



HAL
open science

Contribution à la simulation numérique tridimensionnelle du forgeage à chaud : étude du contact et calcul multigrille

Katia Mocellin

► To cite this version:

Katia Mocellin. Contribution à la simulation numérique tridimensionnelle du forgeage à chaud : étude du contact et calcul multigrille. Mécanique [physics.med-ph]. École Nationale Supérieure des Mines de Paris, 1999. Français. NNT: . tel-00480385

HAL Id: tel-00480385

<https://pastel.hal.science/tel-00480385>

Submitted on 4 May 2010

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

présentée à

L'ÉCOLE NATIONALE SUPÉRIEURE DES MINES DE PARIS

par

Katia MOCELLIN

en vue de l'obtention du titre de

DOCTEUR

en

SCIENCES ET GÉNIE DES MATÉRIAUX

**CONTRIBUTION À LA SIMULATION NUMÉRIQUE
TRIDIMENSIONNELLE DU FORGEAGE À CHAUD :**

ÉTUDE DU CONTACT ET CALCUL MULTIGRILLE

soutenue le 12 juillet 1999 devant le jury composé de:

Mr	Michel RAOUS	Président
Mr	Jean Claude GELIN	Rapporteur
Mr	Hervé GUILLARD	Rapporteur
Mme	Isabelle POITRAULT	
Mr	Jean-Loup CHENOT	
Mr	Lionel FOURMENT	Directeur de thèse

Table des matières

INTRODUCTION	1
La forge en France	1
Avantages métallurgique des pièces forgées	1
Caractère industriel du procédé	2
Le forgeage à chaud	2
La modélisation	3
Objectifs de ce travail	4
1 MODÉLISATION	7
1.1 Le problème mécanique	7
1.1.1 Les équations d'équilibre	7
1.1.2 Les lois de comportement	8
1.1.3 Le système d'équations à résoudre	10
1.1.4 Le formalisme	10
1.2 La discrétisation du problème	11
1.2.1 La forme faible du problème	11
1.2.2 La discrétisation éléments finis	12
1.2.3 Le problème non linéaire	15
1.3 La résolution du système linéaire	18
1.3.1 Stockage de la matrice	18
1.3.2 La méthode de résolution	18
1.3.3 La discrétisation temporelle	18
2 LA GESTION DU CONTACT	21
2.1 Les équations du contact	23
2.1.1 Le traitement du contact en quasi stationnaire	23
2.1.2 La gestion incrémentale du contact	29
2.2 Le contact implicite	34

2.2.1	Principe de la méthode	34
2.2.2	Robustesse de la méthode	35
2.2.3	La procédure de construction des normales	37
2.2.4	Une méthode de lissage pour les angles concaves et convexes	40
2.2.5	Applications	41
2.2.6	Conclusions	44
3	LES MÉTHODES MULTIGRILLES	47
3.1	La méthode de résidu minimal	48
3.1.1	Présentation du problème	48
3.1.2	Préconditionnement	50
3.1.3	Résultats	53
3.2	Les méthodes multigrilles : présentation	59
3.2.1	Principe	59
3.2.2	Construction du problème grossier	62
3.2.3	Le lissage, la correction et les opérateurs	65
3.3	les méthodes multigrilles et les problèmes incompressibles	67
3.3.1	Le problème de Stokes	67
3.3.2	Les méthodes découplées	67
3.4	Mise en place d'une méthode multigrille 3D : Application à la simulation du forgeage à chaud	69
3.4.1	Les différents niveaux de calcul	69
3.4.2	Mise en place de la méthode	74
3.4.3	Test des méthodes 3-grilles	78
3.4.4	Les améliorations du solveur bigrille	81
3.4.5	La procédure de remaillage	90
3.5	Résultats	92
3.5.1	Résultats sur le forgeage du triaxe	92
3.5.2	Résultats sur le martelage d'une tôle épaisse	94
3.5.3	Résultats sur un emboutissage de plaque mince	95
3.6	Conclusion	98
3.7	Perspectives	99
4	CONCLUSION GÉNÉRALE	101
4.1	Conclusion	101
4.2	Perspectives	102

INTRODUCTION

La forge en France

Répartie sur 95 sites, la Forge française représente plus de 445 000 tonnes de pièces livrées dont 32,4% à l'export (chiffres SNEF 1997). Ceci représente une facture de plus de 7300 millions de francs. Ces premiers chiffres suffisent à montrer l'importance de ce secteur d'activité dans l'économie nationale. Le secteur automobile est un gros consommateur de pièces forgées puisqu'il représente 64,5 % du marché et qu'une partie de la production (138 800 tonnes) est réalisée par des forges automobiles intégrées. Le reste de la production est répartie entre la construction mécanique, l'aéronautique, l'ingénierie (robinetterie, pétrole), le matériel TP, le matériel agricole, l'armement et le ferroviaire (voir graphique 1).

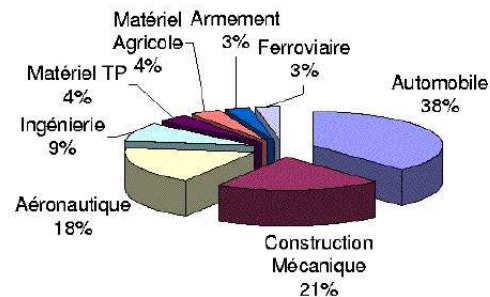


FIG. 1 – Les marchés de la forge en % des ventes (hors forges intégrées) (chiffres SNEF 1997)

Avantages métallurgique des pièces forgées

Le procédé de mise en forme par forgeage est très utilisé car les pièces obtenues présentent une résistance supérieure à celles fabriquées avec d'autres méthodes (usinage, fonderie,...) à masses équivalentes. Elles présentent, en effet, des qualités de tenue en fatigue et de résistance aux chocs bien meilleures, dues à des propriétés métallurgiques acquises pendant la mise en forme.

Caractère industriel du procédé

La mise en forme par forgeage se prête très bien à l'environnement industriel. Les cadences de fabrication peuvent être élevées (production automobile...). La souplesse de production est considérable puisque l'on forge des pièces de quelques grammes à plusieurs tonnes (voir figure 2). Enfin ce procédé est applicable à une large gamme d'aciers et d'alliages.



FIG. 2 – Exemple de matrice utilisée en forgeage (train avant d'airbus)- Document SNEF

Toutefois le procédé de mise en forme par forgeage reste très coûteux en comparaison à d'autres méthodes comme la fonderie (perte de matière à l'ébavurage, coût d'usinage, coût des outils). Il faut donc optimiser le procédé au maximum pour rester compétitif.

Le forgeage à chaud

On trouve sous cette dénomination des méthodes de mise en forme très différentes.

- *La forge par estampage* concerne les métaux ferreux produits avec des outillages spécifiques. On l'utilise donc pour des pièces produites en petite, moyenne ou grande séries (voir figure 3).

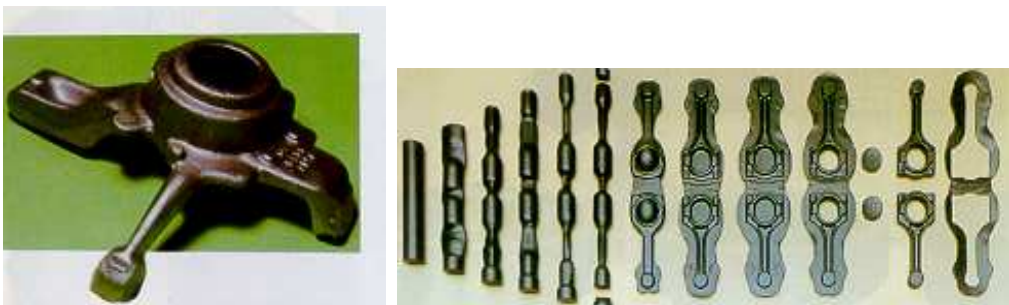


FIG. 3 – Exemples d'estampage

- *La forge libre* concerne des pièces brutes dont le poids est inférieur à 3 tonnes. Elle permet de réaliser des ébauches de prototypes ou de pièces faites à l'unité ou en très faibles séries. Les outillages ne sont pas spécifiques aux pièces forgées.



FIG. 4 – Exemple de forge libre

- *Le laminage circulaire* permet de fabriquer des couronnes.
- *La forge par matriçage* ne se distingue de la forge par estampage que par le fait qu'elle concerne des matériaux non ferreux.

La modélisation

La mise au point d'une gamme de forgeage est difficile car les pièces obtenues doivent vérifier des propriétés de géométries et de métallurgie très précises. Pour obtenir ce type de prédiction, la solution souvent utilisée est la simulation sur matériaux modèles (voir figure 5). Mais si ce type de méthode permet de prévoir la géométrie, elle ne permet pas de visua-

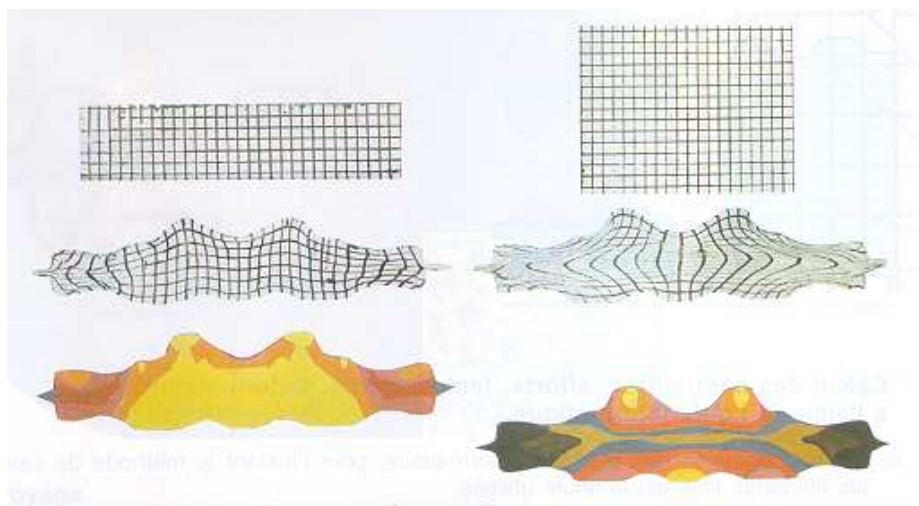


FIG. 5 – Simulation par matériau modèle

liser les zones où les contraintes sur la pièce sont très élevées. Si ce type de phénomène se produit sur la surface, il peut en découler un affaiblissement de l'outillage. Pour les grandes productions (par exemple les forges automobiles intégrées), il est primordial de concevoir des matrices dont la durée de vie soit maximale. Une autre méthode est la simulation numérique. Le logiciel **Forge2** de simulation numérique bidimensionnelle donne des résultats très satisfaisants pour la modélisation de forgeage de pièces axisymétriques ou planes. A

partir de ce logiciel ont été développés différents modules comme de l'optimisation de forme des matrices de forgeage [Balan96], des calculs couplés avec les outillages (déformation des outillages, calcul d'outils frettés,...) [Miles95], [Acr297]. Ce logiciel est désormais largement utilisé dans les forges françaises et permet une bonne prédiction. Toutefois, de nombreuses pièces présentent des caractéristiques 3D comme on peut le voir sur la figure 3.

Le logiciel **Forge3** est développé pour simuler ce type de pièces depuis environ 15 ans au Cemef et à Transvalor. Toutefois, la simulation 3D engendre des problèmes plus lourds que la modélisation 2D puisque la taille des problèmes considérés (maillages) est bien plus grande et que les phénomènes purement 3D demandent des techniques de calculs différentes. Le principal problème rencontré est le temps de calcul très élevé qui résulte du nombre de degrés de liberté. Une première étape pour réduire ce temps de calcul a été la mise en place d'un solveur parallèle [Marie97]. Ainsi, pour le cas de forgeage présenté sur la figure 6, le calcul sur 4 processeurs d'une station DIGITAL Modèle 4100 demande 135 heures de calcul. Le lopin initial comporte 49384 éléments et le lopin final 107122.

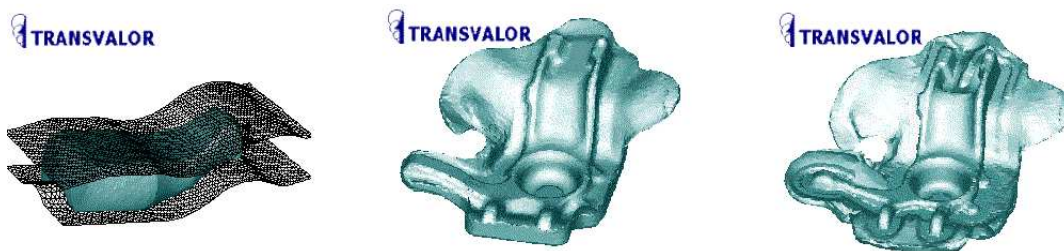


FIG. 6 – Simulation numérique du forgeage d'un pivot de fusée par **Forge3**, préforme, configuration intermédiaire et configuration finale

On voit donc que les délais de calcul sont encore importants et qu'ils constituent toujours un frein à l'utilisation de la simulation numérique comme outil prédictif.

Dans ce qui précède, on a noté que les pièces forgées doivent vérifier des géométries très strictes. Il faut donc que les résultats obtenus par la simulation soient également très précis au niveau de la surface de contact entre la pièce et l'outil. Un premier point est donc que l'outil soit modélisé de façon très fine et dans un second temps, que la méthode de gestion du contact puisse garantir la précision requise

Objectifs de ce travail

Cette thèse s'inscrit dans un contrat de recherche, le GPI ACR2 (Grand Projet Innovant - Action Concertée de Recherche 2), liant 16 entreprises représentant l'essentiel du potentiel français de forge, 10 laboratoires, et bénéficiant du soutien du Ministère de l'Industrie. L'ACR2 s'appuie sur la volonté des principaux forgerons français de considérer la simulation numérique du forgeage comme un axe de progrès stratégique et vital pour l'avenir de cette profession. Elle se décompose en 7 programmes de travail. Cette thèse s'inscrit dans le groupe de travail "Evolution de Forge3".

Dans ce contexte industriel, le but de nos travaux est double. Dans un premier temps, on souhaite améliorer la précision des calculs. Pour cela, on envisage d'améliorer la méthode de gestion du contact pour mieux contrôler la précision. Le second point est la diminution

du temps du calcul. Pour cela, nous avons choisi de mettre en place un nouveau solveur plus compétitif, tout en gardant la compatibilité avec le calcul parallèle.

Le premier chapitre de ce document présente les équations du problèmes de forgeage, le point de départ de notre travail. Dans le deuxième on exposera l'algorithme de contact et dans le troisième, le développement d'une méthode multigrille.

Chapitre 1

MODÉLISATION

1.1 Le problème mécanique

Nous allons décrire les équations qui modélisent le problème mécanique de la mise en forme par forgeage à chaud des métaux. Nous nous limiterons à une description isotherme. Le lecteur intéressé pourra se reporter à [Soyris90] pour la description du cas anisotherme. Cette partie présente les différents éléments de la modélisation de la mise en forme du code **Forge3**. Elle est nécessaire pour introduire les améliorations qu'on lui a apporté.

Cependant, nous n'avons pas abordé tous les aspects de la simulation du forgeage à chaud. Le remaillage automatique n'est, par exemple, pas développé. Le lecteur pourra assouvir sa curiosité sur le sujet en se référant à [Coupez91] ou [Coupez94]. Nous noterons en gras les variables vectorielles et tensorielles ($\mathbf{v}, \boldsymbol{\sigma}, \dots$), contrairement aux variables scalaires (p, \dots).

1.1.1 Les équations d'équilibre

Le lopin est considéré comme un domaine Ω de frontière $\partial\Omega$. Pour modéliser le problème, on a choisi d'exprimer les équations en tout point de Ω en fonction du champ de vitesse \mathbf{v} et du champ de pression p . Ce choix est dicté par l'utilisation d'une formulation à deux champs (vitesse et pression) de nos équations.

1.1.1.1 L'équilibre dynamique

Dans le cadre particulier du forgeage, la pièce ne doit sa déformation qu'à l'action de l'outillage. C'est-à-dire que l'action de l'inertie et de la gravité sont en général négligeables aux vitesses standard de mise en forme. L'équation d'équilibre se résume donc à l'équation suivante :

$$\mathbf{div}\boldsymbol{\sigma} = 0$$

où $\boldsymbol{\sigma}$ est le tenseur des contraintes de Cauchy.

1.1.1.2 L'incompressibilité

Le métal chauffé est un matériau incompressible et homogène. Pour respecter cette propriété, il faut ajouter au problème l'équation de conservation de la matière qui se

traduit par le fait que la divergence du champ de vitesse est nulle :

$$\operatorname{div} \mathbf{v} = 0 \quad (1.1)$$

1.1.1.3 Les conditions limites

On divise la surface de l'outillage en deux parties distinctes

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

– $\partial\Omega_l$ Surface libre

Sur la surface libre, la pièce n'est soumise à aucune contrainte, ce qui se traduit par la condition :

$$\boldsymbol{\sigma} \mathbf{n} = \mathbf{0}$$

où \mathbf{n} est la normale sortante à la pièce (voir figure 1.1).

– $\partial\Omega_c$ Surface en contact avec l'outil

On distingue deux types de conditions imposées sur cette surface :

– Une condition de frottement qui permet d'exprimer la valeur de la cission de frottement $\boldsymbol{\tau}$ définie par :

$$\boldsymbol{\tau} = \boldsymbol{\sigma} \mathbf{n} - (\boldsymbol{\sigma} \mathbf{n} \cdot \mathbf{n}) \mathbf{n}$$

Cette condition est tangentielle et est associée à une loi tribologique (voir plus loin).

– Une condition de contact qui exprime la non pénétration des nœuds dans l'outil. Elle est décrite par les équations suivantes appelées *conditions de Signorini*

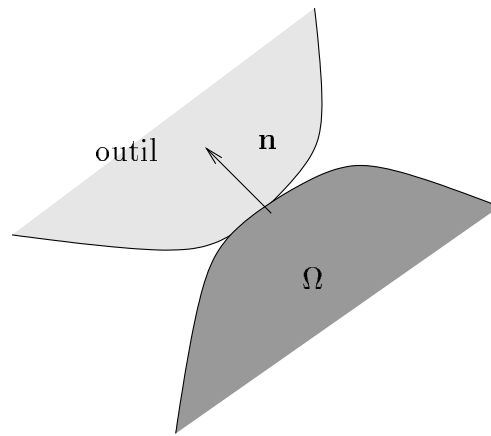
$$\begin{aligned} (\mathbf{v} - \mathbf{v}^{outil}) \cdot \mathbf{n} &\leq 0 \\ \sigma_n &\leq 0 \\ [(\mathbf{v} - \mathbf{v}^{outil}) \cdot \mathbf{n}] \sigma_n &= 0 \end{aligned} \quad (1.2)$$

où \mathbf{v}^{outil} est la vitesse de l'outil et $\sigma_n = \boldsymbol{\sigma} \mathbf{n} \cdot \mathbf{n}$ est la pression de contact. La condition de contact est une condition portée par la normale à la surface de la pièce.

1.1.2 Les lois de comportement

1.1.2.1 La loi rhéologique

Dans les conditions du forgeage à chaud, le métal se comporte généralement comme un matériau viscoplastique isotrope. En effet, compte tenu de la déformation plastique subie par la pièce, la composante élastique de la déformation est considérée comme négligeable. On utilise une loi de type Norton-Hoff pour modéliser le comportement. Cette loi relie le déviateur des contraintes \mathbf{s} au tenseur des vitesses de déformation $\dot{\boldsymbol{\epsilon}}$ par une expression non

FIG. 1.1 – Définition de la normale \mathbf{n}

linéaire fonction de la vitesse de déformation généralisée $\dot{\bar{\epsilon}}$, de la consistance du matériau K et de la sensibilité à la vitesse de déformation m :

$$\mathbf{s} = 2K(\sqrt{3}\dot{\bar{\epsilon}})^{(m-1)}\dot{\boldsymbol{\epsilon}} \quad (1.3)$$

on rappelle que sous l'hypothèse des petites déformations, le tenseur $\dot{\boldsymbol{\epsilon}}$ est le tenseur des vitesses de déformation linéarisé.

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2} (\mathbf{Grad} \mathbf{v} + \mathbf{Grad} \mathbf{v}^t)$$

et

$$\boldsymbol{\sigma} = \mathbf{s} - pI \quad (1.4)$$

où p la pression hydrostatique définie par $p = -\frac{1}{3}\text{Trace}(\boldsymbol{\sigma})$ et I est le tenseur identité.

La vitesse de déformation généralisée est définie par :

$$\dot{\bar{\epsilon}} = \sqrt{\frac{2}{3}\dot{\boldsymbol{\epsilon}} : \dot{\boldsymbol{\epsilon}}}$$

La consistance K du matériau peut-être fonction de la déformation généralisée $\bar{\epsilon}$ et de la température. On se place ici dans le cadre isotherme et on considère que le matériau ne présente pas d'écrouissage.

1.1.2.2 La loi de frottement

Pour traduire l'action du frottement, on utilise une loi de frottement de type Norton. Cette loi est bien adaptée au cas du forgeage. Elle relie la valeur de la cission de frottement $\boldsymbol{\tau}$ à la vitesse de glissement de l'outil $\Delta\mathbf{v}_g$ par le biais de l'expression suivante :

$$\boldsymbol{\tau} = -\alpha_f K \|\Delta\mathbf{v}_g\|^{(p_f-1)} \Delta\mathbf{v}_g \quad (1.5)$$

où

$$\Delta\mathbf{v}_g = \mathbf{v} - \mathbf{v}^{outil} - [(\mathbf{v} - \mathbf{v}^{outil}) \cdot \mathbf{n}] \mathbf{n}$$

Cette loi dépend de deux paramètres de frottement, α_f appelé coefficient de frottement et p_f coefficient de sensibilité à la vitesse de glissement. Souvent, on choisit $p_f = m$, la loi de frottement n'est alors plus dépendante que d'un seul paramètre.

1.1.3 Le système d'équations à résoudre

On aboutit donc au système d'équations suivant :

$$\begin{cases} \operatorname{div} \boldsymbol{\sigma} = 0 & \text{sur } \Omega \\ \operatorname{div} \mathbf{v} = 0 & \text{sur } \Omega \\ \boldsymbol{\tau} = -\alpha_f K \|\Delta \mathbf{v}_g\|^{(p_f-1)} \Delta \mathbf{v}_g & \text{sur } \partial\Omega_c \\ (\mathbf{v} - \mathbf{v}^{outil}) \cdot \mathbf{n} \leq 0 & \text{sur } \partial\Omega_c \end{cases}$$

si on utilise la décomposition de $\boldsymbol{\sigma}$ en fonction de \mathbf{s} et p , on peut aussi écrire :

$$\begin{cases} \operatorname{div} \mathbf{s} - \mathbf{Grad} p = 0 \\ \operatorname{div} \mathbf{v} = 0 \\ \boldsymbol{\tau} = -\alpha_f K \|\Delta \mathbf{v}_g\|^{(p_f-1)} \Delta \mathbf{v}_g & \text{sur } \partial\Omega_c \\ (\mathbf{v} - \mathbf{v}^{outil}) \cdot \mathbf{n} \leq 0 & \text{sur } \partial\Omega_c \end{cases} \quad (1.6)$$

1.1.4 Le formalisme

Au vu des déformations entre le début et la fin de la mise en forme, le forgeage se présente comme un procédé clairement instationnaire. Pour modéliser cette évolution, il faut introduire un formalisme eulérien ou lagrangien. Le choix fait ici est une gestion par lagrangien réactualisé. Si on divise la durée du forgeage en intervalles de temps $[t_n, t_{n+1}]$, la configuration à t_{n+1} sera calculée à partir de la configuration à t_n et non par rapport à la configuration initiale t_0 comme en lagrangien total. Pour calculer la nouvelle configuration par rapport à l'ancienne, il faut introduire un schéma d'intégration en temps.

1.2 La discrétisation du problème

On se limitera, dans la première partie de ce paragraphe, aux équations correspondant à une rhéologie newtonienne c'est-à-dire où $m = p_f = 1$ dans les équations (1.3) et (1.5). On raisonne ainsi sur un système linéaire en vitesse et en pression. Les particularités liées au comportement viscoplastique et au frottement de Norton (traitement des non linéarités de la matrice) seront abordées dans le paragraphe 1.2.3.

1.2.1 La forme faible du problème

1.2.1.1 Simplification du terme de contact

Notons que, dans la suite de ce chapitre, on considérera une condition de contact bilatéral. C'est à dire que, sur la surface de contact, on impose que la vitesse de la pièce suivant la direction normale soit égale à la vitesse de l'outil. On remplace donc l'inégalité de l'équation (1.2) par une égalité.

$$(\mathbf{v} - \mathbf{v}^{outil}) \cdot \mathbf{n} = 0$$

Pour ne pas alourdir la présentation avec les termes de contact, on introduit les contraintes que doit vérifier le champ de vitesse dans l'espace des fonctions tests pour la vitesse. Le traitement du contact sera, par ailleurs, présenté dans le chapitre suivant.

Les fonctions tests utilisées évoluent dans les espaces \mathcal{V}_0^{ca} et \mathcal{P} tels que :

$$\begin{aligned} \mathcal{V}^{ca} &= \{ \mathbf{v} \in (H^1(\Omega))^3 / (\mathbf{v} - \mathbf{v}^{outil}) \cdot \mathbf{n} = 0 \text{ sur } \partial\Omega_c \} \\ \mathcal{V}_0^{ca} &= \{ \mathbf{v} \in (H^1(\Omega))^3 / \mathbf{v} \cdot \mathbf{n} = 0 \text{ sur } \partial\Omega_c \} \\ \mathcal{P} &= L^2(\Omega) \end{aligned}$$

1.2.1.2 Les équations

On écrit le tenseur des contraintes $\boldsymbol{\sigma}$ en fonction de \mathbf{s} et p , on intègre par partie et on obtient le problème suivant :

$$\left| \begin{array}{l} \text{Trouver } (\mathbf{v}, p) \in (\mathcal{V}^{ca}, \mathcal{P}) \text{ tel que } \forall (\mathbf{v}^*, p^*) \in (\mathcal{V}_0^{ca}, \mathcal{P}) \\ \left\{ \begin{array}{l} \int_{\Omega} \mathbf{s} : \dot{\boldsymbol{\varepsilon}}^* d\omega - \int_{\Omega} p \operatorname{div} \mathbf{v}^* d\omega - \int_{\partial\Omega_c} \boldsymbol{\tau} \cdot \mathbf{v}^* dS = 0 \\ \int_{\Omega} p^* \operatorname{div} \mathbf{v} d\omega = 0 \end{array} \right. \end{array} \right.$$

Pour une rhéologie newtonienne, le déviateur \mathbf{s} s'exprime en fonction du tenseur des vitesses de déformation $\dot{\boldsymbol{\varepsilon}}$ par la formule :

$$\mathbf{s} = 2\eta\dot{\boldsymbol{\varepsilon}}$$

où η est la viscosité du matériau.

La loi de frottement s'exprime de la même façon par :

$$\boldsymbol{\tau} = -\alpha_f \eta \Delta \mathbf{v}_g$$

Ce qui donne les équations faibles suivantes à résoudre :

$$\begin{cases} \int_{\Omega} 2\eta \dot{\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}}^* d\omega - \int_{\Omega} p \operatorname{div} \mathbf{v}^* d\omega - \int_{\partial\Omega_c} \boldsymbol{\tau} \cdot \mathbf{v}^* dS = 0 \\ \int_{\Omega} p^* \operatorname{div} \mathbf{v} d\omega = 0 \end{cases}$$

1.2.1.3 Le lagrangien

On introduit le lagrangien du problème \mathcal{L} , défini par :

$$\mathcal{L}(\mathbf{v}, p) = \int_{\Omega} \frac{3}{2} \eta \dot{\boldsymbol{\varepsilon}}^2 d\omega - \int_{\Omega} p \operatorname{div} \mathbf{v} d\omega - \int_{\partial\Omega_c} \frac{\eta}{2} \alpha_f \|\Delta \mathbf{v}_g\|^2 dS \quad (1.7)$$

La solution du problème est la solution du problème de point selle :

$$\inf_{\mathbf{v} \in \mathcal{V}^{ca}} \sup_{p \in \mathcal{P}} \mathcal{L}(\mathbf{v}, p)$$

1.2.2 La discrétisation éléments finis

On introduit une triangulation éléments finis Γ_h de l'espace Ω_h qui est une approximation discrète du domaine de calcul continu Ω .

$$\Omega_h = \bigcup_{\omega_h \in \Gamma_h} \omega_h$$

On définit des approximations \mathcal{V}_h^{ca} et \mathcal{P}_h des espaces \mathcal{V}^{ca} et \mathcal{P} sur la triangulation Γ_h . Ils sont engendrés par les fonctions de base qui discrétisent la vitesse et la pression. On cherche la solution discrète du problème (\mathbf{v}_h, p_h) dans ces espaces de dimension finie. Ces espaces ne peuvent pas être choisis de manière arbitraire, ils doivent vérifier la condition de compatibilité de Brezzi-Babuska (1.8)[Babuska73].

$$\exists \beta > 0, \beta \text{ indépendant de } h, \quad \inf_{p_h \in \mathcal{P}_h \setminus \{0\}} \sup_{\mathbf{v}_h \in \mathcal{V}_h^{ca} \setminus \{0\}} \left| \frac{\int_{\Omega} p_h \operatorname{div} \mathbf{v}_h d\omega}{\|p_h\|_0 \|\mathbf{v}_h\|_1} \right| \geq \beta \quad (1.8)$$

La vérification de (1.8) assure d'une part que la solution du problème existe et est unique et d'autre part, que la solution du problème discret tend vers la solution du problème continu quand le pas de discrétisation h tend vers 0.

$$\lim_{h \rightarrow 0} \mathbf{v}_h = \mathbf{v}; \quad \lim_{h \rightarrow 0} p_h = p$$

Si les espaces d'approximation ne vérifient pas cette condition, la solution du problème n'est pas unique et on peut converger vers des modes en pression dits "parasites". La compatibilité de ces espaces dépend de la nature des éléments finis.

1.2.2.1 Choix des éléments

Un des éléments finis mixtes possible est le P1+/P1 qui est choisi en 2D par [Abdalass87] par exemple et a été introduit dans [Arnold et al.84]. En 3D il s'agit d'un tétraèdre sur lequel le champ de vitesse est décomposé en une partie linéaire sur le tétraèdre et une partie bulle linéaire définie sur les 4 sous tétraèdres formés par les sommets et le centre de gravité du tétraèdre (cf. figure 1.2). La pression est linéaire sur l'élément.

$$\begin{aligned} \mathcal{P}_h &= \left\{ p_h \in C^0(\Omega_h) / \forall \omega_h \in \Gamma_h, p_h / \omega_h \in P^1(\omega_h) \right\} \\ \mathcal{V}_h &= \mathcal{L}_h \oplus \mathcal{B}_h \\ \mathcal{L}_h &= \left\{ \mathbf{v}_h \in (C^0(\Omega_h))^3 / \forall \omega_h \in \Gamma_h, \mathbf{v}_h / \omega_h \in (P^1(\omega_h))^3 \right\} \\ \mathcal{B}_h &= \left\{ \mathbf{b}_h \in (C^0(\Omega_h))^3 / \forall \omega_h \in \Gamma_h, \mathbf{b}_h = 0 \text{ sur } \partial\Omega_h \text{ et } \mathbf{b}_h / \omega_h^k \in (P^1(\omega_h^k))^3 \quad k = 1, 2, 3, 4 \right\} \end{aligned}$$

où $(\omega_h^k)_{k=1,2,3,4}$ sont les 4 sous tétraèdres dont le sommet commun est le centre de gravité du tétraèdre.

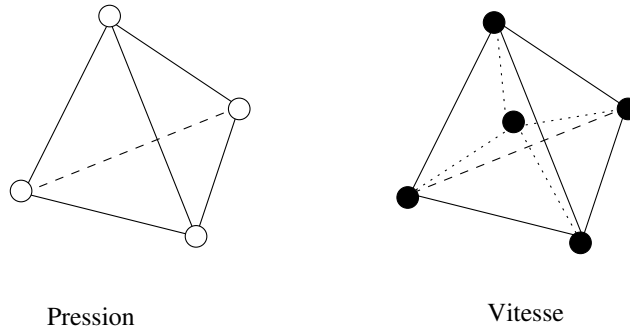


FIG. 1.2 – Elément P1+/P1

Le choix s'est porté sur des éléments tétraédriques pour différentes raisons. Ils permettent de décrire des volumes de géométrie difficile et présentent des facilités pour le maillage et le remaillage [Coupez91]. D'autre part, le rapport précision/temps de calcul est avantageux en comparaison avec d'autres éléments [Coupez95].

Les champs de vitesse et de pression discrets s'écrivent donc :

$$\begin{aligned} \mathbf{v}_h &= \mathbf{v}_h^l + \mathbf{v}_h^b = \sum_{k=1}^{nbnoe} V_k^l N_k^l + \sum_{j=1}^{nbel} V_j^b N_j^b \\ p_h &= \sum_{k=1}^{nbnoe} P_k N_k^l \end{aligned} \quad (1.9)$$

où $(N_k^l)_k$ et $(N_j^b)_j$ sont les fonctions de bases linéaires et bulle.

1.2.2.2 Le problème éléments finis

Le problème discret s'écrit

$$\left| \begin{array}{l} \text{Trouver } (\mathbf{v}_h, p_h) \in (\mathcal{V}_h^{ca}, \mathcal{P}_h) \text{ tel que } \forall (\mathbf{v}_h^*, p_h^*) \in (\mathcal{V}_{0h}^{ca}, \mathcal{P}_h) \\ \left\{ \begin{array}{l} \int_{\Omega} \mathbf{s}_h : \dot{\boldsymbol{\varepsilon}}_h^* d\omega - \int_{\Omega} p_h \operatorname{div} \mathbf{v}_h^* d\omega - \int_{\partial\Omega_c} \boldsymbol{\tau}_h \cdot \mathbf{v}_h^* dS = 0 \\ \int_{\Omega} p_h^* \operatorname{div} \mathbf{v}_h d\omega = 0 \end{array} \right. \end{array} \right. \quad (1.10)$$

où $\mathcal{V}_h^{ca} = \mathcal{L}_h^{ca} \oplus \mathcal{B}_h$

avec $\mathcal{L}_h^{ca} = \left\{ \mathbf{v}_h \in (\mathcal{C}^0(\Omega_h))^3 ; (\mathbf{v}_h - \mathbf{v}^{outil}) \cdot \mathbf{n} = 0 \text{ } \partial\Omega_c / \forall \omega_h \in \Gamma_h, \mathbf{v}_{h/\omega_h} \in (P^1(\omega_h))^3 \right\}$

et $\mathcal{V}_{0h}^{ca} = \mathcal{L}_{h0}^{ca} \oplus \mathcal{B}_h$

avec $\mathcal{L}_{h0}^{ca} = \left\{ \mathbf{v}_h \in (\mathcal{C}^0(\Omega_h))^3 ; \mathbf{v}_h \cdot \mathbf{n} = 0 \text{ } \partial\Omega_c / \forall \omega_h \in \Gamma_h, \mathbf{v}_{h/\omega_h} \in (P^1(\omega_h))^3 \right\}$.

Si l'on considère la décomposition de \mathbf{v}_h en $\mathbf{v}_h^l + \mathbf{v}_h^b$ et $\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^l + \dot{\boldsymbol{\varepsilon}}^b$, on écrit le système ci-dessus de la manière suivante.

$$\left| \begin{array}{l} \text{Trouver } (\mathbf{v}_h, p_h) \in (\mathcal{V}_h^{ca}, \mathcal{P}_h) \text{ tel que } \forall (\mathbf{v}_h^*, p_h^*) \in (\mathcal{V}_{0h}^{ca}, \mathcal{P}_h) \\ \left\{ \begin{array}{l} \int_{\Omega} \mathbf{s}_h(\mathbf{v}_h^l + \mathbf{v}_h^b) : \dot{\boldsymbol{\varepsilon}}_h^{*l} d\omega - \int_{\Omega} p_h \operatorname{div} \mathbf{v}_h^{*l} d\omega - \int_{\partial\Omega_c} \boldsymbol{\tau}_h(\mathbf{v}_h^l) \cdot \mathbf{v}_h^{*l} dS = 0 \\ \int_{\Omega} \mathbf{s}_h(\mathbf{v}_h^l + \mathbf{v}_h^b) : \dot{\boldsymbol{\varepsilon}}_h^{*b} d\omega - \int_{\Omega} p_h \operatorname{div} \mathbf{v}_h^{*b} d\omega = 0 \\ \int_{\Omega} p_h^* \operatorname{div}(\mathbf{v}_h^l + \mathbf{v}_h^b) d\omega = 0 \end{array} \right. \end{array} \right.$$

Remarque :

On peut noter que le terme de frottement n'intervient pas dans la seconde équation car le champ de vitesse bulle est nul aux bords des tétraèdres du maillage et donc nul au bord du domaine.

