



HAL
open science

Sur des approches de haut niveau d'abstraction pour le calcul en mécanique

Dominique Eyheramendy, Fabienne Oudin-Dardun

► **To cite this version:**

Dominique Eyheramendy, Fabienne Oudin-Dardun. Sur des approches de haut niveau d'abstraction pour le calcul en mécanique. 9e Colloque national en calcul des structures, CSMA, May 2009, Giens, France. hal-01413842

HAL Id: hal-01413842

<https://hal.science/hal-01413842>

Submitted on 11 Dec 2016

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.



Distributed under a Creative Commons Attribution 4.0 International License

Sur des approches de haut niveau d'abstraction pour le calcul en mécanique

D. Eyheramendy¹, F. Oudin-Dardun²

¹ Laboratoire de Mécanique et d'Acoustique, CNRS UPR 7051 (Ecole Centrale Marseille)
Technopôle de Château-Gombert, 38 rue F. Joliot-Curie, 13451 Marseille Cedex 20
dominique.eyheramendy@ec-marseille.fr

² Institut Camille Jordan, CNRS UMR5208, (Université de Lyon, Université Lyon 1, ISTIL)
15 Boulevard Latarjet, 69622 Villeurbanne Cedex
fabienne.oudin-dardun@univ-lyon1.fr

Résumé –Aujourd’hui, les problèmes de mécanique font intervenir de plus en plus de phénomènes physiques divers. Ces phénomènes sont plus ou moins couplés et interviennent à des échelles variées. Simultanément, la simulation numérique de problèmes de mécanique se fait de plus en plus souvent sur des systèmes informatiques complexes intégrant le réseau. Il est donc important aujourd’hui de développer des méthodologies capables d’appréhender ces différents niveaux de complexité d’un point de vue global et homogène.

Mots clés – Eléments finis, code multiphysique, Java, parallélisme.

1 Introduction

Aujourd’hui, les problèmes de mécanique font intervenir de plus en plus de phénomènes physiques divers (thermique, chimie,...). Ces phénomènes sont plus ou moins fortement couplés et peuvent intervenir à des échelles variées aussi bien en espace qu’en temps. Dans le même temps, la simulation numérique de problèmes de mécanique se fait de plus en plus souvent sur des systèmes informatiques à processeurs multiples. On peut noter, d’une part, que ces systèmes sont plus accessibles d’un point de vue coût, et d’autre part, plus simples à utiliser d’un point de vue mise en œuvre de simulations. Notre approche consiste donc à des développer des concepts avancés pour appréhender ces différents niveaux de complexité d’un point de vue global et de manière homogène. La programmation orientée objet, au cœur de cette approche, a été appliquée à la méthode des éléments depuis plus de 15 ans (voir par exemple [1] et références). Cette approche informatique est aujourd’hui active et est largement utilisée dans de nombreux domaines du calcul numérique (voir par exemple [2][3][4][5][6][7] et références citées). Dans la première partie, on décrit une approche à objets en Java permettant de gérer efficacement les problèmes à champs multiples dans un cadre éléments finis. On décrit un nouveau principe de programmation, le principe de cohérence global/local dans un contexte éléments finis. Dans une deuxième partie, l’intégration d’outils de simulation au travers du réseau, permettant l’accès aux ressources informatiques diverses, et ce dans des approches intégrées est décrite.

Eléments finis à champs multiples pour multiphysiques couplées

1.1 Architectures de base pour champs multiples

Depuis les années 1990, de nombreuses formes d’architectures logicielles pour les codes éléments finis ont été proposées. Notre approche reprend les principes déjà longuement discutés dans la littérature et générale la description de champs physiques discrets nécessaires à la simulation d’un problème quelconque : champs scalaires, champs vectoriels, champs tensoriels.

