



HAL
open science

Calcul parallèle haute performance dans CAST3M

Pierre Verpeaux, Jean-Yves Cognard

► **To cite this version:**

Pierre Verpeaux, Jean-Yves Cognard. Calcul parallèle haute performance dans CAST3M. 8e Colloque national en calcul des structures, CSMA, May 2007, Giens, France. hal-01504170

HAL Id: hal-01504170

<https://hal.science/hal-01504170>

Submitted on 8 Apr 2017

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.

Public Domain

Calcul parallèle haute performance dans CAST3M

Pierre Verpeaux* — Jean Yves Cognard**

* C.E.A. de Saclay, DMT/SEMT, 91191 GIF SUR YVETTE Cedex
pv@sem2.smts.cea.fr

** Laboratoire de Mécanique des Structures Navales
ENSIETA, 2 rue F. Verny, 29806 Brest Cedex 09, France
jean-yves.cognard@ensieta.fr

RÉSUMÉ. Ce papier présente les principes d'une approche parallèle adaptée à la simulation d'une large classe de problèmes non-linéaires. Elle est basée sur l'utilisation de deux décompositions de domaine, dans le but d'utiliser les propriétés mécaniques des différentes équations à résoudre. L'objectif est de distribuer la charge de calcul sur les différents processeurs en limitant au maximum les phases de redistribution. L'implémentation est réalisée à partir d'extensions des possibilités du langage de commande GIBIANE du code CAST3M. Une procédure automatique a été mise en place pour une large classe de problèmes non-linéaires pour utiliser les possibilités des différents ordinateurs parallèles, et en particulier les architectures performantes et économiques de type PC 64 bits.

ABSTRACT. We present the principles of a parallel approach suited to the simulation of a wide class of non-linear problems. It is based on the use of two domain decompositions, in order to make use of the mechanical properties of the different type of equations to be solved. The goal is to balance the computation load over the various processors by limiting the task redistributions. The implementation of this algorithm is carried out starting from an extension of the possibilities of GIBIANE: the user language of the code CAST3M. Moreover an automatic procedure has been developed in order to use the possibilities of different parallel computers, and in particular efficient and economic configurations of 64 bits PC.

MOTS-CLÉS : Simulations non-linéaires, stratégies parallèles, algorithmes, problèmes de grande taille, équilibrages des charges .

KEYWORDS: Non-linear computations, parallel strategies, algorithms, large scale problems, load balancing.

1. Introduction

La simulation de structures à comportement non-linéaire conduit souvent à des coûts numériques trop élevés pour une utilisation courante sur site industriel. L'utilisation conjointe d'algorithmes performants et des possibilités des calculateurs parallèles est nécessaire pour réduire fortement le coût de ces simulations complexes (Noor *et al.*, 2000 ; Topping, 1999). De plus les performances des architectures de type PC augmentent rapidement et offrent des configurations économiques de calculateurs parallèles (Multi-processeurs 64 bits et Clusters).

L'objectif de cette communication est de présenter, dans le cadre du calcul de structures, les possibilités d'un environnement de programmation qui prend en compte les spécificités des différents types d'ordinateurs, qu'ils soient à mémoire distribuée, partagée ou partagée-distribuée (Cognard *et al.*, 2000). Ce système est compatible avec l'architecture des super-calculateurs, des stations de travail et des PC ; ces ordinateurs ont maintenant en commun une architecture de base commune : un multi-processeur accédant à une mémoire commune.

L'approche présentée du parallélisme est adaptée à la simulation d'une large classe de problèmes non-linéaires d'évolution en quasi statique. Elle est basée sur l'utilisation de deux découpages dont le but est d'équilibrer les charges de calcul des différents processeurs en limitant les redistributions de tâches (Cognard *et al.*, 2004). Ces deux propriétés, équilibrage des tâches et limitation des communications, sont nécessaires pour obtenir un algorithme parallèle efficace. Pour faciliter le développement d'applications parallèles, un concept de langage parallélisable s'appuyant sur un modèle à flux de données a été proposé en utilisant les possibilités de l'environnement de programmation parallèle. Ainsi, l'utilisateur de CAST3M peut distribuer des calculs, sachant que le système assure la cohérence des données et la synchronisation des tâches en limitant les phases d'attente des applications. Pour assurer la résolution en parallèle d'une large classe de problèmes, de façon transparente pour l'utilisateur tout en assurant une bonne efficacité, différents développements récents ont été effectués dans le code CAST3M.

