Implémentation de la méthode asymptotique numérique dans CAST3M

Loïc Daridon* — **Thierry Charras****

* *Institut de Mécanique des Fluides et des Solides, U.M.R. 7507*
2, Rue Boussingault 67000 Strasbourg
loic.daridon@ipst-ulp.u-strasbg.fr

** *CEA/LM2S*
CEN Saclay
91191 Gif/yvette cedex
chat@semt2.smts.cea.fr

RÉSUMÉ. *L'objectif de cet article est de montrer comment il est possible d'implémenter la méthode asymptotique numérique (MAN) dans le cadre d'une programmation objet. Pour ce faire, nous avons choisi d'utiliser le code de calcul par élément finis CAST3M développé par le C.E.A. Premièrement, nous présentons l'implémentation de la MAN, dans le cas de l'élasticité non linéaire géométrique, en utilisant uniquement les opérateurs standard de CAST3M. Nous réalisons également une comparaison avec une approche classique utilisant des opérateurs spécifiques. Nous présentons des applications avec un grand nombre de degrés de liberté, 180 000 ddl et 300 000 ddl. Ces exemples nous permettront de comparer les temps d'inversions de la matrice de rigidité, du calcul des seconds membres et de la résolution des systèmes linéaires dans le cadre d'une application industrielle avec un solveur direct performant.*

ABSTRACT. *This paper deals with the implementation of the numerical asymptotic method in an industrial finite element code. This code is CAST3M, developed by the C.E.A. We present the implementation of the geometrically non linear elasticity case. We propose two approaches: firstly an updated Lagrangian one in which displacements are referred to the updated configuration and secondly a classical Lagrangian one in which displacements are referred to the undeformed configuration. Then, we present a numerical application with a large number of degrees of freedom, 180 000 dof and 300 000 dof. With this application, we can evaluate various calculation times and the efficiency of the method in an industrial frame work.*

MOTS-CLÉS : *Élasticité non linéaire , MAN , Grand nombre de ddl.*

KEYWORDS: *Non linear elasticity , MAN , Large number of dof.*

REEF - 13/2004. Méthodes asymptotiques numériques, pages 165 à 176

1. Introduction

La méthode asymptotique numérique (MAN) est un algorithme destiné à résoudre les problèmes non linéaires. La robustesse de la méthode pour le suivi des courbes et l'adaptation automatique de la longueur de pas en font une méthode très fiable. Le principe général de la méthode consiste à développer chaque inconnue du problème sous forme de série entière puis de les injecter dans le problème initial. Il en résulte une décomposition du problème non linéaire en n problèmes linéaires dont la résolution récursive est faite par la méthode des éléments finis (MEF). Actuellement, l'implémentation des développements de la MAN est faite dans des codes de calcul par élément finis de laboratoire, tous dérivés du code EVE. Aujourd'hui, il existe plusieurs versions de ce code écrites soit en Fortran 77 [COC 94a], soit en Fortran 90 [BAG 01]. Nous proposons, dans ce travail, de montrer qu'il est possible d'implémenter la MAN dans un code industriel de calcul par éléments finis dont la programmation est orientée objet. Pour cela nous avons choisi d'utiliser CAST3M qui est le code développé au C.E.A. Dans ce code il est possible de programmer soit en ESOPE soit en GIBIANE [VER 93]. Ces deux types de langages représentent deux niveaux de programmation. La programmation en ESOPE est à rapprocher de celle en Fortran 77. Elle permet de définir de nouveaux opérateurs dans CAST3M spécifiques à chaque problème. La programmation en GIBIANE est à rapprocher de celle des langages orientés objet [PAS 97]. Elle permet de définir des procédures. Une procédure en langage GIBIANE est un ensemble d'opérateur agissant sur des objets définis dans CAST3M.

