

Mini-symposium SOLVER
Nouveaux champs de recherche et outils associés pour la
résolution de grands systèmes linéaires creux

Résumé

La résolution précise et efficace de grands systèmes linéaires creux et sans structure particulière est un vrai enjeu pour bon nombre de logiciels de simulation en physique des champs. Cette étape peut réellement être sur leurs chemins critiques pour la conduite de certaines typologies d'études. D'où la nécessité d'outils généralistes, robustes, évolutifs et pérennes pour capitaliser les algorithmes et les recherches dans le domaine.

Dans ce mini-symposium SOLVER, on se propose de donner la parole aux développeurs et aux chercheurs de quelque uns de ces outils. Il s'agit, pour la plupart, de produits développés ou co-développés par des organismes de recherche français ; souvent en partenariat avec les meilleurs organismes internationaux du domaine et avec le soutien d'industriels. Seront ainsi présentés les bibliothèques MUMPS, PaStiX/MaPHyS, Scotch/PaMPA, ABCD-Solver et HPDDM, ainsi que des travaux de recherches associés.

Organisateur(s)

1. **Olivier Boiteau**, EDF Lab Clamart.

Liste des orateurs

1. **Patrick Amestoy ou Jean-Yves L'Excellent** , au nom de l'équipe de développement, CERFACS, CNRS, ENS Lyon, INPT, INRIA Lyon, Univ.Bordeaux.
Titre : L'outil MUMPS : résolution de systèmes linéaires creux par méthode directe.
2. **Louis Poirel**, INRIA Bordeaux.
Titre : Préconditionneur multiniveau pour le solveur par décomposition de domaine algébrique MaPHyS.
3. **Cédric Lachat**, INRIA Bordeaux.
Titre : Manipulation parallèle de graphes et de maillages avec Scotch et PaMPA.
4. **Daniel Ruiz** , au nom de l'équipe de développement, CERFACS, CNRS, INPT.
Titre : L'outil ABCD-solver : résolution de systèmes linéaires creux par méthode hybride.
5. **Ryadh Haferssas** , Pierre Jolivet et Frédéric Nataf, UPMC/INRIA Rocquencourt.
Titre : Domain Decomposition Methods within HPDDM framework.

Olivier Boiteau, EDF R&D/SINETICS/I23 1 Av. du Général de Gaulle 92140 CLAMART, olivier.boiteau@edf.fr

Patrick Amestoy ou Jean-Yves L'Excellent au nom de l'équipe de développement de MUMPS, CERFACS, CNRS, ENS Lyon, INPT, INRIA Lyon, Univ.Bordeaux, Patrick.Amestoy@enseeiht.fr, Jean-Yves.L.Excellent@ens-lyon.fr

Louis Poirel, INRIA Bordeaux – Sud-Ouest, 200 avenue de la vieille tour 33405 TALENCE, louis.poirel@inria.fr

Cédric Lachat, INRIA Bordeaux – Sud-Ouest, 200 Avenue de la Vieille Tour, 33405 TALENCE Cedex, cedric.lachat@inria.fr

Daniel Ruiz, au nom de l'équipe de développement de ABCD-solver, CERFACS, CNRS, INPT, Daniel.Ruiz@enseeiht.fr

Ryadh Haferssas, UPMC (Laboratoire Jacques-Louis Lions) / INRIA Rocquencourt(Alpines), 04 place Jussieu 75005 Paris cedex, ryadh.haferssas@ljl11.math.upmc.fr

Introduction

La résolution précise et efficace de grands systèmes linéaires creux et sans structure particulière est un vrai enjeu pour bon nombre de logiciels de simulation (notamment) en physique des champs : CFD, mécanique des structures, électromagnétisme, thermique, acoustique, étude des matériaux... Les consommations en temps et en mémoire de cette étape sont dimensionnantes pour le calcul complet. Cependant elle est souvent méconnue car elle est la dernière strate d'algorithmes englobants plus "métiers" : problèmes non linéaires, recherches de modes propres, intégrations d'EDO...

