



**HAL**  
open science

# Préconditionneur grille grossière pour la résolution par solveur itératif de problème éléments-finis fortement non-linéaires de grandes tailles

Basile Marchand, Stéphane Quilici

## ► To cite this version:

Basile Marchand, Stéphane Quilici. Préconditionneur grille grossière pour la résolution par solveur itératif de problème éléments-finis fortement non-linéaires de grandes tailles. 14e colloque national en calcul des structures, May 2019, Presqu'île de Giens, France. hal-02077554

**HAL Id: hal-02077554**

**<https://hal.science/hal-02077554>**

Submitted on 19 Jul 2022

**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.

# Préconditionneur grille grossière pour la résolution par solveur itératif de problème éléments-finis fortement non-linéaires de grandes tailles.

B. Marchand<sup>1</sup>, S. Quilici<sup>2</sup>

<sup>1</sup> Centre des Matériaux, Mines ParisTech, Université PSL, [basile.marchand@mines-paristech.fr](mailto:basile.marchand@mines-paristech.fr)

<sup>2</sup> Transvalor Innovation, [stephane.quilici@transvalor.com](mailto:stephane.quilici@transvalor.com)

---

**Résumé** — Afin de permettre l'emploi des solveurs itératifs pour les problèmes fortement non-linéaires un nouveau preconditionneur a été développé et implémenté dans Z-set/Zébulon. Ce preconditionneur est basé sur la factorisation de la matrice tangente globale du problème, formulée sur un maillage grossier. La factorisée ainsi calculée, à faible coût, est alors utilisée pour preconditionner le système du problème de référence. L'emploi de ce preconditionneur permet d'améliorer significativement la convergence des solveurs itératifs et ainsi, de réaliser pour des coûts réduits des calculs de structures de grandes dimensions à comportements mécaniques complexes.

**Mots clés** — Comportements non-linéaires, Solveur itératif, Préconditionneur.

---

## 1 Introduction

Les besoins en finesse de modélisation des problèmes actuels entraînent une croissance forte du nombre de degrés de liberté à traiter dans les formulations éléments finis. Le standard industriel actuel, repose sur des calculs comportant de l'ordre du million de degrés de liberté tout en considérant des comportements mécaniques complexes. Pour réaliser ce type de calcul éléments finis il est alors nécessaire de mettre en place des stratégies de résolution adéquates. Les différentes stratégies possibles pour la résolution du système linéaire sont :

- Résolution directe
  - Factorisation Cholesky ou LU en mémoire partagée ou distribuée [1].
- Résolution itérative
  - Solveurs itératifs Gradient Conjugué ou GMRes [6][5].
  - Méthodes de décomposition de domaines BDD ou FETI [2].

La résolution directe basée sur la mise en place de calculs à mémoire distribuée présente des performances tout à fait remarquables. De même, les avancées au sujet des méthodes de décomposition de domaines [3][7] permettent aujourd'hui d'employer ce type d'approche sur une vaste gamme de problèmes. Cependant ces solveurs nécessitent des moyens de calculs qui généralement ne sont accessibles qu'à de très grands industriels. A l'opposée, les solveurs itératifs permettent avec des ressources limitées de réaliser des calculs habituellement coûteux notamment en mémoire. Cependant les solveurs itératifs présentent le défaut majeur d'avoir une convergence de très mauvaise qualité lorsque l'on s'intéresse à des problèmes non-linéaires. Cette perte de qualité de la convergence est due au mauvais conditionnement de la matrice tangente globale dans le cadre de comportement mécaniques non-linéaires complexes. Afin de passer outre ces problèmes de convergence il est nécessaire d'introduire une étape de preconditionnement du système linéaire. Pour une revue détaillée de ces différentes approches le lecteur peut se référer à [10]. Cependant les méthodes de preconditionnement classiques ont tendance à perdre en efficacité sur des problèmes de mécanique des solides non-linéaires représentatifs.

Dans ce travail on propose la formulation d'un nouveau preconditionneur basé sur la formulation et la résolution exacte du problème éléments finis initial exprimé sur un maillage grossier. Pour ce faire différentes stratégies de construction de l'opérateur tangent sur le maillage grossier sont envisageables et étudiées dans ce travail. La stratégie de preconditionnement proposée est mise en œuvre sur différents

problèmes représentatifs considérant des comportements non-linéaires ainsi que des grandes déformations.