L'objectif de ce travail est de montrer qu'il est possible d'implémenter efficacement la MAN dans CAST3M. La philosophie de CAST3M est de résoudre les problèmes non linéaires en lagrangien actualisé, nous avons donc dû adapter la formulation de la MAN à ce cadre. Dans cette première étude nous avons réalisé l'implémentation dans le cas non linéaire géométrique [COC 94a]. Dans un premier temps, nous avons réalisé l'implémentation suivant deux approches : Lagrangien actualisé ou Lagrangien total. Cela nous a permis de mener une étude comparative sur les temps de calcul, au regard de la philosophie de l'implémentation. La première consiste à implémenter la MAN dans le cadre d'une formulation lagrangienne actualisée en développant une procédure utilisant uniquement des opérateurs standard. La deuxième consiste à implémenter la MAN dans le cadre d'une formulation larangienne totale en développant une procédure utilisant des opérateurs spécifiques. Dans un deuxième temps, nous allons évaluer le coût de calcul d'une série entière dans les conditions d'un calcul industriel. Pour ce faire, nous avons étudié un problème ayant 300 000 degrés de libertés (ddl) à l'aide de la méthode asymptotique numérique implémentée dans un code de calcul industriel ayant un solveur direct et une gestion de la mémoire performants.

2. Implémentation dans CAST3M

Contrairement au code d'éléments finis, EVE, utilisé pour implémenter les développements de la MAN, CAST3M a été développé pour une résolution des problèmes

non linéaires en lagrangien actualisé. Nous allons expliciter ici les simplifications que ce type de résolution induit dans le traitement de la non linéarité géométrique. En reprenant les notations classiques [BAG 01], le problème non linéaire peut être écrit sous la forme[1] :

$$F(U,\lambda) = \begin{cases} \int_\Omega \underline{\underline{S}} : \delta\underline{\underline{\gamma}}(U,\delta U)\, d\omega &= \lambda \quad P_e(\delta U) \\ \underline{\underline{S}} &= D : \underline{\underline{\gamma}}(U) \end{cases} \qquad [1]$$

où U est le champ de déplacement, λ le paramètre de charge, $\underline{\underline{S}}$ le tenseur de contrainte de Piola-Kirchhoff du second ordre, P_e les efforts extérieurs et $\underline{\underline{\gamma}}$ le tenseur des déformations de Green-Lagrange. Nous pouvons définir ce tenseur $\underline{\underline{\gamma}}$ par [2] :

$$\underline{\underline{\gamma}}(U,U) = \frac{1}{2}\left(\nabla^t U + \nabla U\right) + \frac{1}{2}\nabla^t U.\nabla U \qquad [2]$$

En supposant que le problème soit résolu pour un point (U_0, S_0, λ_0) et en désignant par Ω_0 le domaine déformé correspondant, nous cherchons la solution sous la forme de séries entières développées par rapport à un paramètre de pilotage a [COC 94a][3] :

$$\begin{cases} S &= S_0 + aS_1 + a^2S_2 + a^3S_3 + \ldots \\ U &= U_0 + aU_1 + a^2U_2 + a^3U_3 + \ldots \\ \lambda &= \lambda_0 + a\lambda_1 + a^2\lambda_2 + a^3\lambda_3 + \ldots \end{cases} \qquad [3]$$

En introduisant dans le principe des puissances virtuelles les développements en séries [3] et en utilisant les notations matricielles classiques [ZIE 97], les problèmes linéaires à résoudre peuvent se mettre sous la forme suivante [COC 94b][4, 5] :

$$ordre\ \ 1 \begin{cases} \left[\hat{U}\right] &= [K_T]^{-1}[F] \\ \lambda_1 &= \frac{1}{\sqrt{1+[\hat{U}]^t[\hat{U}]}} \\ [U_1] &= \lambda_1.\left[\hat{U}\right] \\ [S_1] &= [D][B(U_0)][U_1] \end{cases} \qquad [4]$$

$$ordre\ \ p \begin{cases} \left[\hat{U}\right] &= [K_T]^{-1}\left[F_p^{nl}\right] \\ \lambda_p &= \lambda_1\left[\hat{U}\right]^t[U_1] \\ [U_p] &= \frac{\lambda_p}{\lambda_1}[U_1] + \left[\hat{U}\right] \\ [S_p] &= [D][B(U_0)][U_P] + \left[S_P^{nl}\right] \end{cases} \qquad [5]$$