L'organisation des données est basée sur une structure multi-niveaux. Chaque niveau est lié au précédent. Comme le montre la Figure 1, les données du problème éléments finis sont basées sur 3 niveaux de description. Les géométries de base telles que les lignes ou points (« Geometry level » sur la Figure 1) permettent de définir la topologie générale du domaine. La description du maillage (« Mesh level » sur la Figure 1) est basée sur un ensemble de données décrivant les éléments géométriques. Plusieurs maillages peuvent s'appuyer sur la même géométrie. A ce niveau, le maillage est décrit comme un ensemble de nœuds et d'éléments. Seule la cinématique de base de l'élément est donnée à ce niveau. Toutes les données cinématiques nécessaires au niveau des calculs des formes élémentaires de la méthode des éléments finis seront fournies dans un objet « cinématique » donné par la géométrie élémentaire. Le champ au sens des éléments finis est décrit de manière analogue par un ensemble de champs élémentaires. La description d'un champ peut se faire soit en décrivant les données aux nœuds, soit aux points de Gauss. Cette dernière idée se retrouve dans [8]. La description du champ élémentaire suit le même schéma multi-niveau que le champ. Comme le montre la Figure 1, la définition du champ élémentaire est basée sur la description de l'élément géométrique. Ainsi, un même élément géométrique peut servir de base à plusieurs champs élémentaires. Il peut être composé de nœuds ou de point de Gauss (données aux nœuds ou points de Gauss). Le passage d'informations entre les différents niveaux (champ élémentaire/géométrie élémentaire, champ/maillage,...) peut se faire grâce à une gestion de liens entre les entités des différents niveaux (direction verticale dans les niveaux de la Figure 1). Par exemple, un champ élémentaire connaît la géométrie élémentaire sur laquelle il s'appuie. Il peut ainsi avoir accès à la définition cinématique de l'élément, principalement les fonctions d'interpolations correspondantes et leurs dérivées partielles successives. Il est important de noter que dans une telle organisation, les données ne sont ainsi jamais dupliquées. Cette organisation globale permet ainsi au développeur de se concentrer sur la physique qui couple ces différents champs dans le contexte d'un problème complexe et sur les algorithmes de résolutions globaux. Cette organisation des données permet une gestion assez naturelle de champs multiples dans un contexte multiphysique. De plus, cette organisation permet de faciliter l'implémentation d'algorithmes de résolution en proposant des niveaux globaux et locaux des données éléments finis. Chacune des données est impliquées dans la résolution algorithmique globale (mise en forme des membres de gauche et de droite des systèmes linéaires résolus successivement).

1.2 Application à un problème d'écoulement souterrain

1.2.1 Formulation éléments finis d'un problème d'écoulement souterrain à surface libre

On considère un problème de suintement au travers d'un milieu poreux présenté dans [9]. On considère l'écoulement au travers d'une digue d'un fluide incompressible dans un milieu poreux comme le montre la Figure 2. Le milieu est supposé être soit sec, soit humide (saturé). La capillarité, la saturation partielle et l'évaporation sont négligées. La surface libre, ligne ECD, est composée de la ligne séparant milieux humide Ω_w et sec Ω_d , ligne ED, de la ligne de sol humide, et de la ligne CD. Le flux au travers de la surface libre est nul. Les équations du problème sont données Figure 2. La difficulté du problème est de localiser la surface libre. Un problème variationnel simple est proposé dans [10]. Il s'agit de trouver la pression étendue au domaine complet $p_\varepsilon \in P = H^1(\Omega_w)$ tel que $\forall q \in H_0^1(\Omega_w)$, la relation suivante soit satisfaite :

$$\int_{\Omega} \frac{1}{\gamma} k \nabla p_\varepsilon \nabla q d\Omega = - \int_{\Omega} k H_\varepsilon(p) \nabla y \nabla q d\Omega \quad (1)$$

Où p_ε est l'extension de la pression sur le domaine total et vaut 0 dans le domaine sec, et H_ε est une fonction de Heaviside dépendant de la pression et d'un paramètre ε . Une méthode itérative de type Newton est nécessaire à la résolution du problème discret, la surface libre et donc le domaine étant inconnu a priori (voir [11] pour la définition du problème discret).

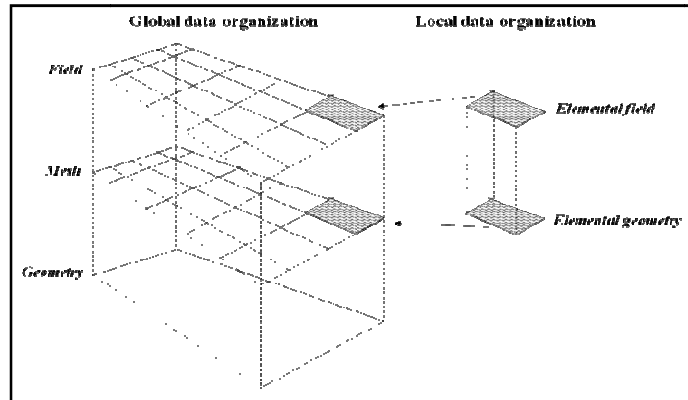


Figure 1 – Organisation multi-niveaux des champs et géométries

Hauteur piezométrique Φ et p pression:

$$\phi = \frac{P}{\rho g} + y = \frac{p}{\gamma} + y \text{ avec } p \geq 0 \text{ in } \Omega_w \text{ et } p = 0 \text{ in } \Omega_d$$

où ρ densité et γ masse spécifique du fluide.

