



HAL
open science

Application d une méthode de Newton-Krylov en mécanique des solide

Nicolas Tardieu, Etienne Cheignon

► **To cite this version:**

Nicolas Tardieu, Etienne Cheignon. Application d une méthode de Newton-Krylov en mécanique des solide. 10e colloque national en calcul des structures, May 2011, Giens, France. pp.Clé USB. hal-00592743

HAL Id: hal-00592743

<https://hal.science/hal-00592743>

Submitted on 3 May 2011

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.

Application d'une méthode de Newton-Krylov en mécanique des solides

N. Tardieu¹, E. Cheignon²

¹ LAMSID, EDF-R&D, 1 av du Gal de Gaulle, 92141 Clamart Cedex, France, nicolas.tardieu@edf.fr

² IRMAR-INSA, 263 avenue du Général Leclerc, Rennes Cedex, France, etienne.cheignon@insa-rennes.fr

Résumé — Dans cette communication est présentée une méthode de Newton-Krylov qui appartient à la famille des méthodes de Newton inexactes. Celles-ci associent des méthodes itératives de résolution de systèmes linéaires, que sont les méthodes de Krylov [1] (gradient conjugué, GMRES,...), à la méthode de Newton, là où, de manière traditionnelle en mécanique des solides, on utilise des méthodes directes. Son efficacité est illustrée sur un problème non-linéaire industriel.

Mots clés — méthode de Newton, solveur itératif, préconditionnement.

1 Méthode de Newton

Cette section rappelle les éléments essentiels de la méthode de Newton classique avant de présenter le principe et les intérêts des méthodes de Newton inexactes. On trouvera dans [2] des informations plus détaillées sur les méthodes de Newton.

1.1 Principe général

La méthode de Newton est une méthode de résolution de systèmes non linéaires, mis sous la forme suivante :

$$f(\bar{x}) = 0 \quad (1)$$

Soient U un ouvert de \mathbb{R}^n , $f \in C^1(U, \mathbb{R}^n)$ et $\bar{x} \in U$ tel que $f(\bar{x}) = 0$. À partir d'une donnée initiale $x_0 \in U$, la méthode de Newton construit une suite $(x^{(k)})_{k \in \mathbb{N}} \subset U$ d'approximations de \bar{x} . En supposant que deux approximations successives sont assez proches l'une de l'autre, le développement de Taylor de la fonction f à l'ordre 1 en $x^{(k+1)}$ donne :

$$f(x^{(k+1)}) = f(x^{(k)}) + \nabla f(x^{(k)})(x^{(k+1)} - x^{(k)}) + o(x^{(k+1)} - x^{(k)}) \quad (2)$$

On considère ensuite que $f(x^{(k+1)}) \simeq 0$, ce qui permet de définir la suite $(x^{(k)})_{k \in \mathbb{N}}$ comme suit :

$$\begin{cases} x^{(0)} = x_0 \\ f(x^{(k)}) + \nabla f(x^{(k)})(x^{(k+1)} - x^{(k)}) = 0 \end{cases} \quad (3)$$

où

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(x) & \cdots & \frac{\partial f_n}{\partial x_n}(x) \end{pmatrix}$$

La construction de la suite s'arrête à l'itération p pour laquelle le critère de convergence est atteint, c'est à dire, pour une précision relative ε donnée :

$$\|f(x^{(p)})\| \leq \varepsilon \|f(x^{(0)})\| \quad (4)$$

À chaque itération k , l'algorithme doit donc effectuer les étapes suivantes :

1. calculer la matrice $A = \nabla f(x^{(k)})$
2. résoudre le système $As^{(k)} = -f(x^{(k)})$, $x^{(k+1)} = x^{(k)} + s^{(k)}$

3. évaluer $f(x^{(k+1)})$

Cependant rien n'assure que $\nabla f(x^{(k)})$ soit inversible ni que la suite des $x^{(k)}$ converge vers \bar{x} . Par contre, dans un voisinage de \bar{x} et sous certaines conditions de régularité, on dispose de résultats de convergence locale d'ordre 2 [2].