Pour résoudre le problème [1] il faut donc calculer $\left[F_p^{nl}\right], \left[S_p^{nl}\right]$ et $[K_T]^{-1}$. Nous proposons ici dans les chapitres 2.2 et 2.1 deux méthodes pour déterminer $\left[F_p^{nl}\right]$ [10], $\left[S_p^{nl}\right]$[9]. L'opérateur tangent $[K_T]$ est donné par [6] :

$$\begin{array}{rcl} [K_T] & = & [K_{rigi}] + [K_\sigma] \\ & = & \int_{\Omega_0} [B\,(U_0)]^t\,[D]\,[B\,(U_0)]\;\; d\omega + \int_{\Omega_0} [G]^t\,[S_0]\,[G] d\omega \end{array} \qquad [6]$$

Cette écriture du problème à l'ordre p [5] reste vraie en formulation lagrangienne actualisée ou totale. La seule différence réside dans le calcul de $[B\,(U_0)]$ dont l'expression générale est donnée par [7] :

$$[B\,(U_0)] = [B_l] + [A\,(U_0)]\,.\,[G] \qquad [7]$$

où $[B_l]$ est le terme linéaire du tenseur des déformations et le produit $[A\,(U_0)]\,.\,[G]$ la partie non linéaire [ZIE 97]. Dans la configuration déformée Ω_0, le terme $[A\,(U_0)]$ s'annule et donc $[B\,(U_0)]$ est réduit à la partie linéaire du tenseur des déformations :

$$[B\,(U_0)] = [B_l] \qquad [8]$$

Cette simplification a des implications sur le calcul de l'opérateur tangent, du tenseur de contrainte et des seconds membres.

2.1. *Programmation en GIBIANE uniquement*

Pour l'implémentation de la MAN, dans le cas de la non linéarité géométrique, en langage GIBIANE, nous avons conservé le cadre du lagrangien actualisé (méthode I). Cette optique permet de conserver l'esprit de programmation du code CAST3M et d'utiliser uniquement des opérateurs standard. Nous allons montrer qu'il est possible de traduire le système d'équation [5] à l'aide d'opérateurs existants dans CAST3M.

L'opérateur $[K_T]$ est l'opérateur tangent classique utilisé dans les méthodes prédiction correction de type Newton-Raphson. Il est construit à l'aide de deux opérateurs : RIGI et KSIGMA. L'opérateur RIGI permet de calculer la matrice de rigidité géométrique linéaire dans une configuration déformée ou non. L'opérateur KSIGMA calcule la matrice de raideur géométrique associée à un champ de contrainte.

Le champ de contrainte $\left[S_p^{nl}\right]$ peut être déterminé en utilisant deux opérateurs : GRAD et ELAS. En effet, en conservant la partie non linéaire du tenseur des déformations sous la forme [2], il est alors clair que ce champ de déformation peut être obtenu par le simple produit scalaire de deux vecteurs gradients [9].

$$\left[S_p^{nl}\right] = [D] \sum_{r=1}^{p-1} \frac{1}{2} \left[A\left(U_r\right)\right] . [G] \left[U_{p-r}\right] = [D] \sum_{r=1}^{p-1} \frac{1}{2} \nabla^t U_r . \nabla U_{p-r} \qquad [9]$$

L'opérateur GRAD calcule les gradients d'un champ de déplacement. L'opérateur ELAS calcule des contraintes à partir de déformations dans l'hypothèse d'un matériau élastique. En combinant ces deux opérateurs il est donc possible de calculer $\left[S_p^{nl}\right]$.

Pour calculer $\left[F_p^{nl}\right]$, nous allons utiliser deux opérateurs de CAST3M : BSIGMA et KSIGMA. En effet, B. Cochelin [COC 94a] a montré que l'on pouvait écrire le second membre à l'ordre p sous la forme suivante :

$$\begin{aligned} \left[F_p^{nl}\right] = \left[{}_1F_p^{nl}\right] + \left[{}_2F_p^{nl}\right] \quad &= \quad - \int_{\Omega_0} \left[B\left(U_0\right)\right] \left[S_p^{nl}\right] d\omega \\ &\quad - \int_{\Omega_0} \sum_{r=1}^{p-1} \left[A\left(U_{p-r}\right)\right]^t . [G]^t . [S_r] d\omega \end{aligned} \qquad [10]$$