En régime établi:

$$\nabla \cdot k \cdot \nabla \phi = 0 \text{ in } \Omega \text{ où } k \text{ tenseur de perméabilité.}$$

Avec les conditions de bord:

- $p = \gamma(H_1 - y)$ on AF
- $p = \gamma(H_2 - y)$ on BC
- $p = 0$ and $n \cdot k \cdot \nabla \phi = 0$ on ED
- $p = 0$ and $n \cdot k \cdot \nabla \phi = 0$ on CD
- $n \cdot k \cdot \nabla \Phi = 0$ on AB

Où n normale extérieure au domaine Ω_w .

Figure 2 – Définition d'un problème d'écoulement dans une digue.

1.2.2 Une implémentation du problème d'écoulement au travers d'un sol en Java

L'implémentation est basée sur l'objet « champ » évoqué dans la partie 0. Dans ce problème, un seul champ inconnu est nécessaire, un champ scalaire, le champ de pression étendu à tout le domaine. Le champ est automatiquement construit sur le domaine discrétisé (éléments géométriques), ses données étant utilisées pour le calcul des matrices élémentaires. La formulation est implémentée dans une classe appelée **PressureDarcyPenalizedProblem**. La classe hérite de sa superclasse **Formulation** le comportement général d'une formulation, et en particulier la construction automatique d'un champ discret sur la géométrie. Les méthodes principales consistent à définir les champs du problème (ici la pression) et à construire les éléments correspondant. Ceux-ci sont des instances de la classe interne **PressureDarcy** dans laquelle est défini le calcul des données élémentaires du problème avec couplages des champs. Nous appelons cette méthode de gestion de la cohérence de la formulation globale et des données élémentaire basée sur une classe interne « principe de cohérence locale/globale ».

1.2.3 Applications numériques

On présente sur la Figure 3 deux exemples de calcul d'écoulement au travers d'une digue. Chaque calcul montre une solution de surface libre. Le premier calcul montre l'influence d'un drain, le second celle d'une surface imperméable sur la partie amont. Les calculs sont réalisés avec des éléments quadrangulaires linéaires et ne nécessite qu'une dizaine d'itérations.

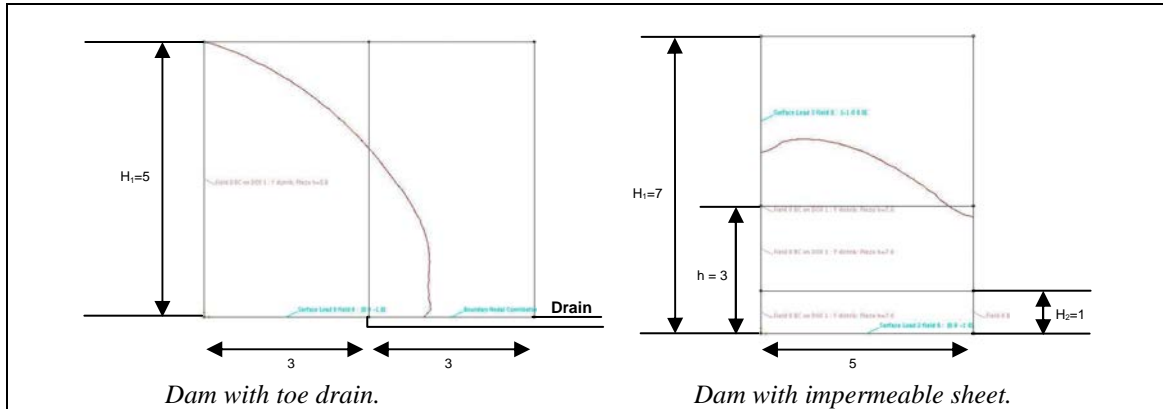


Figure 3 –Exemples d’écoulements dans une digue.

