# Méthode de préconditionnement multigrille hybride et parallèle pour l'accélération des calculs en mise en forme des matériaux

Frédéric Vi, Katia Mocellin, Etienne Perchat, Hugues Digonnet, Lionel Fourment

▶ **To cite this version:**

Frédéric Vi, Katia Mocellin, Etienne Perchat, Hugues Digonnet, Lionel Fourment. Méthode de préconditionnement multigrille hybride et parallèle pour l'accélération des calculs en mise en forme des matériaux. 13e colloque national en calcul des structures, Université Paris-Saclay, May 2017, Giens, Var, France. hal-01899362

**HAL Id: hal-01899362**
**https://hal.science/hal-01899362**

Submitted on 19 Oct 2018

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

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

CSMA 2017
13ème Colloque National en Calcul des Structures
15-19 mai 2017, Presqu'île de Giens (Var)

# Méthode de préconditionnement multigrille hybride et parallèle pour l'accélération des calculs en mise en forme des matériaux

F. Vi[1], K. Mocellin[1], E. Perchat[2], H. Digonnet[3], L. Fourment[1]

[1] *MINES ParisTech, PSL Research University, CEMEF, CNRS UMR 7635, CS 10207 rue Claude Daunesse, 06904 Sophia Antipolis Cedex, France, {frederic.vi, katia.mocellin, lionel.fourment}*@mines-paristech.fr

[2] *Transvalor S.A., Parc de haute technologie de Sophia-Antipolis, 06255 Mougins Cedex, France, etienne.perchat@transvalor.com*

[3] *Ecole Centrale de Nantes, ICI - Institut du Calcul Intensif, 44321 Nantes, France, hugues.digonnet@ec-nantes.fr*

**Résumé** —.Le préconditionnement d'un algorithme itératif de résolution à l'aide d'une méthode multigrille (MG) permet d'approcher le comportement asymptotique en O(N) recherché pour les très grands problèmes. La méthode MG retenue est hybride : algébrique pour la construction des systèmes linéaires des grilles grossières et géométrique pour celle des opérateurs de passage. Adaptée à une formulation mixte en vitesse et pression, compatible avec le calcul parallèle et avec les nombreux remaillages inhérents à la mise en forme, elle accélère les calculs par un facteur compris entre 1,5 et 2.

**Mots clefs** — Méthode itérative de résolution, Multigrille, Calculs Parallèles, Remaillages, Mise en Forme, Forgeage.

## 1. Introduction

La réduction des temps de calcul demeure un problème d'une grande actualité, à la fois d'un point de vue général mais aussi plus spécialement en mise en forme des matériaux où certaines simulations nécessitent encore plusieurs semaines voire plusieurs mois de calcul [1]. L'utilisation intensive des possibilités offertes par le calcul parallèle, qui reste toutefois bornée à 16 ou 64 processeurs pour son utilisation industrielle dans le formage des métaux, ne suffit pas à fournir les accélérations nécessaires. Il s'agit donc de se tourner vers d'autres directions, en observant qu'une partie de ces limitations résulte du coût asymptotique en $O(N^{1,5})$ des meilleures méthodes itératives de résolution utilisées dans une formulation implicite, N étant le nombre d'inconnues du problème [2]. Or, les méthodes de résolution multigrille permettent d'espérer un coût asymptotique linéaire en O(N), qui est donc extrêmement attractif pour aborder ces très grands problèmes [5]. Reposant sur l'utilisation de plusieurs niveaux de grilles sur un même domaine, elles ont plus spécialement été développées pour des problèmes d'écoulements de fluides et des systèmes linéaires résultant de formulations en volumes finis [3][4], leur utilisation pour des problèmes d'éléments finis [6] avec des remaillages fréquents [7][12] et dans un contexte de calculs parallèles soulevant des difficultés particulières qui font l'objet de ce travail.

Le principe des méthodes multigrille consiste à utiliser plusieurs niveaux de grille sur un même domaine. Il repose sur l'observation que les méthodes itératives classiques de résolution sont très efficaces pour calculer les hautes fréquences de la solution mais beaucoup moins pour les basses fréquences [5]. En revanche, ces basses fréquences peuvent être efficacement calculées en utilisant une grille plus grossière avec très peu d'inconnues [11]. La combinaison de ces deux méthodes de résolution constitue la méthode multigrille. Dans les méthodes multigrille algébriques, les différents niveaux de grille sont construits de manière abstraite à partir de la matrice du système, alors que les méthodes géométriques utilisent des maillages explicitement créés comme grille de calcul. L'approche retenue combine l'approche algébrique pour la construction des systèmes linéaires grossiers avec l'approche géométrique pour la définition des opérateurs de passage entre les grilles [7][12][13].