1.2 Recherche linéaire

Dans la pratique, avoir une convergence locale ne suffit pas, puisqu'on ne sait pas à l'avance si x_0 est suffisamment proche de \bar{x} . Pour avoir une convergence plus sûre, il existe des méthodes de globalisation, telle que la recherche linéaire.

La recherche linéaire a pour but de mieux assurer la convergence de la méthode de Newton quand l'itéré initial n'est pas assez près de la solution. Après la résolution du système linéaire, on a déterminé $s^{(k)}$. La recherche linéaire consiste à poser $x^{(k+1)} = x^{(k)} + \theta s^{(k)}$ avec $\theta \in \mathbb{R}_*^+$ et déterminer la valeur θ qui minimise f sur ce segment. Pour effectuer cette minimisation on pose :

$$\phi_k(\theta) = f(x^{(k)} + \theta s^{(k)}).$$

Puis on effectue une recherche de zéro sur la fonction ϕ_k' . Pour faire cette recherche, on pourrait utiliser encore une fois la méthode de Newton, dans la pratique on utilise plutôt la méthode de la corde ou de la sécante.

2 Méthodes de Newton inexactes

Les méthodes de Newton inexactes ont pour but de gagner du temps de calcul, donc de minimiser le nombre d'opérations. Pour cela on peut chercher à diminuer le coût de la construction de la matrice jacobienne ou le coût de la résolution du système linéaire. C'est cette deuxième option qu'utilisent les méthodes inexactes [2].

2.1 Principe

Le principe est le même que pour la méthode classique, l'unique différence se situe lors de la résolution du système linéaire où le vecteur $s^{(k)}$ est seulement une approximation de la solution du système linéaire à résoudre. Ce vecteur est une solution de la condition de Newton inexacte :

$$\|\nabla f(x^{(k)})s^{(k)} + f(x^{(k)})\| \leq \eta_k \|f(x^{(k)})\| \quad (5)$$

La valeur η_k , appelée "forcing term", peut varier à chaque itération. Plus cette valeur est petite, plus on se rapproche de la méthode classique, mais moins on gagnera de temps à la résolution du système linéaire. Cependant si cette valeur est trop grande, on risque de faire plus d'itérations de Newton. Il faut donc trouver un bon compromis entre résoudre rapidement les systèmes linéaires et ne pas effectuer trop d'itérations de Newton en plus de ce qui serait fait par la méthode classique.

2.2 Convergence

Théorème 1 (Convergence locale des méthodes de Newton inexactes) Soit $f : U \subset \mathbb{R}^n \mapsto \mathbb{R}^n$ telle que :

1. l'équation (1) a une solution \bar{x} ,
2. $\nabla f : U \mapsto \mathbb{R}^{n \times n}$ est Lipschitzienne dans un voisinage de \bar{x} ,
3. $\nabla f(\bar{x})$ est inversible,

alors il existe deux réels positifs δ et $\bar{\eta}$, avec $\bar{\eta} < 1$, tels que $\forall x_0 \in B(\bar{x}, \delta)$ et $\{\eta_k\} \subset [0, \bar{\eta}]$, la suite $(x^{(k)})_{k \in \mathbb{N}}$ définie par

$$x^{(k+1)} = x^{(k)} + s^{(k)}$$

et par la condition (5) converge linéairement.
Si de plus, $\eta_k \rightarrow 0$, la convergence est superlinéaire.

On trouvera la démonstration dans [4].

2.3 Méthodes de Newton-Krylov

Les méthodes de Newton-Krylov utilisent le caractère itératif des méthodes de Krylov [1] pour résoudre avec la précision souhaitée le système linéaire à chaque itération. La condition de Newton inexacte (5) va donc servir de test d'arrêt pour la résolution par méthode de Krylov. On cherche à adapter la précision de la résolution à chaque itération de Newton de façon à faire le moins d'itérations de Krylov possible, tout en gardant une bonne convergence de la méthode de Newton.