## 2 Pré-conditionneur grille grossière

### 2.1 Formulation du préconditionneur

Lors de la résolution d'un problème de mécanique non-linéaire par la méthode des éléments finis, le système linéaire à résoudre à chaque itération de l'algorithme de Newton global est le suivant :

$$\mathbf{K}\delta\mathbf{u} = -\mathbf{R} \quad (1)$$

Avec  $\mathbf{K} \in \mathbb{R}^{n_{dof} \times n_{dof}}$  la matrice tangente globale du système,  $\mathbf{R} \in \mathbb{R}^{n_{dof}}$  le résidu et  $\delta\mathbf{u} \in \mathbb{R}^{n_{dof}}$  la correction de Newton de l'incrément de déplacement courant.

Pour résoudre ce problème par une méthode itérative (Gradient Conjugué si la matrice tangente est symétrique et GMRes sinon) il est nécessaire, si l'on souhaite avoir des propriétés de convergence correctes, d'introduire une matrice  $\mathbf{P}$  dite préconditionneur de la manière suivante :

$$\mathbf{P} \cdot \mathbf{K}\delta\mathbf{u} = -\mathbf{P} \cdot \mathbf{R} \quad (2)$$

La matrice de préconditionnement  $\mathbf{P} \in \mathbb{R}^{n_{dof} \times n_{dof}}$  est une matrice qui se doit, pour que le préconditionneur soit efficace, d'être :

- Peu coûteuse à construire. Le préconditionneur idéale est bien entendu  $\mathbf{P} = \mathbf{K}^{-1}$  il est cependant bien trop coûteux à construire.
- Avec des propriétés telles que  $\text{cond}(\mathbf{PK}) < \text{cond}(\mathbf{K})$  afin de garantir une meilleure convergence du solveur itératif.

Dans la littérature il existe toute une zoologie de préconditionneurs, par exemple factorisation incomplète de la matrice  $\mathbf{K}$ , factorisations par blocs. Ces différents préconditionneurs permettent, lorsque le problème initial est relativement bien conditionné, d'améliorer la convergence du solveur itératif. Néanmoins sur des cas plus représentatifs les préconditionneurs classiques de la littérature sont mis en échec et donc les solveurs itératifs ne sont plus utilisables. La stratégie que l'on met en place dans ce travail est de construire le préconditionneur du problème en introduisant une seconde discrétisation spatiale, un maillage déraffiné de la structure, comme définit dans l'équation (3).

$$\mathbf{P} = \mathbf{Z} \cdot \mathbf{K}_{grossier}^{-1} \cdot \mathbf{Z}^T + \lambda \tilde{\mathbf{K}}_{fin}^{-1} \quad (3)$$

La matrice  $\mathbf{K}_{grossier}^{-1} \in \mathbb{R}^{n_{dof}^{grossier} \times n_{dof}^{grossier}}$  symbolise l'inverse de la matrice tangente globale du problème grossier. En pratique cette inverse n'est bien évidemment pas calculée mais on réalise une décomposition de Cholesky (ou LU dans le cas non-symétrique) de la matrice tangente du problème grossier  $\mathbf{K}_{grossier}$ . La matrice  $\mathbf{Z} \in \mathbb{R}^{n_{dof}^{fin} \times n_{dof}^{grossier}}$  est la matrice de passage permettant de projeter un champ nodal du maillage grossier sur le maillage fin, cet opérateur est construit à partir des fonctions de forme éléments finis des maillages grossier et fin. L'opérateur  $\tilde{\mathbf{K}}_{fin}^{-1} \in \mathbb{R}^{n_{dof}^{fin} \times n_{dof}^{fin}}$  symbolise le calcul d'une factorisation, Cholesky ou LU, incomplète de la matrice globale du problème initial. Enfin le coefficient  $\lambda \in \mathbf{R}_+$  permet de pénaliser la factorisation exacte de la matrice globale grossière par la factorisation incomplète. En pratique le coefficient  $\lambda$  est pris égal à 1 ou 10.