Ce qui peut s'écrire sous la forme matricielle :

$$\begin{pmatrix} \mathbf{A}^{ll} & \mathbf{A}^{lb} & \mathbf{A}^{lp} \\ \mathbf{A}^{bl} & \mathbf{A}^{bb} & \mathbf{A}^{bp} \\ \mathbf{A}^{pl} & \mathbf{A}^{pb} & 0 \end{pmatrix} \begin{pmatrix} V^l \\ V^b \\ P \end{pmatrix} = \begin{pmatrix} C \\ 0 \\ 0 \end{pmatrix}$$

où V^l est le champ de vitesse aux nœuds, V^b est la correction bulle du champ de vitesse aux centres de gravité des tétraèdres et P est le champ de pression aux nœuds.

1.2.2.3 L'élimination du terme bulle

Le déviateur des contraintes \mathbf{s} peut s'écrire indépendamment en fonction de la partie linéaire et de la partie bulle du champ de vitesse. Le champ bulle possède la propriété que son gradient est orthogonal à tout tenseur constant sur un élément e du maillage, ce qui donne :

$$\int_{\Omega_e} C : \dot{\boldsymbol{\varepsilon}}_h^b d\omega = 0$$

On peut donc en déduire

$$\int_{\Omega_e} \dot{\boldsymbol{\varepsilon}}_h^l : \dot{\boldsymbol{\varepsilon}}_h^b d\omega = 0 \quad (1.11)$$

où Ω_e est un élément de la triangulation. On peut alors simplifier le système précédent. On trouve, sur une matrice locale associée à l'élément e du maillage :

$$\begin{pmatrix} \mathbf{A}_e^{ll} & 0 & \mathbf{A}_e^{lp} \\ 0 & \mathbf{A}_e^{bb} & \mathbf{A}_e^{bp} \\ \mathbf{A}_e^{pl} & \mathbf{A}_e^{pb} & 0 \end{pmatrix} \begin{pmatrix} V_e^l \\ V_e^b \\ P_e \end{pmatrix} = \begin{pmatrix} C_e \\ 0 \\ 0 \end{pmatrix} \quad (1.12)$$

Si on extrait la seconde ligne du système, le champ de vitesse bulle peut s'exprimer en fonction du champ de pression :

$$\begin{aligned} \mathbf{A}_e^{bb} V_e^b + \mathbf{A}_e^{bp} P_e &= 0 \\ \Leftrightarrow V_e^b &= -\mathbf{A}_e^{bb^{-1}} \mathbf{A}_e^{bp} P_e \end{aligned}$$

On peut donc réinjecter l'expression de V_b dans le système. On aboutit à un système à 4 inconnues par nœud, 3 composantes du champ de vitesse et une composante du champ de pression. Si on exprime le système sur l'ensemble des degrés de liberté, on obtient l'équation (1.13).

$$\begin{pmatrix} \mathbf{A}^{ll} & \mathbf{A}^{lp} \\ \mathbf{A}^{pl} & -\mathbf{A}^{pb} \mathbf{A}^{bb^{-1}} \mathbf{A}^{bp} \end{pmatrix} \begin{pmatrix} V^l \\ P \end{pmatrix} = \begin{pmatrix} C \\ 0 \end{pmatrix} \quad (1.13)$$

Cette élimination du terme bulle permet de plus d'obtenir un système mieux conditionné que le précédent, puisque la matrice ne présente plus de terme nul sur la diagonale. Pour des raisons de simplification d'écriture, on peut aussi l'écrire :

$$\begin{pmatrix} \mathbf{A}^{vv} & \mathbf{A}^{vp} \\ \mathbf{A}^{pv} & -\mathbf{A}^{pp} \end{pmatrix} \begin{pmatrix} V \\ P \end{pmatrix} = \begin{pmatrix} C \\ 0 \end{pmatrix} \quad (1.14)$$

où V et P sont les valeurs du champ de vitesse et du champ de pression aux nœuds du maillage.

Remarque :

La matrice A^{pp} issue de l'élimination du terme bulle est symétrique définie positive. La matrice du système (1.14) n'est donc pas définie positive.

1.2.3 Le problème non linéaire

Jusqu'ici, on a travaillé en supposant que la rhéologie et le frottement étaient newtoniens ($m = p_f = 1$). On considère maintenant le cas où le matériau est viscoplastique. La dépendance du déviateur \mathbf{s} par rapport au tenseur des vitesses de déformation $\dot{\boldsymbol{\varepsilon}}$ et la relation entre la cission de frottement et la vitesse de glissement sont alors non linéaires. Pour écrire les équations discrètes, on repart des équations du système d'équations (1.6) et on introduit un troisième champ $\mathbf{D} = \dot{\boldsymbol{\varepsilon}}$. Si on note $\eta(\mathbf{D}) = K(\sqrt{3}\dot{\boldsymbol{\varepsilon}})^{m-1}$, on écrit le système :

$$\begin{cases} \operatorname{div} 2\eta(\mathbf{D})\dot{\boldsymbol{\varepsilon}} - \mathbf{Grad} p = 0 \\ \operatorname{div} \mathbf{v} = 0 \\ \mathbf{D} - \dot{\boldsymbol{\varepsilon}} = 0 \end{cases}$$

On discrétise le champ \mathbf{D} dans l'espace $P^0(\Omega_e)^{3 \times 3}$ des fonctions constantes par éléments. Le problème mixte discret s'écrit alors :

$$\left| \begin{array}{l} \text{Trouver } (\mathbf{v}_h, p_h, \mathbf{D}_h) \in (\mathcal{V}_h^{ca}, \mathcal{P}_h, P^0(\Omega_e)^{3 \times 3}) \text{ tel que } \forall (\mathbf{v}_h^*, p_h^*, \mathbf{D}_h^*) \in (\mathcal{V}_0^{ca}, \mathcal{P}_h, P^0(\Omega_e)^{3 \times 3}) \\ \left\{ \begin{array}{l} \int_{\Omega} 2\eta_h(\mathbf{D}_h) \dot{\boldsymbol{\varepsilon}}_h : \dot{\boldsymbol{\varepsilon}}_h^* d\omega - \int_{\Omega} p_h \operatorname{div} \mathbf{v}_h^* d\omega - \int_{\partial\Omega_c} \boldsymbol{\tau}_h \cdot \mathbf{v}_h^* dS = 0 \\ \int_{\Omega} p_h^* \operatorname{div} \mathbf{v}_h d\omega = 0 \\ \int_{\Omega} (\mathbf{D}_h - \dot{\boldsymbol{\varepsilon}}_h) : \mathbf{D}_h^* d\omega = 0 \end{array} \right. \end{array} \right.$$

En utilisant les propriétés du champ bulle, on montre que :

$$\mathbf{D}_h = \dot{\boldsymbol{\varepsilon}}(\mathbf{v}_h^l) \text{ au sens faible}$$

Ainsi on peut écrire

$$\eta_h(\mathbf{D}_h) = \eta_h(\dot{\boldsymbol{\varepsilon}}_h(\mathbf{v}_h^l)) \text{ au sens faible}$$

(voir [Coupez95] pour plus de détails). On peut donc écrire (toujours au sens faible) :

$$\mathbf{s}_h(\mathbf{v}_h^l + \mathbf{v}_h^b) = 2\eta_h(\mathbf{v}_h^l + \mathbf{v}_h^b) \dot{\boldsymbol{\varepsilon}}_h(\mathbf{v}_h^l + \mathbf{v}_h^b) = 2\eta_h(\mathbf{v}_h^l) (\dot{\boldsymbol{\varepsilon}}_h(\mathbf{v}_h^l) + \dot{\boldsymbol{\varepsilon}}_h(\mathbf{v}_h^b))$$

On élimine le champ \mathbf{D}_h en le remplaçant par sa valeur, ce qui conduit aux équations suivantes :

$$\left\{ \begin{array}{l} \int_{\Omega} 2\eta_h(\dot{\boldsymbol{\varepsilon}}_h^l) \dot{\boldsymbol{\varepsilon}}_h : \dot{\boldsymbol{\varepsilon}}_h^* d\omega - \int_{\Omega} p_h \operatorname{div} \mathbf{v}_h^* d\omega - \int_{\partial\Omega_c} \boldsymbol{\tau}_h \cdot \mathbf{v}_h^* dS = 0 \\ \int_{\Omega} p_h^* \operatorname{div} \mathbf{v}_h d\omega = 0 \end{array} \right.$$

Par la suite, on note $\eta_h = \eta_h(\dot{\boldsymbol{\varepsilon}}_h^l)$. Si on utilise la propriété (1.11), alors on peut écrire le système de la façon suivante

$$\left\{ \begin{array}{l} \mathbf{R}^l(\mathbf{v}_h^l, \mathbf{v}_h^b, p_h) = \mathbf{R}^{ll} \quad +\mathbf{R}^{lp} \quad +R^l = 0 \\ \mathbf{R}^b(\mathbf{v}_h^l, \mathbf{v}_h^b, p_h) = \quad \mathbf{R}^{bb} \quad +\mathbf{R}^{bp} = 0 \\ \mathbf{R}^p(\mathbf{v}_h^l, \mathbf{v}_h^b, p_h) = \mathbf{R}^{pl} + \mathbf{R}^{pb} = 0 \end{array} \right.$$

où

$$\begin{aligned} \mathbf{R}^{ll} &= \int_{\Omega} 2\eta_h(\mathbf{v}_h^l) \dot{\boldsymbol{\varepsilon}}_h(\mathbf{v}_h^l) : \dot{\boldsymbol{\varepsilon}}_h^{*l} d\omega, & \mathbf{R}^{lp} &= - \int_{\Omega} p_h \operatorname{div} \mathbf{v}_h^{*l} d\omega, \\ \mathbf{R}^{bb} &= \int_{\Omega} 2K\eta_h(\mathbf{v}_h^b) \dot{\boldsymbol{\varepsilon}}_h(\mathbf{v}_h^b) : \dot{\boldsymbol{\varepsilon}}_h^{*b} d\omega, & \mathbf{R}^{bp} &= - \int_{\Omega} p_h \operatorname{div} \mathbf{v}_h^{*b} d\omega, \\ \mathbf{R}^{pl} &= \int_{\Omega} p_h^* \operatorname{div} \mathbf{v}_h^l d\omega, & \mathbf{R}^{pb} &= \int_{\Omega} p_h^* \operatorname{div} \mathbf{v}_h^b d\omega, \\ R^l &= - \int_{\partial\Omega_c} \boldsymbol{\tau}_h(\mathbf{v}_h^l) \cdot \mathbf{v}_h^{*l} dS \end{aligned}$$

On résout le problème non linéaire par une méthode de Newton-Raphson. Pour cela on introduit les matrices de raideur ou hessiens locaux \mathbf{K}^{xy} définis par :

$$\mathbf{K}^{xy} = \frac{\partial \mathbf{R}^x}{\partial y}$$

où x et y évoluent dans $\{l, b, p\}$. A chaque itération $n + 1$ de Newton-Raphson, on aboutit à la résolution du système linéaire suivant :

$$\begin{pmatrix} \mathbf{K}^{ll} & 0 & \mathbf{K}^{lp} \\ 0 & \mathbf{K}^{bb} & \mathbf{K}^{bp} \\ \mathbf{K}^{pl} & \mathbf{K}^{pb} & 0 \end{pmatrix} \begin{pmatrix} \delta V^l \\ \delta V^b \\ \delta P \end{pmatrix} = - \begin{pmatrix} \mathbf{R}^l(V_n^l, V_n^b, p_n) \\ \mathbf{R}^b(V_n^l, V_n^b, p_n) \\ \mathbf{R}^p(V_n^l, V_n^b, p_n) \end{pmatrix} \quad (1.15)$$

où $(\delta V^l, \delta V^b, \delta P)$ sont les corrections apportées au champ à l'itération n : (V_n^l, V_n^b, P_n) .

$$\begin{pmatrix} V_{n+1}^l \\ V_{n+1}^b \\ P_{n+1} \end{pmatrix} = \begin{pmatrix} V_n^l \\ V_n^b \\ P_n \end{pmatrix} + \begin{pmatrix} \delta V^l \\ \delta V^b \\ \delta P \end{pmatrix}$$

1.2.3.1 L'élimination du terme bulle

Par le même type de raisonnement qu'en linéaire, on aboutit à la résolution d'un système d'où on a exclu le champ de vitesse bulle et qui est de la forme :

$$\begin{pmatrix} \mathbf{K}^{vv} & \mathbf{K}^{vp} \\ \mathbf{K}^{pv} & -\mathbf{K}^{pp} \end{pmatrix} \begin{pmatrix} \delta V \\ \delta P \end{pmatrix} = - \begin{pmatrix} \mathbf{R}^l \\ \mathbf{R}^p - \mathbf{K}^{pb} \mathbf{K}^{bb^{-1}} \mathbf{R}^b \end{pmatrix} \quad (1.16)$$

où $\delta V = \delta V^l$ est la correction du terme en vitesse linéaire. \mathbf{K}^{pp} est une matrice symétrique définie positive. Elle est définie comme pour le cadre linéaire au niveau de l'élément par :

$$\mathbf{K}^{pp} = \mathbf{K}^{pb} \mathbf{K}^{bb^{-1}} \mathbf{K}^{bp}$$

En linéaire comme en non linéaire, l'élimination du terme bulle permet de stabiliser la matrice de raideur (meilleur conditionnement) en éliminant les termes diagonaux nuls. L'avantage de cette méthode sur une méthode de stabilisation classique est que le terme de stabilisation \mathbf{K}^{pp} n'est pas fonction d'un paramètre comme dans [Vincent97]. On se ramène à une résolution avec un champ de vitesse linéaire et la condition de compatibilité des espaces (1.8) est vérifiée.

1.3 La résolution du système linéaire

1.3.1 Stockage de la matrice

Si on utilise une méthode de résolution itérative, il est préférable de lui associer une méthode de stockage compact. A l'aide de tableaux de pointeurs lignes et colonnes, on ne stocke que les termes non nuls de la matrice. Comme la matrice résultant du calcul est une matrice creuse, ce type de stockage présente une économie très importante en place disque (voir tableau 1.1).

	Stockage
Méth. directe	$16Nbn oe^{\frac{5}{3}}$
Stock. morse	$192Nbn oe$

TAB. 1.1 – Comparaison des coûts de stockage méthode directe et stockage morse

De plus avec une telle méthode de stockage, le calcul des produits matrice/vecteur, qui sont largement utilisés dans les méthodes itératives, est très efficace. Ce qui représente un gain important en temps de calcul.

1.3.2 La méthode de résolution

Nous discuterons des méthodes de résolution du système ci-dessus dans le paragraphe 3.1. Notre problème peut s'assimiler à la résolution d'un système de Stokes stabilisé, le lecteur intéressé pourra donc se référer aux publications concernant ces problèmes [Wathen et al.93] [Silvester et al.94],[Vincent97],... pour des détails sur différentes méthodes. On note simplement dans le tableau 1.2 les taux de convergence asymptotique pour une méthode directe et une méthode de résidu minimal préconditionné sur un système dont la matrice est creuse.

	Méthode directe	Résidu minimal
Coût de calcul	$O(Nbn oe^{\frac{1}{3}})$	$O(Nbn oe^{\frac{2}{2}})$

TAB. 1.2 – Comparaison des coûts de calcul asymptotiques d'une méthode directe et d'une méthode de résidu minimal préconditionné

1.3.3 La discrétisation temporelle

Pour prendre en compte l'évolution en temps, il faut intégrer à notre problème l'équation d'évolution :

$$\mathbf{v} = \frac{d\mathbf{x}}{dt} \quad (1.17)$$

pendant le temps de forgeage $[t_{deb}, t_{fin}]$. Pour cela, on discrétise l'intervalle $[t_{deb}, t_{fin}]$ en plusieurs pas de temps et on intègre l'équation (1.17) à l'aide d'une méthode différences finies.

$$[t_{deb}, t_{fin}] = \cup_{i=1,N} [t_i, t_i + \Delta t_i]$$

Le schéma choisi dans **Forge3** est un schéma d'Euler explicite :

$$X^{t+\Delta t} = X^t + \Delta t V^t$$

Utiliser ce schéma revient à considérer que la vitesse est constante au cours de l'incrément $[t, t + \Delta t]$. Pour une configuration Ω_t donnée, on calcule le champ de vitesse V_t et on en déduit la configuration au pas de temps suivant $t + \Delta t$. Cette méthode implique que les équations d'équilibre ne sont vérifiées qu'à certains moments de la simulation (méthode de collocation par points).

D'autres schémas plus précis ont été testés. Par exemple, pour des problèmes bien particuliers de laminage circulaire ou transversal où l'outil présente un déplacement en rotation plutôt qu'en translation, le schéma d'Euler explicite engendre des variations de volume considérables. Un schéma de type Runge-Kutta a été mis en place [Boyre99] et s'est montré bien plus adapté à ce type de configuration.

L'évolution temporelle du problème oblige aussi à prendre en compte d'autres problèmes tels que l'évolution du contact que l'on détaillera plus loin (voir § 2) ou la déformation du maillage. Une méthode de remaillage automatique a été développée dans [Coupez91]. Le remaillageur procède par amélioration topologique des éléments tétraédriques qui constituent le maillage. Pour cela, on alterne une procédure surfacique et une procédure volumique qui utilisent des opérations de type inversion de diagonale, déplacement de nœuds ...

Chapitre 2

LA GESTION DU CONTACT

Introduction

Un problème omniprésent

De Coulomb ou Newton à nos jours, le contact et le frottement sont des problèmes étroitement liés qui suscitent toujours de nombreux travaux de recherche. En effet, le contact est une caractéristique intrinsèque de la mécanique par laquelle les corps interagissent entre eux. Une bonne modélisation numérique du contact est un facteur important dans bien des domaines et tout particulièrement en mise en forme des métaux. On peut également citer les problèmes issus de la mécanique des structures ou les problèmes d'impact où la surface de contact évolue très brusquement et très rapidement. D'autre part, les équations du contact font apparaître des problèmes théoriques complexes. Par exemple, la méthode des multiplicateurs de Lagrange, pourtant largement utilisée, soulève encore des problèmes quant à la compatibilité des espaces discrets [Hild98].

La nature du contact

Un autre aspect des problèmes de contact est le choix de la condition à modéliser. Le contact peut en effet être étudié entre un corps déformable et un corps rigide, ou entre deux corps déformables [Laursen et al.93], [Jaouen98], [Chertier97] selon le type de phénomènes que l'on veut prendre en compte. Pour notre part, on considérera la pièce comme déformable et le moule comme rigide, ce qui est une bonne approche du problème de forgeage. La nature du contact amène à considérer des équations différentes suivant la nature du couple outil/matière. Ainsi on utilisera souvent une condition de contact bilatéral collant pour la mise en forme des polymères. En forgeage, la surface en contact frotte avec l'outil. Elle évolue donc au cours de la mise en forme, peut décoller ou entrer en contact. Ceci justifie la condition de contact unilatéral frottant utilisée.

Les méthodes numériques

Il existe plusieurs méthodes pour imposer les conditions de contact. La méthode des multiplicateurs de Lagrange donne des résultats très précis mais présente l'inconvénient d'ajouter des inconnues au problème. La méthode directe, ou élimination des degrés de

liberté, consiste à imposer directement au système les conditions de contact sur les degrés de liberté concernés. La méthode choisie dans le logiciel **Forge3** et sur laquelle nous avons basé nos développements est une méthode de pénalisation nodale explicite. Pour le lecteur intéressé par la problématique du contact, plusieurs articles de synthèse donnent une bonne présentation de l'état de l'art pour ces problèmes par exemple [Wriggers95], [Chabrant et al.98].

Nos travaux

Le but de nos travaux est d'améliorer la précision de la méthode de **Forge3** lorsqu'on utilise des pas de temps relativement grands. Pour cela on a développé un algorithme implicite développé dans le paragraphe 2.2. On commence toutefois par présenter les équations du contact et quelques méthodes utilisées pour les résoudre avant d'introduire la méthode développée dans **Forge3** dans le paragraphe 2.1.

2.1 Les équations du contact

2.1.1 Le traitement du contact en quasi stationnaire

2.1.1.1 La condition de contact bilatéral

Pour des raisons de simplification de l'exposé, on commencera par considérer une condition de contact bilatéral. Cette condition impose que les points matériels en contact restent en contact. Elle s'écrit :

$$\left(\mathbf{v} - \mathbf{v}^{outil} \right) \cdot \mathbf{n} = 0 \quad \text{sur } \partial\Omega_c$$

où \mathbf{n} est la normale entrante à l'outil.

2.1.1.2 La formulation mixte du problème

On commence ici par réécrire la formulation faible en n'imposant pas directement les conditions aux limites du contact. Cela revient à dire que l'on se ramène au problème suivant :

$$\left\{ \begin{array}{l} \text{Trouver } (\mathbf{v}, p) \in (\mathcal{V}, \mathcal{P}) \text{ tel que } \forall (\mathbf{v}^*, p^*) \in (\mathcal{V}, \mathcal{P}) \\ \int_{\Omega} \mathbf{s} : \dot{\boldsymbol{\varepsilon}}^* d\omega - \int_{\Omega} p \operatorname{div} \mathbf{v}^* d\omega - \int_{\partial\Omega_c} \boldsymbol{\tau} \cdot \mathbf{v}^* dS \\ \quad - \int_{\partial\Omega_c} \sigma_n \mathbf{v}^* \cdot \mathbf{n} dS = 0 \\ \int_{\Omega} p^* \operatorname{div} \mathbf{v} d\omega = 0 \\ \text{+condition de contact} \end{array} \right. \quad (2.1)$$

Il convient maintenant d'ajouter un terme à ce problème qui traduise la condition de contact. On peut, pour cela, choisir différentes méthodes.

Les multiplicateurs

La méthode des multiplicateurs consiste à introduire une nouvelle variable λ [Pascoe et al.89], [Fish et al.95] qui est le multiplicateur et qui est associée à la condition de contact par l'équation suivante :

$$\int_{\partial\Omega_c} \lambda \left(\mathbf{v} - \mathbf{v}^{outil} \right) \cdot \mathbf{n} dS = 0$$

que l'on ajoute au lagrangien vitesse/pression ce qui donne, pour une rhéologie quelconque, la fonctionnelle suivante :

$$\begin{aligned} \mathcal{L}_2(v, p, \lambda) &= \int_{\Omega} \frac{K}{m+1} (\sqrt{3}\dot{\boldsymbol{\varepsilon}})^{m+1} d\omega - \int_{\Omega} p \operatorname{div} \mathbf{v} d\omega \\ &+ \int_{\partial\Omega_c} \frac{K\alpha_f}{p_f+1} \|\Delta \mathbf{v}_g\|^{p_f+1} dS + \int_{\partial\Omega_c} \lambda \left(\mathbf{v} - \mathbf{v}^{outil} \right) \cdot \mathbf{n} dS \end{aligned}$$

On a ainsi obtenu un lagrangien en vitesse/pression/multiplicateur \mathcal{L}_2 . La résolution du problème sous la condition de contact revient alors à chercher le point selle de ce lagrangien,

c'est-à-dire à résoudre le système suivant :

$$\left| \begin{array}{l} \text{Trouver } (\mathbf{v}, p, \lambda) \in (\mathcal{V}, \mathcal{P}, H^{-\frac{1}{2}}(\partial\Omega_c)) \text{ tel que } \forall (\mathbf{v}^*, p^*, \lambda^*) \in (\mathcal{V}, \mathcal{P}, H^{-\frac{1}{2}}(\partial\Omega_c)) \\ \left\{ \begin{array}{l} \left(\frac{\partial \mathcal{L}(\mathbf{v}, p, \lambda)}{\partial \mathbf{v}}, \mathbf{v}^* \right) = \left(\frac{\partial \mathcal{L}(\mathbf{v}, p)}{\partial \mathbf{v}}, \mathbf{v}^* \right) + \int_{\partial\Omega_c} \lambda \mathbf{v}^* \cdot \mathbf{n} dS = 0 \\ \left(\frac{\partial \mathcal{L}(\mathbf{v}, p, \lambda)}{\partial p}, p^* \right) = \int_{\Omega} p^* \operatorname{div} \mathbf{v} d\omega = 0 \\ \left(\frac{\partial \mathcal{L}(\mathbf{v}, p, \lambda)}{\partial \lambda}, \lambda^* \right) = \int_{\partial\Omega_c} \lambda^* (\mathbf{v} - \mathbf{v}^{outil}) \cdot \mathbf{n} dS = 0 \end{array} \right. \end{array} \right. \quad (2.2)$$

où \mathcal{L} est le lagrangien vitesse pression défini pour une rhéologie linéaire par l'équation (1.7) et $H^{\frac{1}{2}}(\partial\Omega_c)$ est l'ensemble des traces des fonctions de $H^1(\Omega)$ et $H^{-\frac{1}{2}}(\partial\Omega_c)$ son dual.

On peut vérifier que λ est la pression de contact σ_n en identifiant (2.2) et (2.1).

Cette méthode oblige à ajouter des inconnues (λ) mais elle permet de résoudre exactement le problème et de calculer précisément la contrainte normale sur la surface de contact.

La formulation en lagrangien est souvent utilisée sous la forme du lagrangien augmenté décrite dans [Fortin et al.92] et qui combine la méthode de multiplicateurs et celle de pénalisation [Zavarise et al.95] ou encore sous la forme du Lagrangien perturbé [Simo et al.92]. Ces méthodes permettent de conserver la précision des multiplicateurs sans ajouter de variables supplémentaires lors de la résolution du système. Les multiplicateurs sont en effet, calculés par une méthode découplée itérative.

La pénalisation

La méthode de pénalisation n'augmente pas le nombre d'inconnues du problème [Cescotto et al.93], [Pavanachand et al.96]. La contrainte normale est exprimée en fonction de la vitesse via un opérateur O dont l'expression n'est pas toujours simple. On trouvera son expression discrète dans le paragraphe suivant. On introduit alors un facteur de pénalisation ρ . Le problème de contact est résolu de façon approchée (avec une précision de l'ordre de $\frac{1}{\rho}$). Pour une résolution exacte, il faudrait donc atteindre $\rho = \infty$ ce qui n'est pas possible. Pour des valeurs trop grandes de ce coefficient, on déconditionne la matrice du problème et l'utilisation de méthodes itératives de résolution devient impossible. Il s'agit donc trouver un compromis entre la précision des calculs et le conditionnement de la matrice ou alors de recourir aux méthodes de Lagrangien augmenté ou perturbé que nous venons de mentionner.

2.1.1.3 La discrétisation des équations

Les équations discrètes

Dans la première partie, on a présenté la discrétisation éléments finis de la vitesse et de la pression. Il faut maintenant introduire une nouvelle famille de fonctions de base $(M_m)_m$ pour discrétiser les multiplicateurs de Lagrange λ_h . On écrit donc :

$$\lambda_h = \sum_{m=1}^{Nc} \Lambda_m M_m$$

où Nc est le nombre de paramètres d'interpolation de la condition de contact.

Reprenons l'expression du résidu non linéaire en lui ajoutant la contribution du contact.

On aboutit aux équations suivantes :

$$\begin{cases} \mathbf{R}^l(\mathbf{v}_h, p_h, \lambda_h) = \mathbf{R}^{ll} & +\mathbf{R}^{lb} & +\mathbf{R}^{lp} & +\mathbf{R}^{l\lambda} & +R^l & = 0 \\ \mathbf{R}^b(\mathbf{v}_h, p_h, \lambda_h) = \mathbf{R}^{bl} & +\mathbf{R}^{bb} & +\mathbf{R}^{bp} & & & = 0 \\ \mathbf{R}^p(\mathbf{v}_h, p_h, \lambda_h) = \mathbf{R}^{pl} & +\mathbf{R}^{pb} & & & & = 0 \\ \mathbf{R}^\lambda(\mathbf{v}_h, p_h, \lambda_h) = \mathbf{R}^{\lambda l} & & & & +R^\lambda & = 0 \end{cases}$$

où \mathbf{R}^{ll} , \mathbf{R}^{lb} , \mathbf{R}^{lp} , \mathbf{R}^{bl} , \mathbf{R}^{bb} , \mathbf{R}^{bp} , \mathbf{R}^{pl} , \mathbf{R}^{pb} et R^l restent inchangés et

$$\begin{aligned} \mathbf{R}^{v\lambda} &= \int_{\Omega} \lambda_h \mathbf{v}_h^* \cdot \mathbf{n} \, d\omega, \\ \mathbf{R}^{\lambda v} &= \int_{\partial\Omega_c} \lambda_h^* \mathbf{v}_h \cdot \mathbf{n} \\ R^\lambda &= - \int_{\partial\Omega_c} \lambda_h^* \mathbf{v}^{outil} \cdot \mathbf{n} \end{aligned}$$

On note qu'il n'y a pas de terme $= \mathbf{R}^{b\lambda}$. En effet, la vitesse bulle est nulle aux bords des éléments et donc aux bords du maillage et en particulier sur la surface en contact avec l'outil où sont définis les multiplicateurs λ .

Comme dans le paragraphe 1.2.3, on introduit une procédure de Newton-Raphson pour traiter les non-linéarités et on procède à l'élimination du terme bulle :

$$\begin{pmatrix} \mathbf{K}^{vv} & \mathbf{K}^{vp} & \mathbf{K}^{v\lambda} \\ \mathbf{K}^{pv} & -\mathbf{K}^{pp} & 0 \\ \mathbf{K}^{\lambda v} & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta V \\ \delta P \\ \delta \Lambda \end{pmatrix} = - \begin{pmatrix} \mathbf{R}^v \\ 0 \\ \mathbf{R}^\lambda \end{pmatrix} \quad (2.3)$$

où

$$\begin{aligned} \mathbf{K}^{v\lambda}_{ikm} &= \int_{\partial\Omega_c} M_m N_k n_i \, dS \\ \mathbf{R}_m^\lambda &= \int_{\partial\Omega_c} M_m (\mathbf{v}_h - \mathbf{v}^{outil}) \cdot \mathbf{n} \, dS \end{aligned}$$

On peut noter que $\mathbf{K}^{v\lambda} = \mathbf{K}^{\lambda v^t}$.

La formulation discrète pénalisée

En partant du système (2.3), la formulation pénalisée se présente sous la forme :

$$\begin{pmatrix} \mathbf{K}^{vv}(\mathbf{v}) + \rho \mathbf{K}^{v\lambda} \mathbf{K}^{\lambda v} & \mathbf{K}^{vp} \\ \mathbf{K}^{pv} & -\mathbf{K}^{pp} \end{pmatrix} \begin{pmatrix} \delta V \\ \delta P \end{pmatrix} = - \begin{pmatrix} \mathbf{R}^v + \rho \mathbf{K}^{v\lambda} \mathbf{R}^\lambda \\ 0 \end{pmatrix} \quad (2.4)$$

On introduit donc dans la matrice de raideur des termes reliant la i^{eme} composante du nœud k et la j^{eme} composante du nœud l de la forme suivante :

$$(\mathbf{K}^{v\lambda} \mathbf{K}^{\lambda v})_{jl ik} = \sum_m \int_{\partial\Omega_c} M_m N_k n_i \, dS \int_{\partial\Omega_c} M_m N_l n_j \, dS \quad (2.5)$$

et dans le second membre des termes

$$(\mathbf{K}^{v\lambda} \mathbf{R}^\lambda)_{ik} = \sum_m \int_{\partial\Omega_c} M_m N_k n_i \, dS \int_{\partial\Omega_c} M_m (\mathbf{v}_h - \mathbf{v}^{outil}) \cdot \mathbf{n} \, dS \quad (2.6)$$

Si on écrit cette condition dans un lagrangien discret augmenté $\mathcal{L}_{h,\rho}$ [Fortin et al.92], on obtient en fonction du lagrangien discret \mathcal{L}_h , issu de la discrétisation de (1.7), la formule suivante :

$$\mathcal{L}_{h,\rho}(\mathbf{v}_h, p_h) = \mathcal{L}_h(\mathbf{v}_h, p_h) + \frac{\rho}{2} \sum_m \left[\int_{\partial\Omega_c} M_m \left((\mathbf{v}_h - \mathbf{v}^{outil}) \cdot \mathbf{n} \right) dS \right]^2 \quad (2.7)$$

Suivant la famille de fonctions de base $(M_m)_m$, on obtient différentes formulations du contact. En toute rigueur, ces fonctions doivent vérifier une condition de compatibilité comparable à celle pour la vitesse et la pression. En dehors de [Hild98], peu de travaux ont été consacrés à ce sujet. Pour ce qui nous concerne, on s'appuie sur ces résultats théoriques pour garantir la précision des interpolations étudiées. Dans la suite, on présente trois types de discrétisation. La première est celle que l'on retrouve dans le logiciel **Forge3**.

Le contact nodal

$$M_m = \delta(X_m)$$

La condition de contact est écrite aux nœuds m de la surface. $\delta(X_m)$ désigne la fonction de Dirac au nœud m . On introduit donc, dans la matrice de raideur, les termes pour le couplage des composantes i et j des nœuds k et l qui sont de la forme :

$$(\mathbf{K}^{v\lambda} \mathbf{K}^{\lambda v})_{ikjl} = \delta_{kl} n_i^k n_j^l$$

où n_i^k est la i^{eme} composante de la normale en X^k . Les termes introduits dans le second membre du problème sont de la forme :

$$(\mathbf{K}^{v\lambda} \mathbf{R}^\lambda)_{ik} = \left(\mathbf{v}_h^k - \mathbf{v}^{outil} \cdot \mathbf{n}^k \right) n_i^k$$

Notons que les fonctions de base M_m ne définissent pas une famille d'éléments finis de la surface.

Elles permettent de traiter chaque nœud indépendamment de ses voisins. Le contact ne couple que les différentes composantes du champ de vitesse d'un même nœud. La contribution des termes de contact à la matrice n'apparaît que près de la diagonale (dans le bloc diagonal nodal). Le contact touche donc des coefficients de la matrice qui sont contrôlés par le préconditionnement (voir paragraphe 3.1.2). Ceci permet de minimiser l'effet du coefficient de pénalisation sur la convergence de la résolution itérative [Marie97]. La position des termes de contact dans la matrice implique aussi que l'on n'augmente pas la largeur de bande du système à résoudre. En identifiant les premières lignes des systèmes (2.3) et (2.4), on trouve que la valeur nodale du multiplicateur s'exprime par :

$$\Lambda_k = \rho (V^k - V^{outil}) \cdot n_k$$

Le contact intégré

$$M_m = N_m$$

On choisit de discrétiser les termes de contact à l'aide des mêmes fonctions de base que celles du champ du vitesse. Le terme de pénalisation devient alors :

$$\mathcal{L}_{h,\rho}(\mathbf{v}, p) = \mathcal{L}_h(\mathbf{v}_h, p_h) + \frac{\rho}{2} \sum_m \left[\int_{\partial\Omega_c} N_m \left((\mathbf{v}_h - \mathbf{v}^{outil}) \cdot \mathbf{n} \right) dS \right]^2 \quad (2.8)$$

Les formules des termes introduits dans la matrice de raideur et le second membre sont les expressions (2.5) et (2.6) où l'on remplace M_m par N_m .

D'après la formule (2.5), on voit que l'on introduit des termes de couplages entre différents nœuds de la surface qui ne sont pas nécessairement des voisins directs (cf. figure 2.1). En utilisant le contact intégré, on double donc la largeur de bande du système en reliant

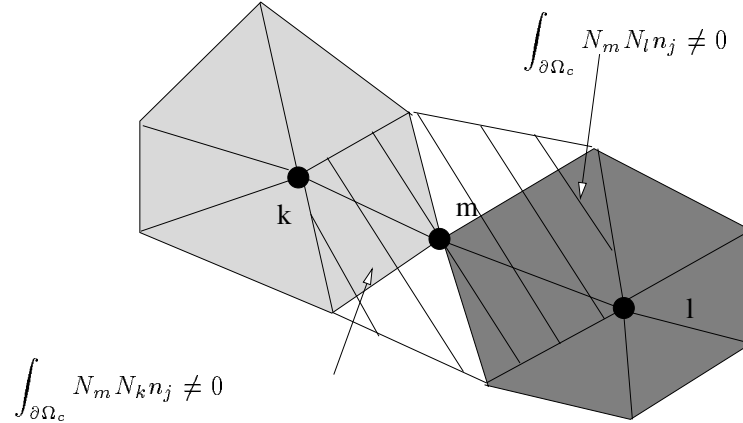


FIG. 2.1 – Relation induite par le contact intégré entre deux nœuds de la surface non voisins k et l

algébriquement des nœuds non voisins. La matrice n'est plus à diagonale dominante car on ajoute des termes de contact en dehors de la diagonale.