2.3.1 Choix de la valeur d'arrêt ou "forcing term"

Pour s'approcher au mieux de cet objectif, il existe des schémas pour calculer la suite des η_k à partir d'un η_0 . L'idée principale est de dire que quand on est loin de la solution, il n'est pas nécessaire d'être très précis, mais plus on s'approche de la solution plus on veut l'être pour garder une convergence quadratique. On trouve dans [2] et dans [3] le schéma suivant :

$$\eta_{k+1}^{Res} = \gamma \frac{\|f(x^{(k)})\|^2}{\|f(x^{(k-1)})\|^2} \quad (6)$$

où $\gamma \in (0, 1]$ est un paramètre fixé par l'utilisateur. Cependant pour éviter que la suite des η_k décroisse trop vite et que l'on fasse des résolutions trop précises ("oversolving"), ou que $\eta_k > 1$, le schéma complet est :

$$\eta_{k+1} = \min \left(\eta_{max}, \max \left(\eta_{k+1}^{Safe}, 0.5\epsilon \frac{\|f(x^{(k)})\|^2}{\|f(x^{(k-1)})\|^2} \right) \right) \quad (7)$$

où ϵ est la précision de l'équation (4) et η_k^{Safe} est défini comme suit :

$$\eta_{k+1}^{Safe} = \begin{cases} \eta_{max} & k = 0 \\ \min(\eta_{max}, \eta_{k+1}^{Res}) & k > 0, \quad \gamma \eta_k^2 \leq 0.1 \\ \min(\eta_{max}, \max(\eta_{k+1}^{Res}, \gamma \eta_k^2)) & k > 0, \quad \gamma \eta_k^2 > 0.1 \end{cases}$$

2.3.2 Oversolving

Dans le contexte des méthode de Newton-Krylov, faire de l'*oversolving*, c'est résoudre précisément les systèmes linéaires inutilement, c'est-à-dire sans que $\|f(x^{(k)})\|$ ne décroisse suffisamment par rapport à la précision du calcul. Il arrive en effet dans de nombreux cas que l'on obtienne la même décroissance du résidu non linéaire avec une résolution du système linéaire moins précise. Dans ce cas on a effectué inutilement de nombreuses itérations de Krylov.

Le but des méthodes de Newton-Krylov est d'adapter toujours au mieux la précision η_k demandée pour limiter au mieux les itérations de Krylov, en faisant le moins possible augmenter le nombre d'itérations de Newton.

Pour illustrer ce phénomène, nous avons réalisé un pas de chargement du calcul présenté au §5.1 de deux manières : d'abord on utilise la méthode de Newton avec un gradient conjugué, la précision relative des résolutions des systèmes linéaires sera constante et égale à 10^{-6} , la valeur par défaut ; ensuite, on utilise la même méthode mais où la précision varie selon le schéma (7).

Pour bien comprendre les résultats qui suivent, on a besoin de clarifier la différence entre itérations de Newton et itérations de Krylov. La méthode de Newton construit une suite $(x^{(k)})_{k \geq 0}$ d'approximation

de \bar{x} . À chaque itération $k + 1$, une méthode de Krylov construit la suite $(s_i^{(k)})_{i \geq 0}$ d'approximation de s solution de $\nabla f(x^{(k)})s = -f(x^{(k)})$. On note alors $x_i^{(k)} = x^{(k)} + s_i^{(k)}$. Les itérations de Krylov s'arrêtent quand on atteint le critère de convergence (5). Si on appelle p , le numéro de l'itération où le critère de convergence est satisfait, on a $x_p^{(k)} = x^{(k+1)}$.

Sur la figure 1, les étoiles représentent les valeurs successives des résidus non-linéaires $f(x^{(k)})$. En traits pleins, sont représentées les valeurs du résidu linéaire $\nabla f(x^{(k)})s_i^{(k)} + f(x^{(k)})$ et en pointillés une estimation grossière de $f(x_i^{(k)})$. On voit très clairement sur la figure de gauche que la grande précision de résolution est inutile puisque le résidu non linéaire ne décroît pas beaucoup plus que sur la figure de droite. Le tableau suivant résume les nombres d'itérations effectuées dans chaque cas.