Comme dans une formulation en lagrangien actualisé $[B(U_0)]$ est identique à $[B_l]$, d'après l'équation [8], le premier terme $\left[{}_1F_p^{nl}\right]$ du second membre est obtenu directement à l'aide de l'opérateur BSIGMA qui calcule le champ de forces nodales résultant de l'intégration d'un champ de contraintes $\left[S_p^{nl}\right]$.

$$\left[{}_1F_p^{nl}\right] = \int_{\Omega_0} \left[B_l\right] \left[S_p^{nl}\right] d\omega \qquad [11]$$

Le second terme du second membre, $\left[{}_2F_p^{nl}\right]$ [10] est obtenu à l'aide de l'opérateur KSIGMA. En effet il peut être exprimé comme le produit d'une matrice $[K_\sigma]$ avec un vecteur déplacement [ZIE 97].

$$\left[{}_2F_p^{nl}\right] = \int_{\Omega_0} \sum_{r=1}^{p-1} \left[A\left(U_{p-r}\right)\right]^t [G]^t [S_r] d\omega = \sum_{r=1}^{p-1} \left[K_{S_r}\right] \left[U_{p-r}\right] \qquad [12]$$

Comme nous utilisons uniquement des opérateurs standard de CAST3M, cette implémentation est valable pour toute la bibliothèque d'éléments. Elle peut, également, être transportée et utilisée directement sur toutes les plate-formes et systèmes d'exploitation supportant CAST3M sans avoir à recompiler le code.

Ce type de programmation, orientée objet, met en lumière la nécessité d'identifier, dans les différents développements de la MAN, des opérateurs généraux. Il suffira, alors, pour ceux qui ne seraient en standard dans CAST3M de les y implémenter. Ensuite, il sera possible de les utiliser dans une procédure écrite en GIBIANE.

2.2. *Programmation en ESOPE*

Nous présentons brièvement maintenant l'implémentation de la MAN, dans le cas de la non linéarité géométrique, en langage ESOPE (méthode II). Cette approche est du même type que celle actuellement utilisée pour développer dans EVE. Nous avons utilisé le cadre du lagrangien total. Pour cela, nous avons dû développer des opérateurs spécifiques à la MAN et en adapter d'autres pour le cadre du lagrangien total. Cette méthode nécessite une expertise supérieure quant à la programmation dans CAST3M. Tous ces opérateurs développés doivent être écrits pour chaque type d'élément. Actuellement nous les avons développés uniquement pour les éléments massifs 2-D. Bien que plus complexe, le développement d'opérateurs spécifiques peut s'avérer nécessaire dans des problèmes non linéaires plus complexes [CAD 01], [DAY 01], [ELH 98].

3. Comparaison entre une implémentation en GIBIANE et en ESOPE

Nous allons comparer les temps de calculs entre les deux approches. Dans les deux cas nous avons développé une procédure en langage GIBIANE permettant d'utiliser la méthode de continuation utilisant soit les séries entières soit les approximants de Padé [ELH 00].

Nous allons considérer le flambage d'une poutre fissurée. Nous avons utilisé un maillage régulier constitué de 30 000 éléments quadrilatères à 8 nœuds à 16 ddl. Les données géométriques utilisées sont les suivantes $\frac{L}{a} = 4$, $\frac{h_0}{a} = 0,2$,$\frac{h_1}{h_2} = 5$,$E = 1E5Mpa$,$a = 10mm$. Pour des raisons de symétrie, nous avons modélisé uniquement la moitié de la structure (figure 1).

