



HAL
open science

Déformations élastiques des presses de forgeage et calcul parallèle

Olga Karaseva

► **To cite this version:**

Olga Karaseva. Déformations élastiques des presses de forgeage et calcul parallèle. Sciences de l'ingénieur [physics]. École Nationale Supérieure des Mines de Paris, 2005. Français. NNT : 2005ENMP1307 . pastel-00001513

HAL Id: pastel-00001513

<https://pastel.hal.science/pastel-00001513>

Submitted on 16 Jan 2006

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.

Remerciements

Ce travail a été effectué au Centre de Mise en Forme des Matériaux (CEMEF) de l'École des Mines de Paris, dans le groupe de recherche Calcul Intensif en Mise en Forme de Matériaux (CIM). Je tiens à remercier la direction de l'École des Mines de Paris de m'avoir acceptée au sein de sa formation doctorale. Je suis aussi très reconnaissante à Jean-Loup Chenot de m'avoir accueillie au sein du CEMEF.

François Xavier Roux m'a fait l'honneur d'accepter de présider ce jury. Je le remercie vivement pour l'intérêt qu'il a porté à mon travail et pour ces conseils très précieux. Mes remerciements les plus vifs à Stéphane Lanteri et Jean-Claude Boyer qui ont accepté la lourde tâche de rapporteur.

Je tiens à exprimer ma profonde reconnaissance à mon directeur de thèse Lionel Fourment pour sa disponibilité, sa rigueur scientifique et sa patience exceptionnelle. Son aide m'a été très précieuse dans les moments de doutes. Merci !

Cette thèse a été financée dans le cadre du projet européen IMPRESS. Je remercie tous les partenaires de ce projet pour nos échanges scientifiques et l'intérêt qu'ils ont porté pour mon travail.

Je voudrais remercier Christine Béraudo de Transvalor S.A. qui n'a pas ménagé son temps pour m'aider avec Forge3[®] avec une grande gentillesse et bonne humeur. Merci à Hugues Dignonnet pour son aide. Merci à tous mes collègues qui ont contribué directement ou indirectement à ce travail : Mihaela, Serge, Julien, Delphine, Fred, Luisa, Cyril, Laurent, Héjin, Sorin, Olga, Tania, Julia (allez, courage !), Mehdi, Céline, Stéphane et beaucoup d'autres... pardon à tous ceux que j'oublie...

Une pensée très affectueuse à mes collègues de l'Université de Liège : Sophie, Martin, Antoine, Mike,... et à ceux de l'École Centrale de Nantes : Laure, Hans, Steven, Grégory, Gille, Frank, sans oublier Marianne !

Je pense aussi à mes amis de Moscou. Надя, Дима, Леша, Антон, Алена, Андрюша. Спасибо.

Enfin, je remercie mes parents qui m'ont aimée et soutenue pendant ces années. Спасибо. За поддержку и участие, за терпение и вашу любовь. Спасибо.

TABLES DES MATIÈRES

TABLES DES MATIÈRES	I
LISTE DES FIGURES.....	V
LISTE DES TABLEAUX.....	IX
INTRODUCTION.....	1
CHAPITRE I. PRISE EN COMPTE DE LA RAIDEUR DES PRESSES DE FORGEAGE	5
I.1. MODÈLES DE RAIDEUR DES PRESSES DE FORGEAGE	5
I.1.1. Approche actuelle standardisée	7
I.1.2. Approche du projet NETTFORM.....	7
I.1.3. Approche du projet IMPRESS	9
I.1.4. Outils flottants dans Forge3®	10
I.2. FORMULATION DU PROBLÈME MÉCANIQUE	11
I.2.1. Équations d'équilibre et de conservation de la masse	11
I.2.2. Lois rhéologiques	12
I.2.3. Conditions aux limites mécaniques	14
I.2.4. Formulation faible	18
I.3. FORMULATION DU PROBLÈME DISCRET	19
I.3.1. Discrétisation temporelle et contact incrémental.....	19
I.3.2. Discrétisation spatiale.....	20
I.3.3. Modèle de presse	23
I.3.4. Linéarisation des équations discrètes.....	25
I.3.5. Implémentation dans Forge3®	27
I.4. VALIDATION PAR RAPPORT AU CAS ANALYTIQUE	30
I.5. CONCLUSION	32
CHAPITRE II. FORGE3® PARALLÈLE.....	33
II.1. STRATÉGIE S.P.M.D.	33
II.2. PARTITIONNEMENT	35
II.2.1. Partitionnement du maillage.....	35

II.2.2. Partitionnement des vecteurs et de la matrice	37
II.3. MÉTHODE DE RÉSIDU MINIMAL	40
II.4. PARTICULARITÉS PARALLÈLES	42
II.4.1. Produits matrice-vecteur	42
II.4.2. Produits scalaires.....	43
II.4.3. Calculs des normes infinies des vecteurs	44
II.4.4. Algorithme parallèle.....	44
II.4.5. Prise en compte de la raideur de presse.....	45
CHAPITRE III. MÉTHODES DE DÉCOMPOSITION DE DOMAINE.....	47
III.1. PRINCIPE DE BASE.....	47
III.2. MÉTHODE DU COMPLÉMENT DE SCHUR	49
III.3. MÉTHODE FETI.....	51
III.4. MÉTHODE PRIMALE-DUALE.....	53
III.5. PRECONDITIONNEMENT.....	56
III.5.1. Méthode du complément de Schur	56
III.5.2. Méthode FETI.....	58
III.5.3. Méthode primale-duale	59
III.6. PROBLÈME DES MODES RIGIDES	60
III.6.1. Extraction des modes rigides	61
III.6.2. Problème hybride aux interfaces.....	62
III.6.3. Méthode itérative projetée	63
III.6.4. Méthode primale-duale	65
III.7. SOLUTION DU PROBLÈME CONDENSÉ AUX INTERFACES.....	66
III.7.1. Méthode OrthoDir préconditionnée.....	67
III.7.2. Méthode OrthoDir projetée.....	68
III.7.3. Méthode OrthoDir projetée de la méthode primale-duale	71
III.8. CONCLUSION	73
CHAPITRE IV. MÉTHODES DE DÉCOMPOSITION DE DOMAINE : TESTS.....	75
IV.1. ÉCRASEMENT D'UNE BARRE LONGUE	75
IV.2. CAS DU TRIAXE	82
IV.2.1. Calcul sur 2 sous-domaines	83
IV.2.2. Calcul sur 3 sous-domaines	84
IV.3. UNE POUTRE ENCASTRÉE	86
IV.4. CONCLUSION	88
CHAPITRE V. RÉSULTATS D'APPLICATIONS	91
V.1. PRISE EN COMPTE DE LA RAIDEUR DE PRESSE	91

V.1.1. Cas Vaccari	91
V.1.2. Cas Smetek.....	93
V.1.3. Cas Iskra Avtoelektrika.....	97
V.2. FORGE3 PARALLÈLE	99
V.2.1. Cas Vaccari	99
V.2.2. Cas Iskra Avtoelektrika.....	101
V.3. CAS DU TRIAXE : CONFRONTATION FORGE3 VS. FETI	103
V.3.1. Cas à 10 602 nœuds	104
V.3.2. Cas à 58 626 nœuds	109
V.4. CONCLUSION	111
CONCLUSIONS ET PERSPECTIVES	113
BIBLIOGRAPHIE	117
ANNEXE A.....	123
ANNEXE B.....	127

LISTE DES FIGURES

Figure I.1. Presse de forgeage avec des conditions de chargement décentrées.....	6
Figure I.2. Diagramme typique charge-déformation.....	6
Figure I.3. Repère (x, y, z) pour une presse.....	8
Figure I.4. Exemples des relations entre « action » et « réaction ».....	9
Figure I.5. Mouvement du corps rigide.....	11
Figure I.6. Description géométrique de la zone de contact entre les deux corps.....	16
Figure I.7. Approximation de la surface de l'outil rigide pour le contact explicite.....	20
Figure I.8. Élément fini mixte P1+/P1.....	21
Figure I.9. Modèle de raideur de presse appliqué aux outils.....	24
Figure I.10. Création des éléments fictifs.....	27
Figure I.11. Forgeage d'un cube : cas test analytique.....	31
Figure I.12. Forgeage d'un cube : validation par rapport au cas analytique.....	31
Figure II.1. Stratégie S.P.M.D. de parallélisation de Forge3.....	34
Figure II.2. Analyse de contact effectuée indépendamment par chaque processeur avec l'ensemble de l'outillage (ici deux simples tas plats).....	34
Figure II.3. Analyse de contact : version multi-coprs.....	35
Figure II.4. Partition d'un maillage par éléments et en deux sous-domaines.....	36
Figure II.5. Minimisation des communications par changement local de couleurs et équilibrage des charges.....	37
Figure II.6. Contributions de deux sous-domaines dans la matrice globale.....	38
Figure II.7. Renumerotation des nœuds en intérieurs et ceux d'interface.....	39
Figure III.1. Décomposition du domaine Ω en deux sous-domaines Ω_1 et Ω_2	48
Figure III.2. Méthode primale : description de l'interface.....	49
Figure III.3. Méthode FETI : <i>tearing and interconnecting</i>	51
Figure III.4. Méthode FETI : opérateurs de trace B_i	52
Figure III.5. Sous-domaines flottants : modes rigides.....	61
Figure IV.1. Écrasement d'une barre longue : partition en 2, 3, 4 et 5 sous-domaines, non nécessairement connexes.....	75

Figure IV.2. Écrasement d'une barre : nombre d'itérations nécessaires pour la résolution du système linéaire par les trois méthodes utilisées sans préconditionnement, précision de 10^{-9}	78
Figure IV.3. Écrasement d'une barre : temps de résolution du système linéaire par les trois méthodes utilisées sans préconditionnement, précision de 10^{-9}	78
Figure IV.4. Écrasement d'une barre : temps pour une itération de la résolution du système linéaire par les trois méthodes utilisées sans préconditionnement, précision de 10^{-9}	79
Figure IV.5. Écrasement d'une barre : nombre d'itération nécessaire pour la résolution du système linéaire par les trois méthodes utilisées sans préconditionnement, précision de 10^{-6}	79
Figure IV.6. Écrasement d'une barre : temps de résolution du système linéaire par les trois méthodes utilisées sans préconditionnement, précision de 10^{-6}	80
Figure IV.7. Écrasement d'une barre : temps pour une itération de la résolution par les trois méthodes utilisées sans préconditionnement, précision 10^{-6}	80
Figure IV.8. Écrasement d'une barre : nombre d'itérations pour la résolution du système linéaire sans et avec préconditionnement, précision de 10^{-6}	81
Figure IV.9. Écrasement d'une barre : temps de résolution du système linéaire sans et avec préconditionnement, précision de 10^{-6}	81
Figure IV.10. Écrasement d'une barre : temps pour une itération sans et avec préconditionnement, précision de 10^{-6}	82
Figure IV.11. Cas du triaxe : calcul sur deux sous-domaines	83
Figure IV.12. Cas du triaxe : calcul sur trois sous-domaines au milieu de procédé.....	85
Figure IV.13. Partition en trois sous-domaines d'une poutre encastree	87
Figure IV.14. Barre encastree : courbes de convergence pour les trois méthodes.	88
Figure V.1. Cas Vaccari : géométrie initiale (a) et forme finale de la pièce forgée obtenue sans prise en compte de la raideur de presse (b)	92
Figure V.2. Cas Vaccari : vitesse dans la direction de l'axe z à la fin de la simulation	93
Figure V.3. Cas Smetek : début de la simulation, isovaleurs de contact.....	95
Figure V.4. Cas Smetek : images des pièces expérimentalement forgées et isovaleurs de déformation équivalente à la fin du procédé	95
Figure V.5. Cas Smetek : valeurs expérimentales et numériques du déplacement selon l'axe x	96
Figure V.6. Cas Smetek : valeurs expérimentales et numériques du déplacement de la presse correspondant à la rotation autour de l'axe y	96
Figure V.7. Cas Iskra : géométries initiales et maillages des corps	97
Figure V.8. Cas Iskra : fin de la simulation.....	98
Figure V.9. Cas Iskra : déplacement selon l'axe z dû aux déformation de la presse.....	99
Figure V.10. Cas Vaccari : partition en 2, 4 et 8 sous-domaines	100
Figure V.11. Cas Iskra : accélérations obtenues sur le cluster en fonction du nombre de processeurs et pour différentes tailles de maillage.....	102
Figure V.12. Cas Iskra : efficacité pour les trois maillages en fonction du nombre de processeurs ...	102
Figure V.13. Cas Iskra : calcul sur deux processeurs.....	103
Figure V.14. Cas du triaxe à 10 602 nœuds : temps de résolution en fonction du nombre de processeurs	104

Figure V.15. Cas du triaxe à 10 602 nœuds : nombre d'itérations (a) et temps nécessaire pour une itération du solveur considéré (b)	105
Figure V.16. Méthode FETI : temps de factorisation, de résolution itérative et temps total	105
Figure V.17. Cas du triaxe à 10 602 nœuds : accélération (a) et efficacité (b)	106
Figure V.18. Cas du triaxe à 10 602 nœuds : accélération (a) et efficacité (b) calculées avec comme référence le temps de calcul sur 2 processeurs	107
Figure V.19. Cas du triaxe à 58 628 nœuds : temps de résolution	109
Figure V.20. Cas du triaxe à 58 628 nœuds : différentes partitions construites mais impropres aux calculs au-delà de 10 sous-domaines	110

LISTE DES TABLEAUX

Tableau IV.1. Écrasement d'une barre longue : paramètres de la rhéologie.....	76
Tableau IV.2. Écrasement d'une barre longue : dimensions des maillages	76
Tableau IV.3. Cas du triaxe : paramètres de la rhéologie et du frottement	82
Tableau IV.4. Cas du triaxe : dimensions de la partition en deux sous-domaines	83
Tableau IV.5. Cas du triaxe : résultats des calculs sur deux sous-domaines.....	84
Tableau IV.6. Cas du triaxe : dimensions de partition en trois sous-domaines.....	85
Tableau IV.7. Cas du triaxe : résultats des calculs sur trois sous-domaines	86
Tableau IV.8. Cas d'une barre encastree : dimensions de partition en trois sous-domaines.....	86
Tableau IV.9. Cas d'une barre encastree : résultats pour les trois méthodes avec les différents préconditionneurs.	87
Tableau V.1. Cas Vaccari : paramètres de rhéologie	91
Tableau V.2 Cas Smetek : dimensions des maillages au début des calculs	93
Tableau V.3. Cas Smetek : paramètres de la rhéologie.....	94
Tableau V.4. Cas Iskra : dimensions des maillages	97
Tableau V.5. Cas Iskra : paramètres de rhéologie.....	97
Tableau V.6. Cas Vaccari : accélérations obtenues sur le cluster de PC.....	101
Tableau V.7. Cas Iskra mono-corps : accélérations obtenues sur le cluster.....	101
Tableau V.8. Cas Iskra parallèle : dimensions des maillages.....	102
Tableau V.9. Méthode FETI : diminution de précision.....	108
Tableau V.10. Méthode FETI sur 8 processeurs avec différents préconditionneurs.....	109
Tableau V.11. Forge3 et PETSc : accélérations obtenues sur le cluster par référence au calcul sur 2 processeurs.....	111
Tableau V.12. Cas du triaxe à 58 628 nœuds : accélérations obtenues sur le cluster par référence au calcul sur 6 processeurs	111
Tableau A.1. Écrasement d'une barre longue : résultats des calculs sans préconditionnement, précision 10^{-9}	123
Tableau A.2. Écrasement d'une barre longue : résultats des calculs sans préconditionnement, précision 10^{-6}	124

Tableau A.3. Écrasement d'une barre longue : résultats des calculs avec les préconditionneurs allégés, précision 10^{-6}	125
Tableau B.1. Cas du triaxe à 10 602 nœuds : résultats des calculs	127
Tableau B.2. Cas du triaxe à 10 602 nœuds : accélérations obtenues sur le cluster	128
Tableau B.3. Cas du triaxe à 10 602 nœuds : efficacité pour les trois solveurs	128
Tableau B.4. Cas du triaxe à 10 602 nœuds : accélérations obtenues sur le cluster avec le temps de résolution sur 2 processeurs comme référence	128
Tableau B.5. Cas du triaxe à 10 602 nœuds : efficacité pour les trois solveurs avec le temps de résolution sur 2 processeurs comme référence	128
Tableau B.6. Cas du triaxe 58 628 nœuds : résultats des calculs	129

INTRODUCTION

La simulation numérique en mécanique est un outil essentiel pour le développement technologique de disciplines très variées telles que le génie civil, l'automobile, l'aérospatial. Le nombre d'analyses effectuées par des ingénieurs dans le monde entier est estimé être de l'ordre d'un million par jour*. Les avancées informatiques offrent des moyens de calcul de plus en plus puissants permettant ainsi de complexifier les modèles utilisés et d'approcher davantage l'expérience lors des simulations. Cet effort constant à améliorer des modèles retenus pour décrire le phénomène considéré conduit à la résolution des équations non-linéaires avec un très grand nombre d'inconnues. En conséquence, les temps de calcul deviennent de plus en plus importants. Un autre problème se pose alors qui est de réduire ce temps de calcul tout en gardant la précision des résultats recherchés.

Ce travail de thèse s'inscrit dans le cadre du projet européen IMPRESS. Ce projet a réuni des laboratoires de recherche danois IPU, italien DIMEG, polonais IPPT et le CEMEF, des forgerons danois BUL et slovène ISKRA Avtoelektrika, des fabricants de presses de forgeage italien VACCARI et danois SMETEK, des sociétés de service slovène C3M et française TRANSVALOR. L'objectif principal de cette collaboration était de mieux comprendre les paramètres qui influencent la précision du forgeage à froid et tout particulièrement l'impact de la raideur des presses de forgeage sur la précision des pièces forgées. Les résultats en écoulement des matériaux et en efforts que l'on peut obtenir par simulation numérique ne sont parfois pas suffisamment précis. L'utilisation des outils rigides pour la simulation de forgeage à froid peut en être la cause. En effet, les déformations élastiques dans les outils (matrices, presse, etc.) peuvent ne pas être négligeables vis-à-vis de la tolérance exigée sur la pièce finale. De ce fait, une approche multi-matériaux est nécessaire pour simuler les déformations des matrices. Par contre, on préfère utiliser un modèle de ressort pour la déformation du bâti de presse, ce qui évite le maillage de la presse et donc d'accroître considérablement le nombre de degrés de liberté.

Ce travail a été effectué sur la base du logiciel Forge3[®], développé au CEMEF et commercialisé par TRANSVALOR. Ce logiciel utilise la méthode des éléments finis et est destiné aux simulations

* Tom Hughes, IMA workshop on compatible discretizations, May 2004

des procédés de mise en forme des métaux. Forge3[®] permet de traiter des outils libres en translation [Boyère 1999] et en rotation [Traoré 2001]. Ce sont les outils pilotés à force imposée et/ou à couple imposé. La dernière version 7 de ce logiciel donne la possibilité de traiter les problèmes où les outils sont considérés comme déformables [Barboza 2004, Terzolo 2004]. Le travail présent est une généralisation des outils flottants au cas où la force et le couple changent selon une loi de type élastique. Cela rend alors possible de mener des calculs sur l'ensemble du système pièce-matrice-presse.

Lors de cette première étude, nous avons constaté que le temps de calcul nécessaire à la simulation du forgeage de composants industriels devient très rapidement un frein à l'utilisation des méthodes numériques. Dans ce contexte, une solution consiste à utiliser le calcul parallèle pour réduire le temps de calcul tout en gardant la précision souhaitable. La version parallèle de Forge3[®] est basée sur une stratégie S.P.M.D.* et la résolution globale du système linéaire [Marie 1997] permettant de rendre parallèle la méthode itérative de résolution. Le couplage des solutions entre les sous-domaines a lieu lors de cette résolution. Cette méthode n'engendre pas de résolutions de sous-problèmes locaux, associés aux sous-domaines, ainsi le problème à résoudre en séquentiel reste le même en parallèle, quelque soit le nombre de processeurs. Afin d'essayer d'améliorer la version parallèle de Forge3[®], nous nous sommes intéressés aux méthodes de décomposition de domaine, inédites au CEMEF. L'idée commune à ces méthodes repose sur le découpage du domaine physique de manière à travailler avec des problèmes locaux. Pour coupler ces problèmes résolus sur les sous-domaines, la méthode recherche itérativement les conditions aux limites à imposer aux frontières de ces sous-domaines. Dans un premier temps, nous nous sommes intéressés à la méthode FETI, à partir de la version fournie par l'Onera. À la suite de cette étude et dans le cadre de ce code, nous avons développé deux autres méthodes : la méthode classique du complément de Schur et une méthode hybride, a priori mieux adaptée à la formulation mixte du problème mécanique utilisée dans Forge3[®]. Nous nous sommes aussi intéressés à l'étude des préconditionneurs indispensables à ces méthodes. Le préconditionneur de Neumann a été implémenté pour la méthode du complément de Schur. Pour la méthode hybride, un préconditionneur simple et un autre plus complexe ont été élaborés. Une attention spéciale a été portée au traitement des modes rigides.

La suite de ce texte est organisée en cinq chapitres. Dans le *premier chapitre*, nous présenterons la formulation mathématique du problème mécanique de forgeage et les principales équations de sa discrétisation par la méthode des éléments finis. En même temps, nous introduirons le modèle de déformation élastiques des presses de forgeage et montrerons comment il a été intégré dans le logiciel Forge3[®]. Un cas test dont la solution analytique est connue validera l'implémentation de ce modèle. Le *deuxième chapitre* s'attachera à présenter la version parallèle de Forge3[®]. On verra aussi les

* « Single Program on Multiple Data » ou « un Seul Programme sur de Multiples Données »

méthodes de résolution du problème de forgeage exposé dans le premier chapitre. Le *troisième chapitre* sera dédié aux méthodes de décomposition de domaine. On y verra les deux méthodes classiques : la méthode du complément de Schur et la méthode FETI, ainsi qu'une nouvelle méthode hybride. Les questions de préconditionnement, traitement des modes rigides et résolution des problèmes d'interface seront abordées. Les trois méthodes développées dans le troisième chapitre seront étudiées dans le *quatrième chapitre* dans une série des tests assez simples, mais néanmoins représentatifs par rapport au problématique de Forge3[®]. Dans le *cinquième chapitre* nous présenterons les applications industrielles. Le modèle de raideur de presse sera validé par rapport aux données expérimentales. Nous verrons aussi l'influence que ce modèle peut avoir dans les simulations. Enfin, les résultats des tests effectués sur le solveur FETI seront présentés.

Chapitre I. PRISE EN COMPTE DE LA RAIDEUR DES PRESSES DE FORGEAGE

I.1. MODÈLES DE RAIDEUR DES PRESSES DE FORGEAGE

Le problème lié à la raideur des presses de forgeage est schématisé sur la Figure I.1, qui montre une presse avec une pièce à forger et l'ensemble des outillages. Une déviation quelconque des conditions symétriques de chargement conduit à l'apparition de forces décentrées dans la presse qui sont à l'origine de forces latérales, de flexion et souvent de torsions. Elles conduisent à différents déplacements des outils de forgeage difficiles à contrôler et par conséquent à une diminution de la précision de forgeage et du respect des tolérances. En outre, des effets nuisibles à la durée de vie de la presse et de l'outillage peuvent se manifester. Des études sur différentes machines industrielles de mise en forme des métaux montrent que le déplacement horizontal et la rotation de la table de la presse peuvent respectivement atteindre des valeurs de 5mm et 6mm/m [Doege 1990]. En général, ces problèmes sont présents dans les procédés avec des outils non-symétriques complexes, ou lors du forgeage de plusieurs opérations en un seul coup de presse, ou pour un positionnement non symétrique de la pièce dans les outils, et causent tout naturellement des complications dans la production des pièces à haute précision, tout particulièrement en forge à froid où les efforts sont très importants et la précision recherchée très grande.

Même si plusieurs logiciels de conception et de calcul sont maintenant à la disposition des ingénieurs pour la préparation et la mise en projet des outils de forgeage à froid, la qualité et les tolérances des composantes dépendent toujours de facteurs importants qui sont les conditions de frottement, l'écoulement de la matière, la température, les déformations de la machine et des outils. On observe que les déformations de la presse peuvent être tout à fait significatives vis-à-vis des tolérances voulues [Bockel 1980].

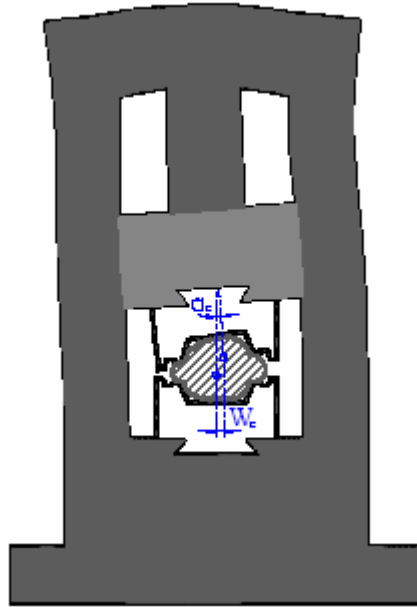


Figure I.1. Presse de forgeage avec des conditions de chargement décentrées

Un chargement non centré appliqué à la presse cause une déformation qui est constituée de trois parties (cf. Figure I.2) : jeux, déformation due au contact et déformation élastique du bâti. La déformation due à la fermeture des jeux entre les différents éléments de la presse peut se traduire par une valeur relativement grande, même si la charge appliquée est faible (zone I). La déformation due à la mise en contact est identifiée à la transition non-linéaire de la courbe (zone II). Enfin, quand toutes les pièces de la presse sont en contact, la raideur de la presse est identifiée à la pente de la courbe (zone III). Le bâti subit alors des déformations élastiques.

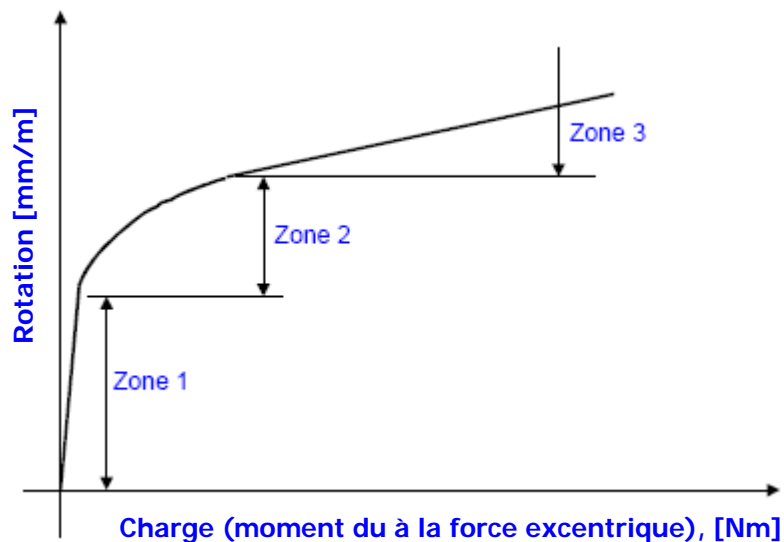


Figure I.2. Diagramme typique charge-déformation

I.1.1. Approche actuelle standardisée

La raideur des presses de forgeage a été l'objet d'études pendant quelques décennies. La première norme établie en 1942 décrit comment estimer la précision géométrique sur une presse non chargée. La norme actuelle DN 55189 décrit la courbe charge/déformation, ainsi que les méthodes de chargement et de mesure des différents efforts sur les presses mécaniques et hydrauliques. Elle introduit aussi les trois définitions suivantes :

- raideur linéaire selon l'axe principal de la presse (l'axe z) :

$$k_{ZZ} = \frac{\Delta F_z}{\Delta z} \quad (I.1)$$

- deux raideurs angulaires autour de l'axe x et y respectivement :

$$k_{AA} = \frac{\Delta F_z L_y}{\Delta A} \text{ et } k_{BB} = \frac{\Delta F_z L_x}{\Delta B} \quad (I.2)$$

où ΔF_z est un incrément de force verticale, Δz désigne la déformation élastique verticale de la presse correspondant à la force ΔF_z ; L_x et L_y sont les distances du centre de la table au point d'application de la force F_z selon les axes x et y respectivement, ΔA et ΔB désignent quant à eux les déformations angulaires autour des axes x et y .

Connaissant k_{ZZ} , il est facile de calculer la déformation verticale Δz pour toute valeur de ΔF_z . En revanche, des couples interviennent dans les équations (I.2), c'est à dire que le point d'application de la force modifie la valeur de ces couples. Ainsi, si les raideurs angulaires k_{AA} et k_{BB} sont connues, les angles ΔA et ΔB sont obtenus à condition que les déformations élastiques de la table et du coulisseau soient négligeables.

I.1.2. Approche du projet NETTFORM

Dans les équations (I.1) et (I.2), seule la composante verticale de la force de forgeage est considérée. Or, dans certains cas, comme par exemple pour le forgeage des aubes de turbine, la composante latérale de la force est toute aussi significative. Une meilleure modélisation du comportement de la presse a été proposée au cours du projet NETTFORM avec l'introduction de la matrice de flexibilité [Chodnikiewicz et al 1994]. L'objectif de ce projet était de définir l'élasticité de la presse due à une charge latérale et de proposer des méthodes appropriées pour en mesurer les coefficients élastiques. Le modèle proposé est celui de la matrice de flexibilité ou de raideur (voir équations et (I.3) et (I.4)). Les coefficients de cette matrice ont une interprétation physique très simple. Par exemple, l'élément c_{xz} définit la déformation du coulisseau dans la direction de l'axe x en

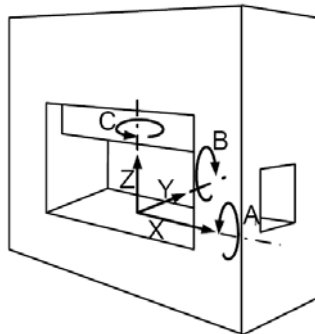
réaction à une force unitaire $F_z = 1$ appliquée dans la direction de l'axe z . Cette matrice est ainsi définie par l'équation :

$$\{R\} = [FM]\{A\} \quad (I.3)$$

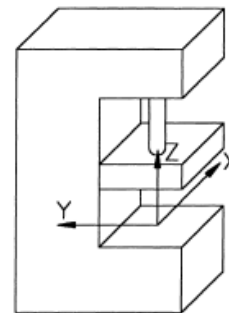
où $\{R\} = \{\Delta x, \Delta y, \Delta z, \Delta A, \Delta B, \Delta C\}^T$ est le vecteur des « réactions » composé des déformations linéaires et angulaires, $\{A\} = \{F_x, F_y, F_z, M_A, M_B, M_C\}^T$ est le vecteur des « actions » constitué des forces et des couples, et $[FM]$ est la matrice de flexibilité de taille 6×6 dans le cas général. La structure de cette matrice et la forme du vecteur $\{R\}$ dépendent du type de presse et de sa symétrie. Ainsi, peut-on considérer différents types de presses.

Pour une presse hydraulique, par exemple, le coulisseau n'atteint jamais la butée de fin de course de sorte qu'il y a toujours une pression de l'huile appliquée dans la direction z et donc la déformation du bâti de presse n'intervient pas. Par conséquent, il n'est pas possible de quantifier la déformation dans la direction de l'axe z (cf. l'orientation des repères sur la Figure I.3). De ce fait, on ne tient pas compte de cette composante verticale du vecteur $\{R\}$ qui prend la forme $\{\Delta x, \Delta y, \Delta A, \Delta B, \Delta C\}^T$ et donc la troisième ligne de la matrice correspondant à cet axe est supprimée. Par ailleurs, pour une presse ayant un bâti en forme de O (Figure I.3a) dû à la présence de deux plans de symétrie (yz et xz) la troisième colonne de la matrice est éliminée, elle aussi. En revanche, les presses hydrauliques à bâti en forme de C (Figure I.3b) ont un seul plan de symétrie, yz , et par conséquent la troisième colonne ne peut pas être éliminée. Pour ces deux cas, les matrices de flexibilité prennent respectivement la forme suivante :

$$[FM] = \begin{bmatrix} c_{XX} & 0 & 0 & c_{XB} & 0 \\ 0 & c_{YY} & c_{YA} & 0 & 0 \\ 0 & c_{AY} & c_{AA} & 0 & 0 \\ c_{BX} & 0 & 0 & c_{BB} & 0 \\ 0 & 0 & 0 & 0 & c_{CC} \end{bmatrix} \quad \text{et} \quad [FM] = \begin{bmatrix} c_{XX} & 0 & 0 & 0 & c_{XB} & c_{XC} \\ 0 & c_{YY} & 0 & c_{YA} & 0 & 0 \\ 0 & c_{AY} & c_{AZ} & c_{AA} & 0 & 0 \\ c_{BX} & 0 & 0 & 0 & c_{BB} & 0 \\ c_{CZ} & 0 & 0 & 0 & 0 & c_{CC} \end{bmatrix} \quad (I.4)$$



(a) bâti en forme O



(b) bâti en forme C

Figure I.3. Repère (x, y, z) pour une presse

I.1.3. Approche du projet IMPRESS

Le modèle de la matrice de flexibilité représente l'inconvénient de ne pas prendre en compte les effets non-linéaires des zones I et II (cf. Figure I.2). Ce concept est élargi par l'introduction des modèles suivants :

1. matrice de flexibilité proportionnelle (PFM) : elle introduit une relation linéaire entre l'action et la réaction, comme illustré sur la Figure I.4a ; c'est le modèle de base du projet NETTFORM ;
2. matrice de flexibilité généralisée (GeFM) : ce modèle prend en compte d'une manière très simple le début non-linéaire du diagramme (Figure I.4b), c'est à dire la fermeture des jeux entre les éléments de la presse ;
3. matrice de flexibilité non-linéaire (NLFM) : elle permet de décrire les relations entre les forces et les déformations d'une façon plus précise et plus fine (Figure I.4c). L'aspect non-linéaire des coefficients de cette matrice peut être représenté par une courbe point à point permettant une description précise du comportement de la presse.
4. matrice de flexibilité graphique (GrFM) : elle est définie individuellement pour chaque presse en tenant compte des asymétries et de l'usure. En effet, les trois matrices précédentes ont été établies sur la base de certaines hypothèses concernant le type et la symétrie des presses permettant à chaque fois d'annuler certains coefficients de la matrice. Cependant, plusieurs facteurs peuvent remettre en question ces hypothèses, comme cela a été souligné par [Chodnikiewicz et al 2000] : réparations effectuées sur la presse, usure, mauvais ajustement du coulisseau, etc. peuvent annuler une hypothèse de symétrie. Pour ces presses, la description de leurs caractéristiques peut être faite en utilisant des graphes correspondant à chacun des 36 éléments.

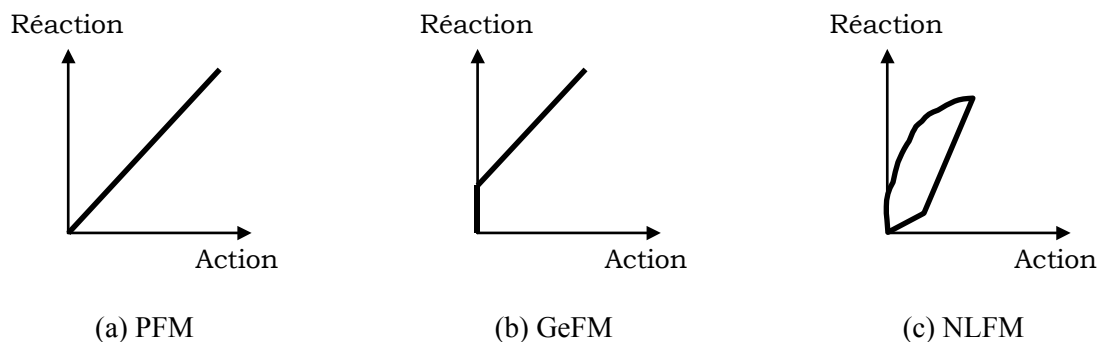


Figure I.4. Exemples des relations entre « action » et « réaction »

Chacun de ces quatre modèles décrit avec une précision croissante la raideur de la presse et implique une étude de plus en plus approfondie du comportement. Il faut souligner que le premier niveau d'approximation, la matrice de flexibilité proportionnelle, représente déjà une amélioration considérable de la description du comportement des presses.

I.1.4. Outils flottants dans Forge3®

Le plus souvent, les matrices de forgeage sont pilotées par une presse considérée suffisamment rigide pour que leur cinématique soit supposée connue à l'avance. Or, certains procédés, comme par exemple le laminage circulaire, nécessitent l'utilisation d'outils flottants qui bougent librement [Boyère 1999] et [Traoré 2001]. C'est l'écoulement de la matière et les forces de contact qui déterminent le mouvement de ces outils flottants.

Dans ce cas, la vitesse $v^fl(M)$ de tout point M de cet outil est définie comme

$$v^fl(M) = v^O + (\omega \wedge OM)$$

$$\begin{Bmatrix} v_1^{fl} \\ v_2^{fl} \\ v_3^{fl} \end{Bmatrix} = \begin{Bmatrix} v_1^O \\ v_2^O \\ v_3^O \end{Bmatrix} + \begin{Bmatrix} OM_2\omega_3 - OM_3\omega_2 \\ OM_1\omega_3 - OM_3\omega_1 \\ OM_1\omega_2 - OM_2\omega_1 \end{Bmatrix} \quad (I.5)$$

avec $v^fl(M)$ la vitesse du point M , v_i^O la vitesse de translation du centre O de gravité de l'outil, ω_i les vitesses angulaires autour des trois axes (cf. Figure I.5).

On remarque que le même phénomène est présent si l'on considère que le bâti de presse subit des déformations élastiques. En effet, comme on l'a vu dans les sections précédentes, des forces décentrées peuvent aboutir à une modification de la cinématique des outils (cf. Figure I.1). Pour prendre en compte la raideur de presse, on choisit l'approche de la matrice de flexibilité (cf. Figure I.4a) que l'on écrit sous la forme inversée :

$$\begin{Bmatrix} F^{pr} \\ C^{pr} \end{Bmatrix} = \begin{bmatrix} k^{xx} & k^{x\theta} \\ k^{\theta x} & k^{\theta\theta} \end{bmatrix} \begin{Bmatrix} \Delta x \\ \Delta \theta \end{Bmatrix} \quad (I.6)$$

avec F^{pr} le vecteur tridimensionnel des forces, C^{pr} le vecteur des couples, K la matrice de raideur de presse, Δx et $\Delta \theta$ les déplacements linéaires et angulaires, respectivement. En considérant la presse comme un corps solide, on associe le modèle de raideur (I.6) à un des outils rigides de sorte que la vitesse $v^{pr}(M)$ de tout point M de cet outil soit régie par la même équation (I.5). On développera la relation entre les déplacements et les vitesses dans la troisième section de ce chapitre.

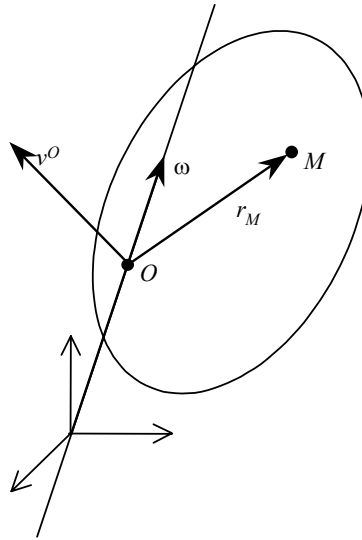


Figure I.5. Mouvement du corps rigide

I.2. FORMULATION DU PROBLÈME MÉCANIQUE

I.2.1. Équations d'équilibre et de conservation de la masse

A l'instant t , l'équation de conservation de la quantité de mouvement s'écrit comme :

$$\operatorname{div}(\sigma) + \rho f = \rho \frac{dv}{dt} \quad (\text{I.7})$$

ou encore avec la notation d'Einstein de sommation des indices répétés :

$$\frac{\partial \sigma_{ij}}{\partial x_j} + \rho f_i = \rho \frac{dv_i}{dt}, \quad \forall x \in \Omega, \quad i = 1, 2, 3 \quad (\text{I.8})$$

où σ_{ij} représente le tenseur des contraintes de Cauchy, ρ désigne la masse volumique, f_i sont les forces massiques, et v_i est le champ des vitesses.

En forgeage, on peut négliger l'inertie. En effet, pour un lopin en acier, il faudrait que localement la vitesse de la matière atteigne 50m/s pour que les forces d'inertie soient égales à un dixième des forces de déformation. Les forces de volume quant à elles sont uniquement la gravité, que l'on peut aussi omettre face aux autres efforts. La hauteur du même lopin d'acier doit être de 2000m pour que les forces de gravité soient de l'ordre des efforts dus à la plasticité [Boyère 1999].

Ainsi, les équations d'équilibre se présentent comme la condition de nullité de la divergence du tenseur des contraintes :

$$\frac{\partial \sigma_{ij}}{\partial x_j} = 0, \quad \forall x \in \Omega \quad (\text{I.9})$$

L'équation de conservation de la masse s'écrit dans sa forme générale :

$$\frac{d\rho}{dt} + \text{div}(\rho \cdot v) = 0 \quad (\text{I.10})$$

Si l'on considère d'une part l'hypothèse d'un cas stationnaire, la dérivée de la masse volumique par rapport au temps s'annule. D'autre part, si le matériau est incompressible, sa déformation élastique est négligeable devant la déformation plastique ($\rho = \text{constante}$), alors la masse volumique est constante et l'équation (I.10) revient à satisfaire la condition d'incompressibilité dans Ω :

$$\frac{\partial v_i}{\partial x_i} = 0 \quad \forall x \in \Omega \quad (\text{I.11})$$

Dans le cas élastique, le gradient de vitesse est égal alors à la trace du tenseur de vitesse de déformation élastique :

$$\frac{\partial v_i}{\partial x_i} = \dot{\varepsilon}_{11}^{el} + \dot{\varepsilon}_{22}^{el} + \dot{\varepsilon}_{33}^{el} \quad \forall x \in \Omega$$

I.2.2. Lois rhéologiques

La rhéologie concerne l'étude de l'écoulement et de la déformation de la matière lorsque des contraintes lui sont appliquées. Ces relations relient les contraintes aux déformations ou vitesses de déformation de la matière. Suivant le cas, le comportement du matériau peut être de type élastique, élasto-plastique, élasto-visco-plastique ou visco-plastique. On présente ci-dessous ces différents types de comportement.

I.2.2.1. Comportement élastique

Il se caractérise tout d'abord par le fait que la déformation est réversible. La loi de Hooke généralisée s'écrit en petites déformations :

$$\sigma_{ij} = ID_{ijkl}^{el} \cdot \varepsilon_{kl}, \quad (\text{I.12})$$

où ID_{ijkl}^{el} est le tenseur d'élasticité d'ordre 4 défini par :

$$ID_{ijkl}^{el} = 2\mu\delta_{ik}\delta_{jl} + \lambda\delta_{kl}\delta_{ij}, \quad (\text{I.13})$$

$$\mu = \frac{E}{2(1+\gamma)}, \quad \lambda = \frac{E\gamma}{(1+\gamma)(1-2\gamma)}$$

où λ et μ sont les coefficients de Lamé, E le module d'Young, γ le coefficient de Poisson, et δ_{ij} le symbole de Kronecker. Dans la relation (I.12) ε_{ij} est le tenseur des déformations linéarisées, défini à partir du gradient des déplacements u_i :

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (\text{I.14})$$

Il convient ici de préciser que, par la suite, c'est une formulation en vitesse qui est adoptée pour résoudre le problème mécanique. Alors la loi de Hooke est reformulée en fonction des dérivées (au sens de Jaumann) des champs des contraintes et de déformations :

$$\dot{\varepsilon}_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \text{ et } \dot{\sigma}_{ij} = ID_{ijkl}^{el} \cdot \dot{\varepsilon}_{kl} \quad (\text{I.15})$$

1.2.2.2. Comportement élasto-plastique

La formulation élasto-plastique introduite dans Forge3[®] est basée sur une décomposition additive de la vitesse de déformation en une partie élastique et une partie plastique :

$$\dot{\varepsilon}_{ij} = \dot{\varepsilon}_{ij}^{el} + \dot{\varepsilon}_{ij}^{pl} \quad (\text{I.16})$$

L'évolution de la plasticité pour un matériau standard associé est décrite par la loi de Prandtl-Reuss. Pour la variable d'écroutissage $\bar{\varepsilon}$, on définit le critère de Von Mises comme :

$$f(\sigma, \bar{\varepsilon}) = \left\{ \frac{1}{2} \left[(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2 + 6\sigma_{12}^2 + 6\sigma_{23}^2 + 6\sigma_{31}^2 \right] \right\}^{1/2} - \sigma_0(\bar{\varepsilon}) \quad (\text{I.17})$$

$\bar{\varepsilon}$ représente la déformation plastique cumulée et $\sigma_0(\bar{\varepsilon})$ le seuil de plasticité en traction simple. La fonction $f(\sigma, \bar{\varepsilon})$ peut être écrite aussi à l'aide de la partie déviatorique s_{ij} du tenseur des contraintes :

$$f(\sigma, \bar{\varepsilon}) = \bar{\sigma} - \sigma_0(\bar{\varepsilon}) \quad (\text{I.18})$$

où la contrainte équivalente est donnée par :

$$\bar{\sigma} = \sqrt{\frac{3}{2} s_{ij} s_{ij}} \quad (\text{I.19})$$

La loi de Prandtl-Reuss s'écrit comme [Montmitonnet 1994] :

- si $f(\sigma, \bar{\varepsilon}) < 0$ ou si $f(\sigma, \bar{\varepsilon}) = 0$ et $\partial f / \partial \sigma : \dot{\sigma} < 0$ (charge ou décharge élastique) alors :

$$\dot{\sigma}_{ij} = ID_{ijkl}^{el} \cdot \dot{\varepsilon}_{kl}^{el} \text{ et } \dot{\varepsilon}_{ij}^{pl} = 0 \quad (\text{I.20})$$

- si $f(\sigma, \bar{\varepsilon}) = 0$ et $\partial f / \partial \sigma : \dot{\sigma} > 0$ (charge plastique) alors :

$$\dot{\varepsilon}_{kl}^{el} = [ID_{ijkl}^{el}]^{-1} \dot{\sigma}_{ij} \text{ et } \dot{\varepsilon}_{ij}^{pl} = ID^{pl} \frac{\partial f}{\partial \sigma} \quad (ID^{pl} > 0) \quad (\text{I.21})$$

Pour ne pas rentrer davantage de détails sur l'écriture tensorielle, nous renvoyons le lecteur à [Montmitonnet 1994].