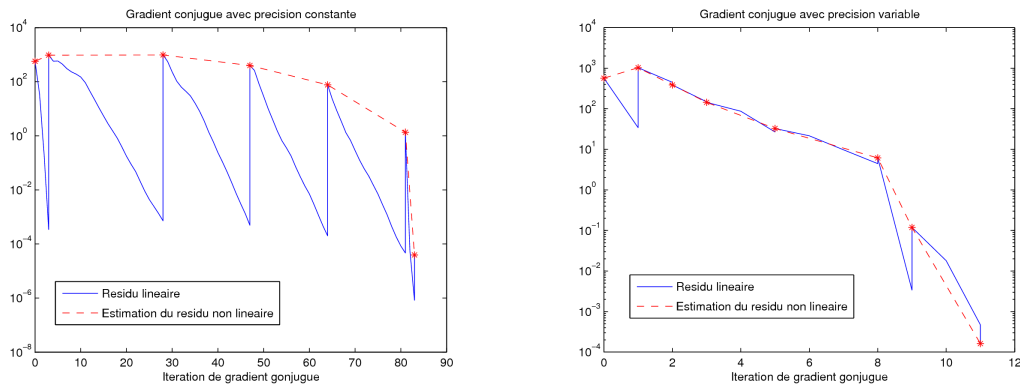


FIG. 1 – Illustration de l'oversolving

	GCPC	GCPC + Forcing term
Nb d'itérations de Newton	6	7
Nb d'itérations de GCPC	83	11
Tps de résolution	284.03 s	255.07 s
Gain de tps	/	10%

On voit donc clairement que le schéma utilisé a très bien annulé l'oversolving, on a ainsi réduit considérablement le nombre d'itérations de Krylov en ne concédant qu'une seule itération de Newton. Le temps de calcul a été lui aussi amélioré, étant donné le nombre d'itérations gagnées on aurait pu s'attendre à mieux, cela est sûrement dû au coût de l'itération de Newton supplémentaire.

3 Implantation, utilisation et résultats

4 Utilisation

Dans l'optique de l'utilisation de la méthode de Newton-Krylov, il faut commencer par choisir un préconditionneur et une méthode itérative. En fonction de la symétrie de la matrice du système, notre choix de méthode itérative s'est porté vers le Gradient Conjugué ou GMRes. En matière de préconditionneur, nous avons utilisé une méthode développée pour *Code_Aster* [5], appelée LDLT_SP [6], basée sur une factorisation simple précision de la matrice du système linéaire. Si on réévalue le préconditionneur à chaque résolution de système linéaire, cette approche est coûteuse en mémoire et en temps CPU. Mais cela n'est pas nécessaire : ce préconditionneur est de très bonne qualité et peut être gardé constant pendant plusieurs résolutions. Il n'est réévalué que lorsque son efficacité décroît. De fait, on dispose ainsi d'un préconditionneur extrêmement robuste.

Dans le même objectif de robustesse, nous avons adapté le schéma de calcul du "forcing term" pour assurer une convergence plus sûre, certainement au prix d'un peu plus d'oversolving. On a donc défini

de manière empirique le schéma suivant :

$$\eta_{k+1} = \begin{cases} \eta_0 & k = 0 \\ \max(\min(\eta_k/2, \eta_{k+1}^{Res}), \eta_{min}) & k > 0, \quad \gamma_k^2 \leq 0.1 \\ \min(\eta_k/2, \max(\eta_{k+1}^{Res}, \gamma_k^2)) & k > 0, \quad \gamma_k^2 > 0.1 \end{cases} \quad (8)$$

Enfin, on choisit les paramètres $\gamma = 0.9$ et $\eta_0 = 0.9$, comme dans [2] et [3], et $\eta_{min} = 10^{-4}$. C'est avec toutes les caractéristiques que l'on vient de décrire que les tests qui suivent ont été effectués.

5 Résultats

Cette section présente les tests de la méthode de Newton-Krylov que l'on a décrite précédemment sur des problèmes de différentes natures. Ils ont été faits dans le but d'évaluer les performances de cette méthode numérique et également pour tester sa robustesse.