## 2. Préconditionneur multigrille pour un problème mixte

Considérons le système linéaire suivant à résoudre :

$$Ax = b \tag{1}$$

Ce système résulte d'une formulation mixte en vitesse et pression discrétisée par éléments finis à l'aide d'une interpolation P1+P1 (linéaire avec ajout d'une fonction bulle pour la vitesse) [10], après condensation des degrés de liberté associés à la fonction bulle, et linéarisation par un algorithme de Newton-Raphson [7][12]. (1) est généralement résolu par une méthode de Résidu Conjugué, préconditionnée par une décomposition LU incomplète, avec un niveau de remplissage supplémentaire par rapport au stockage compact. Cet algorithme, noté PCR-ILU1, est utilisé à partir de la bibliothèque PETSc [9]. Pour les applications de mise en forme étudiées (voir Figure 1 par exemple), son coût asymptotique est de $O(N^{1,5})$ [2].

### 2.1. La méthode multigrille

Une méthode multigrille se définit récursivement à partir d'une méthode deux-grille (2G). La grille la plus fine est le maillage de calcul sur lequel est défini le système à résoudre $A_h x_h = b_h$ (1), et sur laquelle on effectue quelques itérations d'une méthode itérative de résolution aussi simple que possible. Cette opération, appelée « lissage » ou « pré-lissage », sur la variable $x_h$ fournit la valeur $\tilde{x}_h$. L'équation résiduelle qui en résulte, $\tilde{r}_h = b_h - A_h \tilde{x}_h$, est projetée sur la grille grossière grâce à un opérateur de projection (ou de restriction) noté *R,* pour donner l'équation résiduelle sur la grille grossière :

$$A_H X_H = r_H = R\tilde{r}_h \tag{2}$$

(2) est résolue de manière exacte ; les indices h et H se réfèrent respectivement aux grilles fine et grossière. La correction ainsi obtenue, $X_H$, est interpolée sur la grille fine grâce à un opérateur d'interpolation (ou de prolongation) *P*, avant d'être ajoutée à $\tilde{x}_h$ :

$$\bar{x}_h = \tilde{x}_h + P X_h$$

De retour sur la grille fine, on effectue à nouveau quelques itérations de lissage (souvent nommées « post-lissage ») pour obtenir $\tilde{\tilde{x}}_h$, qui constitue la solution d'un cycle multigrille. Si la méthode multigrille est utilisée comme solveur, alors on recommence le cycle jusqu'à convergence. Si elle est utilisée comme préconditionneur, alors $\tilde{\tilde{x}}_h$ est le résultat du préconditionnement. Nous préfèrerons cette seconde approche qui se révèle un peu plus efficace que celle du solveur multigrille, et qui est aussi celle qui est proposée par PETSc [9].

Si la grille grossière contient trop d'inconnues pour être résolue de manière exacte à l'aide d'une méthode directe, alors on la résout par une nouvelle méthode deux-grille. On répète cette approche récursivement jusqu'à obtenir une grille grossière ayant moins de 1 500 nœuds (ou 6 000 inconnues pour notre formulation mixte). Dans la pratique, nous nous limiterons à trois niveaux de grilles, quitte à accentuer l'écart entre les niveaux, de manière à limiter l'augmentation de la largeur de bande du système (2) et à minimiser les difficultés de maillage.

### 2.2. Les éléments de la méthode multigrille

Une méthode multigrille repose sur le choix des opérateurs de lissage (et le nombre d'itérations de

pré-lissage et post-lissage), sur la définition des opérateurs de projection et d'interpolation, et sur la méthode de construction du système de la grille grossière (2) (et la méthode de résolution utilisée). A la suite de [7][12][13], nous utilisons une méthode *hybride* dans laquelle les opérateurs *P* et *R* sont construit de manière *géométrique* à partir des maillages des différents niveaux de grilles mais où la matrice grossière $A_H$ est construite de manière *algébrique* à partir des seuls opérateurs *P* et *R*.