I.2.2.3. Comportement visco-plastique

Lorsque l'on considère les problèmes de mise en forme à chaud des métaux, le comportement de la pièce peut être modélisé comme viscoplastique. Dans ce cas, la composante élastique est négligeable devant la composante visqueuse. On utilise souvent la loi de Norton-Hoff pour modéliser ce comportement visqueux :

$$s_{ij} = 2K(T, \bar{\varepsilon}, \dots) \left(\sqrt{3} \dot{\bar{\varepsilon}} \right)^{m-1} \dot{\varepsilon}_{ij}, \quad (\text{I.22})$$

où $K(T, \bar{\varepsilon}, \dots)$ est la consistance du matériau, m son coefficient de sensibilité à la vitesse, $\dot{\bar{\varepsilon}}$ est la vitesse de déformation équivalente :

$$\dot{\bar{\varepsilon}} = \sqrt{\frac{2}{3} \dot{\varepsilon}_{ij} \dot{\varepsilon}_{ij}} \quad (\text{I.23})$$

Le déviateur des contraintes s_{ij} est défini par :

$$\sigma_{ij} = s_{ij} + \frac{1}{3} \text{Trace}(\sigma_{ij}) \delta_{ij} = s_{ij} - p \delta_{ij} \quad (\text{I.24})$$

où p désigne la pression hydrostatique.

Dans le cas particulier où $m = 1$, il s'agit du comportement newtonien, d'habitude la consistance K est constante.

Mais, dans le cas général, la consistance K suit des lois non-linéaires, comme par exemple la loi puissance définie à l'aide des coefficients K_0 , n , β et ε_0

$$K(T, \bar{\varepsilon}) = K_0 (\bar{\varepsilon} + \varepsilon_0)^n \exp(\beta/T) \quad (\text{I.25})$$

ou encore la loi d'érouissage de Hansel et Spittel :

$$K(T, \bar{\varepsilon}) = K_0 \cdot \exp(m_1 T) \cdot \bar{\varepsilon}^{m_2} \cdot \dot{\bar{\varepsilon}}^{m_3} \cdot \exp(m_4 / \bar{\varepsilon}) \quad (\text{I.26})$$

I.2.3. Conditions aux limites mécaniques

Pour compléter le système constitué des équations de conservation et des lois de comportement, il faut imposer des conditions aux limites. En notant par $\partial\Omega$ le bord du domaine Ω , on décompose celui-ci en une réunion des parties suivantes :

$$\partial\Omega = \partial\Omega_0 \cup \partial\Omega_T \cup \partial\Omega_c \quad (\text{I.27})$$

avec

- $\partial\Omega_0$ la surface libre où la contrainte normale est nulle : $\sigma_{ij} n_j = 0$. On considère que la pression atmosphérique p_{ext} est nulle.
- $\partial\Omega_T$ est la surface où la contrainte est imposée : $\sigma_{ij} n_j = T_i$

- $\partial\Omega_c$ est la surface de contact ; sur cette partie de la surface sont définies des conditions de frottement, comme par exemple sur l'interface entre la pièce avec des outils rigides où des outils déformables.

On présente ici surtout le cas des outils rigides, pour les détails sur les corps déformables nous renvoyons le lecteur à la thèse de J.Barboza [Barboza 2004].

1.2.3.1. Contact avec des outils rigides

On introduit la fonction g qui, pour chaque point de la pièce, représente la distance signée de ce point à la surface de l'outil. Pour tout point x on détermine son projeté $\pi(x)$ sur la surface de l'outil. Par convention, la fonction écart $g(x)$ est positive si le point est extérieur à l'outil, égal à zéro sur sa surface et négative à l'intérieur de celui-ci (Figure I.6a). Un point est donc considéré en contact si $g \leq 0$. On peut ainsi écrire :

$$g(x) = (\pi_i(x) - x_i)n_i \quad (\text{I.28})$$

où n_i est la normale sortante de la pièce et donc rentrante dans l'outil. Pour imposer la non-pénétration sur la surface de contact, il faut que la condition suivante soit vérifiée, pour chaque point du corps déformable potentiellement en contact avec l'outil rigide :

$$g(x) \geq 0 \quad \forall x \in \partial\Omega_c \quad (\text{I.29})$$

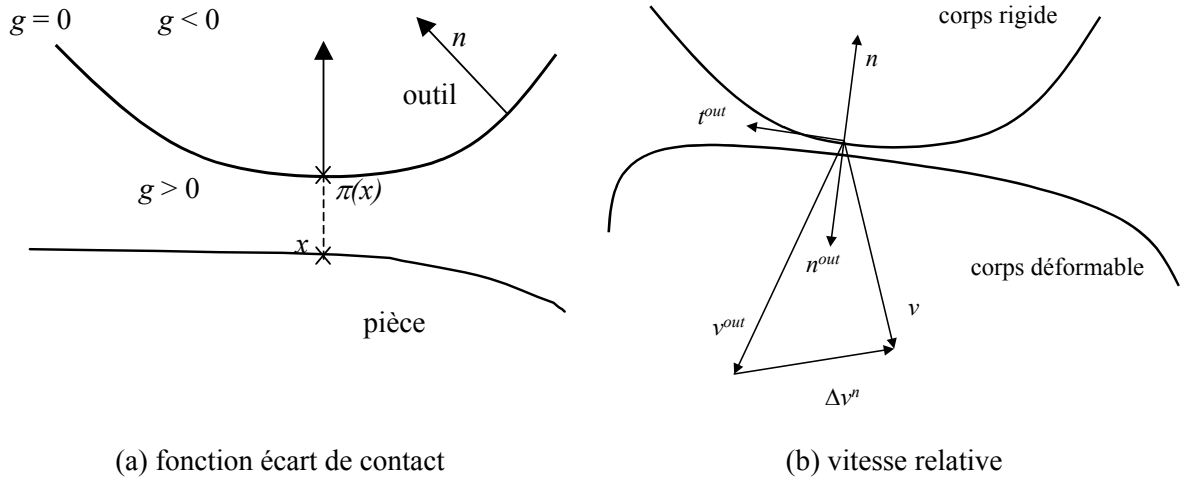
Sur la surface potentiellement en contact $\partial\Omega_c$ (donc pour $g(x) = 0$), la formulation cinématique des conditions de contact utilise les notions suivantes (Figure I.6b) :

- n^{out} est la normale sortante de l'outil et donc rentrante dans le corps déformable $n^{out} = -n$;
- v et v^{out} sont les vitesses de la pièce et de l'outil ;
- $\Delta v_i = v_i - v_i^{out}$ est la vitesse relative ;
- Δv^n est la vitesse normale relative définie par :

$$\Delta v_i^n = (v_j - v_j^{out})n_j n_i = \Delta v_j n_j n_i \quad (\text{I.30})$$

- Δv^t la vitesse tangentielle relative :

$$\Delta v_i^t = (v_i - v_i^{out}) - \Delta v_i^n = \Delta v_i - \Delta v_i^n \quad (\text{I.31})$$



(a) fonction écart de contact

(b) vitesse relative

Figure I.6. Description géométrique de la zone de contact entre les deux corps

Il existe différentes formulations des conditions de contact. Le contact peut être bilatéral. Dans ce cas, une fois le contact établi sur une partie de surface, il est impossible que les points ainsi en contact le quittent. Si en plus on interdit un déplacement tangentiel des points en contact, le contact est dit bilatéral collant. Cette condition s'écrit comme :

$$v_i - v_i^{out} = 0 \quad \forall i = 1,2,3 \quad (I.32)$$

Si le déplacement tangentiel est autorisé le contact est dit glissant :

$$(v_i - v_i^{out})n_i = 0 \quad (I.33)$$

Enfin, les conditions de Signorini décrivent le contact unilatéral glissant :

$$\begin{aligned} (v_i - v_i^{out})n_i &\leq 0 \\ \sigma^n &\leq 0 \\ \sigma^n \cdot (v_i - v_i^{out})n_i &= 0 \end{aligned} \quad (I.34)$$

et le contact unilatéral collant :

$$\begin{aligned} (v_i - v_i^{out})n_i &\leq 0 \\ \sigma^n &\leq 0 \\ (v_i - v_i^{out}) - (v_j - v_j^{out})n_j n_i &= 0 \quad \text{si } \sigma^n < 0 \end{aligned} \quad (I.35)$$

Dans les équations (I.32) et (I.33) la contrainte normale σ^n est définie par :

$$\sigma^n = \sigma_{ij} n_i n_j = \sigma_{ij}^{out} n_i^{out} n_j^{out} \quad (I.36)$$

I.2.3.2. Lois de frottement

Ces lois définissent la relation entre le vecteur de contrainte tangentielle τ_i , dont la norme τ est appelée la scission de frottement, et des différentes valeurs mécaniques et cinématiques de l'interface de contact. τ_i est la composante tangentielle du tenseur des contraintes :

$$\tau_i = \sigma_{ij} n_j - \sigma^n n_i \quad (\text{I.37})$$

En général, les lois de frottement sont formulées comme des lois discontinues avec un seuil de frottement τ^c qui représente la contrainte tangentielle minimale nécessaire pour qu'un glissement apparaisse entre deux corps :

- si $|\tau_i| < \tau^c$ alors $\Delta v_i^t = 0$ (le contact est collant)
- si $|\tau_i| = \tau^c$ alors $\exists \lambda \geq 0 : \Delta v_i^t = -\lambda \tau_i$ (le contact est glissant) (I.38)
- $|\tau_i| > \tau^c$ est impossible physiquement

Loi de Coulomb

Cette loi établit la relation linéaire entre le seuil de frottement et la pression de contact $p^c = -\sigma^n$ ($p^c \geq 0$) :

$$\tau^c = \mu p^c \quad (\text{I.39})$$

où $\mu \geq 0$ est le coefficient de frottement. Cette relation donne :

- si $|\tau_i| < \mu p^c$ alors $\Delta v_i^t = 0$
- si $|\tau_i| = \mu p^c$ alors $\Delta v_i^t = -\frac{\|\Delta v_i^t\|}{\mu p^c} \tau_i$ (I.40)
- $|\tau_i| > \mu p^c$ est impossible physiquement

Loi de Tresca

Si l'un des corps est plastique incompressible, son critère de plasticité limite la scission par $\tau \leq \tau_{\max}$, où τ_{\max} vaut $\sigma_0/\sqrt{3}$ pour le critère de Von Mises ou bien $\sigma_0/2$ pour le critère de Tresca, avec σ_0 la contrainte d'écoulement du matériau (cf. (I.17)). Ainsi, on pose

$$\tau^c = \bar{m} \frac{\sigma_0}{\sqrt{3}} \quad (\text{I.41})$$

avec le coefficient de frottement $0 \leq \bar{m} \leq 1$.

Lois de type viscoplastique

Elles se caractérisent par le fait que, contrairement aux deux lois précédentes, le domaine interdit ($\tau > \tau^c$) n'est pas atteint. Au-delà du seuil (éventuellement nul), la contrainte est d'autant plus forte

que la vitesse relative est grande. La loi de Norton est une loi de frottement viscoplastique sans seuil, et donc elle relie la contrainte tangentielle à la vitesse de glissement :

$$\tau_i = -\alpha_f K \left\| \Delta v_i' \right\|^{p_f - 1} \Delta v_i' \quad (\text{I.42})$$

avec respectivement α_f et p_f les coefficients de frottement et de sensibilité à la vitesse de glissement, et K la consistance du matériau. Si p_f tend vers zéro, la loi de Norton tend vers la loi de Tresca sans seuil. Même pour une faible scission τ_i , il y a glissement.

I.2.4. Formulation faible

On introduit les espaces fonctionnels des vitesses cinématiquement admissibles à condition de contact V_{ca} (qui vérifie les conditions de contact (I.32), (I.33), (I.34) ou (I.35)) et cinématiquement admissibles à zéro V_{ca}^0 (mêmes équations avec $v^{out} = 0$), ainsi que l'espace P des pressions :

$$\begin{aligned} V_{ca} &= \left\{ v, v \in (H^1(\Omega))^3, (v_i - v_i^{out}) n_i \leq 0 \text{ sur } \partial\Omega_c \right\} \\ V_{ca}^0 &= \left\{ v, v \in (H^1(\Omega))^3, v_i n_i \leq 0 \text{ sur } \partial\Omega_c \right\} \\ P &= L^2(\Omega) \end{aligned}$$

$L^2(\Omega)$ est l'espace des fonctions de carré sommable sur Ω , et $H^1(\Omega)$ est l'espace de Sobolev des distributions de carré sommable ainsi que leur dérivée première. Par Q on désigne un espace des vecteurs de vitesses et rotations ponctuels.

Le principe des puissances virtuelles appliqué aux équations de conservation (I.9) et (I.11) donne la forme faible du problème mixte en vitesse/pression :

Pour des variations quelconques $v^* \in V_{ca}^0$ et $p^* \in P$

trouver $v \in V_{ca}$ et $p \in P$ tels que :

$$\begin{cases} \int_{\Omega} s_{ij} \dot{\varepsilon}_{ij}^* d\Omega - \int_{\Omega} p \operatorname{div}(v^*) d\Omega - \int_{\partial\Omega_T} T_i v_i^* dS - \int_{\partial\Omega_c} \tau_i v_i^* dS = 0 \\ - \int_{\Omega} p^* \operatorname{div}(v) d\Omega = 0 \end{cases} \quad (\text{I.43})$$

où le déviateur des contraintes s_{ij} est lié aux champs des vitesses par une des lois rhéologiques présentées dans la section précédente. Le système (I.43) est écrit pour le cas rigide plastique ($\operatorname{div}(v) = 0$), dans le cas élastique la deuxième équation est plus complexe. Pour le système composé de la pièce et de la presse, ce système (I.43) reste valable, mais il faut rajouter les équations pour la presse. En se basant sur la description de la vitesse de l'outil flottant (I.5), on applique le même

principe des puissances virtuelles à l'outil rigide auquel le modèle de presse est associé, ce qui amène à la formulation faible du problème dans la presse :

Pour une variation quelconque $(v^{O*}, \omega^*) \in Q$

trouver $v^{pr} = v^O + \omega \wedge OM$ tels que (I.44)

$$\int_{\partial\Omega_C} \tau_i v_i^{pr*} dS - F_i^{pr} v_i^{O*} - C_i^{pr} \omega_i^* = 0$$

L'intégrale sur volume est ici égale à zéro, puisque cet outil est un corps rigide (il n'a y donc pas de forces intérieures). Cette formulation pour la presse déformable ressemble beaucoup à la formulation des outils flottants présentée dans [Boyère 1999] et [Traoré 2001]. La différence majeure est dans le fait que les forces F_i^{pr} et les couples C_i^{pr} vérifiant l'équation (I.6) ne sont pas imposées ; on développera ceci dans la suite.

I.3. FORMULATION DU PROBLÈME DISCRET

I.3.1. Discrétisation temporelle et contact incrémental

Les problèmes de forgeage se présentent comme des problèmes clairement instationnaires. Dans Forge3[®], on utilise un formalisme de lagrangien réactualisé. Si l'on divise la durée de forgeage en pas de temps, la nouvelle configuration est calculée à partir des valeurs de la configuration précédente et non la configuration initiale [Boyère 1999, Traoré 2001]. On résout à chaque incrément de calcul l'équilibre quasi-statique du matériau en train de se déformer sous l'effet des outils et de la température. Pour modéliser cette évolution, on adjoint aux équations exposées précédemment l'équation différentielle associée à la réactualisation de la configuration :

$$\begin{cases} \frac{dx(t)}{dt} = v(x, t) \\ x(0) = x^0 \end{cases} \quad (I.45)$$

Le plus souvent on utilise un schéma d'Euler explicite (ou implicite pour une loi élasto-viscoplastique) :

$$x(t + \Delta t) = x(t) + v(x, t)\Delta t + O(\Delta t^2) \quad (I.46)$$

Ainsi, en connaissant à l'instant t la configuration $\Omega(t)$ et supposant que la vitesse $v(t)$ est constante au cours de l'incrément, on calcule la nouvelle configuration à l'instant suivant $t + \Delta t$. L'erreur commise sur la réactualisation des positions avec cette méthode est proportionnelle à Δt^2 .

En ce qui concerne le contact, la condition de non-pénétration (I.29) est imposée à l'instant $t + \Delta t$, ainsi le déplacement des points du corps qui peuvent traverser la surface de l'outil rigide est pris en compte :

$$g(x, t + \Delta t) \geq 0 \quad (\text{I.47})$$

On linéarise cette relation en utilisant le schéma d'intégration (I.46) de manière à l'exprimer en fonction des variables connues à l'instant t .

$$\begin{aligned} g(t + \Delta t) &= g(t) + \frac{dg(t)}{dt} \Delta t + O(\Delta t^2) \\ &= g(t) + \frac{d((\pi_i(x(t)) - x_i(t))n_i(t))}{dt} \Delta t + O(\Delta t^2) \\ &= g(t) + \frac{d(\pi_i(x(t)) - x_i(t))}{dt} \Delta t n_i(t) + (\pi_i(x(t)) - x_i(t)) \Delta t \frac{d(n_i(t))}{dt} + O(\Delta t^2) \end{aligned} \quad (\text{I.48})$$

La surface de l'outil peut être approchée localement par son plan tangent $\Pi(x, t)$ défini par $\pi(x(t))$ et $n(t)$ (Figure I.7), ce qui revient à négliger le terme $(\pi_i(x(t)) - x_i(t)) \Delta t \cdot d(n_i(t))/dt$ dans (I.48). Ainsi, la linéarisation de la condition de non-pénétration aboutit à :

$$\begin{aligned} g(t + \Delta t) &= g(t) + \frac{d(\pi_i(x(t)) - x_i(t))}{dt} \Delta t n_i(t) + O(\Delta t^2) \\ &= g(t) + \frac{d\pi_i(x(t))}{dt} \Delta t n_i(t) - \frac{dx_i(t)}{dt} \Delta t n_i(t) + O(\Delta t^2) \\ &= g(t) + (v_i^{out}(t) - v_i(t)) \Delta t n_i(t) + O(\Delta t^2) \end{aligned} \quad (\text{I.49})$$

On introduit la fonction $h(v(t))$ qui exprime la condition de contact unilatéral :

$$\begin{aligned} h(v(t)) &\leq 0 \quad \text{ou} \quad [h(v(t))]^+ = 0 \\ \text{où } h(v(t)) &= (v_i(t) - v_i^{out}(t)) n_i(t) - \frac{g(t)}{\Delta t} \end{aligned} \quad (\text{I.50})$$

ici $[x]^+$ désigne la partie positive de x .

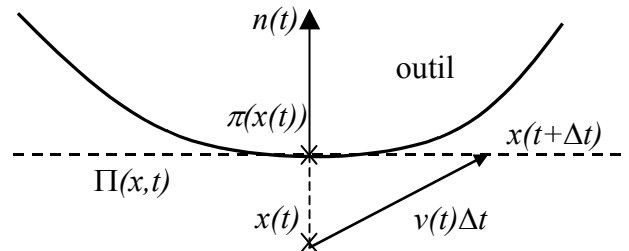


Figure I.7. Approximation de la surface de l'outil rigide pour le contact explicite

I.3.2. Discrétisation spatiale

Soit Ω_h une triangulation du domaine Ω en éléments Ω_e ayant une taille caractéristique h . On note E l'ensemble des éléments, si bien que :

$$\Omega_h = \bigcup_{e \in E} \Omega_e \quad (\text{I.51})$$

1.3.2.1. Élément fini mixte P1+/P1

L'élément utilisé pour la discrétisation spatiale est l'élément mixte P1+/P1 (Figure I.8). C'est un tétraèdre linéaire, sur lequel les coordonnées, les vitesses et la pression sont interpolées linéairement en fonction des valeurs aux sommets. Le « + » indique le rajout d'un degré de liberté au centre du tétraèdre pour l'interpolation de la vitesse. La fonction d'interpolation associée est linéaire et vaut 1 au centre et 0 sur la frontière (d'où l'appellation de fonction bulle). Ce degré de liberté supplémentaire permet à l'élément P1+/P1 de satisfaire la relation de compatibilité de Brezzi-Babuska entre les espaces d'interpolation de la vitesse et de la pression [Coupez 1995].

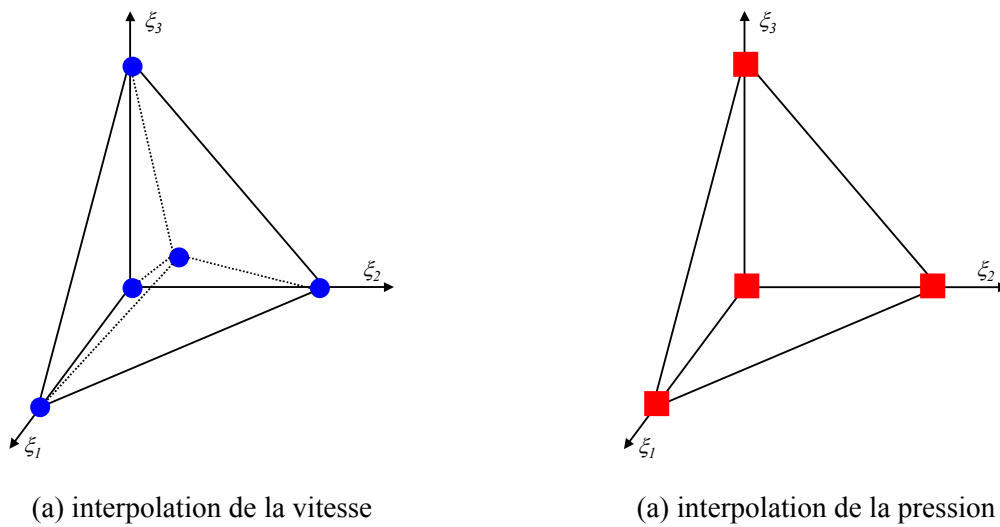


Figure I.8. Élément fini mixte P1+/P1

Le champ de vitesse interpolé w^h se décompose de manière unique en une partie linéaire v^h et en une partie b^h relative au champ bulle :

$$w^h = v^h + b^h \quad \text{où} \quad w^h \in V_h = V_h^l \oplus B_h \quad (\text{I.52})$$

$$V_h^l = \left\{ v^h : v^h \in (C^0(\Omega_h))^3, \forall \Omega_e \in E \quad v^h(\Omega_e) \in (P1(\Omega_e))^3 \right\} \quad (\text{I.53})$$

$$B_h = \left\{ b^h : b^h \in (C^0(\Omega_h))^3, b^h = 0 \text{ sur } \partial\Omega_e, \forall \Omega_e \in E \quad b^h(\Omega_e) \in (P1(\Omega_e^i))^3, i = 1, \dots, 4 \right\} \quad (\text{I.54})$$

ici $(C^0(\Omega_h))^3$ est l'espace des fonctions continues sur le domaine discrétisé $\Omega_h \subset \mathbb{R}^3$, $(P1(\Omega_e))^3$ l'espace des fonctions linéaires sur l'élément $\Omega_e \subset \mathbb{R}^3$, Ω_e^i , $i = 1, \dots, 4$ représente, pour tout élément Ω_e du maillage, les sous-tétraèdres étoilés à partir de son centre. La fonction bulle est donc continue, mais dérivable par morceaux seulement.

Le champ de pression est linéaire et continu :

$$P_h = \{p^h : p^h \in C^0(\Omega_h), \forall \Omega_e \in E \quad p^h(\Omega_e) \in P1(\Omega_e)\} \quad (I.55)$$

Par conséquent, les champs de vitesse et de pression discrétisés, ainsi que les coordonnées, s'écrivent en un point quelconque $x \in \Omega_h$ de l'espace :

$$w^h(x) = \sum_{k=1}^{Nbnoe} V_k N_k^l(x) + \sum_{k=1}^{Nbelt} B_k N_k^b(x) \quad (I.56)$$

$$p^h(x) = \sum_{k=1}^{Nbnoe} P_k N_k^l(x) \quad (I.57)$$

$$x^h = \sum_{k=1}^{Nbnoe} X_k N_k^l(x) \quad (I.58)$$

où $w^h \in V_h$, $N_k^l \in V_h^l$, $k=1, \dots, Nbnoe$ sont les fonctions d'interpolation de l'élément linéaire associées aux nœuds k , $N_k^b \in B_h$, $k=1, \dots, Nbelt$ la fonction bulle associée à l'élément k . $Nbnoe$ et $Nbelt$ sont respectivement le nombre de nœuds et d'éléments du maillage.

I.3.2.2. Traitement du contact

Au niveau discret, la condition de contact (I.50) est imposée en chaque nœud du maillage, c'est ce que l'on appelle le contact nodal. Ainsi, pour chaque nœud k qui se trouve sur la surface du contact de la pièce $\partial\Omega_{hC}$, la fonction discrétisée $h(v^h)$ doit-elle être positive :

$$\forall k \in \partial\Omega_{hC} \quad [h(v^h(X_k))]^+ = 0 \quad \text{avec} \quad h(v^h(X_k)) = (V_k - v^{out})n_k^h - \frac{g_k^h}{\Delta t} \quad (I.59)$$

où $g_k^h = g(X_k)$ et $n_k^h = n(X_k)$ est la normale construite en nœud X_k .

La méthode de pénalisation est utilisée pour imposer les équations de contact [Mocellin 1999]. On introduit un facteur de pénalisation ρ_C . La contrainte normale de contact peut être alors exprimée en fonction de la vitesse. Ainsi, pour un nœud X_k de façon approchée avec une précision $O(1/\rho_C)$ on peut écrire :

$$\sigma_n(X_k) \cong -\rho_C h(v^h(X_k)) \quad (I.60)$$

Pour une résolution exacte il faudrait atteindre $\rho_C = +\infty$ ce qui n'est pas possible. Pour des valeurs trop grandes de ce coefficient, on déconditionne la matrice du problème et l'utilisation des méthodes itératives devient impossible. Dans la pratique, on utilise $\rho_C \approx 10^2$ et donc la condition de contact est vérifiée avec une précision relative de $O(1/\rho_C) \approx 10^{-2}$

Le problème discrétisé (I.43)-(I.44) s'écrit à chaque instant de $(t, t + \Delta t)$:

$$\text{Pour une variation quelconque } (w^{h*}, p^{h*}, v^{pr*} = (v^{O*}, \omega^*)) \in V_h \times P_h \times Q \quad (I.61)$$

Trouver $(w^h, p^h, (v^O, \omega)) \in V_h \times P_h \times Q$ tels que :

$$\left\{ \begin{array}{l} \int_{\Omega} s_{ij}(w^h) \dot{\epsilon}_{ij}(w^{h*}) d\Omega - \int_{\Omega} p^h \operatorname{div}(v^{h*}) d\Omega - \int_{\partial\Omega_T} T_i v_i^{h*} dS - \int_{\partial\Omega_C} \tau_i v_i^{h*} dS + \rho_C \sum_{k \in \partial\Omega_C} h(v^h) v_i^{h*} n_{ki}^h S_k = 0 \\ - \int_{\Omega} p^{h*} \operatorname{div}(v^h) d\Omega = 0 \\ \int_{\partial\Omega_C} \tau_i v_i^{pr*} dS - \rho_C \sum_{k \in \partial\Omega_C} h(v^h) v_i^{pr*} n_{ki}^h S_k - F_i^{press} v_i^{O*} - C_i^{press} \omega_i^* = 0 \end{array} \right.$$

On a noté ici n_{ki}^h la composante i de la normale construite en en nœud X_k , S_k est la surface associée au noeud k , utilisée pour affecter un poids plus fort aux noeuds liés à des faces frontières de plus grande surface :

$$S_k = \int_{\partial\Omega_C} N_k ds$$

I.3.3. Modèle de presse

Selon l'équation de comportement élastique de la presse (I.6), la composante en translation est introduite par l'expression :

$$F_i^{pr} = k_{ij}^{xx} \Delta x_j + k_{ij}^{x\theta} \Delta \theta_j \quad (\text{I.62})$$

Dans le cas le plus général, la matrice de flexibilité est non-symétrique. Toutes les équations suivantes sont donc établies d'une manière générale, sans aucune hypothèse sur la structure de la matrice de flexibilité.

On définit le déplacement Δx du bâti de la presse dû à la déformation élastique comme la différence entre la position du centre de gravité x^p de l'outil auquel le modèle de presse est appliqué et la position x^0 :

$$\Delta x = x^p - x^0 \quad (\text{I.63})$$

La position x^0 est définie en même temps que la cinématique de l'outillage. Par exemple, dans le cas d'une presse mécanique, on peut appliquer le modèle de presse à l'outil inférieur (Figure I.9b) qui est censé d'être immobile, donc x^0 sera nul, ou l'on peut appliquer le modèle à l'outil supérieur (Figure I.9c) dont la cinématique est définie par une vitesse constante ou variable. Dans ce cas, la prise en compte de la raideur de la presse implique des déplacements de l'outil différents de ceux prévus par la mécanique de la presse, cette différence étant exprimée par Δx .

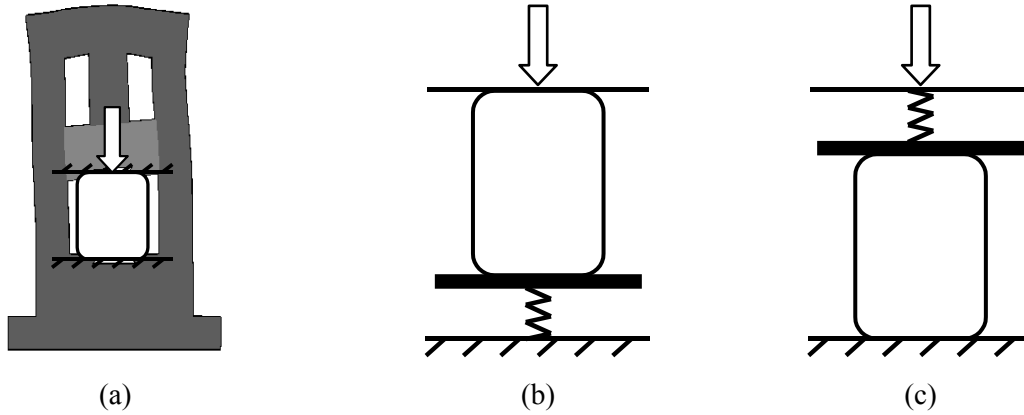


Figure I.9. Modèle de raideur de presse appliqué aux outils.

Pour calculer les valeurs x^p et x^o à l'instant $t + \Delta t$, on applique le même schéma d'intégration explicite (I.46) :

$$\begin{cases} x^p(t + \Delta t) = x^p(t) + v^p(t)\Delta t \\ x^o(t + \Delta t) = x^o(t) + v^o(t)\Delta t \end{cases} \quad (\text{I.64})$$

On utilise les notations suivantes pour désigner les vitesses :

- $v_j^p(t)$ est la composante j du vecteur inconnu de la vitesse de la presse à l'instant t
- $v_j^o(t)$ est la composante j du vecteur de la vitesse connue, prédéfinie de la presse (de l'outil) à l'instant t .

Avec ces notations, on obtient la composante linéaire de la vitesse de la presse v^o (cf. équation (I.5)) comme la somme $v^o = v^{pr} + v^0$, d'une vitesse autorisée, donc connue, v^0 et celle v^{pr} qui dépend des équations d'équilibre.

Par ailleurs, la rotation de la presse au moment $t + \Delta t$ est exprimée d'une manière semblable à (I.64) par :

$$\begin{cases} \theta^p(t + \Delta t) = \theta^p(t) + \omega^p(t)\Delta t \\ \theta^o(t + \Delta t) = \theta^o(t) + \omega^o(t)\Delta t \end{cases} \quad (\text{I.65})$$

ici θ est le vecteur des rotations angulaires suivant chacun des axes, où θ_j , $j = 1,2,3$, est la composante selon l'axe j , et $\omega = \{\omega_1 \quad \omega_2 \quad \omega_3\}^T$ est la vitesse de rotation angulaire. Les indices p et o ont la même signification qu'avec la composante linéaire. Ainsi, on obtient la relation suivante pour réactualiser la valeur de la force au moment $t + \Delta t$:

$$\begin{aligned}
F_i^{pr}(t + \Delta t) &= k_{ij}^{xx} [x_j^p(t + \Delta t) - x_j^0(t + \Delta t)] + k_{ij}^{x0} [\theta_j^p(t + \Delta t) - \theta_j^0(t + \Delta t)] \\
F_i^{pr}(t + \Delta t) &= k_{ij}^{xx} [x_j^p(t) + v_j^p(t + \Delta t)\Delta t - x_j^0(t) - v_j^0(t + \Delta t)\Delta t] \\
&\quad + k_{ij}^{x0} [\theta_j^p(t) + \omega_j^p(t + \Delta t)\Delta t - \theta_j^0(t) - \omega_j^0(t + \Delta t)\Delta t] \\
F_i^{pr}(t + \Delta t) &= F_i^{pr}(t) - k_{ij}^{xx} v_j^0(t + \Delta t)\Delta t - k_{ij}^{x0} \omega_j^0(t + \Delta t)\Delta t \\
&\quad + k_{ij}^{xx} v_j^p(t + \Delta t)\Delta t + k_{ij}^{x0} \omega_j^p(t + \Delta t)\Delta t
\end{aligned} \tag{I.66}$$

Suivant la même méthode qui vient d'être appliquée à la composante de translation, d'après l'équation (I.6), on écrit le couple de la presse comme :

$$C_i^{pr} = k_{ij}^{0x} \Delta x_j + k_{ij}^{00} \Delta \theta_j \tag{I.67}$$

et suivant le même raisonnement que dans (I.66) on obtient :

$$\begin{aligned}
C_i^{pr}(t + \Delta t) &= C_i^{pr}(t) - k_{ij}^{0x} v_j^0(t + \Delta t)\Delta t - k_{ij}^{00} \omega_j^0(t + \Delta t)\Delta t \\
&\quad + k_{ij}^{0x} v_j^p(t + \Delta t)\Delta t + k_{ij}^{00} \omega_j^p(t + \Delta t)\Delta t
\end{aligned} \tag{I.68}$$

Ainsi, la troisième équation de (I.61) décrivant l'équilibre de la presse prend la forme:

Pour une variation quelconque $v^{pr*} = (v^{O*}, \omega^*) \in Q$

trouver $(v^O, \omega) \in Q$ tels que $v^{pr} = v^O + \omega \wedge OM$ et

$$\begin{aligned}
&\int_{\partial\Omega_C} \tau_i v_i^{pr*} dS - \rho_C \sum_{k \in \partial\Omega_C} h(v^h) v_i^{pr*} n_{ki}^h S_k \\
&- (F_i^{pr}(t) - k_{ij}^{xx} v_j^0(t + \Delta t)\Delta t - k_{ij}^{x0} \omega_j^0(t + \Delta t)\Delta t + k_{ij}^{xx} v_j^p(t + \Delta t)\Delta t + k_{ij}^{x0} \omega_j^p(t + \Delta t)\Delta t) v_i^{O*} \\
&- (C_i^{pr}(t) - k_{ij}^{0x} v_j^0(t + \Delta t)\Delta t - k_{ij}^{00} \omega_j^0(t + \Delta t)\Delta t + k_{ij}^{0x} v_j^p(t + \Delta t)\Delta t + k_{ij}^{00} \omega_j^p(t + \Delta t)\Delta t) \omega_i^* = 0
\end{aligned} \tag{I.69}$$

I.3.4. Linéarisation des équations discrètes

En utilisant les indices v pour les vitesses linéaires, b pour les vitesses liées aux termes bulle, p pour la pression et a pour la vitesse de la presse, on peut réécrire le système discret non-linéaire (I.61)

sous une forme vectorielle $R(v^h, b^h, p^h, a) = 0$:

$$\begin{aligned}
R^v &= R^{vv} + R^{vb} + R^{vp} + R^{va} = 0 \\
R^b &= R^{bv} + R^{bb} + R^{bp} = 0 \\
R^p &= R^{pv} + R^{pb} = 0 \\
R^a &= R^{av} + R^{aa} = 0
\end{aligned} \tag{I.70}$$

On résout ce problème non-linéaire par une méthode de Newton-Raphson. Pour cela, on introduit les matrices hessiennes H^{xy} définies par :

$$H^{xy} = \frac{\partial R^x}{\partial y} \tag{I.71}$$

où x et y évoluent dans $\{v, b, p, a\}$. A chaque itération $n+1$ de cette méthode, on aboutit à la résolution du système linéaire suivant :

$$\begin{bmatrix} H^{vv} & 0 & H^{vp} & H^{va} \\ 0 & H^{bb} & H^{bv} & 0 \\ H^{pv} & H^{pb} & 0 & 0 \\ H^{va} & 0 & 0 & H^{aa} \end{bmatrix} \cdot \begin{bmatrix} \delta v \\ \delta b \\ \delta p \\ \delta a \end{bmatrix} = - \begin{bmatrix} R^v(v_n^h, b_n^h, p_n^h, a_n) \\ R^b(v_n^h, b_n^h, p_n^h, a_n) \\ R^p(v_n^h, b_n^h, p_n^h, a_n) \\ R^a(v_n^h, b_n^h, p_n^h, a_n) \end{bmatrix} \quad (\text{I.72})$$

où $(\delta v, \delta b, \delta p, \delta a)^T$ sont les corrections apportées au champ à l'itération n :

$$\begin{cases} v_{n+1}^h = v_n^h + \delta v \\ b_{n+1}^h = b_n^h + \delta b \\ p_{n+1}^h = p_n^h + \delta p \\ a_{n+1}^h = a_n^h + \delta a \end{cases} \quad (\text{I.73})$$

Localement, sur chaque élément Ω_e du maillage, les degrés de liberté liés aux termes bulles peuvent être éliminés (la contribution locale étant identique à la globale). Grâce à cette technique de condensation statique, le système (I.72) devient

$$\begin{bmatrix} K^{vv} & K^{vp} & K^{va} \\ K^{pv} & K^{pp} & 0 \\ K^{av} & 0 & K^{aa} \end{bmatrix} \cdot \begin{bmatrix} \delta v \\ \delta p \\ \delta a \end{bmatrix} = - \begin{bmatrix} R^v(v_n^h, b_n^h, p_n^h, a_n) \\ \tilde{R}^p(v_n^h, b_n^h, p_n^h, a_n) \\ R^a(v_n^h, b_n^h, p_n^h, a_n) \end{bmatrix} \quad (\text{I.74})$$

En décomposant a en vitesses de translation v^o et de rotation ω , et en tenant compte de l'élimination du terme bulle, on aboutit au système suivant :

$$\begin{bmatrix} K^{vv} & K^{vp} & K^{vo} & K^{v\omega} \\ K^{pv} & K^{pp} & 0 & 0 \\ K^{ov} & 0 & K^{oo} & K^{o\omega} \\ K^{\omega v} & 0 & K^{\omega o} & K^{\omega\omega} \end{bmatrix} \cdot \begin{bmatrix} \delta v \\ \delta p \\ \delta v^o \\ \delta \omega \end{bmatrix} = - \begin{bmatrix} R^v \\ \tilde{R}^p \\ R^o \\ R^\omega \end{bmatrix} \quad (\text{I.75})$$

La sous-matrice contenant les termes K^{vv} , $K^{vp} = (K^{pv})^T$, $K^{pp} = -H^{pb} (H^{bb})^{-1} H^{bp}$ est la hessienne habituelle après la condensation de la bulle. Néanmoins, il faut noter que cette sous-matrice n'est pas définie positive. Les zéros apparaissent puisque il n'y a aucune interaction entre la pression (volumique) et la vitesse de l'outil auquel on applique le modèle de presse déformable. Les sous-blocs K^{vo} et $K^{v\omega}$ représentent les contributions du contact et du frottement sur la surface de la pièce en contact avec l'outil rigide, l'écriture précise de ces composantes pour différentes lois de frottement est donnée dans [Boyère 1999, Forestier 2000, Traoré 2001].

I.3.5. Implémentation dans Forge3®

Pour pouvoir prendre en compte les déformations de la presse, l'un des outils rigides est autorisé à avoir des mouvements en translation, ou en rotation, ou les deux. Pour cela, on crée des éléments fictifs associés à cet outil. La création de ces éléments commence par la création d'un nœud associé habituellement au centre de gravité de l'outil choisi. Si l'outil est libre en translation et en rotation, deux nœuds sont alors créés. Physiquement, ils ont les mêmes coordonnées, mais le maillage est tout de même élargi à ces deux nœuds supplémentaires. De cette manière, six vitesses additionnelles associées à ces nœuds fournissent la description des déflexions de la presse recherchée.

Puisque le maillage a été enrichi, de nouveaux éléments sont créés pour connecter chacun de ces nœuds à tous les nœuds de la surface de la pièce en contact avec cet outil (c.f. Figure I.10). Un autre élément « dégénéré » est créé entre les deux nœuds fictifs. Utilisant un élément tétraédrique, celui-ci a donc aussi quatre nœuds, même si en réalité seulement deux nœuds le constituent dans la réalité.

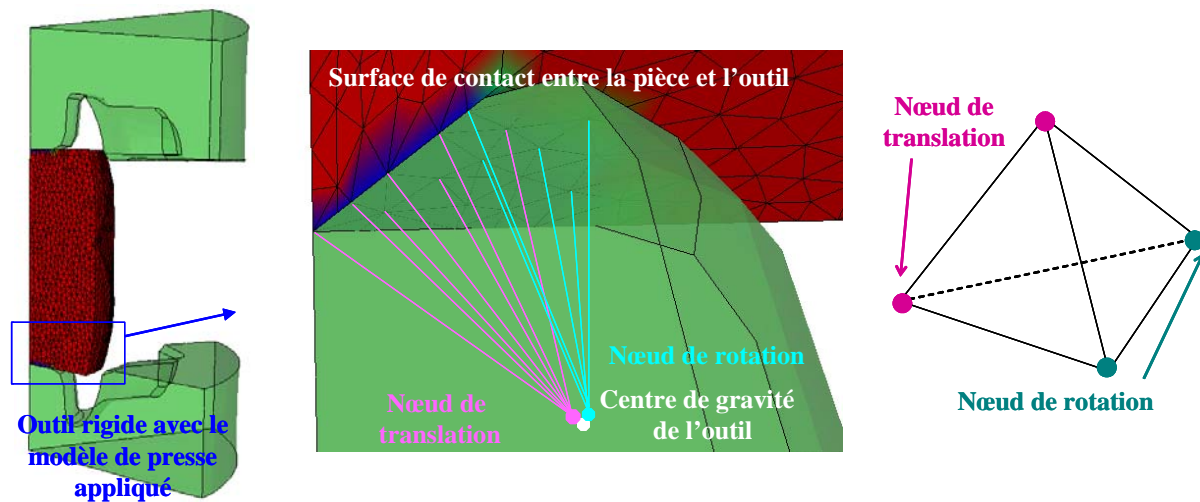


Figure I.10. Création des éléments fictifs

I.3.5.1. Homogénéisation des équations

On remarque que le système d'équations linéaire à résoudre (I.75) n'est pas homogène. La formulation utilisée en Forge3® est mixte en vitesse/pression. Suivant les unités utilisées, les grandeurs relatives des composantes varient, ce qui peut empêcher l'algorithme de converger ou bien faire que l'algorithme converge mieux en vitesse ou en pression. Pour remédier à cela, un changement des variables est effectué et donc la pression est multipliée par le facteur $1/L^2$, où L est une grandeur caractéristique du système. Or, l'introduction du modèle de presse engendre le même type de problème. En effet, si la vitesse linéaire v^o de presse a bien l'unité de vitesse, $[mm/s]$, ω est la vitesse angulaire, son unité est donc de $[1/s]$. On effectue alors un changement de variable suivant :

$$\varpi = \frac{1}{R} \omega \quad (\text{I.76})$$

où R est la grandeur caractéristique de l'outil rigide considéré, par exemple le rayon moyen.

I.3.5.2. Contributions en contact bilatéral

La vitesse de presse donnée par la relation (I.5) s'écrit comme :

$$v_i^{pr} = v_i^O + e_{ijk} \omega_j OM_k \quad (\text{I.77})$$

Le symbole e_{imi} vaut $e_{123} = e_{231} = e_{312} = 1$, $e_{321} = e_{213} = e_{132} = -1$ et zéro dans toutes les autres combinaisons des indices.