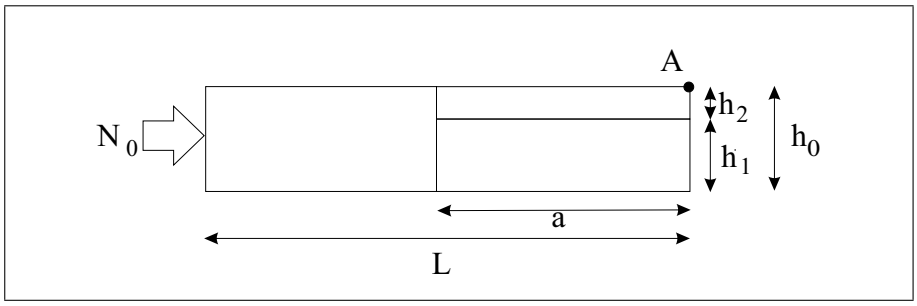

Figure 1. *Poutre fissurée - Conditions aux limites*

Nous présentons dans la figure 2 le déplacement vertical du point A, $V(a)$, en fonction du paramètre de charge, λ . Ce calcul a été réalisé sous un environnement Linux avec un pentium III cadencé à 750 MHz et 256 Mo de mémoire vive. Cette courbe a été obtenue en utilisant la méthode de continuation pour les séries entières. Le rayon de convergence de la série entière est déterminée à l'aide du critère donné par l'équation 13 [COC 94b] :

$$R_{max} = \left(\epsilon \frac{\|U_1\|}{\|U_n\|} \right)^{\left(\frac{1}{n-1}\right)} \quad [13]$$

où ϵ est égal à 10^{-6} et n est l'ordre du développement limité.

La longueur du pas étant directement liée au rayon de convergence de la série, la figure 2 illustre bien que les pas sont plus petits (4 et 5) dans les zones de forte non linéarité. Cette aptitude de la méthode à auto-ajuster la taille des pas est un atout majeur de la MAN. Le choix du paramètre de pilotage a [COC 94b] permet un pilotage automatique de tous les problèmes non linéaires en passant aussi bien les limites en charge ou en déplacement.

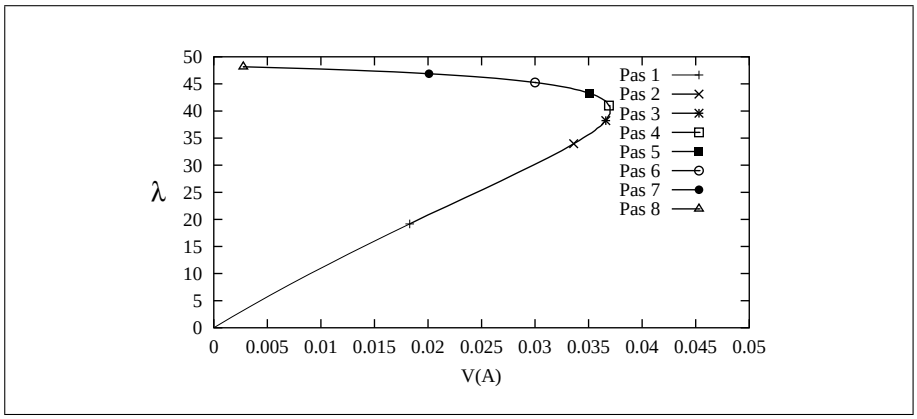

Figure 2. *Déplacement de A vs* λ. $\epsilon = 10^{-6}$, *n=10*

Le tableau 1 donne un récapitulatif des temps de calculs pour les différents membres de l'équation et assemblage des matrices correspondantes. Nous comparons ici ces différents temps à différents ordres. Les résultats sont donnés sous une forme sans dimension où le temps de référence est celui du calcul et de l'assemblage d'un opérateur tangent.

Les résultats du tableau 1 montrent que la programmation en utilisant uniquement des opérateurs standard de CAST3M, méthode I, indiquent des temps de calcul plus importants que ceux obtenus en utilisant des opérateurs spécifiques, méthode II. Cette différence de temps de calcul est plus liée à la gestion spécifique par CAST3M de la place en mémoire centrale qu'à la philosophie de programmation utilisée. En effet, dans le cas de la méthode I, CAST3M à l'ordre 10 fait appel une fois à l'opérateur MENA dans le calcul du champ de contrainte $\left[S_p^{nl}\right]$. Cet opérateur MENA gère l'encombrement de la mémoire centrale au cours du calcul. A l'ordre 25, CAST3M fait appel trois fois à ce même opérateur lors du calcul de $\left[S_p^{nl}\right]$. Cette inconvénient peut être supprimé en utilisant une machine avec une plus grande capacité de mémoire