### 2.2 Interprétation

Au premier abord l'approche proposée ici ressemble fortement aux méthodes de résolution de types multi-grilles très développées dans la communauté de la mécanique des fluides [8]. Il existe en effet une similitude, due notamment au fait que l'on introduit une discrétisation grossière du problème. Cependant cette ressemblance s'arrête ici puisque dans les méthodes multi-grilles classiques la grille grossière ne sert qu'à restreindre l'opérateur  $\mathbf{K}$  en vue d'une factorisation moins coûteuse [9]. La particularité de l'approche que l'on propose vient du fait que l'opérateur grossier  $\mathbf{K}_{grossier}$  n'est pas une restriction de  $\mathbf{K}$  mais provient de l'intégration des lois de comportement directement sur le maillage grossier. De fait, le

maillage grossier ne sert pas uniquement de support de projection mais est véritablement le lieu d'une formulation et d'une résolution éléments finis.

Du fait que la loi de comportement du problème soit réellement intégrée sur le maillage grossier il est possible d'interpréter le préconditionneur (3) comme une approche de préconditionnement multi-échelle. En effet le préconditionneur grille grossière se compose de deux termes :

- Factorisation "exacte" de la matrice tangente globale du problème grossier. Correspond à la prise en compte dans le préconditionneur des interactions lointaines du problème du fait que la taille des éléments-finis est bien plus grande dans le problème grossier que dans le problème initial.
- Factorisation "incomplète" de la matrice tangente globale du problème initial. Correspond à la prise en compte uniquement des interactions locales. En effet, on le rappelle, le principe d'une factorisation incomplète est de construire les matrices triangulaires de la factorisation en ne calculant que les termes initialement non nuls dans la matrice (permettant ainsi d'avoir exactement les mêmes profils de matrice) pour se faire tous les termes d'interaction lointaine sont négligés.

## 2.3 Implémentation dans Z-set/Zébulon et intérêt de l'approche

La stratégie de préconditionnement proposée dans ce travail a été implémentée dans le solveur éléments finis Zébulon de la suite logiciel Z-set [4]. Afin d'être à même de traiter des cas-tests industriels le préconditionneur a été implémenté pour des matrices symétriques ou non. C'est-à-dire que l'étape de factorisation incomplète a été implémentée de deux manières : (i) factorisation de Cholesky incomplète ; (ii) factorisation LU incomplète. De plus pour permettre la résolution de problèmes de grande taille le préconditionneur a été implémenté : (i) en séquentiel ; (ii) en parallèle à mémoire partagée (multi-thread) ; (iii) en parallèle à mémoire distribuée (MPI). L'unique étape n'ayant pas été réalisée en interne mais se basant sur une librairie externe est la factorisation "exacte" de la matrice tangente globale du problème grossier. Cette factorisation est réalisée en utilisant la librairie MUMPS [1]. De plus l'usage de la suite logiciel Z-set offre l'avantage de fournir divers outils de remaillages pour la construction du maillage grossier à partir du maillage initial. Pour ce faire les outils de remaillages existants et interfacés avec Z-set, Distene ou MMG, ont montré de bonnes performances.

Le préconditionneur proposé ici et tel qu'il est implémenté dans Z-set/Zébulon offre deux avantages majeurs pour la résolution de problèmes industriels représentatifs :

- Il permet tout d'abord d'utiliser dans le cadre de problèmes non-linéaires les solveurs itératifs sans dégrader de manière notable les propriétés de convergence de ces derniers. Cette propriété est illustrée dans la partie 3.
- L'utilisateur fixe lui-même le coût qu'il est prêt à payer pour son calcul. En effet c'est l'utilisateur qui fournit le maillage grossier dont la taille pilote directement le coût du calcul. S'il souhaite une solution rapide l'utilisateur fournira un maillage grossier conséquent. En revanche si le critère est que le calcul doit passer sur une machine à faible mémoire vive l'utilisateur fournira alors un maillage grossier très petit.

## 3 Illustrations

Dans cette partie nous allons présenter les performances et applications possibles du préconditionneur grille grossière. L'intégralité des calculs et résultats présentés dans cette partie ont été réalisés avec le code éléments finis Z-set/Zébulon. Permettant ainsi de garantir que les comparaisons en termes notamment de temps de calcul réalisées dans la suite ont un sens.

### 3.1 Performances brutes

Les premières applications consistent en une épreuve de force pour illustrer les performances brutes du solveur itératif avec préconditionneur grille grossière par rapport à d'autres méthodes de préconditionnement plus classiques ainsi que par rapport à un solveur direct.