À chaque nœud k du maillage, les contributions au gradient du système (I.72) s'écrivent pour chaque composante $i = 1, 2, 3$.

$$R_{ki}^v = \rho_C S_k \left(V_{ki} - v_i^{pr} - \frac{\delta^k}{\Delta t} n_{ki} \right) \quad (\text{I.78})$$

$$R_i^o = -\rho_C \sum_{k \in \partial\Omega_C} S_k \left(V_{ki} - v_i^{pr} - \frac{\delta^k}{\Delta t} n_{ki} \right) \quad (\text{I.79})$$

$$\begin{aligned} R_i^\omega &= -\frac{1}{R} \rho_C \sum_{k \in \partial\Omega_C} S_k \left(V_{kj} - v_j^{pr} - \frac{\delta^k}{\Delta t} n_{kj} \right) e_{jmn} \delta_{mj} OM_n \\ &= -\frac{1}{R} \rho_C \sum_{k \in \partial\Omega_C} S_k \left(V_{kj} - v_j^{pr} - \frac{\delta^k}{\Delta t} n_{kj} \right) e_{jin} OM_n \end{aligned} \quad (\text{I.80})$$

Ici, on a noté ρ_C le coefficient de pénalisation de contact, n_k la normale sortante de l'outil considéré. Le symbole de Kronecker δ_{ij} vaut un si $i = j$ et zéro dans les autres cas.

Afin de simplifier l'écriture, on note $v_{ki} = V_{ki} - v_i^{pr} - \frac{\delta^k}{\Delta t} n_{ki}$. L'expression (I.80) pour le gradient en rotation prend forme :

$$\begin{aligned} R_1^\omega &= \frac{1}{R} \rho_C \sum_{k \in \partial\Omega_{\text{contact}}} S_k (v_{k2} OM_3 - v_{k3} OM_2) \\ R_2^\omega &= \frac{1}{R} \rho_C \sum_{k \in \partial\Omega_{\text{contact}}} S_k (v_{k1} OM_3 - v_{k3} OM_1) \\ R_3^\omega &= \frac{1}{R} \rho_C \sum_{k \in \partial\Omega_{\text{contact}}} S_k (v_{k2} OM_1 - v_{k1} OM_2) \end{aligned} \quad (\text{I.81})$$

Contributions à la matrice hessienne pour les composantes $i, j = 1, 2, 3$ sont :

$$K_{ki,j}^{vv} = \frac{\partial R_{ki}^v}{\partial V_{lj}} = \rho_C S_k \delta_{ki} \delta_{ij} \quad (\text{I.82})$$

$$K_{ki,j}^{vo} = K_{i,kj}^{ov} = \frac{\partial R_{ki}^v}{\partial v_j^o} = \frac{\partial R_i^o}{\partial V_{kj}} = -\rho_C S_k \delta_{ij} \quad (\text{I.83})$$

$$K_{ki,j}^{vo} = \frac{\partial R_{ki}^v}{\partial \omega_j} = \rho_C S_k \frac{\partial v_{ki}}{\partial \omega_j} = -\frac{1}{R} \rho_C S_k e_{imn} \delta_{mj} OM_n = -\frac{1}{R} \rho_C S_k e_{ijn} OM_n \text{ ou}$$

$$K^{vo} = \begin{bmatrix} 0 & -\frac{1}{R} \rho_C S_k OM_3 & \frac{1}{R} \rho_C S_k OM_2 \\ \frac{1}{R} \rho_C S_k OM_3 & 0 & \frac{1}{R} \rho_C S_k OM_1 \\ -\frac{1}{R} \rho_C S_k OM_2 & \frac{1}{R} \rho_C S_k OM_1 & 0 \end{bmatrix} \quad (\text{I.84})$$

$$K_{ij}^{oo} = \frac{\partial R_i^o}{\partial v_j^o} = \rho_C \sum_{k \in \partial \Omega_C} S_k \delta_{ij} \quad (\text{I.85})$$

$$K_{ij}^{oo} = \frac{\partial R_i^o}{\partial \omega_j} = -\rho_C \sum_{k \in \partial \Omega_C} S_k \frac{\partial v_{ki}}{\partial \omega_j} = \frac{1}{R} \rho_C \sum_{k \in \partial \Omega_C} S_k e_{imn} \delta_{mj} OM_n = \frac{1}{R} \rho_C \sum_{k \in \partial \Omega_C} S_k e_{ijn} OM_n \text{ ou}$$

$$K^{oo} = \begin{bmatrix} 0 & \frac{1}{R} \sum_{k \in \partial \Omega_C} \rho_C S_k MO_3 & -\frac{1}{R} \sum_{k \in \partial \Omega_C} \rho_C S_k MO_2 \\ -\frac{1}{R} \sum_{k \in \partial \Omega_C} \rho_C S_k MO_3 & 0 & \frac{1}{R} \sum_{k \in \partial \Omega_C} \rho_C S_k MO_1 \\ \frac{1}{R} \sum_{k \in \partial \Omega_C} \rho_C S_k MO_2 & -\frac{1}{R} \sum_{k \in \partial \Omega_C} \rho_C S_k MO_1 & 0 \end{bmatrix} \quad (\text{I.86})$$

$$K_{ij}^{\omega\omega} = \frac{\partial R_i^{\omega\omega}}{\partial \omega_j} = -\frac{1}{R} \rho_C \sum_{k \in \partial \Omega_C} S_k \frac{\partial v_{kp}}{\partial \omega_j} e_{pin} OM_n = \frac{1}{R^2} \rho_C \sum_{k \in \partial \Omega_C} S_k e_{piq} OM_q e_{pin} OM_n \text{ ou}$$

$$K^{\omega\omega} = \frac{1}{R^2} \rho_C \begin{bmatrix} \sum_{k \in \partial \Omega_C} S_k (OM_2^2 + OM_3^2) & -\sum_{k \in \partial \Omega_C} S_k OM_2 OM_1 & -\sum_{k \in \partial \Omega_C} S_k OM_3 OM_1 \\ -\sum_{k \in \partial \Omega_C} S_k OM_1 OM_2 & \sum_{k \in \partial \Omega_C} S_k (OM_1^2 + OM_3^2) & -\sum_{k \in \partial \Omega_C} S_k OM_3 OM_2 \\ -\sum_{k \in \partial \Omega_C} S_k OM_1 OM_3 & -\sum_{k \in \partial \Omega_C} S_k OM_2 OM_3 & \sum_{k \in \partial \Omega_C} S_k (OM_1^2 + OM_2^2) \end{bmatrix} \quad (\text{I.87})$$

Le même type d'équations est obtenu dans le cas du contact unilatéral avec frottement. Leur écriture étant un peu lourde, nous renvoyons le lecteur à la thèse de K. Traoré [Traoré 2001], où ces équations sont détaillées pour le cas de contact unilatéral et frottement visco-plastique.

1.3.5.3. Contributions au système dues à la raideur de presse

Les contributions dans le gradient du problème global dues au modèle de presse sont :

$$R^o = -F^{pr} - k^{xx} (v^p - v^0) \Delta t - k^{x\theta} (\omega^p - \omega^0) \Delta t \quad (\text{I.88})$$

$$R^{\omega} = -C^{pr} - k^{0x}(v^p - v^0)\Delta t - k^{0\theta}(\omega^p - \omega^0)\Delta t$$

Les contributions dans la matrice hessienne sont les suivantes :

$$\begin{aligned} K^{oo} &= k^{xx}\Delta t \\ K^{o\theta} &= k^{x\theta}\Delta t \\ K^{\omega o} &= k^{0x}\Delta t \\ K^{\omega\theta} &= k^{0\theta}\Delta t \end{aligned} \tag{I.89}$$

Le nombre d'inconnues supplémentaires apportées au système afin de prendre en compte la raideur de presse n'est que de 6.

Un autre aspect du problème est la forme de la matrice de raideur de la presse. Si elle s'avère être non-symétrique, le hessien sera non-symétrique. Cette absence de symétrie peut provenir du manque de précision des essais, et dans ce cas on symétrisera les coefficients. Toutefois, la matrice de raideur peut également être non-symétrique compte tenu de la structure de la presse (par exemple, dans le cas d'une presse à vis). Dans ce cas, on devra recourir au solveur non-symétrique, implémenté en Forge3[®] à travers la librairie PETSc [PETSc].

I.4. VALIDATION PAR RAPPORT AU CAS ANALYTIQUE

Pour valider le modèle de la presse déformable, on considère un cas simple de déformation plane dont on connaît la solution analytique. Un cube $l_0 \times a_0 \times h_0$ (Figure I.11a) est écrasé entre deux outils plans : l'outil supérieur descend avec la vitesse $v_z = v_{out} = 0,5 \text{ mm/s}$. À l'outil inférieur, on applique le modèle de presse avec une seule composante de la matrice $k_{33} = 1,0 \cdot 10^4 \text{ N/mm}$, et sa vitesse v_{pr} est donc inconnue.

Le comportement du matériau est newtonien et décrit par (I.22) avec $m=1$. On considère le contact comme parfaitement glissant, sans frottement, afin d'avoir une solution simple à déterminer. Trois plans de symétrie (xOz), (yOz) et (yl_0z) rendent impossible tout déplacement en x et un déplacement dans la direction négative de l'axe y . Avec les notations introduites sur la Figure I.11b, la vitesse de la presse à l'instant t est donnée par :

$$v^{pr}(t) = \frac{4Kl_0a(t)v^{out} + F^{pr}(t)(h(t) - b(t))}{4Kl_0a(t) - k_{33}\Delta t(h(t) - b(t))} \tag{I.90}$$

En utilisant un schéma d'intégration en temps, on écrit la force de presse comme :

$$\begin{cases} F^{pr}(t + \Delta t) = F^{pr}(t) + k_{33}v^{pr}(t + \Delta t)\Delta t \\ F^{pr}(0) = \frac{4Kl_0\alpha_0v^{out}}{h_0} \end{cases} \quad (I.91)$$

La Figure I.12a montre l'évolution de la vitesse de l'outil inférieur qui subit une translation verticale. Il y est aussi présenté la comparaison entre la vitesse calculée pendant la simulation et celle de la solution analytique. La Figure I.12b présente l'évolution de la force de la presse. Comme on peut le voir, les résultats de la simulation correspondent à la solution analytique, avec tout de même un écart qui s'accumule au cours du temps et qui résulte du schéma temporel utilisé pour calculer la solution analytique. Ceci nous permet donc de valider notre implémentation des équations.

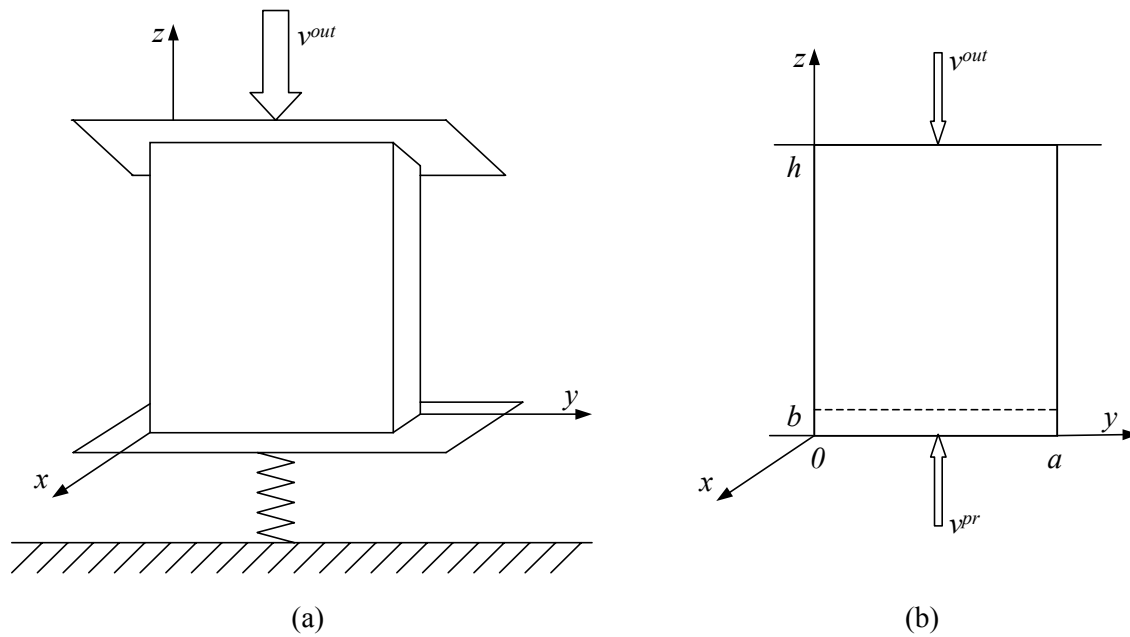
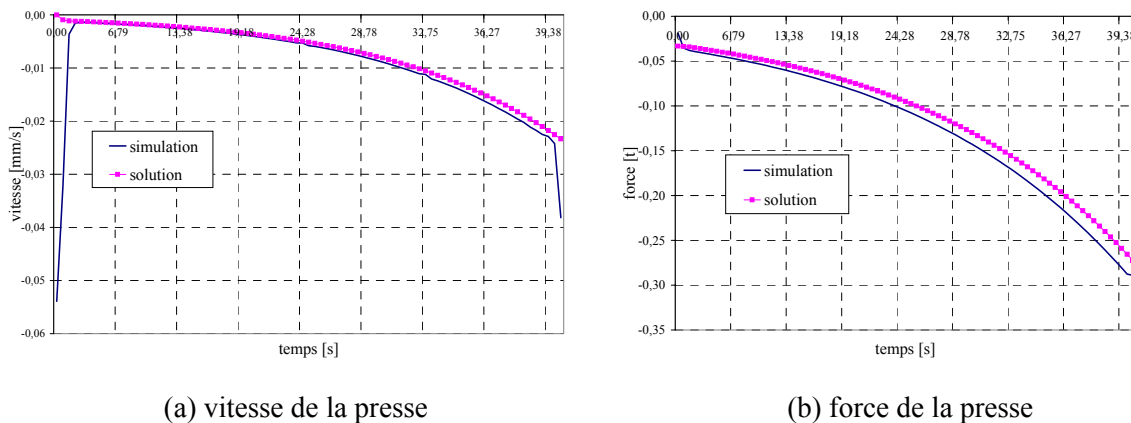


Figure I.11. Forgeage d'un cube : cas test analytique



(a) vitesse de la presse

(b) force de la presse

Figure I.12. Forgeage d'un cube : validation par rapport au cas analytique

I.5. CONCLUSION

Dans ce chapitre, on a introduit un modèle de raideur des presses de forgeage qui se présente comme une généralisation des outils flottants à forces et/ou à couples imposés. Cette généralisation est réalisée à partir des routines concernant les outils flottants, déjà existantes dans le logiciel Forge3[®]. Le modèle proposé a été implémenté dans la version 6.2 du logiciel, ainsi qu'intégré dans la version 7 multi-corps. Le cas d'écrasement d'un cube est étudié. Les résultats issus de la simulation numérique ont été comparés avec un modèle analytique en terme de vitesses et de forces. Une bonne concordance est constatée entre les deux approches ainsi validant l'implémentation du modèle dans le logiciel. La validation numérique de ce modèle pour des applications industrielles de grande taille fera l'objet du cinquième chapitre.

Chapitre II. FORGE3® PARALLÈLE

Dans ce chapitre on présente la méthode de parallélisation du code Forge3. Il s'agit du paradigme S.P.M.D. (Single Program Multiple Data). Cette approche est basée sur le partitionnement du maillage et l'utilisation d'un solveur itératif. Cette méthode peut être aussi qualifiée de globale, puisque elle consiste à résoudre le système linéaire dans son ensemble mais sur des données distribuées, les vecteurs et la matrice étant partagés par les différents processeurs. Le même code marche aussi bien en séquentiel qu'en parallèle, et la solution est la même (aux erreurs d'arrondi près).

II.1. STRATÉGIE S.P.M.D.

La parallélisation du code Forge3 a été effectuée par Stéphane Marie au cours de son doctorat [Marie 1997]. La combinaison d'une méthode de partitionnement et d'un solveur itératif a permis d'adopter une stratégie S.P.M.D. (*Single Program Multiple Data* ou un Seul Programme pour des Multiples Données, cf. Figure II.1) pour rendre Forge3 parallèle.

Le maillage initial est partitionné en plusieurs sous-domaines, leur nombre correspond à N_{proc} , le nombre de processeurs choisi. La version complète du code s'exécute sur chaque processeur, avec comme données initiales une partie du maillage. Certaines données supplémentaires (par rapport à la version séquentielle du code) sont nécessaires pour permettre de décrire les interfaces entre les sous-domaines et échanger les informations entre eux.

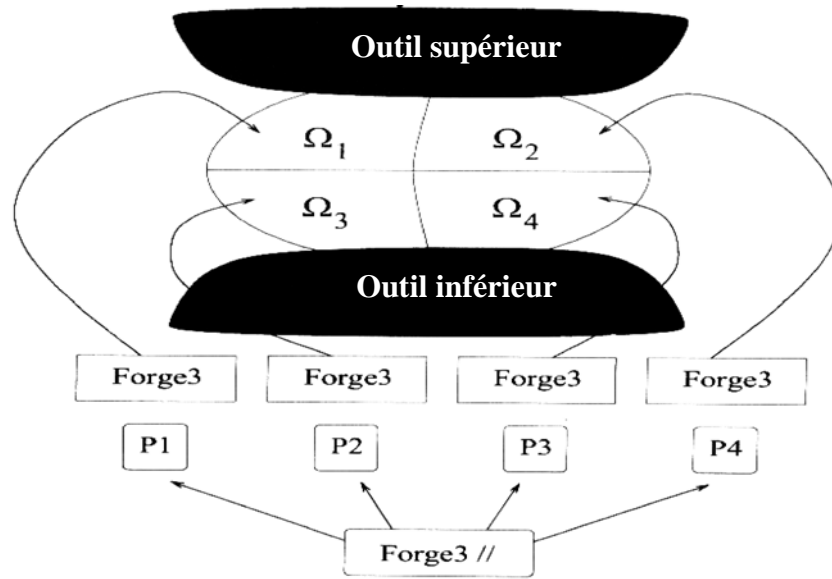


Figure II.1. Stratégie S.P.M.D. de parallélisation de Forge3

Les maillages des outils rigides sont lus par chaque processeur, ainsi que des paramètres définissant leur pilotage et le frottement. En ce qui concerne le modèle de raideur de presse exposé dans le chapitre précédent, il est traité de la même manière. La structure de données de chaque processeur concernant l'outillage est complétée par la matrice de raideur de presse associée à l'un des outils rigides. De cette manière, chaque processeur peut effectuer l'analyse de contact indépendamment des autres (Figure II.2). Au cours de la simulation, chaque processeur construit la matrice et le vecteur de son système à résoudre, tout en tenant compte du contact et du frottement avec les outils rigides. Lors de la résolution des systèmes linéaires locaux, l'échange des parties partagées par les processeurs permet d'obtenir la solution du système global.

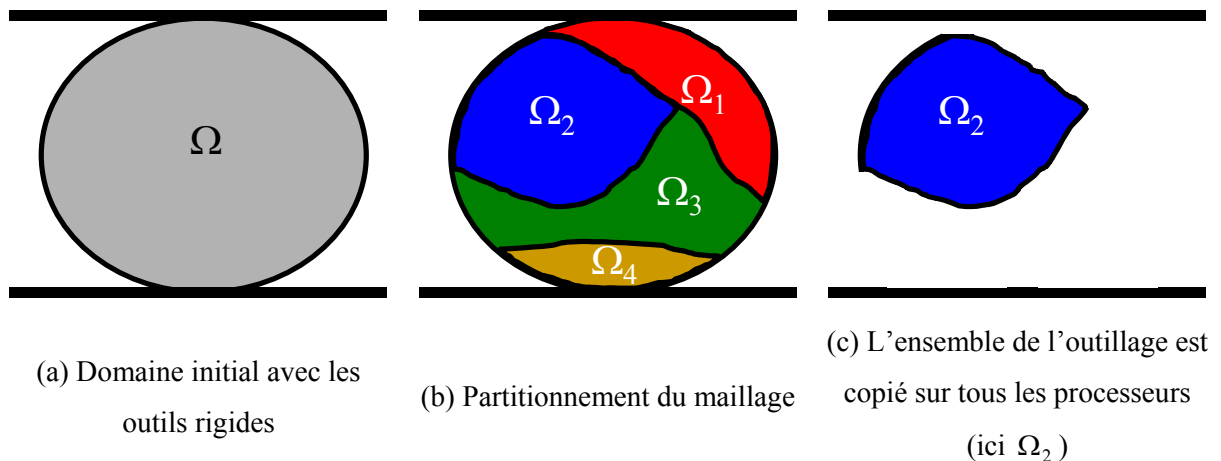


Figure II.2. Analyse de contact effectuée indépendamment par chaque processeur avec l'ensemble de l'outillage (ici deux simples tas plats)

Dans le cas du problème multi-corps, le maillage de chaque corps est partitionné en même nombre de sous-domaines, correspondant au nombre de processeurs. La description des outils rigides est lue par chaque processeur comme dans le cas mono-corps. En revanche, l'enveloppe surfacique du

corps maître (l'équivalent du corps rigide pour l'analyse de contact multi-corps) doit être reconstituée et ensuite recopiée sur l'ensemble des processeurs (Figure II.3). Ainsi, chaque processeur dispose de toutes les informations nécessaires pour effectuer l'analyse de contact. Pour plus de détails sur l'algorithme parallèle de création des éléments de contact, nous renvoyons le lecteur à [Barboza 2004].

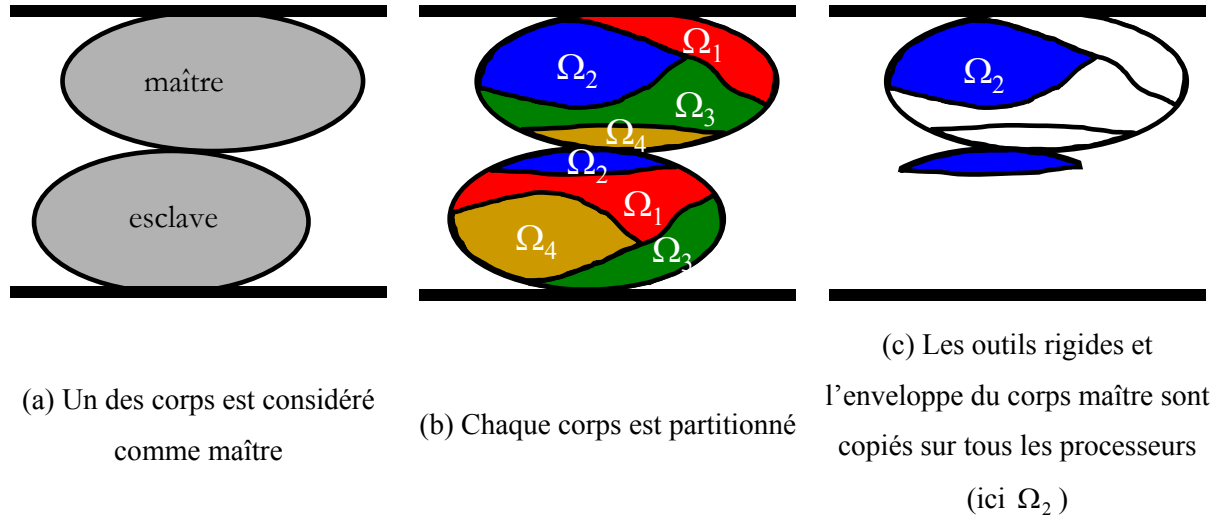


Figure II.3. Analyse de contact : version multi-corps

II.2. PARTITIONNEMENT

II.2.1. Partitionnement du maillage

Soit Ω le maillage discrétisant le corps étudié, on considère une *partition* du maillage en N_{proc} sous-domaines, N_{proc} étant aussi le nombre de processeurs.

$$\Omega = \bigcup_{i=1}^{N_{proc}} \Omega_i \quad (\text{II.1})$$

Cette partition est basée sur les éléments du maillage : chaque élément appartient à un et un seul processeur. En revanche, les nœuds du maillage peuvent être partagés par plusieurs sous-domaines. Ils sont donc représentés plusieurs fois des différents cotés de l'interface, comme montré sur la Figure II.4. D'une manière générale, l'interface Γ_{ij} entre les sous-domaines Ω_i et Ω_j , est définie comme :

$$\Gamma_{ij} = \Gamma_{ji} = \Omega_i \cap \Omega_j \quad (\text{II.2})$$

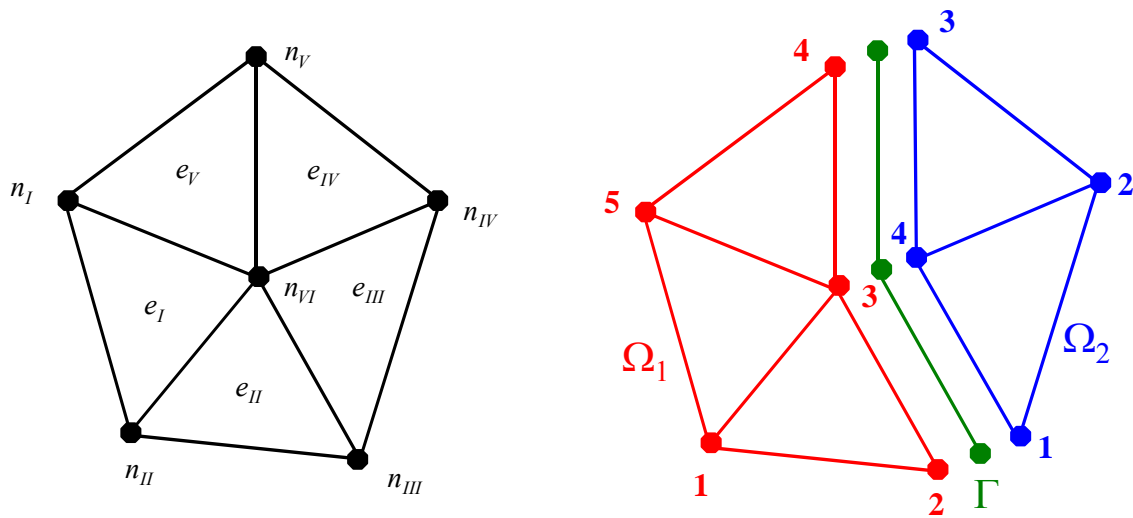
D'autre part, on définit l'interface Γ_i du sous-domaine Ω_i comme :

$$\Gamma_i = \bigcup_{j=1}^{N_{proc}} \Gamma_{ij} \quad (\text{II.3})$$

Et enfin, l'interface globale Γ d'une partition est définie comme l'ensemble des interfaces des sous-domaines :

$$\Gamma = \bigcup_{i=1}^{N_{proc}} \Gamma_i \quad (\text{II.4})$$

La Figure II.4 représente la partition d'un maillage initial constitué de 5 éléments et 6 nœuds, en deux sous-domaines. Chaque sous-domaine a sa propre numérotation locale des nœuds et des éléments. Les nœuds n_{III} , n_{VI} et n_V sont partagés par les deux sous-domaines et constituent l'*interface*. De cette manière, sur le sous-domaine Ω_1 (à gauche et en rouge) le vecteur de l'interface Γ_{12} est représenté dans la numérotation locale comme $(2,3,4)^t$, et sur le sous-domaine Ω_2 (à droite et en bleu) par $(3,4,1)^t$.



(a) Maillage initial avec numérotation globale des nœuds et des éléments

(b) Chaque sous-domaine a sa propre numérotation locale

Figure II.4. Partition d'un maillage par éléments et en deux sous-domaines

La qualité du partitionnement influence les performances du solveur parallèle. Un bon partitionneur doit minimiser la taille des interfaces entre les sous-domaines. Par ailleurs, il doit équilibrer les tâches entre les processeurs, les sous-domaines devant avoir approximativement le même nombre d'éléments. Le partitionneur utilisé dans Forge3 se sert seulement de la topologie du maillage, c'est à dire de la connexion des éléments entre eux. Le partitionnement s'effectue en deux étapes.

Une première partition équilibrée est obtenue par un algorithme dit « glouton ». Pour un nombre donné de processeurs N_{proc} , par chaque sous-domaine le partitionneur détermine un élément aussi éloigné que possible des autres. Cet élément devient le centre du sous-domaine. On attribue une couleur différente à chacun de ces N_{proc} éléments, une couleur représentant un processeur. Ensuite, pour chaque élément colorié, on propage sa couleur à ses éléments voisins non coloriés. Une fois que tous les éléments sont coloriés, on obtient une première partition que l'on améliore ensuite.

La deuxième étape consiste à minimiser les interfaces, donc les communications. La couleur de chaque élément peut être modifiée afin de réduire le nombre de voisins. Un tétraèdre a au plus quatre éléments voisins, et donc de une à quatre couleurs possibles. Il y a donc cinq configurations de couleurs des voisins possibles (Figure II.5a et b) en fonction de la couleur de l'élément considéré. On modifie la couleur du tétraèdre visité afin de minimiser une fonction coût qui prend en compte le nombre d'éléments par couleur (charge des processeurs) et le nombre de communications, c'est à dire le nombre de faces partagées par des éléments de couleurs différentes. L'algorithme itératif se déroule de la manière suivante : on visite un élément du domaine, on optimise les communications locales en modifiant ou non sa couleur, on passe à l'élément suivant non encore visité et qui ne fait pas partie des éléments voisins du tétraèdre que l'on vient de visiter. On procède ainsi jusqu'à ce que tous les éléments aient été visités [Coupez et al 1996].

La procédure décrite est l'algorithme de base. Il en existe des améliorations, tout particulièrement pour rendre cette procédure de partitionnement parallèle [Digonnet 2001].

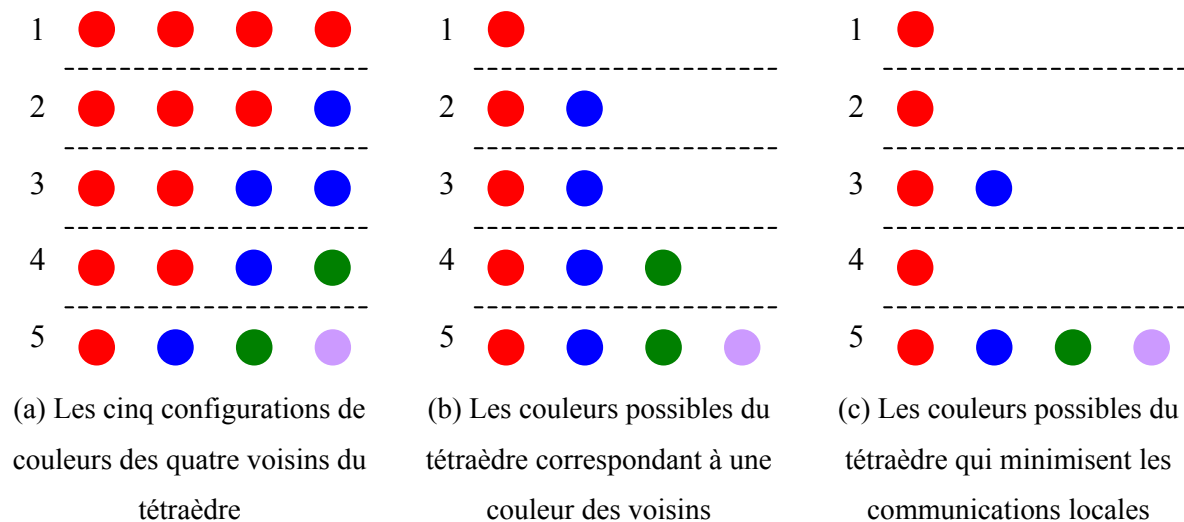


Figure II.5. Minimisation des communications par changement local de couleurs et équilibrage des charges.

II.2.2. Partitionnement des vecteurs et de la matrice

A chaque processeur i , on associe un sous-maillage Ω_i . A la description de la topologie du maillage traditionnelle (numéros des nœuds, des éléments et leurs connectivités) on ajoute alors la structure des interfaces. Chaque sous-domaine dispose des informations suivantes :

- le numéro de ce sous-domaine
- le nombre de sous-domaines voisins
- la liste des sous-domaines voisins
- le nombre de nœuds qu'il partage avec ses voisins

- la liste des numéros locaux de ces nœuds de l'interface pour chaque sous-domaine voisin.

Chaque processeur lit sa partie du maillage initial tout à fait indépendamment des autres, assemble les contributions des éléments lui appartenant, et construit ainsi la matrice locale K_i . Tout ceci est effectué avec la numérotation locale, propre à chaque sous-domaine. Si on reprend l'exemple de la partition montrée sur la Figure II.4, les deux matrices locales construites sur chaque sous-domaine sont représentées sur la Figure II.6a. La matrice globale K (qui aurait été construite sur le domaine global dans la version séquentielle) peut s'écrire comme la somme des matrices locales, préalablement étendues à l'ensemble du maillage :

$$K = \sum_{i=1}^{N_{proc}} \hat{K}_i \quad (\text{II.5})$$

où \hat{K}_i est l'extension de la matrice locale K_i : sa taille est augmentée pour correspondre à la taille de la matrice globale, et d'une numérotation locale propre au Ω_i on passe à la numérotation globale de Ω . L'application de cette stratégie à l'exemple de deux sous-domaines donne la matrice globale de la Figure II.6b.

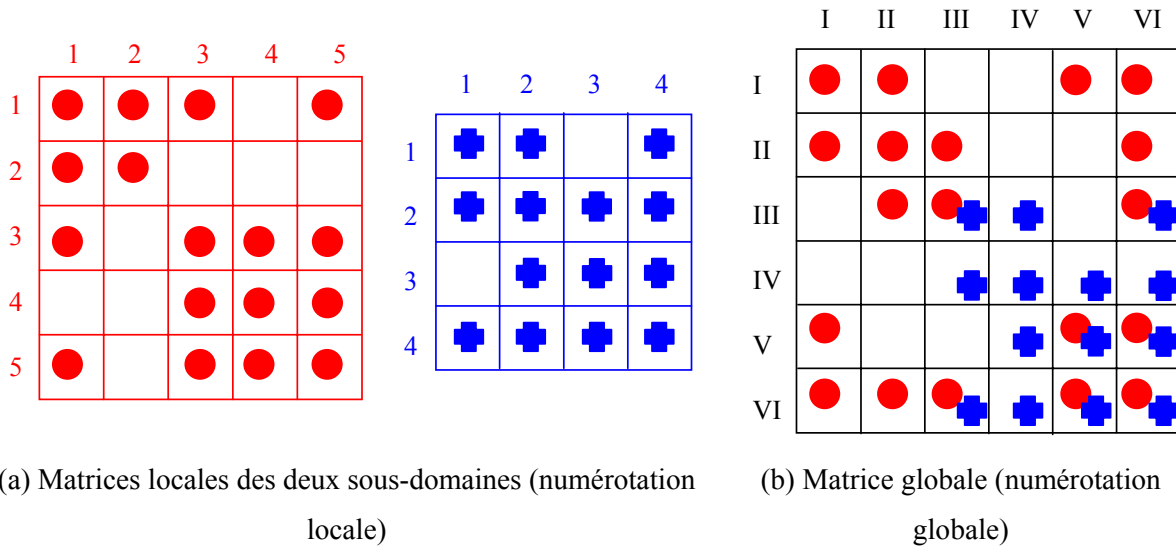


Figure II.6. Contributions de deux sous-domaines dans la matrice globale

Si on rénumérote les degrés de liberté du domaine initial Ω , de manière à distinguer les nœuds intérieurs aux deux sous-domaines Ω_1 et Ω_2 (en rouge et en bleu) de ceux de l'interface Γ (en vert), la matrice globale K a la structure par bloc suivante :

$$K = \begin{bmatrix} K_{11} & 0 & K_{1f} \\ 0 & K_{22} & K_{2f} \\ K_{f1} & K_{f2} & K_{ff} \end{bmatrix} \quad (\text{II.6})$$

où l'indice f désigne les degrés de liberté appartenant à l'interface. La Figure II.7a montre cette sous-structuration de la matrice globale. Il est à noter que l'on a changé l'ordre des numéros des

nœuds par rapport à la Figure II.6 afin de garder les nœuds d'interface à la fin. De la même manière, en numérotant en premier les nœuds strictement intérieurs aux sous-domaines, on peut écrire les matrices locales comme cela est montré sur la Figure II.7b :

$$K_1 = \begin{bmatrix} K_{11} & K_{1f} \\ K_{f1} & K_{ff}^{(1)} \end{bmatrix} \text{ et } K_2 = \begin{bmatrix} K_{22} & K_{2f} \\ K_{f2} & K_{ff}^{(2)} \end{bmatrix} \quad (\text{II.7})$$

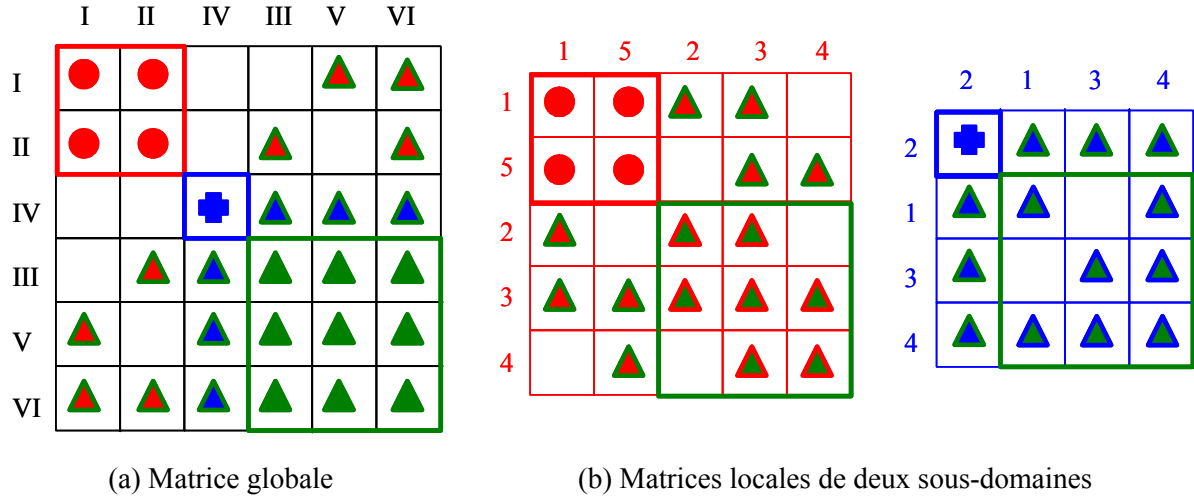


Figure II.7. Renumerotation des nœuds en intérieurs et ceux d'interface

Les coefficients des blocs $K_{ff}^{(1)}$ et $K_{ff}^{(2)}$ représentent respectivement la contribution des sous-domaines Ω_1 et Ω_2 au bloc K_{ff} :

$$K_{ff} = \sum_{i=1}^{Nproc} \hat{K}_{ff}^{(i)} \quad (\text{II.8})$$

Réciproquement, on note x_i la restriction $x_{|\Omega_i}$ d'un vecteur global x au sous-domaine Ω_i , tout en passant à la numérotation locale propre au sous-domaine :

$$x_i = x_{|\Omega_i} \quad (\text{II.9})$$

Sa taille ainsi diminuée, correspond au nombre de nœuds du sous-domaine Ω_i . $x^{(i)}$ étant le sous-vecteur contenant les degrés de liberté intérieurs et $x_f^{(i)}$ ceux de l'interface, la structure par bloc (II.7) introduite pour les matrices est valide pour les vecteurs :

$$x_i = \begin{bmatrix} x^{(i)} \\ x_f^{(i)} \end{bmatrix} \quad (\text{II.10})$$

La construction d'un vecteur global sur le domaine Ω se fait de la même manière que pour les matrices (II.5) en utilisant les extensions $\hat{x}^{(i)}$ des vecteurs locaux et : $x_f = \sum_{i=1}^{Nproc} \hat{x}_f^{(i)}$. Sur deux sous-domaines, l'équivalent de (II.6) pour le vecteur global x est :

$$x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x_f \end{bmatrix} \quad (\text{II.11})$$

II.3. MÉTHODE DE RÉSIDU MINIMAL

La formulation mixte du problème de Stokes généralisé présentée dans le chapitre précédent, stabilisée avec la condensation des degrés de liberté internes à l'élément et linéarisée au sein de l'algorithme de Newton-Raphson, mène à la résolution de systèmes linéaires écrits sous la forme matricielle :

$$Kx = b \quad (\text{II.12})$$

avec

$$K = \begin{bmatrix} A & B \\ B^T & -C \end{bmatrix}, \quad x = \begin{bmatrix} v \\ p \end{bmatrix}, \quad b = \begin{bmatrix} F \\ 0 \end{bmatrix} \quad (\text{II.13})$$

où v représente les variables discrètes en vitesse et p celles en pression. Pour un maillage constitué de $Nbnoe$ nœuds, la dimension du système est de $4 \times Nbnoe$, car on a trois inconnues par nœud en vitesse et une en pression. Les matrices A et C sont définies positives. A cause du bloc matriciel $-C$ le système global, symétrique, n'est pas défini positif.

La méthode du gradient conjugué préconditionnée ne s'applique pas aux systèmes symétriques non définis positifs. Soit $(.,.)$ un produit scalaire de \mathbb{R}^N . Dans la méthode du gradient conjugué, on veut minimiser la fonctionnelle :

$$J(x) = \frac{1}{2}(Kx, x) - (b, x) \quad (\text{II.14})$$

Comme K n'est pas définie positive, la fonctionnelle J n'est pas convexe. En effet, soit \bar{x} la solution du système linéaire et $e = x - \bar{x}$, alors :

$$\exists e \neq 0 \text{ tel que } J(x) - J(\bar{x}) = \frac{1}{2}(Ke, e) < 0$$

et la solution du système linéaire ne réalise donc pas nécessairement le minimum de la fonctionnelle J .

Aussi, une méthode de type résidu minimal préconditionné est utilisée dans le code Forge3 pour la résolution de ce système linéaire (voir la thèse [Marie 1997]). On minimise alors la norme du vecteur résidu :

$$E(r) = (r, r) = (b - Kx, b - Kx) \quad (\text{II.15})$$

Pour accélérer la convergence, on introduit une matrice de préconditionnement Q inversible, symétrique et *a priori* définie positive pour améliorer le conditionnement. Soit une matrice régulière Q_1 telle que $Q = Q_1 Q_1^T$ soit symétrique définie positive. Alors, on applique la méthode du résidu minimal au système préconditionné :

$$\begin{cases} Q_1^{-1} K Q_1^{-T} y = Q_1^{-1} b \\ y = Q_1^T x \end{cases} \quad (\text{II.16})$$

ce qui équivaut à minimiser la fonctionnelle convexe :

$$E(r) = (Q^{-1} r, r) = (Q_1^{-1} r, Q_1^{-1} r) \quad (\text{II.17})$$

A chaque itération k de la méthode, on détermine un **vecteur de descente** w^k et un scalaire α^k pour calculer la solution à l'itération suivante x^{k+1} :

$$x^{k+1} = x^k + \alpha^k w^k \quad (\text{II.18})$$

On définit le résidu à l'itération k :

$$r^k = b - K x^k \quad (\text{II.19})$$

et ensuite pour $k > 0$ la mise à jour du résidu s'effectue comme :

$$r^{k+1} = r^k - \alpha^k K w^k \quad (\text{II.20})$$

La condition de minimisation de $E(r^{k+1})$ définit le coefficient optimal α^k pour r^k et w^k donnés :

$$\alpha^k = \frac{(r^k, Q^{-1} K w^k)}{(Q^{-1} K w^k, K w^k)} \quad (\text{II.21})$$

Enfin, on détermine la direction de descente optimale w^{k+1} de façon à minimiser $E(r^{k+2})$ pour le résidu suivant. On la recherche dans le plan formé par les deux directions r^{k+1} et w^k , avec un paramètre d'optimisation de descente β^k :

$$w^{k+1} = Q^{-1} r^{k+1} + \beta^k w^k \quad (\text{II.22})$$

$$\beta^k = - \frac{(Q^{-1} K w^k, K Q^{-1} r^{k+1})}{(Q^{-1} K w^k, K w^k)} \quad (\text{II.23})$$

Cette méthode peut être généralisée à un nombre quelconque de directions de descente. Dans ce cas, la méthode s'appelle OrthoMin(s), et on cherche la direction de descente comme une combinaison linéaire de r^{k+1} et des vecteurs w^j ($j \leq k$) précédents. Pour des raisons de stockage mémoire, lors de l'implémentation de l'algorithme, cette combinaison linéaire est tronquée aux $(s+1)$ pentes précédentes :

$$w^{k+1} = Q^{-1}r^{k+1} + \sum_{j=k-s_k}^k \beta_k^j w^j \text{ avec } 0 \leq s_k \leq \min(k, s) \text{ et } s \geq 0 \quad (\text{II.24})$$

$$\beta_k^j = -\frac{(Q^{-1}Kw^j, KQ^{-1}r^{k+1})}{(Q^{-1}Kw^j, Kw^j)} \text{ avec } k - s_k \leq j \leq k \quad (\text{II.25})$$

Pour $s = 0$ on retrouve la méthode du résidu minimal.

