 902 nœuds

FIGURE 3.6 – Vue partielle des maillages de calcul et grossier du cas de tas plat pour l'étude du lisseur

Afin d'évaluer les performances des trois lisseurs, nous faisons varier le nombre de pré et de post lissages de 1 à 4 en testant toutes les combinaisons possibles. Les temps de calcul sont présentés dans le tableau 3.1. Ils incluent la résolution du système linéaire mais aussi l'initialisation du préconditionneur multigrille : construction des opérateurs de transfert et de la matrice grossière, factorisation de la matrice grossière, construction du préconditionneur de la grille fine et autres initialisations.

Nous pouvons tout d'abord constater que pour le lisseur PR+ILU0, une combinaison de 4 pré lissages et de 1 post lissage (notée (4,1)) entraîne la non convergence du solveur linéaire après 500 itérations. Nous observons également une convergence difficile et très lente avec

Nb it pré lissage	Nb it post lissage	CPU	CPU	CPU
		PR+ILU0 (s)	PR+Jacobi(s)	PR+SSOR (s)
1	1	9,59	8,99	10,23
1	2	9,46	Div	Div
1	3	10,01	Div	Div
1	4	9,4	Div	Div
2	1	19,85	Div	Div
2	2	9,15	8,78	10,64
2	3	9,95	Div	Div
2	4	9,88	Div	Div
3	1	10,32	Div	Div
3	2	9,97	Div	Div
3	3	9,89	8,41	10,88
3	4	10,34	Div	Div
4	1	Non conv	Div	Div
4	2	10,74	Div	Div
4	3	10,85	Div	Div
4	4	10,35	9,06	12,23

TABLE 3.1 – Comparaison des temps de calculs obtenus avec les trois lisseurs pour la résolution d’un système linéaire issu de la simulation d’un écrasement entre tas plats avec une méthode bigrille idéale.

la combinaison (2,1). En dehors de ces deux combinaisons, le lisseur PR+ILU0 présente des résultats homogènes, compris entre 9,15 secondes et 10,85 secondes. Enfin, à l’exception de la combinaison (1,2), un peu meilleure que la (1,1), nous observons que pour un nombre de pré lissages donné, le meilleur résultat est toujours obtenu avec un nombre de post lissages identique. Ce phénomène se produit plus brutalement avec les lisseurs PR+Jacobi et PR+SSOR : il n’y a pas convergence du solveur linéaire dès lors que nous utilisons une combinaison non symétrique de pré et post lissages.

Pour ce système linéaire, le lisseur PR+Jacobi (3,3) se montre plus performant que les lisseurs plus élaborés que sont PR+ILU0 et PR+SSOR. La différence est néanmoins assez faible : elle représente environ 4% par rapport au temps CPU du meilleur des deux autres, PR+ILU0.

### Forgeage du triaxe

Nous considérons maintenant un problème plus complexe et plus proche des simulations usuelles en mise en forme, le forgeage du triaxe en milieu de simulation, présenté dans la section 2.5.2 et sur la figure 2.7. Nous résolvons ainsi un problème non linéaire sur un maillage

fin comportant 16 902 nœuds (figure 3.7a), à partir duquel nous avons construit un maillage grossier de 936 nœuds (figure 3.7b).

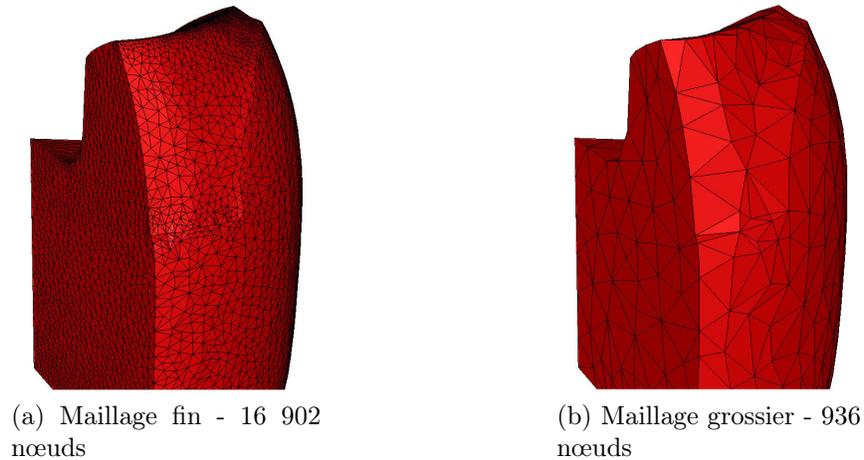


FIGURE 3.7 – Vue partielle des maillages de calcul et grossier pour le cas du forgeage d’un triaxe pour l’étude de lisseurs

Pour le lisseur PR+ILU0, nous limitons notre étude aux 5 combinaisons de pré et post lissage ayant donné les meilleurs temps de calcul sur le cas précédent. Pour les 2 autres lisseurs (PR+Jacobi et PR+SSOR), nous nous limitons aux combinaisons (1,1), (2,2) et (3,3). Les résultats sont présentés sur le tableau 3.2. Ils correspondent ici au calcul d’un incrément entier qui nécessite 12 itérations de Newton-Raphson, et donc autant de résolutions de systèmes linéaires.

Nb it pré lissage	Nb it post lissage	CPU	CPU	CPU
		PR+ILU0 (s)	PR+Jacobi (s)	PR+SSOR (s)
1	1	128,0	313,1	<b>183,0</b>
2	2	158,4	248,9	164,2
3	3	179,2	275,8	191,7
1	2	Non conv		
1	4	198,5		

TABLE 3.2 – Comparaison des temps de calcul obtenus avec les trois lisseurs pour la résolution d’un incrément de temps issu de la simulation de forgeage d’un triaxe avec une méthode bigrille idéale

Notons tout d'abord que pour le lisseur PR+ILU0, les combinaisons symétriques donnent de meilleurs résultats que les deux combinaisons non symétriques testées, la combinaison (1,2) allant même jusqu'à ne pas converger. Sur ce problème plus complexe, les résultats sont moins homogènes, et nous pouvons dégager plus facilement des tendances pour le choix du lisseur et de la combinaison d'itérations de lissage. Ainsi, alors que le PR+Jacobi présentait les meilleurs résultats sur le problème précédent, les lisseurs plus complexes PR+ILU0 et PR+SSOR sont ici plus efficaces. Le lisseur le plus puissant, PR+ILU0, se montre également le plus efficace avec une convergence plus rapide, malgré son coût beaucoup plus grand pour la construction de factorisation incomplète et pour chaque résolution. Par rapport à la meilleure combinaison (2,2) à la fois pour PR+Jacobi et PR+SSOR, la combinaison (1,1) du lisseur PR+ILU0 converge près de deux fois plus rapidement que PR+Jacobi et 1,3 fois plus rapidement que PR+SSOR.

Ainsi, compte tenu de la faiblesse de l'avantage apporté par le lisseur PR+Jacobi sur le cas linéaire, le lisseur PR+ILU0 (1,1) apparaît comme le mieux adapté et sera retenu dans la suite.

### 3.2.2 Facteur de déraffinement

Pour une méthode à 3 grilles, notons 1 le niveau le plus fin, 2 le niveau intermédiaire et 3 le niveau le plus grossier. Le facteur de déraffinement  $F_i$  correspond au ratio entre le nombre de nœuds du niveau de maillage  $i$  sur celui du niveau de maillage  $i + 1$ .

La grille intermédiaire doit être à la fois assez grossière pour minimiser le coût du lissage, et assez fine pour transférer correctement les corrections sur la grille fine. La taille de la grille grossière est ici fixée à 500 nœuds, pour permettre une résolution directe rapide. Nous verrons dans la section suivante 3.2.3 la pertinence de ce choix.

Nous considérons le cas d'écrasement entre tas plats avec un maillage fin d'environ 120 000 nœuds. Différentes combinaisons de facteurs de déraffinement ( $F_1, F_2$ ) sont considérées, permettant :

- de conserver une grille grossière proche de 500 nœuds
- d'étudier des facteurs de déraffinement : homogène (14,14), plus fort entre la grille fine et la grille intermédiaire (24,9) ou plus fort entre la grille intermédiaire et la grille grossière (7,30).

Le tableau 3.3 montre l'influence de ces facteurs sur le temps de calcul.

Maillage fin initial : 119 772 noeuds	Maillage intermédiaire	Maillage grossier	Temps CPU(s)
Facteur de déraffinement	7.11	30.51	86.8
Nombre de noeuds	16 840	552	
Facteur de déraffinement	14.16	14.99	71.61
Nombre de noeuds	8 456	564	
Facteur de déraffinement	24.3	9.11	76.83
Nombre de noeuds	4 929	541	

TABLE 3.3 – Impact de différents facteurs de déraffinement entre les grilles sur les temps de calcul pour une méthode multigrille à 3 niveaux appliquée à la résolution d’un système linéaire provenant de la simulation d’écrasement entre tas plats

Les facteurs de déraffinement homogènes offrent clairement (et logiquement) les meilleures performances. Nous nous efforcerons donc dans la suite de respecter cette propriété.

### 3.2.3 Grille grossière

Nous nous intéressons maintenant à la grille grossière. Dans un schéma multigrille classique, on choisit un facteur de déraffinement "optimal" que l’on utilise ensuite pour déraffiner successivement les différents maillages jusqu’à obtenir un maillage grossier sur lequel une résolution directe est plus efficace qu’une résolution itérative. Le nombre de niveaux dépend donc du nombre de déraffinements effectués et peut devenir élevé. L’approche algébrique nous incite à limiter ce nombre de grilles, à 2 ou 3, afin que l’augmentation de la largeur de bande du système linéaire à chaque nouveau niveau grossier ne pénalise pas la résolution sur la grille grossière. Nous recherchons donc ici un compromis entre la valeur du facteur de déraffinement et la taille de la grille grossière. Il est évident que plus son nombre de nœuds est faible, plus la factorisation et la résolution de son système linéaire seront rapides mais plus le facteur de déraffinement entre les grilles sera élevé. De plus, pour des domaines à géométrie complexe, il n’est pas toujours possible de produire des maillages suffisamment grossiers respectant la géométrie du domaine. Il en résulte que la qualité du maillage entre aussi en considération.

Dans (Rey, 2007), la limite supérieure est fixée à 1 500 nœuds. Au dessus de cette valeur, il est suggéré d’introduire un nouveau niveau de grille ou d’utiliser une méthode de résolution itérative sur la grille grossière.

Suivant cette recommandation, nous étudions l’influence de la taille de la grille grossière sur les performances de la méthode multigrille.

Considérons une nouvelle fois la simulation d’écrasement entre tas plats. A partir d’un maillage initial, deux niveaux de maillages sont générés. Faisant varier la taille de la grille

grossière entre 500 et 1 600 nœuds, l'utilisation de facteurs de déraffinement homogènes entre les différents niveaux de maillage nous permet de déduire la taille de la grille intermédiaire. Nous fixons cette limite à 500 nœuds afin de préserver la géométrie du domaine. Dans le tableau 3.4, nous présentons, pour trois maillages initiaux donnés (de 80 000, 120 000 et 160 000 nœuds), les temps de calcul en fonction de la taille de la grille grossière. Le facteur de déraffinement et le nombre de nœuds de la grille intermédiaire sont également donnés.

	Fact. Déraffinement	Maillage intermédiaire	Maillage grossier	Temps CPU (s)
Maillage initial : 81 244 noeuds	7,3	11 590 noeuds	1 515 noeuds	76,65
	8,6	9 350 noeuds	1 099 noeuds	53,76
	12	6 754 noeuds	567 noeuds	45,76
Maillage initial : 119 772 noeuds	Fact. Déraffinement	Maillage intermédiaire	Maillage grossier	Temps CPU (s)
	8,5	14 622 noeuds	1 665 noeuds	106,14
	10	11 293 noeuds	1 163 noeuds	83,89
	14,5	8 456 noeuds	564 noeuds	76,83
Maillage initial : 162 509 noeuds	Fact. Déraffinement	Maillage intermédiaire	Maillage grossier	Temps CPU (s)
	10	15 976 noeuds	1 615 noeuds	151,48
	12,25	13 338 noeuds	1 083 noeuds	115,32
	16,7	9 635 noeuds	582 noeuds	103,56

TABLE 3.4 – Influence de la taille de la grille grossière pour une méthode trois grilles sur le temps de calcul lors de la résolution d'un système linéaire extrait de l'écrasement entre tas plat avec un maillage initial de 80 000, 120 000 et 160 000 nœuds

Quelque soit la taille du maillage initial étudié, les meilleurs résultats sont obtenus avec le maillage grossier comportant le moins de nœuds. Ainsi, sur le maillage à 160 000 nœuds, l'utilisation d'un facteur de déraffinement de plus de 16 donne la résolution la plus rapide. Notons que 16 peut sembler une valeur importante, mais en considérant les 3 dimensions d'espace du problème, cela correspond à une division par 2,5 de la taille de grille dans chacune des directions, ce qui n'est que légèrement supérieur à la recommandation généralement admise d'une division par 2.

Pour compléter cette étude, nous nous intéressons à la simulation du triaxe en effectuant une reprise à l'état présenté sur la figure 3.8b jusqu'à l'état de la figure 3.8c. A partir du maillage initial ayant 39 238 nœuds (figure 3.8a), nous comparons l'utilisation d'une grille grossière à 500 nœuds (figure 3.9a) avec une grille grossière à 1 500 nœuds (figure 3.9b). Afin de hiérarchiser l'importance de la qualité du maillage vis à vis du nombre de nœuds, des grilles grossières de mauvaise qualité sont également produites (figures 3.9c et 3.9d).

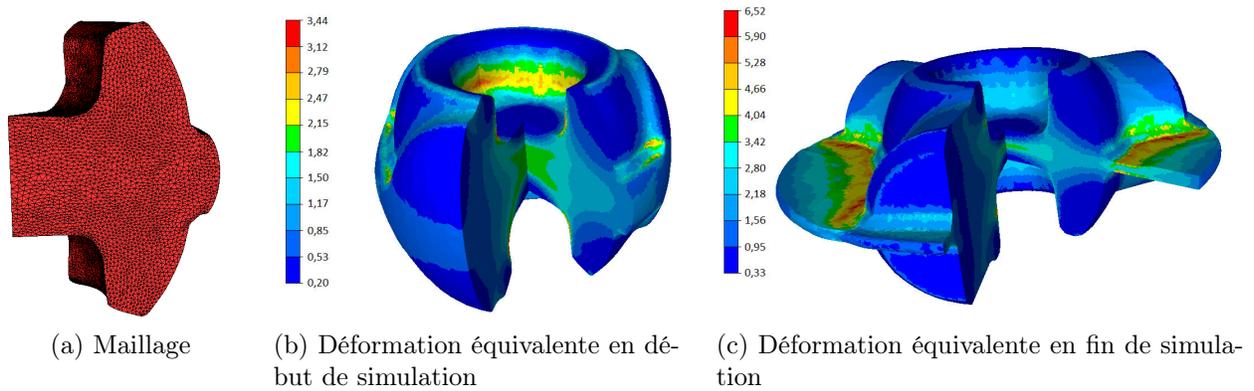


FIGURE 3.8 – Vue partielle du maillage de 39 238 noeuds (a), état à la reprise (b) et en fin de simulation (c) lors du forgeage d’un triaxe pour tests sur la grille grossière

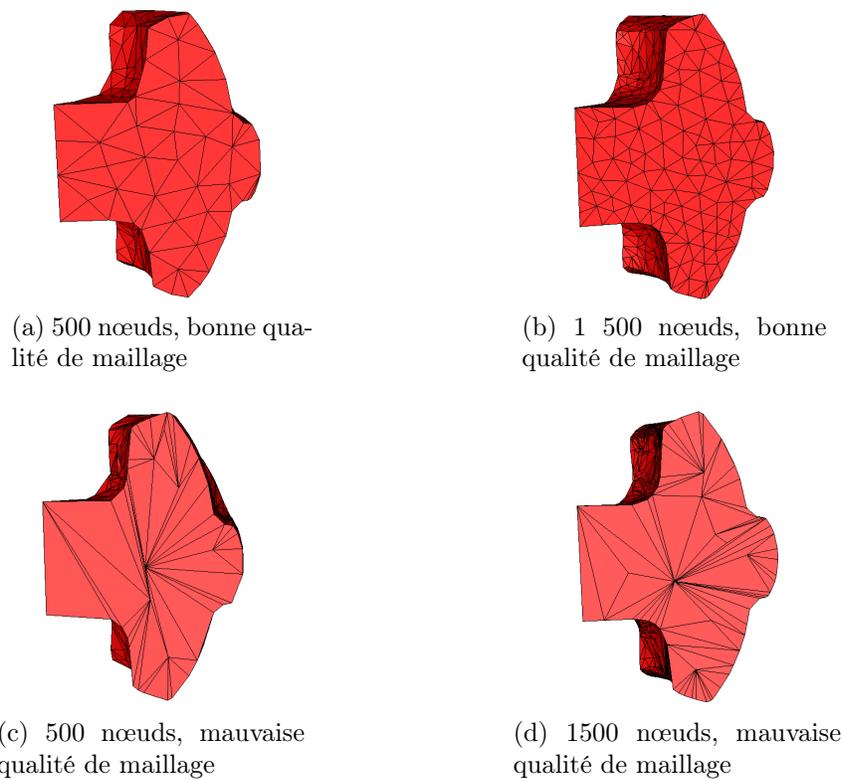


FIGURE 3.9 – Maillages grossiers de différent nombre de noeuds et différente qualité pour la reprise de la simulation de forgeage du triaxe

Les résultats présentés sur la figure 3.10 montrent l'évolution du temps de calcul en fonction des incréments de temps pour ces grilles grossières de 500 et 1 500 nœuds et des maillages de bonne et moins bonne qualité.