Pourtant, d'un point de vue pratique, les ressources machines à allouer et le temps d'attente du résultat final sont très liés aux seules performances de cette étape. La faisabilité du calcul complet peut même dépendre de son bon déroulement. La difficulté de résolution de tel ou tel type de système linéaire peut ainsi brider voire disqualifier le recours à certaines méthodes d'analyse!

C'est aussi évidemment un rouage essentiel dans le schéma de parallélisation d'un logiciel. Certains codes sont ainsi complètement organisés en fonction de cette brique logicielle : mises en données, paradigme de parallélisation, organisation des algorithmes, choix des modélisations déployées en amont...

Bref, cette étape peut réellement être sur le chemin critique d'un code de simulation ou d'une typologie d'étude. D'où la nécessité d'outils généralistes robustes, évolutifs et pérennes pour capitaliser les algorithmes et les recherches dans le domaine. En outre, ces bibliothèques peuvent être des vecteurs très efficaces pour, durablement et concrètement, transférer l'innovation vers les autres équipes de recherche, l'industrie, les éditeurs de logiciels... Ces synergies constituant alors un terreau favorable (et gratifiant) pour lancer des collaborations, améliorer ces outils en les confrontant à une plus large variété de problèmes et d'utilisateurs et, bien sûr, nourrir d'autres projets de recherche...

Dans ce mini-symposium SOLVER, on se propose de donner la parole aux développeurs et aux chercheurs d'un certain nombre de ces outils. Compte-tenu du format limité de ce symposium, il nous est bien sûr impossible d'être exhaustif. On a toutefois essayé de couvrir un large spectre : solveur direct, préconditionneur, solveurs hybrides algébrique et par DD, outils connexes de partitionnement-renumérotation-équilibrage... Il s'agit, pour la plupart, de produits développés ou co-développés par des organismes de recherche français ; souvent en partenariat avec les meilleurs organismes internationaux du domaine et avec le soutien d'industriels. Seront ainsi présentés les produits MUMPS, PaStiX/MaPHYs, Scotch/PaMPA, ABCD-Solver et HPDDM, ainsi que des travaux de recherches associés.

1 L'outil MUMPS : résolution de systèmes linéaires creux par méthode directe [1]

MUMPS (pour MULTifrontal Massively Parallel Solver) est un logiciel libre diffusé sous licence Cecill-C pour la résolution parallèle de systèmes linéaires creux par des méthodes dites directes. Ces méthodes, de par leur robustesse et leur efficacité sont au cœur de nombreuses approches basées sur la simulation numérique dans divers domaines applicatifs. MUMPS résout des systèmes d'équations en factorisant la matrice creuses sous la forme LU ou LDL^T , puis en appliquant des résolutions triangulaires.

Une des originalités de l'approche retenue est de ne pas avoir sacrifié les aspects numériques et notamment le pivotage pour des raisons de performances ou de parallélisation sur calculateurs à mémoire distribuée. De plus, l'ensemble des fonctionnalités disponibles, qui résulte de l'intégration aussi systématique et cohérente que possible de nos travaux de recherche (10 thèses, 2 habilitations) dans une seule et même plate-forme logicielle, est très étendue et permet une large couverture applicative. Parmi ces fonctionnalités, on peut citer :

- différents types de systèmes : symétriques définis positifs, symétriques généraux (indéfinis) ou non symétriques,
- différents formats d'entrées : matrices assemblées ou exprimées comme une somme de matrices élémentaires provenant typiquement de problèmes d'éléments finis, matrices centralisées sur un processeur ou distribuées sur les processeurs,
- différentes précisions arithmétiques pour les données et le calcul (simple ou double précision, réel ou complexe),
- détection des pivots nuls et estimation d'une base du noyau pour le cas des matrices singulières,
- prétraitements et mises à l'échelle (parallélisation de certains prétraitements),
- factorisation partielle et calcul d'un complément de Schur ; condensation d'un second membre sur une interface,

- seconds membres denses ou creux, multiples, solution centralisée ou distribuée,
- pivotage partiel avec seuil,
- approche asynchrone avec recouvrement des calculs et des communications,
- ordonnancement dynamique et distribué des tâches de calcul pour permettre un bon équilibrage de charge en présence de pivotage numérique ou de performances difficilement prédictibles de la machine,
- stockage sur disque des facteurs quand la mémoire disponible n’est pas suffisante,
- calcul de l’inertie, du déterminant, possibilité de ne pas stocker les facteurs.