Le contact "surcontraint"

Une formulation souvent utilisée consiste à écrire que le système pénalisé continu suivant :

$$\left| \begin{array}{l} \text{Trouver } (\mathbf{v}, p) \in (\mathcal{V}, \mathcal{P}) \text{ tel que } \forall (\mathbf{v}^*, p^*) \in (\mathcal{V}, \mathcal{P}) \\ \left\{ \begin{array}{l} \int_{\Omega} \mathbf{s} : \dot{\boldsymbol{\varepsilon}}^* d\omega - \int_{\Omega} p \operatorname{div} \mathbf{v}^* d\omega - \int_{\partial\Omega_c} \boldsymbol{\tau} \cdot \mathbf{v}^* dS + \rho \int_{\partial\Omega_c} (\mathbf{v} - \mathbf{v}^{outil}) \cdot \mathbf{n} \mathbf{v}^* \cdot \mathbf{n} dS = 0 \\ \int_{\Omega} p^* \operatorname{div} \mathbf{v} d\omega = 0 \end{array} \right. \end{array} \right.$$

Ce qui correspond à un Lagrangien qui a pour expression:

$$\begin{aligned} \mathcal{L}_{h,\rho}(\mathbf{v}_h, p_h) &= \mathcal{L}_h(\mathbf{v}_h, p_h) + \frac{\rho}{2} \int_{\partial\Omega_c} \left((\mathbf{v}_h - \mathbf{v}^{outil}) \cdot \mathbf{n} \right)^2 dS \\ &= \mathcal{L}_h(\mathbf{v}_h, p_h) + \frac{\rho}{2} \int_{\partial\Omega_c} \left(\left(\sum_k V^k N_k - \mathbf{v}^{outil} \right) \cdot \mathbf{n} \right)^2 dS \\ &= \mathcal{L}_h(\mathbf{v}_h, p_h) + \frac{\rho}{2} \sum_{fac} \sum_{int} \left[\left(\sum_k V^k N_k - \mathbf{v}^{outil} \right)_{int} \cdot \mathbf{n}_{int} \right]^2 \omega_{int} J_{int} \quad (2.9) \end{aligned}$$

où l'indice $_{int}$ indique que l'on prend la valeur au point d'intégration, ω_{int} est le poids associé au point d'intégration et J_{int} est le jacobien de la transformation entre la configuration actuelle et la configuration de référence. Cette méthode présente l'avantage de ne pas augmenter la largeur de bande du système dans le cas où $M_n = N_n$. En effet, on aboutit à des termes de raideur de la forme

$$\mathbf{K}_{ik \ jl}^{pen} = \int_{\partial\Omega_c} N_k n_i N_l n_j dS$$

Cependant, on note sur l'expression (2.9) que la condition de contact n'est pas exprimée aux nœuds de la surface mais plutôt aux points d'intégration. Cela représente généralement plus de conditions que de nœuds de la surface. Sur des surfaces de contact courbes, on aboutit donc généralement à un système surcontraint. Par exemple, une condition de contact glissant peut se transformer en contact collant [Fourment et al.99].

2.1.1.4 La condition de contact unilatéral

On s'est limité jusque là à l'étude d'une condition de contact bilatéral. On introduit maintenant une condition de contact unilatéral où les nœuds de la surface sont relâchés sous certaines conditions de pression. Cette condition est exprimée par les *Conditions de Signorini* que nous avons introduites dans la première partie.

On aboutit à la même formulation que (2.2). Toutefois, la pression de contact λ et la composante normale à la vitesse de glissement doivent être négatives :

$$\left| \begin{array}{l} \text{Trouver } (\mathbf{v}, p, \lambda) \in (\mathcal{V}, \mathcal{P}, H^{-\frac{1}{2}}(\partial\Omega_c)) \text{ tel que } \forall \lambda \leq 0 \text{ et } (\mathbf{v}^*, p^*, \lambda^*) \in (\mathcal{V}, \mathcal{P}, H^{-\frac{1}{2}}(\partial\Omega_c)) \\ \left\{ \begin{array}{l} \int_{\Omega} \mathbf{s} : \boldsymbol{\varepsilon}^* d\omega - \int_{\Omega} p \operatorname{div} \mathbf{v}^* d\omega - \int_{\partial\Omega_c} \boldsymbol{\tau} \cdot \mathbf{v}^* dS + \int_{\partial\Omega_c} \lambda \mathbf{v}^* \cdot \mathbf{n} dS = 0 \\ \int_{\Omega} p^* \operatorname{div} \mathbf{v} d\omega = 0 \\ \int_{\partial\Omega_c} \lambda^* (\mathbf{v} - \mathbf{v}^{outil}) \cdot \mathbf{n} dS \leq 0 \end{array} \right. \end{array} \right.$$

On ne s'intéressera dans la suite qu'à la formulation pénalisée du problème dans le cas du contact unilatéral, le lecteur intéressé par une résolution en multiplicateurs peut se reporter à [Hild98].

Discrétisation éléments finis

Au niveau discret, résoudre le problème avec un contact unilatéral revient à considérer le lagrangien suivant qui est une modification de (2.7) pour prendre en compte l'inégalité de la dernière équation du système précédent.

$$\mathcal{L}_{h,\rho}(\mathbf{v}_h, p_h) = \mathcal{L}_h(\mathbf{v}_h, p_h) + \frac{\rho}{2} \sum_m \left[\int_{\partial\Omega_c} M_m \left((\mathbf{v}_h - \mathbf{v}^{outil}) \cdot \mathbf{n} \right) dS \right]^{+2}$$

où $[x]^+ = \frac{x+|x|}{2}$.

La condition discrète

Le même raisonnement que précédemment peut être mené pour les différents types de discrétisation présentées.

- Le contact nodal

Dans le logiciel de simulation du forgeage **Forge3**, on utilise ce type de discrétisation associée avec une condition de contact unilatéral. On aboutit aux expressions suivantes :

$$(\mathbf{K}^{v\lambda} \mathbf{K}^{\lambda v})_{ikjl} = \delta_{kl} n_i^k n_j^l H((V^k - V^{outil}) \cdot \mathbf{n}^k)$$

où n^k est la normale en X^k , V^{outil} est la valeur de la vitesse d'outil en X^k et H est la fonction de Heavyside définie par :

$$H(x) = \begin{array}{l} 1 \text{ si } x \geq 0 \\ 0 \text{ sinon} \end{array}$$

Les termes introduits dans le second membre du problème sont de la forme :

$$(\mathbf{K}^{v\lambda} \mathbf{R}^\lambda)_{ik} = [(V^k - V^{outil}) \cdot \mathbf{n}^k]^+ n_i^k$$

– Le contact “surcontraint”

Ce type de méthode est utilisée, au laboratoire, avec une condition de contact unilatérale, dans le code **Thercast** de simulation du refroidissement des pièces de fonderie, où les résultats obtenus ont été validés par rapport à l’expérimentation [Jaouen98]. Toutefois, sur de tels problèmes, la surface de contact présente peu de parties courbes et le frottement ne joue aucun rôle dans la modélisation de l’écoulement.

Dans [Habraken et al.98] les auteurs travaillent aussi avec des conditions de contact exprimées aux points d’intégration de la surface de contact. Sur des calculs 2D, ils utilisent jusqu’à 5 points d’intégration par face.

2.1.2 La gestion incrémentale du contact

2.1.2.1 La condition de contact quasi-statique

La description qui suit est faite avec une formulation de contact aux nœuds de la surface de la pièce.

Fonction distance

On introduit la fonction distance à l’outil d . En un point de la pièce, elle représente la distance de ce point à la surface de l’outil. Pour tout point M de la surface de l’outil, on détermine son projeté P sur la surface de l’outil. Par convention d sera positive si le point est à l’extérieur de l’outil, vaudra zéro sur la surface et sera négative à l’intérieur.

Un nœud est donc considéré en contact si $d \leq 0$ (voir la figure 2.2), on peut écrire :

$$d = \mathbf{MP} \cdot \mathbf{n}$$

On veut imposer la condition de non-pénétration sur la surface, on veut donc vérifier :

$$d \geq 0 \text{ sur } \partial\Omega$$

Cette distance à l’outil est calculée pour tous les nœuds de la surface de la pièce.

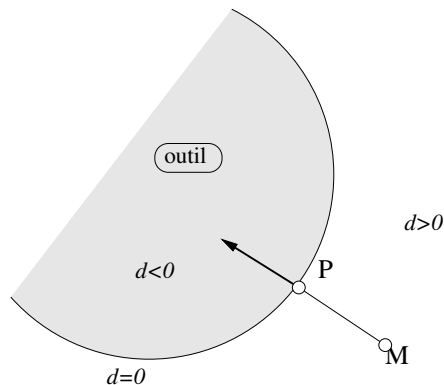


FIG. 2.2 – Définition de la fonction saut du contact

Calcul de la distance

Pour ne pas présenter un coût de calcul exorbitant, le calcul des distances à l'outil utilise une hiérarchisation des facettes décrivant l'outillage. Les facettes sont classées de façon hiérarchiques par leur appartenance à des sphères de tailles croissantes. Cette répartition est faite une fois pour toutes à la lecture du fichier d'outillage.

Dans un second temps, pour ne pas stocker des données de contact pour tous les nœuds de la surface, les normales et distances de contact ne sont calculées que pour des nœuds "proches" de la surface. Pour déterminer numériquement quels sont les nœuds intéressants pour le contact, on se donne une distance minimale autour de l'outil, d_{out} . Cette distance est, à peu près, la distance maximum que peut parcourir un nœud de la pièce en un pas de temps. Si un nœud est à une distance inférieure à d_{out} de l'outil, on lui imposera de respecter les conditions ci-dessus. Sinon, il est considéré comme un nœud de la surface libre (voir figure 2.3).

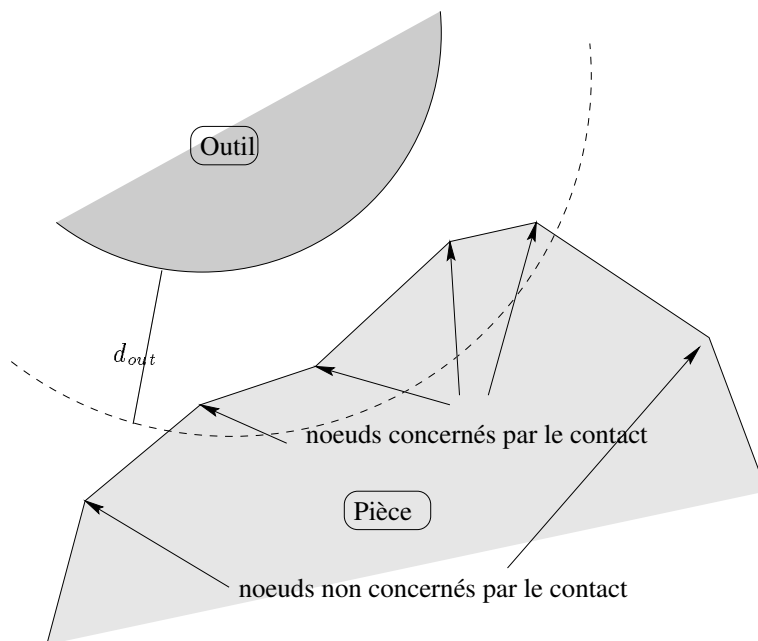


FIG. 2.3 – Définition de la zone d'analyse du contact

2.1.2.2 Le contact incrémental

Le schéma d'intégration

On a vu, dans la première partie, que le schéma d'intégration en temps choisi est un schéma d'Euler explicite. On considère donc la vitesse constante au cours de l'incrément de calcul.

La condition de contact incrémentale

Au cours des incréments de calcul, on veut prendre en compte le déplacement des nœuds en contact qui peuvent engendrer des pénétrations dans l'outil ou des relâchements. La condition de contact est donc modifiée pour empêcher les nœuds de pénétrer dans l'outil. Pour cela, on veut imposer :

$$d^{t+\Delta t} \geq 0$$

On exprime cette condition au temps t après une linéarisation d'ordre 1 en temps :

$$d^{t+\Delta t} = d^t + \frac{d(d^t)}{dt} \Delta t + O(\Delta t^2) \quad (2.10)$$

$$= d^t + \frac{d(\mathbf{M}^t \mathbf{P}^t \cdot \mathbf{n}^t)}{dt} \Delta t + O(\Delta t^2) \quad (2.11)$$

$$= d^t + \frac{d(\mathbf{P}^t - \mathbf{M}^t)}{dt} \cdot \mathbf{n}^t \Delta t + O(\Delta t^2) \text{ car } \frac{d\mathbf{n}^t}{dt} \cdot (\mathbf{M}^t \mathbf{P}^t) = 0 \quad (2.12)$$

$$= d^t + \frac{d(\mathbf{P}^t)}{dt} \cdot \mathbf{n}^t \Delta t - \frac{d(\mathbf{M}^t)}{dt} \cdot \mathbf{n}^t \Delta t + O(\Delta t^2) \\ = d^t + (\mathbf{v}^{outil} - \mathbf{v}) \cdot \mathbf{n}^t \Delta t + O(\Delta t^2) \quad (2.13)$$

On réécrit la condition de contact unilatéral :

$$(\mathbf{v} - \mathbf{v}^{outil}) \cdot \mathbf{n}^t - \frac{d^t}{\Delta t} \leq 0 \quad (2.14)$$

Ce qui conduit aux mêmes équations, en remplaçant simplement au niveau discret la valeur de \mathbf{R}_m^λ par :

$$\mathbf{R}_m^\lambda = \int_{\partial\Omega_c} M_m \left(V^{outil} \cdot n^t + \frac{d^t}{\Delta t} \right) dS$$

2.1.2.3 Le contact explicite

L'algorithme de contact présenté est dit "explicite". Il faut préciser cette appellation. On peut envisager la gestion du contact de deux manières. La première consiste à supposer que la condition de contact reste la même en début et en fin d'incrément. Un nœud en contact reste en contact si la pression de contact est compressive, on lui impose alors $(\mathbf{v} - \mathbf{v}^{outil}) \cdot \mathbf{n} = 0$, et un nœud de surface libre reste de surface libre. Mais cette hypothèse n'est pas vraie puisque certains nœuds peuvent entrer en contact pendant l'incrément. En fin d'incrément, il faut donc rejeter en surface les nœuds qui ont pénétré. Le principal inconvénient de cette méthode réside dans les pertes de volume et de précision engendrés par les reprojctions. Ce sont des algorithmes explicites en temps. Notons que l'on peut améliorer la précision de tels algorithmes par une gestion du pas de temps [Doghri et al.98],[Cescutti89],[Surdon et al.87].

La seconde façon de traiter le contact consiste à vouloir que la condition de contact soit vérifiée à $t + \Delta t$. Pour cela, on écrit la condition à $t + \Delta t$ et on la linéarise pour qu'elle soit exprimée en fonction de variables connues à t . Ce sont les méthodes implicites en temps. C'est ce type de méthode que nous venons de décrire. Mais, les données du contact (normale et distance à l'outil) sont calculées à t et supposées constantes. C'est en cela que la méthode de contact est dite "explicite". Mais avec une telle condition, on aboutit à des problèmes de précision en fin de calcul puisque l'évolution de la surface de contact au cours de l'incrément n'est pas prise en compte. Cette simplification d'ordre 1 revient à approcher localement la surface d'outil par un plan (voir les figures 2.4 sur lesquelles l'outil est vu comme le plan Π^t défini par P^t et n^t et représenté en pointillés). Lorsque l'outil est courbe, il se produit des problèmes de contact. Sur la figure 2.4(a), le nœud ne peut pas déplacer au delà de la ligne pointillée qui est l'approximation plane de l'outil, alors que ce déplacement est licite. Sur la figure 2.4(b), le nœud ne voit pas le changement d'angle de l'outil et le déplacement indiqué est accepté alors que le nœud pénètre dans l'outil. A

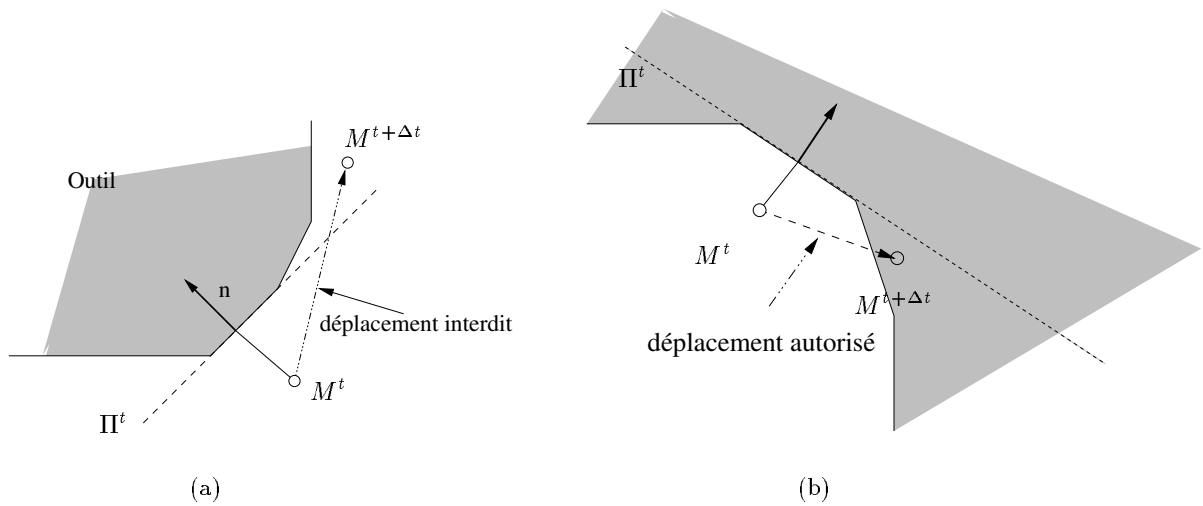


FIG. 2.4 – *Explicite*: L’outil est vu comme le plan Π^t (a) le champ de vitesse proposé est interdit, (b) le champ de vitesse est accepté

l’incrément de temps suivant, la pénétration du nœud est détectée et le nœud est reprojété en surface.

2.1.2.4 Le traitement des coins

Sur la figure 2.4(b), on remarque que le nœud ne voit pas l’angle de l’outil car on identifie l’outillage au plan de la facette d’outil la plus proche. Une manière de résoudre ce type de problème est d’écrire la condition de contact par rapport à toutes les faces de l’outil avec lesquelles il est susceptible d’être en contact. Ainsi, sur la figure 2.5, le nœud reçoit une contribution par rapport aux deux plans Π_1^t et Π_2^t . Ce qui se traduit, au niveau

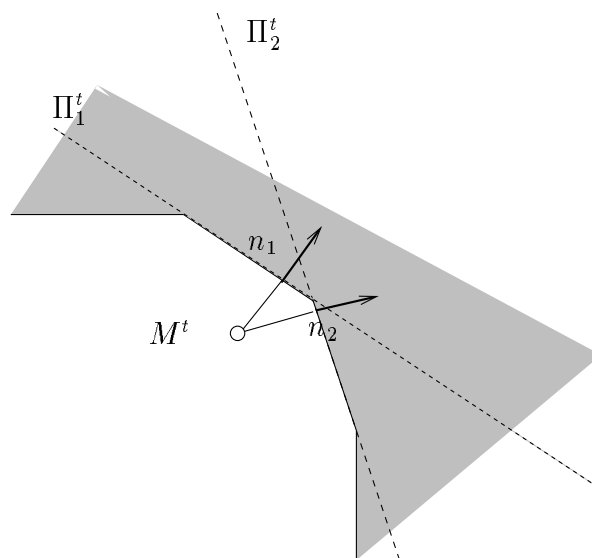


FIG. 2.5 – *Traitement du contact dans les coins*

des équations par résoudre le système :

$$\left| \begin{array}{l} \text{Trouver } (\mathbf{v}, p) \in (\mathcal{V}, \mathcal{P}) \text{ tel que } \forall (\mathbf{v}^*, p^*) \in (\mathcal{V}, \mathcal{P}) \\ \left\{ \begin{array}{l} \int_{\Omega} \mathbf{s} : \dot{\boldsymbol{\varepsilon}}^* d\omega - \int_{\Omega} p \operatorname{div} \mathbf{v}^* d\omega - \int_{\partial\Omega_c} \boldsymbol{\tau} \cdot \mathbf{v}^* dS \\ + \rho \sum_{k \in \partial\Omega_c} \sum_{f \in \mathcal{V}(k)} \left[(V^k - V^{outil}) \cdot \mathbf{n}_f^k - \frac{d_f^k}{\Delta t} \right]^+ \mathbf{n}_{f,i}^k = 0 \\ \int_{\Omega} p^* \operatorname{div} \mathbf{v} d\omega = 0 \end{array} \right. \end{array} \right.$$

où $\mathcal{V}(k)$ est l'ensemble des faces de l'outil avec lesquelles le nœud k est susceptible d'entrer en contact.

Remarque : Cette méthode ne permet pas de résoudre les problème présenté sur la figure 2.4(a).

2.2 Le contact implicite

Motivations

On veut résoudre les problèmes de précision du contact exposés dans le paragraphe précédent. Pour cela, on choisit de modifier l'algorithme de contact en le rendant implicite. Ce type de travaux est assez peu développé dans la littérature sauf, à notre connaissance, dans [Mahajan et al.98] et [Rodic et al.96].

2.2.1 Principe de la méthode

Dans le paragraphe précédent on a vu que la condition de contact à $t + \Delta t$ est approchée à partir de la situation au temps t . Pour éviter les problèmes de précision que cette hypothèse engendre, on choisit d'actualiser la condition de contact à chaque itération de Newton-Raphson à partir de l'itération précédente.

On exprime donc la valeur $d^{t+\Delta t}$ en fonction de la valeur $d_{iter-1}^{t+\Delta t}$, distance du nœud à l'outil au temps t à l'itération $iter - 1$ de Newton-Raphson, par un développement limité du premier ordre en espace. On obtient:

$$\begin{aligned} d^{t+\Delta t} &= d_{iter-1}^{t+\Delta t} + \frac{d(d_{iter-1}^{t+\Delta t})}{dx} \Delta x + O(\Delta x^2) \\ &= d_{iter-1}^{t+\Delta t} - \Delta \mathbf{v}_{iter} \cdot \mathbf{n}_{iter-1}^{t+\Delta t} \Delta t + O(\Delta x^2) \end{aligned}$$

où $\Delta \mathbf{v}_{iter}$ est la variation du champ de vitesse entre la position à l'incrément $iter - 1$ et la position à l'incrément $iter$. $d_{iter}^{t+\Delta t}$ est la distance du point $X_{iter}^{t+\Delta t}$ à l'outil :

$$X_{iter}^{t+\Delta t} = X^t + V_{iter} \Delta t \quad (2.15)$$

$$= X_{iter-1}^{t+\Delta t} + \Delta V_{iter} \Delta t \quad (2.16)$$

On constate que pour faire cette actualisation à partir de l'itération $iter$, il faut connaître la valeur de $d_{iter-1}^{t+\Delta t}$ et $\mathbf{n}_{iter-1}^{t+\Delta t}$. Ceci implique qu'il faut faire le calcul des distances de projection à chaque itération de la procédure de Newton-Raphson. Cela revient à modifier, pour chaque itération, le plan d'approximation de la surface d'outil. Sur la figure 2.6, à l'itération $iter$, l'outil est vu comme le plan $\Pi_{iter-1}^{t+\Delta t}$ et non plus comme le plan Π^t . Après la linéarisation présentée ci-dessus, les contributions du contact implicite à la matrice de raideur et au résidu sont assez différentes de celles exposées en explicite. Par commodité, on souhaite se ramener à une expression fonction de V_{iter} . Pour cela on exprime la distance du point $X_{iter-1}^{t+\Delta t}$ au plan $\Pi_{iter-1}^{t+\Delta t}$ en fonction de V_{iter-1} :

$$d(X_{iter-1}^{t+\Delta t}, \Pi_{iter-1}^{t+\Delta t}) = d_{iter-1}^t + (V^{outil} - V_{iter-1}) \cdot \mathbf{n}_{iter-1}^{t+\Delta t} \Delta t$$

où on a noté $d_{iter-1}^t = d(X^t, \Pi_{iter-1}^{t+\Delta t})$.

En notant que $\Delta V_{iter} + V_{iter-1} = V_{iter}$, on a bien retrouvé une expression de la forme demandée. Le système prend donc une forme similaire à celui du contact explicite (2.14).

$$d^{t+\Delta t} = d_{iter-1}^t + (V^{outil} - V_{iter-1}) \cdot \mathbf{n}_{iter-1}^{t+\Delta t} \Delta t - \Delta \mathbf{v}_{iter} \cdot \mathbf{n}_{iter-1}^{t+\Delta t} \Delta t + O(\Delta x^2)$$

Ce qui donne

$$d^{t+\Delta t} = d_{iter-1}^t - (V_{iter} - V^{outil}) \cdot \mathbf{n}_{iter-1}^{t+\Delta t} \Delta t$$

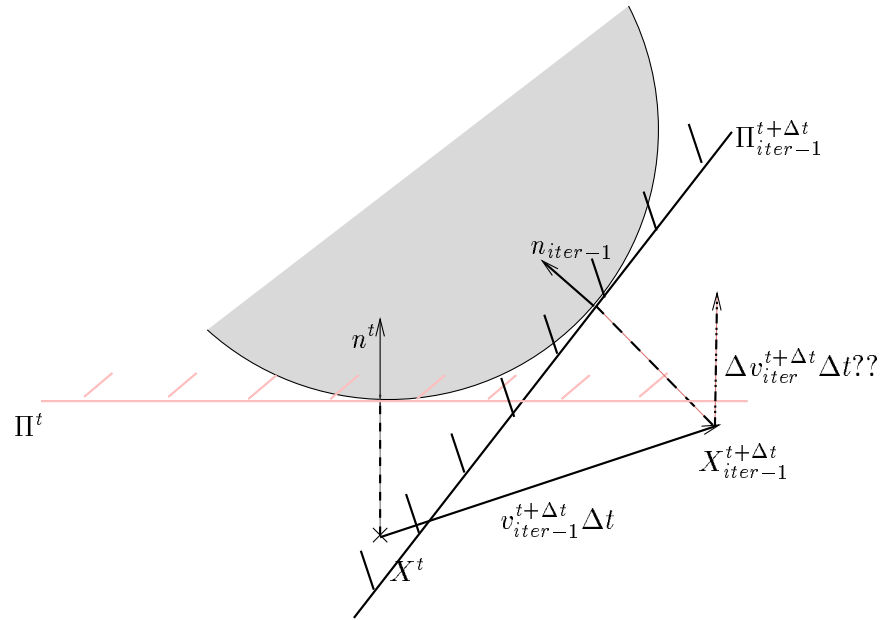


FIG. 2.6 – *Implicite* : Actualisation de l'approximation de l'outillage au cours des itérations

et pour la condition de contact :

$$d_{iter-1}^t - (V_{iter} - V^{outil}) \cdot \mathbf{n}_{iter-1}^{t+\Delta t} \Delta t \leq 0$$

2.2.2 Robustesse de la méthode

On peut mettre en évidence des problèmes de convergence pour la méthode de contact implicite. Sur un cas académique de forgeage de lopin cubique par un outil présentant un angle prononcé (voir figure 2.7(a)), lors du passage de l'angle, on observe un changement brutal de normale. Cette perturbation génère des oscillations numériques. Lors du passage du coin concave, selon que le point se trouve sur une facette ou une autre, l'outil est vu comme le plan Π_1 ou comme le plan Π_2 (voir figure 2.7(b)). Le passage brutal de la position \mathbf{P}_1 à la position \mathbf{P}_2 , d'un état admissible (contact) à un état interdit (pénétration) avec une variation de normale provoque une grande amplitude de variation dans le système à résoudre. De P_2 , le nœud est ramené en \mathbf{P}_3 et de là retourne en \mathbf{P}_2 . Ce qui engendre les oscillations observées.

2.2.2.1 Le lissage d'outil

Pour éviter ce genre de comportement, une approche est de lisser la normale à l'outillage. C'est à dire de construire une normale qui varie continuellement. Nous détaillons dans le paragraphe suivant différentes méthodes de lissage.

2.2.2.2 Différentes méthodes de lissage

- Dans [Rodic et al.96], en 2D, les auteurs choisissent de lisser la normale en faisant une moyenne sur chaque arête. Aux sommets, la normale est calculée comme la moyenne

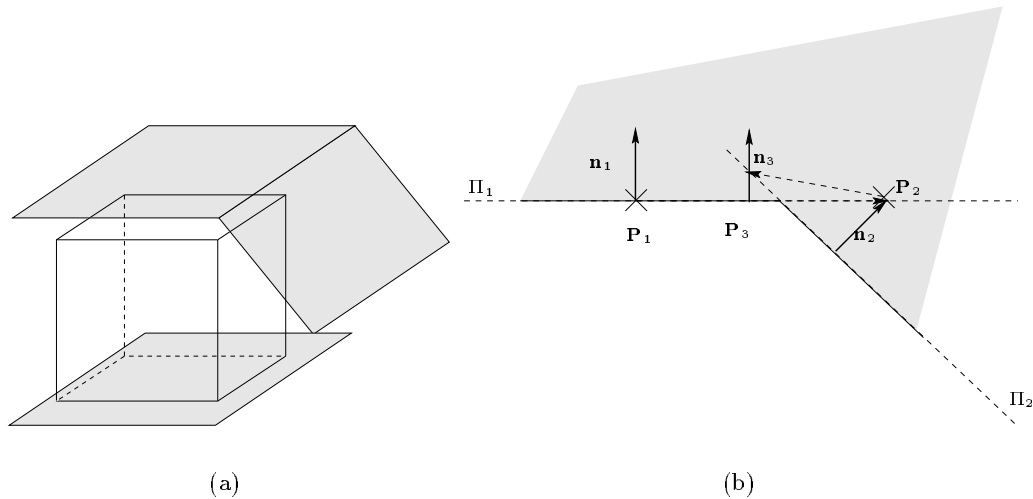


FIG. 2.7 – Cas académique de forgeage

des normales aux deux segments (voir figure 2.8(a)). Ils définissent la normale en tout point d'une arête par combinaison linéaire des normales aux sommets. En 3D, il faudrait calculer la normale en chaque sommet de l'outillage comme moyenne sur les triangles contenant ce sommet. Pour chaque nœud se projetant dans un triangle, il faudrait calculer les coordonnées barycentriques du point de projection. Par la combinaison linéaire des trois normales aux sommets on obtiendrait la normale au point.

L'inconvénient de cette approche est que la "normale lissée" sur une surface plane (triangle de discrétisation) n'est pas constante et n'est pas nécessairement la normale à cette surface. Il en résulte une approximation sur les parties planes de l'outil.

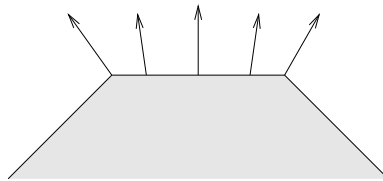


FIG. 2.8 – Méthode de lissage utilisée par [Rodic et al.96]

- Dans le logiciel **Forge2**, on introduit la notion de congé de raccordement sur les arêtes de discrétisation de l'outil. Sur une partie de l'arête proche des sommets, la normale est calculée comme la moyenne de la normale sur l'arête et la normale au sommet (voir figure 2.9). Cette méthode est difficilement applicable en 3D car il faut déterminer la zone sur laquelle on conserve la normale à la face et la zone sur laquelle on fait la moyenne des normales (voir figure 2.10). On obtient différentes procédures suivant que l'on se trouve sur une partie 1 (pas de moyenne), 2 (moyenne pondérée entre deux normales) ou 3 où il est complexe de calculer la normale (moyenne de la normale au sommet et sur les arêtes).

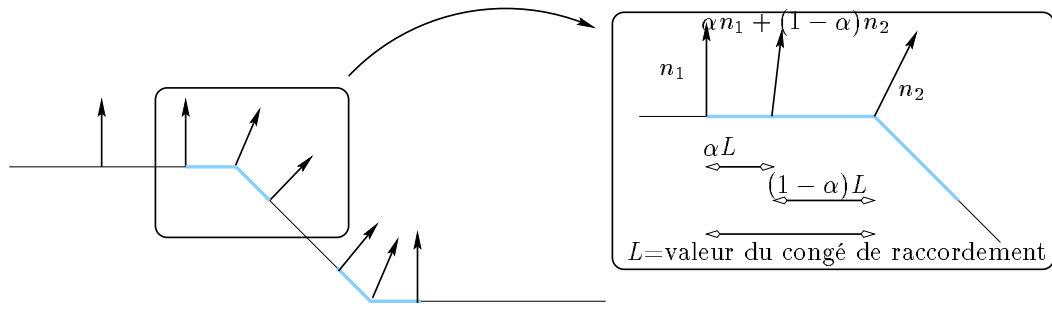
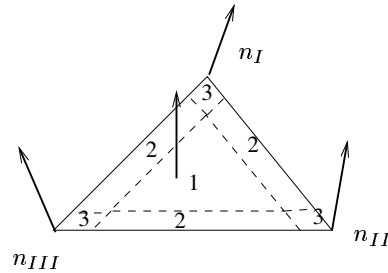
FIG. 2.9 – Lissage des normales par congé de raccordement (*Forge2*)

FIG. 2.10 – Application des congés de raccordement en 3D

2.2.3 La procédure de construction des normales

Pour introduire la méthode de lissage utilisée, nous faisons une parenthèse sur les procédures de construction de normales qui interviennent dans le calcul du terme de contact.

2.2.3.1 La normale de projection

On a vu que l'on devait calculer le point de projection pour les nœuds distants de l'outil, on peut donc se servir de cette projection pour calculer la normale.

$$\mathbf{n} = \frac{\mathbf{MP}}{\|\mathbf{MP}\|}$$

Cependant, pour les nœuds se trouvant exactement sur la surface de l'outil, la projection et le nœud sont confondus. La normale considérée est la normale à la face sur laquelle on se trouve. Si on se trouve sur une arête ou un sommet, on prend la moyenne des arêtes adjacentes. On obtient une normale continue sauf pour les points exactement sur la surface d'outil : c'est à dire la situation la plus fréquente et la plus délicate

Pour éviter les situations où le nœud se trouve exactement sur la surface et où la normale prend une valeur discontinue, on impose aux nœuds en contact d'être situés non pas sur la surface mais légèrement à l'intérieur, d'une distance numérique faible d_{pen} . Ainsi, le point de projection P est distinct du nœud M et ainsi la normale est définie pour les nœuds de contact. La distance d_{pen} est choisie pour induire une pénétration raisonnable (1/10 de la longueur d'une arête).

Cette méthode a pour avantage de donner une définition continue de la normale au passage

des coins concaves de l'outil (voir la figure 2.11). Toutefois, on est obligé d'introduire une pénétration numérique qui nuit à la précision de la détermination de l'interface pièce/outil. Une telle méthode de lissage des normales produit un lissage des angles concaves seulement.

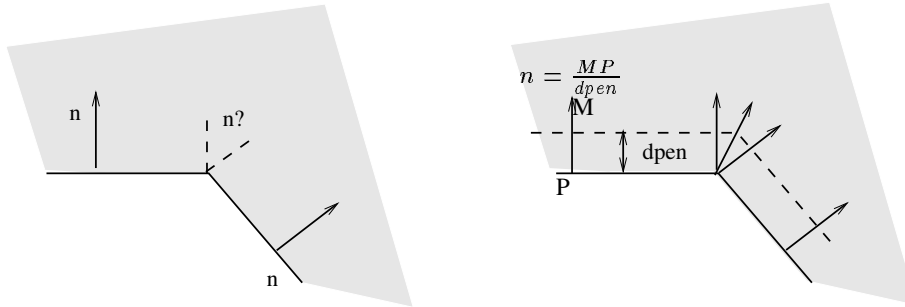


FIG. 2.11 – Définition des normales par le lissage de la surface

Notons que ce lissage des angles est justifié dans le cas de la simulation du forgeage. En effet, les pièces forgées ne peuvent pas être mises en forme dans des matrices présentant des angles vifs. Il s'en suivrait des fissures pour l'outillage ou des problèmes d'extraction de la pièce des matrices [Husson75]. Les angles prononcés sur les pièces finies sont obtenus par usinage, jamais par forgeage. La pièce forgée est donc une forme approchée de la forme voulue sur laquelle les angles ont été arrondis. Ce lissage des outils est bien connu des forgerons. Au niveau du vocabulaire, on parlera de *congé de raccordement*, pour le lissage d'angles convexes et *d'angle d'arrondi*, pour le lissage d'angle concaves.

Malgré le lissage des normales, on observe quand même sur certains cas extrêmes des problèmes de convergence. Pour les résoudre, on souhaite un lissage plus important.

2.2.3.2 Lissage par décalage de l'outil

Pour amplifier le lissage, on choisit d'augmenter la valeur de $dpen$ (voir figure 2.13(b)). On déplace plus loin l'outil (voir figure 2.12), ce qui accentue le lissage des normales mais aussi celui de l'outil. Ce faisant on a augmenté l'imprécision sur le calcul du contact en introduisant une pénétration numérique plus importante.