	Ordre 10		Ordre 25	
	Méthode I	Méthode II	Méthode I	Méthode II
Inversion de $[K_t]$	34.84	34.84	34.84	34.84
Calcul de $[S_p^{nl}]$	16.61	3.07	33.42	8.30
Calcul de $[{}_1F_p^{nl}]$	0.16	0.22	0.17	0.21
Calcul de $[{}_2F_p^{nl}]$	3.07	2.30	19.5	6.08
Calcul de $[U]_p$	2.77	2.35	4.61	2.17

Tableau 1. *Comparaison des temps de calculs pour une série de 25 termes sur un P III 750 Mhz et 256 Mo de Ram*

vive. Nous avons réalisé les mêmes calculs sur une autre machine possédant le double de mémoire vive (Alpha Server 800 5/500 , 512 Mo de Ram). Nous présentons les résultats dans le tableau 2.

	Ordre 10		Ordre 25	
	Méthode I	Méthode II	Méthode I	Méthode II
Inversion de $[K_t]$	14.8	14.8	14.8	14.8
Calcul de $[S_p^{nl}]$	5.11	3.35	17.80	10.24
Calcul de $[{}_1F_p^{nl}]$	0.26	0.23	0.3	0.23
Calcul de $[{}_2F_p^{nl}]$	1.14	2.19	2.09	6.98
Calcul de $[U]_p$	1.09	0.86	1.93	1.44

Tableau 2. *Comparaison des temps de calculs pour une série de 25 termes sur Alpha Server*

Ces résultats [tableaux 1,2] mettent l'accent sur le fait que ce qui est coûteux dans la résolution à l'aide de la MAN dans CAST3M est la gestion de la mémoire vive et non pas la méthode d'implémentation. On note cependant que la méthode I reste plus coûteuse en temps calcul que la méthode II, car elle est plus consommatrice de mémoire. Ce petit surcoût est compensé par le fait qu'en utilisant des opérateurs standard, la procédure est valable pour toute la bibliothèque d'éléments de CAST3M.

Pour étudier l'influence de la méthode d'implémentation sur le rayon de convergence, nous avons considéré la flexion d'une poutre encastrée à une extrémité et soumise à une force verticale à l'autre. Cette poutre a été maillée à l'aide de quadrilatères à 4 nœuds. Nous avons utilisé uniquement des développements en série entière dans ces calculs. Le tableau 3 donne un récapitulatif des rayons de convergence relatif, noté R_{rel} [14], obtenus pour différents pas calculés avec des séries entières ayant 10, 15 et 20 termes :

$$R_{rel} = \frac{RayonLagr.Act.}{RayonLagr.Tot.} \qquad [14]$$

On observe [tableau 3] que pour le premier pas les rayons de convergence sont identiques. En effet, dans cet exemple nous n'avons pas considéré de pré-chargement et donc les deux calcul sont strictement identiques. On note pour les autres pas une très faible augmentation du rayon de convergence quand le problème est formulé en lagrangien actualisé.

PAS	1	2	3	4	5
Ordre 10	1.00	1.0005	1.0012	1.0006	1.0016
Ordre 15	1.00	1.0006	1.0007	1.001	1.0009
Ordre 20	1.00	1.0006	1.0018	1.0013	1.001

Tableau 3. *Comparaison des rayons de convergence*

Dans cet exemple de flexion simple, λ est une fonction strictement croissante du paramètre de pilotage a. Nous avons étudié l'influence de l'ordre de développement limité sur les temps de calculs. Les résultats présentés dans le tableau 4 sont obtenus en utilisant la méthode I sur un Alpha Sever 800 5/500. Les temps présentés dans le tableau sont exprimés en second et correspondent au temps total du calcul. On retrouve les résultats classiques de la MAN liant une augmentation du rayon de convergence avec un augmentation de l'ordre du nombre de ddl. Dans ce calcul à petit nombre de ddl, il apparaît qu'il est préférable d'utiliser un développement à l'ordre 10.