Le parallélisme est exploité grâce à la librairie de transfert de messages MPI (Message Passing Interface) complété par l’utilisation de directives OpenMP et de bibliothèques BLAS multi-cœurs qui permettent d’exploiter efficacement une large gamme d’architectures matérielles. MUMPS est écrit en Fortran 95 et en C et il existe des interfaces depuis Fortran, C, Matlab, Julia, Scilab. Utilisé par des milliers d’utilisateurs et redistribué dans des logiciels publics ou commerciaux, MUMPS est également redistribué dans des paquets Linux ou dans des outils comme PETSc. Il est notamment utilisé quotidiennement chez EDF dans Code_Aster.

Aujourd’hui, un des problèmes critiques issu des besoins applicatifs consiste à résoudre des systèmes linéaires de taille de plus en plus grande, tout en gardant performance et stabilité numérique. Cela nous amène entre autres à mener des recherches sur :

- (i) la scalabilité mémoire sur architectures à grands nombres de cœurs
- (ii) l’étude et la parallélisation pour architectures mixtes distribuées/multicœurs/GPU et la prise en compte de l’architecture de la machine
- (iii) une nouvelle classe de solveurs directs multifrontaux pour les matrices de très grande taille (plusieurs milliards d’inconnues) provenant de la discrétisation d’équations aux dérivées partielles basée sur la représentation de rang faible (dite Low-Rank) de matrices pleines intermédiaires

MUMPS est et a été développé par des chercheurs appartenant à différentes institutions : l’INPT-IRIT, le CNRS, l’Université de Bordeaux, l’ENS Lyon, Inria, et le CERFACS. Les premières versions datent du projet européen PARASOL (Esprit IV, projet LTR 20160, 1996-1999). La dernière version (février 2015) constitue une étape majeure dans le développement du produit et est concomitante à la mise en place d’un consortium d’industriels qui participent au financement du projet sur les aspects ingénierie, support, et aide au transfert des recherches dans l’outil (premiers adhérents : EDF, Michelin, Altair, LSTC, Siemens).

Plus d’informations sur le logiciel MUMPS sont disponibles sur la page [1], qui contient également un ensemble de références sur les travaux de recherche récents qui, avec le retour des utilisateurs, permettent de faire évoluer l’outil MUMPS.

2 Préconditionneur multiniveau pour le solveur de décomposition de domaine algébrique MaPHyS [2] [3]

Dans cet exposé nous présenterons les principales idées qui sous-tendent les méthodes de résolution de systèmes linéaires par décomposition de domaine algébrique mises en œuvre dans l’implantation parallèle du logiciel MaPHyS [4]. Les idées de base s’inspirent des méthodes de décomposition de domaine par sous-structuration originellement introduites en mécanique des structures ; celles-ci consistent à réaliser une élimination directe partielle des inconnues associées à l’intérieur des sous-structures et à résoudre par une méthode de Krylov préconditionnée le problème condensé aux interfaces (complément de Schur). L’un des préconditionneurs utilisés pour la résolution du complément de Schur est une forme algébrique d’une méthode de type Schwarz additif. Pour des systèmes linéaires issus de la discrétisation d’EDP elliptiques, ce préconditionneur essentiellement local ne passe pas à l’échelle et le nombre d’itérations croît avec le nombre de sous-domaines. Cette faiblesse numérique rend nécessaire l’introduction d’un mécanisme global tel qu’une grille grossière algébrique ou des approches par déflation ou augmentation d’espace. La partie significative de l’exposé sera dédiée à la présentation de la construction de l’espace de déflation. Les caractéristiques principales des briques logicielles élémentaires tel que le solveur direct creux supernodal PaStiX [3, 5, 6] seront brièvement décrites.