2 Une approche intégrée pour méthodes de décomposition de domaines en mémoire distribuée ou partagée

2.1 Généralités sur les méthodes de décomposition de domaine

Un algorithme de résolution de problème basé sur une méthode de décomposition de domaine (Schwarz avec ou sans recouvrement, FETI,...) peut se concevoir comme une succession de problèmes définis sur des domaines différents, nécessitant des échanges de données. Sur la Figure 4, on montre un exemple d’algorithme de Schwarz : jusqu’à convergence on résout en alternance les problèmes de type 0, puis ceux de type 1. Après chaque groupe de résolution des échanges de données au bord sont réalisées. Chaque résolution de problème est portée par un processus indépendant (classe **Thread**). Cet algorithme est tout à fait général et ne présuppose pas que les domaines soient situés sur le même espace mémoire. Dans l’algorithme de la Figure 4, le domaine peut être situé sur un processeur soit distant soit local comme le montre la Figure 5. La seule différence se fait sur le domaine de calcul est ou non distribué ; mais l’algorithme de résolution est le même dans les deux contextes. On a ainsi une approche globale de programmation dans des contextes de distributions de données différents. Dun point de vue pratique dans le cas des calculs distribués, le package RMI (Remote Method Invocation) est utilisé pour l’envoi de message sur objets distants (distribués), et les classes sockets de Java pour les échanges de données, plus efficaces que les équivalentes RMI. On peut ainsi de manière simple, naturelle et dans un même concept objet en Java gérer des calculs distants ou non. Une description plus approfondie peut être trouvée dans [12].

<pre> public void solve() { // ... INITIALIZING OF THE SOLVER for(int it = 0 ; (it < maxIteration) && (! iteration.converged()) ; it++) { for(int color = 0 ; color < 2 ; color++) { int anteColor = (color + 1) % 2 ; //***** solution ***** // SOLUTION BLOCK // ... //***** exchanges ***** // EXCHANGES BLOCK // ... } // FINALIZING THE ITERATIONS // ... } } </pre>	<pre> //***** solution ***** Thread[] solutionThreads = this.initializeSolutionThreads (color) ; for(int i = 0 ; i < solutionThreads.length ; i++) solutionThreads[i].start () ; try{ for(int i = 0 ; i < solutionThreads.length ; i++) solutionThreads[i].join () ; }catch(InterruptedException e) System.out.println (e) ; </pre>
<pre> Thread threadii = new Thread (new Runnable () { int number = Ni ; public void run() { domains[number].solveSchwarz () ; } }) ; </pre>	

Figure 4 –Algorithme de Schwarz avec recouvrement.

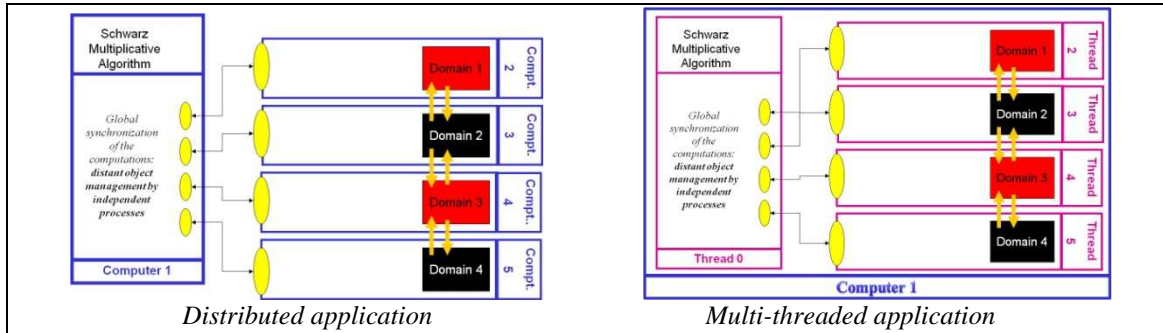


Figure 5 –Schéma de résolution à mémoire partagé et distribué.

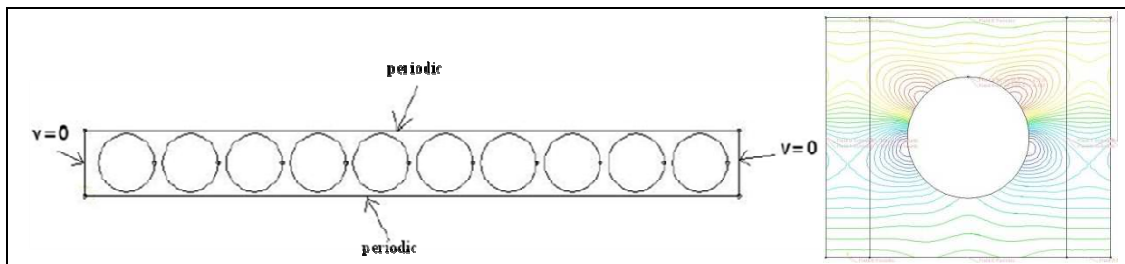


Figure 6 –Ecoulement d'un fluide au travers d'un ensemble de cylindre – Iso-pressions.