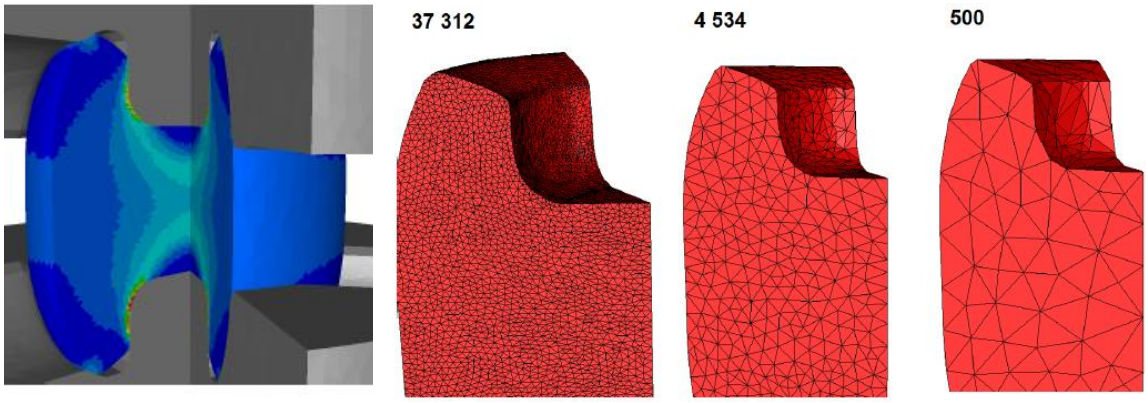


Figure 1: Gauche: forgeage d'un triaxe (isovaleurs de déformation cumulée en cours de forgeage). Droite : maillages fin, intermédiaire et grossier utilisés pour la méthode trois-grille à une étape intermédiaire des calculs.

### 2.2.1. Construction de la grille grossière

Les différents niveaux de grilles sont construits par déraffinement des maillages de niveaux supérieurs, en commençant par le maillage le plus fin (qui est aussi le maillage de calcul). Nous souhaitons imposer un facteur de déraffinement (rapport entre le nombre de nœuds du maillage initial et celui du maillage déraffiné) minimal de 8, ce qui correspondant à un déraffinement par un facteur 2 de la taille de maille dans chacune des directions de l'espace (en 3D). La technique de déraffinement [13] est dérivée de l'algorithme de remaillage parallèle par amélioration topologique [14]. Elle consiste à retirer itérativement des nœuds du maillage jusqu'à obtenir le facteur de déraffinement visé tout en respectant la géométrie du domaine (voir Figure 1). La méthode fonctionne en parallèle sur un domaine partitionné. Chaque processeur déraffine le sous domaine qui lui est affecté mais sans toucher aux nœuds d'interface. Cette interface est ensuite déplacée de manière à pouvoir déraffiner ces parties du maillage au cours de l'itération suivante. L'algorithme se conclut par un équilibrage des charges entre les processeurs [13]. Notons que les différents niveaux de grilles de chacun des sous domaines ne se trouvent pas sur les mêmes processeurs (voir Figure 2).

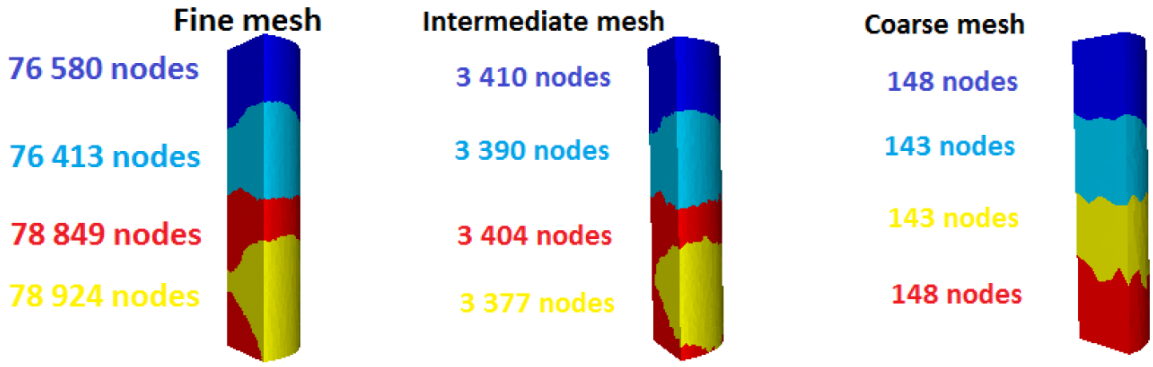


Figure 2: Déraffinement de maillage parallèle sur 4 processeurs et équilibrage des charges.

