



## Applications des techniques de calculs distribués à un code de calcul de houle aléatoire

Emmanuel Lefrançois<sup>(a)</sup>, Jean-Paul Boufflet<sup>(b)\*</sup>, Sébastien Bigot<sup>(b)\*\*</sup>

(a) Laboratoire de Roberval

(b) Maître de conférences\*, ingénieur stagiaire\*\*, Laboratoire Heudiasyc,  
Université de Compiègne (UTC) 60200 Compiègne, Cedex  
jean-paul.boufflet@hds.utc.fr, [emmanuel.lefrancois@utc.fr](mailto:emmanuel.lefrancois@utc.fr)

### Résumé.

Le code de calcul *REFONDE* permet d'effectuer des simulations de houle dans des domaines tels que les bassins de port. Le volume de calcul peut être conséquent. Dans cette étude, nous proposons d'appliquer des techniques de calculs distribués pour pouvoir diminuer le temps de calcul et pour permettre le traitement de problèmes de grandes tailles. Nous utilisons l'approche basée sur les sous-structures qui nécessite une décomposition du domaine. Cette décomposition doit être équilibrée en volume de calcul pour pouvoir traiter efficacement le problème.

### Abstract

*REFONDE* is a code dedicated to calculate swell phenomena in domains such as harbours. Volume data may be large. In this study, we propose to apply distributed tools in order to significantly reduce computing time and to deal with large problems. The approach is based on domain decomposition techniques and uses specific load-balancing strategy in order to be efficient.

**Mots-clés:** Code de houle aléatoire, éléments finis, calculs distribués, complément de Schur, équilibrage de charge

### 1.Objectif de l'étude

La connaissance de la houle revêt une grande importance pour l'ingénierie maritime. Elle intervient en premier lieu dans le dimensionnement des ouvrages maritimes mais s'avère aussi un obstacle majeur à la navigation et aux opérations portuaires. La qualité des simulations numériques est liée à :

- la superposition d'un nombre important de longueurs d'onde et pour un spectre directionnel le plus large possible (houle dite "aléatoire");
- l'utilisation de maillages adaptés aux périodes considérées (pouvant atteindre plusieurs millions de nœuds).

L'étude présentée porte sur l'adaptation du code de calcul de houle *REFONDE*<sup>1</sup> pour effectuer les calculs en distribué. Le modèle *REFONDE* est un code d'agitation de houle qui résout l'équation de réfraction-diffraction (avec conditions de réflexion) de Berkhoff par une méthode éléments finis. L'objectif n'est pas d'exposer le modèle initial du solveur, une présentation détaillée se trouve dans [1]. Le code actuel est séquentiel. Il effectue plusieurs

résolutions avec différentes périodes et directions de houle puis combine les résultats obtenus pour générer la solution globale. L'utilisation des techniques de calculs distribués offre les possibilités suivantes :

1. la résolution simultanée sur plusieurs machines, chacune traitant une période et direction de houle, puis combinaison des résultats ;
2. décomposition du domaine initial en N sous-domaines puis traitement en parallèle pour une direction et période de houle donnée ;
3. l'utilisation d'un modèle du comportement algorithmique du solveur permet d'optimiser le nombre de sous-domaines et leurs volumes de calcul afin d'obtenir le temps de calcul global le plus faible pour les directions/périodes.

Pour réaliser le point 1 il faut en mettre en œuvre de la programmation système, le code séquentiel n'est pas modifié. L'objectif du point 2 est double : réduire le temps de calcul et permettre des calculs sur des tailles de domaines actuellement hors de portée de la version séquentielle. Il faut pour cela utiliser une approche de calcul par sous-structures [2] qui nécessite de décomposer le domaine. Le solveur direct de *REFONDE* doit donc être repensé dans ce sens.

Le point 3 vise l'utilisation efficace des ressources de calcul. En effet, nous avons expérimentalement observé que les temps de calcul des tâches de condensations peuvent significativement varier. Le déséquilibre observé grève l'efficacité d'un calcul et est encore plus réhibitoire dans notre cas puisque plusieurs calculs avec des périodes et directions différentes doivent être menés.