Une autre méthode plus robuste est la méthode OrthoDir. On recherche alors la direction de descente w_{k+1} dans le $(k+1)^{\text{ème}}$ espace de Krylov $\{Q^{-1}r^0, Q^{-1}KQ^{-1}r^0, \dots, (Q^{-1}K)^k Q^{-1}r^0\}$ en posant :

$$w^{k+1} = Q^{-1}Kw^k + \sum_{j=0}^k \gamma_k^j w^j \text{ avec } w^0 = Q^{-1}r^0 \quad (\text{II.26})$$

Avec la condition d'orthogonalité :

$$(Q^{-1}Kw^{k+1}, Kw^j) = 0 \text{ où } \forall j \quad 0 \leq j \leq k \quad (\text{II.27})$$

on déduit les paramètres de descente :

$$\gamma_k^j = -\frac{(Q^{-1}KQ^{-1}Kw^k, Kw^j)}{(Q^{-1}Kw^j, Kw^j)} \text{ avec } 0 \leq j \leq k. \quad (\text{II.28})$$

En pratique, une version tronquée de cette méthode est retenue où l'on ne considère que les $s+1$ pentes précédentes.

II.4. PARTICULARITÉS PARALLÈLES

Les différences majeures entre l'implémentation de la méthode du résidu conjugué préconditionné en séquentiel et en parallèle se résument dans les trois opérations algébriques suivantes.

II.4.1. Produits matrice-vecteur

Sur chaque sous-domaine Ω_i on calcule indépendamment le produit de la matrice locale et d'un vecteur local :

$$y_i = K_i x_i \quad (\text{II.29})$$

Ce vecteur y_i n'est qu'une contribution du sous-domaine Ω_i au produit global $y = Kx$ que l'on cherche à connaître. Pour l'obtenir, il faut rassembler les contributions de tous les sous-domaines. Une fois ce produit global connu, on pourra le restreindre sur chaque sous-domaine comme dans (II.9).

Pour rassembler les différentes contributions, la procédure consiste alors à faire :

1) chaque sous-domaine Ω_i envoie les valeurs $y_f^{(i)}$ des degrés de liberté de l'interface (II.3) aux processeurs qui sont en charge des sous-domaines voisins,

2) chaque sous-domaine Ω_i reçoit les valeurs $y_f^{(j)}$ des degrés de liberté des interfaces de ses voisins (II.2) et les ajoute au vecteur local y_i .

Il faut noter que ces contributions locales $y_f^{(i)}$ et $y_f^{(j)}$ des sous-domaines Ω_i et Ω_j ne sont pas les mêmes. En effet, chacune d'entre elles représente une partie du produit local réduit à l'interface $y_f^{(i)} = y_i|_{\Gamma_y} = K_i x_i|_{\Gamma_y}$ et $y_f^{(j)} = y_j|_{\Gamma_\mu} = K_j x_j|_{\Gamma_\mu}$. Alors, sur sous-domaine Ω_i le vecteur local ainsi complété est la restriction du vecteur global :

$$y_i = \begin{bmatrix} y_f^{(i)} \\ y_f^{(i)} + \sum_{j=1}^{N_{proc}} y_f^{(j)} \end{bmatrix} \quad (\text{II.30})$$

Cette technique peut être appliquée à un nombre quelconque de sous-domaines. Si un nœud appartient à plusieurs interfaces, les contributions locales sont envoyées à tous les processeurs partageant ce nœud, et en retour le même nombre de contributions extérieures est reçu. Ainsi, les points multiples de la partition ne nécessitent pas un traitement spécial.

Un autre aspect important est qu'il n'est pas nécessaire de rénuméroter les nœuds en tant que nœuds intérieurs et nœuds d'interface, comme dans la section II.2.2. Il suffit de savoir trouver, parmi tous les nœuds du sous-domaine, ceux qui constituent l'interface.

II.4.2. Produits scalaires

Puisque les vecteurs globaux sont partagés entre les processeurs, le résultat du produit scalaire global doit être calculé comme la somme des contributions des sous-domaines. Dans cette procédure, il faut prêter attention au fait que les nœuds d'interface s'ajoutent plusieurs fois.

En effet, si on reprend l'exemple de deux sous-domaines, le produit scalaire local des vecteurs x et y s'écrit

$$\begin{aligned} (x_1, y_1) &= \begin{bmatrix} x^{(1)} \\ x_f \end{bmatrix} \cdot \begin{bmatrix} y^{(1)} \\ y_f \end{bmatrix} = (x^{(1)}, y^{(1)}) + (x_f, y_f) \\ (x_2, y_2) &= \begin{bmatrix} x^{(2)} \\ x_f \end{bmatrix} \cdot \begin{bmatrix} y^{(2)} \\ y_f \end{bmatrix} = (x^{(2)}, y^{(2)}) + (x_f, y_f) \end{aligned} \quad (\text{II.31})$$

La somme des deux contributions locales sur Ω_1 et Ω_2 donne :

$$(x^{(1)}, y^{(1)}) + (x^{(2)}, y^{(2)}) + 2 \cdot (x_f, y_f) \quad (\text{II.32})$$

On voit que les contributions des nœuds d'interface sont prises en compte deux fois. Dans le cas plus général d'une partition multi-domaines, les contributions des nœuds d'interface sont sommées autant de fois que le nombre de sous-domaines auxquels ces nœuds appartiennent. Pour remédier à cette redondance, pour chaque nœud, on introduit un facteur de pondération, et le produit scalaire global prend la forme :

$$(x, y) = \sum_{i=1}^{N_{proc}} (x_i, y_i) = \sum_{i=1}^{N_{proc}} \sum_{n=1}^{Nbnoe(j)} (x_i(n), y_i(n)) \cdot pond(n) \quad (\text{II.33})$$

où $Nbnoe(j)$ est le nombre de nœuds du sous-domaine Ω_j , (x_i, y_i) est le produit scalaire local, $pond(n)$ est le facteur de pondération du nœud n , il est égal à l'inverse du nombre de sous-domaines auxquels ce nœud appartient.

Ce calcul du produit scalaire global nécessite une communication globale entre tous les processeurs pour faire la somme de toutes les contributions.

II.4.3. Calculs des normes infinies des vecteurs

Le calcul des normes inf et sup sur un vecteur y s'écrit :

$$\inf_{y \in \Omega} y = \inf_{i=1, N_{proc}} \left(\inf_{y_i \in \Omega_i} y_i \right) \quad (\text{II.34})$$

$$\sup_{y \in \Omega} y = \sup_{i=1, N_{proc}} \left(\inf_{y_i \in \Omega_i} y_i \right) \quad (\text{II.35})$$

Ces opérations nécessitent une communication globale et donc imposent une synchronisation de tous les processeurs.

II.4.4. Algorithme parallèle

L'algorithme parallèle de résolution du système linéaire ressemble beaucoup à la version séquentielle, à l'exception de deux types d'échange des données nécessaires entre les processeurs :

- Échange des données d'interface entre les sous-domaines adjacents pour rassembler les contributions au produit matrice-vecteur,
- Communication globale entre les processeurs pour ajouter les contributions au produit scalaire.

Ainsi, à l'itération $k+1$ de la méthode du résidu minimal parallèle, on fait des opérations suivantes :

1. produit matrice vecteur avec assemblage des contributions en provenance des sous-domaines adjacents :

$$z^k = Kp^k \quad (\text{II.36})$$

2. deux produits scalaires pour trouver le coefficient optimal de descente :

$$\alpha^k = \frac{(r^k, z^k)}{(z^k, z^k)} \quad (\text{II.37})$$

3. deux opérations purement locales pour mettre à jour la solution et le résidu :

$$x^{k+1} = x^k + \alpha^k w^k \quad (\text{II.38})$$

$$r^{k+1} = r^k - \alpha^k z^k \quad (\text{II.39})$$

4. un produit matrice vecteur Kr^{k+1} et un produit scalaire (Kr^{k+1}, z^k) pour trouver le coefficient de conjugaison :

$$\beta^k = -\frac{(Kr^{k+1}, z^k)}{(z^k, z^k)} \quad (\text{II.40})$$

5. deux opérations purement locales pour mettre à jour la direction de descente et le vecteur supplémentaire :

$$w^{k+1} = r^{k+1} + \beta^k w^k \quad (\text{II.41})$$

$$z^{k+1} = Kr^{k+1} + \beta^k z^k \quad (\text{II.42})$$

II.4.5. Prise en compte de la raideur de presse

L'implémentation du modèle de presse exposé dans le chapitre précédant est aisé. Lors de la lecture des données initiales, chaque processeur lit la matrice de raideur de presse parmi d'autres données relatives aux outils rigides. Ainsi, chacun est capable de créer des nœuds fictifs associés à l'outil avec le modèle de presse et construire des éléments nécessaires pour la prise en compte de la raideur de presse. Il suffit alors de rajouter ces nœuds fictifs à la liste des nœuds d'interface pour leur affecter le poids adéquat et les prendre en compte lors de la résolution.

Chapitre III. MÉTHODES DE DÉCOMPOSITION DE DOMAINE

III.1. PRINCIPE DE BASE

La famille des méthodes de décomposition de domaine est souvent utilisée pour résoudre les systèmes linéaires en parallèle. Ces méthodes se présentent comme une combinaison de deux techniques de calcul : les résolutions directe et itérative. D'une part, un algorithme itératif est utilisé pour résoudre le problème condensé aux interfaces entre les sous-domaines. Il nécessite de nombreuses communications entre les processeurs. D'autre part, à chaque itération, un système linéaire propre au sous-domaine est résolu de manière directe et indépendamment des autres processeurs.

Si le domaine Ω est décomposé en deux sous-domaines Ω_1 et Ω_2 ayant pour interface Γ (Figure III.1), un vecteur global x est solution du système linéaire $Kx = b$ construit sur Ω , si et seulement si les restrictions x_1 et x_2 sur chaque sous-domaine sont respectivement solutions des systèmes suivants sur Ω_1 et Ω_2 :

$$K_i x_i = b_i \quad i = 1, 2, \quad (\text{III.1})$$

et satisfont les conditions de continuité sur l'interface Γ :

$$x_1 - x_2 = 0 \quad \text{sur } \Gamma \quad (\text{III.2})$$

$$\frac{\partial x_1}{\partial n_1} + \frac{\partial x_2}{\partial n_2} = 0 \quad \text{sur } \Gamma \quad (\text{III.3})$$

où n_i est la normale sortante du sous-domaine Ω_i (Figure III.1).

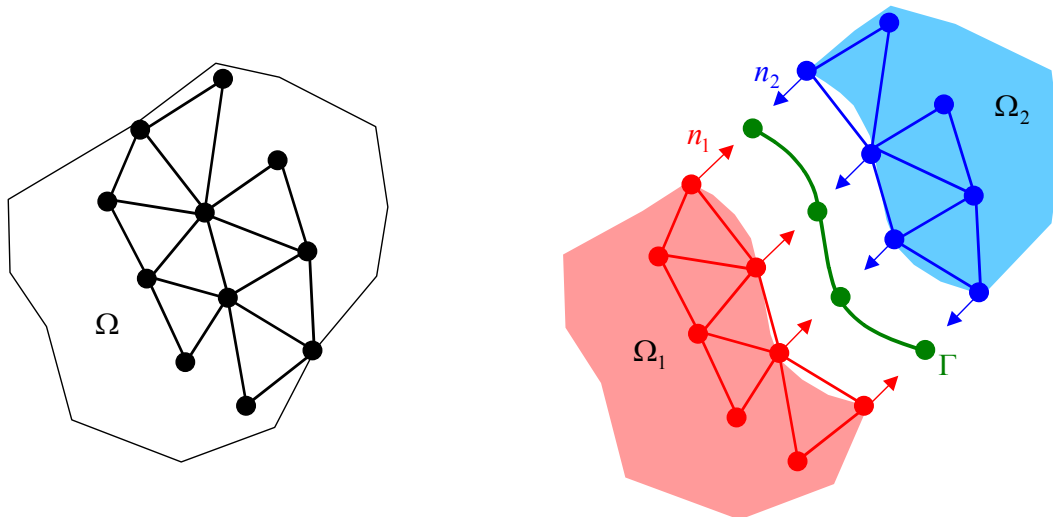


Figure III.1. Décomposition du domaine Ω en deux sous-domaines Ω_1 et Ω_2

Le problème à résoudre consiste alors à trouver les solutions locales x_1 et x_2 du système (III.1) en tenant compte des conditions (III.2) et (III.3). Les problèmes locaux (III.1) sont bien posés si l'une des deux inconnues, x ou $\partial x/\partial n$, est fixée sur l'interface. Ainsi, la plupart des méthodes de décomposition de domaine consistent à effectuer des itérations sur l'une des conditions (III.2) ou (III.3). On peut résoudre ce problème de plusieurs manières, notamment :

- en considérant (III.1), (III.2) et (III.3) comme un problème en x_i et $\partial x_i/\partial n_i$ ce qui conduit à une formulation à trois champs [Brezzi et Marini 1993],
- en choisissant comme inconnue du système la restriction du vecteur x sur l'interface et en vérifiant ainsi *a priori* la continuité (III.2). On obtient alors un système en $x_f = x_f^{(1)} = x_f^{(2)}$: la restriction des vecteurs sur l'interface. Cette approche est la base de la méthode primale de décomposition de domaine, connue comme la méthode du complément de Schur [Tallec 1994], [Chan et Mathew 1994], [Tallec et al 1998], ou *balancing method* [Mandel 1993], [Goldfeld et al 2003], [Pavarino et Widlund 2002],
- en vérifiant *a priori* la condition (III.3), le problème (III.1), (III.2) se pose alors en λ , le multiplicateur de Lagrange associé à la condition (III.2) et identifié à la dérivée normale des champs x sur l'interface. Cette approche est la base de la méthode duale de décomposition de domaine, connue dans la littérature comme la méthode FETI [Farhat et Roux 1991], [Farhat et Roux 1994],
- en combinant les deux approches précédentes, on obtient alors un système en (x_f, λ) . Suivant la stratégie de la résolution, cette méthode conduit à l'approche FETI-DP [Farhat et al 2000b], [Farhat et al 2001], [Mandel et Tezaur 2001], ou à l'approche hybride de décomposition de domaine [Gosselet et al 2002], [Vereecke et al 2003], [Li 2003].

III.2. MÉTHODE DU COMPLÉMENT DE SCHUR

Comme décrit dans la section précédente, l'approche primale consiste à rechercher le champ x_f de l'interface. On réécrit le système global :

$$\begin{bmatrix} K_{11} & 0 & K_{1f} \\ 0 & K_{22} & K_{2f} \\ K_{f1} & K_{f2} & K_{ff} \end{bmatrix} \cdot \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x_f \end{bmatrix} = \begin{bmatrix} b^{(1)} \\ b^{(2)} \\ b_f \end{bmatrix} \quad (\text{III.4})$$

De manière à vérifier automatiquement la condition de continuité (III.2), on a naturellement introduit une unique inconnue à l'interface, x_f . On utilise ici la notation introduite dans le chapitre précédent, section II.2.2, où l'indice f désigne l'interface et où 1 et 2 représentent les quantités propres à l'intérieur des sous-domaines.

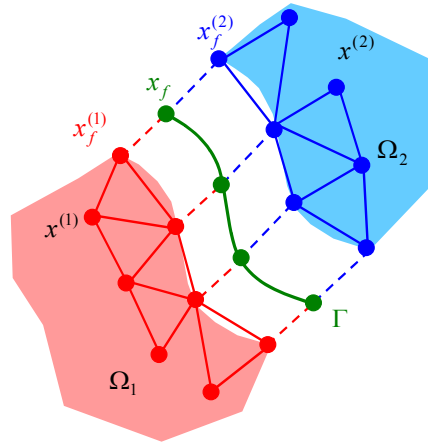


Figure III.2. Méthode primale : description de l'interface

Les inconnues correspondant aux degrés de liberté intérieurs à chaque sous-domaine peuvent être éliminées en résolvant le problème local en fonction des inconnues d'interface :

$$\begin{aligned} K_{11}x^{(1)} &= b^{(1)} - K_{1f}x_f \Rightarrow x^{(1)} = K_{11}^{-1}(b^{(1)} - K_{1f}x_f) \\ K_{22}x^{(2)} &= b^{(2)} - K_{2f}x_f \Rightarrow x^{(2)} = K_{22}^{-1}(b^{(2)} - K_{2f}x_f) \end{aligned} \quad (\text{III.5})$$

L'inversion des matrices K_{ii} correspond à la résolution des problèmes d'équilibre où l'interface est soumise à des conditions de Dirichlet, le déplacement x_f étant imposé. En utilisant (III.5) dans la dernière ligne de (III.4), on obtient :

$$\left(K_{ff} - K_{f1}K_{11}^{-1}K_{1f} - K_{f2}K_{22}^{-1}K_{2f} \right) x_f = b_f - K_{f1}K_{11}^{-1}b^{(1)} - K_{f2}K_{22}^{-1}b^{(2)} \quad (\text{III.6})$$

Avec les notations suivantes :

$$S = K_{ff} - \sum_{i=1}^2 K_{fi} K_{ii}^{-1} K_{if} = \sum_{i=1}^2 [K_{ff}^{(i)} - K_{fi} K_{ii}^{-1} K_{if}] = \sum_{i=1}^2 S^{(i)*} \quad (\text{III.7})$$

$$\tilde{b} = b_f - \sum_i K_{fi} K_{ii}^{-1} b^{(i)} = \sum_i (b_f^{(i)} - K_{fi} K_{ii}^{-1} b^{(i)}) = \sum_i \tilde{b}^{(i)} \quad (\text{III.8})$$

on formule le problème condensé aux interfaces de la méthode du complément de Schur de la manière suivante :

$$S \cdot x_f = \tilde{b} \quad (\text{III.9})$$

où la matrice S définie par (III.7) est nommée la matrice du complément de Schur. La taille du problème (III.9) correspond exactement au nombre de degrés de liberté de l'interface. Ce système (III.9) est résolu par une méthode itérative, en utilisant à chaque itération, le calcul des solutions locales exactes (III.5).

Puisque l'on utilise une méthode itérative pour résoudre le système (III.9), il est nécessaire d'effectuer des produits par la matrice du complément de Schur S . Or, comme on peut le constater, cette matrice S est assemblée à partir des matrices locales $S^{(i)}$ (III.7). Ainsi, le calcul de ce produit global se fait par l'assemblage des produits locaux. Pour un vecteur d'interface w_f connu à l'itération considérée, on calcule le produit $S^{(i)} \cdot w_f$ par la matrice de Schur locale en deux étapes :

1) Résolution du problème local de Dirichlet :

$$\begin{bmatrix} K_{ii} & K_{if} \\ K_{fi} & K_{ff}^{(i)} \end{bmatrix} \cdot \begin{bmatrix} w^{(i)} \\ w_f \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ ou encore } K_{ii} w^{(i)} = -K_{if} w_f \quad (\text{III.10})$$

Il est à noter qu'une fois que la matrice locale K_{ii} a été factorisée, la résolution de ce problème (III.10) se fait très rapidement par une méthode directe et un algorithme de descente/remontée. La solution, décomposée en degrés de liberté internes au sous-domaine, $w^{(i)}$, et ceux de l'interface w_f , peut se mettre sous la forme :

$$\begin{bmatrix} w^{(i)} \\ w_f \end{bmatrix} = \begin{bmatrix} -K_{ii}^{-1} K_{if} w_f \\ w_f \end{bmatrix} \quad (\text{III.11})$$

2) La restriction du produit matrice-vecteur suivant donne exactement les valeurs recherchées de $S^{(i)} \cdot w_f$ à l'interface :

$$\begin{bmatrix} K_{ii} & K_{if} \\ K_{fi} & K_{ff}^{(i)} \end{bmatrix} \cdot \begin{bmatrix} w^{(i)} \\ w_f \end{bmatrix} = \begin{bmatrix} K_{ii} & K_{if} \\ K_{fi} & K_{ff}^{(i)} \end{bmatrix} \cdot \begin{bmatrix} -K_{ii}^{-1} K_{if} w_f \\ w_f \end{bmatrix} = \begin{bmatrix} 0 \\ S^{(i)} \cdot w_f \end{bmatrix} \quad (\text{III.12})$$

* Dans la suite, sauf une note explicite et afin d'alléger les équations, on utilisera par défaut la sommation sur l'indice qui parcourt tous les sous-domaines.

où $S^{(i)} \cdot w_f$ est donc égal à :

$$S^{(i)} \cdot w_f = K_{ff}^{(i)} w_f - K_{fi} K_{ii}^{-1} K_{if} w_f \quad (\text{III.13})$$

Enfin, il suffit d'assembler les contributions locales $S^{(i)} \cdot w_f$ au produit global, comme cela a été montré dans le chapitre précédent, section II.4.1. En utilisant cette procédure, on n'a pas besoin de construire explicitement la matrice globale S .

L'extension de cette méthode à un nombre quelconque de sous-domaines (supérieur à deux) se fait facilement.

III.3. MÉTHODE FETI

La méthode FETI (*Finite Element Tearing and Interconnecting*) est aussi connue dans la littérature comme la méthode du complément de Schur duale [Vanderstraeten and Keunings 1998]. En effet, la condition de continuité (III.2) est imposée à travers les multiplicateurs de Lagrange λ . L'interprétation mécanique de cette méthode est assez simple : en découpant le domaine, on cherche à la fois l'équilibre de chaque sous-domaine indépendamment des autres et les efforts λ nécessaires à l'interface pour équilibrer la structure entière (Figure III.3).

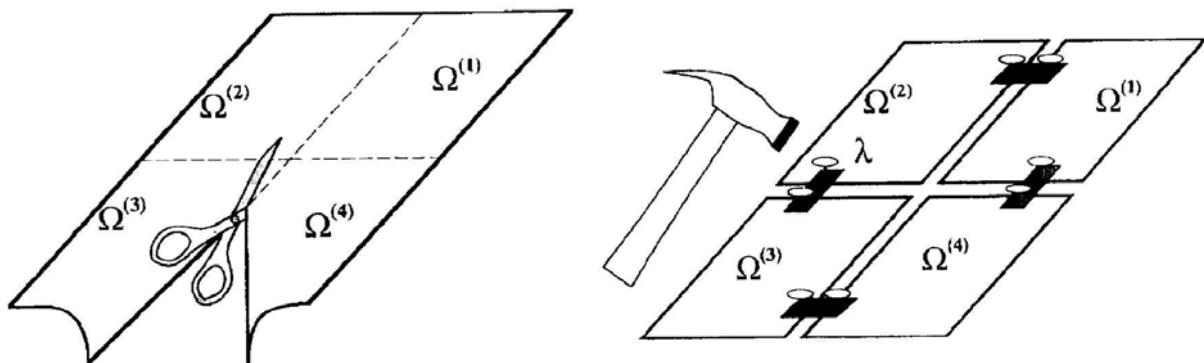
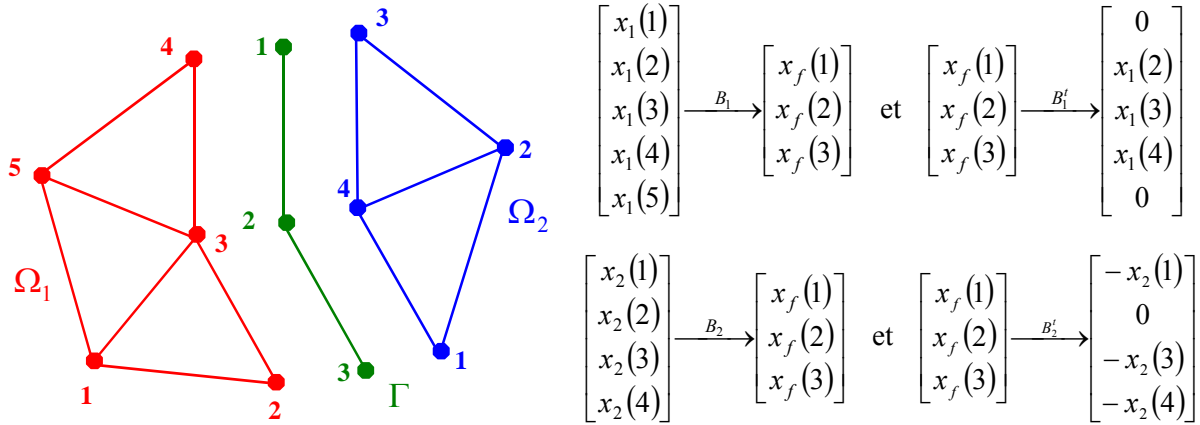


Figure III.3. Méthode FETI : *tearing and interconnecting*

Sur chaque sous-domaine Ω_i , on introduit les opérateurs de trace, B_i , qui construisent les restrictions des vecteurs locaux x_i (définis sur chaque sous-domaines, interface incluse, voir section II.2.2, (II.9)) sur l'interface. L'opérateur transposé B_i^t injecte le vecteur de l'interface $x_f^{(i)}$ sur le sous-domaine entier ; on obtient alors un vecteur local de forme $\begin{bmatrix} 0 & x_f^{(i)} \end{bmatrix}$ (voir Figure III.4 où l'on précise sur un exemple le changement de numérotation associé à l'application de ces opérateurs) :

$$B_i \begin{bmatrix} x^{(i)} \\ x_f^{(i)} \end{bmatrix} = x_f^{(i)} \quad \text{et} \quad B_i^t x_f^{(i)} = \begin{bmatrix} 0 \\ x_f^{(i)} \end{bmatrix} \quad (\text{III.14})$$

Figure III.4. Méthode FETI : opérateurs de trace B_i

Avec ces notations, le système à résoudre sur deux sous-domaines s'écrit sous la forme :

$$\begin{cases} K_1 x_1 = b_1 + B_1^t \lambda \\ K_2 x_2 = b_2 + B_2^t \lambda \\ B_1 x_1 + B_2 x_2 = 0 \end{cases} \quad (\text{III.15})$$

La troisième équation de (III.15) exprime justement la condition de continuité des solutions à travers l'interface (III.2). Pour ce faire, les opérateurs B_i doivent être signés, de sorte que B_{ij} , la restriction de B_i sur Γ_{ij} , ait le signe opposé à B_{ji} , la restriction de B_j sur Γ_{ji} . En outre, dans les deux premières équations, $B_i^t \lambda$ exprime l'effort exercé par le sous-domaine Ω_2 sur Ω_1 et inversement, mais l'effort λ est le même, il change seulement le signe.

En injectant les solutions des deux premières équations dans la troisième (on suppose que les matrices K_i sont non-singulières), on obtient le problème condensé à l'interface :

$$\left[\sum_i B_i K_i^{-1} B_i^t \right] \lambda = - \sum_i B_i K_i^{-1} b_i \quad (\text{III.16})$$

et on note

$$D = \sum_i B_i K_i^{-1} B_i^t \quad (\text{III.17})$$

la matrice du complément de Schur duale. Pour une valeur de λ^k à l'itération k des forces d'interaction entre les sous-domaines, le résidu du problème (III.16) se met sous la forme :

$$(B_1 K_1^{-1} B_1^T + B_2 K_2^{-1} B_2^T) \lambda^k + B_1 K_1^{-1} b_1 + B_2 K_2^{-1} b_2 = B_1 x_1^k + B_2 x_2^k \quad (\text{III.18})$$

Ce résidu est égal au saut à travers l'interface des solutions x_i^k des problèmes de Neumann locaux :

$$\begin{cases} K_1 x_1^k = b_1 + B_1' \lambda^k \\ K_2 x_2^k = b_2 + B_2' \lambda^k \end{cases} \quad (\text{III.19})$$

Lorsque le gradient (III.18) est nul, les valeurs de x_i de deux cotés de l'interface sont les mêmes. Le problème condensé à l'interface (III.16) est résolu par une méthode itérative, comme par exemple la méthode du gradient conjugué ou celle du résidu minimal. À chaque itération, les problèmes locaux de Neumann sont résolus par une méthode directe afin de calculer le saut des solutions et donc le nouveau résidu.

III.4. MÉTHODE PRIMALE-DUALE

Comme on l'a vu dans les chapitres précédents, la formulation mixte implémentée dans Forge3[®] implique qu'en chaque nœud il y a quatre inconnues : trois composantes de la vitesse et une de pression, ce qui conduit à la décomposition suivante du problème local sur le sous-domaine Ω_i :

$$\begin{bmatrix} K_i^{vv} & K_i^{vp} \\ K_i^{pv} & K_i^{pp} \end{bmatrix} \cdot \begin{bmatrix} v_i \\ p_i \end{bmatrix} = \begin{bmatrix} b_i^v \\ b_i^p \end{bmatrix} \quad (\text{III.20})$$

Par v_i on note les vitesses du sous-domaine Ω_i , p_i sont les pressions, par conséquent les indices v et p marquent les contributions dues aux vitesses et aux pressions, respectivement. Le gradient du problème de décomposition de domaine (III.9) ou (III.15) peut être dissocié de la même manière en composantes en vitesses et pressions :

$$g = \begin{bmatrix} g_v \\ g_p \end{bmatrix} \quad (\text{III.21})$$

Dans la méthode FETI appliquée au système (III.20), la partie du gradient correspondant aux vitesses est en fait une vitesse (puisque le gradient est le saut des solutions à travers l'interface, cf. (III.18)) et la partie correspondant aux pressions est une pression. Au contraire, dans la méthode primale de Schur le gradient est le flux des solutions à travers l'interface, donc la partie en vitesse correspond à une force (ou pression), et la partie g_p à une vitesse. Par conséquent, ces deux méthodes appliquées telles qu'elles conduisent à résoudre un problème itératif non-homogène.

Dans le cadre de l'approche primale-duale, on propose d'appliquer la méthode primale sur les pressions et la méthode duale sur les vitesses. Ainsi, le système itératif à résoudre sera-t-il homogène, puisque les deux termes du gradient deviennent homogènes à des vitesses. La méthode primale impose la condition de continuité entre les pressions à l'interface d'une manière directe, alors que la méthode duale introduit des multiplicateurs de Lagrange sur les vitesses de l'interface.

Cette réflexion, qui consiste à combiner deux méthodes, donne lieu à la décomposition suivante du système (III.20) :

$$\begin{bmatrix} K_{ii}^v & K_{ii}^{vp} & K_{if}^{vp} \\ K_{ii}^{pv} & K_{ii}^{pp} & K_{if}^p \\ K_{fi}^{pv} & K_{fi}^p & K_{ff}^p \end{bmatrix} \cdot \begin{bmatrix} v_i \\ p^{(i)} \\ p_f \end{bmatrix} = \begin{bmatrix} b_i^v + B_i^t \lambda \\ b^{p(i)} \\ b_f^{p(i)} \end{bmatrix} \quad (\text{III.22})$$

Ici, le vecteur v_i représente les vitesses partout dans le sous-domaine Ω_i , interface incluse, alors que les pressions sont décomposées en $p^{(i)}$ – pressions à l'intérieur du sous-domaine, et p_f – celles de l'interface. Il faut noter que l'opérateur de la méthode duale B_i (III.14) agit maintenant seulement sur les vitesses, et que le vecteur λ est de taille inférieure au nombre de degrés de liberté de l'interface, et vaut trois fois le nombre de nœuds d'interface.

Pour alléger les équations, introduisons un vecteur $x_i = [v_i \quad p^{(i)}]^t$ qui contient les vitesses partout sur le sous-domaine et les pressions intérieures, et de même le vecteur $b_i = [b_i^v \quad b^{p(i)}]^t$. Alors, en terme d'inconnues « globales » d'interface, λ et p_f , notre système local prend la forme:

$$\begin{bmatrix} K_{ii} & K_{if} \\ K_{fi} & K_{ff}^{(i)} \end{bmatrix} \cdot \begin{bmatrix} x_i \\ p_f \end{bmatrix} = \begin{bmatrix} b_i + B_i^t \lambda \\ b_f^{(i)} \end{bmatrix} \quad (\text{III.23})$$

La première ligne permet d'éliminer les inconnues intérieures :

$$x_i = K_{ii}^{-1} (b_i + B_i^t \lambda - K_{if} p_f) \quad (\text{III.24})$$

La deuxième ligne se transforme en:

$$K_{fi} K_{ii}^{-1} b_i + K_{fi} K_{ii}^{-1} B_i^t \lambda - K_{fi} K_{ii}^{-1} K_{if} p_f + K_{ff}^{(i)} p_f = b_f^{(i)} \quad (\text{III.25})$$

Avec la sommation sur tous les sous-domaines on a :

$$\left[K_{ff} - \sum_i K_{fi} K_{ii}^{-1} K_{if} \right] p_f + \sum_i K_{fi} K_{ii}^{-1} B_i^t \lambda = b_f - \sum_i K_{fi} K_{ii}^{-1} b_i \quad (\text{III.26})$$

La matrice $S = \left[K_{ff} - \sum_i K_{fi} K_{ii}^{-1} K_{if} \right]$ est la matrice du complément de Schur classique (III.7), et

à droite de cette relation (III.26) le vecteur $b_f - \sum_i K_{fi} K_{ii}^{-1} b_i$ représente le second membre classique de la méthode primale. Ainsi, grâce à la méthode primale, la condition de continuité des pressions à l'interface est imposée directement dans le système à résoudre, mais il manque encore l'équation de continuité des vitesses à travers les interfaces. Avec les opérateurs B_i agissant seulement sur les vitesses, cette condition est la suivante :

$$\sum_i B_i x_i = 0 \quad (\text{III.27})$$

À partir de (III.24), et avec la sommation sur tous les sous-domaines, on obtient :

$$\sum_i B_i K_{ii}^{-1} B_i^t \lambda - \sum_i B_i K_{ii}^{-1} K_{if} p_f = - \sum_i B_i K_{ii}^{-1} b_i \quad (\text{III.28})$$

Il est à noter que $D = \left[\sum_i B_i K_{ii}^{-1} B_i^t \right]$ est la matrice du complément de Schur duale (III.17), et que

le second membre correspond tout à fait à la formulation habituelle de la méthode FETI [Farhat et Roux 1991].

Le problème condensé à l'interface (III.26) et (III.28) que l'on doit résoudre se présente sous la forme :

$$\begin{bmatrix} \sum_i B_i K_{ii}^{-1} B_i^t & - \sum_i B_i K_{ii}^{-1} K_{if} \\ - \sum_i K_{fi} K_{ii}^{-1} B_i^t & - K_{ff} + \sum_i K_{fi} K_{ii}^{-1} K_{if} \end{bmatrix} \cdot \begin{bmatrix} \lambda \\ p_f \end{bmatrix} = \begin{bmatrix} - \sum_i B_i K_{ii}^{-1} b_i \\ - b_f + \sum_i K_{fi} K_{ii}^{-1} b_i \end{bmatrix} \quad (\text{III.29})$$

ou sous la forme matricielle :

$$\begin{bmatrix} D & A \\ A^t & S \end{bmatrix} \cdot \begin{bmatrix} \lambda \\ p_f \end{bmatrix} = \begin{bmatrix} c_v \\ c_p \end{bmatrix} \quad (\text{III.30})$$

Ainsi, les inconnues sur l'interface sont λ (attachées aux vitesses) et les pressions p_f . Le gradient du problème condensé (III.29) est composé de deux parties : la partie en vitesse, qui est calculée comme le saut des solutions locales, et la partie en pression. Ce calcul du gradient (III.21) s'effectue en deux étapes, de la manière suivante, pour un vecteur d'interface connu et qui se décompose lui aussi en parties vitesse et pression $w_f = [w_v \quad w_p]^t$

1) Résoudre le problème local suivant pour trouver y_i :

$$\begin{bmatrix} K_{ii} & K_{if} \\ K_{fi} & K_{ff}^{(i)} \end{bmatrix} \cdot \begin{bmatrix} y_i \\ w_p \end{bmatrix} = \begin{bmatrix} B_i^t w_v \\ 0 \end{bmatrix} \quad (\text{III.31})$$

Il est à noter que la partie du vecteur d'interface w_v , correspondant aux vitesses intervient dans le second membre, comme dans la méthode duale (III.19) et représente les conditions aux limites de type Neumann. La partie pression, w_p , représente les conditions aux limites de type Dirichlet (III.10). La forme de la solution de ce système est équivalente à (III.11) :

$$\begin{bmatrix} y_i \\ w_p \end{bmatrix} = \begin{bmatrix} K_{ii}^{-1} (B_i^t w_v - K_{if} w_p) \\ w_p \end{bmatrix} \quad (\text{III.32})$$

La partie recherchée y_i (composée en occurrence de tous les degrés de liberté correspondant aux vitesses et des pressions intérieures) vaut alors $y_i = K_{ii}^{-1}(B_i^t w_v - K_{if} w_p)$

2) Calculer le produit matrice vecteur avec le vecteur solution $[y_i \ w_p]^t$ ce qui donne :

$$\begin{bmatrix} K_{ii} & K_{if} \\ K_{fi} & K_{ff}^{(i)} \end{bmatrix} \cdot \begin{bmatrix} y_i \\ w_p \end{bmatrix} = \begin{bmatrix} 0 \\ s_i \end{bmatrix} \quad (\text{III.33})$$

où le sous-vecteur s_i , correspondant aux degrés de liberté d'interface en pression, est égal à :

$$s_i = K_{ff}^{(i)} w_p - K_{fi} y_i \quad (\text{III.34})$$

Le gradient en vitesse g_v est le saut de y_i à travers les interfaces, comme dans la méthode duale. L'application d'opérateur B_i de la méthode primale-duale permet d'extraire seulement les vitesses du vecteur-solution y_i . Le gradient en pression g_p est le résultat de l'assemblage des contributions de s_i comme cela a été décrit pour la méthode primale. Ainsi, le gradient du problème d'interface à résoudre prend la forme :

$$g = \begin{bmatrix} g_v \\ g_p \end{bmatrix} = \begin{bmatrix} \sum_i B_i y_i \\ \sum_i s_i \end{bmatrix} \quad (\text{III.35})$$

Il est important de souligner que le bloc K_{ii} de la matrice locale est factorisé. La procédure est semblable à celle de la factorisation de la matrice dans la méthode primale, mais cette fois-ci le problème de Dirichlet est posé de manière à bloquer non tous les degrés de liberté d'interface, mais seulement les pressions.

III.5. PRECONDITIONNEMENT

Comme toute méthode itérative, l'efficacité des méthodes de décomposition de domaine est fortement liée à la qualité de leur preconditionnement.

III.5.1. Méthode du complément de Schur

Puisque la matrice du complément de Schur n'est pas assemblée explicitement, il est impossible de la preconditionner algébriquement. Cependant, comme elle est la somme de contributions locales, il est possible d'approcher l'inverse de la somme par la somme des inverses.

Supposons que le sous-problème défini sur chaque sous-domaine ait assez de conditions aux limites de type Dirichlet pour que la matrice K_{ii} ne soit pas singulière.

En se basant sur la définition du complément de Schur (III.7) $S^{(i)} = K_{ff}^{(i)} - K_{fi} K_{ii}^{-1} K_{if}$, on réécrit l'équation (III.12) pour un vecteur d'interface v_f donné :

$$\begin{bmatrix} K_{ii} & K_{if} \\ K_{fi} & K_{ff}^{(i)} \end{bmatrix} \cdot \begin{bmatrix} -K_{ii}^{-1} K_{if} v_f \\ v_f \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ S^{(i)} v_f \end{bmatrix} \quad (\text{III.36})$$

Cela signifie que le vecteur v_f est la restriction à l'interface de la solution du problème de Neumann ayant pour condition aux limites le vecteur $S^{(i)} v_f$ sur l'interface et zéro ailleurs. Considérons un vecteur de l'interface g_f , et la solution w du problème de Neumann suivant :

$$\begin{bmatrix} K_{ii} & K_{if} \\ K_{fi} & K_{ff}^{(i)} \end{bmatrix} \cdot \begin{bmatrix} w^{(i)} \\ w_f \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ g_f \end{bmatrix} \quad (\text{III.37})$$

Alors, compte tenu de (III.36), la restriction de w sur l'interface est égale à :

$$w_f = (S^{(i)})^{-1} g_f \quad (\text{III.38})$$

c'est à dire, le produit de g_f par l'inverse du complément de Schur local. Ainsi, peut-on approcher S^{-1} par la somme des contributions locales $(S^{(i)})^{-1}$ en utilisant les solutions des problèmes locaux de type Neumann (III.37). En effet, contrairement au système (III.4) de la méthode primale où l'on résout le problème de type Dirichlet avec les inconnues d'interface bloquées (données) afin de trouver un vecteur $S^{(i)} v_f$, pour trouver un vecteur $(S^{(i)})^{-1} g_f$ il faut procéder différemment. Il faut résoudre le problème inverse, qui est de type Neumann. Ainsi, on procède à la résolution du système (III.37) avec le second membre $\begin{bmatrix} 0 & g_f \end{bmatrix}$ connu pour trouver la solution $\begin{bmatrix} w^{(i)} & w_f \end{bmatrix}$. La restriction de cette solution sur l'interface w_f est le produit recherché du gradient par l'inverse de la matrice de Schur locale.

La contribution d'un sous-domaine au gradient du problème d'interface est le flux de la solution du problème de Dirichlet local. Le flux complet est la somme de toutes les contributions. Donc, pour préconditionner le problème global condensé aux interfaces il faut prendre la somme des contributions locales. Pour des raisons de symétrie, on introduit aussi un facteur de pondération. Ainsi, le préconditionneur de Neumann de la méthode de complément de Schur (III.9) s'écrit plus précisément :

$$Q_N^{-1} = \sum_i \text{weight}_i (S^{(i)})^{-1} \text{weight}_i^t \quad (\text{III.39})$$

avec weight_i des matrices de pondération telles que leur somme donne la matrice d'identité :

$$\sum_i \text{weight}_i = I \quad (\text{III.40})$$

La résolution de problème de Neumann (III.37) nécessite la factorisation et le stockage de la matrice locale K_i . En outre, l'utilisation de ce préconditionneur pour la résolution du problème condensé aux interfaces implique une augmentation du coût des calculs à chaque itération comparable à celui de résolution du problème de Dirichlet. En effet, non seulement on résout un problème de Dirichlet, mais aussi un problème de Neumann pour préconditionner. Pour éviter de doubler ainsi le coût et l'espace mémoire, un préconditionneur plus simple, par la diagonale, a été implémenté :

$$Q_d^{-1} = \sum_i \text{weight}_i \left(\text{diag}(K_{ff}^{(i)}) \right)^{-1} \text{weight}_i^t \quad (\text{III.41})$$

III.5.2. Méthode FETI

Comme précédemment pour la méthode du complément de Schur, on cherche à approcher l'inverse de la matrice $D = \sum_i B_i K_i^{-1} B_i^t$ de la méthode duale par la somme des inverses locaux. Pour tout vecteur d'interface g_f , on cherche donc le vecteur w_f tel que :

$$(B_i K_i^{-1} B_i^t) w_f = g_f \quad (\text{III.42})$$

Cette équation écrite dans les notations propres à la méthode FETI correspond tout à fait au système (III.37) restreint sur l'interface. À partir de (III.38), on peut alors conclure que :

$$S^{(i)} = (B_i K_i^{-1} B_i^t)^{-1} \quad (\text{III.43})$$

Ceci revient à dire que pour préconditionner le problème d'interface de la méthode FETI il faudrait utiliser le produit par la matrice de complément de Schur S . Or, la façon de ce faire est de résoudre un problème de type Dirichlet. Sur chaque sous-domaine, pour un gradient d'interface g_f donné, on procède comme cela a été montré dans (III.10)-(III.12) et on trouve le produit $S^{(i)} g_f$. La somme des contributions de tous les sous-domaines donne le gradient global préconditionné.

Le préconditionneur de Dirichlet pour le complément de Schur dual D (III.17) peut être écrit comme la somme pondérée des matrices locales de complément de Schur [Farhat et al 1994] :

$$Q_D^{-1} = \sum_i \text{weight}_i (S^{(i)}) \text{weight}_i^t \quad (\text{III.44})$$

Avec ce préconditionneur, le conditionnement de la méthode FETI est borné par $(1 + \log^2(H/h))$ [Farhat et al 1994], où H est la taille des sous-domaines et h est la taille de maille. On dit que ce préconditionneur est optimal. Il est tout de même très coûteux, car non seulement la résolution du système linéaire se fait deux fois par itération, mais on est également obligé de factoriser et de stocker les matrices correspondant aux problèmes locaux de Dirichlet. Il a donc été proposé [Farhat et al 1994] d'utiliser un préconditionneur localisé, qui n'est pas optimal du point de vue mathématique, mais qui

est très économique. Ce préconditionneur est basé sur la définition (III.7) de la matrice du complément de Schur :

$$S^{(i)} = K_{ff}^{(i)} - K_{fi} K_{ii}^{-1} K_{if} \quad (\text{III.45})$$

Ainsi, au lieu de préconditionner par le complément de Schur complet comme dans (III.10)-(III.12), on approche la matrice locale $S^{(i)}$ par $K_{ff}^{(i)}$, et le préconditionneur s'écrit sous la forme :

$$Q_i^{-1} = \sum_i \text{weight}_i (K_{ff}^{(i)}) \text{weight}_i^t = \sum_i \text{weight}_i (B_i K_i B_i^t) \text{weight}_i^t \quad (\text{III.46})$$

Le conditionnement du problème préconditionné par Q_i^{-1} est proportionnel à H/h [Farhat et al 1994].