Le facteur de déraffinement est ajusté de manière à ce que la grille la plus grossière n'ait pas plus de 1500 nœuds. Pour une méthode 3 grilles, il se révèle également optimal d'imposer le même taux de déraffinement entre les différents niveaux de grilles (fin / intermédiaire, intermédiaire / grossier).

### 2.2.2. Construction des opérateurs de passage

Les différents niveaux de maillage permettent de construire les opérateurs d'interpolation/projection $P$ entre le maillage de niveau inférieur et son suivant. Dans le cadre de la méthode des éléments finis, cet opérateur n'est autre que la matrice des paramètres d'interpolation $N_H^l\left(\xi_H^k\right)$ (3) des nœuds du maillage fin à partir des nœuds du maillage grossier :

$$\forall k \in \Omega_h, \quad \exists \xi_H^k \in \left\{\left[0,1\right]\right\}^3, \quad x_k = \sum_{l \in \Omega_H} X_H^l N_H^l\left(\xi_H^k\right) \tag{3}$$

Certains nœuds de la surface du maillage fin peuvent ne pas appartenir au maillage grossier. Ils sont alors reprojetés sur la facette la plus proche avant d'en calculer les coordonnées barycentriques. La construction de ces opérateurs s'effectue en parallèle. Le projeté d'un nœud peut appartenir à un autre processeur (voir Figure 2), de sorte que la recherche de son élément de projection est délicate. Elle repose sur un algorithme de tri hiérarchique efficace et sur une analyse en deux étapes, grossière puis fine, qui permet de minimiser le nombre d'opérations et de communications entre les processeurs [8].

Nous utilisons une méthode de Galerkin dans laquelle l'opérateur de projection/restriction $R$ est la transposée de l'opérateur d'interpolation : $R = P^t$.

### 2.2.3. Construction du système grossier et sa résolution

La matrice du système grossier (2) est construite de manière algébrique à partir des opérateurs d'interpolation et projection :

$$A_H = RA_hP = P^tA_hP \tag{4}$$

Cette méthode algébrique (4) permet de projeter les différentes contraintes du problème sur les niveaux inférieurs, ce qui est particulièrement important pour conserver la condition d'incompressibilité du matériau sur les grilles inférieures, ainsi que les chargements dus au contact. Au contraire, une approche purement géométrique ne permet pas de garantir d'obtenir des conditions de contact sur le maillage grossier lorsqu'il y en a sur le maillage fin. Par ailleurs, la condition $R = P^t$ permet de conserver la symétrie de la matrice grossière, et donc de résoudre plus rapidement le système linéaire associé. En revanche, la méthode algébrique se traduit par une augmentation de la largeur de bande locale du système sur les grilles grossières, de sorte que le nombre de valeurs non nulles de la matrice augmente avec le nombre de grilles, et que l'on préfère donc le limiter à 3.

Les produits de matrices nécessaires à la construction de $A_H$ (4) sont effectués en parallèle et de manière très efficace par les bibliothèques incluses dans PETSc. Sur la grille la plus grossière, le système linéaire (2) est résolu par une méthode directe de factorisation parallèle : MUMPS [15].

### 2.2.4. Sélection des opérateurs de lissage

Les méthodes multigrille utilisent des opérateurs de lissage très simples, tels que l'algorithme de Jacobi. Dans le cadre présent d'une formulation mixte conduisant à une matrice non définie positive, des opérateurs plus complexes s'avèrent plus performants. Pour cela, nous avons considéré un lissage par l'algorithme de Richardson (5) où $\omega$ est un paramètre de relaxation (pris égal à 2/3) et $M^{-1}$ est

l'inverse d'une matrice de préconditionnement qui peut être la diagonale bloc de $A$, ou la matrice correspondant à l'algorithme SSOR, ou encore celle de la décomposition incomplète ILU0.

