Proposition d'ARC – SpaSciAl
Sparse linear Algebra in Scilab

7 janvier 2002

1 Motivation

Les grands challenges passent par la résolution de systèmes linéaires de grande taille, pour lesquels la structure creuse doit être prise en compte: la chimie moléculaire, le global change, le calcul de structures, la simulation du métabolisme humain, ... Ces simulations impliquent aussi le calcul de valeurs propres ou de valeurs singulières, comme par exemple le calcul des fréquences de résonance d'une structure.

L'importance et la variété des domaines d'applications constituent la motivation principale dans la recherche de méthodes de résolution performantes liées aux systèmes linéaires creux (symétriques ou non symétriques, réguliers ou non, carrés ou rectangulaires). Ces systèmes peuvent provenir de la discrétisation par éléments finis ou différences finies d'équations aux dérivées partielles, mais aussi de problèmes d'optimisation, voire d'autres modélisations.

SCILAB est un logiciel développé par l'INRIA dans le projet METALAU; c'est un outil de type MATLAB au succès grandissant, facile d'utilisation, qui contient des centaines de fonctions touchant à de nombreux domaines, et qui est disponible gratuitement. Dans un environnement de type SCILAB, la facilité d'utilisation est primordiale. Compte-tenu de l'augmentation des capacités des stations de travail ou simples PC et de la taille croissante des problèmes à résoudre, les aspects performances sont également à prendre en compte.

Or il apparaît que SCILAB manque d'outils de résolution efficaces pour l'algèbre linéaire creuse (systèmes linéaires, moindres carrés linéaires, problèmes aux valeurs propres, ...).

L'action que nous proposons vise à développer des techniques d'algèbre linéaire creuse et à les rendre accessibles dans SCILAB, comblant ainsi les manques actuels dans les fonctionnalités de SCILAB. Les méthodes choisies respectent à la fois le critère de simplicité d'utilisation et celui de performances.

Dans le cas de problèmes de très grande taille, pour lesquels il n'est pas envisageable d'utiliser une simple station de travail, nous nous intéressons à des méthodes de résolution parallèles. Il s'agit dans ce cadre de rendre de telles méthodes accessibles depuis SCILAB grâce à une approche de type "Grid Computing" (approche client-serveur) qui se situe dans la suite logique des travaux effectués dans le cadre de SCILAB parallèle.

Enfin, SCILAB, ainsi enrichi de ces fonctionnalités pour les systèmes creux, pourra servir de plateforme de valorisation pour les méthodes d'algèbre linéaire creuse développées par les partenaires de l'action. Plus que cela, SCILAB pourra servir de vecteur de diffusion de logiciels pour le creux disponibles gratuitement avec pour but de pouvoir y accéder facilement et de façon cohérente depuis une même plateforme.

2 Situation actuelle

Un premier point a été fait le 7 décembre, lors d'une réunion organisée à l'INRIA Rocquencourt, ayant pour thème Scilab pour le creux. La date a été choisie afin que John Gilbert (Xerox Palo Alto Research Center, California) puisse y participer. John Gilbert a travaillé sur des aspects comparables avec Rob Schreiber et Cleve Moler dans le contexte de Matlab, et il serait désireux de collaborer avec l'INRIA pour le développement de Scilab.

La liste des participants à cette réunion et les transparents correspondant aux présentations effectuées sont disponibles à l'adresse suivante:
`http://www.ens-lyon.fr/~jylexcel/scilab-sparse/`.

Actuellement il existe dans Scilab un support pour le type creux, et un certain nombre de fonctions et méthodes de résolution associées (voir figure 1). Essentiellement, il n'existe que des méthodes directes de résolution basées sur la factorisation de Gauss ou de Cholesky.

À titre de comparaison, le logiciel Matlab permet la manipulation de matrices de type creux et s'est enrichi de nombreuses fonctionnalités dans les dernières versions : factorisation QR, méthodes itératives de résolution, calcul de valeurs et vecteurs propres, calcul de valeurs et vecteurs singuliers, etc.

