HAL
open science

Solveurs directs parallèles et équilibrage de charges

Jean-Paul Boufflet, Emmanuel Lefrançois

To cite this version:

Jean-Paul Boufflet, Emmanuel Lefrançois. Solveurs directs parallèles et équilibrage de charges. 7e colloque national en calcul des structures, CSMA, May 2005, Giens, France. hal-01812887

HAL Id: hal-01812887

https://hal.science/hal-01812887v1

Submitted on 11 Jun 2018

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

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

Public Domain

Solveurs directs parallèles et équilibrage de charges

Jean-Paul Boufflet[1], **Emmanuel Lefrançois**[2]

[1] *Laboratoire Heudiasyc,* [2] *Laboratoire de Roberval*
Université de Compiègne (UTC)
60200 Compiègne, Cedex
jean-paul.boufflet@hds.utc.fr, emmanuel.lefrancois@utc.fr

RÉSUMÉ. *Ce papier présente une approche de la parallélisation d'outils de calcul par éléments finis orientée vers une utilisation rationnelle des ressources informatiques. L'approche est basée sur une technique classique de décomposition de domaine et met l'accent sur la nécessité d'équilibrer le découpage d'un maillage en volumes de calcul et non pas de données, le principal risque avec cette dernière approche étant la dégradation des peformances d'un calcul effectué en parallèle. La technique d'équilibrage exposée est basée sur le transfert d'éléments finis entre sous-domaines et sur la connaissance d'un modèle comportemental du solveur permettant d'estimer a priori le volume de calcul de chacun des sous-domaines. Les possibilités qui en découlent sont de pouvoir volontairement déséquilibrer le découpage pour des applications utilisées sur un environnement de calcul aux performances hétérogènes et de prévoir pour un maillage donné, le nombre optimal de découpages qui conduit au temps de calcul global le plus petit.*

ABSTRACT. *This paper exposes a parallelization approach for finite element solvers with the objective of a rational use of computer resources. This one is based on a classical domain decomposition technique and points out the necessity to ensure a decomposition in terms of loads balancing. The proposed technique is based on finite elements transfers between subdomains and the knowledge of a solver's behaviour model in order to estimate the load assigned to each subdomain. Extensions concern the capability to voluntary unbalance the partitionned load in order to conduct a distributed calculation on heterogeneous computer environment. A second extension concerns the possibility to estimate the optimal number of subdomains that will lead to the minimum global calculation time.*

MOTS-CLÉS : *Eléments finis, calcul parallèle, décomposition de domaines, complément de Schur, équilibrage de charge*

KEYWORDS: *Finite elements, parallel computing, domain decomposition, Schur complement, loads balancing*

1. Objectifs et originalité

L'objectif de ce papier est de réaliser la parallélisation d'applications en vue d'une utilisation rationnelle des moyens de calculs. La démarche proposée s'appuie sur la technique des sous-structures qui a pour objectif de maximiser le nombre de tâches parallélisables, à savoir réalisables en simultané. Selon le type de solveur employé [HEN 00], un découpage équilibré en volumes de données (nombre de nœuds et donc d'inconnues) ne conduit pas nécessairement à un équilibrage en volume de calcul (ou de charge). Les figures 1 et 2 illustrent cette tendance dans le cas d'un calcul de hauteur de houle dans le port de Brest avec le logiciel *REFONDE*[1]. Le maillage a été découpé sous METIS© en cinq sous-domaines équilibrés en volumes de données.

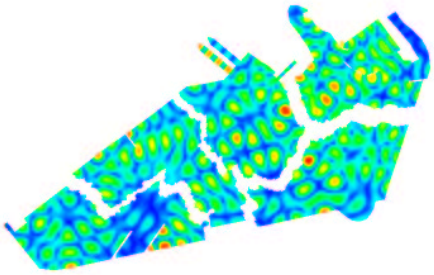

Figure 1. *Partitions en volumes de données équilibrés*

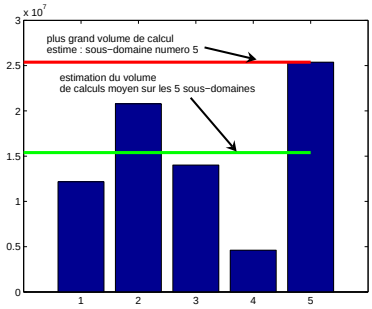

Figure 2. *Volumes de calcul déséquilibrés*

Le volume de calcul est ici estimé *a priori* d'après un modèle comportemental du solveur. On notera le déséquilibre induit par ce découpage sur l'estimation de la charge pour chacun des sous-domaines. Les calculs confirment la tendance estimée.