Pour cela on considère tout d'abord deux cas très simples : (i) un cube élastique linéaire soumis à un chargement de cisaillement ; (ii) une poutre en flexion dont on fait varier l'allongement.

Sur ces deux cas nous allons comparer le nombre d'itération à convergence du gradient conjugué pour

trois préconditionneurs différents : (i) le préconditionneur dit lumped  $\mathbf{P} = \text{diag}(\mathbf{K})^{-1}$  ; (ii) la factorisation de Cholesky incomplète ; (iii) le préconditionneur grille grossière. Les courbes de convergences obtenues en fonction du nombre de degrés de libertés du maillage fin sont représentées sur la Figures 1a (pour le cas du cube) et la Figure 1b (pour le cas de la poutre).

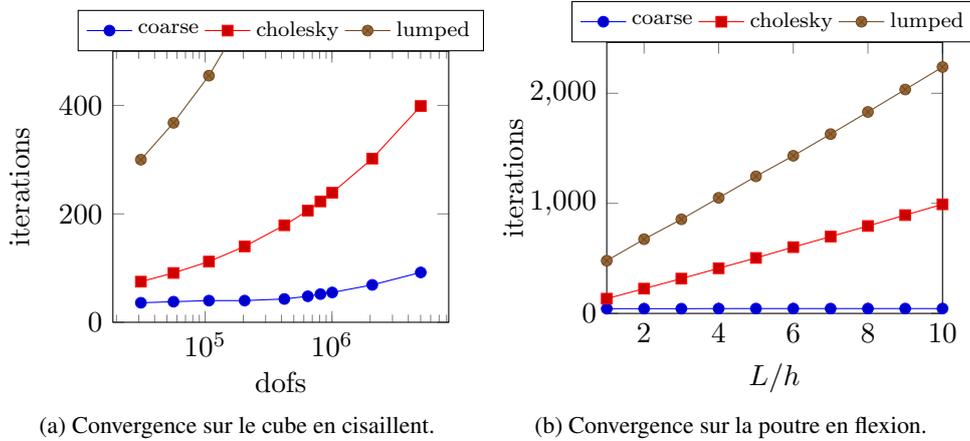


FIGURE 1 – Évolution du nombre d’itérations à convergence en fonction du nombre de degrés de liberté

On peut observer sur la Figure 1a que l’apport du préconditionneur grille grossière en terme de rapidité de convergence est significatif vis-à-vis des préconditionneurs classiques. De plus on constate que le nombre d’itérations à convergence avec le préconditionneur proposé ici évolue presque de manière linéaire pour une discrétisation grossière fixée. Sur le cas-test de la poutre on constate là également un apport significatif du préconditionneur grille grossière sur la qualité de la convergence du gradient conjugué. De plus dans ce cas où le nombre de degrés de liberté du problème grossier croît avec l’allongement de la poutre, le nombre d’itération est constant alors qu’avec les approches classiques le nombre d’itération augmente linéairement avec l’allongement.

Si l’on s’intéresse maintenant aux temps de résolution en séquentiel on obtient les courbes 2a et 2b. On observe que la réduction du nombre d’itération se traduit bien en une réduction du temps de résolution du système linéaire. Ce gain de temps traduit ainsi le fait qu’une itération de gradient conjugué avec le préconditionneur grille grossière n’est pas excessivement plus coûteuse qu’avec un préconditionneur classique. De plus sur la courbe 2a on trace également les temps de résolution du problème avec un solveur direct (les carrés bleus vides), ici MUMPS. On constate alors que pour un million de degrés de liberté l’utilisation du gradient conjugué de Z-set/Zébulon avec le préconditionneur grille grossière permet d’obtenir un speed-up de 10 par rapport à MUMPS.