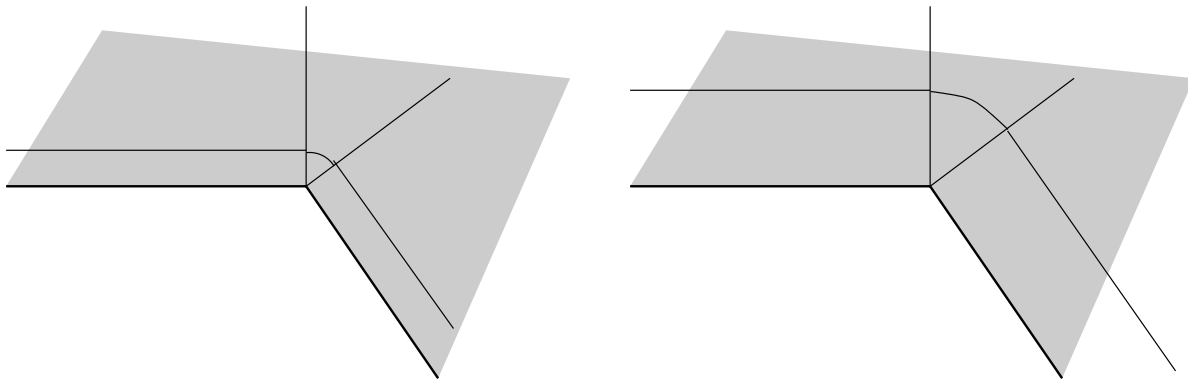


FIG. 2.12 – Lissage par augmentation de la pénétration

Pour conserver ce lissage et un calcul précis, on introduit un outil décalé (voir figure 2.13(c)). Cet outil est déplacé d'une distance de d_{pen} vers l'extérieur de l'outil véritable. Cela permet de ramener les nœuds en surface si on fait le calcul de contact à partir de cet outil (voir figure 2.13(d)).

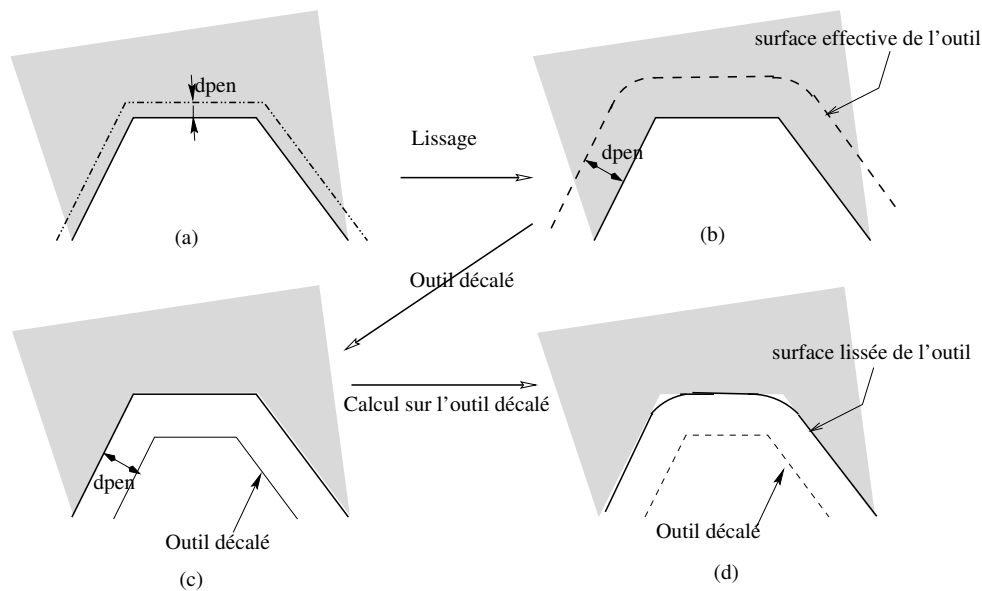


FIG. 2.13 – Lissage de la normale par augmentation de la pénétration d_{pen} et décalage d'outil

Construction de l'outil décalé

- Pour tous les nœuds de l'outil situés sur une partie plane, on décale le nœud suivant la direction normale d'une distance d_{pen} vers l'extérieur de l'outil (voir figure 2.14(a)).
- Pour un nœud situé à l'intersection de deux plans, on calcule les plans parallèles décalés de d_{pen} . Le décalé du nœud est le point de l'intersection des 2 plans, le plus proche (voir figure 2.14(b)).
- Pour un nœud situé à l'intersection de plus de deux plans, son décalé est l'intersection des plans décalés (voir figure 2.14(c)).

Il faut prendre certaines précautions lorsque l'on utilise le décalage des outils. En effet, il faut veiller à ne pas prendre une valeur de d_{pen} trop grande. On pourrait exagérer le lissage des outils et cette approximation ne serait plus justifiée. On pourrait aussi aboutir à des formes d'outil non réalistes (voir figure 2.15).

Le coût d'une telle procédure est négligeable. En effet, cette opération de décalage n'est faite qu'une fois pour toute au début de la simulation. Elle peut aussi être introduite avec une la méthode de gestion du contact explicite. Si elle ne résout pas les problèmes liés à l'approximation de l'outil par une surface plane, elle permet de supprimer la pénétration numérique d_{pen} . On peut voir un exemple de décalage d'outil sur la matrice supérieure du cas du triaxe sur la figure 2.16.

La principale limitation de cette méthode de lissage est qu'elle n'agit que sur les angles concaves de l'outil.

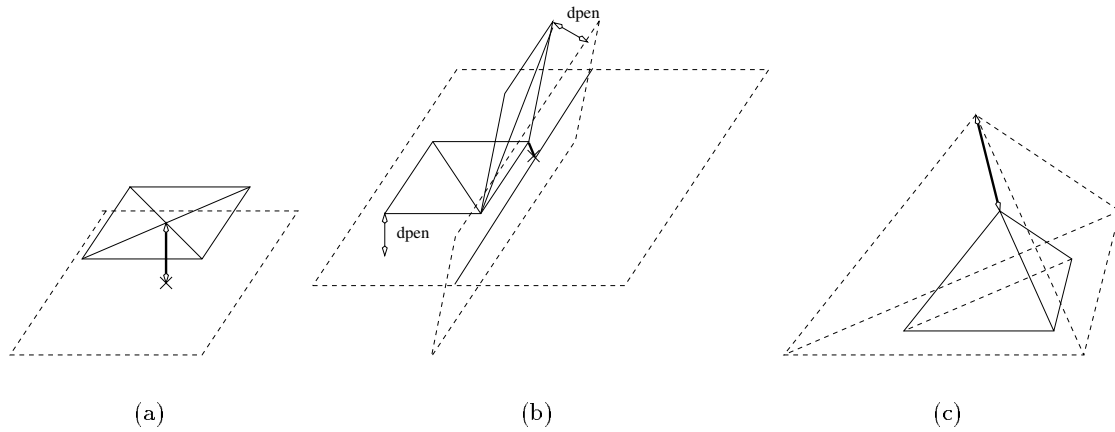


FIG. 2.14 – Détermination des sommets de l'outil décalé

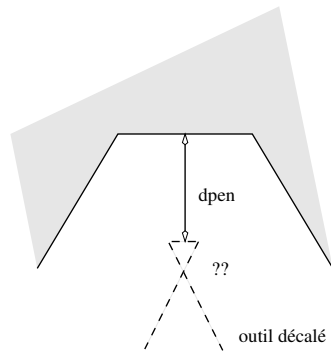


FIG. 2.15 – Utilisation d'un dpen trop grand, illustration 2D

2.2.4 Une méthode de lissage pour les angles concaves et convexes

En suivant l'idée développée dans [Rodic et al.96], on peut proposer une définition continue de la normale basée sur une méthode de moindres carrés. On utilise une famille de fonctions de base $(\bar{N}_k)_k$ et on cherche le champ de normale discrète n_k qui réalise le minimum de la fonctionnelle suivante :

$$\mathcal{M} = \int_{\partial\Omega_{outil}} (n - \sum n_k \bar{N}_k)^2 dS$$

Avec une telle méthode, la normale est lissée partout, même dans les angles convexes de l'outil. La normale est la plus proche possible de la normale d'outil aux points d'intégration. Pour mettre au point cette méthode, il faut cependant choisir une discrétisation éléments finis, assembler un système et le résoudre pour obtenir les n_k , ce qui n'est pas immédiat. Ce calcul n'est toutefois pas coûteux par rapport à la résolution du problème car il n'est fait qu'une seule fois en début de simulation.

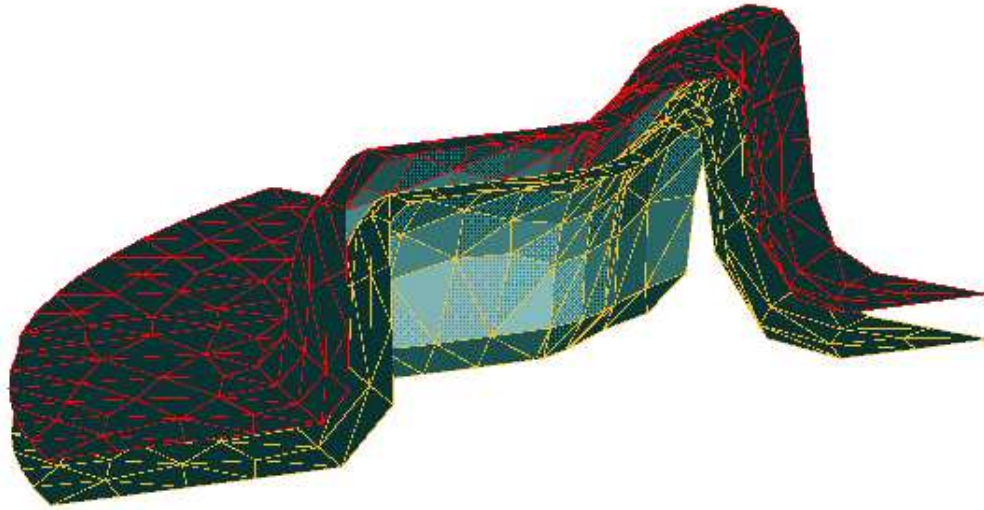


FIG. 2.16 – Outil décalé dans le cas du triaxe

2.2.5 Applications

2.2.5.1 Le cas académique

Le premier cas test, que l'on nommera "cas académique", est l'écrasement d'un lopin cubique de 120mm de coté par un outil avec un angle prononcé. Pour une valeur de $dpen$ de 0.5 mm, le calcul en implicite diverge alors qu'avec une valeur de $dpen$ de 1 mm on obtient la convergence. On a fait le calcul dans le cas d'une rhéologie newtonienne ($m = p_f = 1$) et d'une rhéologie viscoplastique ($m = p_f = 0.139$).

Sur cet exemple, les calculs ont été fait avec décalage d'outil, dans le cas explicite et implicite. Cela montre la précision apportée par l'algorithme implicite sur ce cas extrême.

1. A l'arrivée de la pièce contre l'angle de l'outil, on voit la précision de l'algorithme implicite sur l'algorithme classique sur la figure 2.17. En effet, dans le premier cas la matière pénètre dans l'outil, alors que dans le second, elle est bien maintenue à l'extérieur de l'outillage.
2. On compare les temps de calcul et les pénétrations autorisées avec les deux algorithmes (tableau 2.1). La valeur de $dpen$ est de 1mm. Les temps de calcul sont

		Explicite	Implicite
Newtonien	CPU	8 min 54s	8min 17s
	Pénétration Max.	0.4 mm	$2 \cdot 10^{-5}$ mm
Viscoplastique	CPU	9min 23S	11min 16s
	Pénétration Max.	0.38 mm	$2 \cdot 10^{-4}$ mm

TAB. 2.1 – Comparaison des temps de calcul et des précisions des algorithmes de contact sur le cas académique

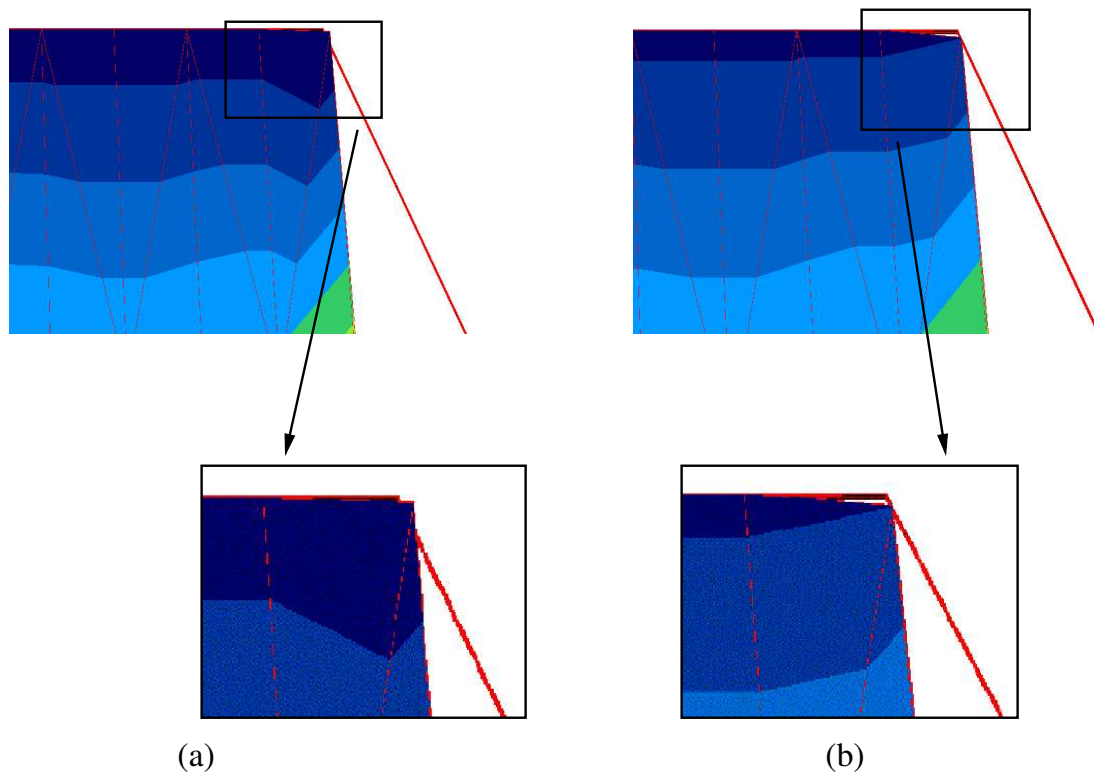


FIG. 2.17 – Arrivée de la matière contre l’angle de l’outillage, (a) avec l’algorithme explicite, (b) avec l’algorithme implicite,

comparables pour les deux algorithmes. Sur le cas viscoplastique la différence n’est pas énorme. Par contre la précision de l’algorithme représentée ici par la pénétration maximale des nœuds dans l’outil au cours du calcul est nettement meilleure pour l’algorithme implicite.

2.2.5.2 Le forgeage partiel d’un triaxe

On a comparé ces algorithmes sur le forgeage d’une pièce automobile, le triaxe (figure 2.18). La pièce finale comporte environ 7000 nœuds. Les calculs ont été menés avec une rhéologie newtonienne et une rhéologie viscoplastique. Les résultats sont donnés dans le tableau 2.2 pour une valeur de d_{pen} de 0.1 mm. La ligne intitulée “précision” donne les pénétrations numériques maximales relevées au cours du calcul (hors remaillage). Dans le cas explicite, les calculs ont été faits sans décalage d’outil pour montrer les apports de l’ensemble de nos travaux sur la gestion du contact. On montre sur le tableau 2.2 que

	Explicite	Implicite
CPU	14h 47 min	18h
Précision	0,15 mm	$4 \cdot 10^{-3}$ mm

TAB. 2.2 – Comparaison de temps de calcul et de précision pour le forgeage à 48 % du triaxe pour une rhéologie viscoplastique

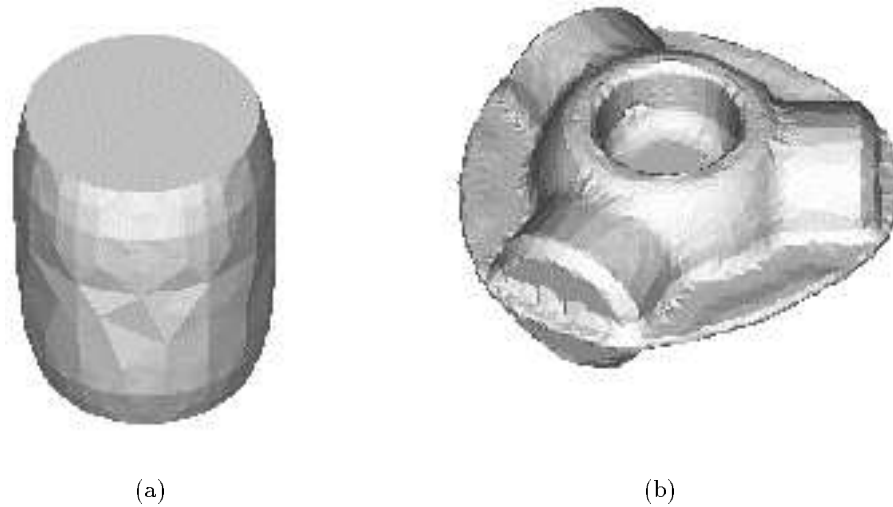


FIG. 2.18 – Cas du triaxe (a) lopin initial (b) forme finale

le contact implicite est plus précis et à peine moins rapide que le contact explicite. On peut expliquer ces résultats par le fait qu'en fin d'incrément implicite, le contact est bien vérifié, alors que pour le cas explicite, il peut y avoir des pénétrations qu'il faut corriger en début d'incrément suivant. Le temps passé à calculer des actualisations du contact en implicite est compensé par le temps qui est nécessaire pour les corrections de pénétration en explicite.

Dans [Mahajan et al.98] lors de l'implémentation en 2D, pour certains cas, le calcul implicite apparaît même plus rapide que le calcul explicite.

2.2.5.3 Le forgeage d'un joint homocinétique

La configuration initiale de ce cas de forgeage est présentée sur la figure 2.19. Le lopin initial comporte 1456 nœuds pour 5625 éléments. la déformation se fait par filage de la matière le long de l'outil inférieur et supérieur.

La pièce finale est représentée sur la figure 2.20. Elle comporte 42427 nœuds pour 154012 éléments. On compare les résultats obtenus pour 60 % du forgeage pour les deux algorithmes de contact dans le tableau 2.3.

Sur ce cas, l'algorithme implicite est beaucoup plus coûteux en temps CPU que l'algo-

	Explicite	Implicite
CPU	4h 36min	23h 36 min
Précision	0,4 mm	$7 \cdot 10^{-3}$ mm

TAB. 2.3 – Comparaison de temps de calcul et de précision sur le cas de filage

gorithme explicite. L'apport en précision est toutefois non négligeable.

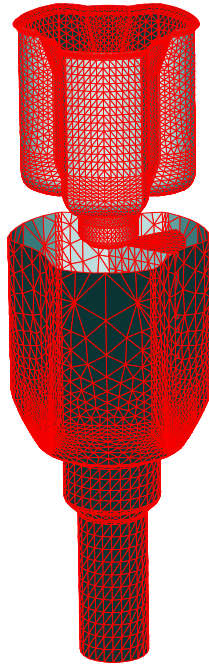


FIG. 2.19 – *Présentation du cas de filage*

2.2.6 Conclusions

On a développé un algorithme de contact implicite destiné à résoudre les problèmes de précision de l'algorithme de **Forge3** lorsque les pas de temps ne sont pas aussi petits qu'il le faudrait pour assurer la précision. Les résultats ont montré que la nouvelle méthode est effectivement plus précise, mais aussi que, dans certains cas, elle n'est pas beaucoup plus coûteuse. La précision obtenue est directement liée à la précision de la méthode de pénalisation et ne dépend plus du pas de temps.

Ce travail repose sur une méthode de lissage des normales qui permet de rendre plus robuste l'algorithme implicite. Cette méthode ne prend en compte que les angles concaves de l'outillage, ce qui semble suffisant sur les cas de forgeage étudiés. Une autre méthode plus générale de lissage des angles convexes et concaves pourrait aisément être utilisée si cette limitation posait des difficultés pour d'autres procédés.

Enfin, on a introduit une technique de décalage de l'outillage qui permet d'éliminer les pénétrations numériques induites par la méthode de calcul des normales. Elle est applicable dans les algorithmes explicite et implicite.

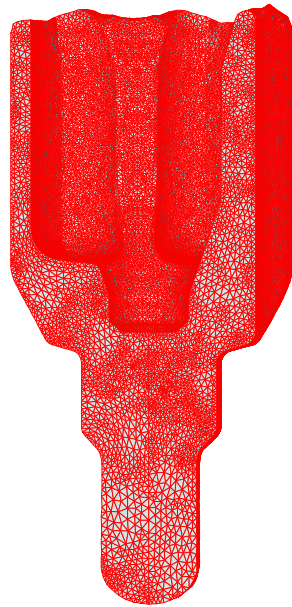


FIG. 2.20 – *Forme de la pièce finale du cas de filage*

Chapitre 3

LES MÉTHODES MULTIGRILLES

Introduction

Le but de notre étude est d'intégrer un solveur plus compétitif que l'algorithme de résidu minimal qui est utilisé dans le logiciel **Forge3**. Pour cela, on s'est dirigé vers les méthodes multigrilles qui présentent des taux de convergence asymptotique très intéressants.

Ce nouveau solveur doit pouvoir être compatible avec la stratégie parallèle utilisée dans le logiciel. Il doit être efficace pour la résolution des systèmes issus de la discrétisation de problèmes incompressibles. Pour cela, on a choisi d'utiliser l'algorithme de résidu minimal comme lisseur de la méthode et on s'est inspiré des travaux présentés dans [Le dain91] proposant le gradient conjugué comme lisseur.

On s'est ensuite efforcé d'adapter cette méthode multigrille aux particularités du système à résoudre et du problème de forgeage dans son ensemble :

- gestion du contact basée sur une méthode de pénalisation et qui génère donc des termes très grands qui peuvent perturber la résolution du système,
- répartition hétérogène de la viscosité qui perturbe également le conditionnement de la matrice,
- déformations subies par le maillage.

Nous allons donc présenter la méthode de résidu minimal ainsi que les différents préconditionneurs qui peuvent y être associés. Ensuite nous exposerons le principe des méthodes multigrilles et nous nous intéresserons à leur application aux problèmes incompressibles. Enfin, nous présenterons le solveur que nous avons développé en présentant ses performances sur des cas académiques de forgeage à la fois 2 et 3 grilles, puis les améliorations que nous avons dû introduire pour traiter les cas de forgeage plus complexes, ce qui nous permettra de terminer sur différents cas d'application.

3.1 La méthode de résidu minimal

Comme nous l'avons vu dans le premier chapitre, la résolution du problème de forgeage conduit à la résolution d'un système en vitesse et pression. L'utilisation d'un élément fini enrichi d'un terme bulle nous permet d'obtenir un système condensé qui ressemble, dans sa forme, à une formulation du problème de Stokes stabilisé.

En simulation tridimensionnelle, le nombre de degrés de liberté est souvent très élevé. On aboutit donc à des systèmes de grande taille. Dans un logiciel tel que **Forge3**, la résolution de ces systèmes représente environ 80 % du temps de simulation. Il est donc fondamental d'utiliser une méthode de résolution efficace. La forme particulière de notre matrice restreint les possibilités de choix du solveur.

Une méthode de type gradient conjugué, *la méthode de résidu minimal* a été utilisée [Marie97]. Nous allons la présenter et décrire ses propriétés de convergence.

3.1.1 Présentation du problème

Les méthodes de gradient conjugué ne permettent d'inverser que des systèmes $\mathcal{A}x = b$ dont la matrice est symétrique définie positive. Le principe est de minimiser la fonctionnelle \mathcal{J} définie par :

$$\mathcal{J}(x) = \frac{1}{2}(\mathcal{A}x, x) - (b, x)$$

Si \mathcal{A} est non définie positive, cette fonctionnelle n'est pas strictement convexe et l'existence et l'unicité de la solution du problème de minimisation n'est pas vérifiée.

Nous avons vu dans le chapitre 1 que la matrice de notre système n'est pas définie positive. Il faut donc trouver une autre méthode de résolution. La méthode choisie ici est *la méthode du résidu minimal* qui est une méthode de type gradient conjugué qui peut être utilisée sur des systèmes de Stokes stabilisé [Silvester et al.94], [Wathen et al.93]. Elle est aussi appelée méthode de résidu conjugué préconditionné. Son principe est de chercher le minimum de la norme du résidu définie par :

$$\mathcal{E}(r) = (r, r) \tag{3.1}$$

où r est le résidu :

$$r = b - \mathcal{A}x$$

On minimise donc la fonctionnelle :

$$\mathcal{E}(r) = (b, b) - 2(\mathcal{A}x, b) + (\mathcal{A}x, \mathcal{A}x)$$

Cette fonctionnelle est bien strictement convexe car $A^t A$ est définie positive; le minimum de la fonctionnelle est bien la solution du problème $\mathcal{A}x = b$. L'algorithme de résolution par minimisation du résidu est donné dans le tableau 3.1 pour un système préconditionné par une matrice \mathcal{M} . Les méthodes de résidus conjugués sont basées sur les relations de conjugaisons entre les différentes pentes p_k de la résolution. La méthode de résidu minimal est le cas particulier où l'on n'utilise qu'une seule relation de conjugaison ($s = 0$) et la méthode Orthomin correspond au cas particulier où $s = k$, c'est-à-dire au cas où l'on stocke toutes les directions de descente déjà calculées.

Algorithme 3.1 Algorithme de minimisation du résidu

Initialisations:

choix du nombre de pente (s+1)

$$r_0 = b$$

$$p_0 = \mathcal{M}^{-1}r_0$$

$$z_0 = \mathcal{A}p_0$$

répète: Itération k

$$\alpha_k = \frac{(r_k, \mathcal{M}^{-1}z_k)}{(\mathcal{M}^{-1}z_k, z_k)}$$

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

$$r_{k+1} = r_k - \alpha_k z_k$$

$$j = k - s_k - 1 \text{ avec } s_k = \text{Min}(k, s)$$

répète:

$$j = j + 1$$

$$\beta_j^k = -\frac{(\mathcal{A}\mathcal{M}^{-1}r_{k+1}, \mathcal{M}^{-1}z_j)}{(\mathcal{M}^{-1}z_j, z_j)}$$

jusqu'à $j = k$

$$p_{k+1} = \mathcal{M}^{-1}r_{k+1} + \sum_{j=k-s_k}^k \beta_j^k p_j$$

$$z_{k+1} = \mathcal{A}\mathcal{M}^{-1}r_{k+1} + \sum_{j=k-s_k}^k \beta_j^k z_j$$

jusqu'à convergence: réduction relative du résidu voulue

fin algorithme

3.1.2 Préconditionnement

Une méthode classique pour accélérer la convergence d'une méthode itérative est d'introduire une matrice de preconditionnement du système \mathcal{M} . Cela revient à résoudre le problème preconditionné :

$$\mathcal{M}^{-1}\mathcal{A}x = \mathcal{M}^{-1}b$$

Si \mathcal{M} est symétrique définie positive, on peut appliquer la méthode de résidu minimal, en introduisant la nouvelle norme à minimiser définie par :

$$\mathcal{E}(r) = (\mathcal{M}^{-1}r, r)$$

Si \mathcal{M}^{-1} est proche de \mathcal{A}^{-1} , alors le conditionnement de $\mathcal{M}^{-1}\mathcal{A}$ est proche de 1 et la méthode itérative sera d'autant plus efficace.

Le choix de \mathcal{M} peut déterminer des conditions de convergence de la méthode. Ainsi, si \mathcal{M} est symétrique définie positive, la convergence de la méthode de résidu minimal preconditionné est monotone. On trouve plusieurs types de preconditionneurs dans la littérature, en particulier :

- Les matrices diagonales blocs qui ont été introduites dans le cas de la modélisation du forgeage dans [Marie97].
- Les factorisations incomplètes que nous avons développées en utilisant les travaux de [Ramage et al.94].

3.1.2.1 Les preconditionneurs blocs diagonaux

Ces preconditionneurs sont construits à partir des termes de la matrice qui couplent les degrés de liberté d'un même nœud . Dans notre cas, on calcule en chaque nœud 4 inconnues (3 pour la vitesse et 1 pour la pression). Le preconditionneur bloc est donc construit à partir de blocs 4x4 (voir figure (3.1)).

Plusieurs matrices de preconditionnement peuvent être construites à partir de cette matrice diagonale bloc.

Preconditionneur diagonal

La matrice de preconditionnement est la valeur absolue de la diagonale (voir figure 3.2(a)). On note \mathcal{M}_{diag} cette matrice. On choisit de prendre la valeur absolue de ces termes pour que le preconditionneur soit défini positif. Son utilisation assure donc la convergence monotone de l'algorithme. Elle présente aussi l'avantage d'être peu coûteuse en place mémoire (1 vecteur à stocker) et en temps de calcul (son inverse est facile à calculer). Elle est cependant moins efficace que les preconditionneurs suivants.

Preconditionneur bloc vitesse et valeur absolue de la diagonale

Le bloc diagonal vitesse pour chaque nœud est une matrice symétrique définie positive. La matrice de preconditionnement \mathcal{M}_{ivp} est définie par l'inverse des blocs 3×3 en vitesse et la valeur absolue du terme diagonal en pression (voir figure 3.2(b)). Elle est définie positive.

Preconditionneur bloc vitesse pression

La matrice de preconditionnement \mathcal{M}_{tot} est la matrice des blocs diagonaux 4x4 vitesse/pression (voir figure 3.2(c)). Une telle matrice n'est pas définie positive et son utilisation ne garantit donc pas la convergence monotone de la méthode de résidu minimal preconditionnée.

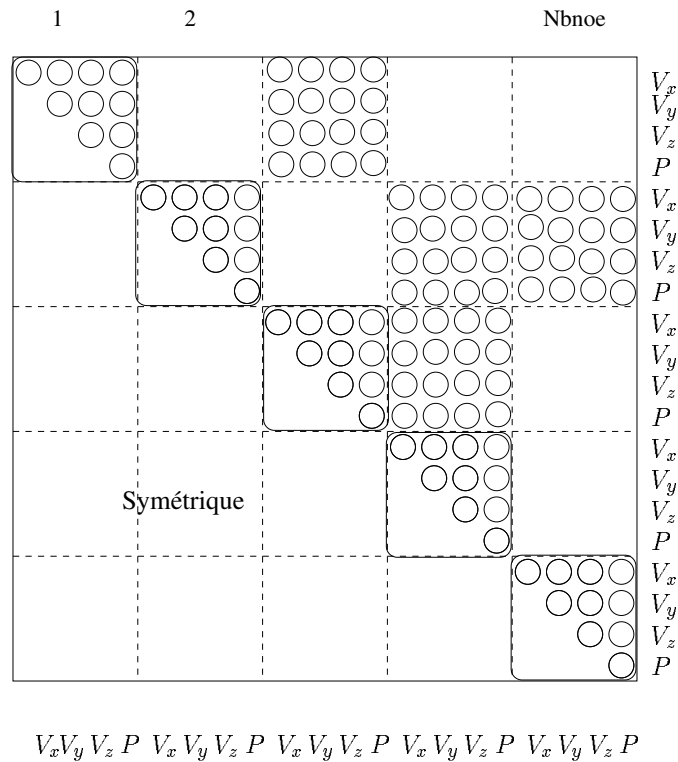


FIG. 3.1 – *Forme de la matrice et termes influencés par le préconditionneur diagonal bloc*

Ces préconditionneurs sont peu coûteux en terme de stockage car on ne stocke que la partie symétrique des blocs 4x4.

On a vu dans le paragraphe 2.1.1.3 que les termes de la matrice où intervient la condition de contact sont les termes couplant les degrés de liberté d'un même nœud, ils se trouvent donc sur les blocs diagonaux. Les deux derniers préconditionneurs prennent en compte ces termes qui sont de forts poids par rapport au reste de la matrice et qui déconditionnent le système.

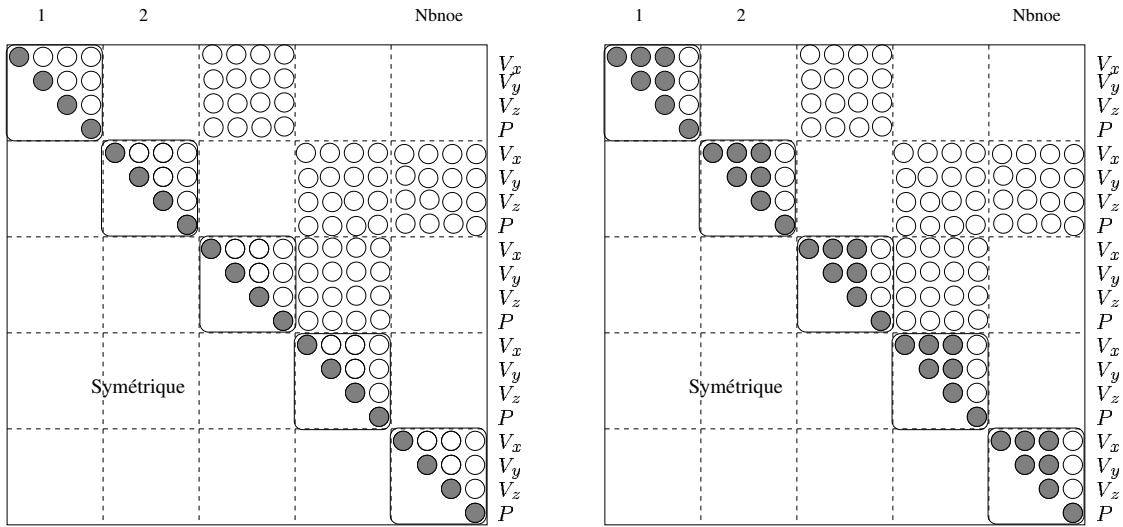
L'utilisation de ces préconditionneurs est basée sur les travaux de Wathen et Silvester [Wathen et al.93], [Silvester et al.94].

Sans entrer dans le détail des calculs, on peut simplement indiquer que l'utilisation de la matrice \mathcal{M}_{diag} donne de moins bons résultats que \mathcal{M}_{ivp} qui est elle même moins efficace que \mathcal{M}_{tot} . Cependant la convergence avec les deux premiers préconditionneurs est monotone alors qu'avec le dernier, elle ne l'est pas. Même si la matrice \mathcal{M}_{tot} n'est pas définie positive, il n'existe pas, à notre connaissance, d'exemples où son utilisation entraîne la non convergence du calcul.

3.1.2.2 Les préconditionneurs de type factorisation incomplète

Une autre classe de préconditionneurs possible est celle des décompositions incomplètes de la matrice [Saad88], [Arany96], [Papadrakakis et al.93] ou des décompositions incomplètes modifiées [Silvester et al.94].

Décomposition de Crout

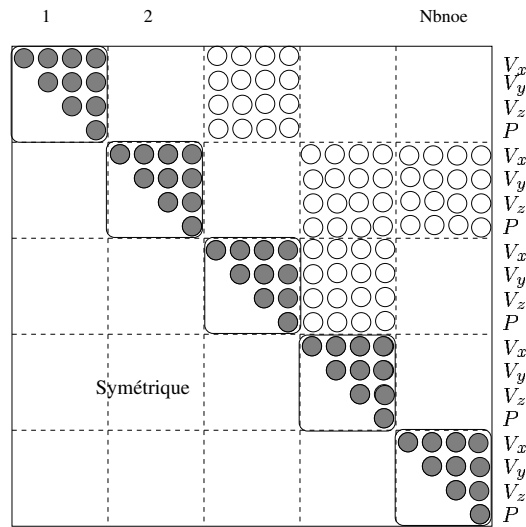


$V_x V_y V_z P$ $V_x V_y V_z P$ $V_x V_y V_z P$ $V_x V_y V_z P$ $V_x V_y V_z P$

(a)

$V_x V_y V_z P$ $V_x V_y V_z P$ $V_x V_y V_z P$ $V_x V_y V_z P$ $V_x V_y V_z P$

(b)



$V_x V_y V_z P$ $V_x V_y V_z P$ $V_x V_y V_z P$ $V_x V_y V_z P$ $V_x V_y V_z P$

(c)

FIG. 3.2 – Termes de la matrice concernés par le préconditionneur (a) préconditionneur diagonal, (b) Préconditionneur bloc vitesse et valeur absolue de la diagonale, (c) Préconditionneur bloc vitesse pression

On considère une décomposition de Crout.

$$\mathcal{A} = \bar{L}\bar{D}\bar{L}^t$$

\bar{L} est une matrice triangulaire inférieure dont la diagonale est 1 et \bar{D} est une matrice diagonale. Une condition suffisante pour qu'une telle décomposition existe est que \mathcal{A} soit symétrique. À partir de cette décomposition, on introduit des décompositions incomplètes de différents ordres. Leur ordre dépend du nombre de coefficients de la matrice de factorisation L qui sont stockés. On n'utilisera ici que des décompositions d'ordre 0.