Les outils standards de décomposition, tels que *METIS* [3] ou *CHACO* [4], équilibrent les volumes de données. Notre expérience pratique sur un solveur parallèle à front multiple [5] et sur le solveur de *REFONDE* (basé sur un stockage partiel des matrices du type SKYLINE [6]) semble confirmer l'analyse présentée par Bruce Hendrickson [7] : répartir équitablement les volumes de données n'induit pas systématiquement des temps de calcul bien équilibrés. Pour être efficace, il faut concevoir des techniques de décomposition de domaines qui tiennent compte du comportement algorithmique du solveur.

## 2.Modèle de calcul du code Refonde

Pour modéliser une agitation dans un port, il faut prendre en compte les phénomènes de réflexion, de diffraction et de réflexion produits par les obstacles à la propagation de la houle (digues, quais, etc). Berkhoff (1978) a établi une équation qui permet de modéliser ces trois aspects. Elle requiert deux hypothèses : une houle linéaire (amplitude des mouvements verticaux des particules situés à la surface suffisamment faible) et une faible pente pour les fonds (équation de pentes douces). Sous ces deux hypothèses, on obtient l'équation de Berkhoff :

$$\vec{\nabla} \cdot (c.c_g \cdot \vec{\nabla} \phi(x, y)) + k^2 \cdot c.c_g \cdot \phi(x, y) = 0, \quad \forall (x, y) \in V.$$

avec la vitesse de phase, la vitesse de groupe, le potentiel des vitesses et le domaine de calcul. Une discrétisation avec un élément fini [6] linéaire de type triangulaire (T3) conduit au système suivant :

$$[K]\{\phi\} = \{F\}, \quad [1]$$

avec  $[K]$  la matrice de rigidité pour une direction et une période données,  $\{\emptyset\}$  le vecteur des inconnues nodales du potentiel des vitesses et  $\{F\}$  le vecteur sollicitation.

### 3.Principe de l'approche du calcul distribué

Le domaine initial est décomposé en  $N$  sous-domaines reliés entre eux par des nœuds d'interface communs : il s'agit d'une décomposition sans recouvrement. Un processeur traite un sous-domaine. Nous distinguons les nœuds 'internes' et les nœuds 'd'interface'. Les nœuds de l'interface sont dupliqués au niveau des sous-domaines.

#### 3.1.Décomposition et calcul

Pour illustrer la démarche nous utilisons le maillage en quadrangle de la figure 1 et une factorisation de type  $LU$ .

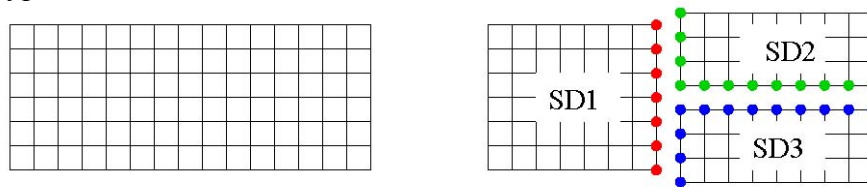


Figure 1. Le domaine initial et sa décomposition en  $N=3$  sous-domaines

La décomposition de la figure 1 met en évidence que les nœuds frontières sont dupliqués. Les sous-domaines sont assemblés en parallèle. Pour un sous-domaine, noté  $SD(i)$  avec  $i \in [1, N]$ ,  $\{\psi_1\}_i$  est le vecteur des inconnues associées aux nœuds internes et  $\{\psi_2\}_i$  est le vecteur des inconnues associées aux nœuds frontières. Les équations associées aux variables de  $\{\psi_1\}_i$  sont complètes, celles associées aux variables  $\{\psi_2\}_i$  ne le sont pas. Nous construisons le système linéaire local au sous-domaine  $i$ :

$$[K]\{\psi\} = \{F\} \Leftrightarrow \begin{bmatrix} L_{11} & 0 \\ L_{21} & Id \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & S_i \end{bmatrix} \begin{Bmatrix} \psi_1 \\ \psi_2 \end{Bmatrix}_i = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix}_i, \quad [2]$$

que nous condensons en factorisant les inconnues. 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.

Pour préparer la future étape de restitution nous posons :

$$\begin{bmatrix} U_{11} & U_{12} \\ 0 & S_i \end{bmatrix} \begin{Bmatrix} \psi_1 \\ \psi_2 \end{Bmatrix}_i = \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix}_i, \quad [3]$$

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

$$[L_{11}]\{y_1\}_i = \{F_1\}_i, \quad \text{et} \quad [L_{21}]\{y_1\}_i + \{y_2\}_i = \{F_2\}_i. \quad [4]$$

A ce stade, nous avons N systèmes interfaces, et N systèmes d'équations [3].