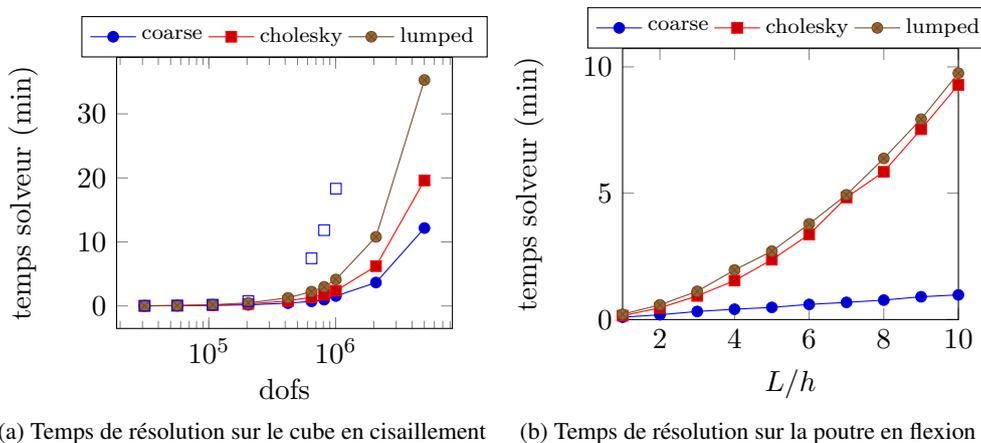


FIGURE 2 – Évolution du temps de résolution en fonction du nombre de degrés de liberté

Pour finir cette première analyse de performance du préconditionneur grille grossière et afin de valider l’implémentation en parallèle à mémoire distribuée il a été réalisé le cas-test classique de l’exten-

sibilité faible. Le motif de base considéré est un cube, de 500 000 degrés de libertés, que l'on répète dans les trois directions de l'espace afin d'obtenir un nouveau cube. Chaque motif représente alors un sous-domaine géré par un CPU. On trace sur la Figure 3 le temps de résolution en fonction du nombre de cœurs utilisés, donc du nombre de sous-domaine.

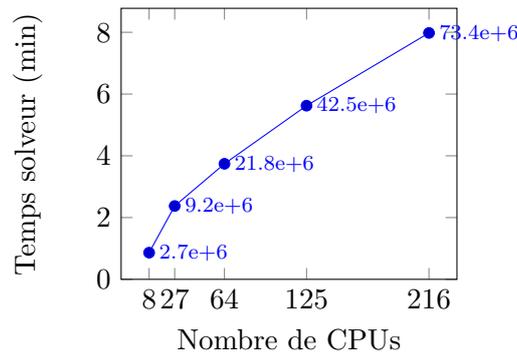


FIGURE 3 – Extensibilité faible du Gradient Conjugué avec préconditionneur grille grossière

On observe alors sur la Figure 3 que l'extensibilité du solveur n'est pas parfaite, puisque l'on a une augmentation notable du temps de calcul en fonction du nombre de sous-domaine (donc de la taille du problème). Cependant le solveur gradient conjugué utilisant le préconditionneur grille grossière présente des performances notables en parallèle. Il permet en effet de résoudre un problème à 70 millions d'inconnues en 8 minutes sur 216 cœurs.

### 3.2 Applications a des problèmes non-linéaires

Pour finir nous allons dans cette partie présenter quatre cas-tests non-linéaires représentatifs afin d'illustrer l'intérêt du préconditionneur grille grossière.

### 3.3 Comportement élasto-plastique

Le premier cas-test que l'on considère est le calcul élasto-plastique d'une culasse de moteur<sup>1</sup>. Sur cet exemple il a été réalisé la comparaison entre : (i) le solveur direct de Z-set (MUMPS) ; (ii) le gradient conjugué de Z-set avec préconditionneur grille grossière ; (iii) le solveur direct d'Abaqus. On présente dans le Tableau 1 les temps de résolution et occupations mémoire obtenus avec les trois solveurs pour un maillage de 2 500 000 degrés de liberté. On constate que le solveur itératif avec préconditionneur grille grossière permet un gain d'un facteur 3 par rapport à Abaqus aussi bien sur l'occupation mémoire que sur le temps de résolution. De plus la même comparaison a été réalisée pour un maillage deux fois plus fin (5 000 000 de degrés de liberté) et il apparait que le coût de la résolution avec Z-set et le préconditionneur grille grossière évolue alors linéairement alors que le coût du calcul explose quant à lui avec un facteur 2.6 sur l'occupation mémoire et un facteur 6 sur le temps de résolution.

	Abaqus direct	Z-set mumps	Z-set CG+grossier
Memoire	30 Go	40 Go	10 Go
# Coeurs	Temps execution		
1	1896 s	2177 s	496 s
4	607 s	786 s	194 s
8	400 s	624 s	153 s

(a) Comparaison entre Abaqus, Z-set avec solveur direct et Z-set avec solveur itératif et préconditionneur grille grossière.