Décomposition incomplète de Crout d'ordre 0 (IC)

Pour une décomposition incomplète d'ordre 0, on ne calcule pas \bar{L} et \bar{D} mais L et D telles que le stockage compact de L soit identique à celui de \mathcal{A} . On ne calcule que les termes L_{ij} tels que $\mathcal{A}_{ij} \neq 0$.

$$\forall (i, j), i < j \quad / \mathcal{A}_{ij} \neq 0, \quad \bar{L}_{ij} = \frac{1}{\bar{D}_j} \left(\mathcal{A}_{ij} - \sum_{k=1, \mathcal{A}_{ik} \neq 0 \text{ et } \mathcal{A}_{kj} \neq 0}^{j-1} \bar{L}_{ik} \bar{L}_{kj} \bar{D}_k \right)$$

$$\forall i, \quad \bar{D}_i = \mathcal{A}_{ii} - \sum_{k=1, \mathcal{A}_{ik} \neq 0}^{i-1} \bar{L}_{ik}^2 \bar{D}_k$$

Décomposition incomplète de Crout modifiée (MIC)

(MIC : Modified Incomplete Crout decomposition)

Pour une décomposition incomplète modifiée d'ordre 0, on calcule aussi les termes L_{ij} tels que $\mathcal{A}_{ij} = 0$ mais on ne les stocke pas, on les ajoute au terme diagonal D_i . On cherche à vérifier la condition suivante :

$$\forall i \sum_j \mathcal{A}_{ij} = \sum_j \sum_{k=1}^j L_{ik} L_{kj} D_k$$

Lorsque l'on utilise un stockage compact sur une matrice creuse, ce type d'information représente un surcoût très important. L'existence de la décomposition incomplète a été montrée pour certains types de matrices (en particulier les M- matrices). Dans le cas plus général des matrices symétriques définies positives, l'existence de cette décomposition n'est pas garantie.

3.1.2.3 Préconditionneur sur la matrice vitesse ou la matrice vitesse-pression

Nous avons testé deux types de décompositions incomplètes. En nous inspirant des différents preconditionneurs blocs diagonaux, nous avons comparé un preconditionneur de type factorisation incomplète sur la matrice globale que l'on notera "ICvp" et un preconditionneur factorisation incomplète sur la matrice vitesse uniquement, en prenant la valeur absolue du terme diagonal en pression, "IC" (voir figure (3.3)).

3.1.3 Résultats

On compare dans ce paragraphe les performances des différents preconditionneurs tant en coût de stockage qu'en coût de résolution CPU.

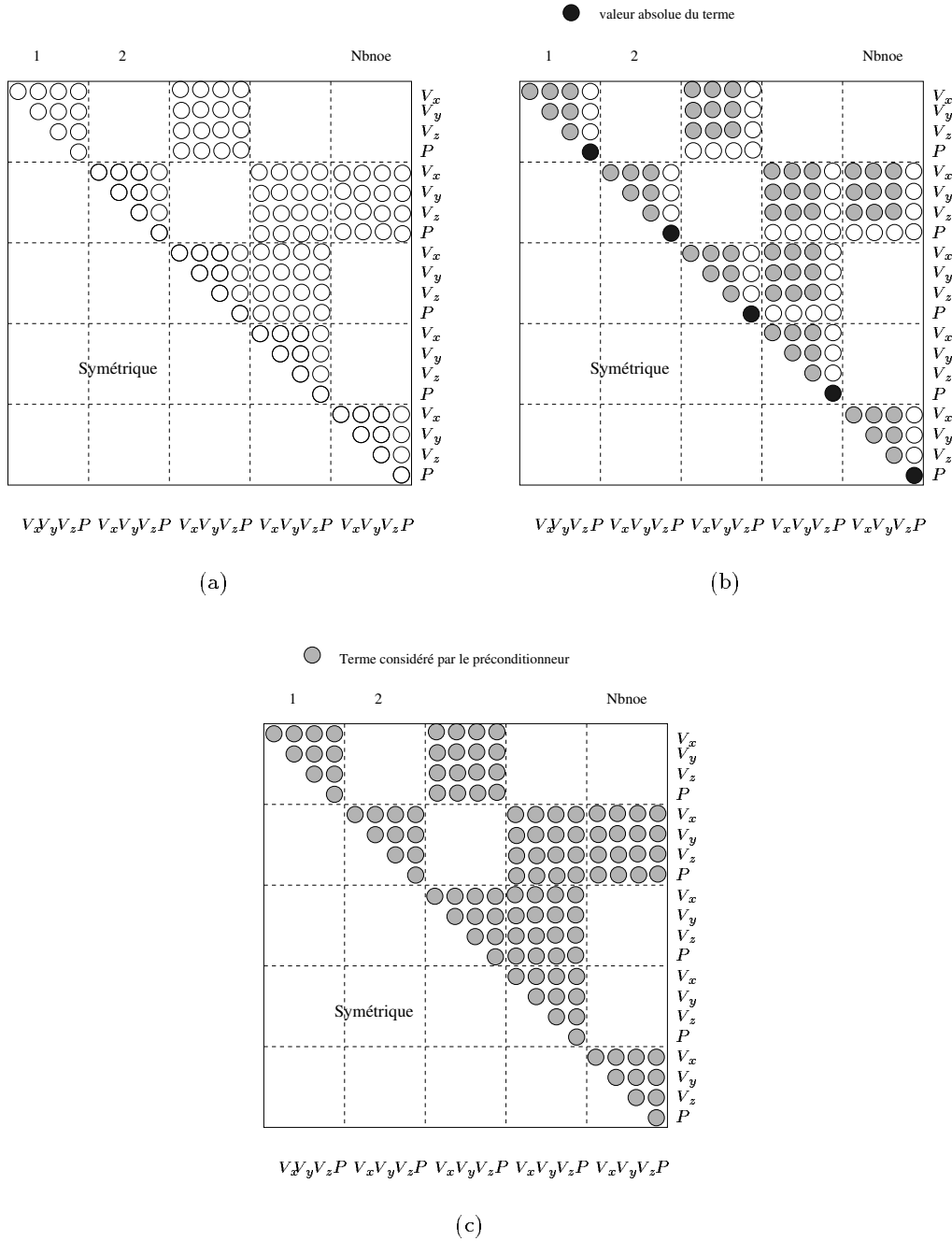


FIG. 3.3 – (a) Termes non nuls de la matrice, (b) Préconditionneur sur le bloc vitesse et valeur absolue du terme diagonal en pression, IC, (c) Préconditionneur sur la matrice globale ICvp

3.1.3.1 Stockage

Les matrices de préconditionnement par factorisation incomplète sont plus coûteuses à stocker. On présente la comparaison de la taille des stockages dans le tableau 3.1. Pour un stockage morse, le nombre de termes est donné par le produit du nombre de nœuds du maillage par le nombre de degré de liberté par nœud (4 pour ICvp, 3 pour IC), multiplié par le nombre de voisins géométriques du nœud considéré multiplié par le nombre de degré de liberté par nœud. Le nombre moyen de voisins pour un nœud d'un maillage 3D est estimé à 12 [Marie97].

	Diagonal bloc matrice M_{tot}	IC matrices L et D	ICvp matrices L et D
Stockage du préconditionneur	$10Nbnoe$	$109Nbnoe$	$192Nbnoe$
Stockage total $A +$ préconditionneur	$202Nbnoe$	$301Nbnoe$	$384Nbnoe$

TAB. 3.1 – Comparaison des coûts de stockage des préconditionneurs en nombre de termes matriciels

3.1.3.2 Temps CPU

Inversion des préconditionneurs

A chaque itération, il faut calculer les préconditionneurs et leur inverse. Pour les préconditionneurs bloc, on inverse effectivement les matrices alors que pour les factorisations incomplètes, pour calculer $\mathcal{M}^{-1}x$, on résout le problème :

$$\left| \begin{array}{l} \text{Trouver } y \text{ tel que} \\ LDL^t y = x \end{array} \right.$$

La complexité du calcul des différentes factorisations est indiquée dans le tableau 3.2.

On notera que les coûts de calcul des matrices L et D sont fortement dépendants de la

Diagonal bloc		IC		ICvp	
Construction de \mathcal{M}^{-1}	Calcul de $\mathcal{M}^{-1}x$	Construction de L et D	calcul de $\mathcal{M}y = x$	Construction de L et D	calcul de $\mathcal{M}y = x$
213 Nbnoe	16 Nbnoe	$81Nbnoe * nbv^3$ ^a	$5Nbnoe + 36Nbnoe * nbv$	$256Nbnoe * nbv^3$	$6Nbnoe + 64Nbnoe * nbv$

TAB. 3.2 – Complexité du calcul et de l'utilisation des préconditionneurs

^a nbv est le nombre de voisins moyen pour un nœud, il peut être estimé à 12 pour un maillage 3D, le point important est qu'il peut être borné indépendamment du maillage

nature du stockage des matrices. Pour une factorisation de Crout complète en stockage bande, le coût de calcul est en $N * Lbg^2$ où N est le nombre de degrés de liberté et Lbg la largeur de bande de la matrice.

Convergence asymptotique

Le taux de convergence d'un préconditionneur utilisant une décomposition incomplète modifiée de Cholesky sur la matrice globale est estimé en $O(N^{\frac{5}{4}})$ par [Ramage et al.94] (voir tableau 3.3), alors que pour un préconditionneur bloc diagonal, elle est en $O(N^{\frac{3}{2}})$, ce taux de convergence théorique ayant été vérifié numériquement par [Marie97] et dans la suite de nos travaux.

	Diagonal bloc	ICvp
taux de convergence	$O(Nbnoe^{\frac{3}{2}})$	$O(Nbnoe^{\frac{5}{4}})$

TAB. 3.3 – Comparaison asymptotique des préconditionneurs

3.1.3.3 Comparaisons sur un cas de forgeage

Sur un exemple d'écrasement d'un lopin cubique entre deux tas plats, on a comparé les résultats asymptotiques numériques d'une résolution par la méthode de résidu minimal préconditionné par bloc diagonal et par décompositions incomplètes de Crout. Sur la figure 3.4, on montre les résultats obtenus et les asymptotes déterminées numériquement.

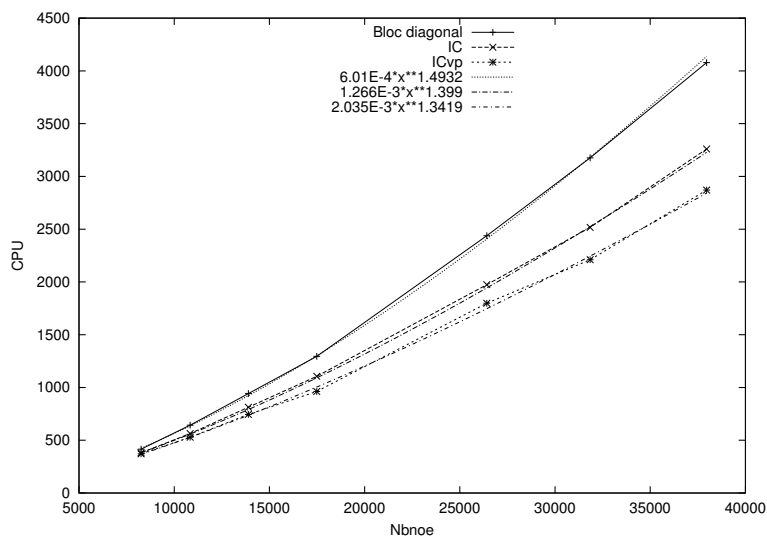


FIG. 3.4 – Comparaison asymptotique des préconditionneurs bloc diagonal et Crout incomplet sur un cas d'écrasement entre tas plats

On remarque que l'on retrouve bien un comportement en $O(Nbnoe^{\frac{3}{2}})$ pour le préconditionneur diagonal bloc. Cependant pour le préconditionneur Crout incomplet on ne retrouve pas un comportement en $O(Nbnoe^{\frac{5}{4}})$ mais plutôt en $O(Nbnoe^{1,34})$. L'implémentation dans le code **Forge2** de ce même préconditionneur conduit aux mêmes constatations. On est donc amené à penser que l'estimation faite dans [Ramage et al.94] est assez optimiste.

On remarque que la convergence des méthodes utilisant des décompositions incomplètes

est meilleure que celle du préconditionneur diagonal bloc. De plus, le préconditionneur ICvp donne de meilleurs résultats que le préconditionneur IC. Sur la figure 3.5, on représente le pourcentage du temps pour la factorisation et pour résolution. On note que le temps nécessaire à la factorisation est en relatif plus important

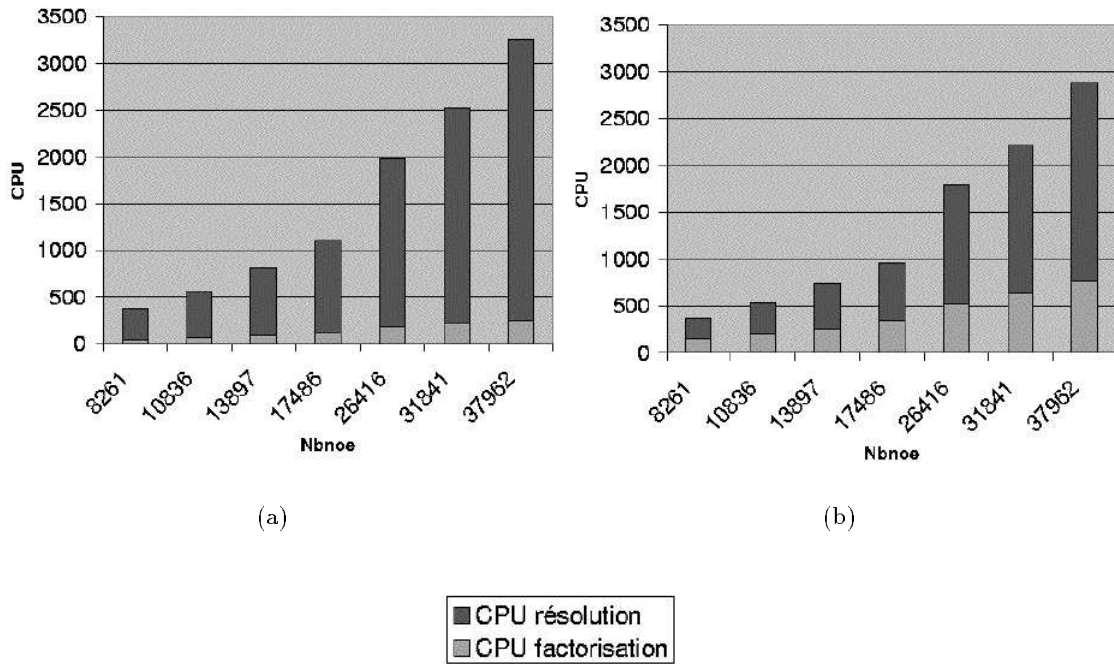


FIG. 3.5 – Comparaison du temps de factorisation et temps de résolution pour la factorisation IC en (a) et la factorisation ICvp en (b)

pour la factorisation ICvp.

L'utilisation de méthodes de factorisation incomplète diminue le temps de calcul mais surtout le nombre d'itérations de la méthode de résidu minimal. On présente dans le tableau 3.4, le nombre d'itérations nécessaires sur un même calcul pour une méthode résidu minimal préconditionné par \mathcal{M}_{ivp} et par des décompositions incomplètes de type IC et ICvp. Ces résultats montrent que les itérations de la méthode ICvp sont plus efficaces (meilleure

	Diagonal bloc		IC		ICvp	
	Itérations	CPU	Itérations	CPU	Itérations	CPU
Incrément1	401	95	122	76	57	48
Incrément2	149	35	47	29	25	21
Incrément3	359	85	111	69	57	48
Incrément4	390	92	121	76	67	56

TAB. 3.4 – Comparaison du nombre d'itérations nécessaires à la résolution d'un même système 3 préconditionneurs différents

réduction du résidu) que les itérations IC.

On ne présente pas de résultats avec une décomposition incomplète modifiée. Contrairement à ce qui est indiqué dans [Ramage et al.94], les résultats que nous avons obtenus avec cette méthode sont décevants. On observe de très long temps de factorisation et de résolution. Sur le tableau 3.5, on présente le nombre d'itérations et le temps CPU pour la résolution de plusieurs résolutions de calcul, sur un cas de poinçonnement de cylindre. Dans [Vasconcelos et al.98], les auteurs font la même constatation pour ce qui concerne le

	MICv		IC	
	Nombre d'itérations	CPU	Nombre d'itérations	CPU
résolution 1	401	317	141	117
résolution 2	1178	930	290	235
résolution 3	826	651	214	173

TAB. 3.5 – *Comparaison des résultats obtenus avec MIC et IC*

préconditionneur de type MIC.

3.1.3.4 Conclusion

- La résolution demande moins d'itérations dans le cas des factorisations incomplètes, à peu près 6 fois moins pour une factorisation ICvp et 3 fois moins pour une factorisation IC.
- L'existence de la factorisation incomplète n'est pas garantie pour nos matrices. Cependant on n'a pas rencontré de cas où on ne pouvait pas la calculer.
- Les préconditionneurs par factorisation incomplète présentent un taux de convergence asymptotique meilleur que celui des préconditionneurs diagonaux.
- Un gros inconvénient des méthodes de décomposition incomplète est qu'elles ne se prêtent pas directement au parallélisme. Toutefois, les résultats récemment obtenus en 2D [Perchat et al.99] permettent d'estimer que les versions parallèles de ces préconditionneurs auront des performances très proches des versions séquentielles étudiées ici.

3.2 Les méthodes multigrilles : présentation

Les méthodes multigrilles sont des méthodes de résolution relativement récentes puisqu'elles sont apparues dans les années 1960-70 [Fedorenko62],[Brandt77]. Elles ont été développées dans le cadre de la résolution d'équations de Poisson. Elles ont été ensuite largement utilisées dans le domaine de la mécanique des fluides. Leur application aux discrétisations de type volumes finis ou différences finies est, en effet, assez facile. Leur utilisation a été ensuite étendue à d'autres problèmes, résolution des équations d'Euler [Mavriplis92], [Van der maarel92], de Stokes [Vincent97] de Navier-Stokes [Perng et al.91], [Oosterlee97], [Dick et al.92], de problèmes de turbulence [Sheng et al.95] [Carre95]. Elles sont assez peu présentes dans le domaine de la mécanique des solides car leur utilisation avec des éléments finis est plus délicate. Citons pourtant des travaux sur l'élasticité [Le dain91] [Lee98], [Mitchell et al.98], mais aussi sur des calculs de viscoplasticité [El hadj87]

Le principal avantage de ces méthodes est leur taux de convergence asymptotique en $O(N \log N)$, ou $O(N)$ sous certaines conditions. Cette propriété en fait des méthodes itératives plus performantes que les méthodes classiques (méthodes de Krylov, gradient conjugué, Choleski, ...) dont le taux de convergence est en $O(N^\alpha)$ avec $1 < \alpha \leq 2$.

L'originalité de ce type de méthode de résolution est l'utilisation de plusieurs maillages de taille décroissante. Le plus fin est le maillage de calcul classique et les plus grossiers sont des maillages destinés à des calcul de correction de la solution. Une variante "algébrique" consiste à construire plusieurs systèmes de taille décroissante.

Dans ce chapitre nous exposerons plus en détail le principe de ces méthodes avant de présenter nos choix de développement pour la simulation 3D du forgeage. Après l'analyse des premiers résultats, nous verrons les améliorations apportées à cet algorithme pour le rendre efficace et robuste dans le cadre du forgeage. Dans ce contexte, il a fallu adapter la méthode au type de système utilisé. En particulier, il faut prendre en compte la forme de la matrice due à la stabilisation par le mini-élément (1.16). Il faut ensuite considérer les problèmes induits par l'utilisation d'une rhéologie non linéaire et par l'algorithme de contact par pénalisation. Il a fallu aussi considérer la déformation du maillage au cours du calcul qui soulève des problèmes de compatibilité entre les niveaux de grilles. Enfin, nous présenterons les résultats obtenus sur différents cas de forgeage.

3.2.1 Principe

Pour exposer le principe de la méthode, nous nous limiterons, dans un premier temps, à une méthode bigrille (le maillage de calcul + un maillage grossier). Nous généraliserons ensuite aux méthodes multigrilles. Nous détaillerons enfin, les éléments qui caractérisent les différentes étapes de telles méthodes.

3.2.1.1 La méthode bigrille

On considère deux maillages de taille de maille croissante Ω_H et Ω_h sur un même domaine.

La phase de lissage

Soit un problème discret sur Ω_h ,

$$\mathcal{A}_h X_h = b_h \tag{3.2}$$

On introduit X_h^ν la solution approchée du problème après ν itérations d'une méthode de résolution itérative avec pour initialisation X_h^0 . On notera :

$$X_h^\nu = \mathcal{I}(X_h^0, \nu)$$

Si on note \bar{X}_h la solution exacte du système, on définit la correction par

$$\delta X_h^\nu = \bar{X}_h - X_h^\nu$$

qui vérifie :

$$\mathcal{A}_h \delta X_h^\nu = b_h - \mathcal{A}_h X_h^\nu = \mathcal{R}_h^\nu \quad (3.3)$$

On veut résoudre l'équation (3.3), mais cela est aussi coûteux que le problème de départ. On va donc transférer ce système sur un maillage plus grossier.

La correction calculée sur la grille grossière est efficace si le problème projeté sur le maillage grossier est proche du problème sur le maillage fin. Sur la figure 3.6, on montre en (a) une approximation δX_H proche de δX_h , et en (b) une mauvaise approximation de δX_h . Le premier cas correspond à une méthode itérative qui présente de bonnes propriétés de lissage de l'erreur contrairement à (b) où le lissage est insatisfaisant.

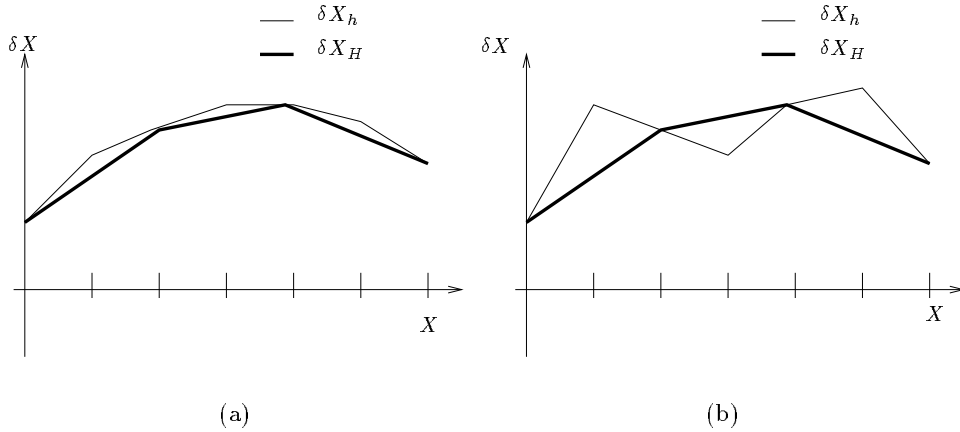


FIG. 3.6 – (a) Erreur lissée sur la grille fine, (b) erreur mal lissée sur la grille fine (Représentation 1D)

La phase de correction

Pour calculer δX_H^ν , correction sur la grille grossière, il faut transférer le problème (3.3) sur cette grille grossière. Pour cela on introduit un opérateur de restriction p_h^H et on résout le problème suivant sur le maillage Ω_H :

$$A_H \delta X_H^\nu = p_h^H \mathcal{R}_h^\nu$$

où A_H est la matrice du problème sur la grille grossière. Une fois cette correction grossière calculée, il faut corriger le vecteur X_h^ν . Pour cela il faut transférer la valeur de la correction δX_H^ν sur la grille fine Ω_h à l'aide de l'opérateur de prolongement p_H^h . On ajoute à X_h^ν cette correction pour obtenir une nouvelle approximation de \bar{X}_h notée $X_{h,1}^\nu$

$$X_{h,1}^\nu = X_h^\nu + p_H^h \delta X_H^\nu$$

Si $X_{h,1}^\nu$ ne vérifie pas l'équation (3.2) à la précision voulue, on itère le processus en posant :

$$X_{h,2}^0 = X_{h,1}^\nu$$

On vient de décrire ci-dessus *un cycle bigrille* n qui se décompose en plusieurs étapes :

1. Lissage : ν itérations d'une méthode itérative

$$X_{h,n}^\nu = \mathcal{I}(X_{h,n}^0, \nu)$$

2. Restriction du résidu au maillage grossier:

$$\mathcal{R}_{H,n}^\nu = p_h^H \mathcal{R}_{h,n}^\nu$$

3. Correction par grille grossière : *résolution du système sur le maillage grossier*

$$\mathcal{A}_H \delta X_{H,n}^\nu = \mathcal{R}_{H,n}^\nu \tag{3.4}$$

4. Prolongement de la correction sur le maillage fin :

$$\delta X_{h,n}^\nu = p_H^h \delta X_{H,n}^\nu$$

5. Calcul de la nouvelle approximation :

$$X_{h,n+1}^0 = X_{h,n}^\nu + \delta X_{h,n}^\nu$$

Remarques :

- La résolution multigrille est un processus itératif (enchaînement de cycles) qui utilise une méthode itérative de résolution de système linéaire (\mathcal{I}), le lisseur.
- On utilise en général des cycles avec une sixième étape, appelée post-lissage, où :

$$X_{h,n+1}^0 = \mathcal{I}(X_{h,n}^\nu + \delta X_{h,n}^\nu, \nu')$$

L'étape 1 est alors, par opposition, appelée pré-lissage.

- Il faut définir plusieurs paramètres de la méthode comme :
 - la méthode itérative de lissage,
 - le nombre d'itérations de pré et post-lissage,
 - les opérateurs de transfert p_H^h et p_h^H ,
 - la méthode de calcul de \mathcal{A}_H ,
 - la méthode de résolution de la correction (comment résoudre $\mathcal{A}_H \delta X_{H,n}^\nu = \mathcal{R}_{H,n}^\nu$),

3.2.1.2 La méthode multigrille

Pour passer d'une méthode bigrille à une méthode multigrille, il suffit de renouveler le raisonnement pour la résolution du problème (3.4) avec une grille encore plus grossière. Il faut alors définir des opérateurs supplémentaires entre les grilles grossières. On peut ainsi, de proche en proche, introduire plusieurs niveaux de grilles.

Lorsqu'on utilise plusieurs niveaux de grilles se pose le problème de savoir comment enchaîner les résolutions grille fine, grille(s) moyenne(s) et grille grossière. Il existe dans la

littérature plusieurs cycles (V-cycles, W-cycles ...). On en présentera quelques uns dans le chapitre 3.4.3.

On détaillera dans la suite quelques choix possibles pour les paramètres de la méthode multigrille. Ceci ne représentera pas une liste exhaustive au vue de l'étendue des applications des méthodes multigrilles [Douglas et al.]. Toutefois, on donnera un aperçu des possibilités qui se sont présentées à nous pour la mise en place d'un solveur multigrille.

3.2.2 Construction du problème grossier

Il nous faut introduire un problème grossier qui soit une approximation du problème sur le maillage fin. Pour cela, deux grandes voies existent. Soit on construit effectivement les différents maillages sur lesquels on calcule la matrice du problème, *c'est l'approche géométrique*. Soit, on n'introduit pas de maillage supplémentaire mais on détermine le système linéaire grossier à partir du système linéaire fin par le biais d'opérateurs, *c'est l'approche algébrique*.

3.2.2.1 L'approche géométrique

On construit différents maillages, le fin et les grossiers. Sur chacun de ces maillages, on calcule la matrice correspondant au problème.

Plusieurs méthodes sont utilisées dans la littérature pour construire les différents maillages. Le lecteur intéressé pourra consulter le chapitre 4.3 de [Mavriplis97] pour y trouver d'autres références et d'autres méthodes.

Les grilles indépendantes

Les différentes grilles sont construites indépendamment, avec pour seul critère une taille de maille décroissante [Mavriplis et al.90], [Mavriplis95], [Zang et al.95]. Sur de tels maillages, il faut ensuite construire des opérateurs de restriction et de prolongement, ce qui nécessite une procédure d'interpolation des nœuds d'un maillage dans le maillage qui est plus grossier et dans celui qui est plus fin (voir figure 3.7).

Le seul impératif que doivent remplir ces maillages est de décrire la même frontière. Le principal inconvénient de ces méthodes est qu'il faut construire et stocker les différents opérateurs de restriction et de prolongement entre les niveaux de maillage, et que le calcul des communications entre grilles par ces opérateurs est coûteux en temps. De plus dans le cas particulier de la simulation du forgeage, les remaillages sont nombreux et coûteux en temps CPU. Avec une telle méthode, il faudrait remailler tous les niveaux de maillages, ce qui alourdirait encore la procédure.

Déraffinement de maillage fin

On part d'un maillage fin et on ne conserve que certains nœuds de ce maillage. On peut, par exemple, extraire ces nœuds en prenant un nœud sur deux [Guillard93] ou en utilisant la théorie des graphes [Dorward et al.97]. Les grilles sont alors emboîtées par nœud. On obtient un nuage de points grossiers que l'on triangule par un algorithme de type Voronoï. Une autre méthode consiste à éliminer certaines arêtes du maillage, à améliorer la topologie par inversion d'arêtes puis à lisser la distribution des nœuds [Hattangady99].

L'avantage est que l'on part du maillage de calcul fin pour construire le maillage grossier. L'inconvénient est que l'on impose aux maillages de respecter approximativement la même frontière, ce qui n'est pas toujours aisé dans le cas de géométries complexes. Il faut alors

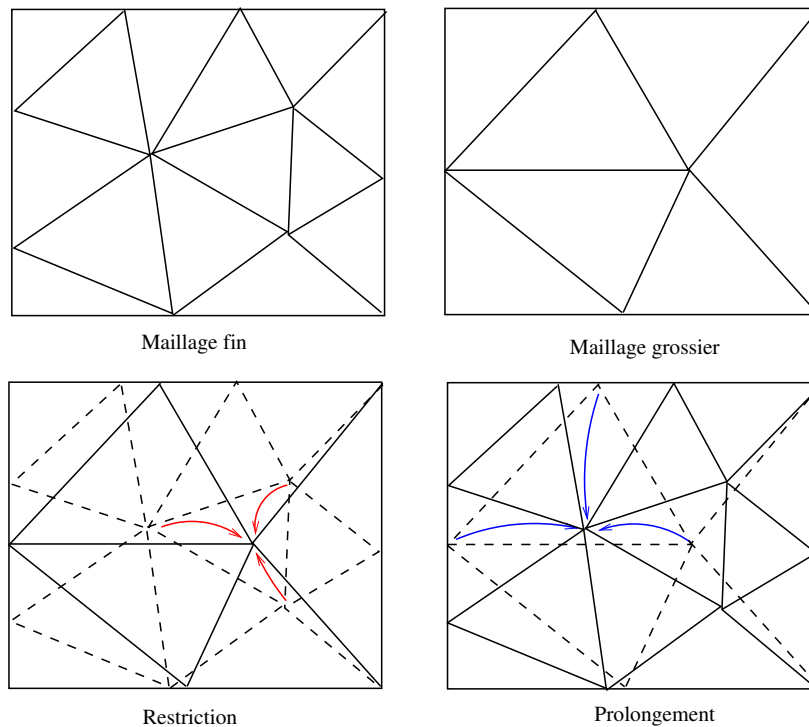


FIG. 3.7 – Opérateurs d’interpolation de deux grilles indépendantes

introduire “manuellement” des nœuds à conserver lors du déraffinement.

Cependant ce type de méthode n’existe que pour des maillages 2D. En 3D, le problème est bien plus compliqué (triangulation des nœuds) et il n’existe pas, à notre connaissance, de déraffinement de maillage tridimensionnel.

Le raffinement de grille grossière

On part ici d’une grille grossière que l’on va raffiner, soit de manière systématique comme dans [El hadj87], [Le dain91], [Muttin et al.95], soit en s’aidant d’un estimateur d’erreur a posteriori [Moukalled et al.91]. C’est une procédure simple, mais en 3D, le raffinement d’un tétraèdre peut mener à des tétraèdres dégénérés [Muttin et al.95] et donc à un maillage éléments finis de mauvaise qualité. De plus, il faut que le maillage ainsi obtenu décrive le contour fin voulu. Des dégénérescences supplémentaires peuvent apparaître lors de la projection des nœuds fins sur la frontière (voir figure 3.8).

3.2.2.2 L’approche algébrique

Les multigrilles algébriques (AMG) sont apparues au début des années 80 [Brandt et al.84]. Elles utilisent les informations contenues dans la matrice du problème pour construire les différents éléments de la méthode (niveaux de résolution, opérateurs de transfert...).

Le principe

Elles sont basées sur l’utilisation d’un seul maillage. On détermine les opérateurs et les matrices grossières et donc les degrés de liberté grossiers à partir de la valeur des termes de la matrice fine. Cette méthode a pour principaux avantages de ne pas obliger à construire différents maillages et possède un caractère automatique quant à la construction des problèmes

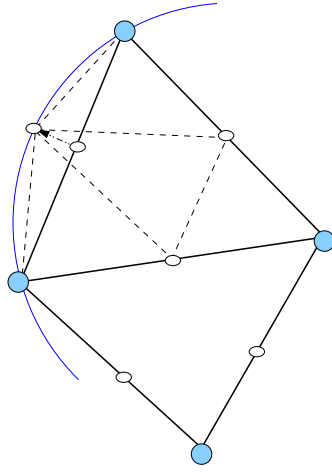


FIG. 3.8 – *Projection de nœuds fins en frontière pour une méthode de raffinement de maillage*

grossiers. Elle est utilisée pour la résolution de problèmes elliptiques [Mandel et al.88], en mécanique des fluides ou des solides [Boersma et al.97]. Cependant elle est généralement moins rapide que les méthodes géométriques d'après [Lallemand92].

Construction de la matrice grossière

Les opérateurs p_h^H et p_H^h sont calculés à partir des valeurs des coefficients de A_h . On note N_i l'ensemble des voisins du degré de liberté i , défini par :

$$\forall i \in \{1, \dots, Nbddl\}, N_i = \{j \in \{1, \dots, Nbddl\} / a_{ij} \neq 0\}$$

où $Nbddl$ est le nombre de degrés de liberté et S_i l'ensemble des points auxquels le degré de liberté i est fortement connecté, c'est à dire :

$$\forall i \in N_i, S_i = \left\{ j \in \{1, \dots, Nbddl\} / |a_{ij}| \geq \varepsilon_1 \max_{k \neq i} |a_{ik}| \text{ avec } \varepsilon_1 \in]0, 1] \right\}$$

On construit l'espace C_i des nœuds grossiers par une procédure itérative décrite par exemple dans [Lallemand92] et utilisant les espaces N_i et S_i . On définit ainsi l'opérateur p_H^h par :

$$(u_h)_i = (p_H^h u_H)_i = \begin{cases} (u_H)_i & \text{si } i \in C_i \\ \sum_{j \in C_i} \frac{a_{ij}}{a_{ii}} (u_H)_j & \text{sinon} \end{cases}$$

L'opérateur p_h^H est défini par $p_h^H = p_H^h{}^t$ et la matrice du problème grossier est déterminée par la relation : $A_H = p_h^H A_h p_H^h$ que l'on nomme *Condition de Galerkin*.

3.2.2.3 Autres méthodes de construction

Les méthodes d'agglomération

Dans [Koobus94] on combine les maillages éléments finis et volumes finis pour la résolution de problèmes hyperboliques. A partir du maillage éléments finis, on construit un maillage

volumes finis à l'aide des volumes de contrôle. Ces volumes sont ensuite agglomérés de façon successive pour construire les maillages volumes finis grossiers. C'est une méthode qui se rapproche des méthodes de déraffinement de maillage.

Les méthodes hiérarchiques

Les différents niveaux sont construits à partir d'une grille grossière en enrichissant successivement les fonctions de base éléments finis [Pressburger et al.95]. Ces méthodes sont difficiles à adapter à notre cas où les fonctions de base pour la vitesse et la pression doivent vérifier des conditions de compatibilité.

3.2.3 Le lissage, la correction et les opérateurs

3.2.3.1 Le lissage

Les méthodes "classiques"

Les méthodes de lissage que l'on trouve dans la littérature sont souvent des méthodes de type Jacobi [Mavriplis et al.90], SOR [Pressburger et al.95] ou Gauss-Seidel, [Perng et al.91], [Elman96]. Elles possèdent, en effet, de bonnes propriétés d'élimination des composantes hautes fréquences au cours des premières itérations (c'est-à-dire de bonnes propriétés de lissage de l'erreur).

On trouve aussi des méthodes de décomposition incomplète ILU étudiées par [Wittum89] et utilisées dans [Elman96] et [Tzanos92]. Si $A = LU - \Delta$ est la décomposition incomplète de la matrice A du problème $AX = b$, on calcule :

$$X^{k+1} = (LU)^{-1} \Delta X^k + (LU)^{-1} b$$

Les méthodes de type gradient conjugué

On trouve peu d'auteurs utilisant des méthodes de gradient conjugué comme lisseur. En effet ce type de méthode réduit simultanément les composantes hautes et basses fréquences de l'erreur [Braess86]. L'utilisation de tels solveurs en tant que lisseur est donc rare [Farhat et al.89], [Bornemann et al.96]. Toutefois, sur des problèmes d'élasticité et de visco-plasticité, son utilisation a donné de bons résultats [El hadj87], [Le dain91]. [Oosterlee97] utilise une méthode GMRES comme lisseur pour des problèmes de mécanique des fluides.