2.2 Application à un écoulement de Navier-Stokes dans un réseau de cylindre résolu par une méthode de Schwarz avec recouvrement

On montre sur la Figure 6, un exemple de calcul réalisé avec le même algorithme de Schwarz en version distribué ou locale, mais à chaque fois à processus multiples. Il s'agit du calcul d'un écoulement périodique de Navier-Stokes au travers d'une série de cylindres. Une formulation éléments finis stabilisée par ajout de terme de type SUPG et PSPG. Les courbes d'iso-pressions sont données. On se référera à [12] pour plus de détails.

3 Conclusion

Dans cette communication, nous avons présenté des éléments d'implémentation à objets pour problèmes à champs multiples dans le cadre des éléments finis. Une approche unifiant dans un même contexte à objet des implémentations parallèles dans un contexte de mémoire partagé ou distribué. Ces éléments sont présentés ici dans une approche en Java. D'autres types d'approches informatiques permettent des structurations de code aussi riches (voir [3] par exemple). Les approches plus classiques en calcul parallèles sont basées sur des librairies type MPI et OpenMP qui nécessitent donc d'appréhender plusieurs méthodologies et technologies ; cet aspect n'existe pas dans notre approche en Java. Cela constitue un point important de notre travail. Les prochaines avancées en matière d'architecture logicielle consistent à intégrer des concepts symboliques au sein des formulations numériques. Ce type d'approche a à ce jour encore été peu étudiée mais permet d'apporter de la modularité et de la souplesse dans certains points précis d'une approche numérique : forme variationnelle, schéma d'intégration numérique,...

Références

- [1] Y. Dubois-Pelerin and Th. Zimmermann, Object-oriented finite element programming: III - An efficient implementation in C++, Computer Methods Appl. Mech. Eng. 108/2 (1993) 165-183.
- [2] D. Eyheramendy and F. Oudin, Advanced object-oriented techniques for coupled

- multiphysics, In *Civil Engineering Computation: Tools and Techniques*, Ed. B.H.V. Topping, ©Saxe-Cobourg Publications, ISBN 978-1-874672-32-6, pp. 37-60, 2007.
- [3] Mackie, R.I., *Programming Distributed Finite Element Analysis: An Object Oriented Approach*", Saxe-Cobourg Publications, ISBN 978-1-874672-31-9, 2007.
 - [4] W.B. VanderHeyden, E.D. Dendy and N.T. Padiyal-Collins, CartaBlanca-a pure-Java, component-based systems simulation tool for coupled nonlinear physics on unstructured grids-an update, *Concurrency and Computation: Practice and Experience* 15 (2003) 431-458.
 - [5] D. Eyheramendy, *Towards global computational frameworks for multi-physics*, 7th World Congress on Computational Mechanics, Los Angeles USA, 2006.
 - [6] D. Eyheramendy, "High abstraction level frameworks for the next decade in computational mechanics", *Innovation in Engineering Computational Technology*, Chapter 3, Eds. B.H.V. Topping, G. Montero and R. Montenegro, ©Saxe-Cobourg Publications, pp. 41-61, 2006.
 - [7] D. Eyheramendy, *Advanced object models for mathematical consistency enforcement in scientific computing*, WSEAS Transactions on Mathematics, 4(4) 457-463, 2005.
 - [8] P. Verpeaux, T. Charras et A. Millard, "CASTEM 2000 : une approche moderne du calcul de structures", *Calcul des structures et intelligence artificielle*, Vol. 2, J.M. Fouet, P. Ladevèze et R. Oyahon Eds., Pluralis, 1988.
 - [9] S.J. Lacy and J.H. Prevost, "Flow through porous media: A procedure for locating the free surface", *Int. J. Num. An. Meth. Geom.*, 11, 585-601, 1987.
 - [10] H. Brezis, D. Kinderlehrer and G. Stampacchia, "Sur une nouvelle formulation du problème de l'écoulement à travers une digue", *C.R. Acad. Sci. Paris*, 287 (ser. A), 711-714, 1978.
 - [11] D. Eyheramendy and D. Guibert, *A Java Approach for Finite Elements Computational Mechanics*, ECCOMAS 2004, Jyvaskyla, Finland (2004).
 - [12] D. Eyheramendy, D. Loureiro, F. Oudin-Dardun, "An integrated object-oriented approach for parallel CFD", 20th Int. Conf. on Paral. Comp. Fluid Dyn., Lyon, May 19th-22nd 2008.