3 Manipulation parallèle de graphes et maillages avec Scotch et PaMPA [7] [8]

De nombreuses simulations numériques ont pour support des maillages non structurés. Ces maillages peuvent eux-mêmes être modélisés sous formes de graphes, dont les sommets représentent certaines entités du maillage (éléments, faces, arêtes, noeuds, etc.) et les arêtes, les liaisons individuelles entre ces entités. La nécessité de réaliser des simulations de précision et de taille croissantes fait que les maillages considérés ne peuvent plus être stockés dans la mémoire d'un unique ordinateur. Il est donc nécessaire de distribuer les données décrivant les maillages sur plusieurs noeuds de traitement à mémoire distribuée, et de gérer tous les surcoûts induits par cette distribution, tels que la duplication, entre les bords des sous-domaines du maillage affectés à chaque noeud de traitement, des informations nécessaires aux calculs des valeurs physiques portées par les entités du maillage, ou encore la synchronisation entre les nouvelles valeurs physiques calculées au sein d'un sous-domaine et les multiples copies de ces valeurs situées sur des noeuds de traitement voisins.

Qui plus est, la pertinence des simulations informatiques dépend fortement de la qualité des maillages. Pour améliorer cette qualité, on recourt à des techniques de remaillage, qui visent à rendre le maillage plus dense aux endroits où les phénomènes physiques critiques se produisent et évoluent au cours du temps. Il est donc nécessaire d'offrir aux scientifiques des méthodes parallèles de remaillage adaptées aux architectures à mémoire distribuée.

PaMPA (pour "Parallel Mesh Partitioning and Adaptation") est une bibliothèque dédiée à la manipulation de maillages non structurés distribués sur les processeurs d'une machine parallèle à mémoire distribuée. Son but est de soulager les auteurs de logiciels de simulation, des tâches répétitives et sujettes à erreurs consistant à écrire à chaque fois les routines de service pour la définition des structures de données des maillages, l'échange des données entre noeuds de traitement, le remaillage et la redistribution des données du maillage aux fins d'équilibrage de la charge sur la machine parallèle.

Certaines des fonctionnalités offertes par PaMPA, à savoir le partitionnement de maillages et le remaillage, nécessitent l'utilisation de la bibliothèque Scotch. Cette bibliothèque, dédiée au partitionnement de graphes, est utilisée pour placer les entités des maillages sur les noeuds de traitement des machines parallèles, afin d'équilibrer la charge de calcul et d'accroître la localité des communications. Scotch permet en effet de minimiser une métrique qui prend en compte les coûts de transfert au sein du réseau de communication reliant, éventuellement de façon hétérogène, les noeuds de traitement de la machine parallèle.

L'objectif de cet exposé est de présenter l'architecture et les principes de fonctionnement des outils PaMPA et Scotch. En partant des besoins exprimés par la communauté scientifique utilisant les maillages non structurés, nous exposerons en quoi ces outils permettent de répondre à nombre de ces besoins.