5 Pas Ordre 10		5 Pas Ordre 15		5 Pas Ordre 20	
Tps(s)	λ_{max}	Tps(s)	λ_{max}	Tps(s)	λ_{max}
324.3	22.11	637.1	45.34	1199	68.28

15 Pas Ordre 10		10 Pas Ordre 10	
Tps(s)	λ_{max}	Tps(s)	λ_{max}
917.8	250.1	612.4	80.11

Tableau 4. *Comparaison des temps de calculs et des λ_{max}*

Il semble que, par comparaison avec la littérature [IMA 01], nos coûts de calcul de seconds membres sont importants et augmentent vite avec l'ordre, même dans le cadre de la méthode II. Typiquement dans la littérature, pour un petit problème (5000 ddl), le coût d'une matrice inversée vaut 15 à 25 seconds membres et ce rapport passe à plus de 100 pour un gros problème (50 000 ddl). Nous ne sommes pas encore en mesure de discuter complètement cette question, mais il se peut qu'une réécriture de certains opérateurs standard de CAST3M puissent diminuer nos coûts de calcul de séries.

4. Cas d'un calcul industriel

Les calculs ont été réalisés avec l'implémentation en lagrangien actualisé. Nous avons considéré une structure de type plaque (figure 3) constituée d'élément DKT à 3 nœuds. La machine que nous avons utilisée est un PC à deux processeurs avec 2 Giga de Ram. Nous avons réalisé des calculs avec respectivement 180 000 ddl et 300 000 ddl. Les temps calculs que nous présentons dans les tableaux 5 et 6 sont exprimés en seconde. La subroutine que nous avons utilisée est celle développée dans le cadre de la méthode I.

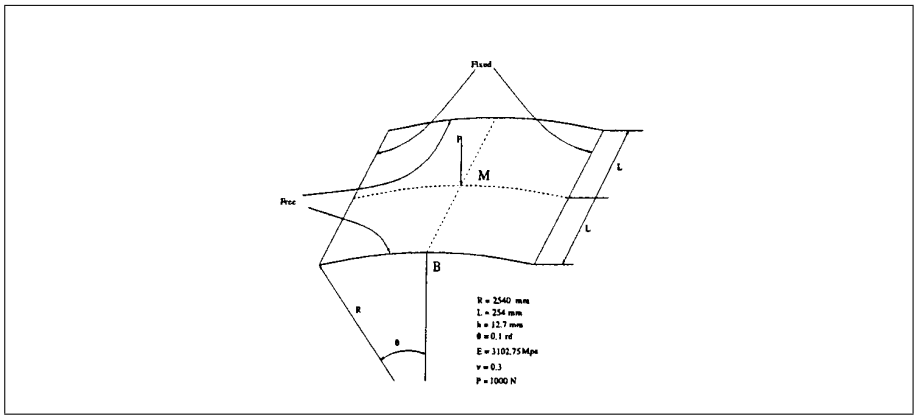

Figure 3. *Toit soumis à une force ponctuelle*

Ordre	Second membre	Résolution	Temps total	Ordre	Second membre	Résolution	Temps total
1			110	6	8.7	4.65	18.15
2	2.5	3.42	11.34	7	11.1	4.30	20.42
3	4.8	4.02	13.66	8	10.2	4.62	19.84
4	5.5	4.66	15.14	9	11.5	5.18	21.53
5	7.9	4.36	16.84	10	13.8	3.31	23.97

Tableau 5. *Comparaison des temps de calculs pour une série de 10 termes et 180 000 ddl*

Le temps total de l'ordre 1 correspond au temps nécessaire à l'inversion de l'opérateur tangent et à la résolution du système linéaire. Pour les autres ordres, le temps total correspond à la somme du temps de calcul du second membre, de la résolution du système, du calcul et du stockage des données nécessaires au calcul à l'ordre suivant.

Les résultats du tableau 5 montrent que dans le calcul avec 180 000 ddl l'inversion de l'opérateur tangent correspond au calcul d'environ 8 termes de la série.