$$\forall i > 0, \quad \tilde{x}^{(i+1)} \leftarrow \tilde{x}^{(i)} + \omega M^{-1}\left(b - A\tilde{x}^{(i)}\right) \tag{5}$$

Pour un problème linéaire, le préconditionnement diagonal bloc (correspondant à l'algorithme de Jacobi) avec 3 opérations de pré et post lissage donne les meilleurs résultats, suivi de près par SSOR puis par ILU0 qui est seulement moins performante de 25%. En revanche, en considérant un problème non linéaire avec une géométrie peu régulière (celle du maillage de la Figure 1), la décomposition $LU$ incomplète ($M^{-1} = ILU0$), avec une seule itération de lissage et de post lissage, fournit de bien meilleures performances: *ILU0* est deux fois plus rapide que la diagonale bloc. Il s'avère donc un lisseur plus robuste et mieux adapté aux problèmes de mise en forme. Ce résultat souligne l'importance d'ajuster les paramètres de la méthode multigrille sur des problèmes représentatifs.

# 3. Résultats et applications

## 3.1. Convergence asymptotique

Pour le forgeage du triaxe (voir Figure 1), la Figure 3 montre l'évolution des temps de calcul en fonction du nombre de nœuds, avec le solveur de référence PCR-ILU1 et avec les préconditionnements 2 et 3 grilles, PCR-2G et PCR-3G respectivement. La courbe de PCR-ILU1 montre une tendance en $O(N^{1,65})$ qui est assez proche de celle en $O(N^{1,5})$ attendue. La tendance asymptotique de PCR-3G est en $O(N^{1,07})$, donc très proche de celle en O(N) espérée. Notons que le préconditionnement multigrille est toujours plus performant que ILU1, et que pour un « faible » nombre de nœuds (un peu plus de 30 000), la méthode 2 grilles est plus performante que la méthode 3 grilles.

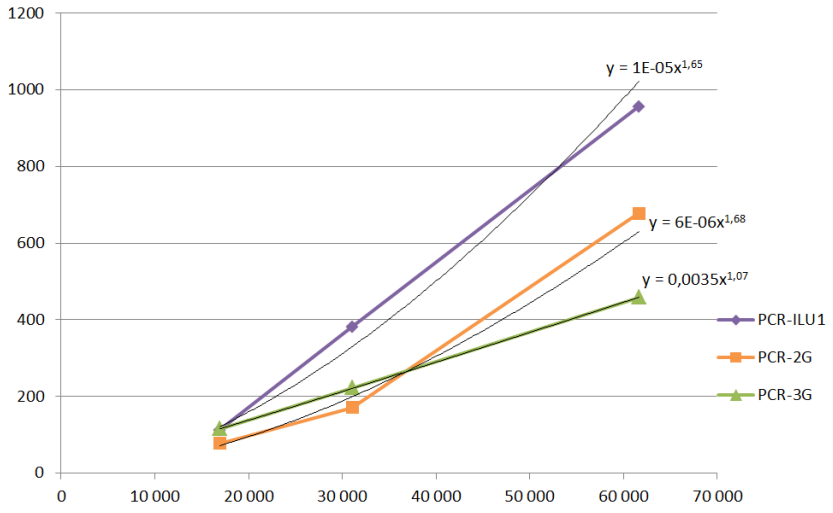


Figure 3: Coût asymptotique des méthodes 2G et 3G en comparaison du solveur PCR-ILU1 sur le problème du triaxe de la Figure 1.

## 3.2. Efficacité parallèle

La Figure 4 montre l'accélération parallèle (rapport entre le temps de calcul sur un seul processeur et celui sur $n$ processeurs) obtenue sur un problème linéaire avec environ 310 000 nœuds. Il se produit une super accélération avec les deux solveurs. Elle s'explique par un effet de mémoire cache qui est un

peu moins marqué avec le préconditionneur multigrille. L'efficacité parallèle de PCR-3G est excellente, semblable à celle de PCR-ILU1. Cette similitude est tout particulièrement marquée en prenant pour temps de référence le calcul sur 2 processeurs, compte-tenu du fait que les préconditionneurs ILU1 et ILU0 ont des comportements différents en séquentiel et en parallèle. Ce résultat montre l'excellent niveau de parallélisation de toute la chaîne des calculs du préconditionneur multigrille, depuis la génération des maillages de bas niveaux jusqu'aux produits matriciels.