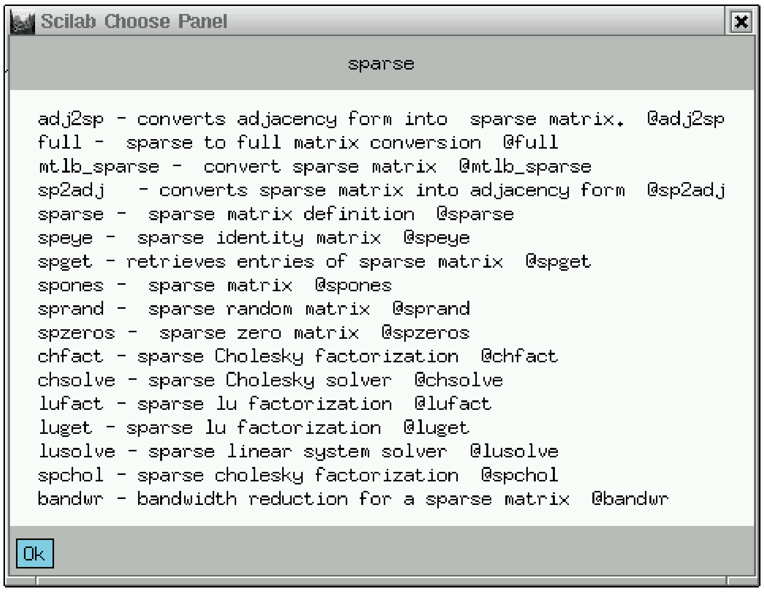

Fig. 1 – Fonctionalités actuelles de Scilab pour le creux.

Il existe des logiciels commerciaux et du domaine public pour le calcul matriciel creux : dans le domaine public, on peut citer PETSC, UMPpack, SuperLU, TauCS, MUMPS, Sparskit, Metis, ARPACK, etc. Certains codes ont des versions parallèles, comme PETSC, SuperLU, MUMPS, etc.

3 Actions proposées

Nous avons identifié cinq actions afin d'atteindre nos objectifs :

- spécifications logicielles et fonctions de base ;
- intégration et développement de méthodes de résolution ;
- interfaçage et développement de méthodes parallèles ;
- algorithmes combinatoires pour les graphes et réordonnancements ;
- validation sur des problèmes de grande taille.

3.1 Spécifications logicielles et fonctions de base

Nous commencerons par identifier de façon précise les fonctionnalités qui doivent être intégrées dans SCILAB et par spécifier une manière consistante de les intégrer, ce qui passe par le choix des types creux à utiliser à l'intérieur de SCILAB, l'amélioration de certaines fonctions de base, l'ajout de procédures pour les interfaces et la surcharge des opérateurs.

1. Choix du type creux et amélioration des performances.

   Le type creux tel qu'il est actuellement défini dans SCILAB est relativement standard. Le plus important est une implantation efficace des opérations de base sur les matrices creuses et une gestion efficace de l'espace mémoire. De manière générale, les outils de manipulations de matrices creuses doivent être améliorés en termes de performances. Une des difficultés, qui existe aussi dans Matlab, est de pouvoir insérer un faible nombre de non zéros dans une matrice creuse sans devoir la réallouer entièrement.

   Un autre exemple concerne la manipulation de matrices triangulaires creuses issues de factorisations de type Gauss. Scilab utilise un pointeur interne, il peut être préférable de choisir une autre représentation.

2. Procédures de conversion et de visualisation.

   Dans tous les cas, il est nécessaire de définir des procédures de conversion vers/depuis d'autres formats pour pouvoir utiliser diverses méthodes de résolution, et des procédures d'import/export de matrices. Par exemple, il est nécessaire de pouvoir utiliser des matrices définies dans des fichiers au format Harwell-Boeing, Rutherford-Boeing, voire Matlab. D'autre part, il apparaît nécessaire d'ajouter des outils de visualisation de matrices creuses.

3. Surcharge de l'opérateur "\"

   Nous souhaitons pouvoir choisir automatiquement une méthode de résolution adaptée lors de l'invocation d'une commande telle que $x = A\backslash b$ (résolution d'un système $Ax = b$). En fonction de propriétés simples de la matrice A (triangulaire, carrée, symétrique ou rectangulaire), un polyalgorithme choisit d'utiliser une résolution triangulaire, une factorisation de Cholesky, une factorisation de Gauss LU ou une factorisation QR.

   Afin d'accroître l'efficacité, l'algorithme pourra garder un pointeur sur la factorisation de manière à n'effectuer qu'une étape de résolution lors d'appels ultérieurs à l'opérateur "\" sur la même matrice.

   Une piste à explorer consiste, sous certaines conditions, à surcharger l'opérateur par un appel à des méthodes itératives ou parallèles comme décrites plus loin.

3.2 Intégration et développement de méthodes de résolution

Nous prévoyons d'intégrer dans SCILAB les méthodes directes et itératives pour résoudre des systèmes linéaires creux ou des problèmes aux moindres carrés creux. Il est prévu également d'intégrer une bibliothèque de résolution de problèmes aux valeurs propres. Les méthodes auront une version pour les matrices de nombres réels et une version pour les matrices de nombres complexes.