Les gains de performance obtenus pour la résolution de problèmes non-linéaires, dans un environnement d'exploitation industrielle, sont présentés.

2. Stratégie utilisée pour la parallélisation des simulations non-linéaires

De nombreux travaux ont été réalisés pour résoudre efficacement les problèmes linéaires de grande taille sur les ordinateurs parallèles. Mais la prise en compte des non linéarités matériaux pose des difficultés supplémentaires. La parallélisation de ces problèmes, généralement résolus par les méthodes de type incrémentale, demande une architecture de code adéquate et un environnement de développement parallèle adapté. En effet, les inconnues du problème dépendent du comportement du matériau via les variables internes utilisées pour décrire les phénomènes

physiques. De plus, le coût numérique d'intégration des lois de comportement est fortement lié à la sollicitation locale du matériau ; il est donc impossible de prévoir l'évolution du coût de cette opération, en chaque point de la structure.

Programme Maître	Programme Assistant i	
	découpage I	découpage II
test de convergence accélération de convergence	calcul du déplacement	loi de comportement calcul du résidu

Figure 1. Principe de l'algorithme - utilisation des deux découpages

La stratégie proposée, pour limiter les transferts de données et les phases de ré-équilibrages, est basée sur l'utilisation de deux découpages dans le but d'équilibrer les charges des processeurs. L'idée consiste à utiliser un découpage classique en sous-domaines pour la résolution des problèmes globaux linéaires, compatible avec les techniques d'accélération de convergence et d'utiliser un second découpage permettant d'équilibrer les charges lors de l'intégration de la loi de comportement en évitant la redistribution des tâches au cours du calcul (Cognard *et al.*, 2004) . Chaque découpage est adapté aux propriétés mécaniques des sous problèmes à résoudre. La figure 1 résume l'enchaînement des différentes phases des simulations.

3. Langage de commande parallèle

Le logiciel CAST3M est une collection d'opérations élémentaires, nommées OPERATEURS, qui travaillent sur des OBJETS qui sont référencés par l'utilisateur. Une application est une séquence d'opérations élémentaires dont la syntaxe est :

$$\text{RES1 RES2} = \text{OPER OBJ1 OBJ2 OBJ3} ;$$

Le langage de commande (GIBIANE) utilise les fonctionnalités d'ESOPÉ 2000 qui est un système de type mémoire virtuelle partagée basée sur la notion d'objets (Cognard *et al.*, 2000). Il permet le développement d'applications parallèles portables, réutilisant une grande partie du code séquentiel ; de plus, il évite au programmeur les tâches fastidieuses de gestion des échanges de données et de gestion de cohérence des données. Pour faciliter les dialogues et les synchronisations entre applications, des activations conditionnelles peuvent être utilisées.

Pour faciliter le développement d'applications parallèles, un concept de langage de commande parallèle a été proposé. Ainsi, l'utilisateur de CAST3M peut distribuer des calculs, sachant que le système assure la cohérence des données lors de l'enchaînement des tâches tout en limitant les phases d'attente des applications. La

distribution des tâches sur les différents assistants se fait en utilisant l'opérateur « ASSISTANT » (« ASSI ») de la façon suivante :

$$\text{RES1 ... RES}_i = \text{'ASSI'}_i \text{'OPER'} \text{OBJ1 ... OBJ}_n ;$$

CAST3M soumet l'exécution de l'opération élémentaire 'OPER' à l'assistant n° i, puis traite l'opération suivante. Les résultats de l'opération ne seront utilisables que lorsque l'opération associée sera terminée. L'assistant effectue l'opération lorsqu'il est disponible et attend éventuellement la disponibilité des données.