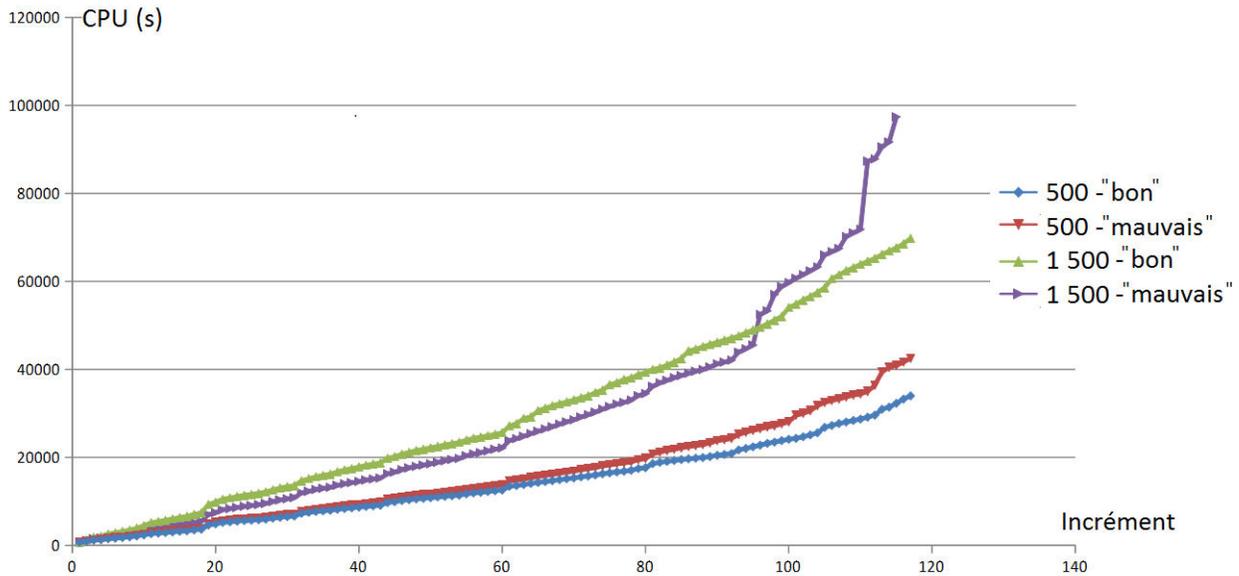


FIGURE 3.10 – Temps de calculs cumulés en fonction des incréments de temps lors de la reprise d'une simulation du forgeage du triaxe en utilisant une méthode multigrille avec des grilles grossières à 500 et 1 500 nœuds, de bonne et mauvaise qualité

Il apparait clairement que les meilleurs résultats sont obtenus avec les maillages grossiers comportant le moins de nœuds, qu'ils soient de bonne ou de mauvaise qualité. Comme attendu, le maillage de bonne qualité donne un temps de calcul plus faible. Le temps gagné en effectuant la factorisation et la résolution sur une grille grossière comportant moins de nœuds compense donc celui passé dans les itérations supplémentaires rendues nécessaires du fait de la mauvaise qualité du maillage pour obtenir la convergence.

En revanche, un maillage de mauvaise qualité, même avec suffisamment de nœuds (1 500) conduit à l'absence de convergence de la méthode multigrille en fin de simulation. La correction grille grossière est alors probablement erronée tout en étant plus directement et plus précisément transmise aux grilles supérieures.

Enfin, le maillage grossier de bonne qualité comportant 1 500 nœuds présente la même évolution du temps de calcul que le maillage grossier de bonne qualité à 500 nœuds, mais est pénalisé par le temps nécessaire pour les factorisations et résolutions sur la grille grossière.

Nous concluons ici sur l'importance d'utiliser des grilles grossières ayant suffisamment peu de nœuds, et secondairement sur celle de construire des maillages de qualité. Dans la suite, nous nous efforcerons de construire des grilles grossières ayant environ 500 nœuds, qui apparait comme le meilleur compromis entre la capacité du mailleur à mailler des géométries complexes avec des éléments de qualité et l'efficacité résultante de la méthode multigrille.

### 3.3 Extensions de la méthode multigrille

Nous terminons ce chapitre par l'utilisation de la méthode multigrille hybride pour résoudre des systèmes linéaires soit en tant que solveur soit en tant que préconditionneur d'un autre solveur plus général.

#### 3.3.1 GMRES + Multigrille

Le préconditionneur multigrille a été associé jusqu'à maintenant à la méthode du Résidu Conjugué, particulièrement adaptée pour la résolution de systèmes symétriques et en particulier ceux obtenus dans FORGE® NxT. Afin d'étudier l'intérêt que pourraient présenter les multigrilles sur d'autres types de systèmes linéaires, nous nous intéressons ici à une méthode de résolution plus générale, la méthode généralisée de minimisation du résidu (GMRES). Proposée par (Saad et Schultz, 1986), elle s'apparente à une généralisation de la méthode du résidu conjugué qui s'applique aux matrices non symétriques.

Nous reprenons les cas issus des simulations d'écrasement entre tas plats et de forgeage du triaxe en milieu de simulation (section 2.5). Nous comparons les temps de calcul obtenus avec la méthode GMRES préconditionnée par une méthode multigrille (GMRES + MG) avec ceux obtenus en utilisant un préconditionneur ILU1 (GMRES + ILU1), considéré comme étant le plus performant pour les problèmes étudiés. Nous supposons que les conclusions établies précédemment pour définir le préconditionneur multigrille de cette étude restent valables :

- Solveur direct LU sur la grille grossière
- Lisseur PR + ILU0 avec 1 itération de pré et post lissage
- Facteurs de déraffinement homogènes
- Grille grossière proche de 500 nœuds

A partir d'un maillage initial de 22 059 nœuds pour le tas plat (figure 3.11a), nous construisons un niveau intermédiaire à 3 773 nœuds (figure 3.11b) et un niveau grossier à 479 nœuds (figure 3.11c). Le tableau 3.5 présente les temps de calcul obtenus avec les méthodes GMRES + ILU1 et GMRES + MG, mais aussi avec la méthode du résidu conjugué préconditionné

par ILU1 (CR + ILU1) et par la méthode multigrille (CR + MG). Les accélérations obtenues figurent dans la dernière colonne du tableau.

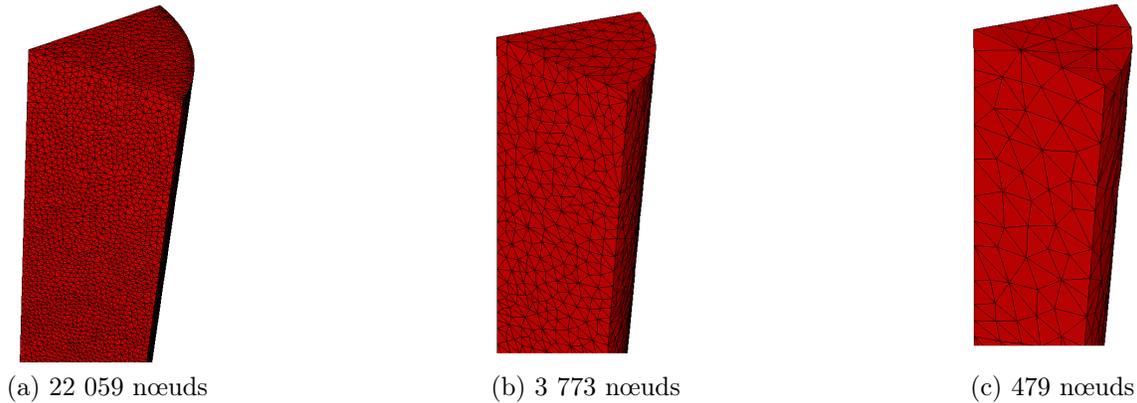


FIGURE 3.11 – Différents niveaux de maillages pour le cas d'écrasement entre tas plats, (a) maillage fin, (b) maillage intermédiaire et (c) maillage grossier

	CPU (s)	Accélération PCMG
CR + ILU1	11,18	
CR + MG	9,62	1,2
<b>GMRES + ILU 1</b>	13,67	
<b>GMRES + MG</b>	11,01	1,2

TABLE 3.5 – Comparaison des méthodes de résolution GMRES et CR préconditionnées par multigrille ou par ILU1 sur le cas d'écrasement entre tas plats

Tout d'abord, et comme nous pouvions nous y attendre, à préconditionneur identique, les temps de calculs obtenus avec la méthode du résidu conjugué sont plus faibles que ceux obtenus avec la méthode GMRES. Le point important est l'accélération apportée par le changement de préconditionneur (entre CR+ILU1 et CR+MG et entre GMRES+ILU1 et GMRES+MG). Cette accélération de 1,2 est parfaitement identique pour les deux méthodes de résolution. Nous pouvons en conclure que les résultats obtenus avec le préconditionneur multigrille dépendent peu du solveur utilisé. Les observations faites sur le résidu conjugué peuvent donc raisonnablement s'étendre à d'autres solveurs, notamment à GMRES, ce qui permet d'envisager d'utiliser ce préconditionneur sur des problèmes non symétriques.

La même étude est réalisée sur le problème du triaxe. Le maillage initial comporte 37 312 nœuds (figure 3.12a) et les deux maillages générés en comportent 4 534 pour l'intermédiaire et 500 pour le grossier (figures 3.12b et 3.12c).

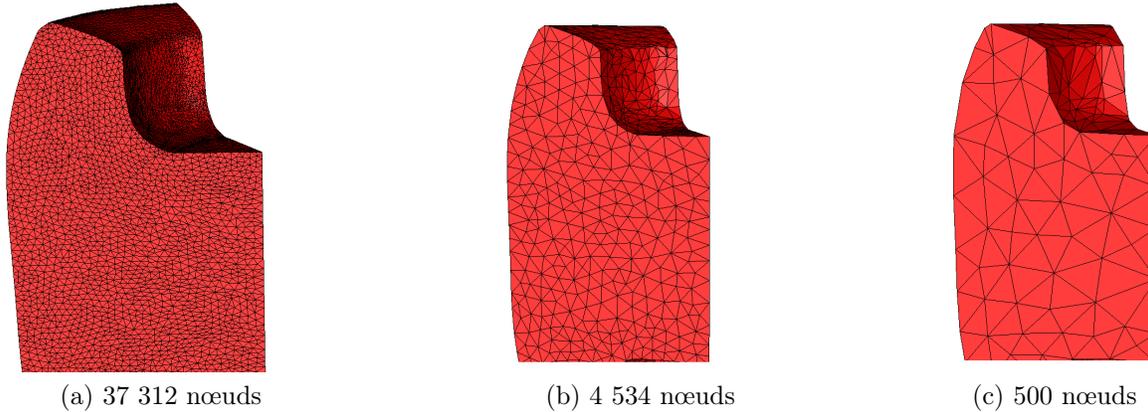


FIGURE 3.12 – Différents niveaux de maillages pour le cas de forgeage du triaxe, (a) maillage fin, (b) maillage intermédiaire et (c) maillage grossier

Les temps de calcul figurant dans le tableau 3.6 correspondent toujours à la résolution d'un incrément de temps comportant 12 itérations de Newton-Raphson.

	CPU (s)	Accélération PCMG
CR + ILU1	552	
CR + MG	390	1,4
<b>GMRES + ILU 1</b>	646	
<b>GMRES + MG</b>	437	1,5

TABLE 3.6 – Comparaison des méthodes de résolution GMRES et CR préconditionnés par multigrille ou par ILU sur le cas de forgeage du triaxe

Les conclusions sont identiques à celles tirées sur le cas d'écrasement entre tas plats, sinon que les accélérations apportées par la méthode multigrille sont encore meilleures avec le solveur GMRES (1,5 au lieu de 1,4).

Pour les problèmes considérés, la méthode du résidu conjugué étant plus efficace, nous continuerons notre étude avec cette seule méthode, tout en considérant que les résultats obte-

pus peuvent raisonnablement s'étendre à la méthode GMRES, et permettent ainsi d'envisager l'utilisation de ce préconditionneur multigrille dans d'autres domaines d'applications.

### 3.3.2 Solveur Multigrille

Pour terminer ce chapitre, nous étudions l'utilisation de la méthode multigrille en tant que solveur. Les motivations derrière ce développement étaient pour partie de se libérer de la "boite noire" multigrille de PETSc, afin notamment d'utiliser un critère de précision pour contrôler le nombre d'itérations de lissage. D'autre part, nous souhaitions comparer les performances des multigrilles en tant que solveur et préconditionneur sur nos problèmes.

Pour développer ce solveur multigrille, nous utilisons toujours la librairie PETSc et l'ensemble de ses fonctions :

- Méthodes itératives de résolution pour les lissages
- Multiplication matrice-vecteur pour le transfert entre les grilles
- Multiplication matrice-matrice pour la construction des matrices grilles grossières

Nous utilisons également la librairie MUMPS pour la résolution directe sur la grille grossière.

Nous utilisons une nouvelle fois les problèmes d'écrasement entre tas plats et de forgeage d'un triaxe. Nous nous plaçons dans la même configuration que celle utilisée pour l'étude du lisseur du préconditionneur, en considérant une méthode bigrille idéale. Pour l'écrasement entre tas plats, le maillage fin comporte 22 059 nœuds et le maillage grossier 902 nœuds (figures 3.6a et 3.6b). Pour le forgeage du triaxe en milieu de simulation, le maillage fin comporte 16 902 nœuds (figure 3.7a) et le maillage grossier 936 nœuds (figure 3.7b).

Avec le solveur multigrille, une plus large variété de lisseurs a pu être testée. En plus des lisseurs Richardson préconditionnés par ILU0, Jacobi ou SSOR préalablement testés sur le préconditionneur (PR+ILU0, PR+Jacobi et PR+SSOR), nous avons considéré le résidu conjugué en tant que lisseur associé aux préconditionneurs ILU0, ILU1 et Jacobi (CR+ILU0, CR+ILU1, CR+Jacobi).

Le tableau 3.7 présente les meilleurs temps de calculs obtenus avec chacun de ces 6 lisseurs pour la résolution d'un système linéaire issu de l'écrasement entre tas plats. Il résulte d'essais menés avec un nombre d'itérations de pré et post lissages allant de 1 à 5. Le nombre total d'itérations de lissage figure également dans ce tableau. Pour rappel, le temps de résolution obtenu avec le préconditionneur multigrille dans sa configuration quasi optimale (lisseur PR+ILU0 (1,1)) est de 9,59 secondes.

Le premier résultat marquant est que pour tous les lisseurs, le meilleur résultat est obtenu

Lisseur	Meilleure combinaison – Nb Itérations de lissage cumulé	CPU (s)
PR + ILU0	(1,5) – 108 itérations	9,03
PR + Jacobi	(1,4) – 150 itérations	8,09
PR + SSOR	(1,5) – 126 itérations	10,95
CR + ILU0	(1,5) – 66 itérations	7,1
CR + ILU1	(1,5) – 72 itérations	14,04
CR + Jacobi	(2,4) – 132 itérations	9,63

TABLE 3.7 – Temps de calcul pour la résolution d’un système linéaire issu de la simulation d’un écrasement entre tas plats par le solveur multigrille avec différents lisseurs

avec une combinaison demandant plus d’itérations de post lissage que de pré-lissage. Dans son utilisation en tant que solveur, le post-lissage joue un rôle plus important et permet de limiter le nombre de cycles multigrilles. Comme dans son utilisation en tant que préconditionneur avec lisseur de type Richardson, le lisseur Résidu Conjugué préconditionné par ILU1 donne des temps de calcul largement supérieurs aux autres, à cause d’un coût de factorisation et de résolution trop élevés et malgré un nombre d’itérations assez faible. Le lisseur SSOR donne de moins bons résultats que les autres lisseurs, comme observé dans l’utilisation des multigrilles en tant que préconditionneur. Le solveur utilisant le lisseur CR+ILU0 se révèle étonnamment plus efficace, et présente des résultats bien meilleurs que ceux obtenus en préconditionnement (7,1 s au lieu de 9,59 s).

Le solveur multigrille autorise de contrôler la diminution du résidu au cours des itérations de lissage comme proposé dans (Mocellin, 1999). Chaque étape de pré et post lissage est alors assujettie à la diminution du résidu par un facteur  $\varepsilon$ . Le critère d’arrêt  $C$  est donné par :

$$C = \max(\varepsilon R_i, C_{glob})$$

avec  $R_i$  le résidu à l’entrée de l’étape de lissage et  $C_{glob}$  le critère d’arrêt global de la résolution linéaire. Nous reprenons notre étude avec les 4 lisseurs ayant donné les meilleurs résultats, CR+ILU0, CR+Jacobi, PR+ILU0 et PR+Jacobi, et avec ce critère de précision pour des valeurs de  $\varepsilon$  égales à 0,1, 0,2 et 0,5. Les temps de calcul figurent dans le tableau 3.8. Pour les lisseurs PR+ILU0 et PR+Jacobi, les résultats obtenus avec  $\varepsilon = 0,1$  n’ont pas été calculés compte tenu de la forte augmentation du temps de calcul déjà présente en passant de  $\varepsilon = 0,5$  à 0,2. Ce tableau présente également le nombre total d’itérations de lissage nécessaire pour atteindre la convergence finale.

Pour tous les lisseurs et pour toutes les précisions étudiées, le nombre total d’itérations de

Lisseur	$\epsilon$					
	0,1		0,2		0,5	
	CPU (s)	Lissage cumulé	CPU (s)	Lissage cumulé	CPU (s)	Lissage cumulé
PR + ILU0	Non testé		16	351 it	11,37	209 it
PR + Jacobi			22,42	1100 it	13,29	550 it
CR + ILU0	8,37	132 it	8,27	123 it	8,17	107 it
CR + Jacobi	10,12	400 it	10,68	396 it	10,6	372 it

TABLE 3.8 – Temps de calcul pour la résolution d’un système linéaire issu de la simulation d’un écrasement entre tas plats par le solveur multigrille avec différents lisseurs contrôlés par la diminution de résidu

lissage est supérieur à ce qui a été obtenu en fixant le nombre de pas de lissage : 107 au mieux, au lieu de 66 pour CR+ILU0. Cela montre une meilleure efficacité de la méthode multigrille lorsque le résidu diminue progressivement, au fur et à mesure des cycles, quitte à augmenter le nombre de cycles et à bénéficier ainsi de davantage de corrections provenant de la grille grossière. Les lisseurs utilisant la méthode de Richardson se montrent particulièrement sensibles à cela, avec une forte augmentation du nombre total de lissages lorsque le critère de précision passe de 0,5 à 0,2. Les lisseurs utilisant la méthode de résidu conjugué y sont moins sensibles ; la différence des temps de calcul est bien moins significative. Le meilleur temps de calcul est obtenu avec le lisseur CR+ILU0 et un CPU de 8,17 secondes. La méthode est moins efficace qu’avec le réglage en nombre de pas de lissages fixé (7,1 secondes) mais reste plus efficace que lorsqu’elle est utilisée en préconditionnement. Nous n’avons pas poursuivi avec ce pilotage en précision qui s’avère au mieux un peu moins efficace qu’en nombre d’itérations.

Pour terminer ce chapitre, nous évaluons les performances du solveur multigrille sur le cas de forgeage du triaxe en milieu de simulation. Nous ne considérons que les lisseurs pour lesquels les meilleurs temps de calcul ont été obtenus. Le tableau 3.9 présente ainsi les résultats obtenus pour la résolution d’un incrément de temps avec les lisseurs CR+ILU0 et PR+ILU0. Les lisseurs PR+Jacobi et CR+Jacobi ont également été testés mais nous n’avons pas réussi à faire converger le solveur multigrille résultant, quelque soit la combinaison de pré et post lissage testée. Le temps de calcul obtenu avec le préconditionneur y est indiqué pour rappel.

Sur ce système, le meilleur temps de calcul est obtenu avec le lisseur PR+ILU0 et la combinaison de pré et post lissage (1,3). Le temps de calcul résultant du CR+ILU0 en est toutefois assez proche, d’autant que la résolution de l’incrément de temps avec ce lisseur a demandé une itération de Newton-Raphson supplémentaire (13 au lieu de 12). Les chemins de convergence de l’algorithme non linéaire pouvant légèrement varier en fonction de la méthode de

Lisseur	Meilleure combinaison – CPU (s)
<b>PR + ILU0</b>	(1,3) – 193 s
<b>CR + ILU0</b>	(1,4) - 214 s
<b>Résultat en tant que réconditionneur</b>	
<b>CR + PCMG</b>	Lisseur PR+ILU0 (1,1) - 128 s

TABLE 3.9 – Temps de calcul pour la résolution d’un incrément de temps issu de la simulation du forgeage d’un triaxe en milieu de simulation, en utilisant le solveur multigrille avec deux lisseurs différents

résolution choisie, il n’est pas exclu, devant la faible différence de temps, que la hiérarchie s’inverse sur un autre incrément de temps. L’écart avec le temps de calcul obtenu en utilisant les multigrilles en tant que préconditionneur est en revanche beaucoup plus important. Ce dernier est, cette fois, bien plus efficace que le solveur (193s contre 128s en préconditionneur).