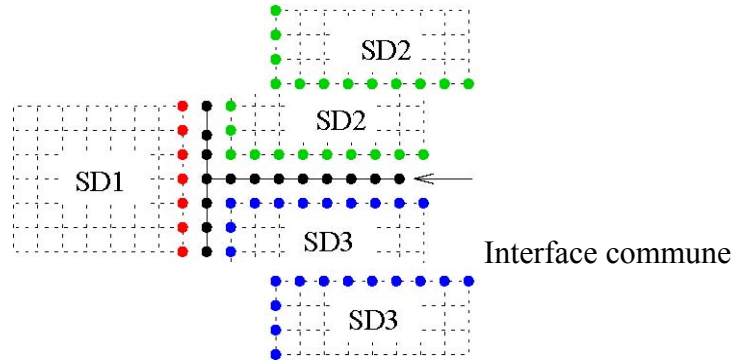


Figure 2. Condensation du domaine découpé en un problème d'interface

Nous assemblons les N systèmes interfaces en un seul avec les procédures d'as-semblage (au sens des éléments finis) :

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

Nous obtenons le système interface global  $[S]\{\psi_2\}_{inter} = \{y_2\}_{inter}$  qui permet de calculer les variables interfaces . La figure 2 illustre cette démarche, les contributions de tous les sous-domaines sont assemblées.  $\{\psi_2\}_{inter}$  est ensuite redistribuée vers chaque SD(i) ce qui permet de calculer les  $\{\psi_1\}_i$  avec [3].

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. Pour cela il faut utiliser un algorithme de prédiction du nombre d'opérations du solveur. Cette démarche a déjà été utilisée avec succès pour un autre type de solveur direct [5] et est actuellement en cours d'intégration pour le solveur de REFONDE.

### 3.2. Technique de décomposition et équilibrage de charge

Un premier code parallèle répartit les données dans l'ordre de leur lecture. Les sous-domaines  $SD(i)$  obtenus ne sont pas connexes. En effet, les données initiales ne sont ni organisées ni structurées pour former des sous-domaines. De fait, la taille de l'interface obtenue rend impraticable et inefficace le calcul parallèle selon l'approche présentée supra. Les données réparties sont donc structurées en parallèle pour former des sous-domaines connexes, ce qui réduit considérablement la taille du problème interface.

Un second code parallèle basé sur des techniques présentées dans [8] permet d'améliorer encore la qualité de l'interface entre les sous-domaines.

A ce stade, nous disposons de  $N$  sous-domaines qui ne sont pas équilibrés en volume de calcul. Pour atteindre cet objectif, nous devons disposer d'un estimateur de volume de calcul pour une tâche de condensation partielle d'un sous-domaine. Il faut établir le comportement algorithmique du solveur modifié sur un sous-domaine. Nous ne présenterons pas les détails de cette modélisation.

Un troisième code parallèle effectue l'équilibrage en volume 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. Il faut choisir judicieusement les éléments à transférer pour ne pas trop faire augmenter la taille du problème interface. La figure 3 illustre le transfert d'éléments finis.

Nous supposons que le sous-domaine  $SD(i)$  est sélectionné, il possède le plus gros volume de calcul estimé.

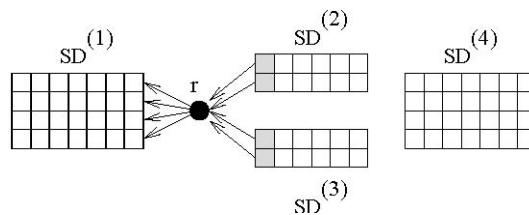


Figure 3. Principe du transfert d'éléments finis entre sous-domaines

Nous agrégeons virtuellement les sous-domaines  $\{SD(2), SD(3)\}$ , voisins de  $SD(1)$ . Nous propageons un front au sein du sous-domaine  $SD(1)$  à partir des éléments frontière de l'agrégat  $\{SD(2), SD(3)\}$ , puis nous sélectionnons les éléments finis à transférer (voir figure 3). Les sous-domaines sont modifiés physiquement puis renumérotés. La procédure est répétée pour obtenir l'équilibrage. Nous disposons d'une décomposition en  $N$  sous-domaines équilibrée en volumes de calcul. Cette technique a déjà été utilisée avec succès lors de travaux précédents [5].