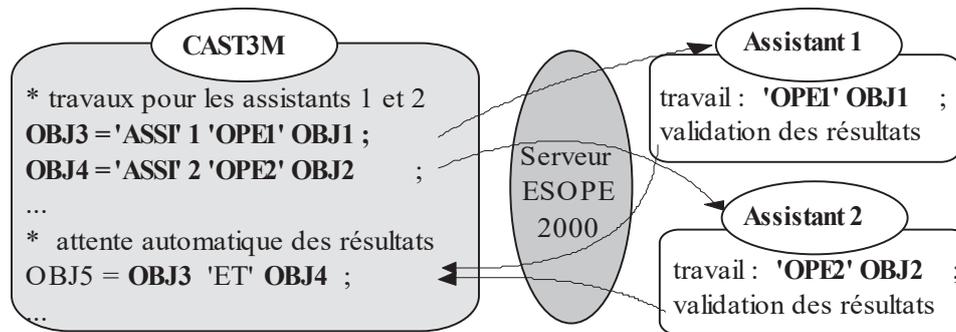


Figure 2. Distribution des calculs à partir du langage de commande

L'exécution asynchrone de calculs au niveau utilisateur, à partir du langage de commande parallèle, a été simplifiée à l'aide d'un objet « conteneur » regroupant les décompositions des objets associé à des fonctionnalités de l'opérateur « ASSI ». La structuration des données proposée pour la mise en œuvre de cette technique de type « data parallèle » pour distribuer les données et les calculs permet d'assurer la compatibilité avec les simulations en séquentiel. De plus, une procédure automatique a été développée pour paralléliser une large classe de problèmes non-linéaires d'évolution, de façon transparente pour l'utilisateur.

4. Exemple numérique

Les possibilités de l'approche parallèle proposée sont présentées à partir d'un exemple d'éprouvette de traction-torsion (figure 3) soumise à des déplacements imposés aux sections extrêmes : un déplacement axial $U_z = U_0 \sin(2\omega t)$ et une rotation $\Theta = \Theta_0 \sin(\omega t)$. Un modèle de comportement viscoplastique de type Chaboche (Z6CNDT17-11) est utilisé (Lemaitre *et al.*, 1994). Les calculs sont réalisés pour 3 cycles de chargement, découpés en 150 incréments. Pour caractériser le comportement de l'algorithme nous utilisons le temps « horloge » qui correspond au temps de restitution du calcul (temps incluant le temps de gestion du système). Nous définissons le facteur de gain comme étant le rapport entre le temps passé pour effectuer le calcul sur 1 processeur et le temps passé pour résoudre le problème avec N processeurs. Les calculs ont été réalisés sur un ordinateur IBM à mémoire partagée avec 4 processeurs. Pour ces simulations, les problèmes linéaires globaux sont résolus par un solveur direct parallèle ; celui-ci est basé sur une technique de

renumérotation de type « nested dissection » (Heath *et al.*, 1995), proche des méthodes de décomposition de domaine (Cognard *et al.*, 2004). Les résultats, résumés dans le tableau 1, montrent une faible variation du facteur de gain en fonction du nombre de degrés de liberté. Pour ce type de problème, la phase la plus coûteuse est l'intégration de la loi de comportement (de l'ordre de 80%). De plus, il est important de noter que les incréments de comportement élastique, qui sont simulés avec la même procédure, pénalisent peu le facteur de gain.