Le solveur multigrille développé s’est montré plus efficace sur la résolution d’un problème "simple" comme l’écrasement entre tas plats, mais est devancé par son utilisation en tant que préconditionneur sur le cas du triaxe qui est toutefois plus représentatif des problèmes rencontrés dans les applications de mise en forme. Son étude mériterait d’être approfondie mais dans cette thèse, nous retenons le choix d’utiliser la méthode multigrille en tant que préconditionneur, comme il est également d’usage ces dernières années dans la littérature.

### 3.4 Conclusion

Dans ce chapitre, nous avons présenté les éléments de la méthode multigrille hybride développée pour résoudre les systèmes linéaires issus des problèmes de mise en forme. Cette méthode est compatible avec le calcul parallèle : construction des niveaux de maillages grossiers, construction des matrices de Galerkin, résolution grille grossière et lisseur. L’utilisation des multigrilles en tant que préconditionneur d’un solveur résidu conjugué apparaît comme mieux adaptée pour les problèmes visés. De nombreux tests nous ont permis de déterminer les meilleurs paramètres de ce préconditionneur multigrille. Ainsi, nous avons retenu l’utilisation d’un lisseur Richardson préconditionné par ILU0 avec une seule itération de pré et post lissage. La grille grossière doit avoir une taille aussi proche que possible de 500 nœuds. Ce critère de nombre de nœuds l’emporte sur celui de la qualité du maillage produit sans pour autant l’exclure. Pour une méthode à 3 grilles, les facteurs de déraffinement entre les différents niveaux doivent être homogènes.

Le dernier chapitre de ce manuscrit porte sur une étude de performance du préconditionneur multigrille ainsi défini. Nous vérifierons tout d’abord son comportement asymptotique. Nous

---

nous intéresserons ensuite à son comportement parallèle, l'ensemble des cas de ce chapitre ayant été traité en séquentiel, avant de terminer sur l'étude de cas d'applications permettant d'évaluer son efficacité sur des cas de forgeage plus complexes.



# 4 | Etude de performances et applications

J'ai refait tous les calculs. Ils confirment l'opinion des spécialistes : notre idée est irréalisable. Il ne nous reste qu'une seule chose à faire : la réaliser.

---

Pierre-Georges Latécoère

Dans ce dernier chapitre, nous évaluons les performances du préconditionneur multigrille hybride développé avec les paramètres décrits dans le chapitre précédent. Seuls les algorithmes 2 grilles et 3 grilles ont été programmés. Dans la première partie, nous commençons par l'étude de la convergence asymptotique de ces deux algorithmes. Les performances en parallèle sont ensuite mesurées sur l'algorithme 3 grilles, plus pertinent pour traiter les problèmes présentant un grand nombre de degrés de liberté voués à être résolus sur plusieurs processeurs. Nous terminons ce chapitre par des cas d'applications de mise en forme afin d'évaluer l'accélération concrètement obtenue.

**Sommaire**

---

<b>4.1</b>	<b>Etude de performances</b>	<b>75</b>
4.1.1	Complexité asymptotique	75
4.1.2	Calcul parallèle	80
<b>4.2</b>	<b>Applications</b>	<b>84</b>
4.2.1	Forgeage du Triaxe	84
4.2.2	Laminage retour	92
4.2.3	Martelage rotatif	97
<b>4.3</b>	<b>Conclusion</b>	<b>103</b>

---

## 4.1 Etude de performances

Avant d'évaluer si le préconditionneur multigrille permet une accélération des calculs sur différents problèmes de mise en forme, nous commençons par étudier la complexité asymptotique de la méthode. Nous évaluons ensuite ses performances dans le contexte du calcul parallèle.

### 4.1.1 Complexité asymptotique

Cette première vérification concerne le comportement de la méthode multigrille en fonction de l'évolution de la taille du problème lorsque celui-ci tend vers l'infini, c'est à dire son comportement asymptotique. Pour cela, nous utilisons la méthode en séquentiel. En tant que solveur, dans la configuration Full Multigrid (section 1.1.3), sa complexité asymptotique théorique optimale est linéaire, en  $O(N)$ ,  $N$  étant le nombre d'inconnues du système. Plus généralement et en tant que préconditionneur, elle est en  $O(N \log N)$ . Cette propriété est la principale motivation de ces travaux en permettant d'envisager des problèmes de mise en forme de tailles beaucoup plus importantes qu'aujourd'hui.

Pour évaluer cette propriété, nous considérons une nouvelle fois la même simulation d'écrasement d'un lopin cylindrique entre tas plats (section 2.5.1) et la résolution d'un système linéaire issue de celle-ci. Nous utilisons des maillages de tailles croissantes, de 22 000 à 160 000 nœuds (table 4.1), et utilisons une grille grossière à 500 nœuds et un facteur de déraffinement homogène pour construire la grille intermédiaire pour la méthode 3 grilles. Pour les plus petites tailles du problème, la méthode deux grilles présente de meilleures performances, la méthode trois grilles devenant plus performante sur les maillages plus importants. Nous évaluons ainsi ces deux algorithmes (PCR+2G et PCR+3G), mais également le solveur de référence, le résidu conjugué préconditionné par ILU1 (PCR+ILU1). Sa complexité asymptotique est habituellement estimée être en  $O(N^{\frac{3}{2}})$ . Les temps de calcul et les accélérations obtenus figurent dans le tableau 4.1.

Pour toutes les tailles de maillages considérées, les temps de calcul des deux algorithmes multigrilles sont inférieurs à ceux du solveur de référence. L'algorithme deux grilles offre une accélération de 1,7 dès la simulation avec le maillage de 20 000 nœuds. Avec le préconditionneur 3 grilles, elle n'est que de 1,1. Ce préconditionneur devient véritablement plus intéressant à partir de 80 000 nœuds. Avec 120 000 nœuds, l'accélération est alors de 2,6. Elle semble stagner en passant au maillage à 160 000 nœuds. Ce phénomène est lié à la mémoire disponible en séquentiel sur un problème de cette taille. Les différentes matrices nécessaires à l'algorithme multigrille requièrent en effet plus de ressources mémoire que le solveur de

Nombre de nœuds	CPU PCR+ILU1(s)	CPU PCR+2G(s)	Accélération 2G	CPU PCR+3G(s)	Accélération 3G
22 059	11,1	6,4	1,73	9,6	1,16
46 622	31,9	15,9	2,01	19,2	1,66
81 244	70,6	36,3	1,94	34,5	2,05
119 772	146	61,6	2,37	55,5	2,63
162 509	232,2	99,0	2,35	88,3	2,63

TABLE 4.1 – Temps de calcul des méthodes PCR+ILU1, PCR+2G et PCR+3G et accélérations multigrilles pour la résolution d'un système linéaire issu de la simulation d'écrasement entre tas plats utilisant des maillages de 22 059 à 162 509 nœuds

référence PCR+ILU1. 160 000 nœuds, qui correspond à près de 650 000 degrés de liberté, semble être cette limite à partir de laquelle le calculateur a besoin de gérer autrement sa mémoire avec le préconditionneur multigrille mais pas encore avec ILU1.

Sur la figure 4.1, nous traçons l'évolution des temps de calcul en fonction du nombre de nœuds ainsi que les courbes de tendance correspondantes qui nous donnent la complexité asymptotique.

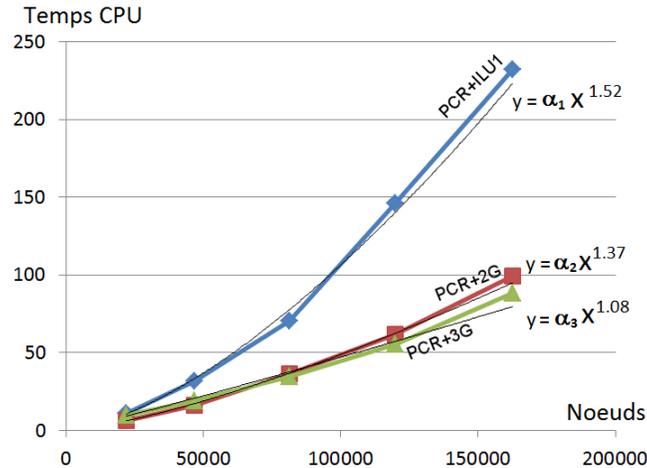


FIGURE 4.1 – Courbes de complexité asymptotique avec les courbes de tendances associées pour les méthodes PCR+ILU1, PCR+2G et PCR+3G lors de la résolution d'un système linéaire issu de la simulation d'un écrasement entre tas plats utilisant des maillages de 22 059 à 162 509 nœuds

Nous remarquons tout d'abord que le comportement asymptotique de la méthode PCR+ILU1

est celui attendu avec une courbe de tendance en  $N^{1.52}$ . Le préconditionneur 3 grilles montre un comportement quasi-linéaire en  $N^{1.08}$ . Ce résultat va de pair avec les excellentes accélérations obtenues sur les maillages les plus fins. Ce comportement nous assure que le préconditionneur multigrille demeure plus efficace que le solveur de référence lorsque le nombre de nœuds augmente et que les accélérations sont de plus en plus grandes. La méthode 2 grilles montre un comportement asymptotique qui pourrait sembler décevant en  $N^{1.37}$  mais qui s'explique parfaitement par le fait que l'on ne peut pousser la complexité asymptotique sans passer à la méthode 3 grilles à partir de 80 000 nœuds. Pour des maillages relativement grossiers, les temps de calcul de la méthode 2 grilles sont proches de ceux du 3 grilles et même significativement inférieurs pour les plus grossiers.

Cette étude souligne la nécessité d'adapter le nombre de grilles en fonction de la taille du problème à résoudre. Dans cette étude, nous nous sommes arrêtés à 3 grilles, mais une manière plus correcte d'effectuer cette analyse de comportement asymptotique, et plus conforme à la pratique, est d'ajouter des grilles au fur et à mesure de l'augmentation de la taille du problème, en suivant une véritable stratégie multigrille. Les méthodes 2 et 3 grilles étant les seules implémentées à ce jour dans le logiciel FORGE® NxT, cette stratégie n'y a pas pu être suivie. Nous présentons néanmoins une étude de ce type réalisée en début de thèse, sur une version préliminaire du préconditionneur multigrille, au sein de la librairie CimLib. Les systèmes traités restent cependant proches de ceux présentés dans le chapitre 2, utilisant la même formulation et ayant les mêmes caractéristiques.

Le point clé de cette étude asymptotique par rapport à la précédente est que chaque nouveau maillage contient 8 fois plus de nœuds et justifie donc l'emploi d'une grille supplémentaire. Cette étude est difficile à étendre sur des géométries complexes mais elle permet d'évaluer le comportement asymptotique de la méthode multigrille de manière plus rigoureuse.

Bien que les stratégies de construction des opérateurs et des systèmes linéaires sur les grilles inférieures soient celles présentées en 3.1, les paramètres de la méthode diffèrent de ceux définis dans la section 3.2. Toutefois, nous avons vu en 3.3 que nous pouvons être relativement confiants quant à la transposition de ces résultats sur notre méthode actuelle.

Ici, la méthode multigrille est utilisée en tant que préconditionneur d'un algorithme GMRES. Le lisseur utilisé est un autre GMRES préconditionné par ILU0, avec 3 itérations de pré-lissages et post-lissages. La méthode de résolution sur la grille la plus grossière reste une méthode directe.

La simulation testée est celle de l'écrasement d'un cube unité entre tas plats, dont le schéma simplifié est présenté sur la figure 4.2. Un contact collant est imposé entre la pièce et les

outils et nous nous intéressons à la résolution du premier système linéaire.

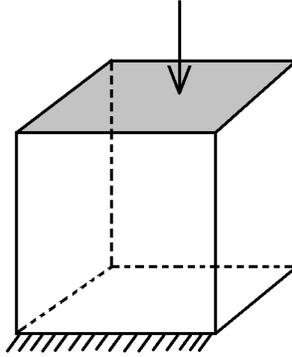


FIGURE 4.2 – Ecrasement d'un cube unité entre tas plat

Les différents niveaux de maillages ont été construits en respectant autant que possible une division de la taille de maille par 2, correspondant à une division du nombre de nœuds par 8. Ainsi, à partir d'un maillage fin comportant 867 889 nœuds, nous obtenons des grilles comportant 323, 2 089, 14 953 et 108 684 nœuds (figure 4.3).

La stratégie de cette étude consiste à partir d'une méthode bigrille, puis à ajouter une grille plus fine à chaque étape de raffinement. Nous avons ainsi successivement :

- Une méthode bigrille avec un maillage fin à 2 089 nœuds et un maillage grossier à 323 nœuds
- Une méthode 3 grilles avec un maillage fin à 14 953 nœuds, un maillage intermédiaire à 2 089 nœuds et un maillage grossier à 323 nœuds
- Une méthode 4 grilles avec un maillage fin à 108 684 nœuds, deux maillages intermédiaires de 14 953 et 2 089 nœuds et un maillage grossier à 323 nœuds
- Une méthode 5 grilles, avec un maillage fin à 867 889 nœuds, 3 maillages intermédiaires de 108 684, 14 953 et 2 089 nœuds et un maillage grossier à 323 nœuds

Les temps de calculs relevés permettent de distinguer les différentes étapes de l'utilisation de la méthode multigrille, à savoir :

- Assemblage du système, noté 'Assemblage'
- Construction des matrices de transfert, noté 'Mat Passage'
- Construction des matrices des grilles intermédiaires et grossières, noté 'Mat Galerkin'
- Initialisations et lectures des paramètres, noté 'Initialisation'

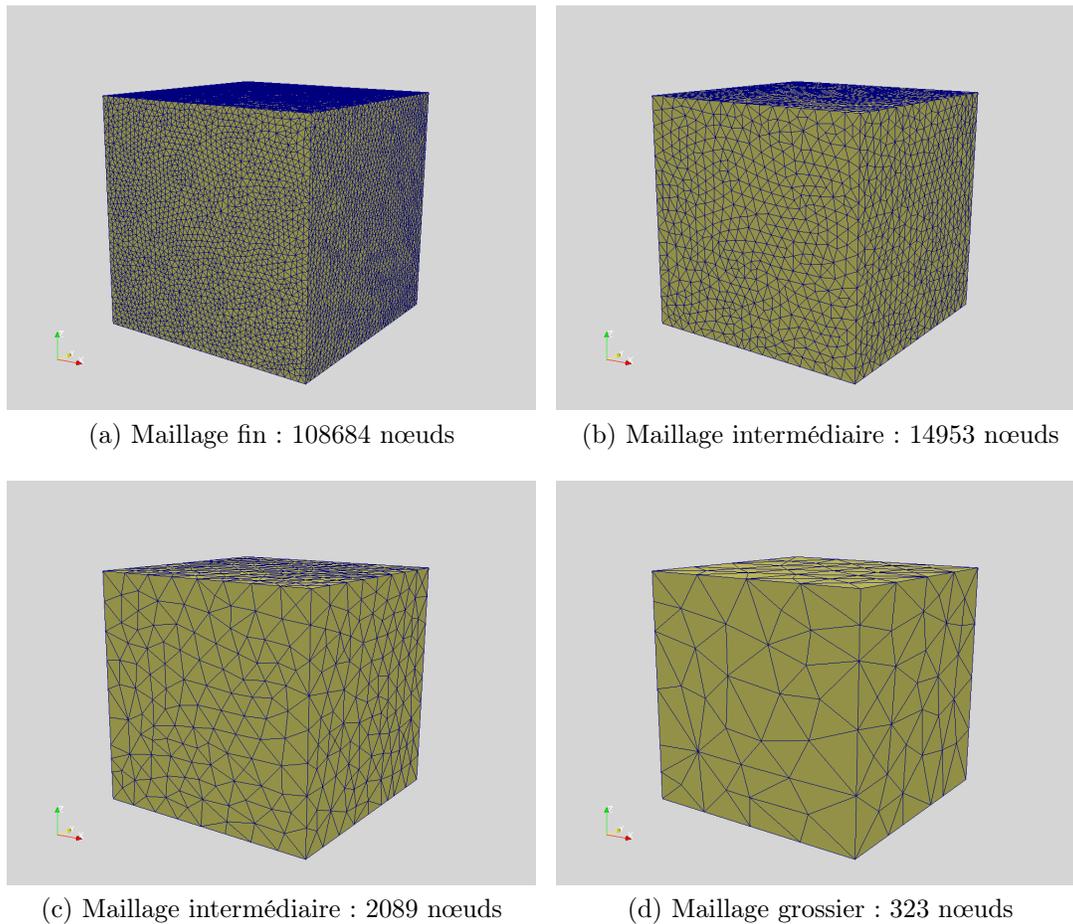


FIGURE 4.3 – Maillages du cas d'écrasement entre tas plats

- Factorisations sur les différentes grilles (ILU et LU sur la grille grossière) et calcul du résidu initial, noté 'Facto + RI'
- Résolution du système, noté 'Résolution'

La figure 4.4 présente sur une échelle logarithmique les temps de calculs de chacune de ces étapes, et le temps de calcul total, en fonction du nombre de nœuds de la grille la plus fine.

L'ensemble des courbes est superposable à la droite  $y=x$ , montrant que chacune des étapes de la méthode multigrille présente le comportement linéaire recherché, sinon pour la méthode de résolution qui se montre un peu moins performante. Ce comportement résulte de

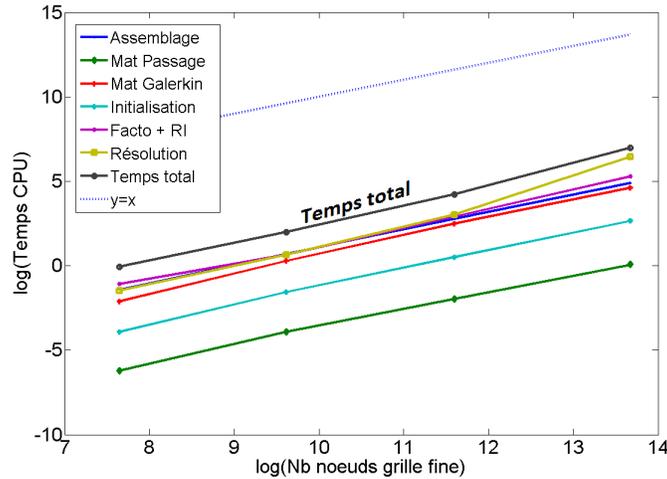


FIGURE 4.4 – Temps de calcul total et des différentes étapes de la méthode multigrille en fonction du nombre de nœuds de la grille fine en échelle logarithmique

la stratégie de Galerkin adoptée. L’augmentation du nombre de niveaux de grilles entraîne le remplissage des matrices des niveaux les plus grossiers. Elles sont de moins en moins creuses et donc leur produit ou résolution directe devient plus coûteux même si leur taille est inchangée. Comme évoqué dans la section 4.1.1, un effet mémoire peut également concourir à cette baisse de performance compte tenu du très grand nombre de nœuds considéré ici.