L'originalité de l'approche proposée ici porte à la fois sur le déploiement d'une stratégie de ré-équilibrage des sous-domaines initiaux en s'appuyant sur la connaissance d'un modèle comportemental du solveur, ce dernier permettant d'estimer *a priori* la charge associée à chacun des sous-domaines, et sur la prédiction avant calcul du nombre de sous-domaines optimal. Nous illustrons Figure 3 l'évolution des performances d'un calcul de déformation d'une poutre en flexion en fonction du nombre de sous-domaines considérés et l'existence d'un découpage optimal. L'objectif visé est donc double et s'oriente pleinement vers une utilisation rationnelle des moyens de calculs disponibles avec la nécessité de fournir les informations utiles *a priori* (ne requérant aucune utilisation au préalable du solveur) en fonction du type d'applications, du solveur, des configurations matérielles ... Une application possible est l'estimation et la répartition de charges au sein d'une grille de calcul en vue d'assurer une prestation de type qualité de service.

1. développement UTC-CETMEF au début des années 1990

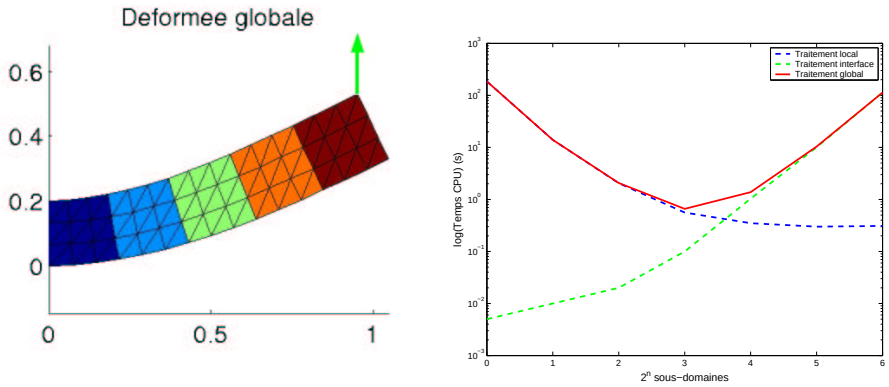

Figure 3. *Découpage d'une poutre en flexion - Evolution du temps global de calcul*

2. Méthodologie

Notre démarche est motivée par notre expérience pratique et par les observations d'autres auteurs [HEN 98, HEN 00, SCO 99]. Elle se place dans la continuité des travaux présentés sur un solveur frontal [DEN 03]. L'objectif de la parallélisation d'un solveur peut être double : le premier est de réduire de manière significative le temps de calcul, le second de réaliser des calculs nécessitant des ressources mémoire dépassant les capacités usuelles d'une unique machine. Nous nous intéressons aux cas de solveurs de type direct couramment employés en mécanique statique ou stationnaire et pour des résolutions temporelles implicites.

2.1. *Principe de l'approche du calcul distribué*

Le traitement d'un problème physique stationnaire par la méthode des éléments finis conduit au système global suivant :

$$[K]\{\phi\} = \{F\},$$

avec $[K]$ la matrice de rigidité du système, $\{\phi\}$ le vecteur des inconnues nodales et $\{F\}$ le vecteur des sollicitations. Les problèmes instationnaires ou non linéaires requièrent la résolution de ce système au sein d'une boucle de calcul, respectivement en temps et itérative. Ce système résulte de la discrétisation des équations d'équilibre du problème traité sur le maillage associé au domaine d'étude. La résolution de ce système par une technique de décomposition de domaine requiert les étapes suivantes :

1) Découpage du domaine initial en N_s sous-domaines ;

2) en simultané sur N_s processeurs :

a) Assemblage du système local ;

b) Factorisation des inconnues internes ;

c) Extraction du complément de Schur local ;

d) Assemblage du problème d'interface.

3) Résolution du problème d'interface ;

4) Redistribution de la solution sur tous les sous-domaines ;

5) en simultané sur N_s processeurs :

a) Phase de remontée pour le calcul des solutions internes aux sous-domaines.

2.2. *Décomposition et calcul*

Pour illustrer la démarche nous utilisons le maillage en quadrangle de la figure 4 et une factorisation de type *LU*.