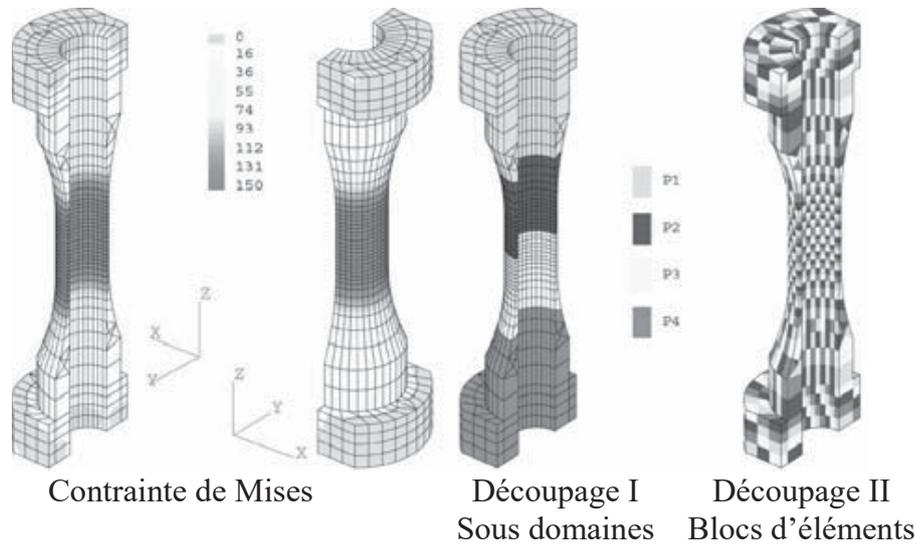


Figure 3. *Modèle étudié, les deux découpages & la contrainte équivalente de Mises.*

Nombre de ddl	temps avec 1 processeur (s)	temps avec 4 processeurs (s)	Facteur de gain (4 processeurs)
5 900	3 177	1 205	2,65
57 000	28 060	8 280	3,39
100 000	42 840	12 910	3,32
160 000	79 500	23 780	3,34
210 000	262 830	68 320	3,85

Tableau 1. *Comportement de l'algorithme parallèle en fonction du nombre de ddl.*

Sous les hypothèses des grandes transformations, une réactualisation de la rigidité globale du système est nécessaire. Ceci pénalise un peu le fonctionnement de l'algorithme, mais des facteurs de gain supérieur à 3 sont obtenus avec 4 processeurs (avec le temps de restitution, temps « horloge ») (Cognard *et al.*, 2004).

5. Simulation de problèmes de grande taille

La résolution de problèmes de grande taille nécessite l'utilisation d'espaces de débordement sur disques. La gestion de la persistance des objets manipulés,

partagés par les différentes applications du programme parallèle, et du débordement sur disque est réalisée par la mise en place de stratégies de récupération de la place mémoire assurant la cohérence des données et limitant les phases de blocage des applications parallèles.

Une version s'appuyant sur le standard Posix « pthread » a été initialement développée pour assurer les performances de l'environnement de programmation parallèle et pour permettre le portage du code sur les ordinateurs à mémoire partagée. L'extension de la stratégie proposée aux ordinateurs à mémoire partagée/distribuée est en cours.

6. Conclusion

L'environnement de programmation parallèle de CAST3M permet la mise en œuvre facile d'algorithmes parallèles soit au niveau programmation, soit au niveau utilisateur. Ce système évite à l'utilisateur les tâches fastidieuses de gestion des échanges de données et de gestion de cohérence des données. Les algorithmes développés pour la résolution de problèmes linéaires et non-linéaires sur des ordinateurs à mémoire partagée donnent de bons résultats ; de plus, leur utilisation est transparente pour l'utilisateur. L'extension de la version « multithreads » du langage de programmation de CAST3M aux ordinateurs à mémoire distribuée-partagée est en cours de réalisation ; ceci permettra, après validation des différents algorithmes, de traiter des problèmes de grande taille en utilisant la puissance et les capacités mémoire des ordinateurs modernes.

7. Bibliographie

- Cognard J. Y., Thomas F., Verpeaux P., « An integrated approach to solving mechanical problems on parallel computers », *Advances in Engineering Software*, vol. 31, 2000, p. 885-899.
- Cognard J. Y., Poulhalec A., Thomas F., Verpeaux P., « A parallel environment and associated strategies in structural non-linear analysis », *Progress In Engineering Computational Technology*, Eds. Topping B. H. V. & Mota Soares C. A., Saxe-Coburg Publications, 2004, Chapter 14, p 323-352.
- Heath M. T., Raghavan P., « A cartesian parallel nested dissection algorithm », *SIAM J. Matrix Anal. Appl.*, vol. 16, 1995, 235-253.
- Lemaitre J., Chaboche J. L., *Mécanique des matériaux solides*, Dunod Paris, 1985.
- Noor A. K., Venneri S. L., Paul D. B., Hopkins M. A., « Structure technology for future aerospace systems », *Computer & Structures*, vol. 74, 2000, p. 507-519.
- Topping B. H. V., *Parallel and distributed processing for computational mechanics : system and tools*, Saxe-Coburg publications, 1999.