Ces résultats prouvent néanmoins que la méthode multigrille hybride développée dans cette thèse reproduit les performances en terme de convergence asymptotique quasi-linéaire attendues, et motive le développement d’une méthode 4 grilles au sein de FORGE® NxT afin de traiter efficacement des problèmes comportant encore plus de degrés de liberté.

### 4.1.2 Calcul parallèle

Nous considérons de nouveau la méthode multigrille avec les paramètres définis dans le chapitre précédent. Pour l’évaluer sur des calculs parallèle, nous reprenons le problème d’écrasement entre tas plats, présentés en section 2.5.1, et considérons la résolution d’un système linéaire issu de cette simulation. Pour la pertinence de cette étude, nous utilisons un maillage initial très fin, comportant 310 000 nœuds. Les systèmes à résoudre comportent donc plus de 1,2 millions de degrés de liberté.

L’efficacité parallèle de la méthode multigrille repose sur un équilibre des charges entre

les différents processeurs qui passe par la construction de partitions de tailles équivalentes sur chacun des niveaux de maillage. De plus, et afin de limiter les communications, il est important que les partitions soient relativement consistantes entre ces différents niveaux, c'est-à-dire que la localisation de ces partitions soit conservée lors du déraffinement. A titre d'exemple, la figure 4.5a présente les partitions du maillage de calcul utilisé sur 4 processeurs et indique le nombre de nœuds pour chacun des processeurs. Les figures 4.5b et 4.5c montrent les mêmes informations sur les maillages intermédiaire et grossier.

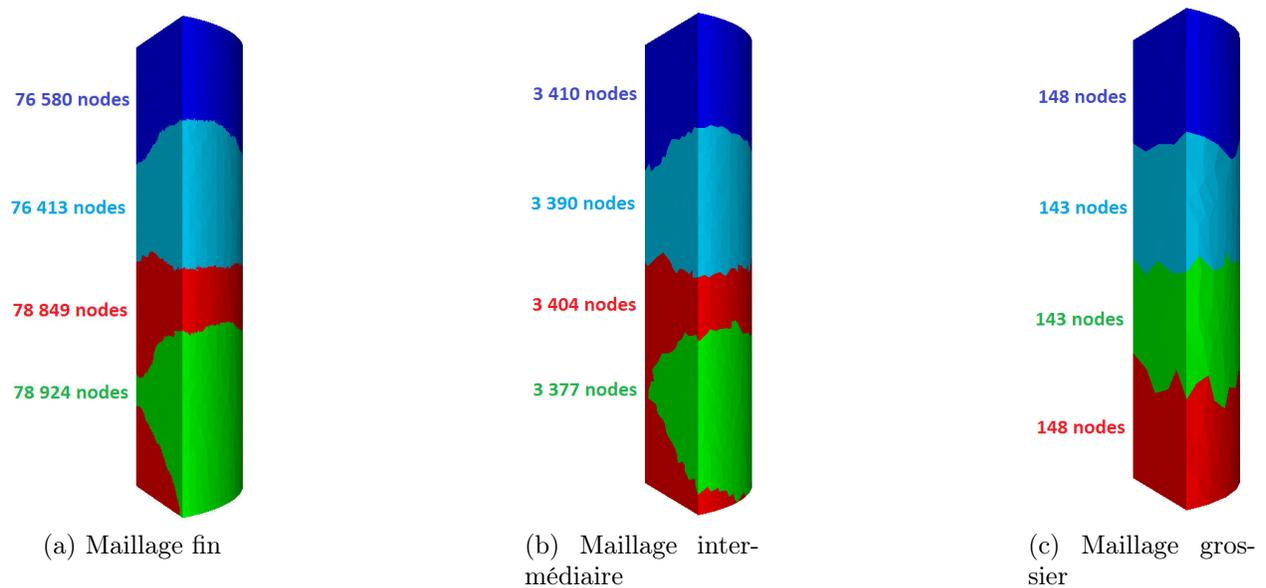


FIGURE 4.5 – Partitions sur 4 processeurs des 3 maillages utilisés par le préconditionneur 3 grilles pour la résolution d'un système linéaire issu de la simulation de l'écrasement d'un lopin cylindrique entre tas plats

Pour les trois maillages, les partitions sont équitablement réparties sur les 4 processeurs, avec, dans tous les cas, moins de 1% d'écart sur le nombre total de nœuds entre les différentes partitions. La conservation des différents niveaux sur une même partition est correctement réalisée, bien qu'imparfaite pour les partitions rouge et verte sur le maillage le plus grossier.

Les calculs utilisent de 1 à 8 processeurs. Ayant observé une accélération multigrille en séquentiel dans la section précédente 4.1.1, nous cherchons ici à obtenir des performances similaires sur plusieurs processeurs. Dans le tableau 4.2 figurent les temps de calcul relevés, ainsi que les accélérations multigrilles pour chacune des simulations.

Nb Procs	CPU		Accélération
	PCR+ILU1 (s)	PCR+3G (s)	
1	900.5	335.3	2.7
2	253.9	137.6	1.8
4	115.1	58.0	2.0
6	83.0	42.8	1.9
8	56.6	30.7	1.8

TABLE 4.2 – Accélération multigrille et temps de calcul du PCR+ILU1 et du PCR+3G pour la résolution d'un système linéaire issu de la simulation d'écrasement d'un lopin entre tas plats sur un maillage à 310 000 nœuds, en utilisant de 1 à 8 processeurs.

Nous pouvons tout d'abord noter que le préconditionneur multigrille apporte une accélération significative en séquentiel comme en parallèle. Comme évoqué dans la section précédente, ce cas de plus de 300 000 nœuds subit l'effet conjugué du manque de mémoire (en séquentiel) et de la nécessité de recourir à un quatrième niveau de grille. L'accélération est de 2,7. Elle est supérieure mais proche de ce qui avait été obtenu avec 120 000 nœuds, et non supérieure à 3 comme on aurait pu l'espérer sur la base du comportement asymptotique. En passant au parallèle, cette accélération décroît mais nous observons une certaine constance dans sa valeur, avec une division du temps de calcul par presque 2, quelque soit le nombre de processeurs.

L'accélération parallèle (ou speed-up) permettant de mesurer l'efficacité parallèle d'un calcul se mesure de la manière suivante :

$$\text{Speed-up} = \frac{\text{Temps de calcul séquentiel}}{\text{Temps de calcul parallèle}} \quad (4.1)$$

Les courbes d'accélération parallèle des méthodes PCR+ILU1 et PCR+3G en fonction du nombre de processeurs sont présentées sur la figure 4.6, accompagnées de la courbe d'accélération "idéale"  $y = x$ .

On observe une hyper accélération des deux solveurs, mais surtout du PCR + ILU1. Cela s'explique par un effet de mémoire cache saturée en séquentiel et déjà évoqué précédemment. Ces courbes sont donc faussées par cet effet et il est beaucoup plus cohérent de prendre comme temps de référence le temps relevé sur 2 processeurs. De plus, les préconditionneurs incomplets ILU0 et ILU1 de PETSc utilisés par chacun des solveurs ont des comportements différents en séquentiel et en parallèle. En effet, la décomposition incomplète réalisée en parallèle ne donne pas la même matrice de factorisation qu'en séquentiel. Elle dépend donc du nombre de processeurs mais présente surtout de grandes différences entre sa valeur séquentielle et parallèle.

Nous définissons alors un speed-up alternatif, dont les courbes en fonction du nombre de

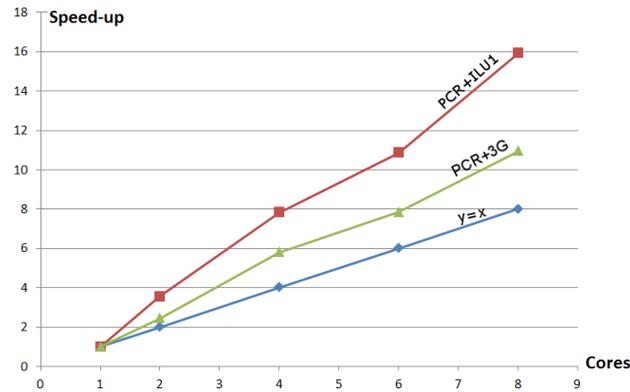


FIGURE 4.6 – Courbes d’accélération parallèle pour les méthodes PCR+ILU1 et PCR+3G pour la résolution d’un système linéaire issu de la simulation d’écrasement d’un lopin entre tas plats, sur un maillage à 310 000 nœuds et utilisant de 1 à 8 processeurs

processeurs sont présentées sur la figure 4.7 :

$$\text{Speed-up}_{2p} = 2 * \frac{\text{Temps de calcul sur 2 proc}}{\text{Temps de calcul parallèle}} \quad (4.2)$$

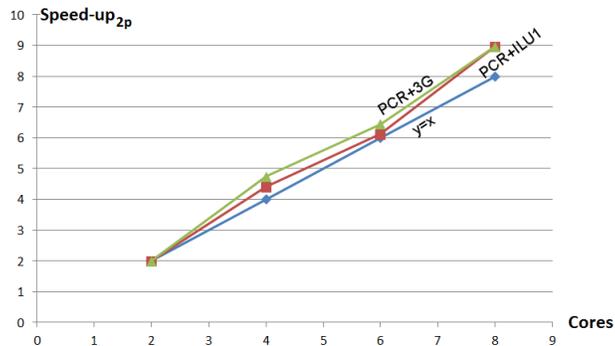


FIGURE 4.7 – Courbes d’accélération parallèle basées sur le temps de calcul sur 2 processeurs pour les méthodes PCR+ILU1 et PCR+3G pour la résolution d’un système linéaire issu de la simulation d’écrasement d’un lopin entre tas plats, sur un maillage à 310 000 nœuds et utilisant 1 à 8 processeurs

Une légère hyper-accélération est toujours visible pour les deux méthodes. L’accès à la mémoire cache permet toujours d’expliquer ce phénomène, si on estime que le problème à 310 000 nœuds partitionné sur deux processeurs équivaut à résoudre deux problèmes à 155 000

nœuds par processeur. Nous observons déjà cet effet mémoire avec 160 000 nœuds dans la section précédente 4.1.1. La figure 4.7 permet surtout de montrer un comportement parallèle quasiment identique avec les solveurs PCR+ILU1 et PCR+3G, mettant en relief la bonne parallélisation du PCR+3G et donc des différentes composantes de l’algorithme développé. La chute de l’efficacité de la méthode multigrille en parallèle s’explique donc davantage par la chute d’efficacité des préconditionneurs incomplets ILU en parallèle, et plus spécialement de ILU0 dont le préconditionneur multigrille fait un usage intensif durant toutes les itérations de lissage.

Ayant mis en évidence les possibilités du préconditionneur multigrille développé dans cette thèse pour la résolution de problèmes de grande taille en parallèle sur le cas simple d’écrasement entre tas plats, nous terminons ce manuscrit par des cas d’applications de mise en forme.

## 4.2 Applications

Pour cette dernière section, nous étudions différents problèmes de mise en forme présentés dans la section 2.5 et nous utilisons le préconditionneur multigrille pour leur résolution afin d’évaluer les gains de cette approche sur la résolution de systèmes associés à des écoulements plus complexes ou à des simulations complètes de procédés.

### 4.2.1 Forgeage du Triaxe

Considérons pour commencer la simulation de forgeage du triaxe, présenté en 2.5.2, et qui a déjà servi dans le chapitre précédent pour établir différents réglages du préconditionneur multigrille. Nous considérons cette fois la simulation complète du procédé, de l’état initial (figure 4.8a) à l’état final (figure 4.8b), sur deux tailles de problème.

#### Maillage à 33 580 nœuds

Présentons tout d’abord les résultats obtenus pour un maillage initial de 33 580 nœuds, figure 4.9. Compte tenu de la taille du problème, et des conclusions que nous avons pu tirer dans la section 4.1.1, le préconditionneur 2 grilles devrait présenter les meilleurs résultats. Nous vérifions cela, en comparant les performances de la méthode de référence PCR+ILU1, du préconditionneur 2 grilles, PCR+2G, et du préconditionneur 3 grilles, PCR+3G. Les figures 4.10a et 4.10b présentent les maillages intermédiaire et grossier de 3 946 et 496 nœuds pour la méthode 3 grilles, la figure 4.10c correspondant au maillage grossier de 528 nœuds pour la méthode 2 grilles.

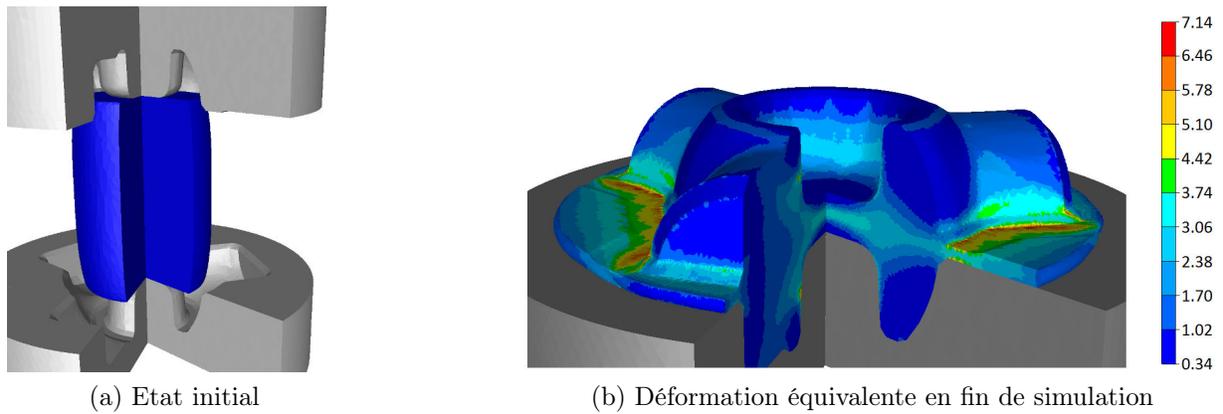


FIGURE 4.8 – Début et fin de simulation de forgeage d'un triaxe

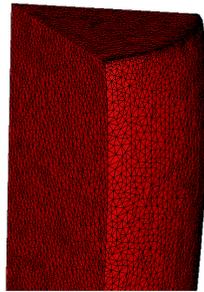


FIGURE 4.9 – Maillage initial de 33 580 nœuds pour la simulation de forgeage du triaxe

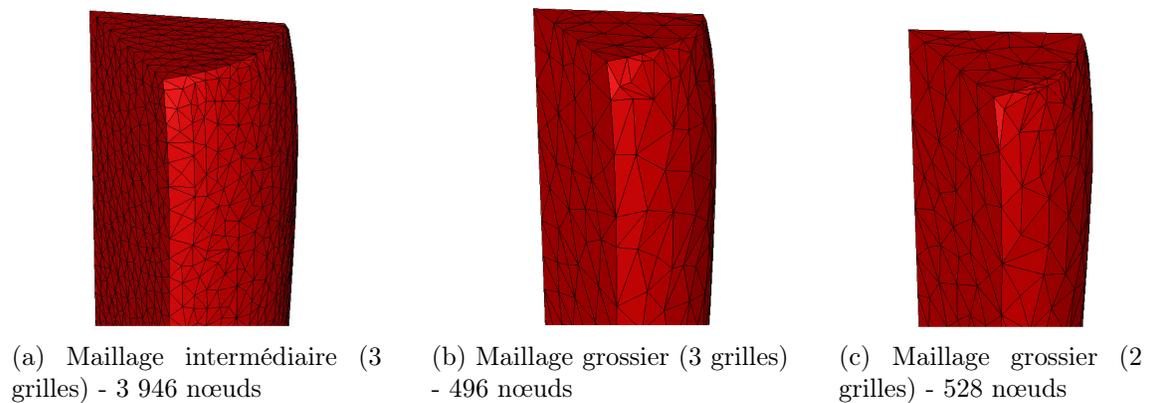


FIGURE 4.10 – Maillages intermédiaire et grossiers générés pour les méthodes 2 et 3 grilles en début de simulation du forgeage du triaxe à partir d'un maillage initial à 33 580 nœuds

Après 550 incréments, 35 remaillages et autant de génération de grilles grossières (et intermédiaire pour la méthode 3 grilles), le maillage fin de la pièce finale comporte 46 500 nœuds (figure 4.11). La génération des grilles repose sur un paramètre de déraffinement défini en début de simulation, lié à la taille du problème. Cette taille grandissant, ce paramètre conduit à générer des grilles intermédiaires et grossières comportant trop de nœuds, ce qui nuit à l'efficacité de la méthode, comme nous avons pu en discuter au chapitre 3. En fin de simulation, les grilles de 6 982 et 1 119 nœuds générées pénalisent l'efficacité du préconditionneur multigrille durant les derniers incréments de calcul. Il devient alors moins efficace que le PCR+ILU1, malgré un début de calcul prometteur, comme le montre la figure 4.12 qui présente les temps cumulés de ces 2 méthodes en fonction des incréments de temps.

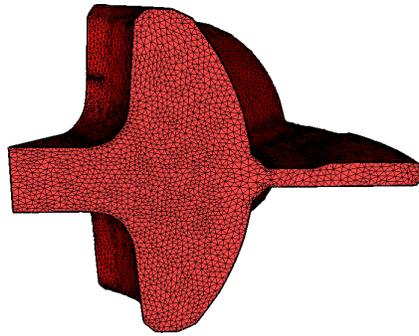


FIGURE 4.11 – Maillage fin final de 46 500 nœuds pour la simulation de forgeage du triaxe

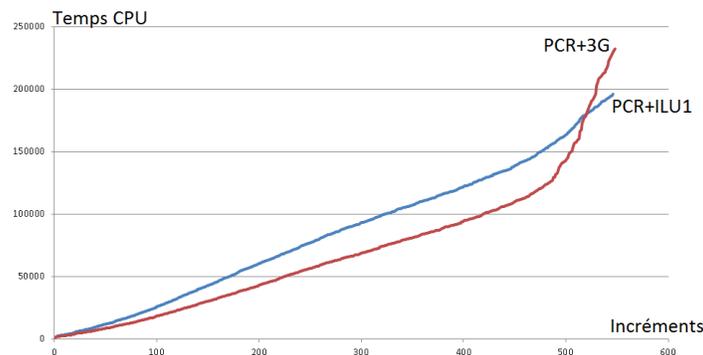
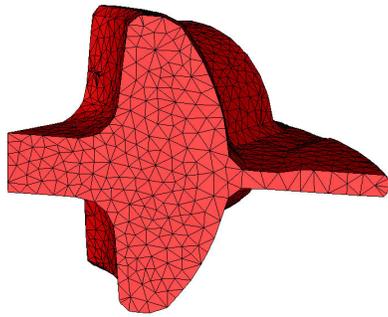
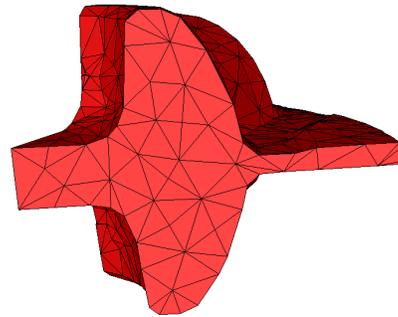


FIGURE 4.12 – Evolution des temps de calcul cumulés du PCR+ILU1 et PCR+3G en fonction du nombre d'incrément pour la simulation de forgeage d'un triaxe, avec un maillage initial de 33 580 nœuds et maillages intermédiaires et grossiers calculés à partir d'un facteur de déraffinement défini en début de calcul

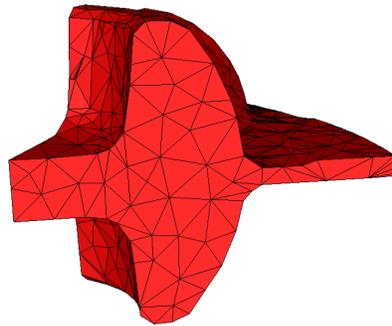
Afin de régler ce problème, nous modifions la valeur de ce paramètre de déraffinement afin d'obtenir en fin de simulation des maillages grossiers de tailles plus réduites. Pour cela, nous calculons ce facteur à partir du nombre de nœuds du maillage final et non du maillage initial. Nous obtenons ainsi un maillage grossier pour la méthode 2 grilles de 1 082 nœuds (figure 4.13c) et les maillages intermédiaire et grossier de 5 469 et 969 nœuds pour la 3 grilles (figures 4.13a et 4.13b). Sur les figures 4.14 et 4.15, nous présentons les temps cumulés globaux et les accélérations correspondantes tout au long de la simulation. Une baisse de performance, dont nous discuterons par la suite, est toujours observée en fin de simulation mais de manière beaucoup moins marquée. Les temps de calculs multigrilles sont tous deux meilleurs que ceux du PCR+ILU1, la méthode 2 grilles restant la plus rapide.