De manière générale, l'association "*multigrille/gradient conjugué*" se fait surtout par l'utilisation des méthodes MG comme préconditionneur des méthodes de gradient conjugué [Silvester et al.94], [Boersma et al.97], [Davids et al.99]. Par exemple dans l'algorithme 3.1, le calcul de $\mathcal{M}^{-1}X$ est assimilé à un calcul approché de $\mathcal{A}y = X$ en faisant quelques cycles d'une méthode multigrille.

3.2.3.2 La correction

Pour résoudre le problème sur la grille la plus grossière, la méthode la plus souvent utilisée est la méthode directe. En effet, on utilise, en général, un nombre élevé de maillages qui sont tels que la grille fine est très fine et la grille grossière est très grossière. Dans [Morano92], l'auteur utilise 6 maillages dont le plus fin possède 12284 nœuds et le plus grossier 19. Une résolution par la méthode directe d'un tel système est donc peu coûteuse.

3.2.3.3 Les opérateurs

Le choix des opérateurs de transfert entre les grilles dépend fortement de la méthode de construction des différents maillages.

- Pour les méthodes algébriques, les opérateurs sont construits à partir des termes de la matrice fine et vérifient $p_h^H = p_H^h$.
- Pour les méthodes hiérarchiques, les opérateurs peuvent être simplement des injections d'une grille sur l'autre [Pressburger et al.95].
- Enfin pour les multigrilles géométriques, les opérateurs sont des interpolations qui peuvent être constantes, linéaires ou quadratiques. Dans [Mavriplis95], l'auteur choisit des opérateurs d'interpolation linéaire entre les différents maillages construits indépendamment. Dans [Oosterlee97], l'opérateur de prolongement est une interpolation linéaire alors que l'opérateur de restriction est un opérateur par moyenne pondérée. Dans [Rubart92], [Le dain91], l'opérateur de prolongement est défini par les fonctions de base éléments finis utilisées et l'opérateur de restriction est défini par la relation :

$$p_h^H = p_H^h$$

3.3 les méthodes multigrilles et les problèmes incompressibles

On présente ici rapidement les différentes méthodes utilisées dans la littérature pour traiter les problèmes incompressibles dans le cadre particulier des méthodes multigrilles.

3.3.1 Le problème de Stokes

Pour des raisons de simplicité, on assimilera notre problème au problème de Stokes pour la présentation des différentes méthodes. Le problème de Stokes est un système d'équation en vitesse et pression qui s'écrit de la manière suivante :

$$\left| \begin{array}{l} \text{Trouver } (\mathbf{v}, p) \\ \left\{ \begin{array}{ll} -\nu \Delta \mathbf{v} + \mathbf{Grad} p = f & \text{sur } \Omega \\ \operatorname{div} \mathbf{v} = 0 & \text{sur } \Omega \\ \mathbf{v} = \mathbf{0} & \text{sur } \partial\Omega \end{array} \right. \end{array} \right.$$

qui donne, après discrétisation le système suivant à résoudre :

$$\begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} \begin{pmatrix} V \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}$$

où A est une matrice symétrique définie positive.

La matrice globale du système précédent est mal conditionnée car elle comporte des zéros sur la diagonale. On ne peut donc pas inverser ce système par des méthodes itératives classiques.

3.3.2 Les méthodes découplées

Une méthode classique pour résoudre les systèmes mixtes issus de problèmes incompressibles est de découpler les équations de la vitesse et de la pression. L'algorithme d'Uzawa (voir l'algorithme 3.2) est, par exemple très utilisé.

Algorithme 3.2 Algorithme d'Uzawa pour la résolution découplée en vitesse/pression

1-Initialisations

choix de p_0

2- A l'itération n

$$(A + rB^tB)u_n = F - B^tp_n$$

3- Calcul de p_{n+1}

$$p_{n+1} = p_n - rBu_n$$

4- Répéter 2 et 3 jusqu'à la convergence

fin algorithme

3.3.2.1 Les multigrilles comme méthode de résolution

Les auteurs utilisent souvent la méthode de découplage d'Uzawa pour la résolution des problèmes incompressible avec des méthodes multigrilles. Dans [El hadj87] et [Rubart92], ils utilisent une méthode multigrille pour inverser le système en vitesse uniquement. Dans [Lonsdale93], l'auteur, avec une autre méthode de découplage, *la méthode SIMPLE*, utilise une méthode multigrille pour résoudre problème en vitesse.

Notons que l'on a traité ici de l'expression du problème en vitesse et pression, mais que l'on trouve aussi dans la littérature l'expression du problème de Stokes en vorticit  et fonction de courant. On peut, dans ce cas aussi découpler les  quations et utiliser la m thode multigrille pour la r solution du syst me en vorticit  [Davis et al.93].

3.3.2.2 La m thode d coupl e comme lisseur

Les m thodes multigrilles peuvent utiliser comme lisseur une r solution partielle (quelques it rations) par la m thode d'Uzawa [Vincent97], [Abdalass87]. Ces travaux ont  t  d velopp  pour la r solution d'un probl me de Stokes stabilis  d fini par :

$$\begin{pmatrix} A & B^t \\ B & -\beta C \end{pmatrix} \begin{pmatrix} V \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}$$

o  β est un param tre et C est une matrice de stabilisation d finie par :

$$C_{ij} = - \sum_{\omega_h \in \Gamma_h} \delta\omega_h \int_{\omega_h} N_i N_j d\omega$$

o  $(N_k)_k$ est la famille des fonctions de base discr tisant le champ de pression.

Quelques it rations d'une m thode it rative (Gauss-Seidel) sur le probl me en vitesse et une correction en pression constituent l' tape de lissage. Dans ces publications, on trouve aussi d'autres lisseurs "d coupl s" comme un lisseur G.S.B.A. (Gauss Seidel Bloc Altern s). Cette m thode utilise la matrice de masse M . Le lisseur est d fini par :

1. A partir de V^0 , on applique deux it rations de Gauss-Seidel au syst me

$$AV = F - B^t P^0$$

2. On r sout le syst me triangulaire inf rieur

$$(M + D - E)\bar{P} = (M + E^t)P^0 + BV$$

o  $C = D - E - E^t$ et E est une matrice triangulaire sup rieure.

3. On r sout le syst me triangulaire sup rieur

$$(M + D - E^t)P = (M + E^t)\bar{P} + BV$$

3.4 Mise en place d'une méthode multigrille 3D : Application à la simulation du forgeage à chaud

On veut mettre en place une méthode de résolution de systèmes linéaires à l'intérieur de l'algorithme de Newton-Raphson.

On expose dans ce paragraphe les choix faits pour la mise en place d'un solveur multigrille. Pour cela, on justifie d'abord de la méthode de calcul des différents niveaux (maillages et matrices sur chacun de ces maillages), on abordera ensuite la question des méthodes de résolution à chacun des niveaux de maillage ainsi que des opérateurs de transfert entre les grilles. Dans une troisième partie, on présente les adaptations faites pour améliorer les résultats de la première version.

3.4.1 Les différents niveaux de calcul

3.4.1.1 Création des différents niveaux de maillage

On a choisi d'utiliser une méthode de raffinement de maillage à partir d'une grille grossière (voir le paragraphe précédent). Dans notre cas 3D, on part de la grille grossière tétraédrique et on choisit de diviser chaque tétraèdre en 8 sous tétraèdres (voir figure 3.9).

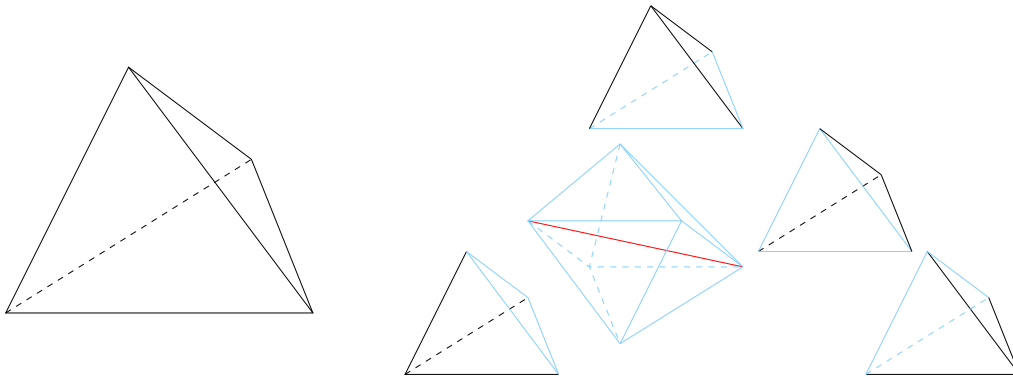


FIG. 3.9 – *Subdivision d'un tétraèdre pour le raffinement de maillage*

On peut noter que les 4 sous-tétraèdres internes ne sont pas homothétiques au tétraèdre de départ. Comme il est noté dans [Muttin et al.95], une telle méthode peut conduire à des éléments de faible qualité. Toutefois dans [Gawrylczyk95], l'auteur indique que cette méthode de décomposition est la façon la plus efficace pour raffiner un maillage tétraédrique.

Propriétés de la méthode de raffinement

- Après la résolution, la partie du calcul la plus coûteuse en modélisation du forgeage est la procédure de remaillage. Si notre maillage de départ est le maillage grossier, c'est celui-ci qu'il faudra remailler. Comme ce maillage est plus petit (en nombre de nœuds), le remaillage devrait être plus rapide.
- Dans l'optique d'une implantation en parallèle cette méthode semble bien adaptée. En effet, pour partitionner le problème, on procède par découpage du maillage. Ensuite la matrice associée à chaque sous maillage est assemblée. Avec cette méthode, on

partitionne le maillage grossier et on le raffine. Cette procédure n'induit aucune communication supplémentaire entre les processeurs.

- Une série de raffinements successifs mène à la dégénérescence du maillage fin. Avec cette méthode, on ne pourra donc pas utiliser, pour un maillage grossier de qualité moyenne, plus de 3 niveaux de maillages.

3.4.1.2 Construction des matrices

Le problème grossier

Sur le maillage grossier, il existe plusieurs méthodes pour construire la matrice. Nous avons testé différentes méthodes.

- Assemblage

La matrice grossière est la matrice du problème sur la grille grossière que l'on note \tilde{A}_H . On introduit une discrétisation éléments finis sur le maillage grossier et on assemble la matrice correspondante. Cette méthode a pour inconvénient de dupliquer les structures de données de la procédure d'assemblage.

- La condition de Galerkin

Cette condition introduite dans le paragraphe 3.2.2.2 est utilisée, en particulier, dans les méthodes de multigrilles algébriques. On calcule la matrice \bar{A}_H définie par

$$\bar{A}_H = p_h^H A_h p_H^h$$

La condition $\bar{A}_H = \tilde{A}_H$ est vérifiée pour deux maillages différences finies emboîtés. Elle n'est pas vérifiée dans le cas de maillages éléments finis non emboîtés.

On choisit de calculer la matrice grossière par la condition de Galerkin même si la condition $\tilde{A}_H = \bar{A}_H$ n'est pas vérifiée. La différence avec les méthodes AMG vient du fait que nos opérateurs de transfert ne sont pas calculés en fonction des termes de la matrice mais en fonction de relations géométriques entre les nœuds fins et grossiers.

Les opérateurs de transfert

Le choix des opérateurs est basé sur la définition des fonctions de base. Ainsi, l'opérateur de restriction est défini de la manière suivante :

$$\forall K \in \Omega_H, \forall l \in \omega_h \quad p_h^H K_l = N_H^k(X^l) \quad (3.5)$$

où (N_H) est l'ensemble des fonctions l'interpolation linéaire sur Ω_H . Pour conserver les propriétés de la matrice (symétrique avec un bloc diagonal défini positif et un bloc diagonal défini négatif), d'une part et pour pouvoir utiliser la condition de Galerkin d'autre part, on définit l'opérateur de prolongement par :

$$p_H^h = p_h^{H^t} \quad (3.6)$$

Ces opérateurs sont définis en début de calcul et ne sont pas actualisés avant le remaillage suivant. Ainsi ils sont constants entre des remaillages.

3.4.1.3 Validité de la construction de la matrice grossière

On justifie ici du choix du calcul de la matrice grossière par la condition de Galerkin.

La différence de frontière entre Ω_H et Ω_h

Le maillage fin est obtenu en projetant sur la surface du domaine les nœuds appartenant aux milieux des éléments grossiers (voir la figure 3.10). Il y a ainsi une différence de volume entre le maillage fin et le maillage grossier. Si on considère la condition d'incompressibilité

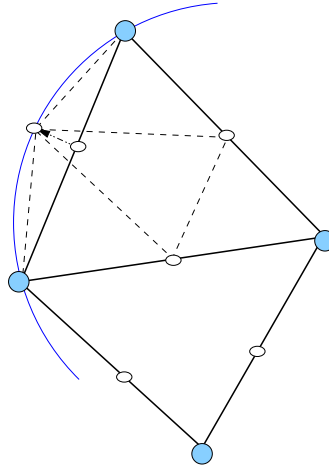


FIG. 3.10 – Différence de volume entre le maillage fin et le maillage grossier

calculée sur la grille grossière par assemblage, on obtient :

$$\forall K, \sum_{\Omega_E \in \Gamma_H} V_i^L \int_{\Omega_E} \sum_{i,L} N^K \frac{dN^L}{dx_i} d\omega = 0$$

où K et L sont des nœuds du maillage grossier. On intègre donc sur le volume du maillage grossier.

En revanche, en utilisant la condition de Galerkin, on obtient la condition suivante :

$$\forall K, \sum_k \alpha_{k,K} \sum_{\Omega_e \in \Gamma_h} \int_{\Omega_e} N^k \frac{dN^l}{dx_i} d\omega \sum_L \alpha_{l,L} V_i^L = 0$$

où les $\alpha_{k,K}$ sont les coefficients des opérateurs de transfert entre les grilles :

$$V_l = \sum_L \alpha_{l,L} V^L$$

La condition est donc intégrée sur la grille fine, ce qui permet d'assurer la conservation du volume du domaine complet.

La stabilisation de la bulle

On rappelle que, pour la simulation du forgeage, la matrice \mathcal{A}_h a une forme particulière puisqu'elle est stabilisée par élimination du mini-élément (1.16). Si on lui applique la condition de Galerkin, on n'obtient pas la même matrice grossière que si l'on appliquait la condition de Galerkin à la matrice du système (1.15) et que l'on procédait ensuite à l'élimination des degrés de liberté bulle.

La seconde solution ne semble pas raisonnable car les opérateurs de transfert sont construits à partir des fonctions de base linéaires. Pour que la condition soit correcte, il faudrait définir les opérateurs à partir des fonctions de bases linéaires et bulle. De plus il faudrait

calculer la matrice du système (1.15), ce qui n'est pas fait explicitement.

L'imposition des degrés de liberté

Pour les degrés de liberté imposés (nœuds de symétrie), le passage de la matrice fine à la matrice grossière par la condition de Galerkin peut perturber l'imposition des degrés de liberté. Pour un nœud N appartenant à la fois au maillage fin et au maillage grossier et appartenant par exemple au plan de symétrie d'équation $x = 0$, on impose la condition $V_x^N = 0$ par élimination en modifiant la matrice fine de la manière suivante :

$$a_{h_{1N},jl} = a_{h_{jl},1N} = 0, \forall (j,l) \neq (1,N) \text{ et } a_{h_{1N},1N} = 1$$

pour le second membre, on impose $R_{1N} = 0$.

Lorsqu'on applique la condition de Galerkin, la matrice grossière ne vérifie plus $a_{H_{1N},jl} = 0$ pour les nœuds l voisins de N sur le maillage grossier car les nœuds voisins de N qui ne sont pas de symétrie ont également apporté leur contribution (non nulle) (voir la figure 3.11).

On a donc comparé les résultats obtenus avec cette matrice et une matrice où on a

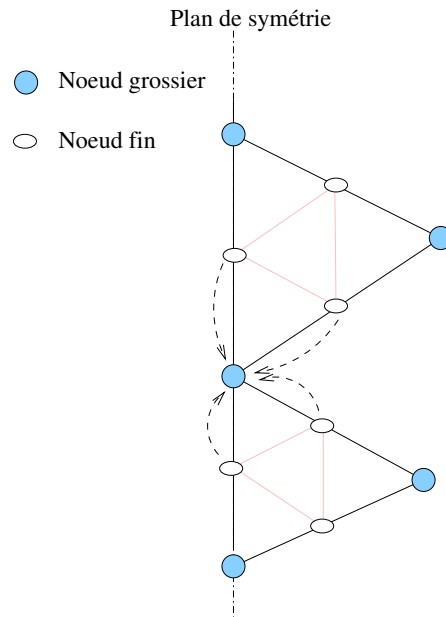


FIG. 3.11 – Influence des nœuds internes sur l'imposition des degrés de liberté de symétrie sur la grille grossière

pris soin de réimposer la condition $V_x^N = 0$ sur la grille grossière, en appliquant le même traitement qu'à \mathcal{A}_h . Les résultats obtenus sont équivalents, sauf sur les plans de symétrie où la condition $V_x = 0$ est respectée à une certaine précision avec la première méthode alors qu'avec la seconde elle est vérifiée exactement.

Sur un cas de forgeage où la vitesse de référence est de $1ms^{-1}$, la condition est vérifiée à une précision de $10^{-7}ms^{-1}$. La méthode de Galerkin est donc tout à fait satisfaisante.

Les repères locaux

Pour des raisons de simplicité, les degrés de liberté sont imposés dans un repère local au plan de symétrie (voir figure 3.12). Si on veut utiliser la condition de Galerkin, il faut

ramener l'expression de tous les degrés de liberté dans le repère global, comme nous allons le voir. Considérons la figure 3.13. Si on fait le calcul sans prendre le soin d'exprimer toutes

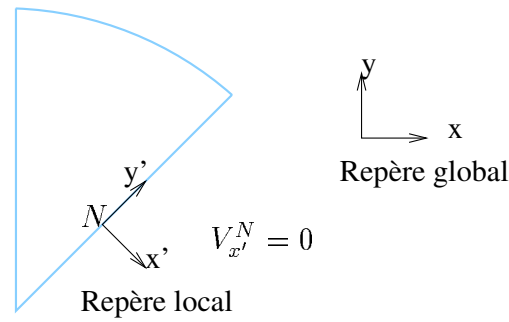


FIG. 3.12 – Repères local et global. Le premier sert à l'imposition des conditions de symétrie

les équations dans le même repère, alors en calculant la restriction du résidu sur la grille grossière on obtient :

$$R_{xH}^N = \underbrace{R_{xh}^N + \frac{1}{2}R_{xh}^i + \frac{1}{2}R_{xh}^j}_{\text{repère } (x',y')} + \underbrace{\frac{1}{2}R_{xh}^k + \frac{1}{2}R_{xh}^l}_{\text{repère } (x,y)}$$

car $N, i, \text{ et } j$ appartiennent au plan de symétrie donc R_{xh}^N, R_{xh}^i et R_{xh}^j sont exprimés dans le repère local, contrairement à R_{xh}^k et R_{xh}^l qui restent exprimés dans le repère global.

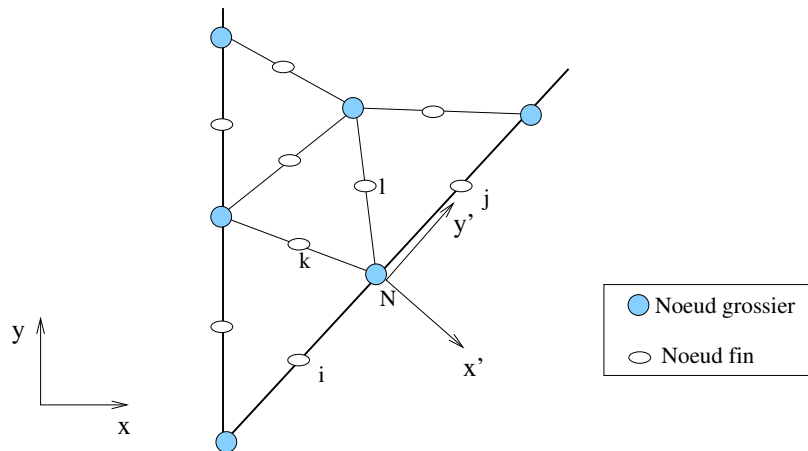


FIG. 3.13 – Repères local et global. Application à la restriction du résidu

Cela conduit à sommer des contributions de natures différentes. Une alternative est de conserver les degrés de liberté dans le repère global. Ainsi la matrice grossière peut être calculée sans ambiguïté. Cette alternative permet d'améliorer nettement la convergence de la résolution multigrille. Toutefois, si on ne l'utilise pas, la méthode converge toujours, mais beaucoup moins rapidement.

3.4.1.4 Lissage et correction

Le lisseur

Nous avons choisi de résoudre le système (1.16) à l'aide d'une méthode multigrille dont le lisseur est une méthode de résidu minimal préconditionné. Cette méthode est originale par plusieurs points.

- Le lisseur est de type gradient conjugué, ce qui est assez peu courant dans la littérature.
- On applique la méthode multigrille au problème complètement couplé.

La correction

Il reste à choisir la méthode pour résoudre le problème sur la grille grossière. Dans notre cas, le nombre de niveaux de grilles est limité à 3 ou 4. La matrice grossière n'est donc pas suffisamment petite pour être résolue par une méthode directe. On fait donc le choix d'utiliser la méthode de résidu minimal préconditionné pour résoudre le système. Ceci est justifié par le fait que la matrice grossière obtenue est symétrique et possède les mêmes propriétés que la matrice fine.

3.4.2 Mise en place de la méthode

3.4.2.1 Les bi-grilles idéales

Dans un premier temps, il s'agit de trouver le nombre optimal d'itérations du lisseur de chaque cycle. On se place donc dans le cadre des bigrilles idéales comme dans [Carre95]. Cela consiste à imposer une résolution exacte sur la grille grossière de manière à n'étudier que l'influence du lissage sur la résolution multigrille. On résout exactement en utilisant la méthode de résidu minimal avec une précision 10^{-8} . On reporte dans le tableau 3.6 les résultats obtenus sur la première résolution d'un cas d'écrasement d'un lopin cubique entre deux tas plats. La rhéologie du lopin est considérée newtonienne et le contact est frottant avec un coefficient de 0.2. ν représente de nombre d'itérations imposées sur la grille fine. Le maillage comporte environ 14000 nœuds fins et 2200 nœuds grossiers. Le temps de résolution par la méthode de résidu minimal est de 94 secondes pour 401 itérations.

On note sur ces résultats que le nombre de cycles et le temps CPU se stabilisent au bout d'une vingtaine d'itérations. En revanche, on observe un minimum du nombre d'itérations cumulées sur la grille fine pour $\nu = 7$. Dans la littérature, le nombre de pas de lissage optimum est souvent de l'ordre de 2 ou 3 pour la méthode de Jacobi et 3 ou 4 pour le gradient conjugué [Le dain91].

3.4.2.2 Nombre d'itérations fixé ou précision fixée

La question se pose de savoir si ce nombre optimal d'itérations dépend du problème considéré. Pour un lisseur de type gradient conjugué, une condition sur la précision est parfois plus adaptée. Dans [Oosterlee97], l'auteur préconise ce type de condition pour optimiser la convergence de la méthode, avec un lisseur de type GMRES. Le même type d'étude est effectué en étudiant diverses précisions de "convergence" du lisseur.

On constate que fixer la précision à 0.2 donne des résultats équivalents à un nombre de pas de lissage fixé à 7 sur ce cas.

ν	2	3	4	5	6	7	8	9
Nombre de Cycles	58	32	17	9	8	6	6	5
Nombre total d'itérations	115	95	65	45	44	38	43	43
CPU	1510	826	440	228	203	149	149	122
	10	11	12	15	20			
	5	5	4	4	4			
	42	45	46	50	63			
	121	122	97	98	101			

TAB. 3.6 – Résultats pour un test de bigrille idéale sur un écrasement de lopin cubique entre tas plats à nombre d'itération de lissage fixée. Influence du nombre ν d'itérations de lissage.

ϵ^f	0.5	0.2	0.15	0.1	0.05	0.01
Nombre de Cycles	12	5	5	4	3	3
Itérations	46	39	41	49	69	154
CPU	300	120	121	97	76	97

TAB. 3.7 – Résultats pour un test de bigrille idéale sur un écrasement de lopin cubique entre tas plats à précision fixée pour le lisseur

On décide de retenir cette condition $\epsilon^f = 0.2$, a priori, mieux adaptée pour prendre en compte tout type de problème.

3.4.2.3 Vérification des propriétés multigrille

On souhaite vérifier que le lisseur que nous avons choisi conduit à une méthode multigrille qui en vérifie les propriétés élémentaires. On vérifie tout d'abord que le processus de convergence est indépendant du maillage. Pour cela, on trace l'évolution du résidu au cours des calculs pour plusieurs maillages. Les résultats sont reportés sur la figure 3.14.

On constate que les différentes courbes obtenues, pour différents maillages, sont superposées, ce qui montre bien l'indépendance de la convergence par rapport au maillage. Cette propriété est nécessaire pour assurer une convergence en $O(N \log N)$.

3.4.2.4 Détermination de la précision de convergence optimale sur la grille grossière

Sur le tableau 3.7, on remarque que le temps de calcul est bien plus élevé qu'avec la méthode de résidu minimal (94 secondes). Cela vient du fait que l'on a imposé une résolution exacte sur la grille grossière, ce qui n'est pas nécessaire. On recherche donc la

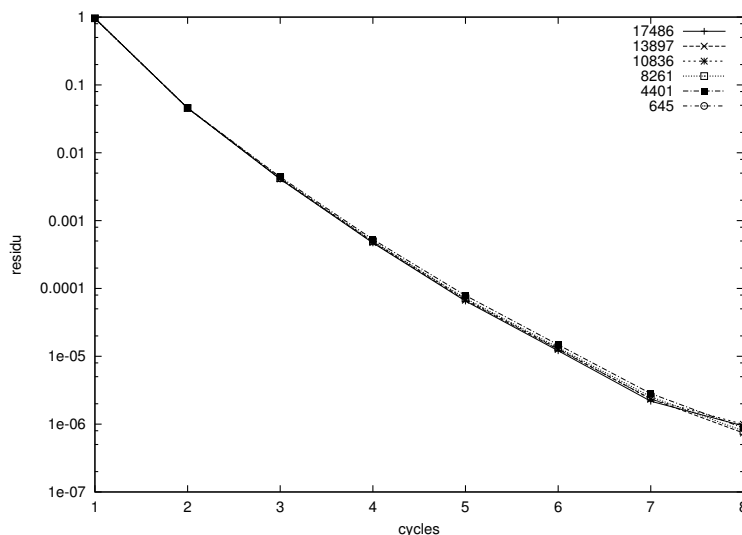


FIG. 3.14 – Valeur du résidu relatif à la fin de chaque cycle multigrille pour plusieurs maillages de nombre de nœuds croissants (Problème d'écrasement d'un cube newtonien entre tas plats frottants)

précision optimale ε^g de la résolution sur la grille grossière. Pour cela, on fixe la précision grille fine à $\varepsilon^f = 0.2$. Les résultats obtenus pour différentes valeurs de ε^g sont reportés dans le tableau 3.8. Sur la deuxième ligne du tableau 3.8, on voit que lorsque la résolution n'est

ε^g	10^{-8}	10^{-5}	10^{-3}	0.05	0.1	0.15	0.2	0.5
Cycles	5	5	5	5	5	6	6	6
Nbit	39	39	39	39	39	59	56	312
CPU	120	97	57	27	25	29	29	87

TAB. 3.8 – Détermination de la précision grille grossière ε^g pour une précision grille fine $\varepsilon^f = 0.2$

pas assez précise (précision > 0.1), le nombre d'itérations sur la grille fine augmente pour compenser la perte de précision. Sur la troisième ligne on observe que lorsque la résolution grille grossière est trop précise, le temps de calcul est inutilement grand.

On remarque que l'on réduit très sensiblement le temps de calcul par rapport à la méthode de résidu minimal. Le meilleur temps CPU est de 25 secondes, ce qui représente une accélération de 3,7.

Ces résultats sont très encourageants, tout particulièrement pour des problèmes incompressibles. L'utilisation du résidu minimal comme lisseur paraît donc parfaitement justifiée.

Les résultats sur ce cas test particulier ont été confirmés sur tous les autres cas de forgeage étudiés, nous permettant de considérer les valeurs de ε^f et ε^g comme indépendantes du problème.

Remarque :

Ces calculs ont été effectués pour la résolution d'un système linéaire issu de Forge3 avec

la précision standard demandée : 10^{-4} . Les résultats peuvent être un peu différents si on demande une précision plus élevée de l'ordre de 10^{-6} . En particulier, l'accélération est meilleure (environ 5). En règle générale plus la précision demandée est grande, plus la méthode de résidu minimal a du mal à converger plus l'accélération multigrille est importante.

3.4.2.5 Vérification des propriétés multigrille

On souhaite vérifier de nouveau l'indépendance de la convergence de la méthode multigrille ainsi optimisée à la taille de maille et estimer le taux de convergence asymptotique de la méthode.

– L'indépendance au maillage

Comme précédemment, on relève la valeur du résidu relatif en fin de cycle pour 5 maillages différents comportant respectivement 13897, 17486, 26416, 31841 et 37962 nœuds sur la grille fine. On constate sur la figure 3.15 que les profils obtenus sont superposables. La convergence est donc bien indépendante du maillage utilisé.

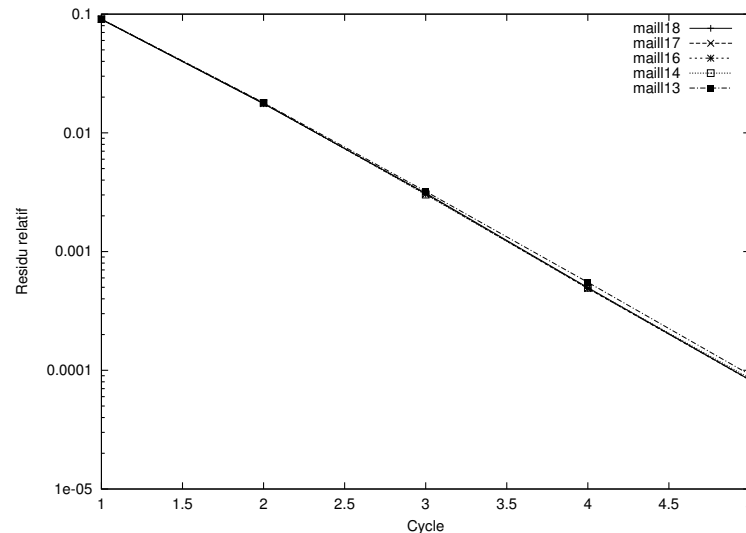


FIG. 3.15 – Indépendance de la convergence de la méthode multigrille au maillage avec $\varepsilon^f = 0.2$ et $\varepsilon^g = 0.1$

– Le taux de convergence asymptotique

Pour déterminer le taux de convergence asymptotique de la méthode multigrille, on utilise 6 maillages dont le nombre de nœuds sur la grille fine varie de 11000 à 38000. Pour chacun de ces maillages, on effectue 10 incréments de calcul, puis on relève le temps de calcul cumulé. La courbe obtenue avec ces 6 valeurs est approchée par une asymptote linéaire, puissance et logarithmique. On ne montre, sur la figure 3.16, que la courbe la plus proche, celle en $N \log N$.

Numériquement, on retrouve bien le taux de convergence théorique en $O(N \log N)$ des méthodes multigrilles, pour 10 incréments d'un écrasement entre tas plats frottants d'un lopin de rhéologie newtonienne.

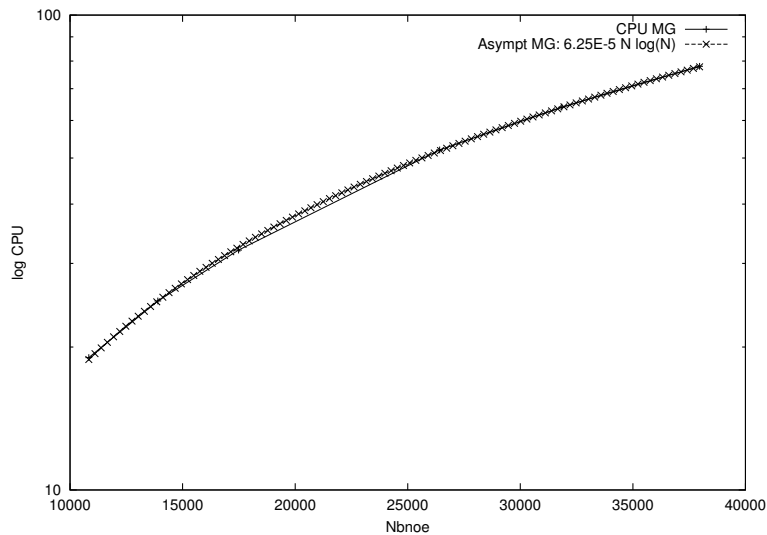


FIG. 3.16 – *Taux de convergence asymptotique de la méthode multigrille*

3.4.3 Test des méthodes 3-grilles

La méthode bigrille vérifie les propriétés multigrille et donne des résultats encourageants sur un cas de forgeage simple. On souhaite donc tester maintenant ces développements sur un calcul 3 grilles.

3.4.3.1 Problématique 3 grilles

Le problème est de définir l'enchaînement des calculs sur les 3 niveaux différents de grilles.

Bi-grille récursive

La première idée est d'utiliser la méthode bigrille de façon récursive. C'est-à-dire que l'on remplace la résolution par résidu minimal sur la grille grossière par une méthode bigrille. Cette solution est efficace si la méthode multigrille est compétitive avec la méthode de résidu minimal pour une précision de l'ordre de 0.1 (valeur ressortie de l'analyse bigrille). Sur le cas présent, une dizaine d'itérations de résidu minimal est suffisante pour diviser le résidu par 10. On ne peut donc espérer accélérer la méthode de cette manière.

Cycles classiques

Il existe dans la littérature différents types de cycles (voir par exemple figure 3.17).

3.4.3.2 Tests en V-cycle

Comme pour la méthode bigrille, on commence par faire des tests en multigrille idéale. On s'est limité à des cycles de type V-cycle, c'est à dire sans cycle intermédiaire lors de la descente et de la remontée de la grille fine à la grille grossière (contrairement au W-cycle). On a imposé que la précision sur la grille intermédiaire soit égale à la précision sur la grille fine. On a ainsi testé plusieurs cycles sur un cas de tas plat où la grille fine comporte 4233

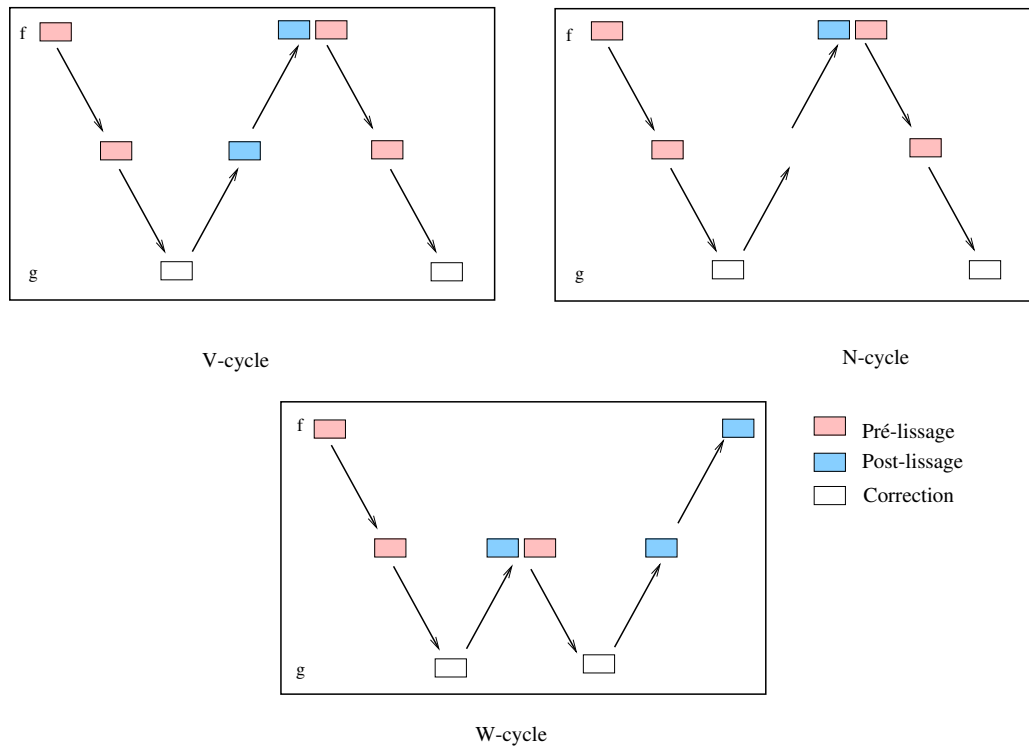


FIG. 3.17 – Exemples de cycles 3-grilles envisageables

nœuds et la grille la plus grossière 115.