	Abaqus	Z-set CG+grossier
Memoire	80 Go	20 Go
Temps 8 coeurs	2433 s	350 s

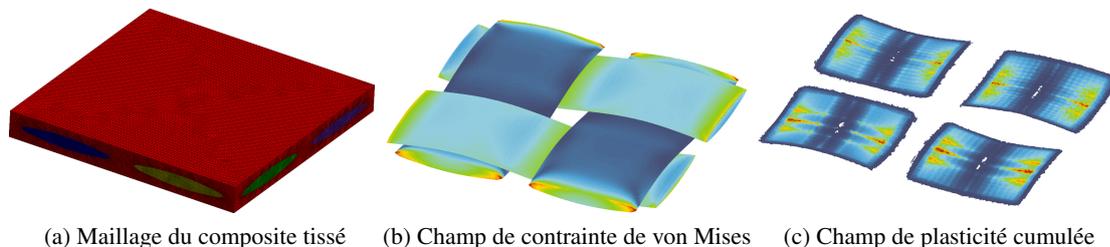
(b) Comparaison Abaqus direct et Z-set solveur itératif préconditionneur grille grossière pour le maillage de la culasse raffiné à 5 millions de ddl.

TABLE 1 – Comparaison Z-set Abaqus sur le calcul de culasse.

1. Les auteurs tiennent à remercier la société Montupet pour le maillage initial de la culasse

### 3.3.1 Matériaux composite tissé

Considérons à présent le calcul d'un Volume Élémentaire Représentatif d'un matériau composite tissé. Les torons sont élastique linéaire tandis que la matrice est élasto-visco-plastique. Le ratio d'hétérogénéité entre torons et matrice est de  $10^3$ . On représente sur la Figure 4a le maillage du problème fin<sup>2</sup>. Ce maillage mène à un problème constitué de 2 800 000 degrés de liberté. La résolution de ce problème avec le préconditionneur introduit ici, repose sur la construction d'un problème grossier associé à 47 000 degrés de libertés. Le problème est résolu en 12 incréments, 2 itérations par incrément. On représente sur les Figures 4b et 4c respectivement la contrainte de Mises et le champ de plasticité cumulée au dernier incrément de chargement.



On indique dans le Tableau 2 les temps de résolution : (i) temps du solveur cumulé, i.e. somme pour toutes les résolutions de systèmes linéaires ; (ii) le temps d'exécution total du calcul. Les temps de résolution sont alors comparés à ceux obtenus en utilisant le solveur direct MUMPS.

Solveur	Temps solveur cumulé (s)	Temps execution (s)
Mumps	5241	7 360
CG grille grossière	1501	4156

TABLE 2 – Temps de résolution du calcul sur composite tissé

On constate sur ce cas que l'on obtient avec le préconditionneur grille grossière un speed-up de 1.77 par rapport à la résolution avec MUMPS pour ce qui est des résolution de systèmes linéaires. Concernant l'exécution totale le speed-up obtenu tombe à 1.42.

### 3.4 Comportement élasto-visco-plastique fortement non-linéaire

Le second exemple est le calcul d'un éprouvette plate en traction ayant un trou conique en son centre. Le matériau associé présente un comportement élasto-visco-plastique. Le maillage de l'éprouvette mène à un problème constitué de 1 674 000 degrés de liberté. La résolution est faite en 20 incréments de chargement, chaque incrément nécessite entre 2 et 4 itérations pour converger. Le comportement faisant intervenir des écrouissages cinématiques non-linéaires, la matrice tangente globale devient au cours du calcul non-symétrique, nécessitant donc l'emploi du GMRes comme solveur itératif. Tout d'abord on présente sur la Figure 4 les courbes de convergence du solveur GMRes au premier incrément de chargement et à l'incrément de chargement le plus critique. Ces courbes sont tracées pour le GMRes avec : (i) le préconditionneur grille grossière ; (ii) le préconditionneur factorisation LU incomplète.

On observe sur la Figure 4 que la convergence du GMRes dans le cadre de l'utilisation du préconditionneur grille grossière reste d'excellente qualité au fur et à mesure du calcul. Par opposition la convergence avec la factorisation incomplète classique présente une convergence initialement acceptable, bien que très lente, mais en revanche se dégrade fortement au cours du calcul jusqu'à ne plus converger lorsque les non-linéarités deviennent trop fortes.