(a) Maillage intermédiaire (3 grilles) - 5 469 nœuds



(b) Maillage grossier (3 grilles) - 969 nœuds



(c) Maillage grossier (2 grilles) - 1 082 nœuds

FIGURE 4.13 – Maillages intermédiaire et grossiers générés pour les méthodes 2 et 3 grilles en fin de simulation du forgeage du triaxe à partir d'un maillage à 46 500 nœuds

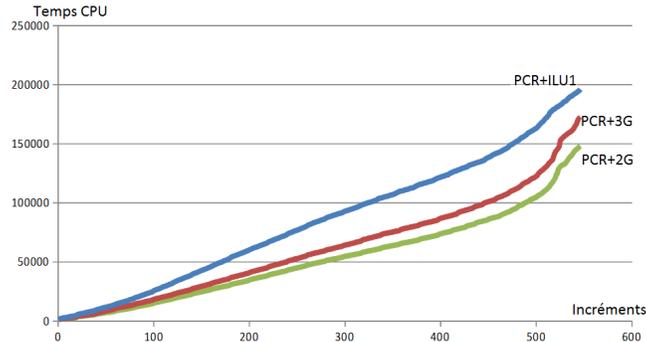


FIGURE 4.14 – Evolution du temps de calcul cumulé du PCR+ILU1, PCR+3G et PCR+2G en fonction du nombre d'incrément pour la simulation de forgeage d'un triaxe, avec un maillage initial de 33 580 nœuds

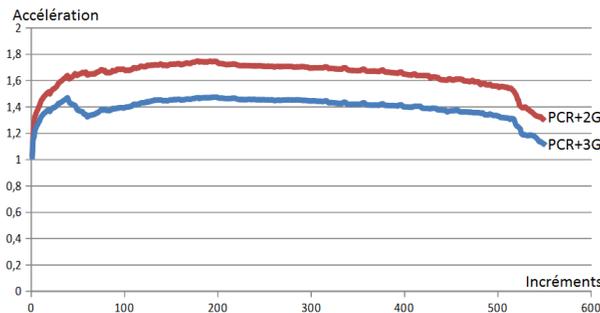


FIGURE 4.15 – Accélération cumulée des temps de calcul du PCR+2G et PCR+3G par rapport au PCR+ILU1 en fonction du nombre d'incrément pour la simulation de forgeage du triaxe, avec un maillage initial de 33 580 nœuds

L'accélération finale globale de 1,3 observée en fin de simulation (pour la méthode 2 grilles) ne reflète cependant pas parfaitement l'influence du préconditionneur multigrille. En effet, les temps de calculs comprennent des opérations sur lesquelles le solveur linéaire n'a aucun impact. Ces opérations représentent environ 30% du temps de calcul global de la simulation. Le tableau 4.3 permet de vérifier cette répartition grâce au détail des temps de calcul.

La répartition des temps de calcul sur la simulation avec PCR+ILU1 correspond parfaitement avec les 30% attendus. Cette part passe à 40% avec le multigrille, principalement en raison de la diminution du temps consacré aux résolutions de systèmes linéaires. Le surcoût apporté par la construction de maillages supplémentaires pour le préconditionneur multigrille n'est finalement que léger, faisant passer de 16 à 17 heures le temps passé sur les autres opé-

	PCR+ILU1		PCR+2G		Accélérations
<b>Résolutions de systèmes linéaires</b>	38 heures	70%	25 heures	60%	1,5
<b>Autres opérations</b>	16 heures	30%	17 heures	40%	0,9
<b>Temps de calcul global</b>	54 heures	100%	42 heures	100%	1,3

TABLE 4.3 – Répartition du temps de calcul du PCR+2G et du PCR+ILU1 pour la simulation de forgeage du triaxe et accélérations associées, avec un maillage initial de 33 580 nœuds

rations que la résolution (soit un incrément de 6 %). Cette heure supplémentaire sur les 42 heures de la simulation complète ne représente que 2% du temps de calcul.

L’apport du préconditionneur multigrille se mesure sur l’accélération des résolutions de systèmes linéaires. Pour le PCR+2G, la figure 4.16 présente l’accélération cumulée tout au long de la simulation en ne prenant en compte que ces temps de résolution. Elle met en évidence une accélération totale en fin de simulation de 1,5 (au lieu de 1,3 en considérant les temps globaux).

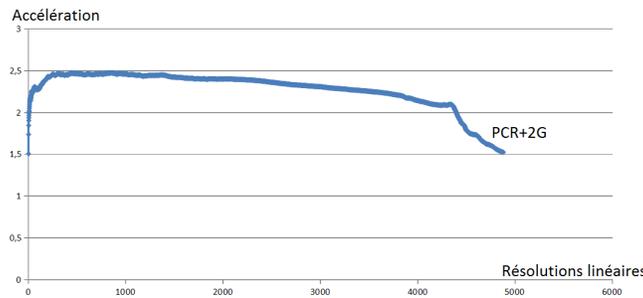


FIGURE 4.16 – Accélération cumulée sur les résolutions linéaires du PCR+2G par rapport au PCR+ILU1 en fonction du nombre de systèmes linéaires résolus pour la simulation de forgeage du triaxe, avec un maillage initial de 33 580 nœuds

Ce résultat présente deux caractéristiques remarquables : l’accélération cumulée du solveur seul atteint d’abord un maximum de 2,5, avant de décroître progressivement et finalement de chuter brutalement. La décroissance s’explique d’abord par l’augmentation progressive du nombre de nœuds du maillage grossier. La chute brutale observée en fin de simulation provient pour partie de l’apparition d’un repli de matière (après une course de 55mm de l’outil supérieur, correspondant à l’incrément 440, figure 4.17). Ce phénomène entraîne un repli du maillage et la possibilité que certains éléments s’interpénètrent. Les opérateurs de transfert peuvent alors être faussés, interpolant les champs sur le mauvais bord du repli, ce qui

peut être la cause d’erreurs lors du passage entre les grilles et ainsi ralentir la convergence. Pour autre partie de cette explication, en fin de simulation l’écoulement dans la bavure est très aplati (figure 4.11). L’épaisseur comporte alors peu de nœuds de sorte que la grille intermédiaire (figure 4.13a) n’y possède qu’un seul élément. La grille grossière ne peut alors pas être déraffinée dans cette direction et la méthode perd en efficacité dans le calcul des basses fréquences.

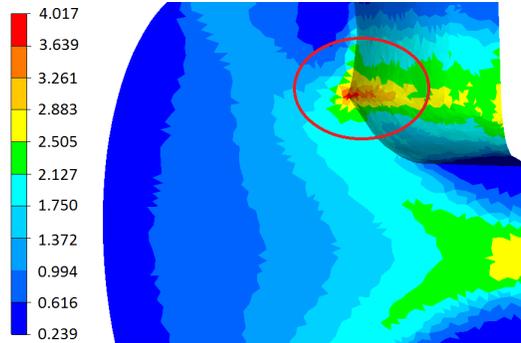


FIGURE 4.17 – Apparition du repli au cours de la simulation de forgeage du triaxe. Déformation équivalente après une course de 55mm de l’outil supérieur

### Maillage à 59 389 nœuds

Nous utilisons maintenant le même cas avec un maillage initial de 59 389 nœuds (figure 4.18a), sur lequel nous évaluons les performances parallèles. Nous utilisons pour cela la méthode 3 grilles, avec une grille intermédiaire à 4 416 nœuds et une grille grossière à 418 nœuds. En fin de simulation, le maillage de calcul atteint 77 696 nœuds (figure 4.18b), dont les maillages intermédiaire et grossier de 5 834 et 830 nœuds sont présentés figures 4.19a et 4.19b.

Le tableau 4.4 présente les temps de calcul relevés en séquentiel, puis sur 2 et 4 processeurs, ainsi que les accélérations multigrilles et parallèles (entre 2 et 4 processeurs, conformément à la discussion de la section 4.1.2) correspondantes.

En séquentiel, le PCR+3G permet une accélération globale d’un facteur 1,57. Elle correspond à une accélération moyenne du solveur linéaire de l’ordre de 1.8. conforme à ce qui avait été observé dans la section 4.1.1 sur la résolution d’un seul incrément.

En parallèle, nous observons une légère chute de l’accélération multigrille, semblable à celle déjà constatée dans la section 4.1.2, et principalement expliquée par le comportement des pré-conditionneurs ILU utilisés. L’accélération multigrille autour de 1,3 pour les calculs utilisant

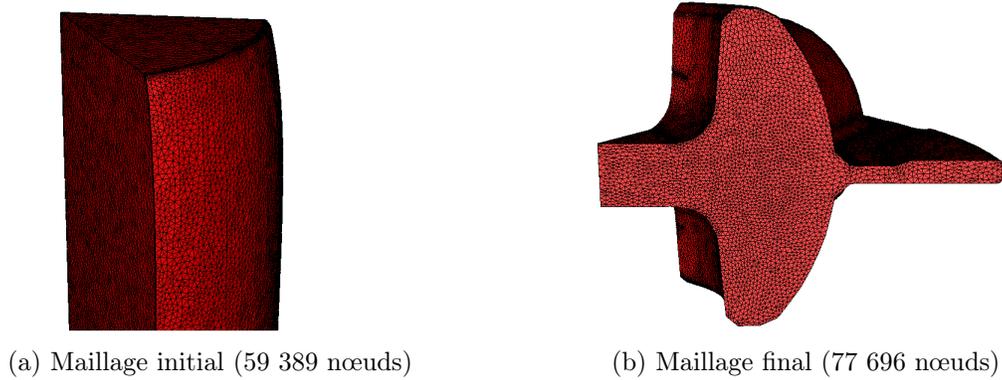


FIGURE 4.18 – Maillages de calcul en début et fin de simulation de forgeage d'un triaxe

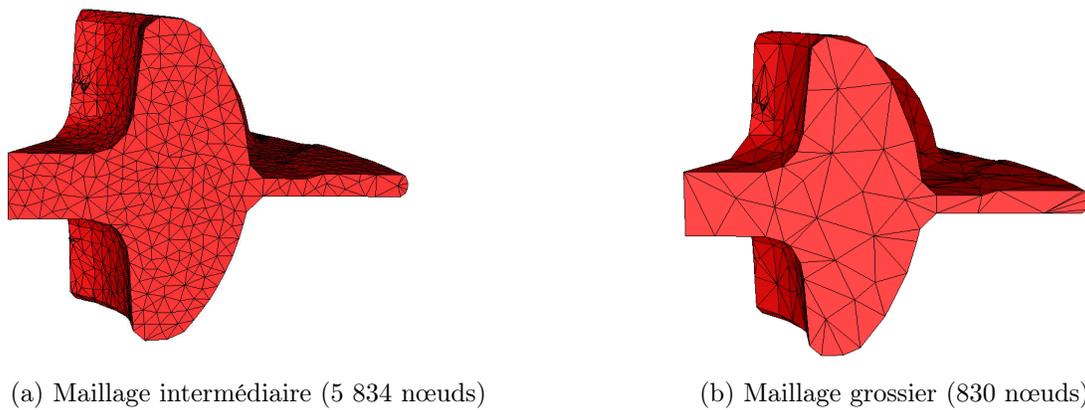


FIGURE 4.19 – Maillages intermédiaire et grossier pour la méthode 3 grilles en fin de simulation de forgeage d'un triaxe

	1 processeur	2 processeurs	4 processeurs	Accélération parallèle
<b>PCR+ILU1</b>	4 jours 9 heures	2 jours 4 heures	1 jour 2 heures	<b>2.00</b>
<b>PCR+3G</b>	2 jours 19 heures	1 jour 15 heures	20 heures	<b>1.95</b>
<b>Accélération MG</b>	<b>1.57</b>	<b>1.33</b>	<b>1.30</b>	

TABLE 4.4 – Temps de calcul des méthodes PCR+ILU1 et PCR+3G et accélérations multi-grille et parallèle (entre 2 et 4 processeurs) pour la simulation de forgeage d'un triaxe avec un maillage initial de 59 389 nœuds

2 et 4 processeurs reste néanmoins satisfaisante. Elle résulte d’une accélération du solveur linéaire supérieure à 1,5. Enfin, l’accélération parallèle est quasi-optimale et surtout similaire à celle du PCR+ILU1.

Pour la simulation de forgeage du triaxe, nous retrouvons ainsi les performances observées sur l’analyse d’un seul incrément de calcul, sauf en toute fin de simulation où les particularités de l’écoulement (bavure et repli) entraînent une réduction des performances des multigrilles, qui restent néanmoins très encourageantes.

### 4.2.2 Laminage retour

Intéressons nous maintenant au laminage retour, présenté dans la section 2.5.3. Nous présentons à nouveau l’état initial et l’état final sur les figures 4.20a et 4.20b. Une nouvelle fois, nous considérons deux tailles de problème.

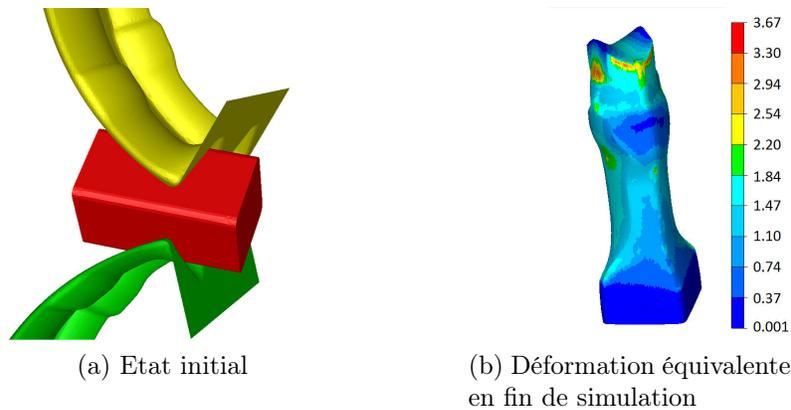


FIGURE 4.20 – Simulation de laminage retour

Tout d’abord, considérons un maillage initial de 42 639 nœuds, présenté sur la figure 4.21, sur lequel nous pouvons comparer la méthode de référence PCR+ILU1, le préconditionneur 2 grilles, PCR+2G, et le préconditionneur 3 grilles, PCR+3G. Le maillage grossier de 431 nœuds (en début de simulation) pour le préconditionneur 2 grilles est présenté sur la figure 4.22c, tandis que les maillages intermédiaire et grossier du préconditionneur 3 grilles, de 4 318 et 478 nœuds, apparaissent sur les figures 4.22a et 4.22b.

Après 133 incréments de temps et 40 remaillages, la pièce finale comporte 49 635 nœuds (figure 4.23a). L’algorithme 2 grilles parvient alors à générer un maillage grossier de 1 126

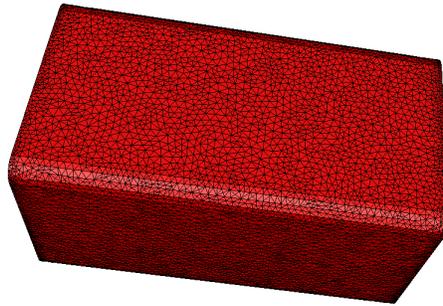
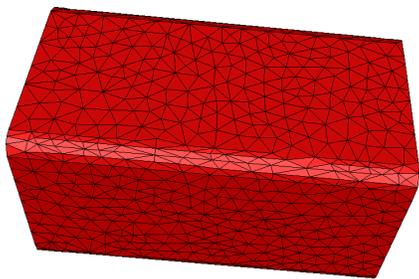
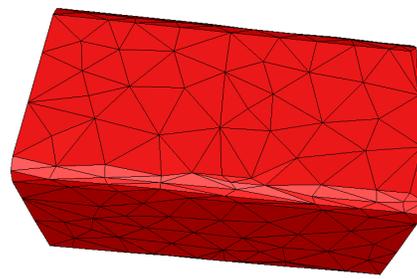


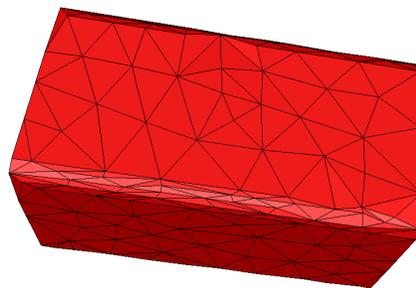
FIGURE 4.21 – Maillage initial de 42 639 nœuds pour la simulation de laminage retour



(a) Maillage intermédiaire (3 grilles) - 4 318 nœuds



(b) Maillage grossier (3 grilles) - 478 nœuds



(c) Maillage grossier (2 grilles) - 431 nœuds

FIGURE 4.22 – Maillages intermédiaire et grossiers générés pour les méthodes 2 et 3 grilles en début de simulation du laminage retour à partir d'un maillage initial à 42 639 nœuds

nœuds (figure 4.23b). Les temps de calcul cumulés sont présentés sur la figure 4.24, en fonction des incréments de temps.