Ordre	Second membre	Résolution	Temps total	Ordre	Second membre	Résolution	Temps total
1			327	9	25.09	11.08	46.25
2	6.9	12.98	29.27	10	28.71	10.47	49.08
3	11.0	8.82	29.46	11	29.85	10.60	51.26
4	13.01	11.98	34.81	12	31.53	11.90	53.74
5	16.49	13.52	39.33	13	35.00	11.52	57.42
6	19.35	10.62	39.85	14	37.88	11.52	59.70
7	20.62	10.93	41.19	15	38.69	10.37	59.66
8	22.85	10.16	42.51				

Tableau 6. *Comparaison des temps de calculs pour une série de 15 termes et 350 000 ddl*

Les résultats du tableau 6 montrent que dans le calcul avec 300 000 *ddl* l'inversion de l'opérateur tangent correspond au calcul d'environ 10 termes de la série. Dans les deux cas, le temps moyen de calcul d'un ordre ≥ 2 est de l'ordre de 13 % du temps de calcul du premier ordre. Sachant que le temps de calcul de du $10^{ième}$ ordre est double de celui du premier ordre, ces résultats montrent qu'il ne faut pas considérer des séries de plus de 15 termes pour être numériquement efficace. Nous avons pu vérifier cette tendance dans le calcul de la flexion simple de la poutre (tableau 4). Ces conclusions pourraient être modulées en fonction de la remarque faite à la fin du paragraphe 3.

5. Conclusion

La MAN est adaptée à la programmation objet, si on prend le temps de déterminer des opérateurs généraux pour chaque cas étudié. CAST3M est un code industriel adapté à l'implémentation aussi bien de lois de comportement, que de méthodes numériques. Dans le cas de la non linéarité géométrique, il suffit d'utiliser la programmation objet pour implémenter la MAN. L'écriture de la méthode numérique dans un langage objet ne pénalise pas beaucoup son efficacité. Les résultats présentés montrent qu'il n'est pas souhaitable d'utiliser un ordre élevé du développement limité pour être numériquement efficace. Cependant, pour déterminer un ordre optimal, il est nécessaire de tenir également compte des rayons de convergence.

6. Bibliographie

[BAG 01] BAGUET S., « Stabilité des structures minces et sensibilité aux imperfections par la méthode asymptotique numérique », Thèse de Doctorat, Université d'Aix-Marseille II, 2001.

[CAD 01] CADOU J. M., POTIER-FERRY M., COCHELIN B., DAMIL N., « Asymptotique Numerical Method for stationary Navier-stokes equations and with Petrov-Galerkin for-

mulation », *International Journal for Numerical Methods in Engineering*, vol. 50, 2001, p. 825-845.

[COC 94a] COCHELIN B., « Méthodes Asymptotiques-Numériques pour le calcul non-linéaire géométrique des structures élastiques », Habilitation à diriger des recherches, Université de Metz, 1994.

[COC 94b] COCHELIN B., DAMIL N., POTIER-FERRY M., « The Asymptotic-Numerical-Method : an efficient perturbation technique for nonlinear structural mechanics », *Revue Européenne des Eléments Finis*, vol. 37, 1994, p. 1181-119.

[DAY 01] DAYA E. M., POTIER-FERRY M., « A numerical method for nonlinear eigenvalue problems : application to vibration of viscoelastic structures », *Computers & Structures*, vol. 79, 2001, p. 533-541.

[ELH 98] ELHAGE-HUSSEIN A., DAMIL N., POTIER-FERRY M., « An asymptotic numerical algorithm for frictionless contact poblems », *Revue Européenne des Eléments Finis*, vol. 17, 1998, p. 119-130.

[ELH 00] ELHAGE-HUSSEIN A., DAMIL N., POTIER-FERRY M., « A numerical continuation method based on Padé approximants », *International Journal of Solids and Structures*, vol. 37, 2000, p. 6981-7001.

[IMA 01] IMAZATÈME A., CADOU J. M., ZAHROUNI H., POTIER-FERRY M., « A new reduced basis method for non-linear problems », *Revue Européenne des Eléments Finis*, vol. 1, 2001, p. 55-79.

[PAS 97] PASQUET P., « Langage et procédures », Notice n° 1, octobre1997, CEA.

[VER 93] VERPEAUX P., ROBEAU M., CHARRAS T., COMBESCURE A., « Manuel d'utilisation ESOPE - GEMAT », Rapport DMT/93 n° 617, novembre1993, CEA.

[ZIE 97] ZIENKIEWICZ O. C., TAYLOR R. L., *The finite element method - volume 2*, Editions Mc Graw Hill, Londres, 1997.