5.1 Groupe Sécheur-Surchauffeur (GSS)

Ce calcul est un problème de mécanique des solides ; il modélise un organe de centrale nucléaire appelé Groupe Sécheur-Surchauffeur, dont la fonction est d'améliorer la qualité de la vapeur d'eau du circuit secondaire avant réinjection. Le matériau obéit à une loi de comportement élasto-plastique avec écrouissage isotrope non-linéaire et une cinématique grandes déformation est utilisée. Le maillage est tétraédrique d'ordre 2 et engendre un problème à 1 084 773 inconnues (figure 2). Les matrices issues de ce problème sont toutes symétriques.

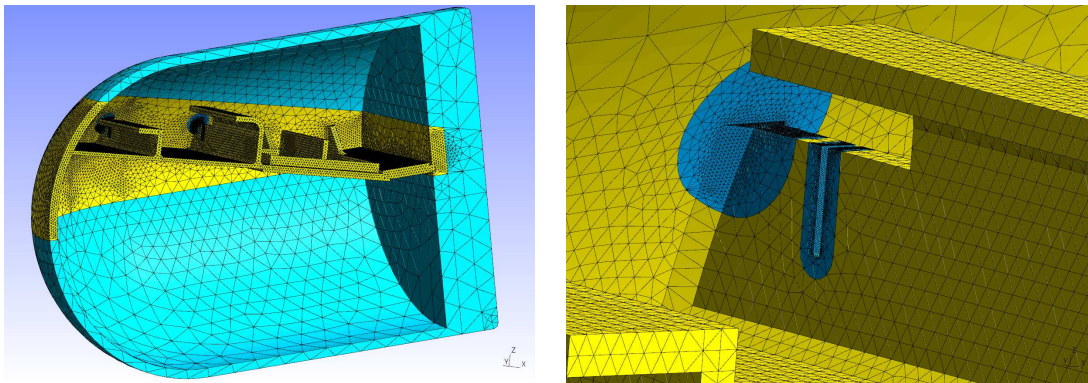


FIG. 2 – Maillage du Groupe Sécheur-Surchauffeur

Pour réaliser des comparaisons, nous avons d'abord lancé le calcul avec la méthode de Newton classique, en utilisant le solveur direct MUMPS [8] comme solveur linéaire, puis le gradient conjugué de la bibliothèque PETSc [7] préconditionné par LDLT_SP et enfin avec la méthode de Newton-Krylov en utilisant le gradient conjugué de PETSc avec LDLT_SP. Le tableau qui suit présente les différents temps de calcul et le gain par rapport au calcul avec MUMPS. Les figures 3 et 4 indiquent respectivement le nombre d'itérations de Newton effectuées pour les trois calculs et le nombre d'itérations de gradient conjugué pour les deux derniers calculs, pour chaque pas de temps.

Solveur linéaire	Temps de résolution du problème		
	MUMPS	PETSc+LDLT_SP	PETSc+LDLT_SP+Forcing term
Tps de résolution	9176 s	8530 s	5554 s
Gain tps CPU / MUMPS	/	7%	39%

On remarque que le temps gagné par le solveur itératif n'est pas très important par rapport au temps gagné par Newton-Krylov. Ceci est sûrement dû à la plus grande non-linéarité du problème qui permet au