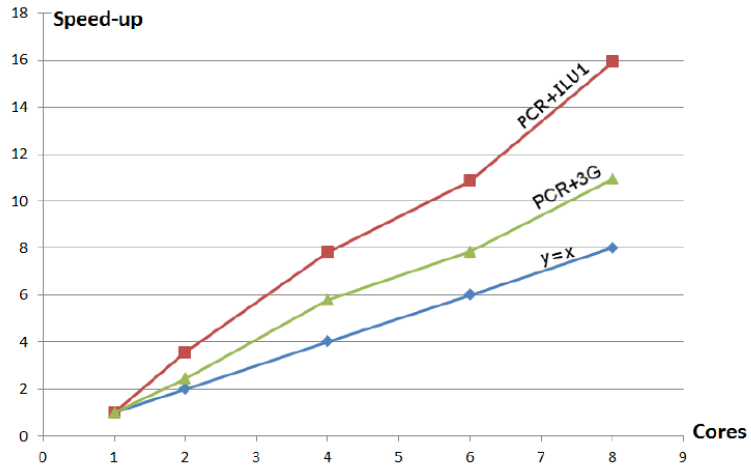


Figure 4: Accélération parallèle obtenue en fonction du nombre de processeurs (de 1 à 8) pour le forgeage d'un lopin cylindrique entre deux plans, avec PCR-ILU1 et avec le préconditionneur multigrille (PCR-3G). Maillage de 310 000 nœuds, loi de comportement linéaire et contact parfaitement collant

## 3.3. Application

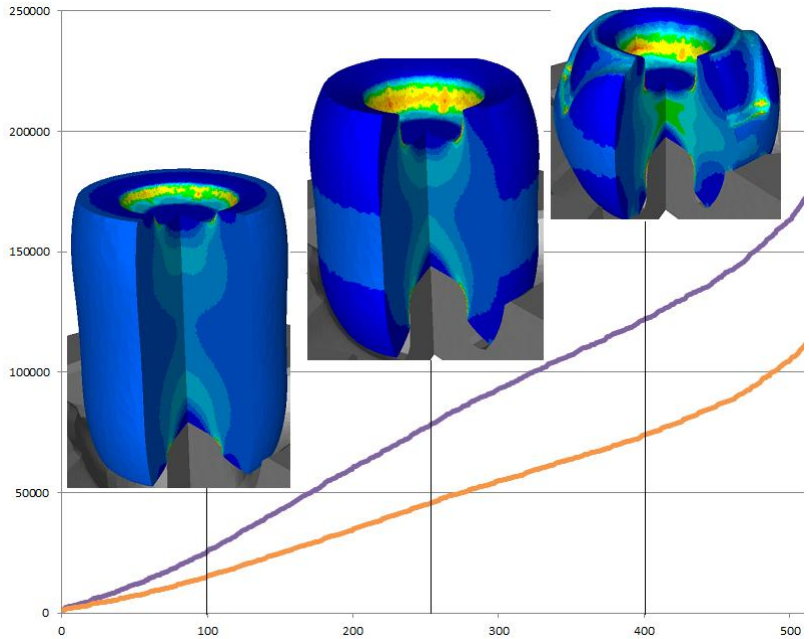


Figure 5: Evolution temporelle (en incréments de calcul) du temps de calcul total nécessaire pour la simulation du forgeage du triaxe de la Figure 1, avec le solveur de référence PCR-ILU1 (violet) et avec le préconditionnement multigrille PCR-3G (jaune-orange), avec au-dessus les isovaleurs de déformations cumulées aux incréments correspondants.

Le préconditionnement multigrille est appliqué au forgeage du triaxe de la Figure 1. Le maillage

initial comporte 34 000 nœuds. Après 26 remaillages et 500 incréments de temps, le maillage final est constitué de 43 000 nœuds. La Figure 5 montre l'évolution des temps de calcul en fonction des incréments de temps ; elle donne aussi une idée des déformations subies par le matériau. La résolution des systèmes linéaires représente environ 70% du temps total de simulation, de sorte qu'avec une accélération multigrille proche de 2 (observée sur la résolution d'un seul système linéaire) nous n'obtenons finalement qu'une accélération maximale d'environ 1,5 sur le temps total de simulation. En toute fin de procédé, le préconditionnement multigrille s'avère un peu moins efficace lorsque la matière s'écoule en bavure ; l'accélération finale obtenue est alors de 1,33.