III.5.3. Méthode primale-duale

En premier lieu, un préconditionneur simple pour la méthode primale-duale a été envisagé. Pour la partie en vitesse, il correspond au préconditionneur localisé (III.46) de la méthode FETI, et pour la pression au préconditionneur par la diagonale (III.41). Avec les opérateurs B_i , on extrait les composantes des blocs K_{ii} correspondant aux vitesses à l'interface. La partie pression du vecteur préconditionné est divisée par la diagonale du bloc $K_{ff}^{(i)}$ correspondant à la pression à l'interface. En gardant la structure des matrices par blocs utilisée plus haut, on peut écrire ce préconditionneur sous la forme :

$$Q_{id}^{-1} = \begin{bmatrix} Q_i^{-1} & 0 \\ 0 & Q_d^{-1} \end{bmatrix} \text{ ou } Q_{id}^{-1} = \sum_i \text{weight}_i \begin{bmatrix} B_i K_{ii} B_i^t & 0 \\ 0 & (\text{diag}(K_{ff}^{(i)}))^{-1} \end{bmatrix} \text{weight}_i^t \quad (\text{III.47})$$

Une autre manière de préconditionner serait de combiner le préconditionneur de Dirichlet en vitesse et le préconditionneur de Neumann en pression. On rappelle que le gradient de la méthode primale-duale est composé de deux parties. Pour le trouver, on résout les problèmes locaux (III.31), dont les solutions (III.32) fournissent le gradient en vitesse, cette partie du gradient suivant la logique de la méthode FETI. En ce qui concerne le gradient en pression, suivant la méthode primale, on l'obtient par le calcul des produits matrice vecteur (III.33).

Pour préconditionner le gradient $[g_v \ g_p]$ connu à l'itération courante, on doit alors résoudre un problème où la partie vitesse g_v impose les conditions de type Dirichlet, tandis que la partie pression g_p correspond aux conditions de type Neumann. On introduit des opérateurs C_i qui restreignent un vecteur aux pressions, comme les opérateurs B_i restreignent un vecteur aux degrés de liberté en vitesse. En outre, la matrice locale K_i est restructurée comme dans (III.23) :

$$K_i = \begin{bmatrix} \tilde{K}_{ii} & \tilde{K}_{if} \\ \tilde{K}_{fi} & \tilde{K}_{ff}^{(i)} \end{bmatrix} \quad (\text{III.48})$$

Il faut souligner que le bloc $\tilde{K}_{ff}^{(i)}$ de la matrice locale K_i correspond aux degrés de liberté bloqués aux interfaces, mais cette fois-ci ce sont les vitesses d'interface. Le bloc intérieur \tilde{K}_{ii} comprend alors les contributions relatives à toutes les pressions (intérieures au sous-domaine et celles d'interface) et aux vitesses intérieures.

Le système que l'on doit résoudre pour préconditionner ressemble au système (III.31) et s'écrit comme :

$$\begin{bmatrix} \tilde{K}_{ii} & \tilde{K}_{if} \\ \tilde{K}_{fi} & \tilde{K}_{ff}^{(i)} \end{bmatrix} \cdot \begin{bmatrix} y_i \\ g_v \end{bmatrix} = \begin{bmatrix} C_i^t g_p \\ 0 \end{bmatrix} \quad (\text{III.49})$$

La solution de ce problème est équivalente à la relation (III.32) et s'écrit :

$$y_i = \tilde{K}_{ii}^{-1} [C_i^t g_p - \tilde{K}_{if} g_v] \quad (\text{III.50})$$

Le produit matrice vecteur (III.33) donne

$$\begin{bmatrix} \tilde{K}_{ii} & \tilde{K}_{if} \\ \tilde{K}_{fi} & \tilde{K}_{ff}^{(i)} \end{bmatrix} \cdot \begin{bmatrix} y_i \\ g_p \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{K}_{ff}^{(i)} g_p - \tilde{K}_{fi} y_i \end{bmatrix} = \begin{bmatrix} 0 \\ q_i \end{bmatrix} \quad (\text{III.51})$$

Ainsi, le gradient préconditionné $Q_{DN}^{-1} g$ est organisé comme dans (III.35) de deux parties : le saut des solutions locales y_i à travers l'interface pour le gradient en vitesse, et l'assemblage des vecteurs additionnels q_i pour la partie pression :

$$Q_{DN}^{-1} g = \begin{bmatrix} Q_{DN}^{-1} g_v \\ Q_{DN}^{-1} g_p \end{bmatrix} = \begin{bmatrix} \sum B_i y_i \\ \sum q_i \end{bmatrix} \quad (\text{III.52})$$

III.6. PROBLÈME DES MODES RIGIDES

En pratique, dans l'approche duale, le procédé de partitionnement produit très souvent des sous-structures « flottantes » (Figure III.5), c'est-à-dire des sous-domaines qui n'ont pas assez des déplacements prescrits pour éliminer les mouvements de corps rigide du sous-domaine. Dans le cas de sous-domaine flottant, la matrice locale K_i est singulière.

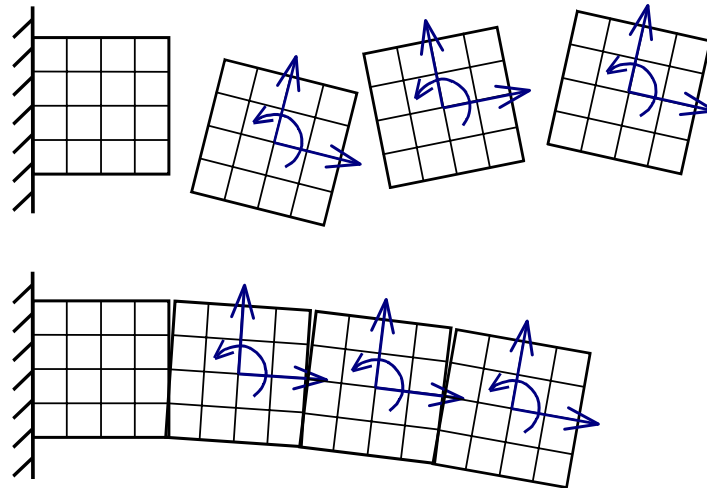


Figure III.5. Sous-domaines flottants : modes rigides

III.6.1. Extraction des modes rigides

Considérons une matrice K , non-définie positive, et qui peut être décomposée de la manière suivante :

$$K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \quad (\text{III.53})$$

où le bloc K_{11} est défini positif et sa dimension est égale au rang de la matrice entière K . Si on applique la factorisation de Gauss-Jordan à la matrice K on a :

$$\begin{bmatrix} I & 0 \\ -K_{21}K_{11}^{-1} & I \end{bmatrix} \cdot \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \cdot \begin{bmatrix} I & -K_{11}^{-1}K_{12} \\ 0 & I \end{bmatrix} = \begin{bmatrix} K_{11} & 0 \\ 0 & K_{22} - K_{21}K_{11}^{-1}K_{12} \end{bmatrix} \quad (\text{III.54})$$

Comme le rang de la matrice résultante est égal à la dimension de K_{11} , le complément de Schur $K_{22} - K_{21}K_{11}^{-1}K_{12}$ doit être égal à zéro. De plus, suite à l'expression suivante qui est vraie :

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \cdot \begin{bmatrix} -K_{11}^{-1}K_{12} \\ I \end{bmatrix} = \begin{bmatrix} 0 \\ K_{22} - K_{21}K_{11}^{-1}K_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (\text{III.55})$$

on peut déduire que les colonnes de la matrice N qui sont définies comme :

$$N = \begin{bmatrix} -K_{11}^{-1}K_{12} \\ I \end{bmatrix} \quad (\text{III.56})$$

forment une base du noyau de la matrice K .

Maintenant, si on considère le système d'équations non-défini :

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (\text{III.57})$$

la même procédure donne :

$$\begin{cases} x_1 = K_{11}^{-1}b_1 - K_{11}^{-1}K_{12}x_2 \\ (K_{22} - K_{21}K_{11}^{-1}K_{12})x_2 = b_2 - K_{21}K_{11}^{-1}b_1 = 0 \end{cases} \quad (\text{III.58})$$

Ainsi, le problème admet au moins une solution si seulement le second membre b vérifie la condition suivante :

$$\begin{bmatrix} -K_{11}^{-1}K_{12} \\ I \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = b_2 - K_{21}K_{11}^{-1}b_1 = 0 \quad (\text{III.59})$$

Dans ce cas, il existe un nombre infini de solutions de la forme :

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} K_{11}^{-1} & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} -K_{11}^{-1}K_{12} \\ I \end{bmatrix} \cdot x_2 \quad (\text{III.60})$$

Cette équation (III.60) montre que les solutions du problème (III.57) sont définies comme une somme de la solution particulière $[K_{11}^{-1}b_1 \quad 0]^t$ et d'un élément quelconque du noyau. La solution particulière $x = K^+ [b_1 \quad b_2]^t$ est obtenue avec la matrice inverse généralisée K^+ :

$$K^+ = \begin{bmatrix} K_{11}^{-1} & 0 \\ 0 & 0 \end{bmatrix} \quad (\text{III.61})$$

La condition d'admissibilité (III.59) exprime le fait que le second membre b doit être orthogonal au noyau de la matrice K , ce qui équivaut à dire qu'il appartient à l'espace image de K .

En pratique, il est impossible de numérotter la matrice K comme dans (III.53) pour calculer directement la matrice N suivant la définition (III.56). Alors, lors la factorisation de la matrice K , on repère les pivots nuls. Les colonnes de la matrice K correspondant à ces pivots nuls forment une matrice rectangulaire $R = [K_{12} \quad K_{22}]^t$. Si la taille du problème (III.57) est de n , et le nombre de pivots nuls de m , la dimension de cette matrice est de $n \times m$. À l'aide de la matrice K^+ , on résout m fois le système (III.57) ayant pour second membre une colonne R_j de la matrice R , $j = 1, \dots, m$. Les m vecteurs solutions forment une matrice rectangulaire $n \times m$ $N = [-K_{11}^{-1}K_{12} \quad I]^t$ qui est une base de noyau recherché. La matrice carrée I est la matrice d'identité de la taille $m \times m$.

III.6.2. Problème hybride aux interfaces

Revenant à la méthode FETI, on réécrit pour chaque sous-domaine Ω_i le problème de Neumann à résoudre et la condition de continuité :

* pour la clarté de la démonstration, on utilise ici la numérotation (III.53).

$$\begin{aligned} K_i v_i &= b_i + B_i' \lambda \\ \sum_i B_i v_i &= 0 \end{aligned} \quad (\text{III.62})$$

En présence des modes rigides, la solution du problème (III.62) prend la forme de la somme d'une solution particulière v_i^+ , obtenue avec l'inverse généralisée K_i^+ , et des déplacements de corps rigide :

$$v_i = v_i^+ + N_i \alpha_i = K_i^+ (b_i + B_i' \lambda) + N_i \alpha_i \quad (\text{III.63})$$

où N_i est une base du noyau de K_i (III.56). Le vecteur α_i , dont la dimension correspond à m_i , le nombre des modes rigides du sous-domaine Ω_i , contient les coordonnées de v_i dans cette base. La condition d'orthogonalité :

$$N_i' K_i v_i = N_i' (b_i + B_i' \lambda) = 0 \quad (\text{III.64})$$

exprime le fait que les modes de corps rigide ne produisent pas d'énergie interne. Elle fournit aussi la dernière équation pour que le système suivant soit fermé :

$$\begin{bmatrix} D & E \\ E' & 0 \end{bmatrix} \cdot \begin{bmatrix} \lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} c_\lambda \\ c_\alpha \end{bmatrix} \quad (\text{III.65})$$

Ce système, écrit pour m sous-domaines flottants parmi N_{proc} , est une généralisation du problème condensé aux interfaces de la méthode FETI (III.16). On a noté :

- $D = \sum_i B_i K_i^+ B_i'$ est la matrice généralisée du complément de Schur duale,
 - $E = [B_1 N_1 \quad \dots \quad B_m N_m]$, et $E \alpha = \sum_{j=1}^m B_j N_j \alpha_j$ représente le saut à travers l'interface des modes rigides,
 - $(E' \lambda)_j = N_j' B_j' \lambda$, pour chaque sous-domaine flottant,
 - $c_\lambda = -\sum_i B_i K_i^+ b_i$ et $c_\alpha = [N_1' b_1 \quad \dots \quad N_m' b_m]$ composent le second membre du problème
- (III.65)

III.6.3. Méthode itérative projetée

Puisque les matrices K_i^+ sont symétriques, la matrice D l'est aussi. Donc, le problème hybride aux interfaces (III.65) peut être interprété comme un problème de minimisation sous contrainte :

$$\min J(\lambda) = \frac{1}{2} \lambda' D \lambda - \lambda' c_\lambda \quad \text{sous la condition } E' \lambda = c_\alpha \quad (\text{III.66})$$

Ce problème peut être résolu avec une méthode de gradient conjugué (par exemple) si à chaque itération k la contrainte $E'\lambda = c_\alpha$ est satisfaite, ce qui peut être obtenu à l'aide de l'algorithme du gradient conjugué projeté où, à chaque itération k , un projecteur P élimine de la solution λ^k les composantes qui ne satisfont pas la contrainte $E'\lambda = c_\alpha$.

Supposons que la valeur initiale λ^0 soit choisie telle qu'elle satisfasse l'équation de contrainte : $E'\lambda^0 = c_\alpha$. Pour chaque itération suivante, λ^k doit satisfaire la condition $E'\lambda^k = 0$. λ^k est alors décomposé en :

$$\lambda^k = \lambda_1^k + \lambda_2^k, \text{ avec } E'\lambda_1^k = 0 \text{ et } E'\lambda_2^k \neq 0 \quad (\text{III.67})$$

Il est donc nécessaire de construire un projecteur adéquat P qui est une matrice qui projette λ^k sur le noyau de E' :

$$\lambda^k \rightarrow P\lambda^k = \lambda_1^k = \lambda^k - \lambda_2^k. \quad (\text{III.68})$$

Forcément, cette matrice n'est pas unique, mais on va choisir un projecteur simple, tout en sachant que dans la littérature il en existe bien d'autres. Pour une matrice F arbitraire, symétrique, définie positive, un opérateur $P(F)$ de projection orthogonale sur le noyau de E' peut être écrit algébriquement comme :

$$P(F) = F - FE(E'FE)^{-1}E' \quad (\text{III.69})$$

Pour des raisons d'efficacité [Farhat et al 2000a], et en premier lieu, la matrice F est prise égale à la matrice d'identité I et l'opérateur de projection $P(F)$ s'écrit :

$$P = I - E(E'E)^{-1}E' \quad (\text{III.70})$$

La matrice $E'E$ est symétrique, définie positive et creuse. Sa structure est déterminée par l'interconnectivité entre les sous-domaines, donc au cours des itérations elle ne varie pas.

Pour une approximation λ^k donnée, on trouve sur chaque sous-domaine des solutions particulières v_i^{k+} des problèmes de Neumann à l'aide des matrices locales généralisées K_i^+ :

$$v_i^{k+} = K_i^+(b_i + B_i'\lambda^k) \quad (\text{III.71})$$

Le gradient g^k du problème d'interface étant égal au saut de ces solutions particulières v^{k+} , sa projection Pg^k prend la forme, suite à (III.69) :

$$Pg^k = g^k + E\alpha^k = \sum_i B_i v_i^{k+} + \sum_i B_i N_i \alpha_i^k \quad (\text{III.72})$$

Ainsi, le gradient projeté Pg^k est égal au saut des solutions particulières v_i^{k+} complétées par les mouvements de corps rigide (III.63) de chaque sous-domaine.

Le calcul du gradient projeté consiste alors en celui des coefficients α^k des mouvements de corps rigide optimaux. À chaque itération et pour un gradient g^k donné, on résout le problème suivant pour trouver les coefficients nécessaires au calcul du gradient projeté :

$$(E^t E)\alpha^k = E^t g^k \quad (\text{III.73})$$

La résolution à chaque itération de ce problème de type « grille grossière » peut être interprétée comme un lissage des solutions v_i^k approchantes. Il induit que la convergence de la méthode soit asymptotiquement indépendante du nombre de sous-domaines [Farhat et al 1994]. n étant la taille du problème et \bar{m} le nombre total des modes rigides, la taille de la matrice E est de $n \times \bar{m}$, alors que la taille de la matrice $E^t E$ est de $\bar{m} \times \bar{m}$, ce qui est relativement petit par rapport aux autres matrices utilisées dans la méthode. Dans la pratique, la matrice $E^t E$ est habituellement rassemblée et inversée une seule fois, ainsi la résolution de ce problème (III.73) se fait facilement et ne demande pas beaucoup de temps ni de ressources.

III.6.4. Méthode primale-duale

Dans le cas mixte, l'analogue de l'équation (III.63) de la méthode FETI est la solution particulière x_i^+ du problème en présence des modes rigides :

$$x_i^+ = K_{ii}^+ (b_i + B_i^t \lambda - K_{if} p_f) \quad (\text{III.74})$$

Cette solution doit être complétée par les mouvements de corps rigide (qui existent en vitesse seulement) :

$$x_i = x_i^+ + N_i \alpha_i = K_{ii}^+ (b_i + B_i^t \lambda - K_{if} p_f) + N_i \alpha_i \quad (\text{III.75})$$

Cette démarche entraîne la transformation de l'équation (III.25) en :

$$K_{ff} K_{ii}^+ b_i + K_{ff} K_{ii}^+ B_i^t \lambda - K_{ff} K_{ii}^+ K_{if} p_f^{(i)} + K_{ff} N_i \alpha_i + K_{ff}^{(i)} p_f^{(i)} = b_f^{(i)} \quad (\text{III.76})$$

Ou encore, avec la sommation sur tous les sous-domaines, on obtient :

$$\sum_i B_i K_{ii}^+ B_i^t \lambda - \sum_i B_i K_{ii}^+ K_{if} p_f + \sum_i B_i N_i \alpha_i = - \sum_i B_i K_{ii}^+ b_i \quad (\text{III.77})$$

$$\left[K_{ff} - \sum_i K_{ff} K_{ii}^+ K_{if} \right] p_f + \sum_i K_{ff} K_{ii}^+ B_i^t \lambda + \sum_i K_{ff} N_i \alpha_i = b_f - \sum_i K_{ff} K_{ii}^+ b_i \quad (\text{III.78})$$

Pour chaque sous-domaine flottant, la condition d'orthogonalité écrite dans la méthode FETI par (III.64) devient dans la méthode primale-duale l'équation suivante :

$$N_i^t (b_i + B_i^t \lambda - K_{if} p_f) = 0 \quad (\text{III.79})$$

On aboutit alors au système suivant, que l'on doit résoudre :

$$\begin{bmatrix} D & A & E \\ A & S & G \\ E^t & G^t & 0 \end{bmatrix} \cdot \begin{bmatrix} \lambda \\ p_f \\ \alpha \end{bmatrix} = \begin{bmatrix} c_v \\ c_f \\ c_\alpha \end{bmatrix} \quad (\text{III.80})$$

m étant le nombre de sous-domaines flottants, on complète les notations de (III.29)-(III.30) avec les notations analogues au (III.65) et celles propres à la méthode primale-duale :

- $E = [B_1 N_1 \quad \dots \quad B_m N_m]$, et $E\alpha = \sum_{j=1}^m B_j N_j \alpha_j$ représente le saut à travers l'interface des modes rigides. Contrairement à la formulation (III.65), et puisque les opérateurs B_i sont définis seulement pour les vitesses, il s'agit ici de calculer le saut des vitesses à travers l'interface.
- $(E^t \lambda)_j = N_j^t B_j^t \lambda$, pour chaque sous-domaine flottant, $j = 1, \dots, m$,
- $G = [K_{f1} N_1 \quad \dots \quad K_{fm} N_m]$ et $G\alpha = \sum_{j=1}^m K_{jf} N_j \alpha_j$
- $(G^t p_f)_j = N_j^t K_{jf}^t p_f$, pour chaque sous-domaine flottant,
- $c_\alpha = [N_1^t b_1 \quad \dots \quad N_m^t b_m]$ complète le second membre du problème. Il faut noter qu'ici chaque vecteur b_j représente le second membre du problème duquel les pressions sur l'interface sont exclues (voir la formulation du problème (III.22)).

Si l'on introduit une notation pour laquelle les variables d'interface λ et p_f forment un seul vecteur y , on retrouve la formulation du problème hybride de la méthode FETI (III.65) :

$$\begin{bmatrix} M & H \\ H^t & 0 \end{bmatrix} \cdot \begin{bmatrix} y \\ \alpha \end{bmatrix} = \begin{bmatrix} c_y \\ c_\alpha \end{bmatrix} \quad (\text{III.81})$$

En ce qui concerne les modes rigides, la différence entre la méthode FETI et la nouvelle méthode repose sur le fait que maintenant les opérateurs B_i agissent seulement sur les vitesses d'interface et non sur tous les degrés de liberté correspondant aux interfaces. Le problème condensé aux interfaces (III.81) de la méthode primale-duale peut être résolu de la même manière par une méthode de gradient conjugué projeté avec calcul des modes rigides à chaque itération.

III.7. SOLUTION DU PROBLÈME CONDENSÉ AUX INTERFACES

Comme on l'a vu dans le chapitre précédent, les systèmes linéaires issus de Forge3[®] sont symétriques mais non définis positifs. Ainsi, la méthode de gradient conjugué habituellement utilisée

pour la résolution itérative du problème condensé aux interfaces ne peut être retenue. On présente ici une méthode OrthoDir préconditionnée et OrthoDir projetée préconditionnée, utilisées pour cette résolution.

III.7.1. Méthode OrthoDir préconditionnée

La méthode de résolution du problème global suivant l'approche de la méthode du complément de Schur (III.9) est la suivante :

Initialisation:

- Initialisation : les valeurs à l'interface sont mises à zéro :

$$p_f^0 = 0 \quad (\text{III.82})$$

- Initialisation : on résout le problème de Dirichlet local sur chaque sous-domaine et trouve ainsi la solution p_i^0 à l'intérieur du sous-domaine et le vecteur s_i^0 des efforts sur l'interface

$$\begin{bmatrix} K_{ii} & K_{if} \\ K_{fi} & K_{ff}^{(i)} \end{bmatrix} \cdot \begin{bmatrix} p_i^0 \\ 0 \end{bmatrix} = \begin{bmatrix} b_i \\ b_f^{(i)} \end{bmatrix} \quad (\text{III.83})$$

$$p_i^0 = K_{ii}^{-1} b_i \quad \text{et} \quad s_i^0 = b_f^{(i)} - K_{fi} p_i^0 \quad (\text{III.84})$$

- Initialisation : calculer le résidu initial par assemblage des contributions s_i^0 de tous les sous-domaines :

$$g^0 = \sum_i s_i^0 \quad (\text{III.85})$$

- Initialisation : définition de la direction de descente initiale w^0 :

$$w^0 = g^0 \quad (\text{III.86})$$

- Initialisation : vecteur

$$z^0 = Q^{-1} \cdot w^0 \quad (\text{III.87})$$

Considérons maintenant qu'à l'itération k les valeurs p_f^k , p_i^k , g^k et w^k sont connues. L'itération $k+1$ consiste à effectuer des étapes suivantes :

- Calculer le champ local w^k intérieur au sous-domaine Ω_i et qui est la solution du problème de Dirichlet :

$$K_{ii} w_i^k = -K_{if} w^k \quad (\text{III.88})$$

- Calculer le produit de w^k par la matrice du complément de Schur, comme cela a été montré dans la relation (III.12) :

$$Sw^k = \sum_i K_{f_i} w_i^k \quad (\text{III.89})$$

- Calculer le coefficient de descente ρ^k et mettre à jour les champs :

$$\rho^k = -\frac{(z^k, Sw^k)}{(Sw^k, Sw^k)} \quad (\text{III.90})$$

$$p_f^{k+1} = p_f^k + \rho^k \cdot w^k \quad (\text{III.91})$$

$$g^{k+1} = g^k + \rho^k \cdot Sw^k \quad (\text{III.92})$$

- Calculer le produit de g^{k+1} par la matrice du complément de Schur comme présenté dans (III.10)-(III.12) :

$$Sg^{k+1} \quad (\text{III.93})$$

- Préconditionner :

$$z^{k+1} = Q^{-1} \cdot Sg^{k+1} \quad (\text{III.94})$$

- Calculer le coefficient de conjugaison γ^{k+1} et mettre à jour les champs :

$$\gamma^{k+1} = -\frac{(z^{k+1}, Sg^k)}{(Sg^k, Sg^k)} \quad (\text{III.95})$$

$$w^{k+1} = g^{k+1} + \gamma^{k+1} \cdot w^k \quad (\text{III.96})$$

$$z^{k+1} = Sg^{k+1} + \gamma^{k+1} \cdot z^k \quad (\text{III.97})$$

La plupart de ces opérations sont purement locales et donc aisément parallélisables. L'assemblage des contributions s_i^k demande néanmoins un échange des données, ainsi que le calcul du coefficient de descente ρ^k et du coefficient de conjugaison γ^k . Chaque processeur calcule sa contribution au produit scalaire, puis les contributions sont rassemblées et diffusées à tous. Cette opération implique une communication globale entre les processeurs et ne dépend pas de la topologie de partition.

III.7.2. Méthode OrthoDir projetée

La méthode de résolution du problème global suivant l'approche de la méthode FETI (III.16) est la suivante :

Initialisation:

- Initialisation : calculer les valeurs initiales de λ^0 . Sur chaque sous-domaine, on cherche à calculer le second membre initial c_i .

$$c_i = -N_i^t b_i \quad (\text{III.98})$$

Ces vecteurs sont ensuite assemblés pour former un vecteur global c . Les produits matrices vecteur suivants donnent la valeur initiale de λ^0 :

$$\lambda^0 = E(E^t E)^{-1} c \quad (\text{III.99})$$

- Initialisation : on résout le problème de Neumann local sur chaque sous-domaine et trouve ainsi la solution v_i^0 à l'intérieur du sous-domaine. En présence des modes rigides, c'est la solution particulière v_i^{0+} que l'on trouve ; elle doit être complétée par la suite par les mouvements de corps rigide.

$$K_i v_i = b_i + B_i^t \lambda^0 \quad (\text{III.100})$$

$$v_i^{0+} = K_{ii}^+ (b_i + B_i^t \lambda^0) \quad (\text{III.101})$$

- Initialisation : calculer le résidu initial g^0 comme le saut des solutions particulières v_i^{0+} à travers les interfaces :

$$g^0 = \sum_i B_i v_i^{0+} \quad (\text{III.102})$$

- Initialisation : calculer le gradient projeté Pg^0 . D'abord, on résout le problème (III.73) pour trouver les coefficients α^0 des mouvements de corps rigide. La matrice $E^t E$ étant déjà inversée, il s'agit de faire des produits matrice vecteurs, ce qui donne :

$$\alpha^0 = -(E^t E)^{-1} E^t g^0 \quad (\text{III.103})$$

Enfin, le gradient projeté Pg^0 est donné par (III.72) :

$$Pg^0 = \left[I - E(E^t E)^{-1} E^t \right] g^0 = g^0 + E\alpha^0 \quad (\text{III.104})$$

- Initialisation : solution initiale particulière v_i^{0+} est complétée par les modes rigides :

$$v_i^0 = v_i^{0+} + N_i \alpha_i^0 \quad (\text{III.105})$$

- Initialisation : vecteur le gradient :

$$z^0 = Q^{-1} \cdot Pg^0 \quad (\text{III.106})$$

- Initialisation : afin de garder la symétrie après le préconditionnement, le vecteur z^0 doit être projeté selon (III.103)-(III.104). En absence de préconditionnement, une seule projection appliquée au gradient suffit.

$$\beta^0 = -(E^t E)^{-1} E^t z^0 \quad (\text{III.107})$$

$$Pz^0 = \left[I - E(E^t E)^{-1} E^t \right] z^0 = z^0 + E\beta^0 \quad (\text{III.108})$$

- Initialisation : définition de la direction de descente initiale w^0 qui est égale au gradient projeté préconditionné :

$$w^0 = Pz^0 \quad (\text{III.109})$$

- Initialisation : pour un champs w^0 donné, on résout le système suivant pour trouver les vecteurs x_i^{0+} :

$$K_i x_i^0 = B_i^t w^0 \quad (\text{III.110})$$

$$x_i^{0+} = K_i^+ B_i^t w^0 \quad (\text{III.111})$$

- Initialisation : calculer le produit de w^0 par la matrice du complément de Schur dual D comme le saut des vecteurs x_i^{0+} à travers les interfaces :

$$Dw^0 = \sum_i B_i x_i^{0+} \quad (\text{III.112})$$

- Initialisation : suivre les étapes (III.103)-(III.104) pour trouver la projection PDw^0 , vecteur (III.106) par la matrice Q et projeter encore une fois (III.107)-(III.108). On obtient à la fin de cette démarche un vecteur $P(Q \cdot P(Dw^0))$

Considérons maintenant qu'à l'itération k les valeurs p_f^k , p_i^k , g^k et w^k sont connues. L'itération $k+1$ consiste à effectuer des étapes suivantes :

- Calculer le coefficient de descente ρ^k et mettre à jour les champs :

$$\rho^k = - \frac{(Pz^k, Dw^k)}{(Dw^k, Dw^k)} \quad (\text{III.113})$$

$$\lambda^{k+1} = \lambda^k + \rho^k \cdot w^k \quad (\text{III.114})$$

$$g^{k+1} = g^k + \rho^k \cdot Dw^k \quad (\text{III.115})$$

- La nouvelle direction de descente est

$$w^{k+1} = g^{k+1} \quad (\text{III.116})$$

- Pour la nouvelle direction de descente w^{k+1} connue, résoudre le problème (III.110)

$$K_i x_i^{k+1} = B_i^t w^{k+1} \quad (\text{III.117})$$

Et trouver sa solution particulière :

$$x_i^{k+1+} = K_i^+ B_i^t w^{k+1} \quad (\text{III.118})$$

- Calculer le gradient qui est le produit de w^{k+1} par la matrice du complément de Schur dual D .

$$g^{k+1} = Dw^{k+1} = \sum_i B_i x_i^{k+1+} \quad (\text{III.119})$$

- Suivre les étapes de projection et préconditionnement (III.103), (III.104), (III.106), (III.107), (III.108). Le vecteur z^k est alors le gradient g^k préconditionné projeté.
- Calculer le coefficient de conjugaison γ^k et mettre à jour les champs :

$$\gamma^k = -\frac{(Dw^{k+1}, Dw^k)}{(Dw^k, Dw^k)} \quad (\text{III.120})$$

$$w^{k+1} = \sum_{j=0}^k \gamma^j \cdot w^j \quad (\text{III.121})$$

$$z^{k+1} = \sum_{j=0}^k \gamma^j \cdot z^j \quad (\text{III.122})$$

III.7.3. Méthode OrthoDir projetée de la méthode primale-duale

On présente ici le schéma de résolution du système linéaire (III.29) de la méthode primale-duale. Comme d'accoutumé, on trouvera des passages caractéristiques à la méthode du complément de Schur, ainsi qu'à la méthode FETI, notamment le traitement spécial des certaines matrices singulières K_{ii} qui peuvent contenir des pivots nuls dans les degrés de liberté correspondant aux vitesses.

- Initialisation : les valeurs initiales en pression sont mises à zéro :

$$p_f^0 = 0 \quad (\text{III.123})$$

Calcul des valeurs initiales de λ^0 se fait selon (III.98)-(III.99) à l'exception du fait que les vecteurs b_i ne contiennent pas des pressions d'interface (voir la formulation (III.22)). Sur chaque sous-domaine flottant, on cherche à calculer le second membre initial c_i et ainsi trouver λ^0 .

$$c_i = -N_i^t b_i \quad (\text{III.124})$$

$$\lambda^0 = E(E^t E)^{-1} c \quad (\text{III.125})$$

- Initialisation : calcul de la solution initiale. On résout le problème local (III.23) sur chaque sous-domaine pour trouver la solution x_i^{0+} composée de toutes les vitesses et des pressions intérieures, ainsi que le vecteur s_i^0 des efforts sur l'interface. En présence des modes rigides, la solution x_i^{0+} doit être complétée par les mouvements de corps rigide en vitesse.

$$\begin{bmatrix} K_{ii} & K_{if} \\ K_{fi} & K_{ff}^{(i)} \end{bmatrix} \begin{bmatrix} x_i^0 \\ 0 \end{bmatrix} = \begin{bmatrix} b_i + B_i^t \lambda^0 \\ b_f^{(i)} \end{bmatrix} \quad (\text{III.126})$$

$$x_i^{0+} = K_{ii}^+ (b_i + B_i' \lambda^0) \quad \text{et} \quad s_i^0 = b_f^{(i)} - K_{fi} x_i^{0+} \quad (\text{III.127})$$

- Initialisation : calculer le résidu initial g^0 comme (III.35). La partie vitesse étant le saut des solutions à travers les interfaces, la partie pression est assemblée des contributions s_i^0 de tous les sous-domaines

$$g^0 = \begin{bmatrix} g_v^0 \\ g_p^0 \end{bmatrix} = \begin{bmatrix} \sum_i B_i x_i^{0+} \\ \sum_i s_i^0 \end{bmatrix} \quad (\text{III.128})$$

- Initialisation : calculer le gradient projeté Pg^0 . Pour cela, on résout le système (III.73) et on trouve des coefficients des mouvements de corps rigide α^0 :

$$\alpha^0 = -(E' E)^{-1} E' g^0 \quad (\text{III.129})$$

Le gradient projeté Pg^0 est calculé alors comme :

$$Pg^0 = [I - E(E' E)^{-1} E'] g^0 = g^0 + E \alpha^0 \quad (\text{III.130})$$

- Initialisation : direction de descente initiale w^0 :

$$w^0 = Pg^0 \quad (\text{III.131})$$

- Initialisation : calculer le nouveau gradient comme le produit de la direction de descente w^0 par la matrice M . Cela se fait en deux étapes : on résout d'abord le problème suivant pour trouver la solution x_i^0 et le champs d'interface s_i^0 .

$$\begin{bmatrix} K_{ii} & K_{if} \\ K_{fi} & K_{ff}^{(i)} \end{bmatrix} \begin{bmatrix} x_i^0 \\ w_p^0 \end{bmatrix} = \begin{bmatrix} B_i' w_v^0 \\ 0 \end{bmatrix} \quad (\text{III.132})$$

$$x_i^{0+} = K_{ii}^+ (B_i' w_v^0 - K_{if} w_p^0) \quad \text{et} \quad s_i^0 = K_{ff}^{(i)} w_p^0 - K_{fi} x_i^{0+} \quad (\text{III.133})$$

Le produit Mw^{0+} est composé de deux parties : en vitesse c'est le saut des solutions particulières x_i^{0+} , en pression – l'assemblage des efforts d'interface s_i^0 :

$$Mw^{0+} = \begin{bmatrix} Mw_v^{0+} \\ Mw_p^{0+} \end{bmatrix} = \begin{bmatrix} \sum_i B_i x_i^{0+} \\ \sum_i s_i^0 \end{bmatrix} \quad (\text{III.134})$$

- Initialisation : calculer le vecteur z^0 qui est la projection du vecteur Mw^{0+} (III.129)-(III.130), suivi par l'étape de préconditionnement et de re-projection.

À l'itération $(k+1)$ les champs λ^k , p^k , x^k , s^k , g^k et w^k étant connus, elle consiste à effectuer des étapes :

- Calculer le coefficient de descente ρ^k et mettre à jour les champs :

$$\rho^k = -\frac{(z^k, Mw^k)}{(Mw^k, Mw^k)} \quad (\text{III.135})$$

$$\begin{bmatrix} \lambda^{k+1} \\ p_f^{k+1} \end{bmatrix} = \begin{bmatrix} \lambda^k \\ p_f^k \end{bmatrix} + \rho^k \cdot \begin{bmatrix} w_v^k \\ w_p^k \end{bmatrix} \quad (\text{III.136})$$

$$g^{k+1} = g^k + \rho^k \cdot Mw^k \quad (\text{III.137})$$

- La nouvelle direction de descente est

$$w^{k+1} = g^{k+1} \quad (\text{III.138})$$

- Pour la nouvelle direction de descente w^{k+1} connue, résoudre le problème

$$\begin{bmatrix} K_{ii} & K_{if} \\ K_{fi} & K_{ff}^{(i)} \end{bmatrix} \begin{bmatrix} x_i^{k+1} \\ w_p^k \end{bmatrix} = \begin{bmatrix} B_i^t w_v^k \\ 0 \end{bmatrix} \quad (\text{III.139})$$

Et trouver sa solution particulière x_i^{k+1+} et le vecteur d'efforts s_i^{k+1}

$$x_i^{k+1+} = K_{ii}^{-1} [B_i^t w_v^k - K_{if} w_p^k] \text{ et } s_i^{k+1} = K_{ff}^{(i)} w_p^k - K_{fi} x_i^{k+1+} \quad (\text{III.140})$$

- Calculer le produit de w^{k+1} par la matrice du complément de Schur primale-duale M .

$$Mw^{k+1} = \begin{bmatrix} Mw_v^{k+1} \\ Mw_p^{k+1} \end{bmatrix} = \begin{bmatrix} \sum B_i x_i^{k+1+} \\ \sum s_i^{k+1} \end{bmatrix} \quad (\text{III.141})$$

- Suivre les étapes de projection du vecteur Mw^{k+1} (III.129)-(III.130), de préconditionnement et de re-projection pour trouver le vecteur z^k .
- Calculer le coefficient de conjugaison γ^k et mettre à jour les champs :

$$\gamma^k = -\frac{(Mw^{k-1}, Mw^k)}{(Mw^{k-1}, Mw^{k-1})} \quad (\text{III.142})$$

$$w^{k+1} = \sum_{j=0}^k \gamma^j \cdot w^j \quad (\text{III.143})$$

$$z^{k+1} = \sum_{j=0}^k \gamma^j \cdot z^j \quad (\text{III.144})$$

III.8. CONCLUSION

Dans ce chapitre, on a présenté trois méthodes de décomposition de domaine, dont deux méthodes classiques : celle du complément de Schur et FETI. La troisième méthode hybride se présente comme une généralisation des approches primale et duale. L'ensemble des concepts relatifs

au préconditionnement (éventuellement approché) et aux problèmes des modes de corps rigide sont respectés.

Les nouvelles opportunités offertes par l'approche hybride pour la résolution des problèmes mécaniques issus de Forge3[®] sont évoquées dans le quatrième chapitre.

Nous n'avons pas cherché dans le cadre de ces travaux d'utiliser des versions plus sophistiquées de deux méthodes classiques. On pourrait citer ici la méthode FETI à deux niveaux [Farhat et al 2000a], ou le choix du projecteur (III.69) est plus compliqué que présenté dans ce travail. Il existe une famille des méthodes FETI-DP [Farhat et al 2000b] consacrées au traitement spécial des coins multiples des partitions. La méthode BDD (Balancing Domain Decomposition, [Mandel 1993, Tallec 1994]) est une méthode primale qui incorpore dans son algorithme de résolution un aspect multi-échelle augmentant ainsi son taux de convergence. Pour cette première expérience avec des méthodes de décomposition de domaine, nous nous sommes alors limités à une étude assez basique qui reste néanmoins un challenge passionnant.

Chapitre IV. MÉTHODES DE DÉCOMPOSITION DE DOMAINE : TESTS

On présente dans ce chapitre un certain nombre des cas test, assez simples, qui ont permis néanmoins de valider les équations du chapitre précédant et même d'évaluer les comportements des trois méthodes de décomposition de domaines. Tous les calculs ont été effectués sur une machine mono-processeur. Il n'est pas donc question ici de mesurer les performances des méthodes. Néanmoins, il est intéressant de comparer les temps de résolution pour les trois méthodes étudiées entre elles.

IV.1. ÉCRASEMENT D'UNE BARRE LONGUE

Il s'agit du test d'écrasement entre outils plans d'une barre assez longue, caractéristique d'un écoulement de forgeage et choisit pour ne pas produire de sous-domaines flottants lors du partitionnement (Figure IV.1).

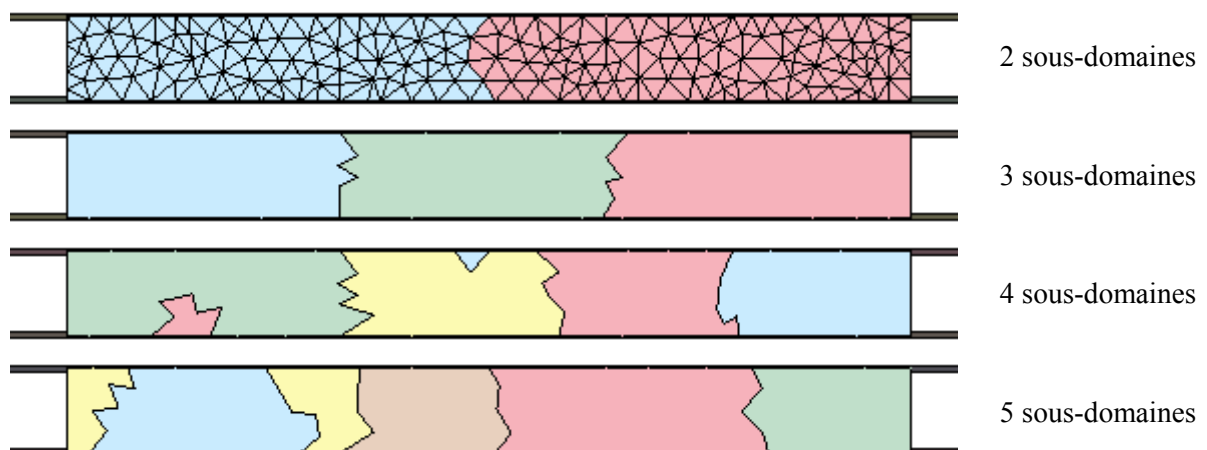


Figure IV.1. Écrasement d'une barre longue : partition en 2, 3, 4 et 5 sous-domaines, non nécessairement connexes

Le comportement du matériau est élasto-plastique avec une loi d'écroutissage de type puissance (I.25) avec $\beta = 0$. Les paramètres sont donnés dans le Tableau IV.1. On n'introduit pas de frottement et donc considère le contact comme bilatéral collant.

Paramètre	K_0 (MPa)	ε_0	n	E (MPa)	γ
Valeur	84,99	3,059	0,585	$1,1 \cdot 10^5$	0,3

Tableau IV.1. Écrasement d'une barre longue : paramètres de la rhéologie

Le maillage initial est constitué de 1 546 nœuds (6 184 degrés de liberté) et 6 333 éléments tétraédriques. Il est ensuite découpé en 2, 3, 4 et 5 sous-domaines (voir Figure IV.1), dont les caractéristiques sont données dans le Tableau IV.2, avec Sd le numéro de sous-domaine, $Nbnoe$ le nombre de nœuds par maillage, $n.d.d.l$ le nombre de degrés de liberté.

Partition	Ns	$Nbnoe$	$n.d.d.l$	Nombre de domaines voisins	$Nbnoe$ d'interface	$n.d.d.l$ d'interface
2 sous-domaines	1	806	3 224	1	44	176
	2	784	3 136	1		
3 sous-domaines	1	554	2 216	1	38	152
	2	540	2 160	1	51	204
	3	452	1 808	2	89	356
4 sous-domaines	1	474	1 896	3	174	696
	2	379	1 516	2	65	260
	3	382	1 528	2	132	528
	4	311	1 244	3	117	468
5 sous-domaines	1	359	1 436	3	112	448
	2	342	1 368	2	80	320
	3	304	1 216	1	33	132
	4	281	1 124	3	187	748
	5	260	1 040	3	100	400

Tableau IV.2. Écrasement d'une barre longue : dimensions des maillages

Les calculs ont été effectués avec pour critère de convergence 10^{-9} et 10^{-6} . Le Tableau A.1 et le Tableau A.2 de l'Annexe A montrent les résidus finaux obtenus avec ces critères pour les trois méthodes utilisées sans préconditionnement. Dans toutes les partitions considérées et pour toutes les méthodes, l'ordre de grandeur des résidus à la convergence est de 10^{-10} et de 10^{-7} , respectivement, ce qui correspond à la précision imposée.

On constate que le nombre d'itérations augmente avec le nombre de sous-domaines, ce qui est logique, puisque l'interface, et donc la dimension du problème itératif à résoudre, devient plus grande. En outre, la partition de ce genre dite « par bande » n'est pas très avantageuse. L'augmentation du temps de résolution s'explique par le fait que les calculs ont été effectués sur une machine mono-

processeur. Ces résultats sont néanmoins intéressants pour comparer les comportements des trois méthodes étudiées.

Pour une convergence à 10^{-9} , FETI est plus efficace que la méthode duale primale, ce qui apparaît sur le nombre d'itérations à convergence, cette dernière ayant plus de difficulté à atteindre la norme de résidu demandé (cf. Figure IV.2). La Figure IV.3 et la Figure IV.4 montrent les temps de résolution et le coût d'une itération pour les trois méthodes.

En revanche, pour atteindre la précision de 10^{-6} , et pour plus de 3 sous-domaines, la méthode primale-duale s'avère plus rapide que FETI. En fait, le nombre d'itérations à convergence est semblable (cf. Figure IV.5), mais le coût d'une itération de primale-duale est inférieur, ce qui explique la différence.

La Figure IV.6 présente les temps de résolution du même système linéaire issu de Forge3, et la Figure IV.7 présente le temps de calcul pour une seule itération. On note que la méthode du complément de Schur nécessite le plus de temps et d'itérations pour converger, et que la méthode primale-duale se comporte mieux que les deux autres méthodes. Les résultats de la méthode FETI sur quatre et cinq sous-domaines semblent moins bons. Cela est probablement dû aux partitions non connexes, les deux autres méthodes gérant mieux cette particularité. Par ailleurs, sur plus de deux sous-domaines, le temps pour effectuer une itération est comparable pour les trois méthodes, avec toutefois un avantage pour la méthode hybride.