La procédure peut 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. En effet, si l'on suppose 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.

## 4. Résultats

Le domaine d'application retenu est la rade du port de Brest. Le maillage initial comporte un nombre modeste d'éléments afin de servir d'exemple de validation à la version parallèle de Refonde. Le calcul correspond à la prise en compte d'une période unique de houle à  $\lambda=16\text{m}$  pour une direction de propagation  $\theta=130^\circ$ . Deux découpages 'bruts', respectivement en cinq et huit sous-domaines ont été retenus (voir figure 4).

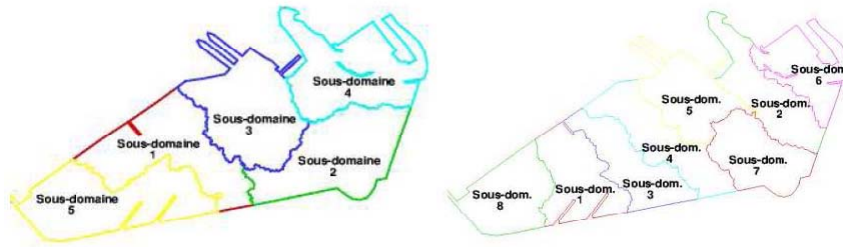


Figure 4. Découpage du port de Brest en cinq et huit sous-domaines

Les caractéristiques des maillages des sous-domaines pour les deux découpages sont regroupés dans les deux tableaux ci-dessous :

Partition 1	Bloc 1	Bloc 2	Bloc 3	Bloc 4	Bloc 5			
NEQ	3919	3854	3741	3847	3742			
Partition 2	Bloc 1	Bloc 2	Bloc 3	Bloc 4	Bloc 5	Bloc 6	Bloc 7	Bloc 8
NEQ	2493	2533	2203	2582	2515	2239	2391	2444

Les temps de calcul mesurés et nombre d'opérations estimées sont donnés figure 5.

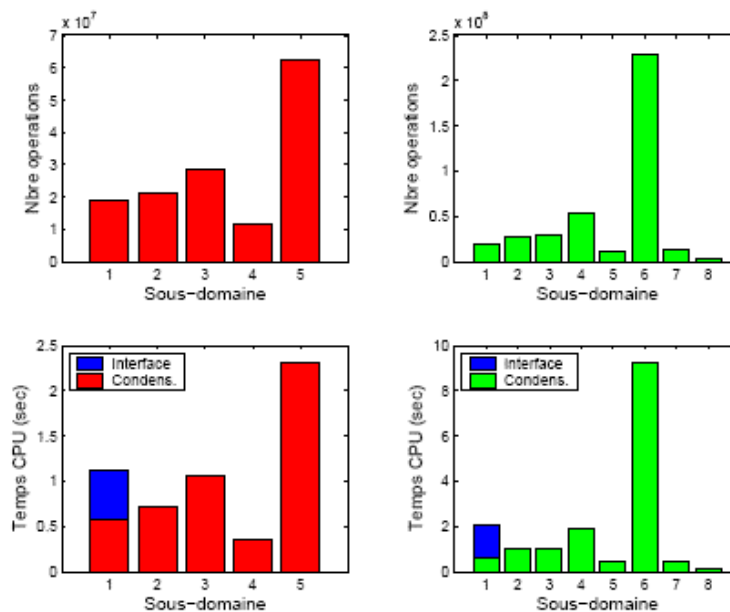


Figure 5. Partition cinq blocs (rouge) -partition huit blocs (vert)

Le nombre d'opérations est ici estimé et basé sur la connaissance de la hauteur moyenne de colonne de la matrice [K](stockage partiel de type *SKYLINE*, voir [6]) du bloc considéré :

$$\text{Nbre opérations} = \frac{1}{2}NEQ * \bar{H}^2 + 2 * NEQ * \bar{H} + NEQ. \quad [5]$$

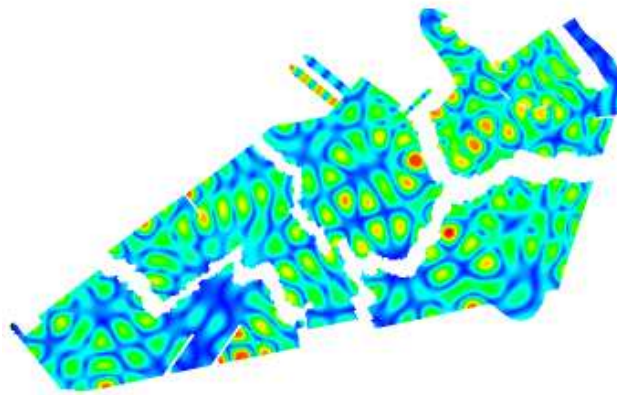
où NEQ est le nombre d'inconnues et H la hauteur moyenne des colonnes de stockage de la matrice. Le premier terme est l'estimation du nombre d'opérations pour effectuer la triangularisation du système, le second terme correspondant au nombre d'opérations pour le résoudre.