Dans un premier temps, on a voulu vérifier l'utilisation d'une grille fine assez fine et d'une seule grille grossière, très grossière améliorerait les résultats. Cela revient à ne pas faire d'itérations de lissage sur la grille intermédiaire, mais seulement des opérations de restriction et de prolongement. On compare les résultats obtenus avec l'algorithme bigrille idéale et le cycle défini sur la figure 3.18. Les résultats sont reportés dans le tableau 3.9. On constate

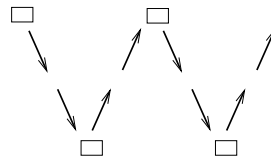
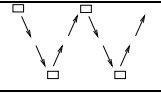


FIG. 3.18 – Méthode pseudo 2-grilles, sans lissage sur la grille intermédiaire

que la méthode bigrille est plus efficace en terme de cycles multigrilles et d'itérations de lissage. Cependant, on voit clairement l'utilité d'une grille grossière de plus petite taille au niveau du temps CPU, la seconde méthode étant globalement moins coûteuse.

Dans un second temps on compare un V-cycle classique avec un V-cycle auquel on a retiré les opérations de pré-lissage sur la grille grossière sur la grille intermédiaire lors de la descente (voir figure 3.19). Les résultats sont reportés dans le tableau 3.10 et montrent que la seconde méthode bien que plus efficace n'apporte pas une très grande amélioration.

	méthode bigrille	
Nombre de cycles	17	48
CPU	917	730

TAB. 3.9 – Comparaison de la méthode bigrille avec une pseudo méthode 2-grille

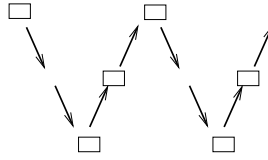


FIG. 3.19 – Méthode 3-grilles V-cycle modifié

3.4.3.3 Résultats 3 grilles

A la lueur de ces résultats, on a choisi de conserver le V-cycle classique. On a testé cette méthode sur plusieurs maillages toujours pour le même problème test de forgeage. Sur les grilles fines, on conserve les précisions $\varepsilon^f = 0.2$ et $\varepsilon^g = 0.1$. Les résultats sont reportés dans le tableau 3.11.

Sur la figure 3.20, on compare les trois méthodes sur un graphique. Bien qu'il soit difficile de faire un calcul d'asymptote sur trois points, il semble que la méthode trois-grilles ait un taux de convergence meilleur que la méthode 2-grilles.

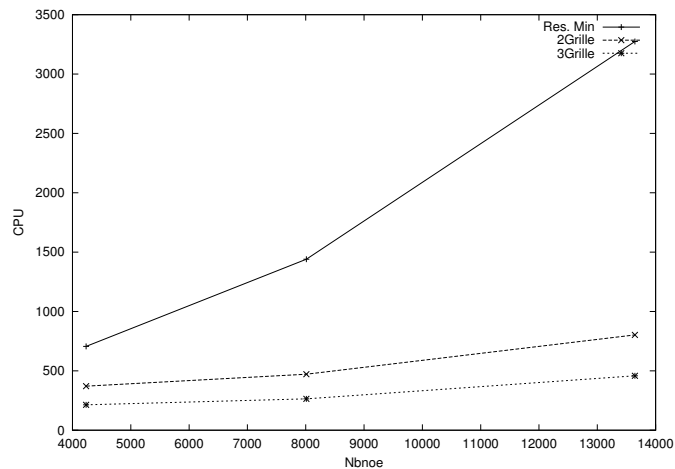
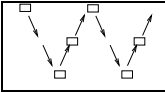
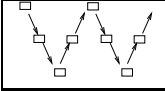


FIG. 3.20 – Comparaison des méthodes 3-grilles, 2-grilles et résidu minimal

Les premiers résultats obtenus avec trois niveaux de grilles sont très encourageants, les accélérations obtenues étant bien meilleures qu'avec la méthode deux grilles. Sur la grille intermédiaire, on a imposé la même précision que sur la grille fine. Comme on le verra dans la suite, la valeur de ces précisions est bien adaptée à tous les cas de forgeage que nous avons rencontré.

On a arrêté là l'étude des méthodes 3-grilles pour optimiser l'algorithme bigrille sur des

Cycles	Nb de cycles	CPU
	20	336
	18	324

TAB. 3.10 – Comparaison de différents cycles en multigrilles idéales

Nb nœuds grille fine	CPU résidu min.	CPU 2G	efficacité 2G	CPU 3G	efficacité 3G
4233	706 s	371 s	1.9	214 s	3.3
8011	1440 s	472 s	3	264 s	5.5
13465	3274 s	803 s	4	458 s	7

TAB. 3.11 – Résultats 3-grilles sur différents maillages

cas de forgeage plus complexes. Cette première approche nous a tout de même permis de montrer l'efficacité de plusieurs niveaux de maillages, et l'accélération supplémentaire qui en résulte.

3.4.4 Les améliorations du solveur bigrille

Tous les calculs précédents ont été faits sur un cas de tas plat avec une rhéologie newtonienne. Il s'agit maintenant de vérifier que nos conclusions restent valables pour des lois de comportement non linéaires et sur des cas de forgeage plus complexes.

3.4.4.1 Calculs avec une rhéologie non linéaire

Dans un premier temps, on vérifie que l'algorithme conserve de bonnes propriétés de convergence lorsque la rhéologie est non linéaire ($m=p=0.139$) c'est-à-dire lorsque la viscosité du matériau est hétérogène et que la matrice du problème est moins bien conditionnée.

Premiers résultats

On considère l'écrasement entre tas plats d'un cube dont le maillage fin comporte environ 14000 nœuds, pour des rhéologies newtonienne et viscoplastique. Dans le premier cas, on obtenait un facteur d'accélération de l'ordre de 5. Avec les mêmes paramètres de l'algorithme, pour le problème non linéaire, le facteur d'accélération tombe à 2,9. Si on examine en détail le comportement de l'algorithme, on se rend compte que la première étape de lissage sur la grille fine est beaucoup plus coûteuse que les autres. Elle prend environ 5 fois plus de temps durant le premier cycle que lors des suivants.

Pour circonvenir ce problème, on commence par effectuer une initialisation sur la grille grossière de manière à améliorer le premier lissage. Cela revient à initialiser l'algorithme multigrille par l'interpolation de la solution calculée sur la grille grossière. Cette méthode multigrille avec ajout d'une phase d'initialisation sur la grille grossière est appelée *full multigrid*, ce que nous traduirons dans la suite par *méthode multigrille complète*.

Dans notre cas test, cette méthode a permis de diminuer par 2 le temps du premier cycle, et de passer d'une accélération de 2,9 à 4,8.

Comparaison multigrille et multigrille complète

En forgeage, on utilise très rarement des rhéologies newtoniennes. Toutefois, on souhaite mettre en place une méthode multigrille qui soit efficace pour toutes les rhéologies. On vérifie donc que la méthode multigrille complète est aussi adaptée au problème linéaire. Pour cela, on compare les temps CPU des méthodes de résidu minimal, multigrille et multigrille complète (voir figure 3.21).

On observe que la méthode multigrille est un peu meilleure que la multigrille complète.

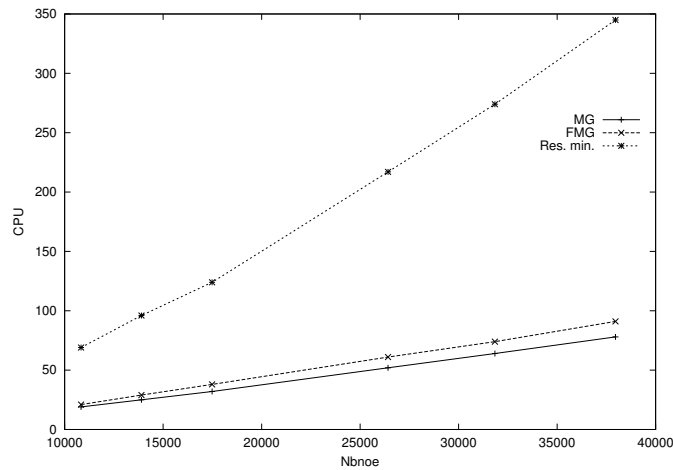


FIG. 3.21 – Comparaison des méthodes de résidu minimal multigrille et multigrille complète pour une rhéologie newtonienne

Cependant, elles ont des vitesses de convergence équivalentes et la méthode complète est plus robuste. On retient donc cette méthode pour les calculs avec une rhéologie newtonienne ou viscoplastique.

Sur la figure 3.22, on compare les temps de calcul de la méthode multigrille complète et de la méthode de résidu minimal. Ces résultats sont obtenus lors de la résolution de la première itération de Newton-Raphson du 20^{ème} incrément de déformation. On constate que l'accélération de la méthode multigrille reste équivalente à celle observée sur la figure 3.21.

Taux de convergence des méthodes multigrilles complètes

Le taux de convergence asymptotique des méthodes multigrilles complètes est généralement en $O(N)$. Pour vérifier cette propriété, nous approchons la courbe numérique expérimentale par une droite. Ces deux courbes sont tracées sur la figure 3.23, pour la résolution de la première itération de Newton-Raphson du second incrément de déformation avec une rhéologie non linéaire. Le taux de convergence asymptotique est bien linéaire.

Ce comportement asymptotique en $O(N)$ n'est pas toujours parfaitement vérifié. Par exemple, si on effectue la même analyse au dixième incrément de déformation, on trouve un taux de convergence en 1,27 (voir figure 3.24). Cela s'explique sans doute par les modifications du domaine (maillages non emboîtés), aspect que nous étudierons dans les paragraphes suivants.

On a donc mis au point une méthode multigrille complète qui vérifie les propriétés

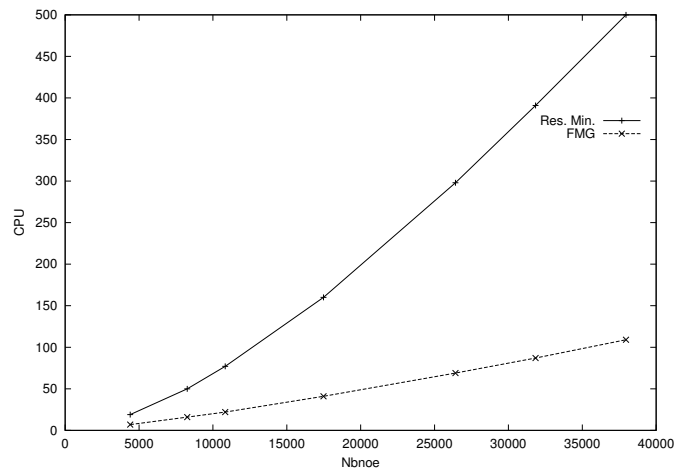


FIG. 3.22 – Comparaison des méthodes de résidu minimal et multigrille complète pour une rhéologie viscoplastique

théoriques des méthodes multigrilles. Il nous reste maintenant à la tester sur des cas de forgeage plus complexes en terme de géométrie d'outil.

3.4.4.2 Le poinçonnement de cylindre

Pour continuer à valider la méthode, nous passons à des cas plus complexes, avec des outils non plans et un contact qui évolue. Nous étudions en particulier un cas de poinçonnement d'un cylindre par un seul outil courbe (voir figure 3.25). Le nombre de nœuds sur la grille fine est de 15 628. On construit le maillage fin en raffinant le maillage grossier et en projetant les nœuds fins sur la surface.

Les opérateurs de transfert

Lorsque le maillage est déformé, les nœuds du maillage fin qui se trouvaient au milieu d'une arête du maillage grossier n'ont aucune raison de rester au milieu de cette arête. Cela signifie que les opérateurs de prolongement et restriction définis par les relations (3.5) et (3.6) doivent être réactualisés et prendre une forme plus complexe (d'où un surcoût à leur utilisation). Nous avons donc étudié l'utilité de cette mise à jour en comparaison avec la conservation des mêmes opérateurs, en négligeant la déformation relative des grilles. On a fait ces comparaisons sur l'écrasement d'un cylindre entre tas plats. Pour calculer les opérateurs modifiés sans perturber toute la structure du code, on a forcé les nœuds fins internes à rester au milieu des arêtes du maillage grossier (voir figure 3.26(b)) et calculé les opérateurs de façon exacte seulement pour les nœuds frontières en utilisant les formules (3.5) et (3.6). Cela revient à utiliser une méthode ALE pour les nœuds internes. Pour l'autre calcul, on a gardé les opérateurs constants sans prendre en compte le déplacement relatif des nœuds fins (voir figure 3.26(a)).

Les tests effectués montrent que les deux méthodes donnent des résultats tout à fait équivalents. La première étant plus complexe et plus coûteuse en temps, nous avons retenu la seconde.

Premiers résultats

Lors des premiers calcul sur le cas test de poinçonnement, on constate que le facteur d'ef-

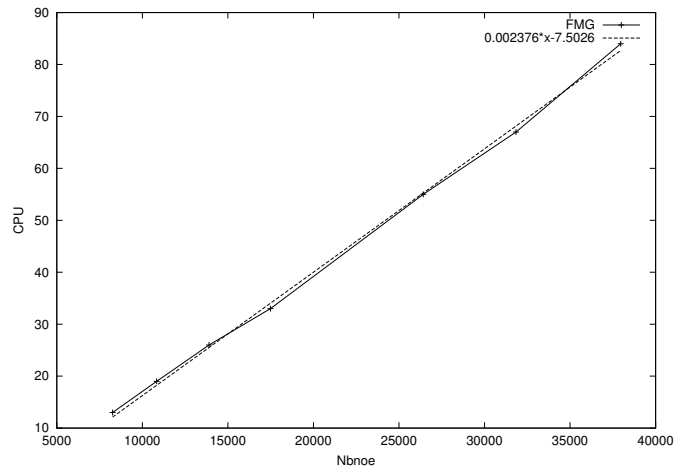


FIG. 3.23 – Taux de convergence asymptotique numérique en $O(N)$ pour la méthode multigrille complète. Deuxième incrément de déformation de l'écrasement entre tas plats d'un lopin de rhéologie viscoplastique

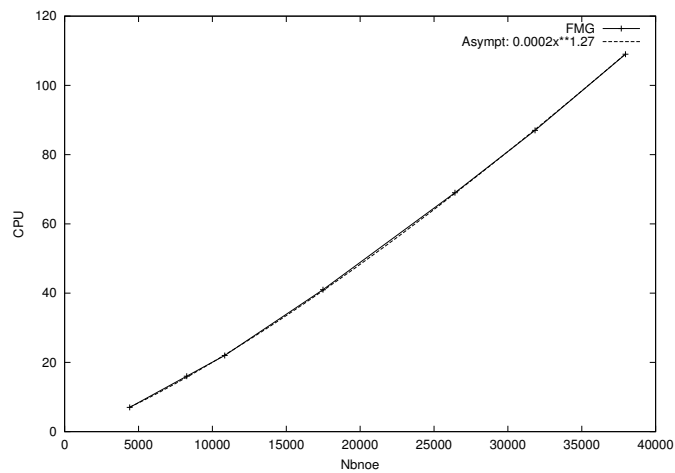


FIG. 3.24 – Taux de convergence asymptotique numérique pour la méthode multigrille complète avec une rhéologie non linéaire. Dixième incrément de déformation

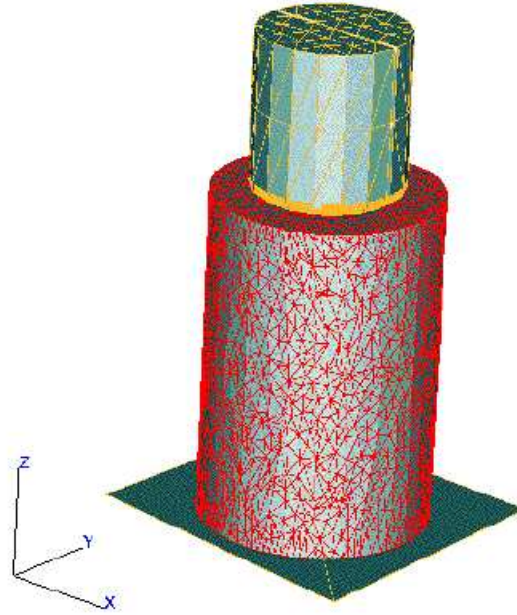


FIG. 3.25 – Cas test du poinçonnement de cylindre, le maillage grossier et les outils

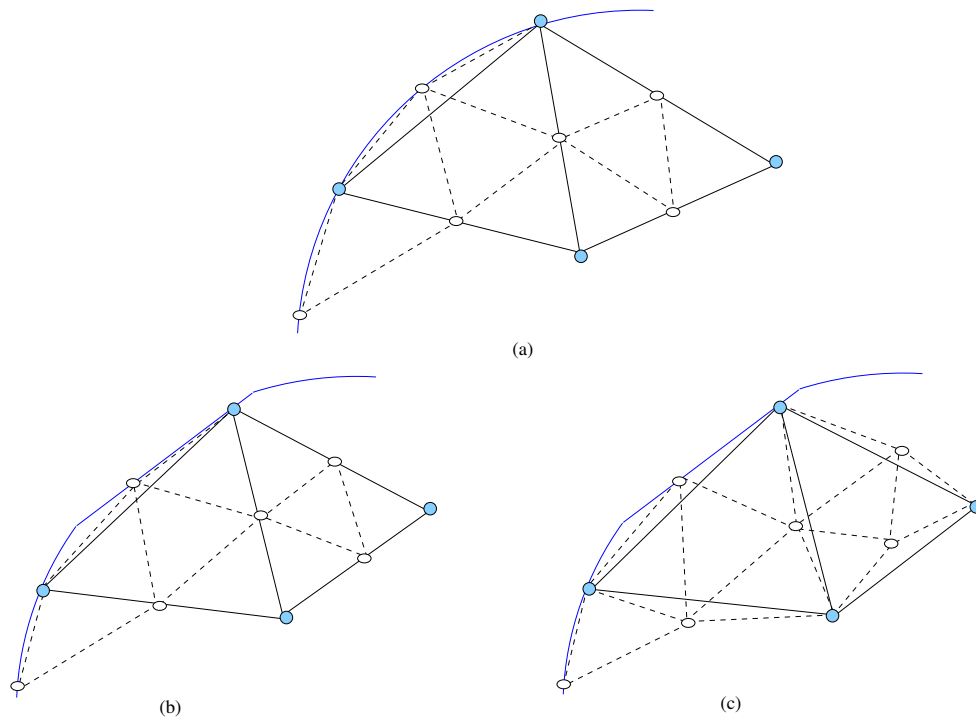


FIG. 3.26 – Comparaison des différents opérateurs: (a) Configuration initiale, (b) Après déformation, avec remaillage des noeuds internes au milieu des arêtes, les opérateurs sont les opérateurs exacts, (c) Après déformation, sans remaillage, les opérateurs sont malgré tout considérés constants

ficacité de la méthode multigrille est moins bon que pour le problème d'écrasement entre tas plats (voir tableau 3.12). La seule différence entre les deux problèmes provient de l'uti-

Itérations	CPU Résidu minimal	CPU multigrille	Accélération
1	142	42	3.4
2	303	86	3.5
3	210	57	3.7
4	209	61	3.4
5	515	262	1.9
6	199	94	2.1
7	62	21	2.9
8	181	65	2.8

TAB. 3.12 – Accélération de la méthode bigrille sur le cas de poinçonnement aux différentes itérations de Newton-Raphson du premier incrément

lisation d'un outil courbe. On s'est donc intéressé aux phénomènes de contact en relation avec la méthode multigrille.

Contact et condition de Galerkin

– Contact sur la frontière

L'utilisation d'un outil courbe implique qu'au sommet du cylindre certains nœuds sont en contact et d'autres non. Si on écrit un terme de la matrice grossière reliant deux composantes de la vitesse du nœud I de la figure 3.27, on obtient :

$$\mathbf{A}_{I_1 I_2}^H = \sum_{j,l} p_{H I_1 j}^h \mathbf{A}_{jl}^h p_{H I_2 l}^h$$

Bien que I ne soit pas en contact avec l'outil, la proximité du nœud k (maillage fin) en contact avec l'outil se traduit par une contribution de contact sur $\mathbf{A}_{I_1 I_2}^H$. Par exemple, \mathbf{A}_{kk}^h est un terme de pénalisation qui contribue à \mathbf{A}_{II}^H .

$$\mathbf{A}_{II}^H = \dots + p_{H I k}^h \mathbf{A}_{kk}^h p_{H I k}^h$$

Le terme matriciel contenant I prend donc des valeurs du même ordre de grandeur que si le nœud I était en contact.

Pour résoudre ce problème, on a essayé de rendre la matrice \mathbf{A}^H (du problème sur le domaine Ω_H) plus proche de la matrice $\tilde{\mathbf{A}}_H$ éléments finis. Pour cela, on a décomposé la matrice \mathbf{A}^h en

$$\mathbf{A}^h = \mathbf{A}_{comp}^h + \mathbf{A}_{\rho}^h$$

où \mathbf{A}_{comp}^h représente la partie issue du comportement et \mathbf{A}_{ρ}^h la partie issue du contact. De la même façon, on décompose le résidu R^h en :

$$R^h = R_{comp}^h + R_{\rho}^h$$

1. On a appliqué la condition de Galerkin uniquement à \mathbf{A}_{comp}^h et assemblé les termes de contact sur la matrice grossière :

$$\mathbf{A}^H = p_h^H \mathbf{A}_{comp}^h p_H^h + \tilde{\mathbf{A}}_{comp}^H$$

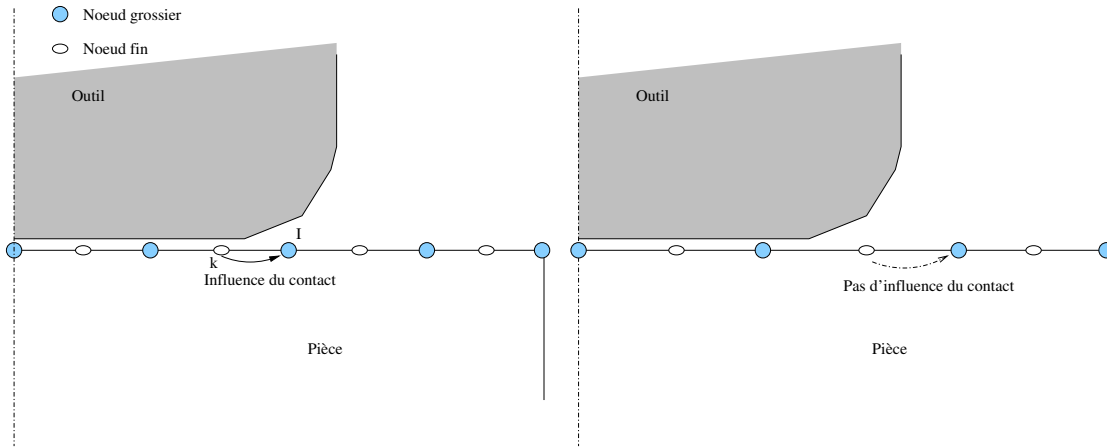


FIG. 3.27 – Influence du contact grille fine sur les nœuds frontières grossiers

Cette méthode conduit à des processus divergents.

2. Dans le même ordre d'idée, on a utilisé des opérateurs de transfert différents pour la partie comportementale et la partie contact :

$$\mathbf{A}^H = p_h^H \mathbf{A}_{comp}^h p_H^h + I_H^h \mathbf{A}_\rho^h I_H^h$$

où I_h^H représente l'opérateur d'injection défini par :

$$I_h^H(v_k) = \begin{cases} v_k & \text{si } k \in \text{ grille grossière} \\ 0 & \text{sinon} \end{cases}$$

3. Une autre méthode a consisté à conserver la condition de Galerkin pour la construction de la matrice grossière dans son ensemble mais en utilisant un opérateur qui vérifie :

$$p_h^H I_k = 0$$

avec les notations de la figure 3.27.

Cette méthode a permis d'améliorer l'efficacité des multigrilles sur certaines résolutions mais celle-ci reste globalement insatisfaisante par rapport aux résultats obtenus sur le tas plat.

– Influence du contact sur la matrice grossière

On s'est alors intéressé à l'influence de la condition de Galerkin sur les termes reliant les degrés de liberté de 2 nœuds distincts dans la matrice grossière. Par exemple, sur la figure 3.27, le terme de contact apparaissant dans le bloc diagonal de la matrice fine pour le nœud k apporte une contribution aux termes reliant les nœuds I et K .

On a :

$$\mathbf{A}_{IK}^H = \sum_{j,l} p_{HIj}^h \mathbf{A}_{jl}^h p_{HKl}^h = \dots + p_{HIk}^h \mathbf{A}_{kk}^h p_{HKk}^h + \dots$$

Comme \mathbf{A}_{kk}^h contient des termes de pénalisation du contact, \mathbf{A}_{IK}^H prend aussi des valeurs élevées d'un terme de pénalisation. On obtient ainsi avec des termes de pénalisation hors du bloc diagonal. L'utilisation d'un préconditionneur diagonal bloc n'est pas très efficace sur de telles matrices, les termes extra-diagonaux n'étant pas

contrôlés par le préconditionneur. Sur le tableau 3.13, on voit qu'effectivement le temps de calcul sur grille grossière est la partie la plus coûteuse des cycles de la méthode multigrille.

Il s'avère donc nécessaire d'utiliser un préconditionnement plus performant qui agisse

Cycle	1	2	3	4	5	6	7	8	9	10
CPU correction	2	4	6	2	3	6	1	2	6	2
CPU cycle	2	7	9	4	6	8	3	6	7	4

TAB. 3.13 – Comparaison du temps de CPU de la correction grille grossière par rapport au temps CPU du cycle pour le cas de poinçonnement

sur les termes extra-diagonaux. Nous avons recours à celui de Crout incomplet, présenté en début de chapitre.

On a donc testé son influence, non seulement sur la grille grossière mais aussi sur la grille fine. Les résultats sont reportés dans les tableaux 3.14 et 3.16, pour les préconditionneurs IC et Icvp respectivement.

Itérations de N.Raphson	Résidu min.	Multigrille \mathcal{M}_{ivp}	Multigrille (ICvp GF/GG) ^a	Multigrille (ICvp GG) ^b
1	142	42	29+54 ^c	39 (3.6) ^d
2	303	86	45+54	56 (5.4)
3	210	57	36+54	43 (4.9)
4	209	61	38+54	46 (4.5)
5	515	262	109+54	139 (3.7)
6	199	94	37+54	40 (5)
7	62	21	20+54	22 (2.9)
8	181	65	39+54	49 (3.7)
Total	1821	688	785	434 (4.2)

TAB. 3.14 – Influence du préconditionnement sur le CPU de la méthode bigrille dans le cas de poinçonnement et pour différentes itérations de Newton-Raphson

^a méthode multigrille utilisant le préconditionneur Crout incomplet sur la grille fine et sur la grille grossière

^b méthode multigrille utilisant le préconditionneur Crout incomplet sur la grille grossière uniquement

^c Le second chiffre représente le temps de factorisation de la décomposition incomplète grille fine

^d Entre parenthèse figure l'accélération par rapport à la méthode résidu minimal préconditionné par \mathcal{M}_{ivp}

On compare les résultats sur plusieurs itérations de Newton-Raphson. En effet, on remarque que l'accélération des multigrilles par rapport au résidu minimal peut varier suivant les itérations et leur complexité.

On voit sur la troisième colonne du tableau 3.14 que le temps nécessaire à la factorisation de la matrice fine pour le calcul du préconditionnement est très important par rapport au temps de la méthode multigrille. Il n'est donc pas avantageux de modifier le préconditionnement de la matrice fine pour les itérations de lissage.

Par contre, le préconditionneur par factorisation incomplète semble régler les problèmes de convergence sur la grille grossière. Dans la quatrième colonne du tableau,

on voit en effet que le CPU est bien inférieur à celui de la deuxième (sauf pour l'itération 7).

Si on considère le temps nécessaire à la correction grille grossière par rapport au temps d'un cycle avec le préconditionneur ICvp, on constate qu'il est redevenu négligeable. Les résultats sont reportés dans le tableau 3.15.

Cycle	1	2	3	4	5	6	7	8	9
CPU correction	6	1	2	2	2	2	1	2	0
CPU cycle	6	7	13	11	12	11	10	11	10

TAB. 3.15 – Comparaison du temps de CPU de la correction par rapport au temps CPU du cycle sur une résolution avec préconditionneur ICvp

On réalise les mêmes calculs en effectuant le préconditionnement par factorisation incomplète sur le bloc vitesse uniquement. Les résultats sont donnés dans le tableau 3.16.

Itérations de N.Raphson	Résidu min.	Multigrille \mathcal{M}_{ivp}	Multigrille (IC GF/GG) ^a	Multigrille (IC GG) ^b
1	142	42	47+17 ^c	43 (3.3) ^d
2	303	86	79+17	79 (3.8)
3	210	57	54+17	53 (4)
4	209	61	56+17	60 (3.5)
5	515	262	197+17	233 (2.2)
6	199	94	48+17	56 (3.6)
7	62	21	26+17	22 (2.8)
8	181	65	58+17	57 (3.2)
Total	1821	701	701	603 (3)

TAB. 3.16 – Influence du préconditionnement sur le CPU dans le cas de poinçonnement

^a méthode multigrille utilisant le préconditionneur de Crout incomplet sur la grille fine et sur la grille grossière

^b méthode multigrille utilisant le préconditionneur de Crout incomplet sur la grille grossière uniquement

^c Le second chiffre représente le temps de factorisation de la décomposition incomplète grille fine

^d Entre parenthèse figure l'accélération par rapport à la méthode résidu minimal préconditionné par \mathcal{M}_{ivp}

On fait le même type de constatations que précédemment tout en observant que les résultats sont moins probants. On retrouve en cela la différence de performance des deux types de factorisation incomplète que l'on a observé au début du chapitre.

Le préconditionneur par ICvp sur la grille grossière uniquement permet ainsi de retrouver les performances qui avaient été obtenues sur les cas d'écrasement entre tas plats, malgré les conditions de contact plus complexes.

3.4.5 La procédure de remaillage

La procédure de remaillage doit être modifiée dans le cadre de l'utilisation des méthodes multigrilles. Nous avons en effet choisi de générer les différents niveaux de maillages en partant du maillage grossier. Ce choix nous permet de remailler la grille grossière et non la grille fine, ce que nous espérons moins coûteux. Toutefois, avec cette stratégie, il faut faire face à plusieurs problèmes.

3.4.5.1 Remaillage grossier

Pendant la procédure de remaillage du maillage grossier, il faut se rappeler que ce maillage sera raffiné et que le maillage obtenu sera un maillage de calcul éléments finis. Sur la figure 3.28, on montre que lors de la projection des nœuds milieux sur la surface, on peut obtenir des éléments fins aplatis.

Il faut donc introduire dans la procédure de remaillage un critère pour prendre en compte

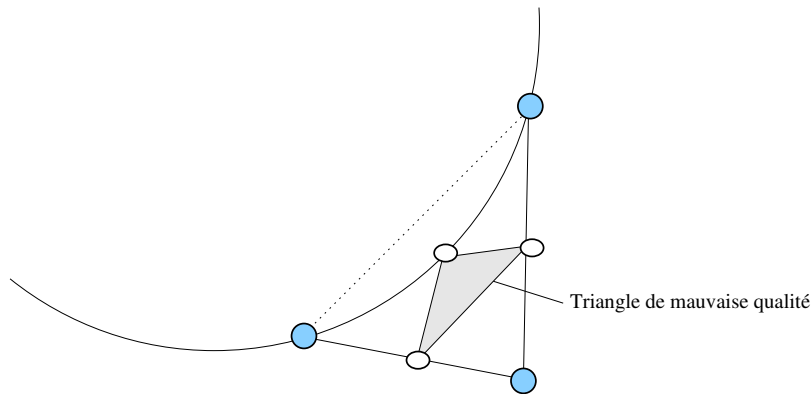


FIG. 3.28 – *Obtention d'un élément de mauvaise qualité après raffinement*

la géométrie issue de la grille fine.

3.4.5.2 Sauvegarde du maillage surfacique fin

Lors d'un remaillage classique, la surface de la pièce est conservée au cours du remaillage par la prise en compte de la courbure du maillage (conservation des coins). Si l'on veut remailler une surface fine par un maillage grossier, il faut garder en mémoire ce contour fin. Sur la figure 3.29, on montre que les angles ne sont pas conservés pendant le remaillage si on ne sauvegarde pas le contour fin.

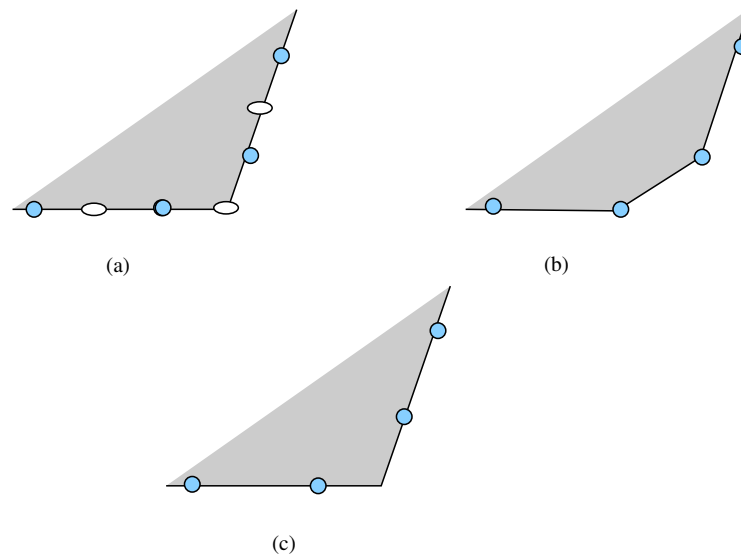


FIG. 3.29 – (a) *Maillage initial*, (b) *Maillage considéré lors du remaillage si on ne sauvegarde pas le maillage fin* (c) *Maillage considéré lors du remaillage si on sauvegarde le maillage fin*

En considérant cette remarque, on voit qu'il faut modifier le critère de remaillage pour prendre en compte lors du remaillage un critère de respect de la surface et non le critère de courbure classique dans le remaillageur [Coupez91]. Ce type de modification du critère de remaillage a déjà été abordé dans [Rodriguez villa97].

3.5 Résultats

3.5.1 Résultats sur le forgeage du triaxe

Comme nous n'avons pas développé de procédure de remaillage associée au raffinement de maillage, nous procédons au forgeage du lopin du triaxe jusqu'à dégénérescence du maillage. On présente les résultats obtenus sur un lopin comportant 14785 nœuds et 71048 éléments sur le maillage fin. Le coefficient de frottement est de 0,2.

3.5.1.1 Rhéologie newtonienne

On compare les temps de calcul des méthodes de résidu minimal préconditionné par bloc diagonal \mathcal{M}_{ivp} ou par Crout incomplet (ICvp) et de la méthode multigrille utilisant ces mêmes préconditionneurs (pour le calcul grille grossière) et un préconditionneur bloc diagonal \mathcal{M}_{ivp} sur la grille fine. Dans la pratique, on ne simule qu'un sixième du problème. On montre dans le tableau 3.17 les résultats obtenus après 360 incréments de calcul ce qui représente 77% du forgeage (voir figure 3.30).

Solveur	Résidu minimal		Multigrille	
	Bloc Diago	ICvp	Bloc Diago	ICvp
CPU	51 h 32 mn	41h 44mn	34h 3mn	28h41mn
Accélération		1,23	1,51	1,8

TAB. 3.17 – Comparaison de temps de forgeage sur le cas du triaxe pour 360 incréments de déformation pour une rhéologie newtonienne

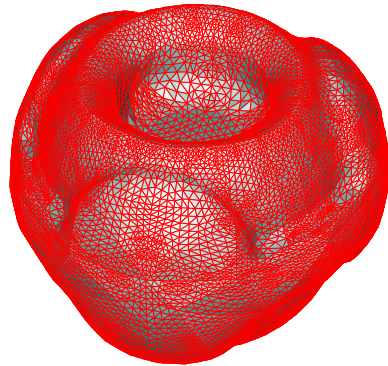


FIG. 3.30 – Déformation du lopin du triaxe newtonien après 360 incréments

On constate que le préconditionneur Crout incomplet améliore les résultats de la méthode de résidu minimal (42h au lieu de 51h30) et que l'utilisation de ce même préconditionneur sur le calcul grille grossière permet d'accélérer la méthode multigrille.

On trouve un facteur d'accélération de l'ordre de 1,8 entre la meilleure méthode multigrille et le solveur initial. Ce résultat peut paraître décevant mais après plusieurs dizaines

d'incrément, les opérateurs de transfert que nous avons considérés constants sont assez éloignés des opérateurs linéaires, qu'il faudrait calculer par interpolation. L'accélération multigrille s'en ressent donc. Toutefois, l'exemple considéré représente un cas extrême que l'on ne considère jamais en pratique : faire plus de 300 incréments de calcul sans remaillage. Avec un remaillage plus habituel (par exemple tous les 10 incréments) et une réactualisation en conséquence des opérateurs de transfert, l'efficacité multigrille devrait être bien supérieure.