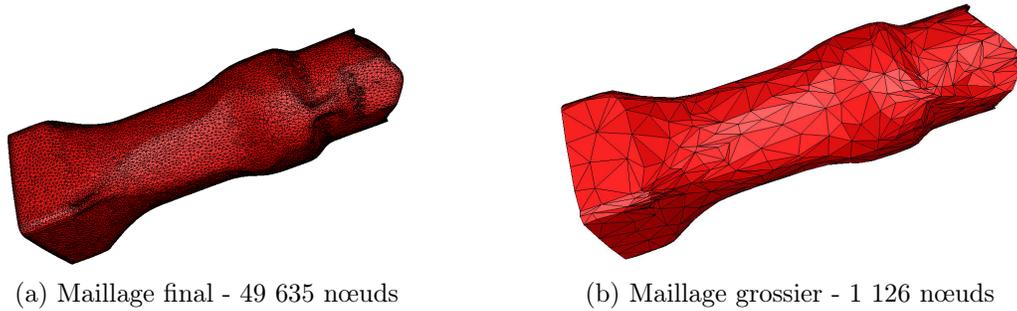


FIGURE 4.23 – Maillages en fin de simulation de laminage retour

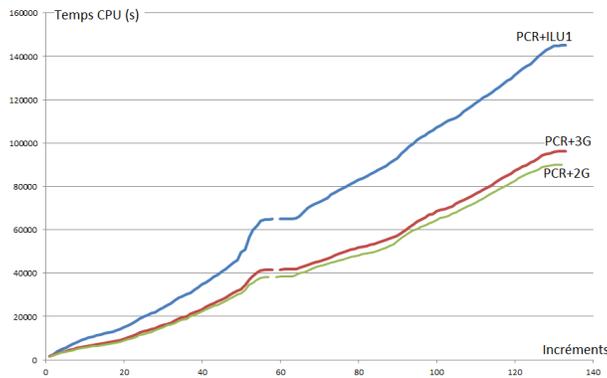


FIGURE 4.24 – Evolution des temps de calcul cumulé du PCR+ILU1, PCR+2G et PCR+3G en fonction du nombre d'incrément pour la simulation de laminage retour avec un maillage initial de 43 000 nœuds

Tout d'abord, nous pouvons voir que les 3 méthodes présentent une évolution identique et relativement linéaire sur la globalité de la simulation, avec des courbes superposables. Les méthodes multigrilles apparaissent clairement plus efficaces avec, comme prévu, une supériorité du préconditionneur 2 grilles, compte tenu de la taille de problème. L'accélération totale en fin de simulation atteint 1,5 pour le préconditionneur 3 grilles et 1,6 pour le 2 grilles. La figure 4.25 présente les accélérations "instantanées" tout au long de la simulation pour la méthode PCR+2G. Ces accélérations sont "lissées" car l'utilisation de deux solveurs différents ne permet pas de suivre strictement la même séquence de remaillages. Il arrive donc qu'à un incrément donné, une seule des simulations remaille, et inversement sur un incrément suivant. Cela provoque une forte différence de temps de calcul à cet incrément de remaillage tant pour les opérations de remaillage que pour la résolution de systèmes non linéaires, mais elle est compensée lorsque le remaillage se produit avec l'autre solveur. Nous pouvons ainsi voir que

le PCR+2G offre une accélération qui peut aller jusqu'à 2,5. Sur l'ensemble de la simulation, la grande majorité des incréments profite d'une accélération proche de 1,5. L'accélération finale est ainsi de 1,6.

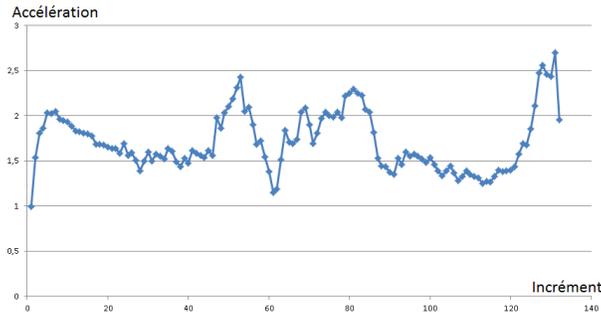


FIGURE 4.25 – Accélération "lissée" par incrément du PCR+2G par rapport au PCR+ILU1 pour la simulation de laminage retour avec un maillage initial de 42 639 nœuds

Comme précédemment, ces temps de calculs prennent en compte toute une partie des opérations sur lesquelles le préconditionneur multigrille n'a pas d'effet. Afin de mesurer l'impact réel du développement effectué, nous traçons sur la figure 4.26 l'accélération cumulée sur les seuls temps consacrés aux résolutions de systèmes linéaires, en fonction du nombre de systèmes linéaires résolus.

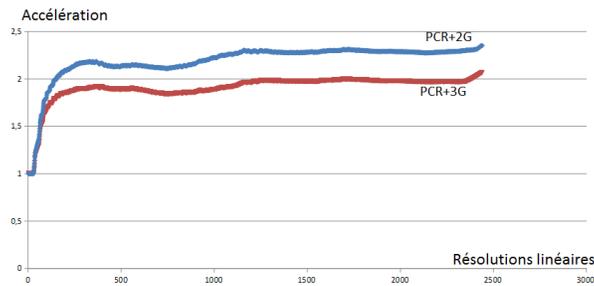


FIGURE 4.26 – Accélération cumulée sur les résolutions linéaires des PCR+2G et PCR+3G par rapport au PCR+ILU1 sur la simulation du laminage retour avec un maillage initial de 42 639 nœuds

Les accélérations de 1,5 et 1,6 obtenues sur le temps de simulation global, respectivement par le PCR+3G et le PCR+2G correspondent à des accélérations de 2,1 et 2,3 sur les résolutions linéaires. Le tableau 4.5 permet d'observer la répartition du temps de calcul pour le

PCR+2G et le PCR+ILU1. Cette répartition respecte une nouvelle fois parfaitement les 70% passés dans la résolution linéaire pour le PCR+ILU1. Surtout, le préconditionneur 2 grilles apporte une accélération de 2,3 sur les résolutions de systèmes linéaires, rendant cette partie plus rapide que les autres opérations de la simulation. Une nouvelle fois, le surcoût apporté par la construction de maillages supplémentaires est limité. Il représente seulement 1 heure sur les 25 heures de la simulation globale (soit 4 %), et ce malgré les nombreux remaillages (40), et autant de générations de grilles supplémentaires.

	PCR+ILU1		PCR+2G		Accélérations
<b>Résolutions de systèmes linéaires</b>	28 heures	70%	12 heures	48%	2,3
<b>Autres opérations</b>	12 heures	30%	13 heures	52%	0,9
<b>Temps de calcul global</b>	40 heures	100%	25 heures	100%	1,6

TABLE 4.5 – Répartition du temps de calcul du PCR+2G et du PCR+ILU1 et accélérations associées pour la simulation de laminage retour, avec un maillage initial de 42 639 nœuds

Complétons ces résultats satisfaisants avec la simulation du même procédé mais avec un maillage initial plus fin comportant 59 491 nœuds. Après 132 incréments et 50 remaillages, la pièce finale comporte 69 041 nœuds (figure 4.27a). Les maillages générés par le PCR+3G en fin de simulation, de 6 155 et 1 005 nœuds, sont présentés sur les figures 4.27b et 4.27c.

Le tableau 4.6 fournit les temps de cacul du PCR+ILU1 et du PCR+3G pour la totalité de la simulation mais aussi pour la résolution des systèmes linéaires.

	PCR+ILU1		PCR+3G		Accélérations
<b>Résolutions de systèmes linéaires</b>	50 heures	68%	21 heures	47%	2,4
<b>Autres opérations</b>	23 heures	32%	24 heures	53%	0,96
<b>Temps de calcul global</b>	73 heures	100%	45 heures	100%	1,6

TABLE 4.6 – Répartition du temps de calcul du PCR+ILU1 et du PCR+3G et accélérations associées pour la simulation de laminage retour, avec un maillage initial de 59 491 nœuds

L'accélération globale de 1,6 est obtenue grâce à une accélération de 2,4 sur les résolutions de systèmes linéaires. Cette performance fait passer la part de temps de calcul consacrée aux systèmes linéaires de 68 % pour le PCR+ILU1 à 47 %. Les opérations annexes occupent donc 53 % du temps de calgul global dans le cas du PCR+3G et le surcoût multigrille provenant de la construction des niveaux intermédiaire et grossier à chaque remaillage est une nouvelle fois négligeable (1 heure supplémentaire sur une simulation de 45 heures, soit 2%).

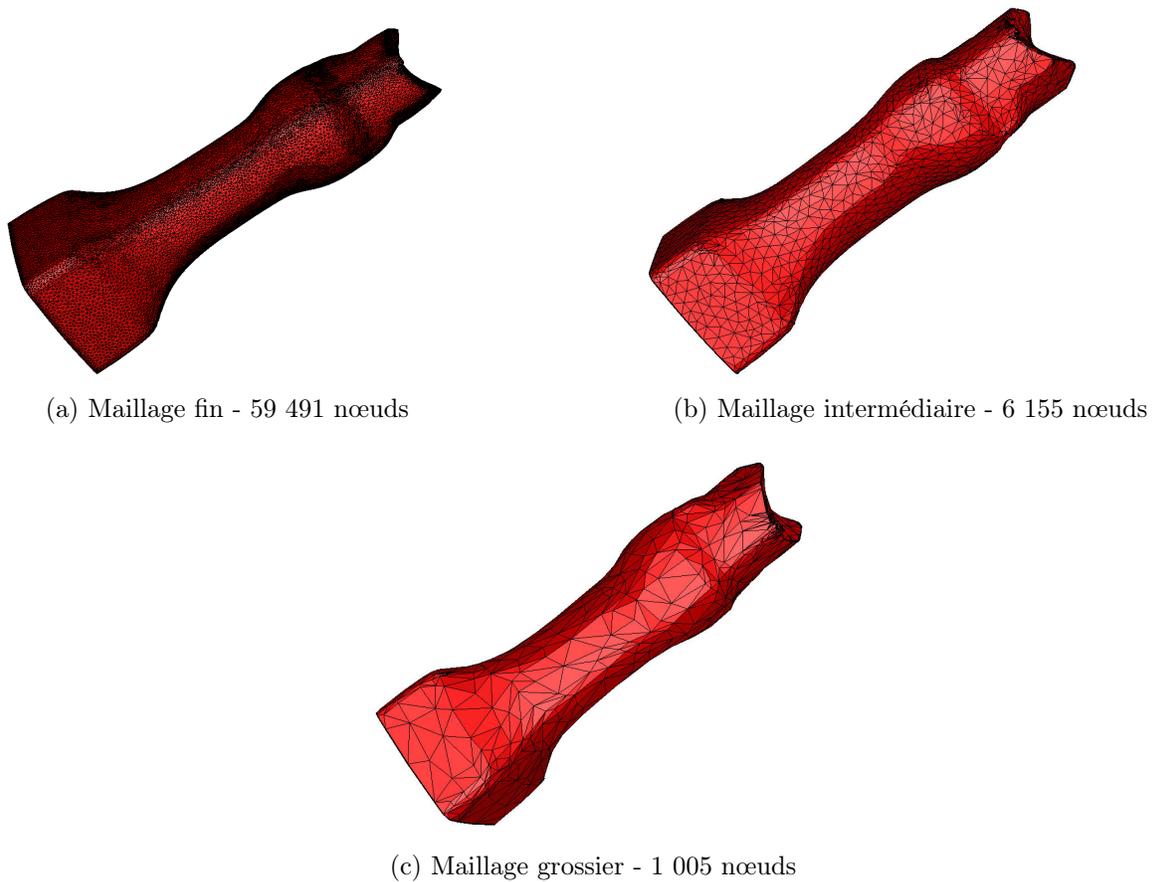


FIGURE 4.27 – Maillages en fin de simulation utilisés par le PCR+3G pour la simulation du laminage retour, pour un maillage en début de simulation de 59 491 nœuds

Comme avec le maillage initial à 42 639 nœuds, l'utilisation du préconditionneur multigrille sur la simulation du procédé de laminage retour diminue très significativement les temps de calcul. Son impact important sur la réduction du temps consacré à la résolution de systèmes linéaires, est néanmoins pondéré par l'importance du temps dédié aux opérations annexes, et notamment ici les nombreux remaillages.

### 4.2.3 Martelage rotatif

Intéressons-nous, pour terminer, au procédé de martelage rotatif. Cette simulation, présentée dans la section 2.5.4, nécessite un grand nombre de nœuds afin d'être discrétisée

correctement et nécessite donc des temps de calculs très longs.

Nous discuterons ici des résultats obtenus pour la simulation de la première passe (sur quatre) de ce procédé, correspondant à 430 incréments (sur 1200). L'état initial et l'état final sont présentés sur les figures 4.28a et 4.28b. Nous évaluerons les performances parallèles du PCR+3G sur cette simulation en utilisant 8 processeurs.

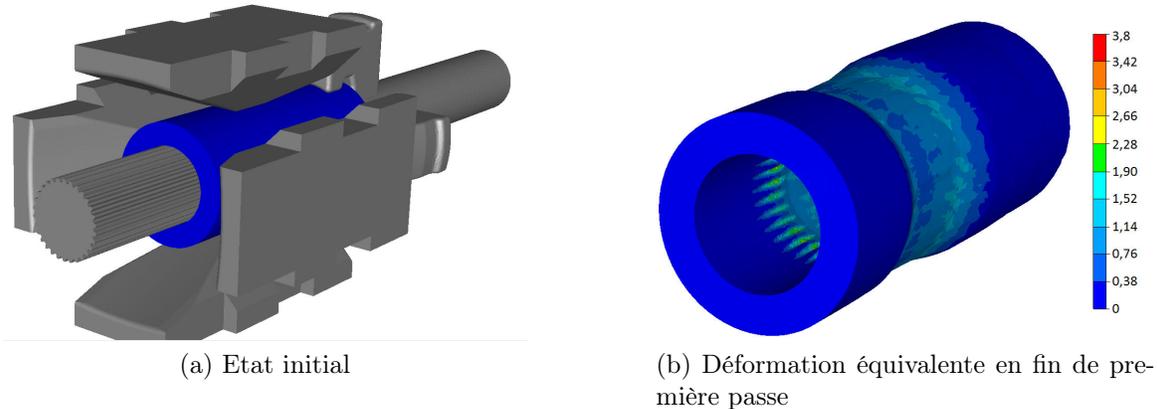


FIGURE 4.28 – Simulation de martelage rotatif

Le maillage initial comporte 213 903 nœuds en début de simulation (figure 4.29a). Ce grand nombre de nœuds mais surtout la géométrie de la pièce n'ont pas permis de respecter les préconisations établies dans le chapitre précédent. Il a ainsi été impossible avec des facteurs de déraffinement homogènes d'obtenir un maillage suffisamment grossier. Les performances du préconditionneur multigrille dépendant en grande partie de la taille de la grille grossière, nous avons utilisé un facteur de déraffinement de 20 puis de 10 pour obtenir un maillage intermédiaire à 10 662 nœuds et un maillage grossier à 1 111 nœuds, présentés sur les figures 4.29b et 4.29c. Les performances de la méthode multigrille sont alors limitées et pourraient être améliorées avec un travail sur le mailleur. Pour simuler cette première passe, 12 remaillages sont nécessaires et aboutissent à un maillage final de 372 715 nœuds et une géométrie complexe avec des cannelures intérieures (figure 4.30a). Les maillages générés en fin de calcul pour le PCR+3G de 14 659 et 2 531 nœuds sont présentés sur les figures 4.30b et 4.30c. Leurs tailles traduisent la difficulté à déraffiner davantage tout en conservant la géométrie du domaine et tout spécialement les fines cannelures intérieures.

Les courbes de la figure 4.31 présentent l'évolution du temps de calcul en fonction du nombre d'incrémentes pour les deux méthodes PCR+ILU1 et PCR+3G.

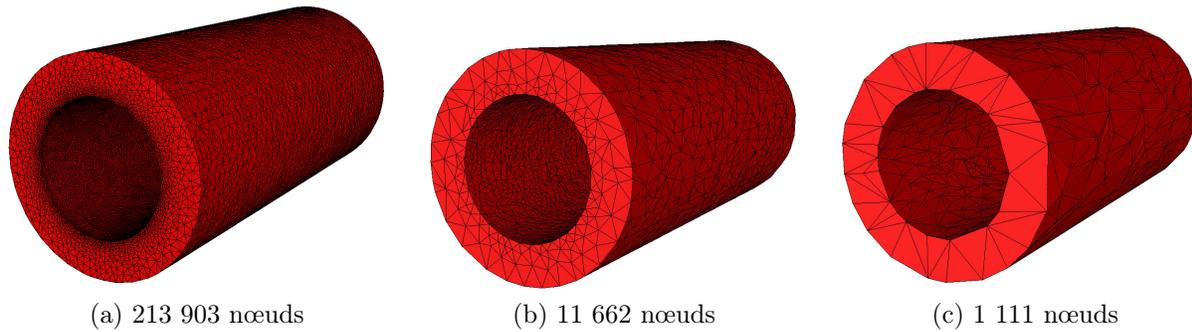


FIGURE 4.29 – Maillages de calcul initial (a) et maillages intermédiaire et grossier (b et c) utilisés par le PCR+3G pour la simulation du martelage rotatif

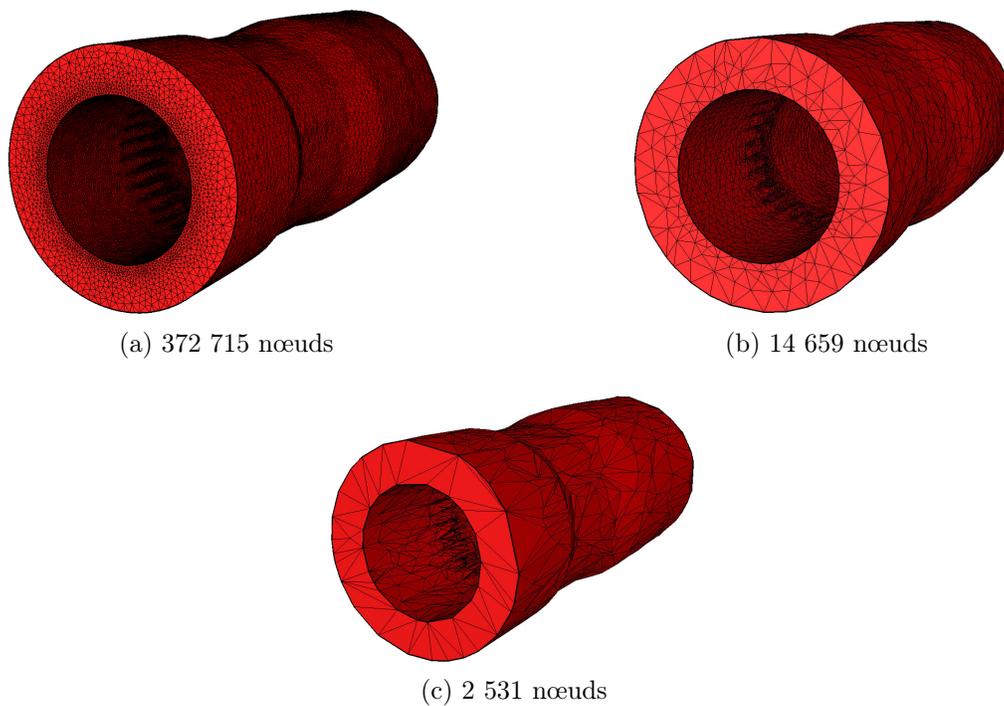


FIGURE 4.30 – Maillage de calcul final (a) et maillages intermédiaire et grossier (b et c) utilisés par le PCR+3G pour la simulation du martelage rotatif

Alors que le PCR+ILU1 effectue cette simulation en approximativement 4 jours et 14 heures, le PCR+3G termine en 2 jours et 22h, soit une accélération d'un facteur 1,6. Nous pouvons constater que cette accélération subit une chute autour de l'incrément 360. Un dé-

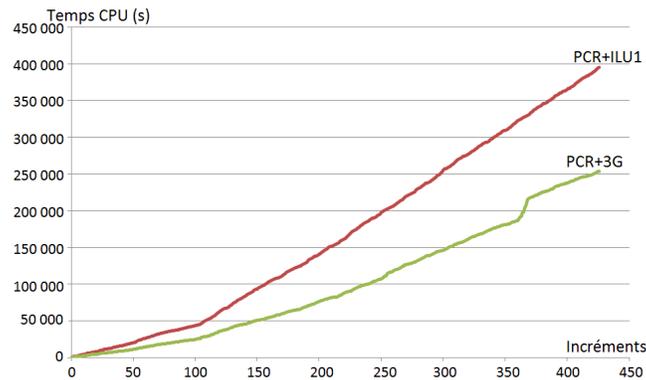


FIGURE 4.31 – Evolution des temps de calcul cumulés du PCR+ILU1 et PCR+3G en fonction du nombre d'incrément pour la simulation de martelage rotatif

crochage de la pente peut être observé sur la courbe d'évolution du temps de calcul. Nous pouvons voir plus en détail ces phénomènes sur les figures 4.32 et 4.33, présentant respectivement les accélérations cumulées et les accélérations "lissées" par incrément du PCR+3G sur le PCR+ILU.