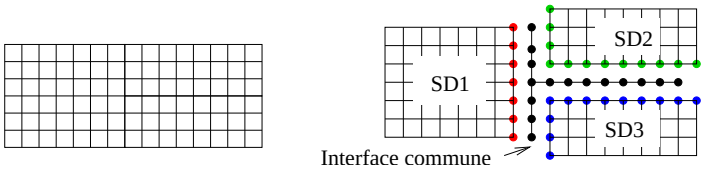

Figure 4. *Le domaine initial et sa décomposition en $N_s = 3$ sous-domaines*

La décomposition met en évidence les nœuds frontières dupliqués. Pour un sous-domaine, noté $SD^{(i)}$ avec $i \in [1, N_s]$, $\{\psi_1\}_i$ est le vecteur des inconnues associées aux nœuds internes et $\{\psi_2\}_i$ est le vecteur des inconnues associées aux nœuds frontières. Nous construisons le système linéaire local au sous-domaine i :

$$[K]\{\psi\} = \{F\} \Leftrightarrow \begin{bmatrix} L_{11} & 0 \\ L_{21} & Id \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & S_i \end{bmatrix} \begin{Bmatrix} \psi_1 \\ \psi_2 \end{Bmatrix}_i = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix}_i, \qquad [1]$$

que nous condensons en factorisant les inconnues ψ_1. Cette factorisation est incomplète. Pour chaque sous-domaine, nous obtenons un système dit "interface" associé exclusivement aux variables $\{\psi_2\}_i$. Sa matrice est $[S_i]$, matrice du complément de Schur local du sous-domaine i. Elle traduit une condensation de toutes les informations (rigidité par exemple) sur les nœuds d'interface. Nous posons :

$$\begin{bmatrix} U_{11} & U_{12} \\ 0 & S_i \end{bmatrix} \begin{Bmatrix} \psi_1 \\ \psi_2 \end{Bmatrix}_i = \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix}_i, \qquad [2]$$

Puis nous calculons $\{y_1\}_i$ et $\{y_2\}_i$ avec :

$$[L_{11}]\{y_1\}_i = \{F_1\}_i, \quad \text{et} \quad [L_{21}]\{y_1\}_i + \{y_2\}_i = \{F_2\}_i. \qquad [3]$$

A ce stade, nous avons N_s systèmes interfaces, et N_s systèmes d'équations [2]. Nous assemblons les N_s systèmes interfaces en un seul avec les procédures d'assemblage (au sens des éléments finis) :

$$[S] = \sum_{i=1}^{N_s} [S_i], \quad \{\psi_2\}_{inter} = \sum_{i=1}^{N_s} \{\psi_2\}_i, \quad \{y_2\}_{inter} = \sum_{i=1}^{N_s} \{y_2\}_i.$$

nous obtenons le système interface global $[S]\{\psi_2\}_{inter} = \{y_2\}_{inter}$ qui permet de calculer les variables interfaces $\{\psi_2\}_{inter}$, qui est ensuite redistribuée vers chaque $SD^{(i)}$ pour calculer les $\{\psi_1\}_i$ avec [2]. Selon ce schéma, le temps global de calcul est la somme du temps maximal de condensation d'un sous-domaine, du temps d'assemblage des problèmes interfaces, du temps de résolution du problème d'interface et du temps de restitution. Un calcul efficace nécessite de synchroniser la fin des tâches de condensation. Or, le temps de factorisation partielle d'un $SD^{(i)}$ n'est pas proportionnel à son volume de données et dépend de la numérotation du sous-domaine. Le gain net en temps de calcul peut ainsi fortement varier : il faut chercher à équilibrer les volumes de calcul et non plus des volumes de données. Ce pilotage de la charge requiert la connaissance d'un modèle comportemental du solveur. Ce dernier permet d'estimer en fonction de la table des connectivités, la largeur de bande moyenne de la matrice de rigidité associée à chacun des sous-domaines. Cette démarche à déjà été utilisée avec succès pour plusieurs types de solveur direct [DEN 03]. Pour la version Skyline du solveur de *REFONDE*, cet estimateur est donné par :

$$\text{Nbre opérations} = \frac{1}{2} NEQ * \bar{H}^2 + 2 * NEQ * \bar{H} + NEQ. \qquad [4]$$

où NEQ est le nombre d'inconnues et $\bar{H}$ la hauteur moyenne des colonnes de stockage de la matrice.