On compare aussi le pourcentage du forgeage calculé avec les différents algorithmes au bout d'un temps de calcul donné. Les résultats sont donnés sur la tableau 3.18.

Solveur	Résidu minimal		Multigrille	
	Bloc Diago	ICvp	Bloc Diago	ICvp
% du forgeage	57,1%	67,9 %	76,7 %	80,7 %

TAB. 3.18 – *Comparaison du pourcentage du forgeage total atteint au bout de 33h de calcul avec les différents algorithmes*

3.5.1.2 Rhéologie viscoplastique

On fait les mêmes calculs avec une rhéologie viscoplastique ($m=p=0.139$). La dégénérescence du maillage est atteinte à l'incrément de déformation 57 (voir figure 3.31). On compare à nouveau les temps de calcul de la méthode de résidu minimal avec les préconditionneurs \mathcal{M}_{ivp} ou Crout incomplet et la méthode multigrille avec préconditionneur Crout incomplet sur la grille grossière. Les résultats sont reportés dans le tableau 3.19. L'accélération de la méthode multigrille est ici aussi de l'ordre de 1,9. La dégénérescence

Solveur	Résidu minimal		Multigrille
	Bloc Diago	ICvp	ICvp
CPU	29 h 32 mn	16h 26mn	15h 9mn
Accélération		1,8	1,95

TAB. 3.19 – *Comparaison de temps de forgeage sur le cas du triaxe viscoplastique pour 57 incréments de déformation*

du maillage et l'approximation des opérateurs de transfert peuvent également expliquer cette baisse de performance par rapport aux résultats obtenus précédemment. En effet, même si la déformation semble peut importante sur la figure 3.31, le maillage est dégénéré.

Il faut également rappeler la remarque faite au paragraphe 3.4.2.4, c'est à dire que le solveur multigrille est d'autant plus efficace (par rapport au résidu conjugué) que la précision demandée est grande. Or l'algorithme de Newton-Raphson de **Forge3** a été particulièrement optimisé. Lorsque l'itérée se trouve loin de la solution du problème non linéaire, la précision demandée pour les calcul de m'itérée suivant est faible. Dans ces conditions, la méthode multigrille n'apporte pas une amélioration significative.

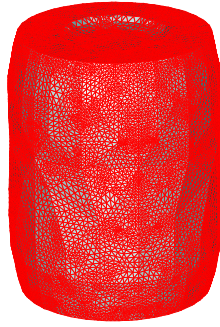


FIG. 3.31 – Déformation du lopin du triaxe viscoplastique après 57 incréments

3.5.2 Résultats sur le martelage d'une tôle épaisse

On étudie un cas industriel de martelage de tôle épaisse. On considère un lopin parallélépipédique de taille $2,3m \times 3,1m \times 1m$, de rhéologie viscoplastique ($m=p=0,18$) discrétisé en 35202 nœuds et 186344 éléments. Le coefficient de frottement entre la pièce et l'outil vaut 0,8. Ce cas est présenté sur la figure 3.32.

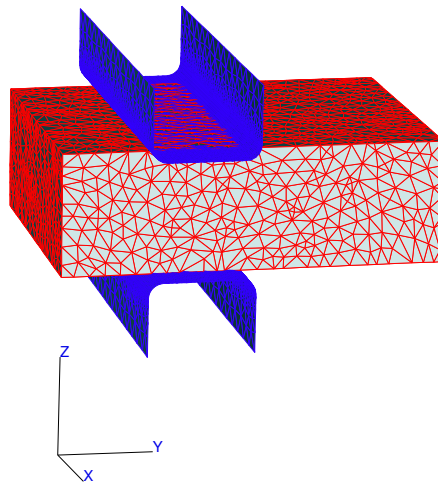


FIG. 3.32 – Description du cas de martelage d'une tôle épaisse

On compare les résolutions par les méthodes de résidu minimal et multigrille sur une dizaine d'incrément. On résume dans le tableau 3.20 quelques uns des rapports d'accélération observés.

Notons que pour les résolutions marquées d'une *, la méthode de résidu minimal n'a pas convergé (nombre maximum d'itérations atteint, 7000) alors que la méthode multigrille a atteint le résidu relatif demandé. Lors de ces résolutions l'accélération est donc supérieure à celles notées. On montre dans le tableau 3.21, les comparaisons de temps de calcul né-

Incrément	Résolution	Résidu minimal	Multigrille	Accélération
1	1	2025*	369	5,48
	2	1916	358	5,35
2	1	1576	246	6,4
	2	1994	383	5,2
3	1	1748	272	6,42
	2	1991	485	4,1
4	1	2019*	383	5,27
	2	1493	401	3,72
5	1	1960	314	6,24
	2	1569	287	5,46

TAB. 3.20 – Comparaison des coûts de résolution par les algorithmes de résidu minimal et de multigrille sur le cas de martelage

cessaires aux deux solveurs pour calculer 7 incréments du cas de martelage. Sur la figure 3.33, on montre la configuration finale du cas après 60 incréments de déformation.

Solveur	Résidu minimal	Multigrille	Accélération
Préconditionneur	Bloc Diago	ICvp	
CPU	37h 7 min	7h 20 min	~ 5

TAB. 3.21 – Comparaison de temps de forgeage sur le cas de martelage pour le calcul de 7 incréments

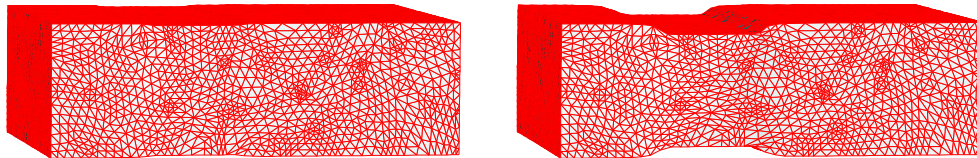


FIG. 3.33 – Déformation du lopin après 7 et 60 incréments sur le cas de martelage

3.5.3 Résultats sur un emboutissage de plaque mince

Le dernier cas présenté est l'emboutissage d'une plaque mince. Le lopin initial est une plaque rectangulaire ($4m \times 1,7m$) d'épaisseur $5,6cm$ qui est emboutie entre deux outils cylindriques (voir figure 3.34(a)). Le lopin comporte 12815 nœuds et 54400 éléments. On montre sur la figure 3.34 les configurations initiale et finale du cas d'emboutissage. On montre sur le tableau 3.22, les temps de calcul sur plusieurs résolutions du cas d'emboutissage. Les chiffres marqués de * sont des résolutions pour lesquelles la méthode de résidu minimal n'a pas atteint le résidu relatif demandé en 7000 itérations.

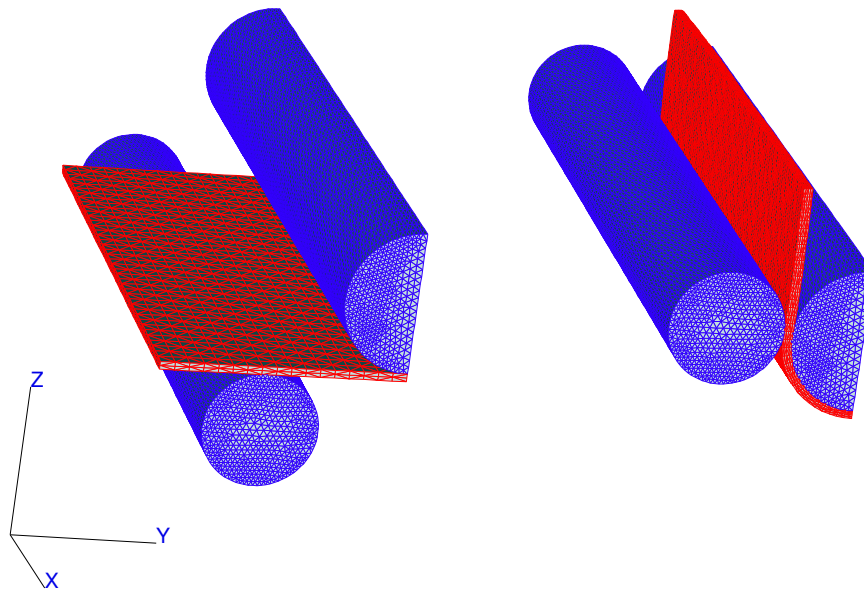


FIG. 3.34 – (a) Configuration initiale du cas d'emboutissage, (b) configuration finale du cas d'emboutissage

Incrément	Résolution	Résidu minimal	Multigrille	Accélération
1	1	650 s	79 s	8,22
2	1	919 s	96 s	9,57
	2	1440*s	132 s	10,9
	3	1439* s	109 s	13,2
	4	1440* s	128 s	11,25
3	1	554 s	72 s	7,69
	2	1440* s	182 s	7,9
	3	1371 s	147 s	9,32
9	1	381 s	91 s	4,18
	2	339 s	87 s	3,89
	3	1440* s	569 s	2,53
	4	1440* s	289 s	4,98

TAB. 3.22 – Comparaison des coûts de résolution résidu minimal et multigrille sur le cas d'emboutissage pour quelques résolutions

On observe sur le tableau 3.22 de très bons facteurs d'accélération. Pour le calcul de 9 incréments de déformation, les accélérations varient de 2,53 à 13,2, la moyenne se situant à 4,4.

3.6 Conclusion

Le but de cette étude était la mise en place d'un solveur multigrille adapté à la simulation numérique du forgeage. Nous avons fait les choix des différents éléments de la méthode en tenant compte des impératifs dictés par le procédé et le code de simulation avec lequel nous avons travaillé.

La forme de la matrice issue de la modélisation du problème à deux champs (vitesse et pression) a inspiré le lisseur. La méthode de résidu minimal préconditionné utilisée est peu commune dans la littérature multigrille. Toutefois, les résultats obtenus en bigrille idéale puis en bigrille et 3-grilles ont validé ce choix.

Le calcul de la matrice grossière par la condition de Galerkin se prête bien à la parallélisation du code. Son implémentation optimisée la rend, par ailleurs, très compétitive par rapport à un assemblage classique.

L'algorithme de résolution multigrille a été amélioré pour traiter la plus grande gamme de cas de forgeage. Ainsi, nous avons opté pour une méthode de multigrille complète pour une meilleure efficacité dans le cas de rhéologies viscoplastiques. De la même façon, un préconditionneur par factorisation de Crout incomplète a été mis en place pour gérer les problèmes de convergence générés par la méthode de gestion du contact par pénalisation.

Par souci d'efficacité, on a utilisé des opérateurs de transfert simplifiés. Ils montrent leurs limites pour les grandes déformations du maillage (les nœuds fins quittent le milieu des arêtes grossières). Ces fortes déformations sont dues au fait que nous n'avons pas abordé, faute de temps, la procédure de remaillage des différents niveaux de grilles.

Au final, on a obtenu un solveur multiniveau qui donne de bons résultats sur différents cas de forgeage et qui présente des taux de convergence prometteurs pour le calcul sur des cas de calcul plus importants. Il est difficile de donner une estimation générale du gain apporté par les méthodes multigrilles sur un cas quelconque de forgeage. En effet, sur le cas du triaxe et de l'emboutissage, pour des lopins de taille équivalente (environ 15000 nœuds), les accélérations observées sont assez différentes.

Si on se place dans le dernier cas de forgeage où l'accélération moyenne est de 4,4 pour 13000 nœuds environ, on peut estimer à 10 l'accélération moyenne pour 65000 nœuds et à 14 pour 130000 nœuds.

Sur les différents cas de forgeage présentés, il semble que l'accélération est d'autant plus élevée que la déformation du maillage est localisée. Ceci pourrait s'expliquer par le fait que les méthodes multigrilles captent ces déformations assez rapidement à l'aide de la grille grossière alors que la méthode de résidu minimal doit converger sur le maillage global.

3.7 Perspectives

A la vue des résultats obtenus, plusieurs développements peuvent être envisagés pour améliorer l'efficacité du solveur multigrille.

La méthode 3-grilles brièvement abordée dans cette étude donne des résultats encourageants. On a, par la suite, intégré une série d'améliorations à la méthode bigrille. L'application des optimisations à la méthode 3-grilles pourrait mener à un solveur encore plus efficace.

On a montré sur la figure 3.5 que le pourcentage représenté par la factorisation dans un calcul préconditionné par la méthode IC est non négligeable. On peut envisager une procédure de type quasi-Newton (construction du préconditionneur seulement tous les X incréments) pour améliorer le temps de calcul.

On a montré l'efficacité de la méthode de résidu minimal comme lisseur pour notre problème. L'application de lisseurs plus classiques, même si leur convergence sur notre type de système n'est pas prouvée, pourrait donner des résultats de convergence intéressants.

Enfin, on peut tester l'utilisation de la méthode multigrille comme préconditionneur de la méthode de résidu minimal en remplaçant la résolution du système

$$\mathcal{M}x = y$$

par la résolution, sur la grille grossière, de

$$A_H x_H = y_H$$

et

$$x = p_H^h x_H.$$

Chapitre 4

CONCLUSION GÉNÉRALE

4.1 Conclusion

La première partie de ce travail a consisté à améliorer la précision de l'algorithme de contact. La gestion du contact entre la pièce et l'outil par une méthode de pénalisation explicite amenait à des pénétrations de nœuds de la pièce dans l'outil ou à des blocages de nœuds sur des angles convexes en fin d'incrément. La mise en place d'une procédure implicite du contact par pénalisation a permis de résoudre ces problèmes de précision. La réactualisation de la condition de contact à chaque itération de Newton-Raphson a permis de tenir compte de l'évolution du vis à vis avec l'outil au cours du calcul. Pour donner à cet algorithme la robustesse nécessaire pour s'appliquer à différents cas de forgeage, on a mis en place une méthode de lissage des normales basée sur la construction d'un outil décalé. Cette procédure a permis d'augmenter la robustesse de l'algorithme. Mais aussi d'améliorer la précision de la description de la surface de contact entre la pièce et l'outil en annulant la pénétration numérique qui était nécessaire au calcul de normale. Cette amélioration de la précision se fait au prix d'une augmentation du coût de calcul. Ce qui nous amène au second point de cette étude.

Dans la seconde partie de cette étude, nous avons développé un nouveau solveur dans le but de diminuer le temps de calcul des simulations. La méthode multigrille développée est originale et donne des résultats encourageants. Le taux de convergence en $O(N)$ au lieu de $O(N^{\frac{3}{2}})$ pour le résidu minimal nous permet d'espérer des accélérations de l'ordre 10 pour des maillages de 50000 nœuds ou 14 pour 130000 nœuds (ces estimations étant faite pour une accélération de 4,4 à 13000 nœuds).

4.2 Perspectives

Les perspectives de ce travail sont nombreuses. Elles sont le reflet du peu de travaux liés à l'utilisation de méthodes multigrilles dans le domaine des problèmes incompressibles et instationnaires en éléments finis.

Dans un premier temps, il est primordial de développer une méthode de remaillage des différents niveaux de grilles. Ceci soulève des problèmes de respect d'une frontière fine lors du remaillage d'une grille grossière. De plus la méthode de raffinement utilisée risque de générer des tétraèdres de mauvaise qualité. Ces deux phénomènes sont les problèmes à résoudre pour mettre en place de la procédure de remaillage.

La mise en place d'un solveur 3-grilles intégrant les améliorations développées pour la méthode bigrille nous semble une première étape facile pour augmenter l'efficacité de la méthode.

On a fait le choix d'utiliser une méthode géométrique. Une autre optique qui peut être envisagée est de développer une méthode algébrique. Plusieurs questions restent en suspens quant à l'application de ces méthodes au type de matrices que nous avons :

- Quelle sera l'influence des termes de pénalisation du contact sur la construction des problèmes grossiers?
- Comment gérer la formulation mixte et les multigrilles algébriques?
- Comment utiliser ce solveur avec la procédure de calcul parallèle?

Mais cette méthode présente aussi un avantage certain. On n'utilise qu'un seul maillage ce qui élimine les problèmes liés à la procédure de remaillage et au nombre maximal de grille utilisé.

Pour l'algorithme de gestion du contact, on peut envisager d'autre type de lissage des normales qui permettraient de lisser les angles concaves et convexes et d'utiliser cette méthode pour d'autres procédés.

Enfin pour une plus grande précision encore, on peut envisager d'intégrer une méthode de multiplicateurs de Lagrange ou de Lagrangien augmenté.

Bibliographie

- [Abdalass87] Abdalass (E.). – *Résolution performante de problème de Stokes par mini-éléments, maillages auto adaptatifs et méthodes multigrilles. Applications.* – Thèse de doctorat, Ecole Centrale de Lyon, 1987.
- [Acr297] ACR2. – *Journées bilan deuxième année.* – 1997.
- [Arany96] Arany (I.). – The preconditioned conjugate gradient method with incomplete factorization preconditioners. *Computers Math. Applic.*, vol. 31, 1996, pp. 1 – 5.
- [Arnold et al.84] Arnold (D.N.), Brezzi (F.) et Fortin (M.). – A stable finite element for the Stokes equations. *Calcolo.*, vol. 21, 1984, pp. 337 – 344.
- [Babuska73] Babuska (I.). – The finite element method with penalty. *Math. Comp.*, vol. 27, 1973, pp. 221 – 228.
- [Balan96] Balan (T.). – *Optimisation de forme des outils de forgeage par méthode inverse.* – Thèse de doctorat, ENSMP, 1996.
- [Boersma et al.97] Boersma (A.) et Wriggers (P.). – An algebraic multigrid solver for finite element computations in solid mechanics. *Engineering Computations*, vol. 14, n2, 1997, pp. 202 – 215.
- [Bornemann et al.96] Bornemann (F. A.) et Deuffhard (P.). – The cascadic multigrid method for elliptic problems. *Numer. Math.*, vol. 75, 1996, pp. 135–152.
- [Boyre99] Boyère (E.). – *Contribution à la modélisation numérique thermo-mécanique tridimensionnelle du forgeage.* – Thèse de doctorat, ENSMP, 1999.
- [Braess86] Braess (D.). – On the combination of the multigrid method and conjugate gradients. *Multigrid Methods II*, éd. par Hackbusch (W.) et Trottenberg (U.). pp. 52–64. – Berlin, 1986.
- [Brandt et al.84] Brandt (A.), McCormick (S.) et Ruge (J.). – *Sparsity and its applications*, chap. Algebraic multigrid (AMG) for sparse matrix equations. – Cambridge University Press, 1984, d.j. evans (ed.) édition.
- [Brandt77] Brandt (A.). – Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, vol. 31, n138, 1977, pp. 333 – 390.
- [Carre95] Carré (G.). – *Simulation numérique d'écoulements turbulents compressibles et stationnaires par méthodes multigrilles.* – Thèse de doctorat, Université de Nice-Sophia Antipolis, 1995.
- [Cescotto et al.93] Cescotto (S.) et Charlier (R.). – Frictional contact finite elements based on the mixed variational principles. *International Journal*

- for *Numerical Methods in Engineering*, vol. 36, 1993, pp. 1681 – 1701.
- [Cescutti89] Cescutti (J.P.). – *Contribution à la simulation numérique du forgeage*. – Thèse de doctorat, ENSMP, Juin 1989.
- [Chabrant et al.98] Chabrant (P.), Dubois (F.) et Raous (M.). – Various numerical methods for solving unilateral contact with friction. *Math. Comp. Mod.*, vol. 28, 1998, pp. 97 – 108.
- [Chertier97] Chertier (O.). – *Contact et frottement entre solides déformables en grandes déformations*. – Thèse de doctorat, Institut de Mécanique de Marseille, 1997.
- [Coupez91] Coupez (T.). – *Grandes déformations incompressibles-remailage automatique*. – Thèse de doctorat, ENSMP, 1991.
- [Coupez94] Coupez (T.). – Maillage automatique et remailage. *Séminaire de plasticité. Éléments finis et mise en forme des matériaux. Sophia Antipolis, 26-30 Septembre, Chapitre 17, Tome 3*, 1994.
- [Coupez95] Coupez (T.). – Stable stabilized finite element for 3D forming calculation. – 1995. communication interne.
- [Davids et al.99] Davids (W.G.) et Turkiyyah (G.M.). – Multigrid preconditioner for unstructured nonlinear 3d fe models. *Journal of Engineering Mechanics*, vol. 125, n2, février 1999, pp. 186 – 196.
- [Davis et al.93] Davis (M. B.) et Carey (G. F.). – Iterative solution of the stream function vorticity equations using a multigrid solver with finite elements. *Comm. Numer. Meth. Engrg.*, vol. 9, 1993, pp. 587–594.
- [Dick et al.92] Dick (E.) et Linden (J.). – A multigrid method for steady incompressible Navier Stokes equations based on flux difference splitting. *Int. J. Numer. Meth. Fluids*, vol. 14, 1992, pp. 1311–1323.
- [Doghri et al.98] Doghri (I.), Muller (A.) et Taylor (R.L.). – A general three dimensional contact procedure for implicit finite element codes. *Engineering Computations*, vol. 15, n2, 1998, pp. 233 – 259.
- [Dorward et al.97] Dorward (S.E), Matheson (L.R.) et Tarjan (R.E). – Toward efficient multigrid preprocessing. *The International Journal of Supercomputer Applications and High Performance Computing*, vol. 11, n1, 1997, pp. 12 – 33.
- [Douglas et al.] Douglas (C.C.) et Douglass (M.B.). – Mnet bibliography. In URL <http://www.mgnet.org/bib/mgnet.bib>, Department of Mathematics and the Center for Computational Sciences, University of Kentucky, Lexington, KY, USA and Department of Computer Science, Yale University, New Haven, CT, USA. Last modified on September 4, 1998.
- [El hadj87] El Hadj (M.). – *Application des méthodes multi-grilles aux calculs des structures élastiques et visco-plastiques*. – Thèse de doctorat, ENSMP, 1987.
- [Elman96] Elman (H.C.). – Multigrid and krylov sspace methods for the discrete stockes equations. *International Journal for Numerical Methods in Fluids*, vol. 22, 1996, pp. 755 – 770.

- [Farhat et al.89] Farhat (C.) et N.Soh. – A coarse/fine mesh preconditionner ofr very ill-conditionned finite element problems. *Int. J. Numer. Methods in Engrg*, vol. 28, 1989.
- [Fedorenko62] Fedorenko (R.P.). – A relaxation method for solving elliptic difference equations. *U.S.S.R. Comput. Math. and Math. Phys.*, 1962.
- [Fish et al.95] Fish (J.), Belsky (V.) et Pandheeradi (M.). – Iterative and direct solvers for interface problems with lagrange multipliers. *Computing Systems in Engineering*, vol. 6, n3, 1995, pp. 261 – 273.
- [Fortin et al.92] Fortin (M.) et Glowinski (R.). – *Méthodes du lagrangien augmenté*. – Dunod, Paris, 1992, *Collection méthodes Mathématiques de l'Informatique*.
- [Fourment et al.99] Fourment (L.), Chenot (J.L.) et Mocellin (K.). – Numerical formulations and algorithms for solving contact problems en metal forming simulation. *INT. J. for Numerical Methods in Engineering*, 1999.
- [Gawrylczyk95] Gawrylczyk (K.M.). – Adaptative refinement and optimization of first order tetrahedral meshes for electrodynamics. *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 14, n4, 1995, pp. 27 – 31.
- [Guillard93] Guillard (H.). – *Node-nested multi-grid with Delaunay coarsening*. – Research Report n1898, INRIA, 1993.
- [Habraken et al.98] Habraken (A.M.) et Cescotto (S.). – Contact between deformable solids. the fully coupled approach. *Mathematical and Computer Modelling*, vol. 28, 1998.
- [Hattangady99] Hattangady (N.V.). – Coarsening of mesh models for representation of rigid objects in finite element anlysis. *International Journal For Numerical methods in En gineering*, vol. 44, 1999, pp. 3130–0326.
- [Hild98] Hild (P.). – *Problèmes de contact unilatéral et maillages éléments finis incompatibles*. – Thèse de doctorat, Université Paul Sabatier de Toulouse, 1998.
- [Husson75] Husson (J.). – *Pratique du forgeage*. – PYC Edition, 1975.
- [Jaouen98] Jaouen (O.). – *Modélisation tridimensionnelle par éléments finis pour l'analyse thermo-mécanique du refroidissement des pièces de fonderie*. – Thèse de doctorat, ENSMP, Décembre 1998.
- [Koobus94] Koobus (B.). – *Algorithmes multigrille et algorithmes implicites pour les écoulements compressibles turbulents*. – Thèse de doctorat, Université de Nice-Sophia Antipolis, 1994.
- [Lallemand92] Lallemand (M.H.). – Algebraic multigrid. *Multigrid methods for computational Fluid Dynamic*. ERCIM. – INRIA, Sophia Antipolis, France, 1992.
- [Laursen et al.93] Laursen (T.A.) et Simo (J.C.). – A continuum-based finite element formulation for the implicit solution of multibody, large deformation frictional contact problems. *International Journal for Numerical Methods in Engineering*, vol. 36, 1993, pp. 3451 – 3485.

- [Le dain91] Le Dain (M.A.). – *Méthodes multigrilles et H-adaptation: Application au calcul de structures élastiques bidimensionnelles*. – Thèse de doctorat, ENSMP, 1991.
- [Lee98] Lee (C.O.). – multigrid methods for the pure traction problem of linear elasticity: mixed formulation. *SIAM J. Numer. Anal.*, vol. 35, n1, 1998, pp. 121 – 145.
- [Lonsdale93] Lonsdale (R.D.). – An algebraic multigrid solver for the navier-stokes equations on unstructured meshes. *Int. J. Num. Meth. heat fluid flow*, vol. 3, 1993, pp. 3 – 14.
- [Mahajan et al.98] Mahajan (P.), Fourment (L.) et Chenot (J.L.). – Implicit scheme for contact analysis in non steady state forming. *Engineering Computations*, vol. 15, n7, 1998, pp. 908 – 924.
- [Mandel et al.88] Mandel (J.), McCormick (S.) et Ruge (J.). – An algebraic theory for multigrid methods for varational problems. *SIAM J. Numer. Anal.*, vol. 25, 1988, pp. 91 –110.
- [Marie97] Marie (S.). – *Un modèle de parallélisation S.P.M.D. pour la simulation numérique de procédés de mise en forme des matériaux*. – Thèse de doctorat, ENSMP, Mars 1997.
- [Mavriplis et al.90] Mavriplis (D.J.) et Jameson (A.). – A multigrid solution of Navier Stokes equations on triangular meshes. *AIAA*, vol. 28, n8, 1990, pp. 1415 –1425.
- [Mavriplis92] Mavriplis (D.J.). – Three-dimensional unstructured multigrid for the Euler equations. *AIAA J.*, vol. 30, 1992, pp. 1753–1761.
- [Mavriplis95] Mavriplis (D.J.). – Three dimensional multigrid Reynolds averaged Navier Stokes solver for unstructured meshes. *AIAA*, vol. 33, n3, 1995, pp. 445 –453.
- [Mavriplis97] Mavriplis (D.J.). – Unstructures grid techniques. *Annu. Rev. Fluid. Mech.*, vol. 29, 1997, pp. 473 – 514.
- [Miles95] Miles (M.). – *Calcul thermomécanique des outillages pour le forgeage à chaud*. – Thèse de doctorat, ENSMP, 1995.
- [Mitchell et al.98] Mitchell (J.A) et Reddy (J.N.). – A multilevel hierarchical preconditioner for thin elastic bodies. *International Journal for Numerical Methods in Engineering*, vol. 43, 1998, pp. 1383 – 1400.
- [Morano92] Morano (E.). – *Résolution des équations d'Euler par une méthode multigrille stationnaire*. – Thèse de doctorat, Université de Nice-Sophia-Antipolis, 1992.
- [Moukalled et al.91] Moukalled (F.) et Acharya (S.). – A local adaptive grid procedure for incompressible flows with multigridding and equidistribution concepts. *Int. J. Numer. Meth. Fluids*, vol. 13, 1991, pp. 1085–1111.
- [Muttin et al.95] Muttin (F.) et Chenot (J.L.). – On a conjugate two-grid method for three dimensional elasticity. *Engineering Computations*, vol. 12, 1995, pp. 3 – 20.
- [Oosterlee97] Oosterlee (C.W.). – A gmres-based plane smoother in multigrid to solve 3d anisotropic fluid flow prolems. *Journal of Computational Physics*, n130, 1997, pp. 41 – 53.

- [Papadrakakis et al.93] Papadrakakis (M.) et Bitoulas (N.). – Accuracy and effectiveness of preconditioned conjugate gradient algorithms for large and ill-conditioned problems. *Computers Methods in Applied Mechanics and Engineering*, vol. 109, 1993, pp. 219 – 232.
- [Pascoe et al.89] Pascoe (S.K.) et Mottershead (J.E.). – Two new finite element contact algorithm. *Computer and Structures*, vol. 32, n1, 1989, pp. 137 – 144.
- [Pavanachand et al.96] Pavanachand (CH.) et Krishnakumar (R.). – A new one-pass approach for large deformation multibody frictional contact analysis. *Communications In Numerical Methods in Engineering*, vol. 12, 1996, pp. 569 – 579.
- [Perchat et al.99] Perchat (E.), Fourment (L.) et Coupez (T.). – Mixed formulations and iterative solver for the parallelisation of a forging simulation by mesh partitioning. *Euroconference: Parallel and Distributed Computing for Computational Mechanics 1999*. – 1999.
- [Perng et al.91] Perng (C.Y.) et Street (R.L.). – A coupled multigrid-domain splitting technique for simulating incompressible flows in geometrically complex domains. *International Journal for Numerical Methods in Fluids*, vol. 13, 1991, pp. 269 – 286.
- [Pressburger et al.95] Pressburger (Y.) et Perucchio (R.). – A hierarchical two-level multigrid solver. *Computers and Structures*, vol. 55, n3, 1995, pp. 471 – 483.
- [Ramage et al.94] Ramage (A.) et Wathen (M. A.J.). – Iterative solution techniques for the Stokes and Navier Stokes equations. *International Journal For Numerical methods in fluids*, vol. 19, 1994, pp. 67 – 83.
- [Rodic et al.96] Rodic (T.) et Wensel (O.). – Direct and inverse analyses of piercing test. *Nuphymat'96 Workshop on Numerical and physical Study of Material Forming Process. Final Report of Human Capital and Mobility program*, pp. 29 – 34. – Sophia Antipolis, 1996.
- [Rodriguez villa97] Rodriguez-Villa (A.). – *Etude théorique et expérimentale de l'extrusion-soufflage de corps creux en polymère*. – Thèse de doctorat, ENSMP, 1997.
- [Rubart92] Rubart (L.). – A multigrid algorithm for mixed finite-element problems. *Communications in Applied Numerical Methods*, vol. 8, 1992, pp. 661 – 668.
- [Saad88] Saad (Y.). – Preconditioning techniques for nonsymmetric and indefinite linear systems. *Journal of Computational and Applied Mathematics*, vol. 24, 1988, pp. 89 – 105.
- [Sheng et al.95] Sheng (C.), Taylor (L. K.) et Whitfield (D. L.). – Multigrid algorithm for three dimensional incompressible high Reynolds number turbulent flows. *AIAA J.*, vol. 33, 1995, pp. 2073–2079.
- [Silvester et al.94] Silvester (D.) et Wathen (A.). – Fast iterative solution of stabilized Stokes systems, part i: using general block preconditioners. *Siam J. Numer. Anal.*, vol. 31, n5, 1994, pp. 1352 – 1367.
- [Simo et al.92] Simo (J.C.) et Laursen (T.A.). – An augmented lagrangian treatment of contact problems involving friction. *Computers & Structures*, vol. 42, n1, 1992, pp. 97 – 116.

- [Soyris90] Soyris (N.). – *Modélisation tridimensionnelle du couplage thermique en forgeage à chaud*. – Thèse de doctorat, ENSMP, 1990.
- [Surdon et al.87] Surdon (G.) et Chenot (J.L.). – Finite element calculation of three-dimensional hot forging. *International Journal for Numerical Methods in Engineering*, vol. 24, 1987, pp. 2107 – 2117.
- [Tzanos92] Tzanos (C.). – Higher order differencing method with a multigrid approach for the solution of the incompressible flow equations at high Reynolds number. *Numerical Heat Transfer, Part B*, vol. 22, 1992, pp. 179 – 198.
- [Van der Maarel92] Van Der Maarel (H.T.M.). – Adaptive multigrid for the steady Euler equations. *Comm. Appl. Num. Methods*, vol. 8, 1992, pp. 749–760.
- [Vasconcelos et al.98] Vasconcelos (P.B.) et D’Almeida (F.D.). – Preconditioned iterative methods for coupled discretizations of fluids flow problems. *IMA Journal of Numerical Analysis*, vol. 18, 1998, pp. 385 – 397.
- [Vincent97] Vincent (C.). – *Méthodes de gradient conjugué préconditionné et techniques multigrilles pour la résolution du problème de Stokes par éléments finis mixtes Q1-P0 stabilisés. Applications*. – Thèse de doctorat, Université de Provence, Centre Saint Charles, 1997.
- [Wathen et al.93] Wathen (A.) et Silvester (D.). – Fast iterative solution of stabilized Stokes systems, part I: using simple diagonal preconditioners. *Siam J. Numer. Anal.*, vol. 30, n3, 1993, pp. 630 – 649.
- [Wittum89] Wittum (G.). – Multi-grid methods for Stokes and Navier–Stokes equations—transforming smoothers: algorithms and numerical results. *Numer. Math.*, vol. 54, 1989, pp. 543–563.
- [Wriggers95] Wriggers (P.). – Finite element algorithms for contact problems. *Computational Methods in Engineering*, vol. 2, 1995, pp. 1 – 49.
- [Zang et al.95] Zang (Y.) et Street (R.L.). – A composite method for calculating unsteady incompressible flows in geometrically complex domains. *International Journal For Numerical Methods In Fluids*, vol. 20, 1995, pp. 341 – 361.
- [Zavarise et al.95] Zavarise (G.) et Wriggers et B.A. Schrefler (P.). – On augmented Lagrangian algorithms for thermomechanical contact problems with friction. *International Journal for Numerical Methods in Engineering*, 1995, pp. 2929 – 2949.

Contribution à la simulation numérique tridimensionnelle du forgeage à chaud : étude du contact et calcul multigrille.

Résumé :

Ce travail se situe dans le cadre de la modélisation numérique par éléments finis du procédé de forgeage à chaud des métaux. Il a pour but l'optimisation du logiciel de simulation numérique tridimensionnelle Forge 3. Il s'oriente autour de deux points bloquants pour une utilisation de ce logiciel : des problèmes de précision sur la surface de contact entre la pièce et l'outil et un temps de calcul élevé. Dans la première partie, on étudie le problème du contact unilatéral. En partant de la méthode de pénalisation existant, on améliore l'algorithme de contact par la mise en place d'un algorithme implicite. On valide ces développements sur plusieurs cas de forgeage où l'on montre l'apport du nouvel algorithme en terme de précision de gestion de la surface entre la pièce et l'outil. Dans la seconde partie, on traite du problème de temps de calcul. Malgré la mise en place d'une méthode de calcul parallèle les délais de calcul pour une simulation tridimensionnelle restent très élevés. Nos travaux ont donc consisté à remplacer le solveur de type résidu minimal implémenté dans Forge 3 par une méthode asymptotiquement optimale basée sur l'utilisation de multigrilles. Nous avons développé une méthode originale utilisant la méthode de résidu minimal comme lisseur et une méthode de raffinement de maillage pour la construction des différents niveaux. Nous avons optimisé cet algorithme en introduisant en particulier un préconditionneur utilisant une factorisation de Crout incomplète pour les résolutions grille grossière. Les résultats obtenus en bigrille sur différents cas de forgeage sont excellents et les quelques résultats sur un calcul avec trois niveaux de maillage sont très encourageants.

Contribution to 3D numerical simulation of hot forging process : Study of contact and multigrid computation.

Abstract:

This work is within the framework of numerical modeling by finite elements of the process of hot forging of metals. The purpose of it is optimization of the software Forge 3 of three-dimensional digital simulation. It is directed around two points blocking for a use of this software: problems of precision on the surface of contact between the part and the tool and a high calculating time. In the first part, we study the problem of the unilateral contact. On the basis of the method of penalization existing, we improve the algorithm of contact by the installation of an implicit algorithm. We validate these developments on several cases of forging where one shows the contribution of the new algorithm in term of precision on the management of surface between the part and the tool. In the second part, we deal with problem of calculating time. In spite of the installation of a method of parallel computation the time of calculation for a three-dimensional simulation remain very high. Our work thus consisted in replacing the solver of minimal residual type implemented in Forge 3 by a method asymptotically optimal based on the use of multigrid. We developed an original method using the method of minimal residual like smoother and a method of grid refinement for the construction of the various levels. We optimized this algorithm by introducing in particular a preconditioner using an incomplete Crout factorization for the resolutions on the coarse grid. The results obtained with two-grids on various cases of forging are excellent and the few results on a calculation with three levels of grid are very encouraging.