On a effectué les mêmes calculs, pour le même système linéaire, avec les trois méthodes préconditionnées cette fois avec les préconditionneurs simples (allégés) : Q_d^{-1} pour la méthode du complément de Schur (III.41), Q_l^{-1} pour la méthode FETI (III.46) et Q_{ld}^{-1} pour la méthode primale-duale (III.47). Le critère de convergence est de 10^{-6} . Le Tableau A.3 de l'Annexe A donne les valeurs des résidus finaux, les nombres d'itérations et les temps de résolution pour les trois méthodes et les quatre partitions. Le critère de convergence est atteint avec toutes les méthodes. La méthode FETI converge même avec un résidu de l'ordre de 10^{-8} . L'utilisation des préconditionneurs simples (non optimaux) diminue d'une manière considérable le nombre d'itérations (cf. Figure IV.8) et donc le temps de résolution (cf. Figure IV.9) sans pour autant augmenter de manière trop importante le temps nécessaire pour une itération (Figure IV.10).

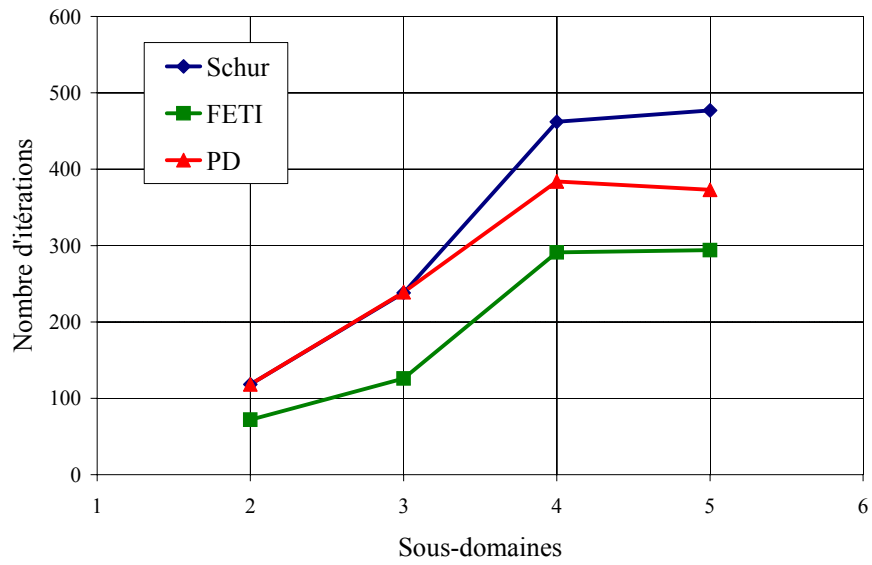


Figure IV.2. Écrasement d'une barre : nombre d'itérations nécessaires pour la résolution du système linéaire par les trois méthodes utilisées sans préconditionnement, précision de 10^{-9}

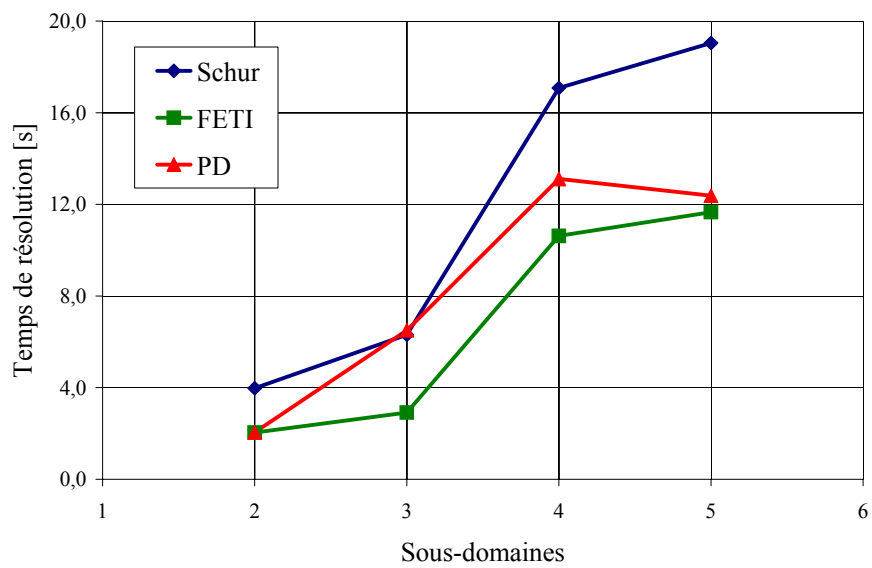


Figure IV.3. Écrasement d'une barre : temps de résolution du système linéaire par les trois méthodes utilisées sans préconditionnement, précision de 10^{-9}

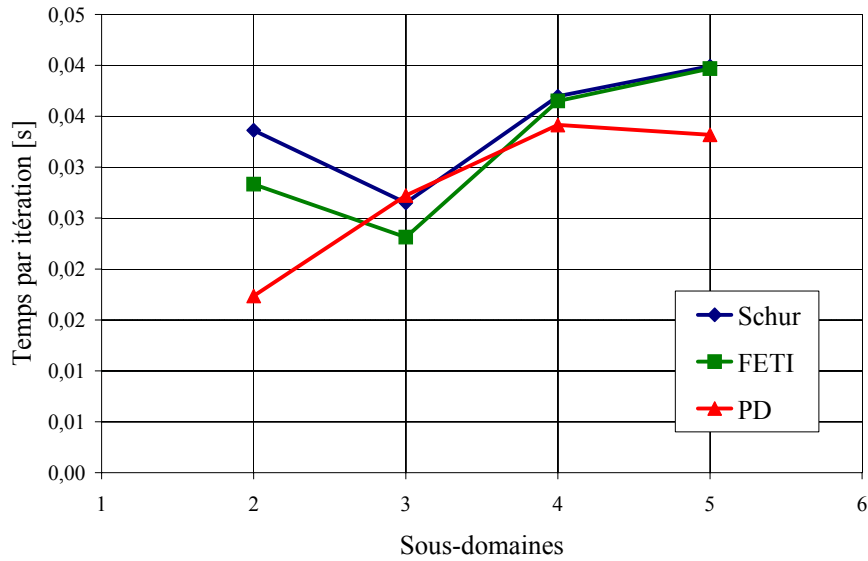


Figure IV.4. Écrasement d'une barre : temps pour une itération de la résolution du système linéaire par les trois méthodes utilisées sans préconditionnement, précision de 10^{-9}

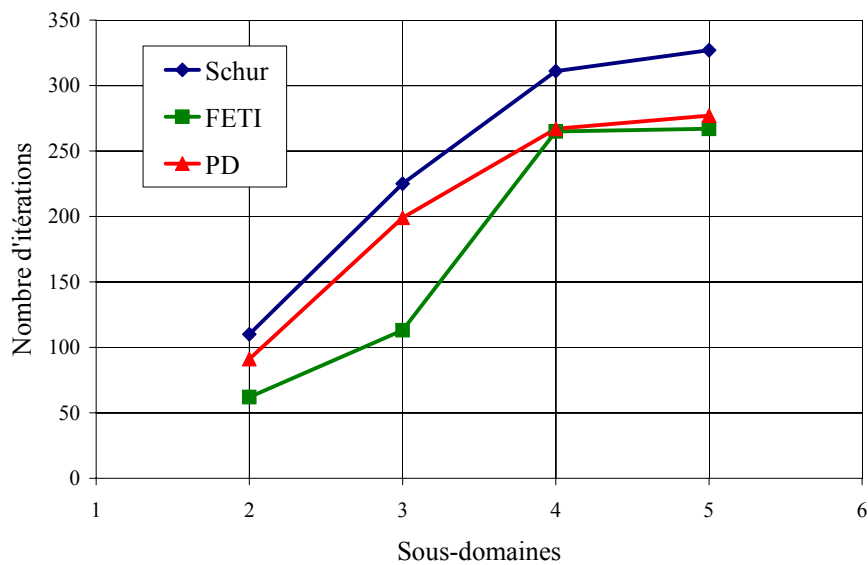


Figure IV.5. Écrasement d'une barre : nombre d'itération nécessaire pour la résolution du système linéaire par les trois méthodes utilisées sans préconditionnement, précision de 10^{-6}

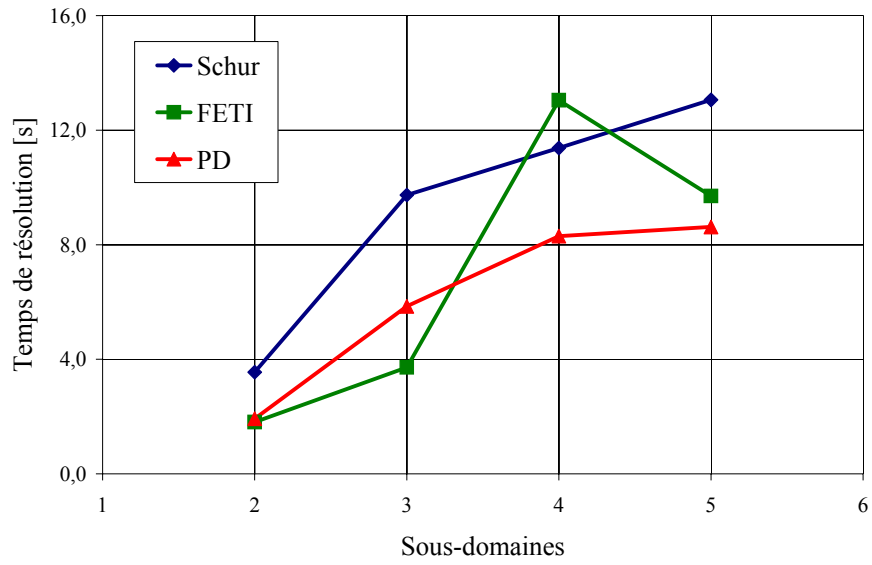


Figure IV.6. Écrasement d'une barre : temps de résolution du système linéaire par les trois méthodes utilisées sans préconditionnement, précision de 10^{-6}

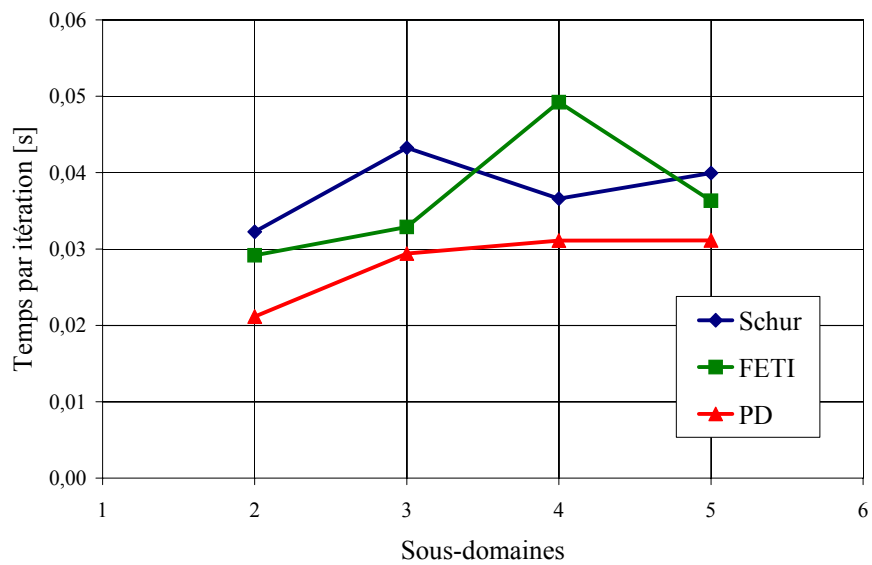


Figure IV.7. Écrasement d'une barre : temps pour une itération de la résolution par les trois méthodes utilisées sans préconditionnement, précision 10^{-6}

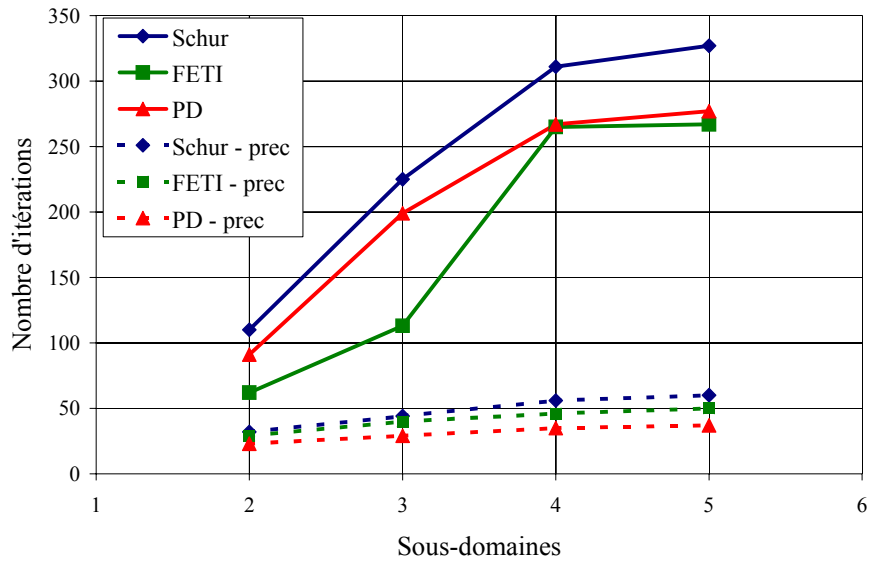


Figure IV.8. Écrasement d'une barre : nombre d'itérations pour la résolution du système linéaire sans et avec préconditionnement, précision de 10^{-6}

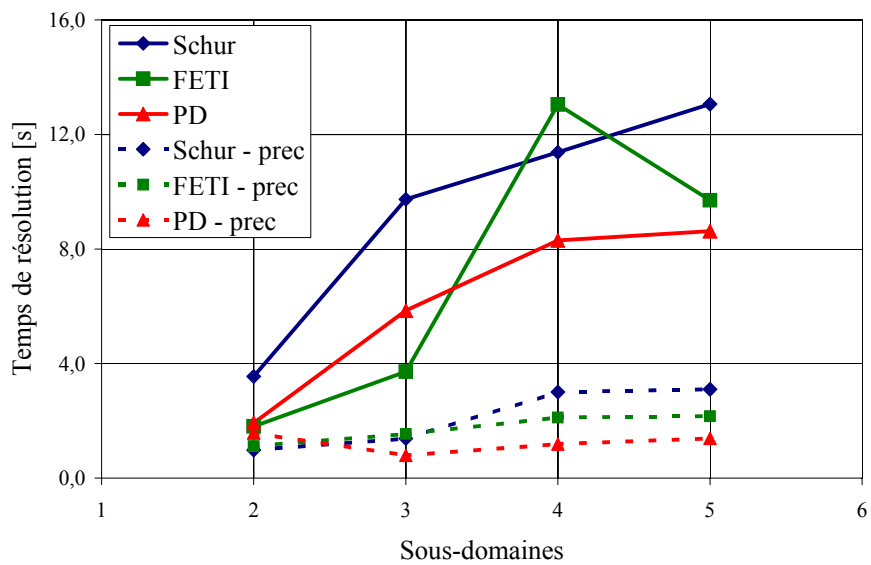


Figure IV.9. Écrasement d'une barre : temps de résolution du système linéaire sans et avec préconditionnement, précision de 10^{-6}

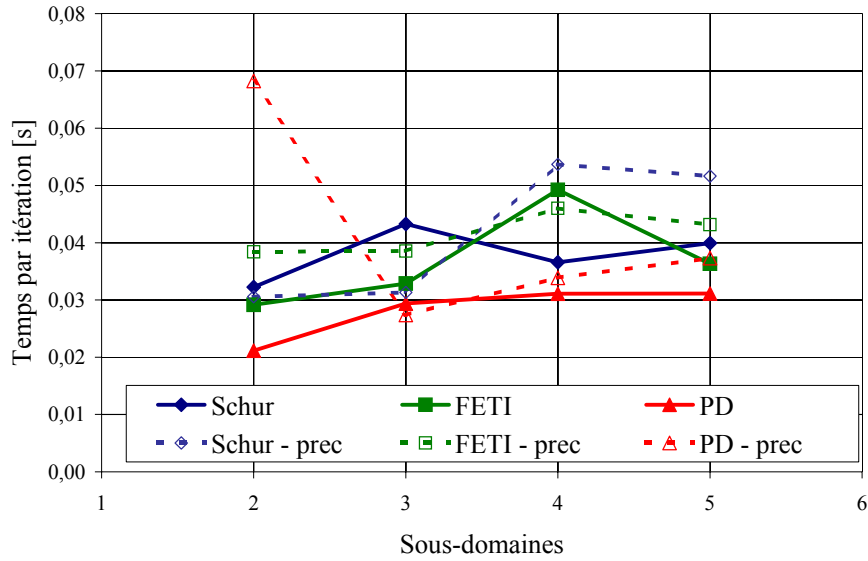


Figure IV.10. Écrasement d'une barre : temps pour une itération sans et avec préconditionnement, précision de 10^{-6}

On constate que le coût d'une itération de la méthode primale-duale est inférieur à celui de deux autres méthodes. Les résultats de la méthode hybride sur 2 sous-domaines semblent étranges, mais en dehors de ce cas, le comportement est bien régulier : la nouvelle méthode primale-duale se montre meilleure que la méthode FETI, qui est à son tour plus efficace que la méthode du complément de Schur.

IV.2. CAS DU TRIAXE

Considérons le cas de forgeage d'un triaxe (Figure IV.11 et Figure IV.12). Le comportement du matériau est modélisé par la loi visco-plastique de Norton-Hoff (I.22) avec la loi d'érouissage donnée par $K(T, \bar{\varepsilon}) = K_0 (\bar{\varepsilon} + \varepsilon_0)^n \exp(-\beta T)$. Le contact entre la pièce et les outils est considéré comme unilatéral glissant et le frottement est décrit par la loi de type viscoplastique (I.42). Les paramètres de ces lois sont donnés dans le Tableau IV.3. La vitesse de pilotage de l'outil supérieur suivant la direction de l'axe z est de 100 mm/s .

Paramètre	K_0 (MPa)	m	ε_0	β	n	α_f	p_f
Valeur	1 693,0	0,1390	$1,0 \cdot 10^{-3}$	$0,244 \cdot 10^{-2}$	0,113	0,4	0,139

Tableau IV.3. Cas du triaxe : paramètres de la rhéologie et du frottement

IV.2.1. Calcul sur 2 sous-domaines

Le maillage initial, constitué de 1 053 nœuds et 3 602 éléments, est partitionné en deux sous-domaines dont les caractéristiques sont données dans le Tableau IV.4.

<i>Sous-domaine</i>	<i>Nbnoe</i>	<i>n.d.d.l</i>	<i>Nbelt</i>	<i>Nbnoe d'interface</i>	<i>n.d.d.l. d'interface</i>
1	549	2 196	1 801	51	204
2	555	2 220	1 801		

Tableau IV.4. Cas du triaxe : dimensions de la partition en deux sous-domaines

On considère le système linéaire issu du premier incrément de la simulation avec Forge3®. La Figure IV.11 montre la partition en deux sous-domaines (a), le champ de déformation équivalente pendant le premier incrément de la simulation (b) et le champ de pression (c).

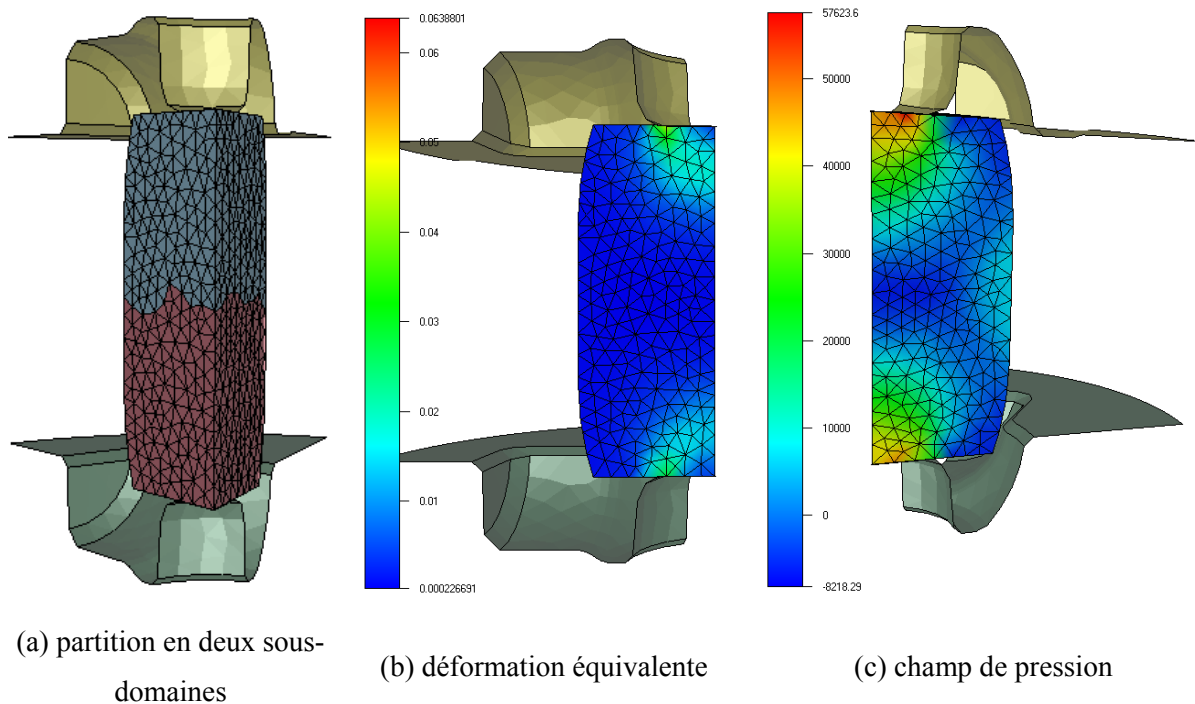


Figure IV.11. Cas du triaxe : calcul sur deux sous-domaines

La résolution du système linéaire a été effectuée avec un critère de convergence de 10^{-6} . Les résultats des calculs pour les trois méthodes de décomposition de domaine, sans préconditionnement et avec les deux types de préconditionneurs, sont donnés dans le Tableau IV.5. Pour les trois méthodes et tous les préconditionneurs (sauf le préconditionneur complet de la méthode primale-duale) les normes des résidus finaux correspondent bien à la précision voulue.

<i>Méthode</i>	<i>Préconditionneur</i>	<i>Résidu final</i>	<i>Nombre d'itérations</i>	<i>Temps (s)</i>	<i>Temps par itération(s)</i>
Schur	–	9,79E-07	135	3,216	0,024
	Diagonal	9,22E-07	41	1,005	0,025
	Neumann	4,48E-07	14	0,691	0,049
FETI	–	3,86E-08	87	8,954	0,103
	« <i>Lumped</i> »	1,19E-09	33	4,387	0,133
	Dirichlet	5,05E-10	25	6,173	0,247
Primale-Duale	–	1,86E-06	88	2,336	0,027
	Allégé	2,41E-07	19	0,446	0,023
	Complet	1,75E-05	29	2,067	0,071

Tableau IV.5. Cas du triaxe : résultats des calculs sur deux sous-domaines

Dans la méthode du complément de Schur, l'utilisation des préconditionneurs par la diagonale et de Neumann réduit le nombre d'itérations de 70% et 90% respectivement par rapport au calcul sans préconditionnement. La performance du préconditionneur de Neumann qui est le préconditionneur optimal de la méthode primale, est spectaculaire. Même si le temps d'une itération augmente par un facteur deux par rapport au préconditionneur allégé, le gain de 66% en nombre d'itérations se traduit par un gain en temps complet de la résolution du système.

Pour la méthode FETI, les résultats sont un peu moins brillants. En utilisant le préconditionneur de Dirichlet, on gagne 71% en nombre d'itérations par rapport au cas sans préconditionneur. Cependant, l'augmentation du temps d'une itération avec ce préconditionneur coûteux anéantit les 15% de gain par rapport au calcul avec le préconditionneur allégé.

Quant à la méthode primale-duale, l'utilisation du préconditionneur simple réduit le nombre d'itérations de 88% sans pour autant augmenter le temps d'une itération. Par ailleurs, le comportement du préconditionneur complet n'est pas du tout satisfaisant. Non seulement le résidu final n'atteint pas le critère de convergence, mais en plus le nombre d'itérations augmente par rapport au préconditionneur simple. Néanmoins, la méthode primale-duale se comporte assez bien et voire mieux (avec le préconditionneur simple) que les deux méthodes classiques.

IV.2.2. Calcul sur 3 sous-domaines

On considère maintenant une étape plus avancée de simulation, où l'outil est descendu de 60mm à 13,6mm, et où la bavure commence à se former (Figure IV.12). La procédure de partitionnement ne produit pas de sous-domaines flottants. Chaque sous-domaine dispose de suffisamment de conditions

aux limites, ce qui permet d'éviter le calcul des modes rigides. Les caractéristiques des trois sous-maillages sont données dans le Tableau IV.6. Chaque sous-domaine a deux sous-domaines voisins. Il est à noter que la partition considérée possède quelques points multiples, c'est-à-dire des nœuds partagés par les trois sous-domaines.

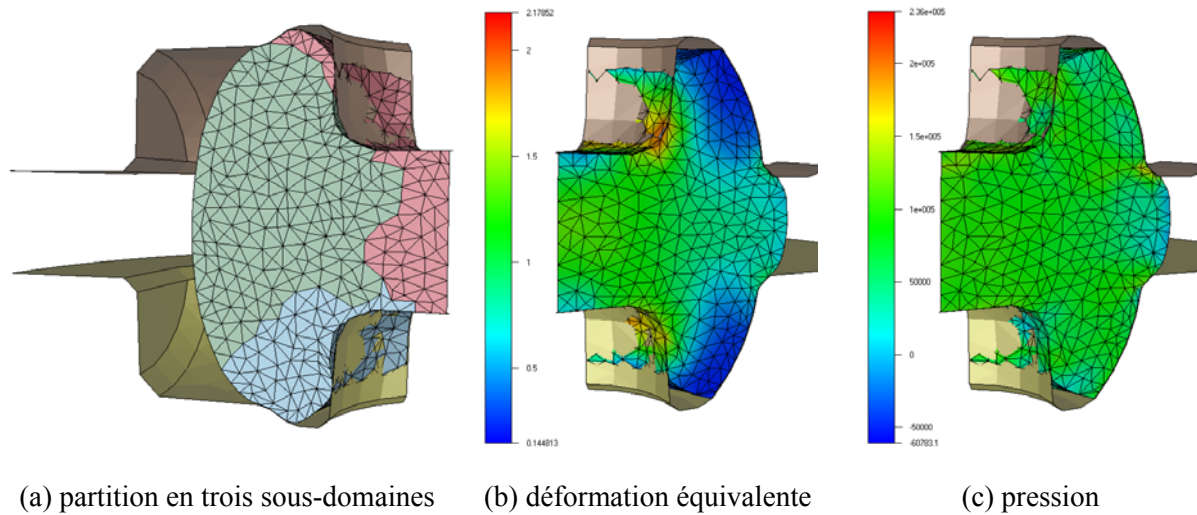


Figure IV.12. Cas du triaxe : calcul sur trois sous-domaines au milieu de procédé

<i>Sous-domaine</i>	<i>Nbnoe</i>	<i>n.d.d.l</i>	<i>Nbelt</i>	<i>Nbnoe d'interface</i>	<i>n.d.d.l. d'interface</i>
1	638	2 552	1 773	141	564
2	594	2 376	1 719	97	388
3	567	2 268	1 713	156	624

Tableau IV.6. Cas du triaxe : dimensions de partition en trois sous-domaines

La résolution du système linéaire a été effectuée avec le critère de convergence de 10^{-6} . Les résultats des calculs pour les trois méthodes de décomposition de domaine sans préconditionnement et avec des préconditionneurs allégés sont donnés dans le Tableau IV.7. La méthode primale-duale se comporte d'une manière assez satisfaisante, même si dans ce cas on observe un moins bon respect de la norme de convergence avec préconditionneur allégé, et que la méthode apparaît ainsi un peu moins performante que FETI.

<i>Méthode</i>	<i>Préconditionneur</i>	<i>Résidu final</i>	<i>Nombre d'itérations</i>	<i>Temps (s)</i>	<i>Temps par itération(s)</i>
Schur	–	8,97E-07	327	17,504	0,054
	Allégé	9,67E-07	125	6,606	0,053
FETI	–	9,47E-07	577	62,407	0,108
	Allégé	3,70E-08	54	4,181	0,077
Primale-Duale	–	8,82E-07	373	22,095	0,059
	Allégé	1,72E-05	72	4,196	0,058

Tableau IV.7. Cas du triaxe : résultats des calculs sur trois sous-domaines

IV.3. UNE POUTRE ENCASTRÉE

L'objectif de ce cas test est de valider le traitement des modes rigides dans la méthode primale-duale. Cela se fait par rapport à la solution fournie par la méthode FETI. On teste aussi le comportement des préconditionneurs de la méthode de Schur et de la méthode hybride qui dans ce cas nécessitent un traitement spécial dû à la présence des mouvements de corps rigide.

Le comportement du matériau considéré est élastique avec un module d'Young $E = 2,0 \cdot 10^5 \text{ MPa}$ et un coefficient de Poisson $\gamma = 0,3$. On n'introduit pas de frottement et on considère donc le contact comme bilatéral collant. Le maillage initial est constitué de 934 nœuds et 3 902 éléments. Il est partitionné en trois sous-domaines (Figure IV.13), les paramètres de ces sous-maillage sont donnés dans le Tableau IV.8. Le sous-domaine #3 partage son interface avec les deux autres sous-domaines. L'interface globale est constituée de 98 nœuds (392 degrés de liberté). Cette partition par bandes génère des modes rigides dans le sous-domaine #3, en vert sur la Figure IV.13.

<i>Sous-domaine</i>	<i>Nbnoe</i>	<i>n.d.d.l</i>	<i>Nbelt</i>	<i>Nbnoe d'interface</i>	<i>n.d.d.l. d'interface</i>
1	347	1 388	1 311	52	208
2	329	1 316	1 272	46	184
3	356	1 424	1 319	98	392

Tableau IV.8. Cas d'une barre encastrée : dimensions de partition en trois sous-domaines

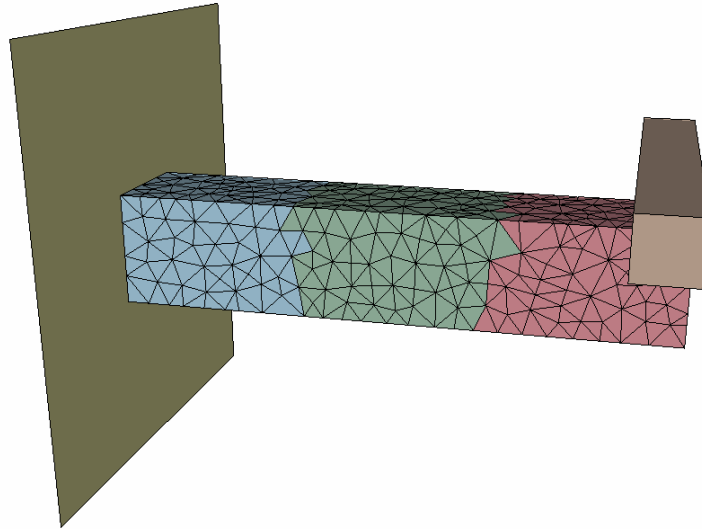


Figure IV.13. Partition en trois sous-domaines d'une poutre encastree

Les calculs ont été effectués avec un critère de convergence de 10^{-9} . Les résultats sont donnés dans le Tableau IV.9. La solution obtenue par la méthode FETI a été choisie comme référence pour les deux méthodes. L'écart entre les solutions obtenues par ces deux méthodes et celle de FETI est tout à fait satisfaisant, ce qui montre que la méthode primale-duale (ainsi que la méthode de Schur) est capable de résoudre le problème en présence des modes rigides.

Méthode	Préconditionneur	Nombre d'itérations	Résidu final	Norme de solution $\ x\ $	$\frac{\ x(FETI) - x\ }{\ x\ }$
FETI	–	51	2,45E-10	17,7527508	–
	« Lumped »	36	1,29E-10	17,7527509	–
	Dirichlet	31	7,89E-11	17,7527508	–
Schur	–	139	5,81E-10	17,7527508	1,4464E-16
	Diagonal	142	4,85E-10	17,7527508	5,8002E-15
	Neumann	50	2,76E-10	17,7527508	4,3162E-17
Primale-Duale	–	50	3,41E-10	17,7527508	4,9871E-19
	Allégé	35	2,76E-10	17,7527508	5,9729E-15
	Complet	50	3,41E-10	17,7527508	2,3207E-18

Tableau IV.9. Cas d'une barre encastree : résultats pour les trois méthodes avec les différents préconditionneurs.

Étant donné qu'aucun traitement spécial n'a été implémenté pour gérer les modes rigides dans la méthode du complément de Schur, elle a mis davantage d'itérations pour trouver la solution avec la

précision donnée. Les calculs avec les préconditionneurs accélèrent la convergence. L'utilisation du préconditionneur de Neumann donne, notamment, le même nombre d'itérations que la méthode FETI non préconditionnée.

Le comportement de la méthode primale-duale ressemble beaucoup à celui de la méthode FETI, surtout dans le cas sans préconditionnement et avec le préconditionneur allégé. Dans ces deux cas, les courbes sont presque identiques (cf. Figure IV.14). Par ailleurs, encore une fois, l'utilisation du préconditionneur complet pour la méthode primale-duale s'avère inutile. Même si la méthode converge, aucun gain en vitesse de convergence n'est observé. Dans ce cas, le résidu suit tout simplement celui du résidu non-préconditionné.

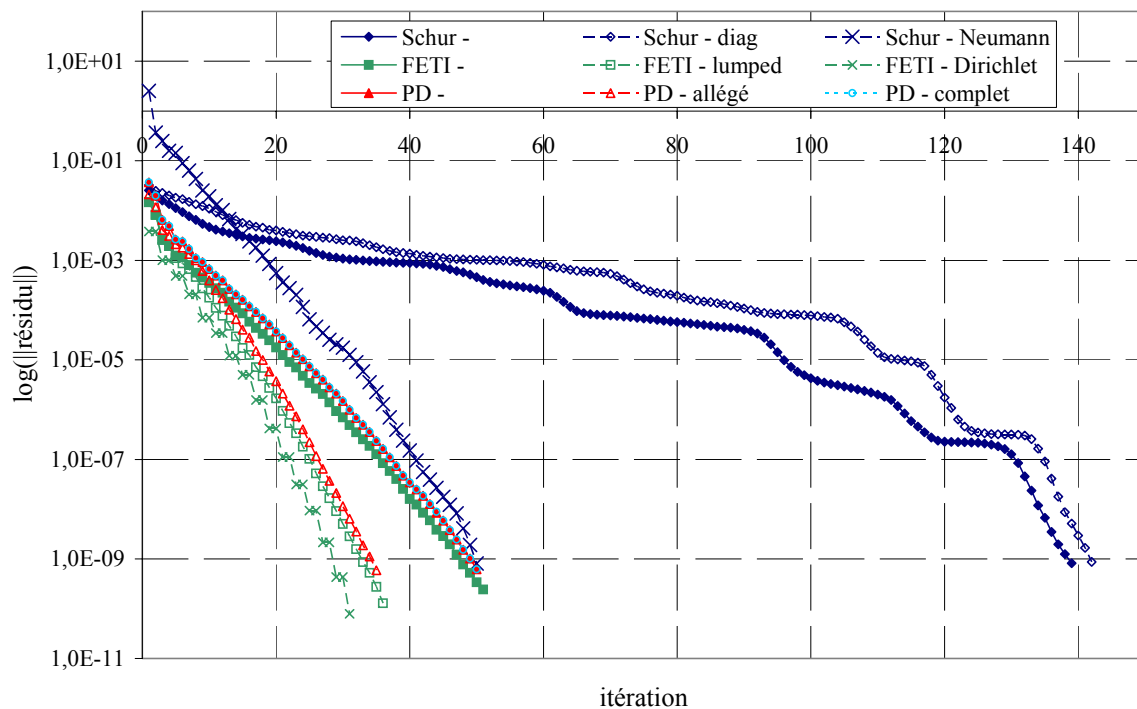


Figure IV.14. Barre encastrée : courbes de convergence pour les trois méthodes.

Il faut noter tout de même que la partition choisie ne génère pas des points multiples (des nœuds appartenant à plus de deux sous-domaines). Les essais effectués sur la même barre, mais partitionnée pour produire des modes rigides et des points multiples, on démontré la nécessité de ce traitement pour la méthode primale-duale.

IV.4. CONCLUSION

Les deux méthodes classiques de décomposition de domaines convergent dans tous les cas testés sans traitement spécifique lié à la formulation vitesse/pression. La nouvelle méthode hybride converge dans tous les cas étudiés, avec ou sans modes rigides.

Les préconditionneurs simples de toutes les trois méthodes sont manifestement efficaces. Il n'y a que pour la méthode du complément de Schur que la nécessité du préconditionneur complet est apparue. Cela est probablement dû au fait que le préconditionneur allégé rapproche moins le préconditionneur de Neumann que le préconditionneur localisé de la méthode FETI ne rapproche le préconditionneur de Dirichlet.

Le préconditionneur complet de la méthode hybride est encore à améliorer. Le coût de son utilisation reste assez important puisqu'il ne réduit pas assez le nombre d'itérations. En outre, le contrôle du résidu est moins bon qu'avec le préconditionneur allégé ou sans préconditionnement. La construction de ce préconditionneur suivait la même logique que l'élaboration de la méthode elle-même : mélange de deux méthodes classiques pour prendre en compte la formulation vitesse/pression. S'agit-il alors d'une faute dans le raisonnement (et donc du problème de fond) ou est-ce simplement une erreur d'implémentation d'une idée juste ? La réponse à cette question reste à découvrir...

Les modes rigides sont traités pour la méthode primale-duale. Dans le cas de la poutre encastree, le comportement de la méthode (à l'exception du calcul avec le préconditionneur complet) est semblable à celui de la méthode FETI dans lequel les modes rigides reçoivent un traitement spécial et bien efficace. Néanmoins, pour que la méthode soit efficace il faut apprendre à traiter des partitions arbitraires, surtout celles qui produisent des modes rigides et des points multiples, ce qui se produit très souvent dans la réalité.

Néanmoins, dans des nombreux cas d'étude présentés ici, la nouvelle méthode hybride s'avère plus efficace que la méthode FETI, elle même plus efficace que la méthode du complément de Schur. Ceci permet d'affirmer l'utilité du travail effectué et l'intérêt de l'utilisation de cette nouvelle méthode.

Chapitre V. RÉSULTATS D'APPLICATIONS

V.1. PRISE EN COMPTE DE LA RAIDEUR DE PRESSE

V.1.1. Cas Vaccari

Ce cas test a été proposé par une société de forge italienne, Vaccari, dans le cadre du projet IMPRESS. Il est relatif au forgeage d'une aube de compresseur. La Figure V.1a montre la géométrie initiale de la pièce à forger, où la géométrie de l'outil supérieur est représentée par des points noirs. C'est un cas de grande taille pour les calculs, car la pièce est maillée assez finement, le nombre de nœuds étant de 18 278 au début du forgeage et le nombre d'éléments de 82 517.

La pièce est en acier X20CrMo13, sa rhéologie est modélisée par la loi visco-plastique de Norton-Hoff (I.22) avec l'écrouissage de type Hansel et Spittel (I.26), dont les paramètres sont donnés dans le Tableau V.1. Ce cas de forgeage est réalisé à chaud, avec une température initiale de la pièce de 1 150°C. Celles des outils supérieur et inférieur sont de 240°C et 160°C, respectivement. Le contact avec les outils est considéré comme unilatéral glissant et le frottement est décrit par la loi de Coulomb (I.40) avec les coefficients $\bar{m} = 0,15$ et $\mu = 7,5 \cdot 10^{-2}$.

Paramètre	K_0 (MPa)	m_1	m_2	m_3	m_4
Valeur	2 435,5051	-0,00264	0,18735	0,08211	-0,00175

Tableau V.1. Cas Vaccari : paramètres de rhéologie

L'outillage est piloté par une presse à vis de 2500t. La Figure V.1b montre la forme de la pièce obtenue lors d'une simulation sans raideur de presse.

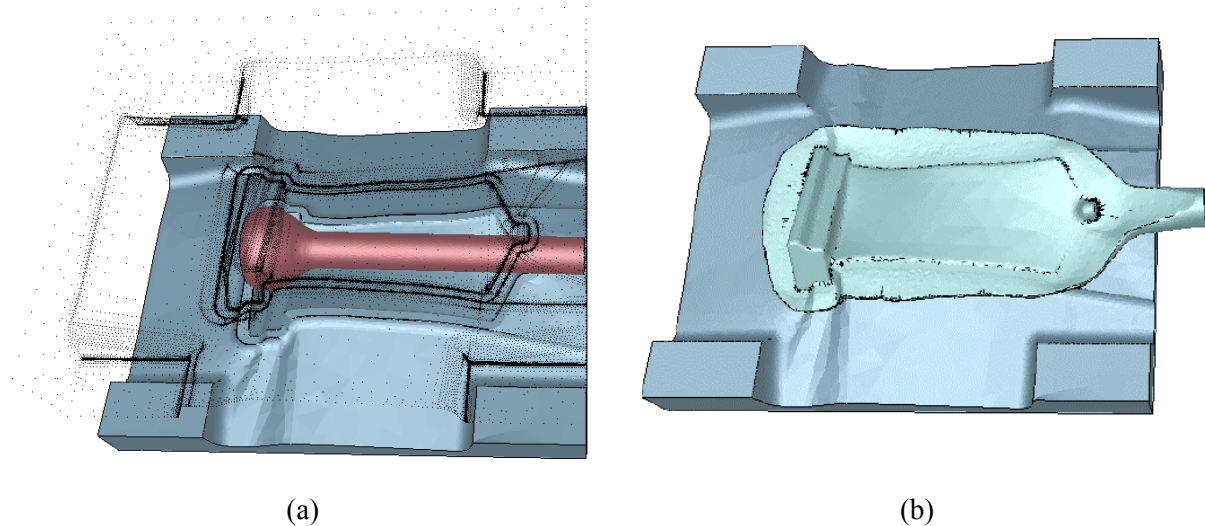
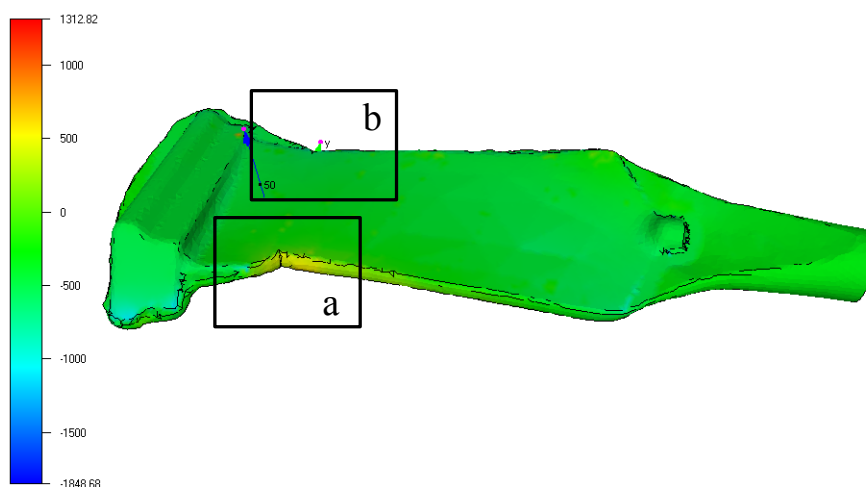


Figure V.1. Cas Vaccari : géométrie initiale (a) et forme finale de la pièce forgée obtenue sans prise en compte de la raideur de presse (b)

Trois coefficients de la matrice de rigidité de presse ont été ensuite introduits dans la simulation, assez arbitrairement, faute de données expérimentales à ce moment des travaux : $k_{11} = k_{22} = 5,0 \cdot 10^6 \text{ N/mm}$ et $k_{33} = 5,0 \cdot 10^9 \text{ N/mm}$. On constate que l'écoulement du matériau est significativement différent de celui simulé avec une presse parfaitement rigide. La forme finale prévue n'a pas été atteinte car la déformation élastique de la presse (exagérée, ici) modifie la course de forgeage. La presse à vis étant pilotée en énergie, et l'énergie de déformation élastique n'ayant pas été prise en compte lors de ces calculs, la matrice de forgeage ne va pas jusqu'au bout de la course réelle. On n'a donc pas simulé le forgeage jusqu'à l'obtention de la bavure. Toutefois, à cette étape de la simulation des replis (Figure V.2) sont observés. Ils sont bien plus importants que dans la simulation sans prise en compte de déformation de la presse, et sont aussi davantage conformes à la réalité rapportée par les experts forgerons du projet.