2.3. *Equilibrage en volumes de calcul*

L'idée consiste à transférer des éléments finis entre le sous-domaine qui possède le plus gros volume de calcul estimé et ses sous-domaines voisins. Pour cela nous avons utilisé et adapté des algorithmes issus de la théorie des graphes. L'idée est de partir d'une partition initiale équilibrée en volume de données fournie par un outil standard puis de la corriger de façon à obtenir une partition équilibrée en volume de calculs. La procédure est répétée de façon itérative jusqu'à obtenir l'équilibrage. Elle peut ainsi être utilisée pour équilibrer une décomposition en N sous-domaines mais aussi comme outil d'aide à la décision pour déterminer le nombre "optimal" de sous-domaines avant tout calcul. Si l'on suppose en effet qu'il existe une énergie de calcul minimale pour traiter un maillage donné, nous pouvons simuler le comportement du solveur pour $2, 3, 4, ...$ sous-domaines et sélectionner la meilleure décomposition équilibrée en volumes de calcul.

Trois stratégies ont ainsi pu être testées : la première stratégie de pilotage de l'équilibrage détecte le sous-domaine de plus grand volume de calcul, puis transfère une partie des éléments finis frontières de ce sous-domaine vers ses sous-domaines voisins. Il s'agit d'une approche *push*. La seconde est élaborée selon une stratégie *pull* : le sous-domaine de plus petit volume de calcul estimé demande à ses voisins de lui fournir des éléments finis frontières. Enfin, nous avons mixé ces deux stratégies : une itération s'effectue en mode *push*, la suivante en mode *pull*. Nous illustrons figure 5, le résultat de l'équilibrage du découpage du port de Brest présenté sur les figures 1 et 2. Les temps de calcul mesurés sont conformes aux estimations. L'équilibre est meilleur et le temps de calcul moyen a diminué. La barrière de synchronisation induite par le

sous-domaine ayant le plus grand temps de calcul est abaissée : le temps de calcul diminue et on utilise mieux les ressources de calculs.

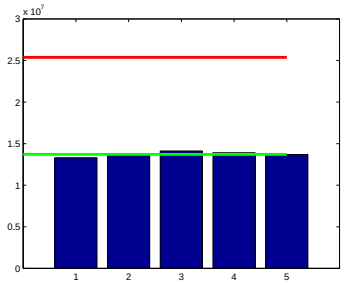

Figure 5. *Estimation des volumes de calculs, découpage initial (N_s =5)*

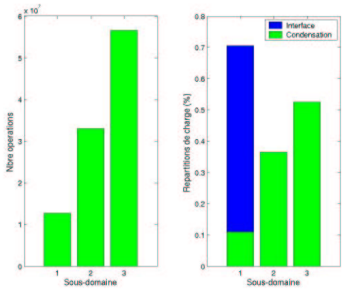

Figure 6. *Equilibrage avec vecteur de consignes (*$10\%,\ 30\%,\ 60\%$*)*

Une possibilité inhérente à la capacité d'équilibrer est de pouvoir justement déséquilibrer pour des applications amenées à être exécutées sur un environnement constitué de machines avec des performances hétérogènes (application autour des grilles de cluster différents par exemple). La figure 6 illustre le cas d'un découpage en trois blocs et volontairement déséquilibré selon les proportions 10%, 30% et 60%. Les deux courbes illustrent respectivement la répartition de la charge obtenue et les temps de calculs mesurés.

3. Conclusions

Les résultats présentés ne prétendent nullement justifier l'utilisation du calcul distribué, mais servent à la fois de cas-test de validation des outils nouvellement développés et d'exemple de démonstration de la procédure. Le travail actuel porte sur l'estimation du nombre optimal de sous-domaines menant au temps de calcul global le plus petit, toujours dans l'optique d'une démarche de type qualité de service. Ce travail est le fruit d'une collaboration entre une équipe d'informaticiens et une équipe de numériciens auquel participent des étudiants de l'UTC dans le cadre de mini-projets. Il répond à la démarche formulée autour du projet de plateforme PILCAD (Plateforme Inter-LAboratoire de CAlculs Distribués) soutenu par un abondement ANVAR.

4. Bibliographie

[DEN 03] DENIS C., BOUFFLET J., BREITKOPF P., RASSINEUX A., VAYSSADE M., « Equilibrage en Volume de Calcul pour un Solveur Parallèle Multi-niveau », *6ème Colloque National en Calcul des Structures*, 2003, p. 349-356, Giens, France.

[HEN 98] HENDRICKSON B., « Graph Partitioning and Parallel Solvers : Has the Emperor No Clothes ? », *Irregular'98, Lecture Notes in Computer Science*, vol. 1457, 1998, p. 218-225.

[HEN 00] HENDRICKSON B., « Load balancing fictions, falsehoods and fallacies », *Applied Mathematical Modelling*, vol. 25, 2000, p. 99-108.

[SCO 99] SCOTT J., « The Design of a Parallel Frontal Solver,Technical Report RAL-TR99-075 », rapport, 1999, Rutherford Appleton Laboratory.