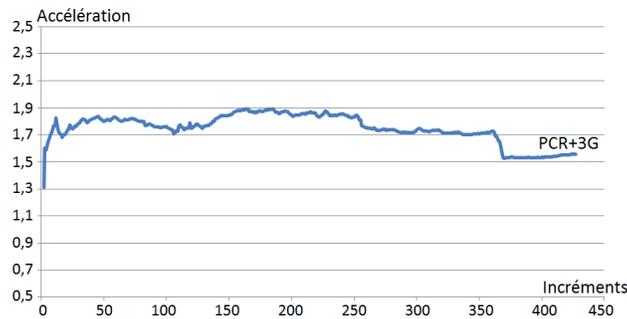


FIGURE 4.32 – Accélération cumulée du PCR+3G sur le PCR+ILU1 lors de la simulation du martelage rotatif

De manière similaire à la simulation de forgeage du triaxe, nous observons qu'une accélération maximum sur les temps cumulés de 1,9 est atteinte autour de l'incrément 180, avant de décroître progressivement jusqu'à la fin de la simulation. Une explication de ce comportement peut être tirée des accélérations par incrément (figure 4.33). La première observation est que cette accélération par incrément se situe globalement autour de 2, avec des montées à 2,5 (et un maximum à 3,15). Nous observons également, à deux reprises, une

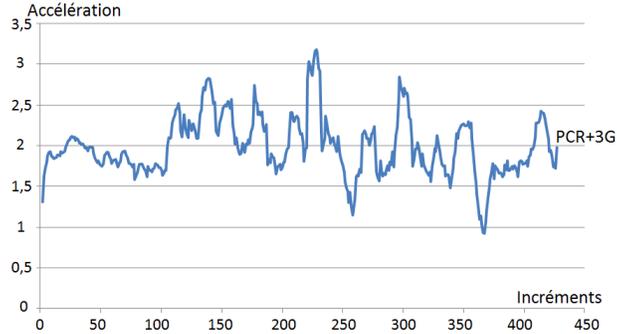


FIGURE 4.33 – Accélération "lissée" par incrément du PCR+3G sur le PCR+ILU1 lors de la simulation du martelage rotatif

forte chute sur une série d'incrément, d'abord autour de l'incrément 250 puis de l'incrément 360. Le comportement du solveur finit dans les deux cas par s'améliorer pour retrouver des accélérations dans la moyenne. Ces chutes d'efficacité proviennent d'increments où le solveur multigrille rencontre des difficultés à converger. Dans les deux cas, cela intervient juste après un remaillage et nous porte à croire que les maillages générés sont en cause. Après le remaillage de l'incrément 362, le maillage de calcul comporte 367 968 nœuds (figure 4.34a) tandis que les maillages intermédiaire et grossier en comportent 13 894 et 2 327 (figures 4.34b et 4.34c). Cette taille de grille grossière ne permet pas d'obtenir de bonnes performances. De plus, les facteurs de déraffinement (26,5 entre le maillage fin et le maillage intermédiaire et 6 entre le maillage intermédiaire et le maillage grossier) ne sont pas homogènes et le premier s'avère bien trop grand. Enfin, nous voyons que les zones très raffinées à l'intérieur du cylindre présentent des éléments de mauvaise qualité tant sur le maillage intermédiaire que grossier. Nous avons vu dans les sections 3.2.2 et 3.2.3 combien ces différents facteurs (taille de la grille grossière, facteurs de déraffinement homogènes et qualité des maillages générés) influaient sur les performances de la méthode multigrille. Cela n'empêche pas malgré tout le PCR+3G de converger et d'être presque tout le temps plus rapide que le PCR+ILU1, tout en donnant bien entendu les mêmes résultats (figure 4.35), donnant ainsi preuve de sa robustesse.

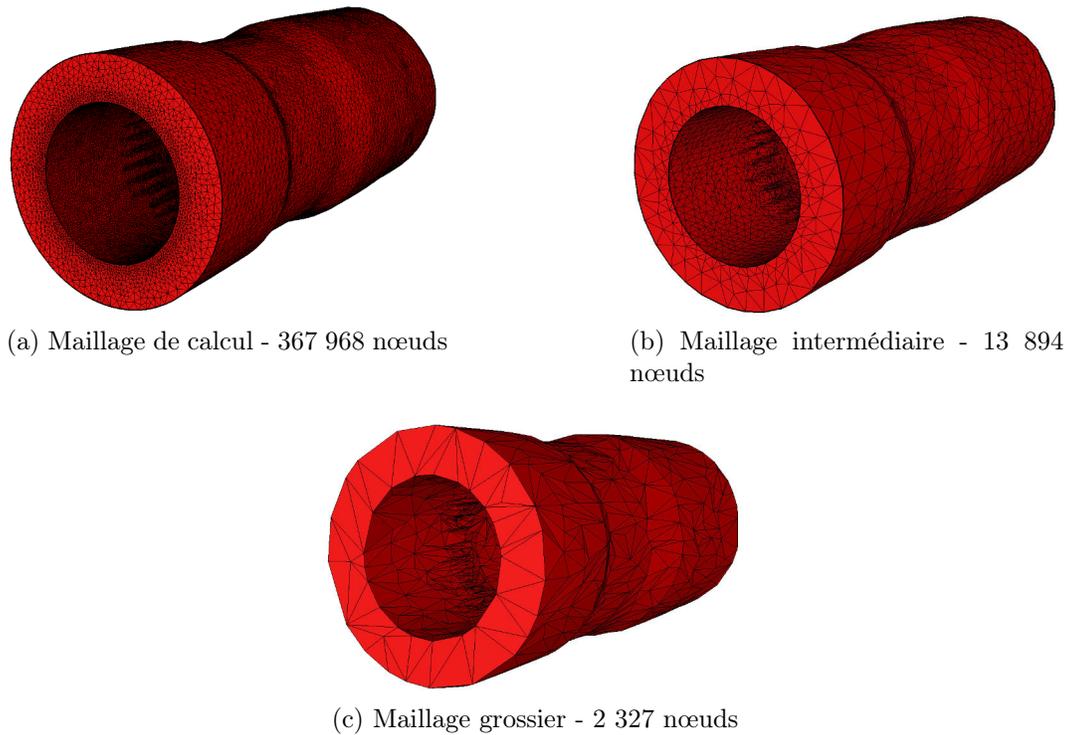


FIGURE 4.34 – Maillage de calcul et maillages intermédiaire et grossier utilisé par le PCR+3G à l'incrément 362 de la simulation du martelage rotatif

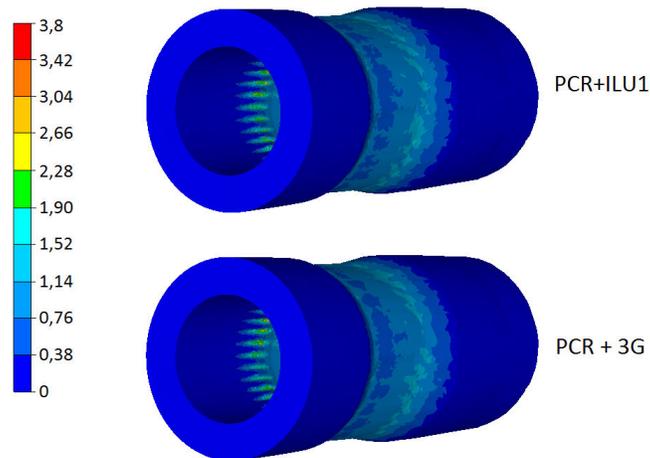


FIGURE 4.35 – Déformation équivalente en fin de simulation obtenue par le PCR+ILU1 et le PCR+3G sur la simulation du martelage rotatif

Cet exemple illustre combien le point clé de la méthode multigrille en mise en forme est la génération optimale des différentes grilles grossières, tant du point de vue du nombre de nœuds, des facteurs de déraffinement que de la qualité du maillage. Notons aussi que ces calculs ont été faits en parallèle, où la méthode multigrille est un peu moins performante.

Concernant la temps consacré uniquement à la résolution des systèmes linéaires, la figure 4.36 montre l'accélération cumulée en fonction du nombre de systèmes résolus. Nous observons une accélération relativement constante autour de 2,5 avant une chute, qui coïncide avec l'incrément 360, réduisant cette accélération à une valeur finale à 1,9. Le tableau 4.7 résume les temps de calcul, les accélérations et la part consacrée à la résolution linéaire sur cette simulation de martelage rotatif.

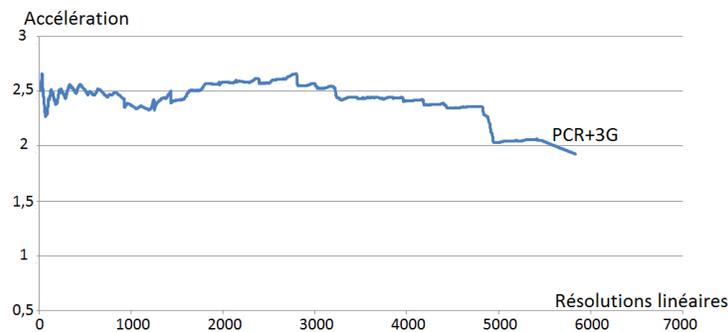


FIGURE 4.36 – Accélération du PCR+3G sur le PCR+ILU1 sur les temps cumulés passés dans les résolutions de systèmes linéaires lors de la simulation du martelage rotatif

	PCR+ILU1		PCR+3G		Accélérations
<b>Résolutions de systèmes linéaires</b>	88 heures	80%	45 heures	64%	1,96
<b>Autres opérations</b>	22 heures	20%	25 heures	36%	0,9
<b>Total</b>	110 heures	100%	70 heures	100%	1,6

TABLE 4.7 – Répartition du temps de calcul du PCR+3G et du PCR+ILU1 et accélérations associées pour la simulation de martelage rotatif

## 4.3 Conclusion

Dans ce chapitre, nous avons évalué les performances de la méthode multigrille hybride développée dans cette thèse. Devant être particulièrement adaptée à la résolution de systèmes

linéaires comportant un très grand nombre de degrés de liberté, l'étude de son comportement asymptotique a permis de vérifier que son comportement est quasi-linéaire de type  $O(N \log N)$ . Elle a également permis de mettre en avant le bon comportement de la méthode 2 grilles sur les problèmes les plus grossiers, où elle est la plus efficace. Nous avons ainsi pu établir un seuil de passage du 2 au 3 grilles entre 45 000 et 80 000 nœuds.

L'étude parallèle a montré l'excellent niveau de parallélisation de la méthode, parfaitement semblable à celui du PCR+ILU1. En parallèle, le comportement des solveurs est différent, tant pour le PCR+ILU1 que pour les multigrilles, en raison du préconditionneur ILU utilisé dans les deux méthodes et qui diffère entre sa version séquentielle et sa version parallèle. Le préconditionneur multigrille faisant un usage intensif de ILU0 dans ses itérations de lissages, ses performances sont davantage affectées lors du passage du séquentiel au parallèle. La méthode multigrille reste néanmoins plus rapide très performante.

L'application du préconditionneur multigrille sur des simulations de procédés de mise en forme a finalement permis de mettre en évidence les possibilités effectives de cette méthode au sein du logiciel FORGE® NxT. Le forgeage du triaxe en séquentiel est ainsi accéléré par un facteur 1,3 ou 1,6 fois selon sa taille. Nous retrouvons pratiquement les mêmes accélérations que celles obtenues sur un seul incrément. Cette simulation a également révélé quelques points faibles de la méthode multigrille : une possible erreur dans le transfert des champs en présence d'un repli de matière et une baisse de performance lorsque l'écoulement devient très contraint en fin de simulation et que la matière s'écoule en bavure où les propriétés de la méthode multigrille ne peuvent être efficacement utilisées. Sur le forgeage du triaxe, comme sur le martelage rotatif au delà des bons résultats obtenus en terme d'accélération et de robustesse de la méthode, la problématique de déraffinement automatique n'a pas permis de conserver les performances de la méthode tout au long de la simulation. Ces exemples ont souligné l'importance des conclusions tirées dans le chapitre précédent quant à la taille et la qualité des grilles grossières ainsi qu'à l'homogénéité des facteurs de déraffinement. Cela n'a pas empêché la méthode multigrille sur le problème le plus massif et le plus coûteux numériquement du martelage rotatif, d'offrir en parallèle, sur 8 processeurs, une accélération de 1,6, faisant passer les temps de calcul de 4 jours et 14 heures à moins de 3 jours, et démontrant sa robustesse.

Les meilleures performances ont été obtenues sur le cas du laminage retour avec une accélération solveur par un facteur moyen de 2,4, qui s'explique probablement par la plus grande homogénéité de l'écoulement sur ce problème. Au final, l'accélération globale est semblable, de 1,6, limitée par le temps des autres opérations qui prennent une part plus importante que les résolutions linéaires avec le préconditionneur multigrille.

Enfin, nous avons vu que la génération de maillages supplémentaires nécessaires à la méthode multigrille présente un coût négligeable et que, malgré ce surcoût, des accélérations sont obtenues sur chacun des procédés traités.



# Conclusions et Perspectives

## Conclusion générale

L'enjeu de cette thèse était de contribuer à la réduction des temps de calcul pour les simulations de mise en forme. Constatant qu'au moins 70% du temps de calcul global est dédié à la résolution de systèmes linéaires, nous nous sommes intéressés au développement d'une méthode permettant d'accélérer ces résolutions en séquentiel comme en calcul parallèle. Notre choix s'est porté sur une méthode multigrille, dont les perspectives d'une complexité asymptotique en  $O(N \log N)$  promettaient une plus grande efficacité face à la méthode du résidu conjugué préconditionné utilisé aujourd'hui, qui présente un comportement asymptotique en  $O(N^{\frac{3}{2}})$ .

Nous avons vu que les grandes déformations subies par la pièce lors des problèmes de mise en forme impliquaient des remaillages fréquents et que la condition d'incompressibilité menait à des matrices indéfinies. La prise en compte du contact rendant impossible l'utilisation des multigrilles géométriques et la gestion de ce contact par pénalisation rendant complexe l'utilisation des multigrilles algébriques, nous avons fait le choix d'une méthode multigrille hybride. Nous avons présenté une approche de génération de grilles grossières par déraffinement de maillage non emboîté, permettant d'obtenir des maillages suffisamment grossiers pour permettre une résolution directe efficace tout en conservant avec qualité la géométrie initiale. En parallèle, cet algorithme permet de plus d'obtenir des partitions de maillages équilibrées sur chacun des niveaux. Afin de conserver les informations de contact et d'incompressibilité définies sur la grille la plus fine et d'éviter un assemblage du système sur chaque niveau grossier, nous avons opté pour une stratégie algébrique en construisant les matrices de calcul sur la grille grossière à partir des opérateurs de transfert.

La recherche des paramètres les plus adaptés et les plus performants pour des problèmes représentatifs de la mise en forme nous a permis de retenir une méthode multigrille utilisant un lisseur ILU0, avec une seule itération de pré-lissage et post-lissage. En effet, alors que les premiers tests sur un problème linéaire nous orientaient vers un lisseur Jacobi, classiquement utilisé dans la littérature, l'utilisation du lisseur ILU0, plus complexe, a permis de gagner en robustesse et de réduire les temps de calculs sur des problèmes non linéaires et plus représentatifs. De la même manière, sur un problème linéaire simple, l'utilisation en solveur de la méthode proposait les meilleurs résultats mais se révèle moins efficace qu'en préconditionneur sur des problèmes plus représentatifs. Enfin, bien que le préconditionneur multigrille offrait une bonne accélération avec le solveur GMRES sur les différents problèmes de mise en forme, nous nous orientons finalement vers un solveur résidu conjugué préconditionné, plus adapté et plus efficace.

Avec la méthode 3 grilles, l'utilisation de facteurs de déraffinement homogènes est nécessaire pour obtenir les meilleures performances. Ces facteurs de déraffinement doivent permettre d'atteindre une grille grossière la plus proche possible de 500 nœuds pour une résolution directe rapide et efficace en parallèle grâce à la librairie MUMPS.

L'expérience montre que la méthode multigrille développée répond aux attentes, tant au niveau de la robustesse que de l'efficacité. Elle présente une complexité asymptotique de  $O(N^{1,08})$ , quand elle est de  $O(N^{1,52})$  pour le solveur habituel. Cela permet, en séquentiel, d'obtenir dès 22 000 nœuds une accélération de 1,7, d'atteindre une accélération de 2,6 sur 160 000 nœuds et promet des accélérations encore meilleures sur plus de noeuds, si les ressources de calcul (notamment la mémoire) le permettent. En calcul parallèle, nous obtenons la même efficacité parallèle que le solveur de référence, avec un speed-up parallèle quasi optimal, malgré des difficultés provenant des différences entre les versions séquentielle et parallèle des préconditionneurs ILU. Nous répondons donc à ce qui représentait une difficulté majeure au début de cette thèse et permettons de combiner les accélérations multigrilles et parallèle. Par exemple, sur une résolution de système linéaire d'un problème à 310 000 nœuds, l'utilisation de la méthode multigrille en parallèle, utilisant 8 processeurs, permet de diviser le temps de calcul par près de 30 par rapport à la résolution de ce même système en séquentiel, avec le solveur de référence.