# 4. Conclusion

L'utilisation d'une méthode multigrille hybride, géométrique pour la construction des opérateurs de passage et algébrique pour la construction des systèmes linéaires grossiers, fournit un préconditionnement très robuste et efficace de la méthode du Résidu Conjugué, dans le cadre de la mise en forme des matériaux. La méthode développée présente une excellente scalabilité parallèle. Elle s'avère également robuste vis-à-vis des nombreux remaillages nécessaires durant une simulation de forgeage. Elle permet d'accélérer des calculs complexes en réduisant les temps de 30% à 50%, ce qui est un résultat intéressant compte-tenu de l'ensemble des optimisations déjà apportées aux solveurs. Le principal intérêt de cette méthode, qui reste à confirmer et à exploiter, est son comportement asymptotique en O(N) car il ouvre de nouvelles perspectives pour considérer des problèmes de plus grandes tailles.

## 2.3. Références


[1] Pérémé M, Marie S, Barbelet M, Perchat E, Ducloux R, Fourment L, Chenot JL. Benefits of High Performance Computing applied to the numerical simulation of forged parts. 20th International Forging Congress, Hyderabad, India, 2011.

[2] Coupez T, Marie S. From a direct solver to a parallel iterative solver in 3-d forming simulation. International Journal of High Performance Computing Applications Jan 1997; 11(4):277–285, doi: 10.1177/109434209701100402.

[3] Ghia U, Ghia K, Shin C. High-re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. Journal of Computational Physics 1982; 48(3):387 – 411, doi:10.1016/0021-9991(82)90058-4.

[4] Vanka S. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. Journal of Computational Physics 1986; 65(1):138–158, doi:10.1016/0021-9991(86)90008-2.

[5] Yavneh I. Why Multigrid Methods Are So Efficient. Computing in Science & Engineering Nov 2006; 8(6):12–22, doi:10.1109/MCSE.2006.125.

[6] Feng YT, Peric D, Owen DRJ. A non-nested Galerkin multi-grid method for solving linear and nonlinear solid mechanics problems. Computer Methods in Applied Mechanics and Engineering May 1997; 144(3–4):307–325, doi:10.1016/S0045-7825(96)01183-8.

[7] Rey B, Mocellin K, Fourment L. A node-nested Galerkin multigrid method for metal forging simulation. Computing and Visualization in Science 2008; 11(1):17–25, doi:10.1007/s00791-006-0054-5.

[8] Ramadan M, Fourment L, Digonnet H. Fast resolution of incremental forming processes by the multi-mesh method. Application to cogging. International Journal of Material Forming 2014; 7(2):207–219,

doi:10.1007/s12289-012-1121-8.

[9] Balay S, Abhyankar S, Adams MF, Brown J, Brune P, Buschelman K, Dalcin L, Eijkhout V, Gropp WD, Kaushik D, et al.. PETSc user's manual. Technical Report ANL-95/11 - Revision 3.6, Argonne National Laboratory 2015.

[10] Arnold DN, Brezzi F, Fortin M. A stable finite element for the Stokes equations. CALCOLO Dec 1984; 21(4):337–344, doi:10.1007/BF02576171.

[11] Wesseling P. An introduction to Multigrid Methods. John Wiley & Sons, Chichester, 1992.

[12] Mocellin K, Fourment L, Coupez T, Chenot JL. Toward large scale F.E. computation of hot forging process using iterative solvers, parallel computation and multigrid algorithms. International Journal for Numerical Methods in Engineering 2001; 52(5-6):473–488, doi:10.1002/nme.304.

[13] Carte G, Coupez T, Guillard H, Lanteri S. Coarsening techniques in multigrid applications on unstructured meshes. European congress on computational methods in applied sciences and engineering, ECCOMAS, 2000.

[14] Coupez T, Digonnet H, Ducloux R. Parallel meshing and remeshing. Applied Mathematical Modelling Dec 2000; 25(2):153–175, doi:10.1016/S0307-904X(00)00045-7

[15] Amestoy PR, Guermouche A, L'Excellent JY, Pralet S. Hybrid scheduling for the parallel solution of linear systems. Parallel Computing 2006; 32(2):136–156.