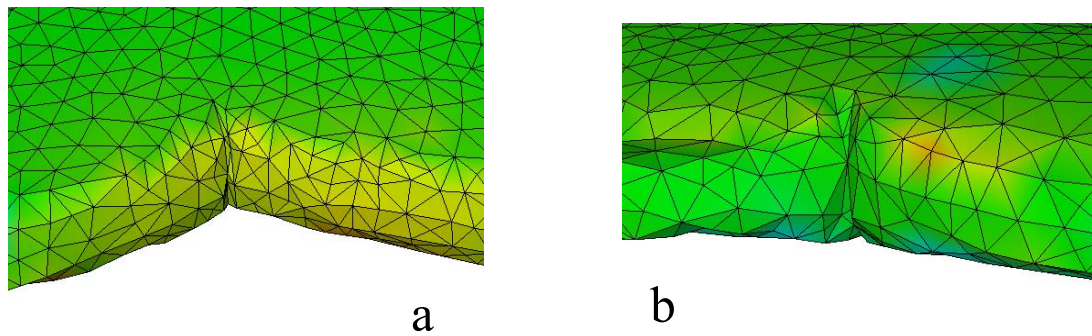


Figure V.2. Cas Vaccari : vitesse dans la direction de l'axe z à la fin de la simulation

V.1.2. Cas Smetek

Ce cas industriel, fourni par la compagnie danoise Smetek dans le cadre du projet IMPRESS, est constitué de trois passes de forgeage et d'une passe d'ébavurage, effectuées en une seule opération sur une unique presse. La première passe est la réalisation d'une collerette sur la partie supérieure du lopin, qui correspond à la collerette de la pièce finale. La deuxième passe est une opération durant laquelle l'ensemble de la géométrie est préformé. La troisième passe est la finition de la collerette avec la formation des 18 dents. Pour terminer, l'enlèvement de la bavure est effectué. Cette dernière étape d'ébavurage n'a pas été considérée lors de la simulation, car elle porte sur des efforts de forgeage très faibles vis-à-vis des autres opérations. La vue globale des trois lopins est présentée sur la Figure V.3 où un nuage de points représente les outils.

Pour pouvoir réaliser la simulation simultanée des trois passes, il a fallu d'abord effectuer les calculs séparés de chaque pièce afin d'obtenir les géométries initiales des deuxième et troisième lopins. Ces deux simulations consécutives ont aussi permis de calculer les champs de déformation, contrainte, température, etc... dans les préformes. Les dimensions des maillages (par passe) au début de la simulation sont données dans le Tableau V.2 :

	Passe №1	Passe №2	Passe №3	Total
Nombre de nœuds	7 070	7 086	20 179	34 335
Nombre d'éléments	35 055	28 774	82 604	146 433

Tableau V.2 Cas Smetek : dimensions des maillages au début des calculs

La rhéologie du matériau est modélisée par le comportement élasto-visco-plastique avec une loi d'écroutissage de type Hansel et Spittel (I.26) avec seulement deux coefficients non-nuls. Les paramètres sont donnés dans le Tableau IV.1.

Paramètre	K_0 (MPa)	m_2	E (MPa)	γ
Valeur	679,24	0,2624	$2,0 \cdot 10^5$	0,3

Tableau V.3. Cas Smetek : paramètres de la rhéologie

L'outillage est piloté par une presse hydraulique de 1000t dont la vitesse suivant la direction négative de l'axe z est de 100 mm/s . Le modèle de presse est appliqué à l'outil supérieur, et la matrice de raideur expérimentalement mesurée dans le cadre du projet IMPRESS est :

$$\begin{bmatrix} 3,5 \cdot 10^8 \frac{N}{mm} & 0 & 0 & 0 & 2,855 \cdot 10^8 \frac{N \cdot m}{mm} & 0 \\ 0 & 3,022 \cdot 10^8 \frac{N}{mm} & 0 & -3,665 \cdot 10^8 \frac{N \cdot m}{mm} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3,665 \cdot 10^8 \frac{N \cdot m}{mm} & 0 & 2,57 \cdot 10^9 \frac{N \cdot m^2}{mm} & 0 & 0 \\ 2,855 \cdot 10^8 \frac{N \cdot m}{mm} & 0 & 0 & 0 & 1,4 \cdot 10^9 \frac{N \cdot m^2}{mm} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{V.1})$$

Le contact avec les outils est considéré comme unilatéral glissant et le frottement est décrit par la loi de Coulomb (I.40) avec les coefficients $\bar{m} = 0,12$ et $\mu = 6,0 \cdot 10^{-2}$.

Sur la Figure V.4, on observe les formes finales des trois pièces forgées, expérimentalement et par la simulation numérique. Enfin, les déplacements de l'outil supérieur dus à la raideur de presse sont confrontés aux données expérimentales mesurées par IPU, un laboratoire danois du projet IMPRESS. Deux séries de mesures du déplacement selon l'axe x sont présentées sur la Figure V.5, ainsi que la courbe tracée à partir des données numériques. Même si le déplacement calculé à la fin de la simulation est plus grand que celui mesuré de $0,05 \text{ mm}$, les courbes se ressemblent de manière très satisfaisante. Quatre séries expérimentales de mesure des déplacements liés à la rotation autour de l'axe y sont présentées sur la Figure V.6. Le même déplacement calculé par la simulation numérique offre une évolution tout à fait similaire, validant ainsi le modèle de raideur de presse, son implémentation et les valeurs mesurées pour la matrice de raideur.

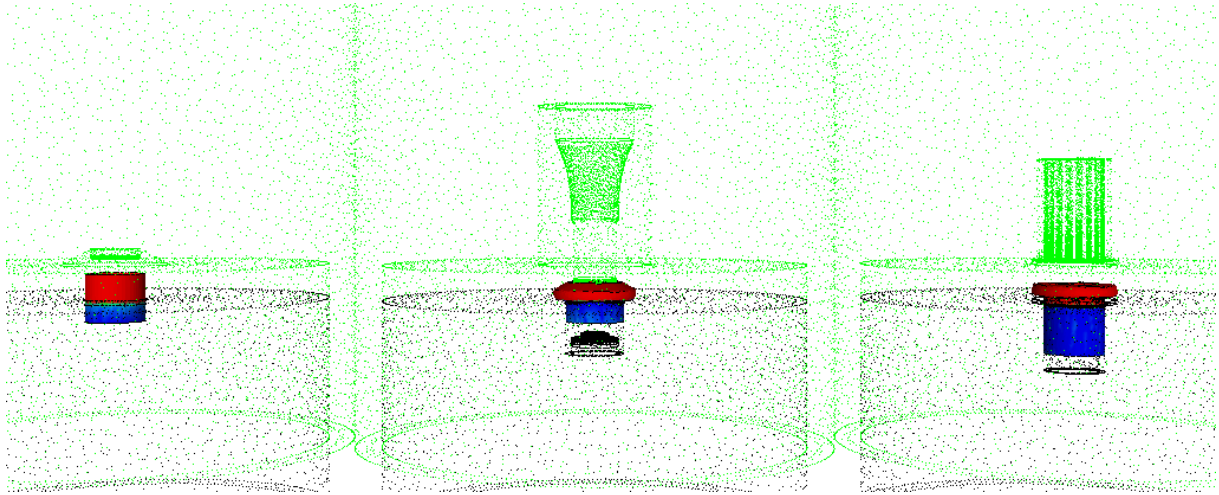


Figure V.3. Cas Smetek : début de la simulation, isovaleurs de contact

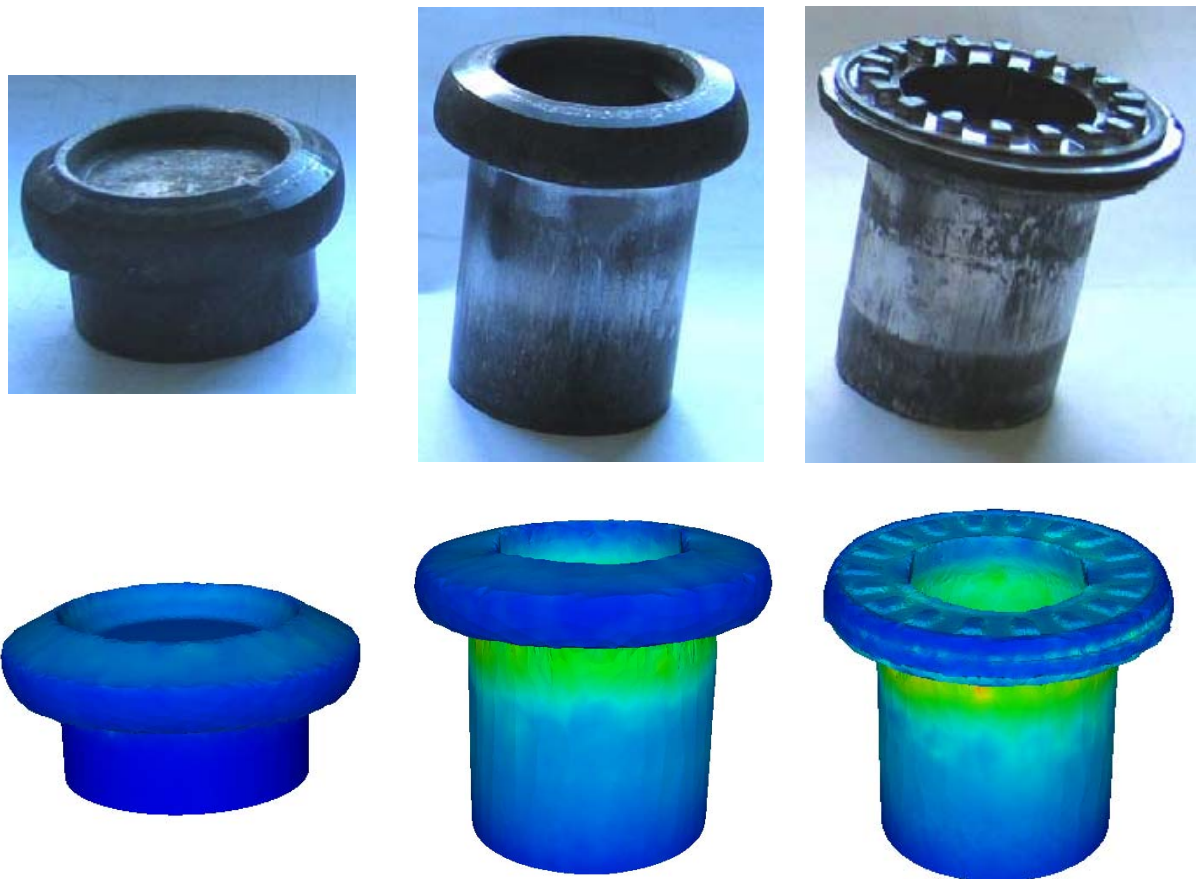


Figure V.4. Cas Smetek : images des pièces expérimentalement forgées et isovaleurs de déformation équivalente à la fin du procédé

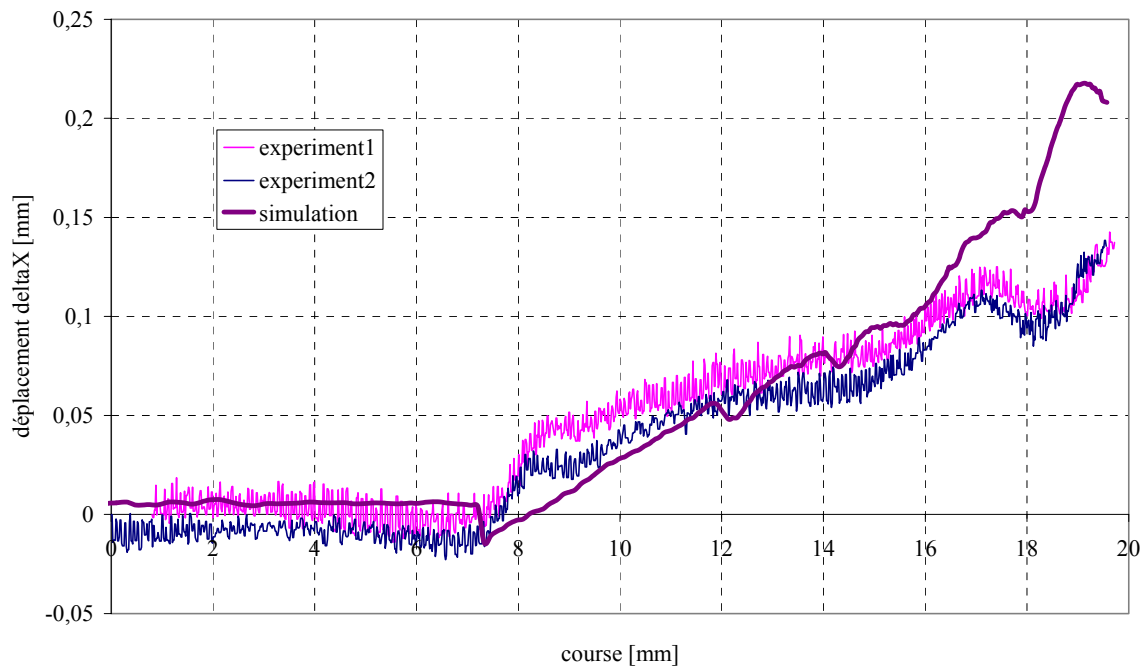


Figure V.5. Cas Smetek : valeurs expérimentales et numériques du déplacement selon l'axe x

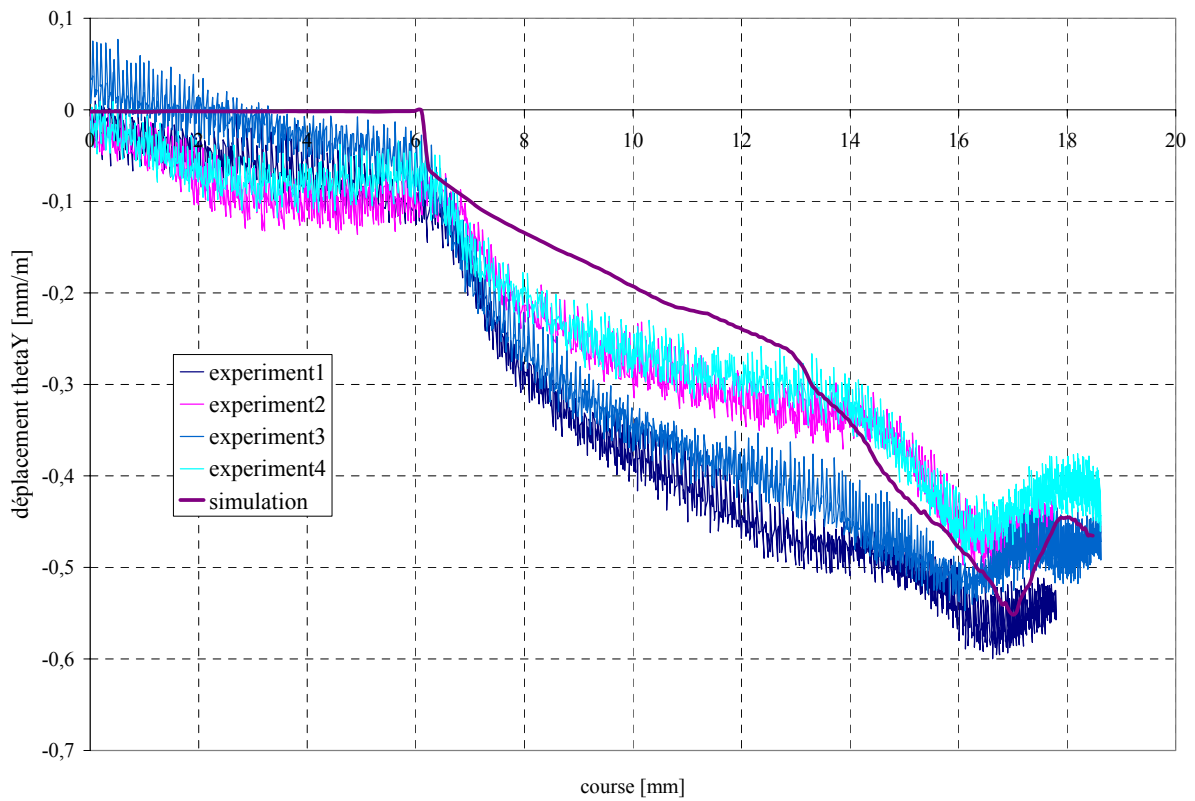


Figure V.6. Cas Smetek : valeurs expérimentales et numériques du déplacement de la presse correspondant à la rotation autour de l'axe y

V.1.3. Cas Iskra Avtoelektrika

Ce cas test a été proposé par une société slovène, Iskra Avtoelektrika, dans le cadre du projet IMPRESS. Il est relatif au forgeage d'une pièce de démarreur d'automobile. En considérant la symétrie du problème, on se contente d'effectuer le calcul sur seulement 1/7 des corps. La Figure V.7 montre la géométrie initiale et les maillages des corps sur lesquels sont effectués les calculs. Les dimensions des différents maillages au début de la simulation sont données dans le Tableau V.4. Au cours du calcul, suite aux remaillages de la pièce et de l'outil supérieur, ces valeurs sont légèrement modifiées.

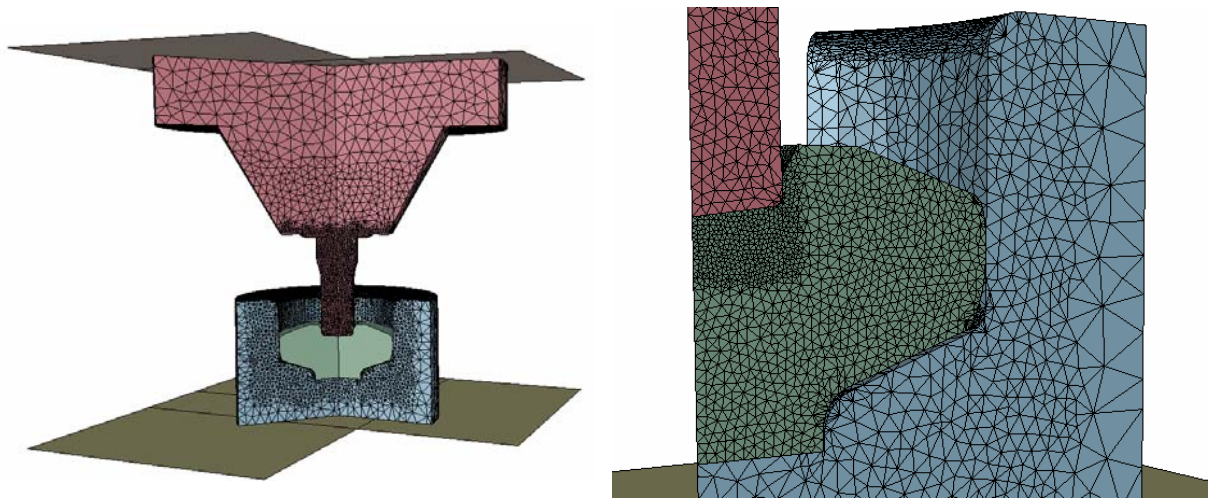


Figure V.7. Cas Iskra : géométries initiales et maillages des corps

Corps	Pièce	Outil supérieur	Outil inférieur	Total
<i>Nbnoe</i>	9 859	3 365	6 879	20 103
<i>Nbelt</i>	41 601	11 689	32 495	85 785

Tableau V.4. Cas Iskra : dimensions des maillages

La rhéologie de la pièce est modélisée par la loi élasto-visco-plastique avec l'écroissage de type Hansel et Spittel (I.26), dont les paramètres sont donnés dans le Tableau V.5. Le matériau des outils déformables suit la loi visco-plastique de Norton-Hoff (I.22) avec l'écroissage de type Hansel et Spittel (I.26), ses coefficients non-nuls sont $K_0 = 2500\text{MPa}$ et $m_3 = 0,01$

Paramètre	K_0 (MPa)	m_1	m_2	m_3	m_4	E (MPa)	γ
Valeur	862,90897	-0,00092	0,1291	0,01283	-0,00977	$2,0 \cdot 10^5$	0,3

Tableau V.5. Cas Iskra : paramètres de rhéologie

Le contact entre la pièce et les outils déformables est considéré comme unilatéral glissant et le frottement est modélisé par la loi de Coulomb (I.40) avec les coefficients $\bar{m} = 0,1$ et $\mu = 5,0 \cdot 10^{-2}$. Le contact entre les outils déformables et les outils rigides (qui servent uniquement à imposer la cinématique des matrices supérieure et inférieure) est bilatéral collant (I.32). L'outillage est piloté par une presse hydraulique de 1000t dont la vitesse de pilotage suivant l'axe z est de 250mm/s . Le modèle de presse est appliqué à l'outil inférieur avec des valeurs suivantes de la matrice de flexibilité, choisies d'une manière arbitraire, faute des données expérimentales : $k_{11} = k_{22} = 3,0 \cdot 10^9\text{ N/mm}$ et $k_{33} = 3,0 \cdot 10^6\text{ N/mm}$

La convergence de la méthode est atteinte tout au long de la simulation. La Figure V.8a montre les maillages à la fin de la simulation ainsi que les isovaleurs de contact. La Figure V.8b présente la contrainte équivalente en fin de course. La Figure V.9 montre le déplacement selon l'axe z dû aux déformations élastiques de presse.

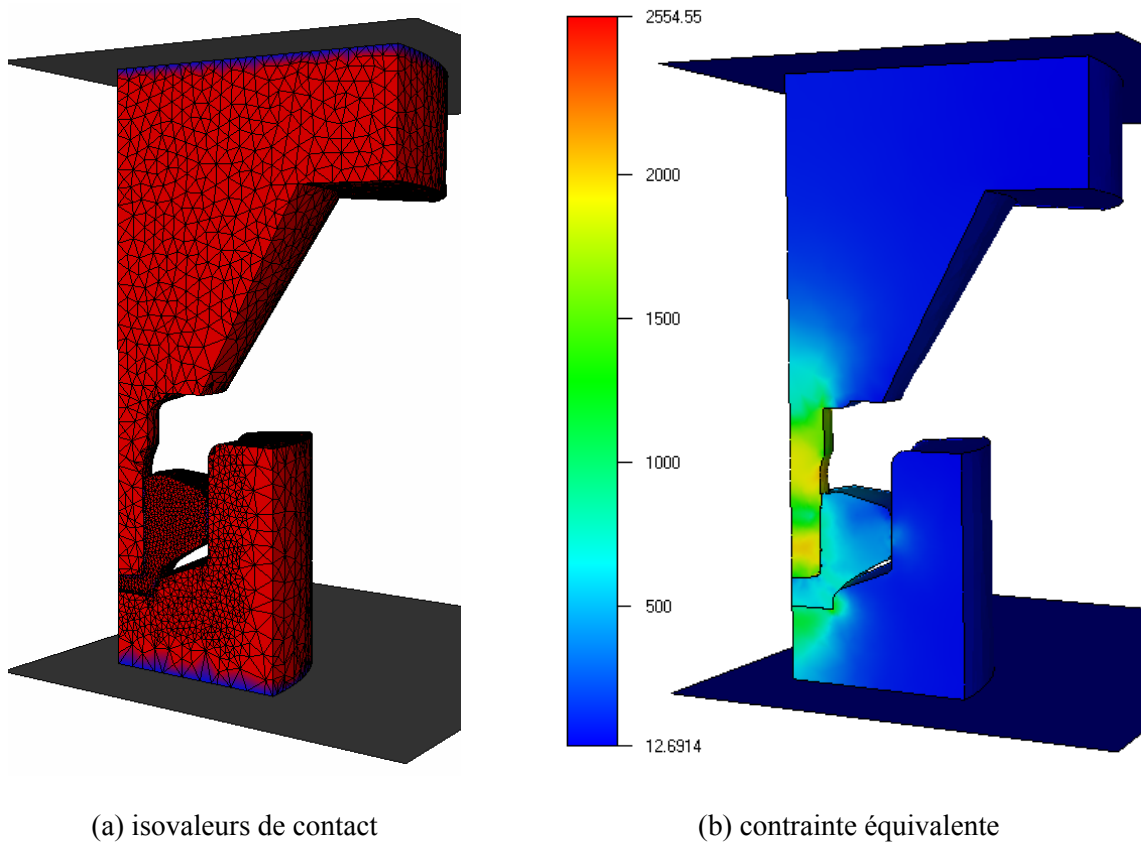


Figure V.8. Cas Iskra : fin de la simulation

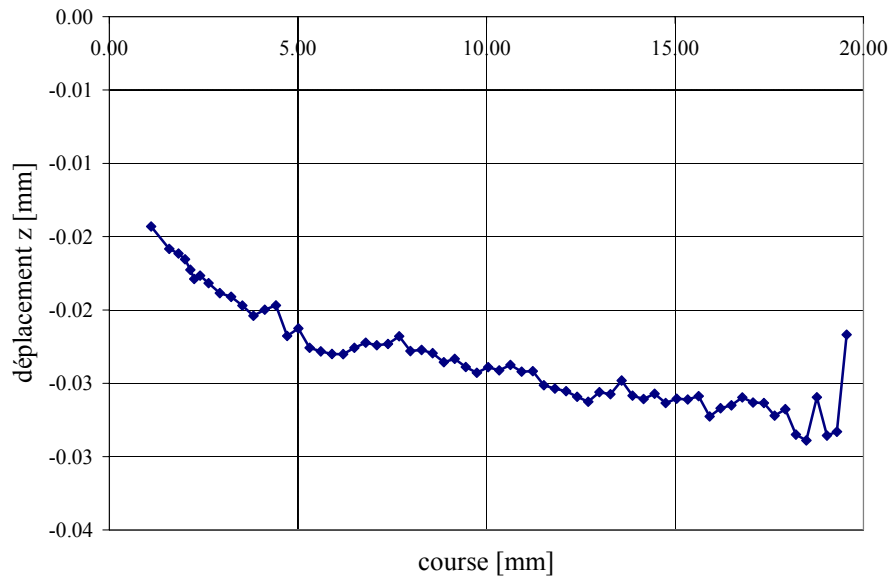


Figure V.9. Cas Iskra : déplacement selon l'axe z dû aux déformation de la presse

V.2. FORGE3 PARALLÈLE

V.2.1. Cas Vaccari

Les calculs de ce cas ont également été effectués avec la version parallèle de Forge3. Les caractéristiques essentielles d'un code parallèle sont le *speed-up* (l'accélération) $Sp(N, p)$ et l'efficacité $E(N, p)$ qui sont calculés comme :

$$Sp(N, p) = \frac{T(N, 1)}{T(N, p)} \quad (\text{V.2})$$

$$E(N, p) = \frac{Sp(N, p)}{p} \quad (\text{V.3})$$

où $T(N, p)$ est le temps de calcul du problème de taille N (nombre de degrés de liberté) sur p processeurs.

Il y a lieu de souligner ici que lors de la simulation en parallèle, le déroulement du calcul et les résultats sont identiques à ceux obtenus en séquentiel (aux erreurs d'arrondi près). En revanche, deux facteurs majeurs entrent en jeu : le remaillage, qui se fait en parallèle et le repartitionnement après le remaillage, qui se fait en séquentiel. Souvent et surtout dans les cas de grande taille et de géométrie complexe, le remaillage est déclenché plus fréquemment en parallèle qu'en séquentiel, ce qui conduit à une augmentation du temps de simulation. Cette problématique du remaillage et du

repartitionnement parallèles ayant fait l'objet d'une thèse [Digonnet 2001], et notre centre d'intérêt étant plutôt tourné vers les méthodes itératives de la résolution du système linéaire, nous nous sommes limités à l'étude des vingt premiers incréments de la simulation pendant lesquels aucun remaillage n'est survenu.

La partition de l'ébauche en 2, 4 et 8 sous-domaines est présentée sur la Figure V.10. Chacune de ces partitions est équilibrée par rapport au nombre de nœuds et constitue approximativement de problèmes de 9 100, 4 500 et 2 200 nœuds, respectivement.

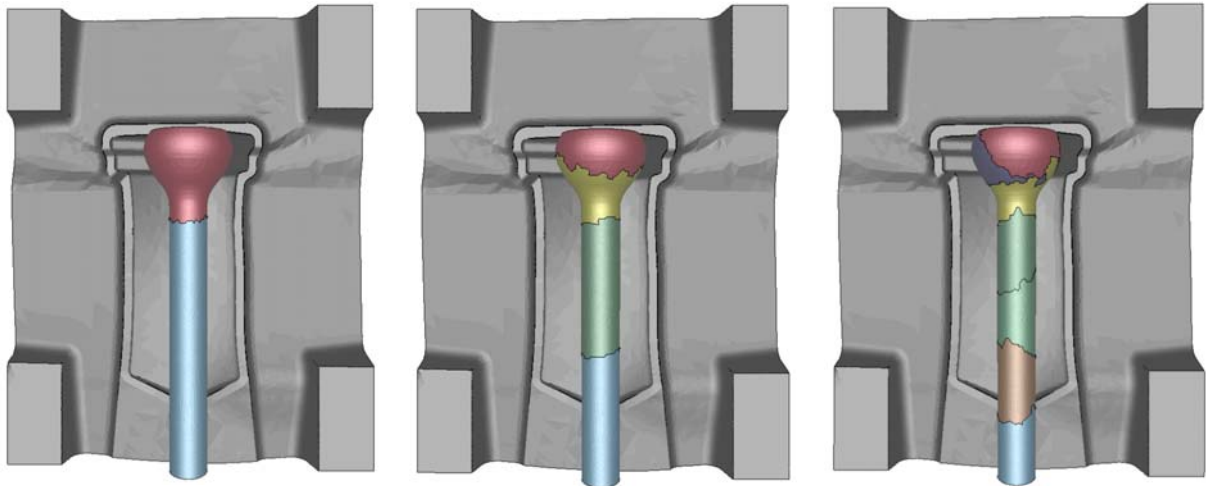


Figure V.10. Cas Vaccari : partition en 2, 4 et 8 sous-domaines

Les accélérations obtenues sur le cluster de 32 bi-processeurs Pentium-III (1GHz, 512Mo) sont données dans le Tableau V.6. Les résultats sur 2 et 8 processeurs sont surprenants au premier regard mais conformes aux performances de Forge3 avec le préconditionneur de Choleski incomplet [Marie 1997, Perchat 2000]. En effet, ce préconditionneur est très performant mais se parallélise assez difficilement, ce qui se manifeste par une efficacité parallèle à peine supérieure à 60%. D'autres préconditionneurs par la diagonale se parallélisent mieux, leurs efficacités parallèles étant proche de 90%. Néanmoins, même avec une efficacité de seulement 60%, le préconditionneur de Chloeski incomplet reste plus rapide que les préconditionneurs par la diagonale. Par ailleurs, l'efficacité de 101,5% sur 4 processeurs s'explique par un effet de mémoire cache et a déjà été rapporté, par exemple, dans [Barboza 2004].

p	$T(N, p)$ (s)	$Sp(N, p)$	$E(N, p)$ (%)
1	14h 26Mn 39s = 51999	-	-
2	11h 30Mn 55s = 41635	1,25	62,4
4	3h 33Mn 25s = 12805	4,06	101,5
8	2h 59Mn 34s = 10774	4,83	60,3

Tableau V.6. Cas Vaccari : accélérations obtenues sur le cluster de PC

V.2.2. Cas Iskra Avtoelektrika

Pour le cas mono-corps de forgeage présenté dans le paragraphe V.1.3, où les calculs sont effectués seulement dans la pièce et où les outils sont considérés comme parfaitement rigides, nous avons réalisé plusieurs simulations (de vingt incréments sans remaillage) avec différents maillages de quatre, dix et vingt-huit mille nœuds. Nous avons obtenu les accélérations suivantes en fonction du nombre de processeurs :

Nombre de nœuds	p	$Sp(N, p)$	$E(N, p)$
~4 000	2	1,36	68%
	4	2,47	62%
	6	3,66	61%
~10 000	2	1,74	87%
	4	2,96	74%
	6	2,60	43%
~28 000	2	1,92	96%
	4	2,86	72%
	6	4,22	70%

Tableau V.7. Cas Iskra mono-corps : accélérations obtenues sur le cluster

Les courbes tracées à partir de ces données sont présentées sur la Figure V.11 et la Figure V.12. En ce qui concerne l'accélération, les résultats sont conformes aux observations habituelles, sauf pour le cas à dix mille nœuds sur 6 processeurs. Cette chute de performance est probablement due à la mauvaise qualité de la partition. Toutefois, la meilleure performance (meilleure également que celles obtenues dans le cas Vaccari) est obtenue avec le plus grand maillage, de 28 000 nœuds. En speed-up, comme en efficacité, les courbes sont bien linéaires. Le cas à quatre mille nœuds est trop petit pour bénéficier pleinement des avantages du calcul parallèle. Néanmoins, les résultats sont tout à fait en accord avec les accélérations habituellement obtenus avec Forge3 parallèle.

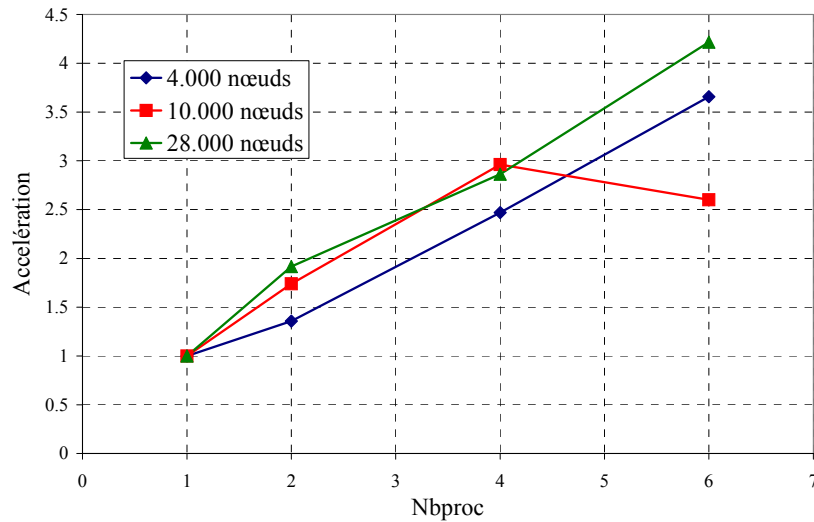


Figure V.11. Cas Iskra : accélérations obtenues sur le cluster en fonction du nombre de processeurs et pour différentes tailles de maillage

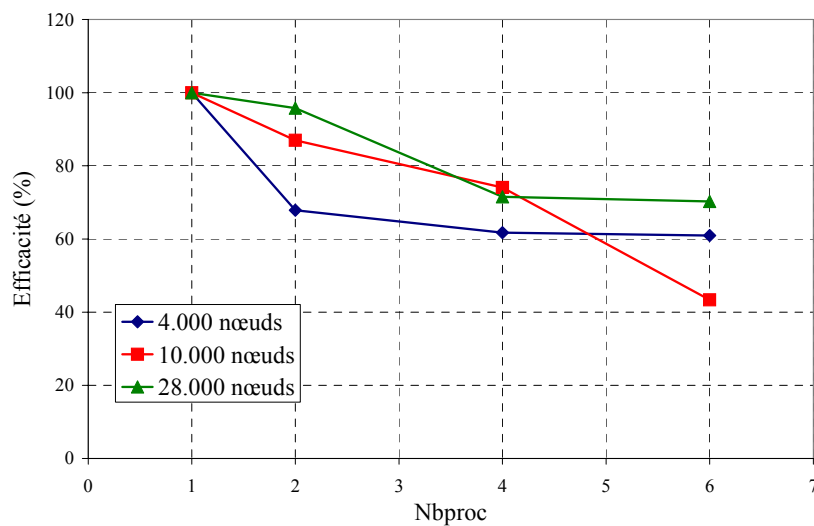
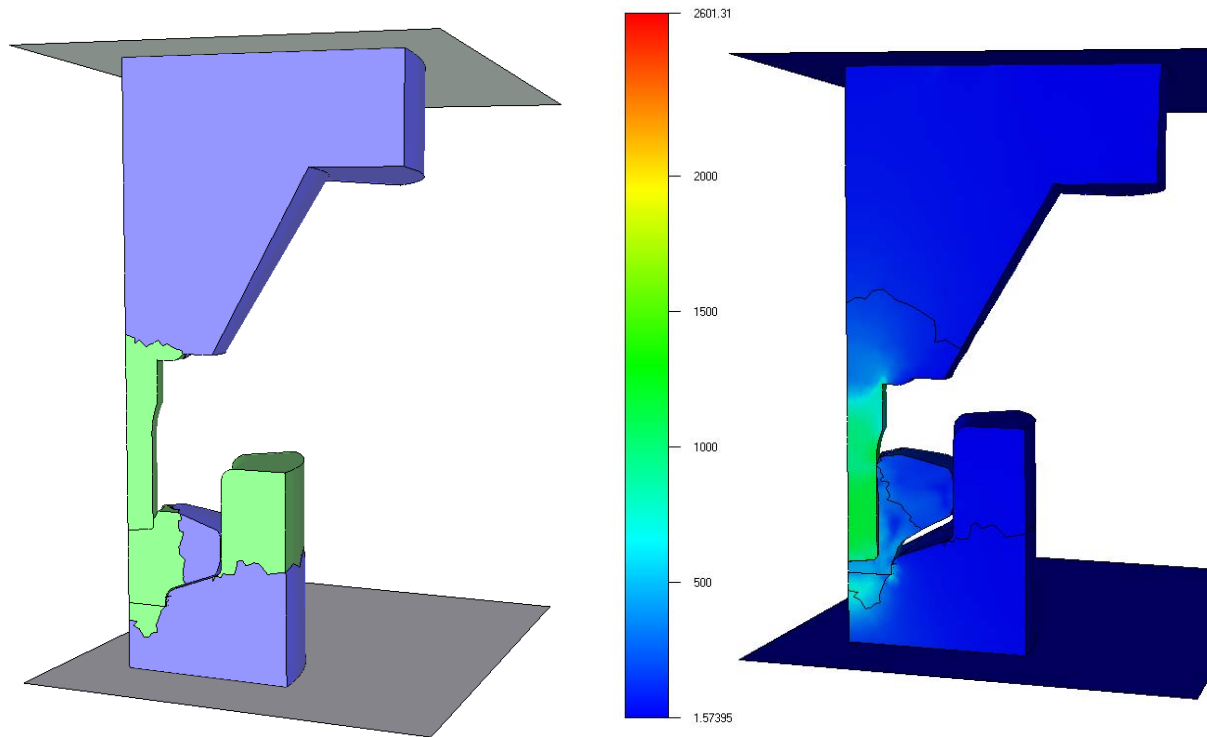


Figure V.12. Cas Iskra : efficacité pour les trois maillages en fonction du nombre de processeurs

La simulation de ce cas de forgeage avec des calculs couplés dans les outils a été également effectuée en parallèle sur deux processeurs. Les dimensions du problème sont données dans le Tableau V.8. La Figure V.13 présente la partition de trois corps considérés au début de la simulation et la contrainte équivalente à la fin de la simulation. La convergence a été atteinte tout au long de la simulation.

	Total	Processeur 1	Processeur 2
<i>Nbnoe</i>	20 103	10900	10113
<i>Nbelt</i>	85 785	42894	42891

Tableau V.8. Cas Iskra parallèle : dimensions des maillages



(a) partition des corps sur deux processeurs

(b) contrainte équivalente à la fin de course

Figure V.13. Cas Iskra : calcul sur deux processeurs

V.3. CAS DU TRIAXE : CONFRONTATION FORGE3 VS. FETI

Afin de confronter le solveur FETI à celui de Forge3, nous retenons le problème du triaxe, présenté dans le chapitre précédant IV.2. Cet exemple est caractéristique du forgeage 3D avec un matériau non linéaire, le contact traité par pénalisation, la formation d'une bavure et l'apparition de replis de matière. Les calculs sont effectués sur des maillages de différentes tailles et pour des instants de forgeage différents (début du calcul, comme fin de la simulation) afin de tester les capacités de la méthode FETI à gérer les différents aspects associés aux problèmes résolus par Forge3 (matériau non linéaire, frottement, formulation mixte en vitesse/pression, et surtout le contact pénalisé). FETI s'est alors montré suffisamment robuste pour résoudre l'ensemble de ces systèmes issus de Forge3. En outre, pour une précision du calcul imposée, la méthode FETI converge généralement avec un résidu plus petit que celui de Forge3, et donc avec une solution plus précise. Ceci étant, la principale question qui nous a occupée est la suivante : est-ce que FETI est plus rapide que le solveur actuel de Forge3 ?

Une série de tests a été réalisée sur un cluster de calcul constitué de 32 bi-processeurs Pentium-IV (2,8GHz, 3Go) avec le réseau myrinet 2000.

V.3.1. Cas à 10 602 nœuds

On a d'abord considéré un maillage à 10 602 nœuds. La taille du problème global à résoudre est donc de 42 408 degrés de liberté. Le critère de convergence choisi est de 10^{-4} , ce qui correspond à la précision de calcul utilisée habituellement dans Forge3.

Les systèmes linéaires issus du deuxième incrément de la simulation ont été résolus par les trois solveurs. Celui de Forge3, présenté dans le chapitre II, utilise la méthode du résidu minimal préconditionnée par Choleski incomplet. Le deuxième solveur testé est ce même solveur mais en provenance de la librairie PETSc [PETSc]. Enfin, la méthode FETI a été utilisée avec le préconditionneur allégé (III.44). Le Tableau B.1 de l'Annexe B montre les résidus obtenus avec ces trois solveurs, ainsi que les temps de résolution correspondants.

La Figure V.14 montre l'évolution du temps de calcul en fonction du nombre de processeurs. On constate que le comportement de PETSc et de Forge3 se ressemblent, ces deux solveurs utilisant le même algorithme de résolution à quelques détails d'implémentation près. Néanmoins, le solveur PETSc est manifestement plus efficace grâce à ses optimisations et son utilisation des librairies mathématiques performantes (produit matrice-vecteur, produit scalaire). Cependant, notre but étant de comparer des méthodes de résolution et non des algorithmes de produits algébriques, nous nous attachons à comparer FETI à Forge3, tout en nous réservant la possibilité, ultérieurement, d'utiliser des librairies mathématiques performantes pour la méthode la plus efficace.

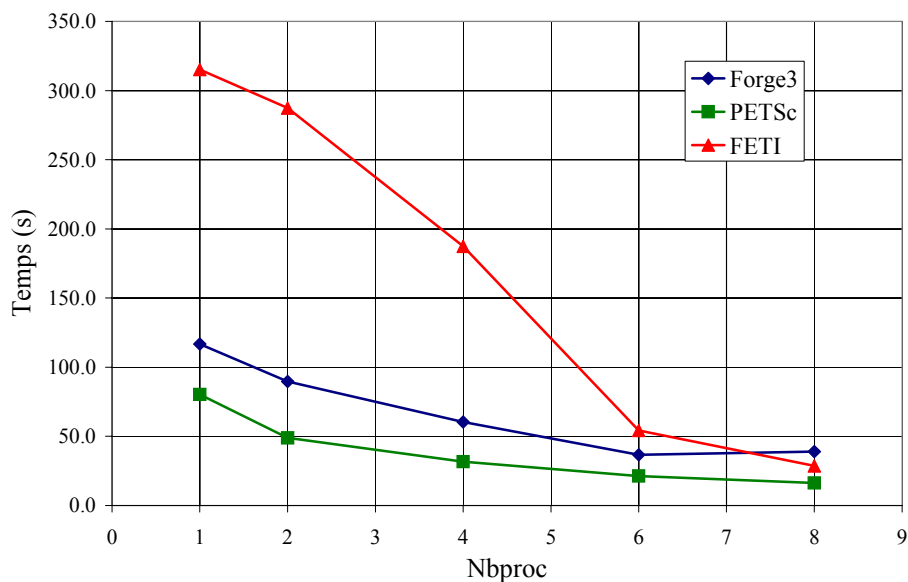


Figure V.14. Cas du triaxe à 10 602 nœuds : temps de résolution en fonction du nombre de processeurs

Par ailleurs, il faut souligner que le temps de résolution par la méthode FETI comprend non seulement le temps de la résolution itérative, mais aussi celui de la factorisation des matrices locales. Ainsi, le temps de 315,20s pour 1 processeur est en réalité le temps de factorisation de la matrice globale, puisque dans ce cas FETI est une méthode de résolution directe. Cette étape de factorisation

peut être très pénalisante en temps de calcul. En effet, la différence du nombre d'itérations de FETI (moins d'une centaine) et des deux autres solveurs (jusqu'à un millier) est frappante (cf. Figure V.15a). Certes, les itérations de FETI sont plus coûteuses que celles des deux autres solveurs (cf. Figure V.15b), mais le problème de fond réside dans le fait que la plus grande partie de temps de calcul (et ce jusqu'à 6 processeurs) est consacrée à factoriser les matrices locales. Il est alors clair que le solveur FETI atteint sa meilleure efficacité quand le temps de factorisation est égal à celui de la résolution itérative. Ainsi, sur 6 processeurs, la taille des sous-problèmes est approximativement de 2 000 nœuds, cet écart est assez faible (cf. Figure V.16) et le temps total de résolution du problème est comparable à celui de Forge3. Sur 8 processeurs, les problèmes locaux ayant environ 1 500 nœuds, il n'y a plus de cet écart : FETI est plus efficace que Forge3.