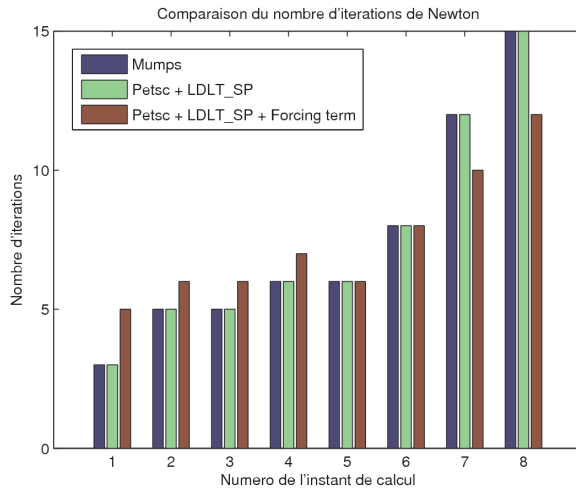


FIG. 3 – Comparaison du nombre d'itérations de Newton

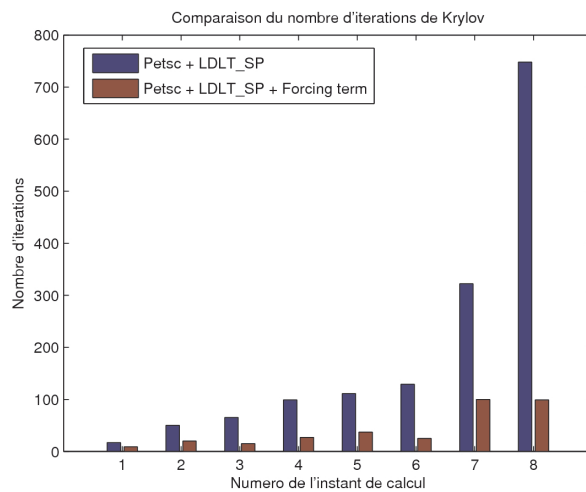


FIG. 4 – Comparaison du nombre d'itérations de Krylov

caractère inexact de s'exprimer plus fortement : plus on avance dans le temps, plus le nombre d'itérations de Newton est grand (figure 3), ce qui convient très bien à cette méthode. Sur la figure 3, on remarque que la méthode inexacte fait plus d'itérations de Newton sur les premiers pas de temps, mais à partir du pas numéro 5 le nombre d'itérations est égal à celui des deux autres méthodes et devient même inférieur à partir du pas numéro 7. Il s'agit d'une situation décrite dans la littérature [3], due au fait qu'une résolution précise du système linéarisé loin de l'équilibre peut-être contre-productive. Le nombre d'itérations de Krylov gagnées s'en trouve alors accru (voir la figure 4).

Voici d'autres informations sur les calculs données par *Code_Aster* :

MUMPS :

STATISTIQUES SUR LE TRANSITOIRE

NOMBRE DE PAS DE TEMPS	:	8
NOMBRE D'ITERATIONS DE NEWTON	:	60
NOMBRE D'ITERATIONS DE RECH.LINE	:	48
NOMBRE DE CREATION DE NUMEROTATION	:	1
NOMBRE DE FACTORISATION DE MATRICE	:	60
NOMBRE D'INTEGRATION DE COMPORTEMENT	:	168
NOMBRE DE RESOLUTION K.U = F	:	60
TEMPS POUR CREATION NUMEROTATION	:	41 s
TEMPS POUR FACTORISATION MATRICE	:	1 h 46 m 1 s
TEMPS POUR INTEGRATION COMPORTEMENT	:	19 m 9 s

TEMPS POUR RESOLUTION K.U = F : 7 m 10 s

Gradient conjugué :

STATISTIQUES SUR LE TRANSITOIRE

NOMBRE DE PAS DE TEMPS : 8
NOMBRE D'ITERATIONS DE NEWTON : 60
NOMBRE D'ITERATIONS DE RECH.LINE : 48
NOMBRE DE CREATION DE NUMEROTATION : 1
NOMBRE DE FACTORISATION DE MATRICE : 60
NOMBRE D'INTEGRATION DE COMPORTEMENT : 168
NOMBRE DE RESOLUTION K.U = F : 60
TEMPS POUR CREATION NUMEROTATION : 41 s
TEMPS POUR FACTORISATION MATRICE : 38 m 43 s
TEMPS POUR INTEGRATION COMPORTEMENT : 18 m 52 s
TEMPS POUR RESOLUTION K.U = F : 1 h 6 m 47 s

Gradient conjugué + FORCING TERM :