On donne dans le Tableau 3 les temps de résolution associés au calcul de l'éprouvette en comparant trois cas de figures : (i) l'utilisation du solveur direct MUMPS ; (ii) l'utilisation du GMRes avec le préconditionneur grille grossière en considérant un problème grossier à 1 467 ddl ; (iii) l'utilisation du GMRes avec le préconditionneur grille grossière en considérant un problème grossier à 29 862 ddl.

2. Les auteurs tiennent à remercier le pôle M&S de Safran Tech pour le maillage initial qui a ensuite été raffiné

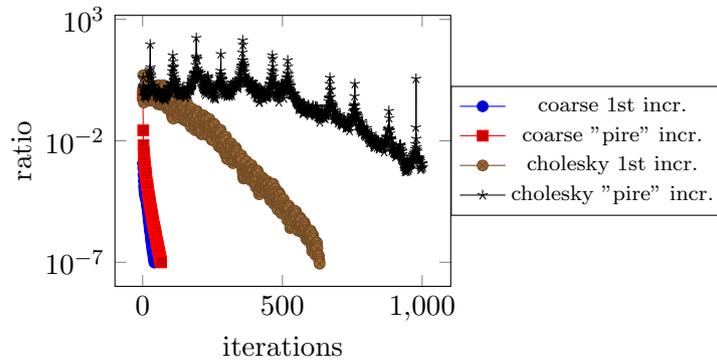


FIGURE 4 – Comparaison de la convergence du GMRes entre préconditionneur grille grossière et préconditionneur factorisation incomplète seul sur la meilleure et la pire itération de Newton

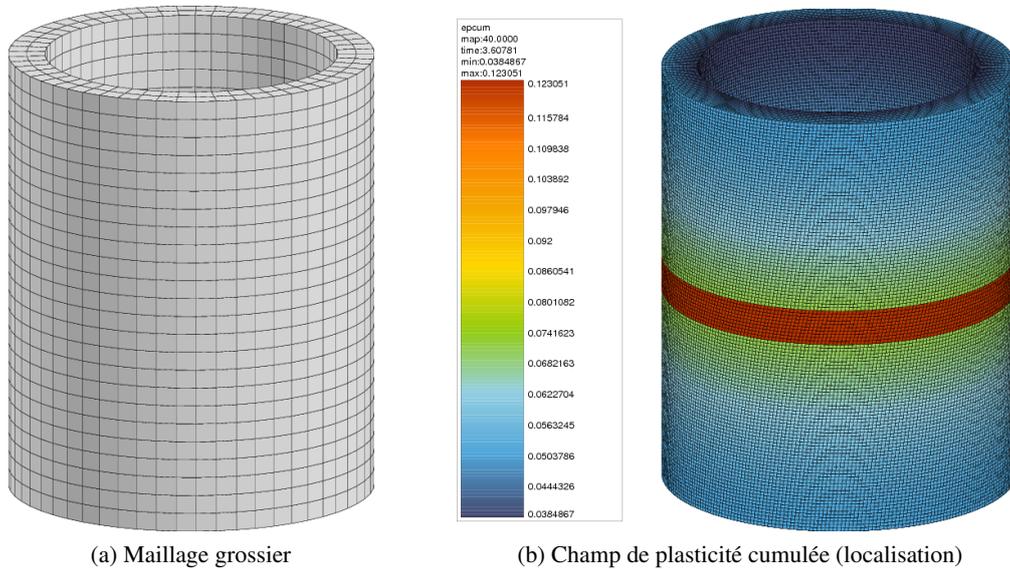
solveur	Temps solveur cumulé (s)	Temps Execution	#Itération min/max
mumps	7 859	62 327	
GMRes + grossier (1467 dofs)	7 809	62 280.0	73/361
GMRes + grossier (29 892 dofs)	3 654	58 050.0	42/66

TABLE 3 – Temps de résolution pour le cas-test de l'éprouvette a perforation conique