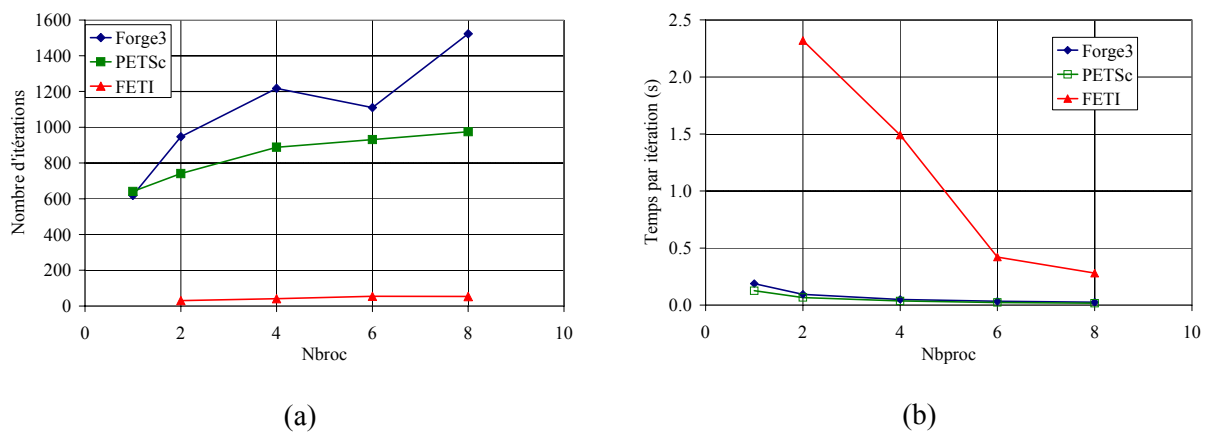


Figure V.15. Cas du triaxe à 10 602 nœuds : nombre d'itérations (a) et temps nécessaire pour une itération du solveur considéré (b)

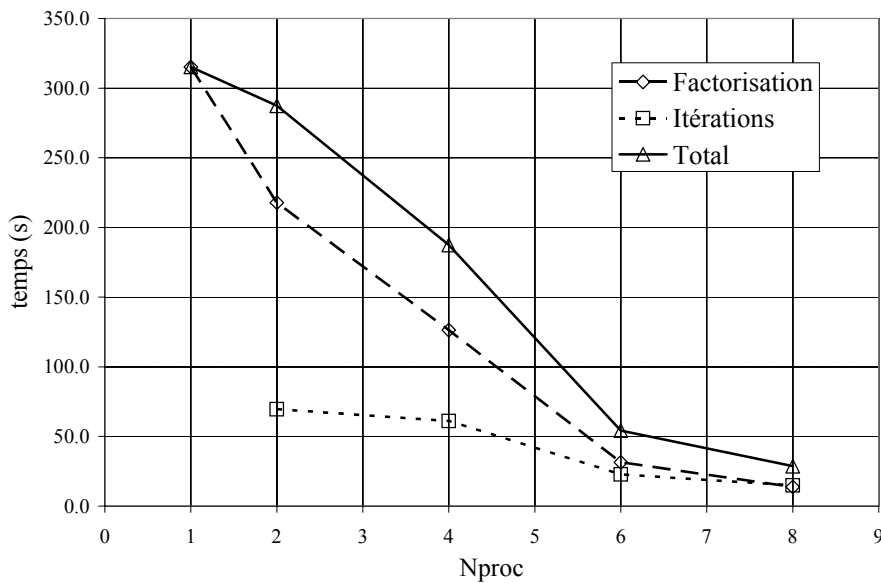
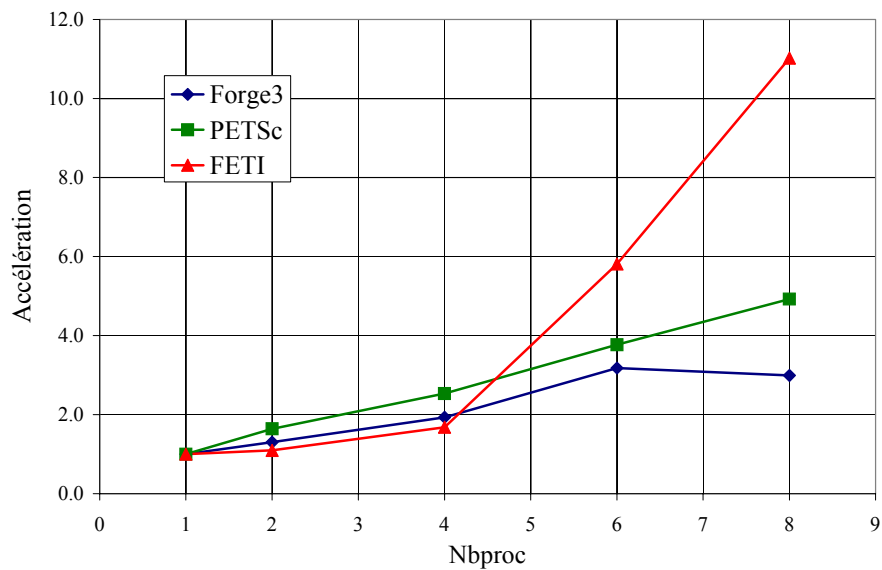
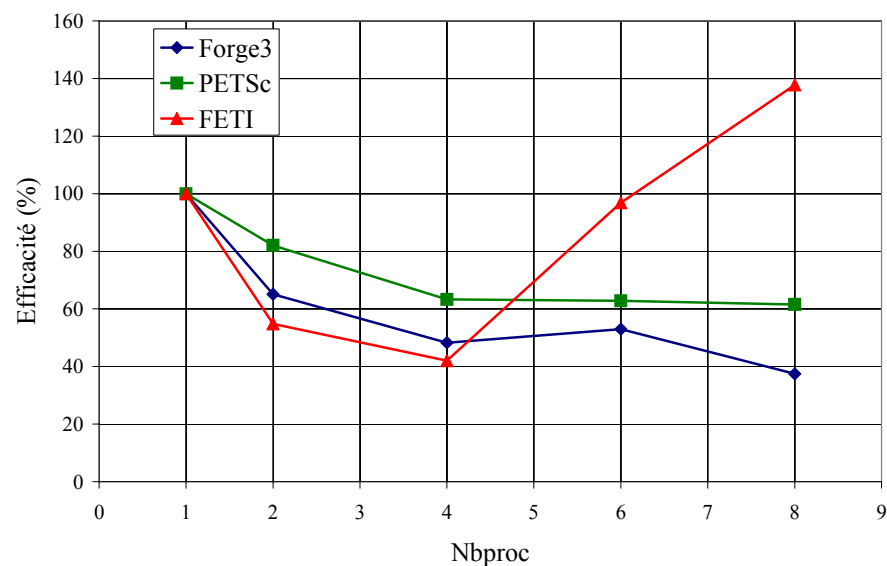


Figure V.16. Méthode FETI : temps de factorisation, de résolution itérative et temps total

La Figure V.17 montre l'accélération (a) et l'efficacité (b) des trois solveurs en fonction du nombre de processeurs (les chiffres exacts sont donnés dans les Tableaux B.2 et B.3 de l'Annexe B). Tout d'abord, on constate que le comportement du solveur PETSc est parfaitement linéaire en accélération et stable en efficacité, qui varie entre 82% sur 2 processeurs et 62% sur 8. Les performances de Forge3 chutent à 38% sur 8 processeurs, et sont moins bonnes que celles obtenues dans d'autres simulations présentées dans le paragraphe V.2. En ce qui concerne la méthode FETI, à partir de 4 processeurs elle suit un comportement non-linéaire pour atteindre, sur 8 processeurs, une accélération de 11 avec 138% d'efficacité. Ce caractère non-linéaire du solveur FETI est dû à sa nature mixte, celle d'un solveur direct au niveau local et d'un solveur itératif au niveau de l'interface.



(a)



(b)

Figure V.17. Cas du triaxe à 10 602 nœuds : accélération (a) et efficacité (b)

D'autre part, il n'est peut être pas tout à fait justifié d'évoquer la méthode FETI sur 1 processeur où elle se réduit à une méthode directe. Nous présentons donc de nouveau ces résultats en utilisant le

temps de résolution sur 2 processeurs comme référence. Les résultats sont donnés dans le Tableau B.4 et le Tableau B.5 de l'Annexe B et les courbes sont tracées sur la Figure V.18. Tout d'abord, on constate une efficacité de 81% pour Forge3 sur 6 processeurs et de 57% sur 8 processeurs. Ces valeurs sont davantage représentatives du niveau de parallélisme de Forge3, puisque la comparaison au calcul sur 2 processeurs élimine l'effet de la chute de performances dû au changement de type de préconditionneur de Choleski incomplet entre le séquentiel et le parallèle. Par ailleurs, le caractère non linéaire de FETI est toujours évident et même accentué. À partir de 4 processeurs, on constate un effet de sur-accélération produit par le raffinement de la partition.

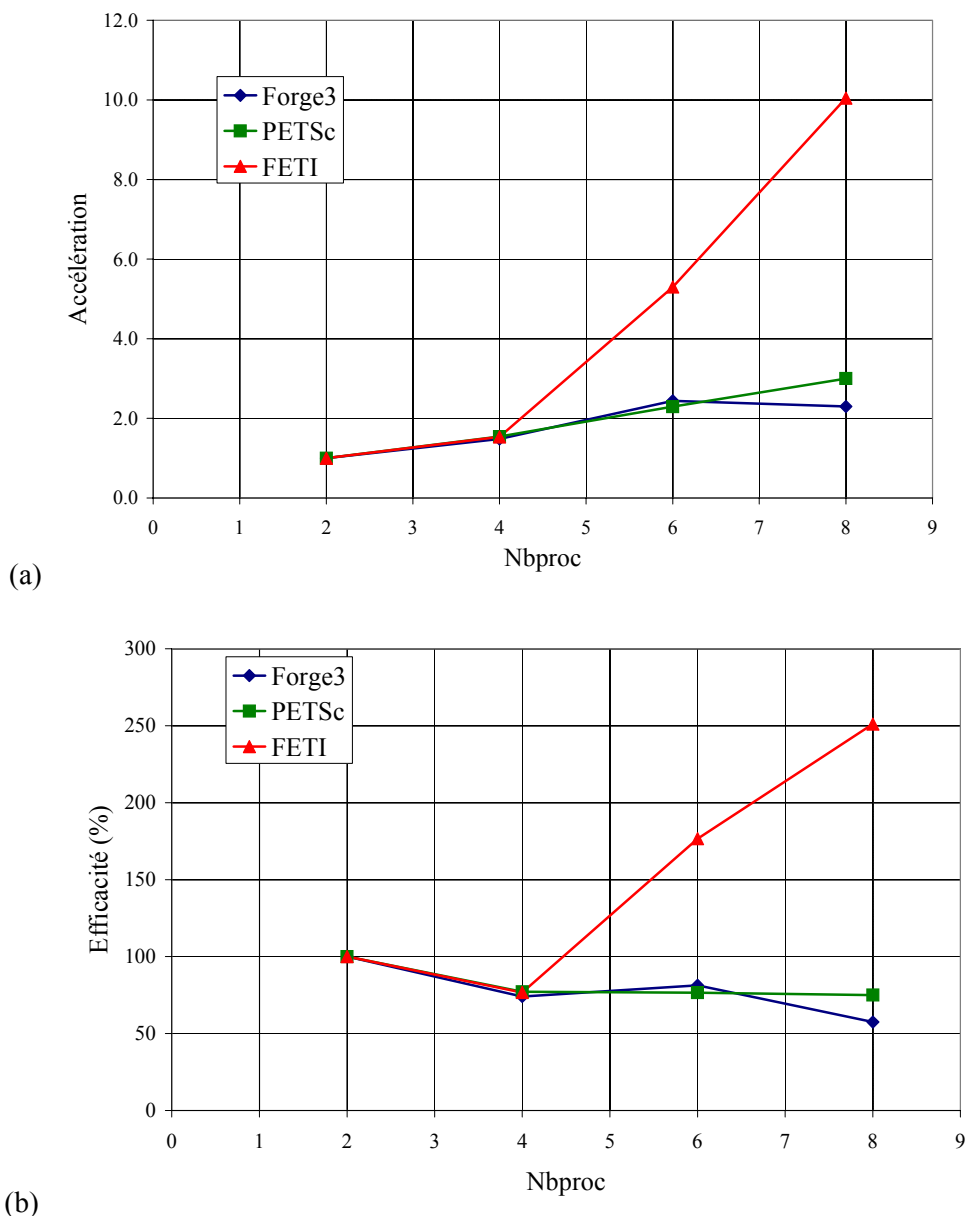


Figure V.18. Cas du triaxe à 10 602 nœuds : accélération (a) et efficacité (b) calculées avec comme référence le temps de calcul sur 2 processeurs

Vue que la méthode FETI converge avec une solution plus précise que Forge3 pour le critère donné, nous avons étudié une diminution de la précision de ce critère d'arrêt. Le Tableau V.9 présente les résultats obtenus pour des calculs Forge3 avec le critère de 10^{-4} et ceux de FETI avec trois valeurs décroissantes. Le temps de factorisation reste le même quelque soit ce critère, alors que le temps de résolution itérative (et donc le temps total) ne diminue pas d'une manière significative. En outre, le critère de 10^{-2} s'avère trop faible pour que la solution fournie par FETI soit correcte. Finalement, cela ne permet pas d'accélérer significativement la convergence. Néanmoins, FETI se présente comme une méthode très robuste, capable de trouver une solution au problème avec beaucoup de précision et en peu d'itérations.

<i>Solveur</i>	<i>Résidu final</i>	<i>Nombre d'itérations</i>	<i>Temps total (s)</i>	<i>Temps par itération (s)</i>
Calcul sur 6 processeurs				
Forge3	9,08E-05	1110	36,71	0,033
FETI 10^{-4}	9,29E-07	54	54,24	0,421
FETI 10^{-3}	8,53E-06	46	53,55	0,440
FETI 10^{-2}	7,28E-05	37	50,57	0,471
Calcul sur 8 processeurs				
Forge3	8,82E-05	1523	38,98	0,026
FETI 10^{-4}	9,56E-07	53	29,51	0,280
FETI 10^{-3}	9,33E-06	43	27,27	0,294
FETI 10^{-2}	1,10E-04	31	24,45	0,317

Tableau V.9. Méthode FETI : diminution de précision

Pour la partition en 8 sous-domaines et le critère de convergence de 10^{-4} , nous avons aussi étudié les calculs sans préconditionnement et avec le préconditionneur de Dirichlet. Les résultats en temps de calcul et nombre d'itérations sont présentés dans le Tableau V.10. Pour le cas sans préconditionnement, le temps d'une itération est assez faible, mais loin de celui de Forge3, et le nombre d'itérations est trop grand pour rivaliser avec Forge3. En ce qui concerne le préconditionneur de Dirichlet, on constate ici les mêmes tendances que celles observées dans le chapitre précédant. Même si le nombre d'itérations diminue significativement, le temps nécessaire pour la résolution itérative est seulement légèrement plus faible par rapport au cas du préconditionneur allégé. Non seulement le temps nécessaire pour effectuer une itération a doublé, mais aussi le temps de factorisation a augmenté, ce qui conduit à ce que le calcul avec le préconditionneur allégé soit plus

rapide. De ce point de vue, il n'y a pas nécessité d'utiliser un préconditionneur plus élaboré et donc le préconditionneur allégé est plus avantageux.

<i>Solveur</i>	<i>Nombre d'itérations</i>	<i>Temps de factorisation (s)</i>	<i>Temps pour les itérations (s)</i>	<i>Temps total (s)</i>	<i>Temps par itération (s)</i>
Forge3	1523			38,984	0,026
FETI – non-prec.	492	14,531	124,537	139,068	0,283
FETI – allégé	43	14,629	12,640	27,269	0,634
FETI – Dirichlet	28	22,979	11,665	34,644	1,237

Tableau V.10. Méthode FETI sur 8 processeurs avec différents préconditionneurs

V.3.2. Cas à 58 626 nœuds

Un autre maillage considéré est celui de 58 626 nœuds, ce qui offre un problème à résoudre d'assez grande taille, avec 234 504 degrés de liberté. Le critère de convergence est toujours de 10^{-4} pour les trois solveurs. Les préconditionneurs sont inchangés.

Le calcul sur un processeur n'a pas été effectué en raison du manque de ressources en mémoire. Par ailleurs, la mémoire disponibles sur chaque processeur du cluster n'a pas été suffisante pour mener les factorisations de la méthode FETI ni sur 2 processeurs (30 000 nœuds par processeur), ni sur 4 (environ 15 000 nœuds par processeur).

La Figure V.19 montre le temps de résolution du système linéaire pour les trois solveurs, ainsi que les temps de factorisation et de résolution itérative pour la méthode FETI.

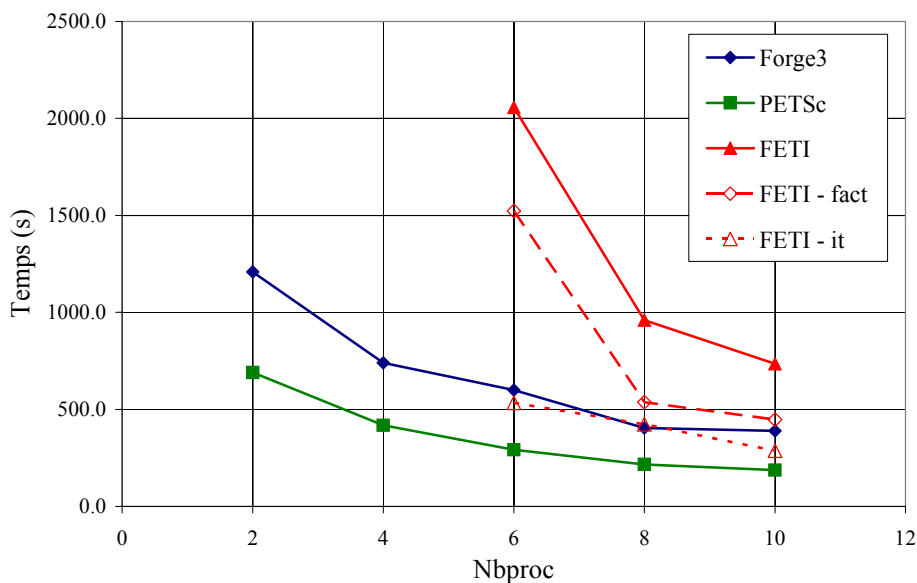
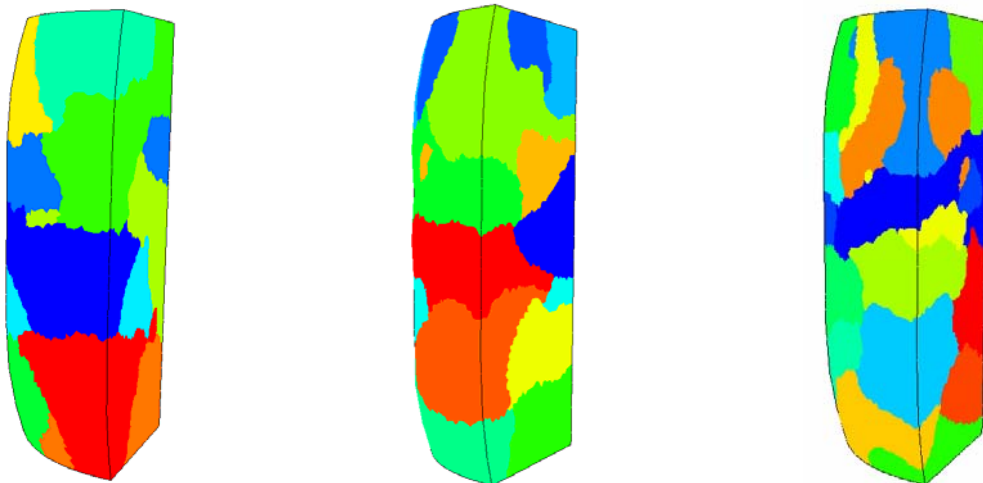


Figure V.19. Cas du triaxe à 58 628 nœuds : temps de résolution

Comme l'on a vu dans le cas précédent à 10 602 nœuds, pour obtenir les meilleures performances de FETI, il faut que le temps incompressible de factorisation soit équivalent au temps de résolution itérative. Pour cela il faut que les sous-domaines soient suffisamment petits. Par conséquent, dans le cas présent, le nombre des sous-domaines devrait être plus élevé. En se basant sur la partition en 8 sous-domaines du cas à 10 602 nœuds, on peut estimer qu'il faudrait découper le domaine à 58 628 nœuds en une cinquantaine parties pour obtenir des sous-domaines à environ 1 500 nœuds, cette taille de 6 000 degrés de liberté étant la limite d'efficacité du solveur directe vis-à-vis du solveur itératif. Cette relation ne prend probablement pas en compte le caractère non-linéaire de l'accélération de la méthode FETI. En effet, sur la Figure V.19 on voit déjà qu'entre 8 et 10 processeurs, l'écart entre ces deux temps a considérablement diminué par rapport au calcul sur 6 processeurs. Forcément, il serait très intéressant de mener les calculs sur un plus grand nombre de processeurs. Malheureusement, il s'est avéré impossible de le faire à cause des partitionneurs utilisés. Avec le partitionneur standard de Forge3 nous n'avons pu construire des partitions satisfaisantes que jusqu'à 8 processeurs. Avec le partitionneur dynamique [Digonnet 2001] nous avons pu améliorer considérablement les partitions existantes et générer la partition en 10 sous-domaines. En revanche, les partitions en 12 et 16 sous-domaines (cf. Figure V.20) que nous avons construites se sont avérées de trop mauvaise qualité (partitions pas assez compactes) pour pouvoir être utilisées par Forge3.



(a) 10 sous-domaines

(b) 12 sous-domaines

(c) 16 sous-domaines

Figure V.20. Cas du triaxe à 58 628 nœuds : différentes partitions construites mais impropres aux calculs au-delà de 10 sous-domaines

Même si les calculs sur 1 processeur n'ont pas pu être réalisés, on présente les accélérations pour Forge3 et PETSc dans le Tableau V.11 en prenant comme référence le temps sur 2 processeurs. On observe de nouveau une efficacité de Forge3 entre 80 et 60%, comme dans les cas précédents. Par ailleurs, les performances de PETSc sont toujours meilleures de Forge3 pour les raisons indiquées plus haut. Pour compléter cette étude sur FETI, nous présentons dans le Tableau V.12 les accélérations pour les trois solveurs avec comme référence le temps sur 6 processeurs. Pour Forge3 et PETSc, le

rapport entre les calculs sur 6 et 8 processeurs est très bon et laisse même évoquer l'effet de sur-accélération, tandis que l'efficacité sur 10 processeurs est supérieure à 90%. Quant à FETI, son caractère non-linéaire est toujours évident avec l'efficacité supérieure à 160% sur 8 et 10 processeurs.

<i>Partition</i>	<i>Accélération</i>		<i>Efficacité (%)</i>	
	Forge3	PETSc	Forge3	PETSc
2 (1)	1	1	100	100
4 (2)	1,63	1,65	81,7	82,49
6 (3)	2,02	2,36	67,2	78,77
8 (4)	2,99	3,20	74,85	79,91
10 (5)	3,11	3,69	62,1	73,84

Tableau V.11. Forge3 et PETSc : accélérations obtenues sur le cluster par référence au calcul sur 2 processeurs

<i>Partition</i>	<i>Accélération</i>			<i>Efficacité (%)</i>		
	Forge3	PETSc	FETI	Forge3	PETSc	FETI
6 (1)	1	1	1	100	100	100
8 (1,3)	1,49	1,35	2,14	111,39	101,45	160,58
10 (1,7)	1,54	1,56	2,80	92,42	93,74	167,97

Tableau V.12. Cas du triaxe à 58 628 nœuds : accélérations obtenues sur le cluster par référence au calcul sur 6 processeurs

V.4. CONCLUSION

Les résultats présentés dans ce chapitre nous ont permis de valider le modèle de raideur des presses de forgeage présenté dans le chapitre I. D'une part, une comparaison avec des données expérimentales a montré un excellent accord entre les résultats numériques et expérimentaux. D'autre part, l'utilisation du modèle de presse dans des cas industriels ne perturbe pas la robustesse des simulations, tant en séquentiel qu'en parallèle.

Les tests effectués sur les trois solveurs parallèles : Forge3, PETSc et FETI, bien que pas très favorables à FETI, ont permis de voir quelques tendances. Les meilleures performances de PETSc vis-à-vis de deux autres solveurs sont manifestes, ce qui nous incite à utiliser des bibliothèques informatiques efficaces une fois la méthode de résolution mise à point. Si nous voulons utiliser FETI d'une manière efficace, la taille des sous-domaines doit être assez faible pour que le coût de factorisation soit comparable à celui d'une résolution itérative du problème condensé aux interfaces. Nous avons été

alors confronté à des difficultés de partitionnement, de sorte que nous n'avons pu rencontré qu'une seule situation (cas du triaxe à 10 602 nœuds sur 8 processeurs) où cette condition a été réalisée. Dans ce cas, FETI s'est avéré plus efficace que Forge3.

Nous pouvons également conclure que les problèmes de forgeage ne sont pas assez raides pour utiliser le potentiel de FETI. Il serait donc intéressant d'étudier cette méthode sur d'autres problèmes de mise en forme, plus raides, par exemple en fonderie, pour lesquels le solveur de Forge3 rencontre des difficultés à converger. D'autre part, vu le nombre très faible d'itérations nécessaire à FETI, son utilisation est très avantageuse sur des grilles de calcul où une partie considérable du temps de calcul est consacrée aux communications. FETI permet alors de réduire très significativement l'échange d'informations entre les sous-domaines.

CONCLUSIONS ET PERSPECTIVES

La première partie de ce travail a consisté à développer un modèle de raideur des presses de forgeage. Les déformations de la structure de la presse sont le résultat combiné du système pièce-outillage-presse qui évolue pendant le procédé de forgeage puisque la géométrie de la pièce évolue aussi. Pour calculer les réactions sur les outils et sur la presse, nous avons choisi de coupler fortement les calculs dans la pièce et dans l'outillage. Le modèle proposé se présente comme une généralisation des outils flottants pilotés à forces et à couples imposés où ces forces et couples suivent une loi de type élastique. En premier temps, l'implémentation du modèle a été validée par une comparaison des résultats de calcul avec une solution analytique connue pour un cas d'écrasement d'un cube. Ensuite, une étude a été menée sur quelques applications industrielles. La prise en compte de la raideur de presse dans ces applications s'est avérée justifiée et efficace. En effet, si la presse n'est pas assez rigide, l'écoulement de matière peut être très perturbé par rapport à la forme finale voulue. Par ailleurs, la comparaison des résultats de simulation avec les données expérimentales a montré une excellente concordance, validant ainsi l'approche du projet IMPRESS pour la raideur des presses de forgeage, et son implémentation dans Forge3[®]. En outre, la prise en compte de ce modèle lors des simulations des applications industrielles ne perturbe pas la convergence de la méthode, ni l'efficacité du logiciel en parallèle.

En deuxième partie de notre travail, nous nous sommes intéressés au calcul parallèle, et plus précisément aux méthodes de décomposition de domaine. La question que nous nous sommes posée a été : est-ce que l'on peut améliorer les performances parallèles du solveur Forge3[®] en utilisant ces méthodes réputées pour leurs efficacité ?

À partir du code FETI, nous avons développé la méthode primale de décomposition de domaine, la méthode du complément de Schur, ainsi qu'une nouvelle approche hybride. Cette nouvelle méthode se présente comme une généralisation des deux méthodes classiques, en étant mieux adaptée à la formulation mixte en vitesse/pression caractéristique de Forge3[®]. Les aspects du préconditionnement et des modes de corps rigides ont été également abordés. L'étude des performances de ces trois méthodes a montré leur convergence pour des problèmes issus de Forge3[®], à savoir, une formulation mixte en vitesse/pression, un matériau non-linéaire, un contact pénalisé, la formation de bavures et de

replis de matière. Dans les exemples étudiés, la nouvelle méthode hybride dotée d'un préconditionneur allégé s'est montrée plus efficace que les deux autres méthodes. En outre, les résultats montrent que les préconditionneurs simples sont efficaces, voire, dans le cas de la méthode FETI, plus efficace que le préconditionneur optimal de Dirichlet. D'autre part, la confrontation du solveur Forge3[®] à celui de FETI n'a pas été tout à fait favorable à FETI. Un point très positif réside dans la convergence de la méthode FETI vers une solution plus précise que celle de Forge3[®], pour un même critère d'arrêt donné. Un autre résultat très encourageant est obtenu lors du forgeage du triaxe à 10 602 nœuds, où le temps de résolution du problème par la méthode FETI est inférieur à celui de Forge3[®]. Néanmoins, nous pouvons conclure que FETI est mieux adapté aux problèmes plus raides que ceux résolus par Forge3[®]. Nous avons aussi mis en avant une condition nécessaire pour que la méthode FETI soit efficace : le temps de factorisation des matrices locales doit être équivalent au temps de résolution itérative du problème condensé aux interfaces. Une façon de remplir cette condition est de décomposer le domaine initial en un nombre assez élevé de sous-domaines. Dans ce cas nous risquons d'être confrontés à des difficultés de partitionnement, comme cela s'est produit lors du forgeage du triaxe à 58 628 nœuds.

Les résultats obtenus à l'issue de cette étude sur les méthodes de décomposition de domaine sont prometteurs. Toutefois certains points sont encore à améliorer et de nombreux axes de travaux futurs apparaissent.

En ce qui concerne la méthode primale-duale, le préconditionneur complet doit être amélioré. Le problème de contrôle du résidu est manifeste, ainsi que celui du nombre d'itérations trop élevé par rapport au préconditionneur allégé. Un traitement spécial des points multiples de partition doit également être implémenté afin de pouvoir gérer correctement les modes rigides dans des partitions arbitraires. Et certainement, il sera très intéressant d'effectuer des tests de plus grande taille sur les machines parallèles.

L'utilisation de la méthode FETI dans les cas de grande taille a soulevé un problème de partitionnement. En effet, ces problèmes doivent être découpés en un grand nombre de sous-domaines pour que l'équilibre entre la taille des sous-domaines et la taille de l'interface globale soit atteint, et que la méthode devienne efficace. Or, dans le cas de forgeage du triaxe à 58 626 nœuds nous n'avons pas réussi à générer des partitions sur plus de dix processeurs de sorte qu'elles soient gérables lors des calculs.

Dans les tests effectués avec FETI, nous n'avons pas pu obtenir des gains considérables en temps de résolution par rapport au solveur de Forge3[®]. Néanmoins, il sera intéressant d'utiliser la méthode FETI dans les simulations de fonderie, où les problèmes linéaires à résoudre sont plus raides qu'en forgeage et où nous pourrions nous attendre à ce que FETI soit plus efficace que le solveur actuel. Une autre considération concerne le nombre d'itérations lors de la résolution par FETI qui est significativement plus faible que celui de la méthode itérative de Forge3[®]. De ce fait, l'utilisation du

solveur FETI sur les grilles de calcul où les communications entre les processeurs représente une partie considérable du temps de calcul global, pourrait être très avantageuse.

BIBLIOGRAPHIE

- [Barboza 2004] **Josué Barboza**, Traitement du contact entre corps déformables et calcul parallèle pour la simulation 3D du forgeage multicorps, *Thèse de Doctorat de l'École des Mines de Paris, Sophia-Antipolis*, (2004)
- [Bhardwaj 2000] **M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson and D. Rixen**, Application of the FETI method to ASCI problems – scalability results on 1000 processors and discussion of highly heterogeneous problems, *Int. J. Numer. Meth. Engng.* Vol.47, 2000, pp 513-535
- [Bockel 1980] **G. Bockel**, Statische und dynamische Untersuchungen von Auffederung, kippung und Versatz an Spindelpressen, *Fakultät für Maschinenwesen der technischen Universität Hannover*, 1980
- [Boyère 1999] **Emmanuel Boyère**, Contribution à la modélisation numérique thermo-mécanique tridimensionnelle du forgeage, *Thèse de Doctorat de l'École des Mines de Paris, Sophia-Antipolis*, (1999)
- [Brezzi et Marini 1993] **F. Brezzi et L. Marini**, A three-field domain decomposition method, dans *Proceedings of the sixth international conference on domain decomposition methods*, 1993, pp 27–34
- [Chodnikiewicz et al 1994] **K. Chodnikiewicz, R. Balendra, T. Wanheim**, A new concept for the measurement of press stiffness, *J. Mater. Process. Technol.* 44 (1994) pp. 293–299

- [Chodnikiewicz et al 1996] **K. Chodnikiewicz, H. Gjorstvang, T. Wanheim, R. Balendra**, New machine test for presses, *Proceedings of the BAMFAC Seminar, Helsinki, 1996*
- [Chodnikiewicz et al 2000] **K. Chodnikiewicz, R. Balendra**, The calibration of metal-forming presses, *J. Mater. Process. Technol.* 106 (2000) pp. 28–33
- [Coupez 1995] **T. Coupez**, Stable Stabilised finite element for 3D forming calculation, *communication interne CEMEF, ENSMP, Sophia-Antipolis*, (1995)
- [Coupez et al 1996] **T. Coupez, S. Marie and R. Ducloux**, Parallel 3d simulation of forming processes including parallel remeshing and reloading, *Numer. Meth. in Engng*, 1996, pp. 738-743
- [Digonnet 2001] **Hugues Digonnet**, Répartition dynamique et maillage parallèle, *Thèse de Doctorat de l'École des Mines de Paris, Sophia-Antipolis*, (2001)
- [Doege 1990] **E. Doege**, *CIRP Annals* 39/1/90
- [Dureisseix et Farhat 2001] **D. Dureisseix and C. Farhat**, A numerically scalable domain decomposition method for the solution of frictionless contact problems, *Int. J. Numer. Meth. Engng*. Vol.50 2001, pp 2643-2666,
- [Farhat et al 2000a] **C. Farhat, K. Pierson and M. Lesoinne**, The second generation FETI methods and their application to the parallel solution of large-scale and geometrically non-linear structural analysis problems, *Comput. Methods Appl. Mech. Engng*. Vol.184 (2000) pp. 333-374
- [Farhat et al 2000b] **C. Farhat, M. Lesoinne and K. Pierson**, A scalable dual-primal domain decomposition method, *Numer. Linear Algebra Appl.* Vol.7, 2000, pp 687-714
- [Farhat et al 2001] **C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson and D. Rixen**, FETI-DP: a dual-primal unified FETI method – part I: A faster alternative to the two level FETI method, *Int. J. Numer. Meth. Engng* Vol.50, 2001, pp 1523-1544

- [Forestier 2000] **Romain Forestier**, Modélisation d'outils rigides libres en rotation Application à la simulation du laminage circulaire, *Rapport de stage, CEMEF*, (août 2000)
- [Gosselet 2003] **Pierre Gosselet**, Méthodes de décomposition de domaine et méthodes d'accélération pour les problèmes multichamps en mécanique non-linéaire, *Thèse de doctorat de l'Université Paris 6, Paris*, 2003
- [Gosselet et al 2002] **P. Gosselet, C. Rey, P. Dasset et F. Léné**, A domain decomposition method for quasi incompressible formulations with discontinuous pressure field, *Revue européenne des éléments finis*, vol. 11, 2002, pp 363–377
- [Klawonn 2002] **A. Klawonn, O.B. Widlund and M. Dryja**, Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients, *SIAM J. Numer. Anal.* Vol.40, no.1, 2002, pp 149-179
- [Li 2003] **Jing Li**, A Dual-Primal FETI Method for solving Stokes/Navier-Stokes Equations, dans *Proceeding of the 14th International Conference on Domain Decomposition Methods*, 2003
- [Mandel 1993] **J. Mandel**, Balancing domain decomposition, *Communications in Applied Numerical Methods*, vol. 9, 1993, pp. 233–241
- [Mandel et Tezaur 2001] **J. Mandel and R. Tezaur**, On the convergence of a dual-primal substructuring method, *Numer. Math.*, vol.88, pp 543-558, 2001
- [Marie 1997] **Stéphane Marie**, Un modèle de parallélisation S.P.M.D. pour la simulation numérique de procédés de mise en forme des matériaux, *Thèse de doctorat de l'École des Mines de Paris, Sophia-Antipolis* (1997).
- [Mocellin 1999] **Katia Mocellin**, Contribution à la simulation numérique tridimensionnelle du forgeage à chaud : étude de contact et calcul multigrille, *Thèse de Doctorat de l'École des Mines de Paris, Sophia-Antipolis*, (1999)

- [Montmitonnet 1994] **Pierre Montmitonnet**, Simulation des écoulements élastoplastiques par éléments finis. In *Seminaire de plasticité*, CEMEF École des mines de Paris Sophia-Antipolis, France, 1994
- [Perchat 2000] **Etienne Perchat**, Mini-élément et factorisations incomplètes pour la parallélisation d'un solveur de Stokes 2D. Application au forgeage, *Thèse de Doctorat de l'École des Mines de Paris, Sophia-Antipolis*, (2000)
- [PETSc] **Portable, Extensible Toolkit for Scientific computation**, <http://www.mcs.anl.gov/petsc/>
- [Rey et Gosselet 2004] **C. Rey and P. Gosselet**, An hybrid domain decomposition method for multifield problems, dans *Proceedings of the 7th ESAFORM conference on material forming*, 2004
- [Rixen et al 1998] **D. Rixen, C. Farhat and M. Géradin**, A two-step, two-field hybrid method for the static and dynamic analysis of substructure problems with conforming and non-conforming interfaces, *Comput. Methods Appl. Mech. Engng.* Vol.154, 1998, pp 229-264
- [Series et al 2003] **L. Series, F. Feyel et F.X. Roux**, Une méthode de décomposition de domaine à deux multiplicateurs de Lagrange, dans *Actes du 16eme congrès français de mécanique*, 2003.
- [Tallec 1994] **P. Le Tallec**, Domain decomposition methods in computational mechanics, *Computational Mechanics Advances* 1 (1994) pp. 121–220
- [Terzolo 2004] **Laurent Terzolo**, Vers une prédiction de la durée de vie des outils de forge à chaud par une détermination numérique du régime thermique établi et de l'usure abrasive, *Thèse de Doctorat de l'École des Mines de Paris, Sophia-Antipolis*, (2004)
- [Traoré 2001] **Karim Traoré**, Simulation thermo-mécanique du laminage circulaire Développement d'une formulation quasi-Eulérienne tridimensionnelle sur une architecture parallèle, *Thèse de Doctorat de l'Ecole des Mines de Paris, Sophia-Antipolis*, (2001)

[Vereecke et al 2003]

B. Vereecke, H. Bavestrello, D. Dureisseix, An extension of the FETI domain decomposition method for incompressible and nearly incompressible problems, *Comput. Methods Appl. Mech. Engng.* Vol.192, 2003, pp 3409-3429

ANNEXE A

Cette annexe contient des résultats des tests sur la barre encastrée du chapitre IV.1

<i>Partition</i>	<i>Méthode</i>	<i>Résidu final</i>	<i>Nombre d'itérations</i>	<i>Temps (s)</i>	<i>Temps par itération(s)</i>
2 sous-domaines	Schur	8,07E-10	118	3,969	0,034
	FETI	4,40E-10	72	2,040	0,028
	Primale-Duale	9,31E-10	118	2,048	0,017
3 sous-domaines	Schur	7,74E-10	238	6,306	0,026
	FETI	9,19E-10	126	2,914	0,023
	Primale-Duale	7,80E-10	239	6,505	0,027
4 sous-domaines	Schur	8,22E-10	462	17,080	0,037
	FETI	8,22E-10	291	10,621	0,036
	Primale-Duale	8,72E-10	384	13,114	0,034
5 sous-domaines	Schur	9,28E-10	477	19,041	0,040
	FETI	9,64E-10	294	11,666	0,040
	Primale-Duale	9,15E-10	373	12,377	0,033

Tableau A.1. Écrasement d'une barre longue : résultats des calculs sans préconditionnement, précision

10^{-9} .

<i>Partition</i>	<i>Méthode</i>	<i>Résidu final</i>	<i>Nombre d'itérations</i>	<i>Temps (s)</i>	<i>Temps par itération(s)</i>
2 sous-domaines	Schur	5,54E-07	110	3,548	0,032
	FETI	7,10E-07	62	1,808	0,029
	Primale-Duale	9,60E-07	91	1,924	0,021
3 sous-domaines	Schur	8,87E-07	225	9,734	0,043
	FETI	8,84E-07	113	3,716	0,033
	Primale-Duale	9,26E-07	199	5,849	0,029
4 sous-domaines	Schur	9,96E-07	311	11,379	0,037
	FETI	7,50E-07	265	13,046	0,049
	Primale-Duale	9,99E-07	267	8,303	0,031
5 sous-domaines	Schur	8,97E-07	327	13,061	0,040
	FETI	7,79E-07	267	9,698	0,036
	Primale-Duale	9,81E-07	277	8,623	0,031

Tableau A.2. Écrasement d'une barre longue : résultats des calculs sans préconditionnement, précision

10^{-6}

<i>Partition</i>	<i>Méthode</i>	<i>Résidu final</i>	<i>Nombre d'itérations</i>	<i>Temps (s)</i>	<i>Temps par Itération (s)</i>
2 sous-domaines	Schur	5,67E-07	32	0,978	0,031
	FETI	1,06E-08	29	1,114	0,038
	Primale-Duale	7,78E-08	23	1,569	0,068
3 sous-domaines	Schur	5,14E-07	44	1,378	0,031
	FETI	2,28E-08	40	1,543	0,039
	Primale-Duale	8,60E-07	29	0,794	0,027
4 sous-domaines	Schur	9,55E-07	56	3,006	0,054
	FETI	6,12E-08	46	2,116	0,046
	Primale-Duale	9,95E-07	35	1,186	0,034
5 sous-domaines	Schur	8,02E-07	60	3,098	0,052
	FETI	3,88E-08	50	2,158	0,043
	Primale-Duale	8,06E-07	37	1,381	0,037

Tableau A.3. Écrasement d'une barre longue : résultats des calculs avec les préconditionneurs allégés, précision 10^{-6}

ANNEXE B

Cette annexe contient des résultats des tests sur le forgeage d'un triaxe du chapitre V.4

<i>Nombre de processeurs</i>		<i>1</i>	<i>2</i>	<i>4</i>	<i>6</i>	<i>8</i>
<i>Résidu final</i>	Forge3®	7,69E-05	9,90E-05	9,33E-05	9,08E-05	8,82E-05
	PETSc	2,62E-08	2,92E-08	3,98E-08	2,57E-08	4,34E-08
	FETI		3,98E-07	6,53E-07	9,29E-07	9,56E-07
<i>Nombre d'itérations</i>	Forge3®	619	947	1218	1110	1523
	PETSc	641	741	888	931	975
	FETI		30	41	54	53
<i>Temps (s)</i>	Forge3®	116,70	89,60	60,43	36,71	38,98
	PETSc	80,28	48,93	31,70	21,31	16,30
	FETI	315,20	287,38	187,36	54,23	28,61
<i>Temps par itération (s)</i>	Forge3®	0,1885	0,0946	0,0496	0,0331	0,0256
	PETSc	0,1252	0,0660	0,0357	0,0229	0,0167
	FETI		2,32	1,49	0,42	0,28
<i>Temps de factorisation (s)</i>	FETI	315,20	217,82	126,27	31,48	13,74
<i>Temps pour itérations (s)</i>	FETI		69,57	61,09	22,76	14,87

Tableau B.1. Cas du triaxe à 10 602 nœuds : résultats des calculs

<i>Partition</i>	<i>Forge3</i>	<i>PETSc</i>	<i>FETI</i>
1	1,00	1,00	1,00
2	1,30	1,64	1,10
4	1,93	2,53	1,68
6	3,18	3,77	5,81
8	2,99	4,93	11,02

Tableau B.2. Cas du triaxe à 10 602 nœuds : accélérations obtenues sur le cluster

<i>Partition</i>	<i>Forge3</i>	<i>PETSc</i>	<i>FETI</i>
1	100,00%	100,00%	100,00%
2	65,12%	82,04%	54,84%
4	48,28%	63,31%	42,06%
6	52,98%	62,79%	96,86%
8	37,42%	61,56%	137,74%

Tableau B.3. Cas du triaxe à 10 602 nœuds : efficacité pour les trois solveurs

<i>Partition</i>	<i>Forge3</i>	<i>PETSc</i>	<i>FETI</i>
2	1,00	1,00	1,00
4	1,48	1,54	1,53
6	2,44	2,30	5,30
8	2,30	3,00	10,05

Tableau B.4. Cas du triaxe à 10 602 nœuds : accélérations obtenues sur le cluster avec le temps de résolution sur 2 processeurs comme référence

<i>Partition</i>	<i>Forge3</i>	<i>PETSc</i>	<i>FETI</i>
2	100,00%	100,00%	100,00%
4	74,14%	77,18%	76,69%
6	81,36%	76,54%	176,63%
8	57,46%	75,05%	251,16%

Tableau B.5. Cas du triaxe à 10 602 nœuds : efficacité pour les trois solveurs avec le temps de résolution sur 2 processeurs comme référence

<i>Nombre de processeurs</i>		<i>2</i>	<i>4</i>	<i>6</i>	<i>8</i>	<i>10</i>
<i>Résidu final</i>	Forge3®	8,07E-05	9,67E-05	9,83E-05	9,04E-05	8,33E-05
	PETSc	6,14E-08	5,54E-08	7,42E-08	9,94E-08	7,60E-08
	FETI			5,79E-07	9,03E-09	6,89E-07
<i>Nombre d'itérations</i>	Forge3®	2138	2360	2927	2508	3311
	PETSc	1748	2076	2147	2167	2224
	FETI			73	97	87
<i>Temps (s)</i>	Forge3®	1208,78	739,80	599,63	403,73	389,28
	PETSc	689,94	418,19	291,96	215,85	186,87
	FETI			2056,73	960,61	734,66
<i>Temps par itération (s)</i>	Forge3®	0,5654	0,3135	0,2049	0,1610	0,1176
	PETSc	0,3947	0,2014	0,1360	0,0996	0,0840
	FETI			7,3142	4,3638	3,3009
<i>Temps par factorisation(s)</i>	FETI			1522,80	537,32	447,49
<i>Temps pour itérations (s)</i>	FETI			533,94	423,29	287,18

Tableau B.6. Cas du triaxe 58 628 nœuds : résultats des calculs