4 L'outil ABCD-solver : résolution de systèmes linéaires creux par méthode hybride [9]

Nous considérons l'utilisation de ABCD-solver, l'acronyme signifiant "Augmented Block Cimmino Distributed solver", qui est une méthode itérative de type "hybride" pour la résolution de grands systèmes linéaires creux sur calculateurs parallèles. La méthode est basée sur des techniques de projection par blocs de lignes (du type des méthodes itératives de Cimmino ou de Kaczmarz), et offre la possibilité d'exploiter trois niveaux de parallélisme différents et complémentaires. La gestion combinée de ces divers niveaux de parallélisme constitue un premier aspect du caractère "hybride" de cette approche. Le deuxième aspect provient d'une fonctionnalité spécifique proposée dans ABCD-solver, à savoir qu'il est possible de plonger le système linéaire à résoudre dans un sur-espace, en incorporant donc des méta-variables et des contraintes additionnelles, pour aboutir au final à une pseudo méthode directe en ce sens que la méthode appliquée à la résolution du problème ainsi augmenté convergera à coup sûr en une seule itération. Cette fonctionnalité ne présente de réel intérêt que dans les cas où le surcoût induit par l'introduction de ces variables et contraintes additionnelles est largement compensé par le gain en itérations relativement à la méthode de Cimmino par bloc classique (c'est à dire ABCD-solver sans l'activation de cette fonctionnalité) pour la résolution du système linéaire considéré. Bien évidemment, ceci dépend de l'application ou de la classe de problèmes considérée, et nous analysons expérimentalement l'utilisation de ABCD-solver pour la résolution de problèmes de grande taille, développés par EDF en particulier pour la simulation

numérique de problèmes d'équations aux dérivées partielles en mécanique des structures ou en mécanique des fluides. Nous comparons aussi les performances de ABCD-solver avec d'autres méthodes disponibles, à la fois sur le plan des performances que sur le plan de la quantité mémoire utilisée, ou encore en termes d'efficacité parallèle sur des calculateurs à hautes performances.

ABCD-solver (release 1.0) est un logiciel libre diffusé sous licence Cecill-C, et est accessible à l'adresse [9]. Ce logiciel a été développé par des chercheurs de l'INPT et de l'IRIT-CNRS, en collaboration avec le CERFACS. La première version du logiciel a été achevée dans le cadre du projet BARESAFE, supporté par l'Agence Nationale pour la Recherche.

5 Domain Decomposition Methods within HPDDM framework [10] [11]

Domain decomposition methods are widely used in applied mathematics and regarded as highly scalable algorithms, alongside multigrid methods. Making those methods scalable to thousands of processors is however not a straightforward task. The aim of this talk is to showcase recent advances in the construction of robust coarse spaces for Schwarz domain decomposition methods. We formulate and analyse a two level preconditioner as well as its implementation inside a C++ open source framework [10] interfaced with different libraries tools such as MUMPS, MKL and Scotch. A broad spectrum of applications will be covered, ranging from a scalar diffusion equation to incompressible linear elasticity equations, and including Stokes problem. Numerical results with hundreds of processes will be provided, clearly showing the effectiveness and the robustness of the proposed approach.

Références

- [1] PRODUIT MUMPS, *site internet*, <http://mumps-solver.org>.
- [2] PRODUIT MAPHYS, *site internet*, <https://project.inria.fr/maphys>.
- [3] PRODUIT PASTIX, *site internet*, <http://pastix.gforge.inria.fr/files/README-txt.html>.
- [4] E. AGULLO, L. GIRAUD, A. GUERMOUCHE, A. HAIDAR, J. ROMAN, *Parallel algebraic domain decomposition solver for the solution of augmented systems*, Advances in Engineering Software, July 2012.
- [5] P. HÉNON, P. RAMET, J. ROMAN, *PaStiX : A High-Performance Parallel Direct Solver for Sparse Symmetric Definite Systems*, Parallel Computing, 28(2) :301–321, January 2002.
- [6] X. LACOSTE, *Scheduling and memory optimizations for sparse direct solver on multi-core/multi-gpu cluster systems*, PhD thesis, LaBRI, Université Bordeaux, February 2015.
- [7] PRODUIT SCOTCH, *site internet*, <https://labri.fr/perso/pelegrin/scotch>.
- [8] PRODUIT PAMPA, *site internet*, <https://project.inria.fr/pampa>.
- [9] PRODUIT ABCD-SOLVER, *site internet*, <http://abcd.enseeiht.fr>.
- [10] PRODUIT HPDDM, *site internet*, <http://github.com/hpddm/hpddm>.
- [11] R.HAFERSSAS, P.JOLIVET AND F.NATAF, *A Robust Coarse Space for Optimized Schwarz methods SORAS-GenEO-2*, submitted, 2015.