Sur les simulations complètes de procédés de mise en forme, soumises à des remaillages fréquents, des premières difficultés sont apparues avec l'augmentation du nombre de noeuds des maillages de calcul pendant une simulation. Compte tenu de l'importance de conserver des maillages grossiers à faible nombre de noeuds, il a été nécessaire d'adapter manuellement et à posteriori le facteur de déraffinement. Des réglages à effectuer sur le mailleur restent nécessaires pour obtenir des performances optimales de la méthode multigrille, d'autant plus que des particularités d'écoulement (bavure sur le forgeage du triaxe ou déformations très

localisées sur le martelage rotatif) rendent encore plus difficile la génération de maillages grossiers suivant les recommandations. Toutefois, sur tous les procédés abordés, le préconditionneur multigrille apporte d'excellents résultats. Tout d'abord, nous constatons que le surcoût apporté par la génération des maillages grossiers est négligeable, même avec de nombreux remaillages. Ensuite, et même si le préconditionneur multigrille n'intervient que sur les résolutions linéaires qui ne représentent que 70 à 80% du temps du temps de calcul global, les temps de calcul des différentes simulations sont significativement réduits.

Sur 60 000 nœuds en séquentiel, une accélération de 2,4 sur les résolutions linéaires est obtenue sur la simulation du laminage retour, permettant de réduire d'un facteur 1,6 le temps de calcul global. Cela représente un gain d'un peu plus d'une journée sur une simulation de 3 jours.

En parallèle, l'utilisation de la méthode multigrille sur la simulation d'un procédé de martelage rotatif à 214 000 nœuds sur 8 processeurs permet une division par 2 du temps de résolution de systèmes linéaires, correspondant à une accélération d'un facteur 1,6 sur le temps de calcul global et représentant un gain de près de 2 jours sur 4,5 jours de calcul.

## Perspectives

Ce travail de thèse permet d'ouvrir de larges perspectives pour aller plus loin dans la réduction des temps de calcul.

Tout d'abord, des améliorations peuvent être apportées au déraffinement de maillage. Nous avons vu que lorsque les grilles générées n'étaient pas suffisamment grossières, les performances chutaient. Relâcher davantage le critère de conservation du volume du domaine ou utiliser des facteurs de déraffinements locaux, basés sur un estimateur d'erreur et permettant de déraffiner plus ou moins sur différentes zones de la géométrie, sont deux pistes qu'il serait intéressant d'explorer pour mieux gérer les particularités d'écoulements. S'il s'avère impossible de générer des grilles suffisamment grossières, l'utilisation d'une méthode itérative sur la grille grossière peut également être envisagée, avec la nécessité de déterminer les paramètres de convergence de celle-ci.

Sur l'aspect automatique du déraffinement de maillage, il semble crucial de faire varier le facteur de déraffinement tout au long de la simulation, suivant l'évolution du nombre de nœuds du maillage de calcul après chaque remaillage. La connaissance de ce nombre de nœuds

permettra le calcul d'un nouveau facteur de déraffinement afin de toujours générer des grilles grossières comportant suffisamment peu de nœuds et de bonne qualité. De plus, parce que cette évolution du nombre de nœuds peut parfois être importante, une gestion automatique du nombre de niveaux de la méthode multigrille peut être envisagée, pour permettre par exemple, l'ajout d'un niveau supplémentaire en milieu de calcul. Pour cela, les méthodes considérant plus que 3 grilles pourront être implémentées, d'autant plus que nous avons vu que la saturation de la largeur de bande commençait à apparaître à partir de 5 grilles.

Afin d'évaluer davantage la méthode multigrille hybride développée, les performances, en parallèle comme en séquentiel, pourront être évaluées sur d'autres cas industriels liés à la mise en forme, mais aussi à d'autres types de problèmes. Nous avons en effet montré que le préconditionneur multigrille que nous avons développé pouvait offrir de bonnes performances avec une méthode GMRES, plus générale et non limitée aux matrices symétriques. De plus, les débuts de développement d'un solveur multigrille permettent d'aller encore plus loin dans cette extension.

Pour encore réduire les temps de calculs consacrés à la résolution linéaire, un travail sur les lisseurs ILU en parallèle pourrait être effectué, ceux-ci causant une baisse d'efficacité lors du passage de séquentiel vers le calcul parallèle. Enfin, de nouveaux travaux sur une méthode multigrille pourraient être effectués, par exemple sur l'utilisation d'une méthode entièrement algébrique et parallèle, adaptée aux problèmes de mise en forme, avec la difficulté de traiter la pénalisation provenant du contact.

Sur la problématique plus générale de réduction des temps de calcul, nous avons vu que l'accélération apportée par la méthode multigrille provoquait un changement dans la répartition des temps de calcul. Ainsi, le temps consacré aux résolutions linéaires, jusqu'à présent largement dominant, est maintenant équivalent, si ce n'est inférieur, au temps consacré aux autres opérations. Le solveur linéaire n'est donc plus la principale piste pour l'accélération des simulations et il est alors nécessaire de s'intéresser à d'autres types d'approches, comme la réduction de modèle par exemple.

# Bibliographie

- ADAMS, M. (2002). Evaluation of three unstructured multigrid methods on 3D finite element problems in solid mechanics. *International Journal for Numerical Methods in Engineering*, 55(5):519–534.
- ADAMS, M., BREZINA, M., HU, J. et TUMINARO, R. (2003). Parallel multigrid smoothing : polynomial versus Gauss–Seidel. *Journal of Computational Physics*, 188(2):593–610.
- ADAMS, M. F. et TAYLOR, R. (2000). Parallel multigrid solvers for 3D unstructured finite element problems in large deformation elasticity and plasticity. *International Journal for Numerical Methods in Engineering*, 48(8):1241–1262.
- AMESTOY, P. R., DUFF, I. S., KOSTER, J. et L’EXCELLENT, J.-Y. (2001). A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41.
- AMESTOY, P. R., GUERMOUCHE, A., L’EXCELLENT, J.-Y. et PRALET, S. (2006). Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136–156.
- ARNOLD, D. N., BREZZI, F. et FORTIN, M. (1984). A stable finite element for the Stokes equations. *CALCOLO*, 21(4):337–344.
- BABUŠKA, I. (1973). The finite element method with penalty. *Mathematics of computation*, 27(122):221–228.
- BALAY, S., ABHYANKAR, S., ADAMS, M. F., BROWN, J., BRUNE, P., BUSCHELMAN, K., DALCIN, L., EIJKHOUT, V., GROPP, W. D., KAUSHIK, D., KNEPLEY, M. G., MCINNES, L. C., RUPP, K., SMITH, B. F., ZAMPINI, S. et ZHANG, H. (2015a). PETSc users manual. Rapport technique ANL-95/11 - Revision 3.6, Argonne National Laboratory.
- BALAY, S., ABHYANKAR, S., ADAMS, M. F., BROWN, J., BRUNE, P., BUSCHELMAN, K., DALCIN, L., EIJKHOUT, V., GROPP, W. D., KAUSHIK, D., KNEPLEY, M. G., MCINNES, L. C., RUPP, K., SMITH, B. F., ZAMPINI, S. et ZHANG, H. (2015b). PETSc Web page.

- BALAY, S., GROPP, W. D., MCINNES, L. C. et SMITH, B. F. (1997). Efficient management of parallelism in object oriented numerical software libraries. In ARGE, E., BRUASET, A. M. et LANGTANGEN, H. P., éditeurs : *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press.
- CARTE, G., COUPEZ, T., GUILLARD, H. et LANTERI, S. (2000). Coarsening techniques in multigrid applications on unstructured meshes. In *European congress on computational methods in applied sciences and engineering, ECCOMAS*.
- CHENOT, J.-L., FOURMENT, L. et MOCELLIN, K. (2002). Numerical treatment of contact and friction in finite element simulation of forming processes. *Journal of Materials Processing Technology*, 125:45–52.
- CHOW, E., FALGOUT, R. D., HU, J. J., TUMINARO, R. S. et YANG, U. M. (2006). A survey of parallelization techniques for multigrid solvers. In HEROUX, M. A., RAGHAVAN, P. et SIMON, H. D., éditeurs : *Parallel Processing for Scientific Computing*, pages 179–201. Society for Industrial and Applied Mathematics.
- COUPEZ, T. (1991). *Grandes transformations et remaillage automatique*. Thèse de doctorat, École Nationale Supérieure des Mines de Paris.
- COUPEZ, T. (1994). A mesh improvement method for 3d automatic remeshing. *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, pages 615–626.
- COUPEZ, T. (2000). Génération de maillage et adaptation de maillage par optimisation locale. *Revue européenne des éléments finis*, 9(4):403–423.
- COUPEZ, T., DIGONNET, H. et DUCLOUX, R. (2000). Parallel meshing and remeshing. *Applied Mathematical Modelling*, 25(2):153–175.
- COUPEZ, T. et MARIE, S. (1997). From a direct solver to a parallel iterative solver in 3D forming simulation. *International Journal of High Performance Computing Applications*, 11(4):277–285.
- DELAUNAY, B. (1934). Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2.
- DIGONNET, H. (2001). *Repartitionnement dynamique, mailleur parallèle et leurs applications à la simulation numérique en mise en forme des matériaux*. Thèse de doctorat, École Nationale Supérieure des Mines de Paris.
- DIGONNET, H., BERNACKI, M., SILVA, L. et COUPEZ, T. (2007). Adaptation de maillage en parallèle, application à la simulation de la mise en forme des matériaux. *18ème Congrès Français de Mécanique (Grenoble 2007)*.
- FARHAT, C., MANDEL, J. et ROUX, F. X. (1994). Optimal convergence properties of the feti domain decomposition method. *Computer methods in applied mechanics and engineering*, 115(3-4):365–385.

- FEDORENKO, R. P. (1961). A relaxation method for solving elliptic difference equations. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 1(5):922–927.
- FENG, Y. T., PERIĆ, D. et OWEN, D. R. J. (1997). A non-nested Galerkin multi-grid method for solving linear and nonlinear solid mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 144(3–4):307–325.
- FOURMENT, L., CHENOT, J.-L. et MOCELLIN, K. (1999). Numerical formulations and algorithms for solving contact problems in metal forming simulation. *International Journal for Numerical Methods in Engineering*, 46(9):1435–1462.
- FREY, P. et GEORGE, P. (1999). *Maillages : applications aux éléments finis*. Hermès Science Publications.
- GEORGE, P. (2001). *Maillage et adaptation*. LAVOISIER.
- GHIA, U., GHIA, K. et SHIN, C. (1982). High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48(3):387 – 411.
- GMEINER, B., RUDE, U., STENGEL, H., WALUGA, C. et WOHLMUTH, B. (2015). Performance and scalability of hierarchical hybrid multigrid solvers for stokes systems. *SIAM Journal on Scientific Computing*, 37(2):C143–C168.
- GRUAU, C. et COUPEZ, T. (2005). 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric. *Computer Methods in Applied Mechanics and Engineering*, 194(48–49):4951–4976.
- IWAMURA, C., COSTA, F. S., SBARSKI, I., EASTON, A. et LI, N. (2003). An efficient algebraic multigrid preconditioned conjugate gradient solver. *Computer Methods in Applied Mechanics and Engineering*, 192(20):2299–2318.
- KARASEVA, O. (2005). *Déformations élastiques des presses de forgeage et calcul parallèle*. Thèse de doctorat, École Nationale Supérieure des Mines de Paris.
- KIMMRTZ, M. et RICHTER, T. (2011). Parallel multigrid method for finite element simulations of complex flow problems on locally refined meshes. *Numerical Linear Algebra with Applications*, 18(4):615–636.
- KPODZO, K. W. (2014). *Accélération des calculs pour la simulation du laminage à pas de pèlerin en utilisant la méthode multimaillages*. Thèse de doctorat, Ecole Nationale Supérieure des Mines de Paris.
- LEBON, F., RAOUS, M. et ROSU, I. (2007). Multigrid methods for unilateral contact problems with friction. In *IUTAM Symposium on Computational Methods in Contact Mechanics*, pages 1–16. Springer.
- LEE, C.-O. (1998). Multigrid methods for the pure traction problem of linear elasticity : mixed formulation. *SIAM journal on numerical analysis*, 35(1):121–145.

- MARIE, S. (1997). *Un Modèle de parallélisation SPMD Pour la simulation numérique de procédés de mise en forme de matériaux*. Thèse de doctorat, École Nationale Supérieure des Mines de Paris.
- MOCELLIN, K. (1999). *Contribution à la simulation numérique tridimensionnelle du forgeage à chaud : Etude du contact et calcul multigrille*. Thèse de doctorat, Ecole Nationale Supérieure des Mines de Paris.
- MOCELLIN, K., FOURMENT, L., COUPEZ, T. et CHENOT, J.-L. (2001). Toward large scale FE computation of hot forging process using iterative solvers, parallel computation and multigrid algorithms. *International Journal for Numerical Methods in Engineering*, 52(5-6):473–488.
- NOTAY, Y. (2016). A new algebraic multigrid approach for stokes problems. *Numerische Mathematik*, 132(1):51–84.
- PARSONS, I. et HALL, J. (1990a). The multigrid method in solid mechanics : part ii—practical applications. *International Journal for Numerical Methods in Engineering*, 29(4):739–753.
- PARSONS, I. D. et HALL, J. F. (1990b). The multigrid method in solid mechanics : Part I—Algorithm description and behaviour. *International Journal for Numerical Methods in Engineering*, 29(4):719–737.
- PERCHAT, E. (2000). *MINI-élément et factorisation incomplète pour la parallélisation d'un solveur de Stokes 2D : application au forgeage*. Thèse de doctorat, École Nationale Supérieure des Mines de Paris.
- PÉRÉMÉ, M., MARIE, S., BARBELET, M., PERCHAT, E., DUCLOUX, R., FOURMENT, L. et CHENOT, J.-L. (2011). Benefits of High Performance Computing applied to the numerical simulation of forged parts. In *20th International Forging Congress*, Hyderabad, India.
- RAMADAN, M. (2010). *Une méthode MultiMaillages MultiPhysiques parallèle pour accélérer les calculs des procédés incrémentaux*. Thèse de doctorat, École Nationale Supérieure des Mines de Paris.
- RASSINEUX, A. (1995). *Maillage automatique tridimensionnel par une méthode frontale pour la méthode des éléments finis*. Thèse de doctorat.
- REY, B. (2007). *Méthode multigrilles pour les grandes déformations et simulation numérique du procédé de roulage*. Thèse de doctorat, Ecole Nationale Supérieure des Mines de Paris.
- RUGE, J. W. et STÜBEN, K. (1987). Multigrid methods.
- SAAD, Y. (2003). 13. multigrid methods. In *Iterative Methods for Sparse Linear Systems*, pages 407–449. Society for Industrial and Applied Mathematics, second édition.

- SAAD, Y. et SCHULTZ, M. H. (1986). Gmres : A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869.
- STÜBEN, K. (2001). A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, 128(1):281–309.
- STRANG, G. (2007). *Computational Science and Engineering*. Wellesley-Cambridge Press.
- TRAORÉ, K. (2001). *Simulation thermomécanique du laminage circulaire : développement d'une formulation quasi-eulérienne tridimensionnelle sur une architecture parallèle*. Thèse de doctorat, École Nationale Supérieure des Mines de Paris.
- VANKA, S. (1986). Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *Journal of Computational Physics*, 65(1):138–158.
- WATANABE, K., IGARASHI, H. et HONMA, T. (2005). Comparison of geometric and algebraic multigrid methods in edge-based finite-element analysis. *IEEE transactions on magnetics*, 41(5):1672–1675.
- WESSELING, P. (1992). *An introduction to Multigrid Methods*. John Wiley & Sons, Chichester.
- YAVNEH, I. (2006). Why Multigrid Methods Are So Efficient. *Computing in Science & Engineering*, 8(6):12–22.
- YERRY, M., SHEPHARD, M. S. *et al.* (1983). Modified quadtree approach to finite element mesh generation. *IEEE COMP. GRAPHICS & APPLIC.*, 3(1):39–46.



## Résumé

Cette thèse porte sur le développement d'une méthode multigrilles parallèle visant à réduire les temps de calculs des simulations éléments finis dans le domaine de la mise en forme de pièces forgées en 3D. Ces applications utilisent une méthode implicite, caractérisées par une formulation mixte en vitesse/pression et une gestion du contact par pénalisation. Elles impliquent de grandes déformations qui rendent nécessaires des remaillages fréquents sur les maillages tétraédriques non structurés utilisés. La méthode multigrilles développée suit une approche hybride, se basant sur une construction géométrique des niveaux grossiers par déraffinement de maillage non emboîtés et sur une construction algébrique des systèmes linéaires intermédiaires et grossiers. Un comportement asymptotique quasi-linéaire et une bonne efficacité parallèle sont attendus afin de permettre la réalisation de simulations à grand nombre de degrés de liberté dans des temps plus raisonnables qu'aujourd'hui. Pour cela, l'algorithme de déraffinement de maillages est compatible avec le calcul parallèle, ainsi que les opérateurs permettant les transferts de champs entre les différents niveaux de maillages partitionnés. Les spécificités des problèmes à traiter ont mené à la sélection d'un lisseur plus complexe que ceux utilisés plus fréquemment dans la littérature. Sur la grille la plus grossière, une méthode de résolution directe est utilisée, en séquentiel comme en calcul parallèle. La méthode multigrille est utilisée en tant que préconditionneur d'une méthode de résidu conjugué et a été intégrée au logiciel FORGE NxT et montre un comportement asymptotique et une efficacité parallèle proches de l'optimal. Le déraffinement automatique de maillages permet une compatibilité avec les remaillages fréquents et permet à la méthode multigrilles de simuler un procédé du début à la fin. Les temps de calculs sont significativement réduits, même sur des simulations avec des écoulements particuliers, sur lesquelles la méthode multigrilles ne peut être utilisée de manière optimale. Cette robustesse permet, par exemple, de réduire de 4,5 à 2,5 jours le temps de simulation d'un procédé.

## Mots Clés

Méthode multigrilles, Calcul parallèle, Forgeage, Remaillage automatique, Déraffinement de maillage, Éléments finis, Solveur linéaire

## Abstract

A parallel multigrid method is developed to reduce large computational costs involved by the finite element simulation of 3D metal forming applications. These applications are characterized by a mixed velocity/pressure implicit formulation with a penalty formulation to enforce contact and lead to large deformations, handled by frequent remeshings of unstructured meshes of tetrahedral. The developed multigrid method follows a hybrid approach where the different levels of non-nested meshes are geometrically constructed by mesh coarsening, while the linear systems of the intermediate and coarse levels result from the algebraic approach. A close to linear asymptotical behavior is expected along with parallel efficiency in order to allow simulations with large number of degrees of freedom under reasonable computation times. These objectives lead to a parallel mesh coarsening algorithm and parallel transfer operators allowing fields transfer between the different levels of partitioned meshes. Physical specificities of metal forming applications lead to select a more complex multigrid smoother than those classically used in literature. A direct resolution method is used on the coarsest mesh, in sequential and in parallel computing. The developed multigrid method is used as a preconditioner for a Conjugate Residual algorithm within FORGE NxT software and shows an asymptotical behavior and a parallel efficiency close to optimal. The automatic mesh coarsening algorithm enables compatibility with frequent remeshings and allows the simulation of a forging process from beginning to end with the multigrid method. Computation times are significantly reduced, even on simulations with particular material flows on which the multigrid method is not optimal. This robustness allows, for instance, reducing from 4.5 to 2.5 days the computation of a forging process.

## Keywords

Multigrid method, Parallel Computing, Metal Forming, Automatic Remeshing, Mesh Coarsening, Finite Element, Linear Solver