STATISTIQUES SUR LE TRANSITOIRE

NOMBRE DE PAS DE TEMPS : 8
NOMBRE D'ITERATIONS DE NEWTON : 60
NOMBRE D'ITERATIONS DE RECH.LINE : 26
NOMBRE DE CREATION DE NUMEROTATION : 1
NOMBRE DE FACTORISATION DE MATRICE : 60
NOMBRE D'INTEGRATION DE COMPORTEMENT : 146
NOMBRE DE RESOLUTION K.U = F : 60
TEMPS POUR CREATION NUMEROTATION : 41 s
TEMPS POUR FACTORISATION MATRICE : 38 m 55 s
TEMPS POUR INTEGRATION COMPORTEMENT : 16 m 44 s
TEMPS POUR RESOLUTION K.U = F : 19 m 7 s

En comparant ces statistiques, on perçoit les avantages de l'utilisation d'une méthode directe ou itérative pour la méthode de Newton classique :

- le coût des méthodes directes est lié à la factorisation de la matrice, la résolution à proprement parler (descente-remontée) étant très rapide
- le coût des méthodes itératives est lié à la résolution du système linéaire, la factorisation (ici il s'agit de la phase de préconditionnement) étant assez rapide

Grâce à l'utilisation de la méthode de Newton-Krylov, on constate que :

- on conserve le faible coût de la phase de préconditionnement
- on parvient de plus à limiter le temps de résolution

C'est grâce à ces deux propriétés que la méthode de Newton-Krylov permet d'obtenir de bonnes performances sur des problèmes de grande taille.

6 Conclusions

Cette communication présente une application industrielle d'une méthode de Newton inexacte sur un problème non-linéaire de mécanique des solides. Pour mener à bien cette étude, nous avons proposé un calcul particulier du "forcing term" ainsi qu'une méthode de préconditionnement particulière. Ainsi équipée, notre méthode de Newton-Krylov s'est montrée à la fois performante et robuste sur le problème étudié qui est très non-linéaire. D'autres calculs non-linéaires, notamment de thermo-hydro-mécanique, ont été menés à bien avec cette méthode.

Cette méthode permet de gagner du temps seulement sur les phases de préconditionnement / factorisation (par rapport à une méthode classique avec solveur direct) et de résolution des systèmes linéaires (par rapport à une méthode classique avec solveur itératif). Ainsi, si la majorité du temps du calcul est passée dans l'intégration de la loi de comportement, on ne pourra espérer qu'un faible gain de temps. De plus, cette méthode effectuée dans la majorité des cas plus d'itérations de Newton et donc plus d'intégrations de la loi de comportement. Si ce coût est trop important, il annulera et même dépassera le temps que l'on espère gagner sur la factorisation et la résolution. Cette méthode de Newton-Krylov est donc très efficace pour des problèmes où la majeure partie du temps est passée dans la résolution des systèmes

linéarisés. Elle est donc très adaptée aux problèmes de grande taille.

Références

- [1] Yousef Saad. *Iterative Methods for Sparse Linear Systems*, Second Edition 2000.
- [2] C.T. Kelley, *Solving Nonlinear Equations with Newton's Method*, Siam, 2003.
- [3] Stanley C. Eisenstat et Homer F. Walker, *Choosing the Forcing Terms in an Inexact Newton Method*, Siam J. Sci. Comput., 17 (1996).
- [4] C.T. Kelley, *Iterative Methods for Solving Linear and Nonlinear Equations*, Vol. 16 in *Frontiers in Applied Mathematics*, Siam, Philadelphia, 1995.
- [5] Site de *Code_Aster*, <http://www.code-aster.org>
- [6] Documentation *Code_Aster*, *Généralités sur le gradient conjugué : GCPC Aster et utilisation de PETSc*, (R6.01.02).
- [7] PETSc Web page, <http://www.mcs.anl.gov/PETSc>, Satish Balay and Kris Buschelman and William D. Gropp and Dinesh Kaushik and Matthew G. Knepley and Lois Curfman McInnes and Barry F. Smith and Hong Zhang, 2010
- [8] P. R. Amestoy and A. Guermouche and J.-Y. L'Excellent and S. Pralet, *Hybrid scheduling for the parallel solution of linear systems*. *Parallel Computing* Vol 32 (2), pp 136-156 (2006)