On constate que les temps de solveur cumulé sont quasi-identiques entre MUMPS et le GMRes grille grossière pour le problème grossier de 7 809 ddl. Cela peut au premier abord sembler être décevant il faut néanmoins pondérer cela par le fait que même si le temps de résolution est identique l'occupation mémoire du calcul n'est en rien comparable. De plus on constate que si l'on augmente la taille du problème grossier (29 892 ddl) le temps de résolution est quasiment divisé par 2 tout en conservant une occupation mémoire quasiment négligeable vis-à-vis de celle engendré par la résolution direct avec MUMPS. Pour finir nous pouvons évoquer le fait que dans ce cas le temps de résolution du système linéaire ne représente environ que 10% du temps d'exécution du calcul. Cette sous-représentation du solveur linéaire provient du fait que la loi de comportement utilisée est d'une complexité et d'une non-linéarité telle que son intégration représente une grande part du temps d'exécution.

### 3.5 Comportement élasto-plastique en grande déformations

Pour finir le dernier cas-test présenté est le calcul d'un tube en torsion. La formulation employée pour ce calcul est celle des grandes déformations et la loi de comportement est une loi élasto-plastique à écrouissage isotrope non-linéaire. La discrétisation initiale du tube conduit à un problème de 4 408 000 degrés de liberté. Le maillage grossier considéré, représenté sur la Figure 5a, mène quant à lui à un problème de 12 825 degrés de liberté. On présente sur la Figure 5b le champ de plasticité cumulée obtenu à la fin du calcul. On voit clairement apparaître sur ce dernier une bande de localisation. Ce problème est résolu en 40 incréments de chargement. Il n'a malheureusement pas été possible sur ce cas-test de comparer les temps de résolution. En effet la résolution avec le solveur direct MUMPS nécessite plus de 256Go de RAM et n'est donc pas possible sur les machines que nous avons à disposition. En revanche en utilisant le solveur GMRes avec le préconditionneur grille grossière le calcul a pu être réalisé en deux jours en occupant uniquement 30 Go de mémoire vive.



## 4 Synthèse

Nous avons dans ce travail proposé un nouveau préconditionneur dédié à la résolution par solveur itératif de problèmes faisant intervenir des lois de comportement fortement non-linéaires. Ce préconditionneur est basé sur l'introduction d'une discrétisation grossière du problème. Le préconditionneur grille grossière a été implémenté dans le code de calcul éléments finis *Z-set/Zébulon*. Cette implémentation nous a permis de mettre à l'épreuve le préconditionneur sur un certain nombre de cas-tests représentatif. Nous avons alors pu observer que l'utilisation de ce préconditionnement permet la résolution à moindre coût d'une vaste gamme de problèmes. Les prochains axes de travail sont l'amélioration des performances en parallèles (multi-thread et MPI).

## Références

- [1] P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent. *Multifrontal parallel distributed symmetric and unsymmetric solvers*, Computer Methods in Applied Mechanics and Engineering, 501-520, 2000.
- [2] P. Gosselet, C. Rey. *Non-overlapping domain decomposition methods in structural mechanics*, Archives of Computational Methods in Engineering, 515, 2006.
- [3] A. Parret-Fréaud and B. Marchand and C. Bovet, P. Gosselet, N. Spillane, C. Rey, F. Feyel. *Méthodes de décomposition de domaine robustes appliquées au calcul en régime linéaire et non linéaire de structures industrielles de grande taille*, 13eme colloque national en calcul des structures, Giens (France), 2017.
- [4] Z-set Material & Structure analysis suite, [www.zset-software.com](http://www.zset-software.com).
- [5] Y. Saad, M.H. Schultz. *GMRES : A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal Sci. and Stat. Comput., 856-869, 1986.
- [6] Y. Sadd. *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, 2003.
- [7] C. Bovet, P. Gosselet, N. Spillane. *Multipreconditioning for nonsymmetric problems : the case of orthomin and biCG*, Comptes rendus de l'Académie des sciences, 354-358, 2017.
- [8] Wesseling P. *An introduction to Multigrid Methods*, John Wiley & Sons, Chichester, 1992.
- [9] F. Vi, K. Mocellin, E. Perchat, H. Dignonnet, L. Fourment. *Méthode de préconditionnement multigrille hybride et parallèle pour l'accélération des calculs en mise en forme des matériaux*, 13eme colloque national en calcul des structures, Giens (France), 2017.
- [10] M. Benzi, *Preconditioning techniques for large linear systems : A survey*, J. Comput. Phys., 418-477, 2002.