Deux remarques peuvent être formulées quant aux résultats obtenus :

1. malgré des volumes de données assez proches, les deux découpages montrent des disparités importantes en temps de calcul sur l'ensemble des sous-domaines;

2. la forte corrélation qui existe entre le nombre d'opérations estimées et le temps mesuré, renforce l'intérêt de coupler la technique de découpage avec cet estimateur pour espérer obtenir à l'équilibre, des temps mesurés quasiment égaux.

À titre de comparaison, le calcul mono-bloc nécessite sur la même machine (Pentium IV à 2GHz) 0.39 s pour la phase d'assemblage (contre 0.05 à 0.07 s en moyenne pour l'approche calcul distribué) et 1.75 s pour la phase de résolution. L'efficacité de l'approche par le calcul distribué nécessite donc de coupler l'utilisation d'une technique de découpage avec un algorithme de prédiction correct du temps de calcul pour espérer obtenir une homogénéisation des temps mesurés.



**Figure 6.** Calcul multiblocs (5) pour une houle à  $\lambda=16m$  et  $\theta=130^\circ$

Le résultat du calcul de la partition à cinq blocs est illustré figure 6. Les blocs ont volontairement été 'détachés' afin de bien les distinguer, les solutions aux frontières étant parfaitement continues entre deux blocs voisins.

### **5. Conclusions et perspectives**

Les résultats présentés servent pour la validation de l'approche et d'exemple de démonstration. Un maillage  $1.8 \cdot 10^6$  de éléments est actuellement en cours de traitement, cas sur lequel l'approche proposée dans ce papier peut trouver toutes ses justifications. Un travail d'intégration de l'estimateur du nombre d'opérations dans l'algorithme de découpage est en cours.

Des résultats antérieurs [5] encourageants permettent d'envisager une optimisation globale du temps de calcul sur l'ensemble des processeurs utilisés et par la suite, une étude du nombre optimal de sous-domaines à fournir pour viser une utilisation 'intelligente' des ressources de calcul.

Ce travail est actuellement le fruit d'une collaboration entre une équipe d'informaticiens et une équipe de numériciens et répond à la démarche formulée autour du projet de plate-forme PILCAD (Plate-forme Inter-LABoratoire de CALculs Distribués) soutenu par un abondement ANVAR.

## **6. Références**

1.F. Ropert. *Modélisation numérique du comportement hydraulique des caissons Jarlan*. PhD thesis, Université de Technologie de Compiègne, Juin 1999.

2.Y. Escaig and G. Touzot. Application des méthodes de décomposition de domaines au calcul parallèle. *Revue Européenne des Eléments Finis*, 6(5):589–609, 1997.

3.G. Kapyris and V. Kumar. Metis a software package for partitioning unstructured graph, partitioning meshes, and computing fill-reducing orderings of sparse matrices. Technical report, University of Minnesota, Dept. of Comp. Sci., 1998.

4.B. Hendrickson and R. Leland. The chaco user's guide, version 2.0. Technical report, Sandia National Laboratories, 1995.

5.C. Denis, J.P. Boufflet, P. Breitkopf, A. Rassineux, and M. Vayssade. Equilibrage en volume de calcul pour un solveur parallèle multi-niveau. In *6ème Colloque National en Calcul des Structures*, pages 349–356, 2003. Giens, France.

6.G. Dhatt and G. Touzot. *Une présentation de la méthode des éléments finis*. Maloine S.A. Editeur Paris, 1984.

7.B. Hendrickson. Load balancing fictions, falsehoods and fallacies. *Applied Mathematical Modelling*, 25:99–108, 2000.

8.C. M. Fiduccia and R. M. Mattheyses. A linear-time for improving network partitions. *19th Designation Automation Conference.*, Paper 13.1:175–181, 1982.