1. En ce qui concerne les méthodes directes, la première tâche consiste à intégrer un ou plusieurs nouveaux solveurs creux efficaces. Dans ce contexte, nous utiliserons des codes du domaine public. Par exemple, il apparaît que UMFPACK (Tim Davis) est meilleur que le solveur actuellement disponible dans SCILAB. La deuxième tâche est le développement d'un logiciel effectuant la factorisation QR pour matrices creuses, afin de résoudre des problèmes aux moindres carrés ou des problèmes d'optimisation [2]. Il n'existe pas de code performant dans le domaine public pour cette tâche.
2. En ce qui concerne les méthodes itératives, la première tâche est d'intégrer dans SCILAB la boîte à outils SCILIN [1] développée par le projet ALADIN, tout en l'étendant au cas des nombres complexes. La deuxième tâche est de compléter SCILIN par des méthodes adaptées aux matrices symétriques indéfinies (SYMMLQ, MINRES) [8] et par des méthodes pour les problèmes aux moindres carrés (LSQR) [9]. Ces méthodes seront codées par des scripts écrits en Scilab. En effet, les performances sont liées aux manipulations des matrices creuses qui, elles, sont ou seront optimisées. Pour accélérer les méthodes itératives, il est nécessaire de les préconditionner. Des techniques de factorisation incomplètes de type LU incomplet sont interfacées par SCILIN. Notre objectif est de les compléter par des factorisations incomplètes de type Cholesy incomplet [34] et de mettre au point un préconditionnement de type QR incomplet [34, 9], basé sur la factorisation QR citée ci-dessus. Dans ce cadre, nous garderons pour objectif de permettre à un utilisateur de fournir et de manière simple ses propres préconditionneurs.
3. Nous intégrerons la bibliothèque ARPACK, du domaine public, pour la résolution de problèmes aux valeurs propres. C'est la solution retenue par Matlab pour ce type de problème.

3.3 Interfaçage et développement de méthodes parallèles

Pour les problèmes de très grande taille, nous travaillerons sur des méthodes parallèles de résolution de systèmes linéaires creux.

1. Méthodes directes.
   - En collaboration avec l'ENSEEIHT-IRIT, le projet ReMaP travaille sur la factorisation parallèle en environnement distribué. Cette collaboration a besoin d'être renforcée, et a pour objectif de fournir un logiciel très général, MUMPS [4, 7] [2], pour les problèmes de grande taille. MUMPS permet de résoudre aussi bien des problèmes symétriques que non symétriques avec matrice d'entrée assemblée ou sous forme élémentaire, éventuellement déjà distribuée. L'approche algorithmique

[1] `http://www.irisa.fr/aladin/codes/SCILIN/`
[2] http://www.enseeiht.fr/apo/MUMPS

adoptée est une approche multifrontale avec scheduling dynamique des tâches de calcul que nous souhaitons adapter aux machines actuelles de type réseaux de nœuds SMP.

– L'équipe ALiENor du LaBRI a conçu et développé une chaîne complète pour la résolution parallèle haute performance de très grands systèmes linéaires creux symétriques définis positifs. Celle-ci est basée sur un logiciel de partitionnement et de renumérotation, SCOTCH [31], qui combine finement les approches globales de dissections emboîtées et locales de degré minimum approché. Les différents niveaux de parallélisme induits par cette renumérotation sont alors exploités par le solveur parallèle PaStiX qui est de type supernodal avec agrégation des calculs de contribution, et qui est piloté par une phase de prétraitement statique (distribution mixte par bloc des données et ordonnancement des calculs) tenant compte des caractéristiques de la plate-forme d'exécution (réseau homogène de processeurs identiques et réseaux de nœuds SMP). Une technique d'agrégation partielle permet de maintenir une bonne scalibilité mémoire et la mise en œuvre d'un mécanisme d'Out-Of-Core distribué permettra de traiter des systèmes de très grandes tailles [24, 19, 23, 22, 21, 20]. Ces travaux ont été validés dans le cadre d'un contrat industriel avec le CEA/CESTA (applications de mécanique des structures et d'électromagnétisme avec des systèmes pour des problèmes 3D atteignant 5.3 millions d'inconnues) et font partie de l'avant projet INRIA ScAlApplix en cours d'évaluation (responsable scientifique Jean Roman).

2. Méthodes itératives. Le projet ALADIN a développé une version parallèle de l'algorithme GMRES [16]. Ce logiciel doit être modifié pour intégrer un préconditionnement quelconque et être utilisable facilement dans un environnement distribué.
3. D'autre part, une interface avec des outils de metacomputing comme Netsolve ou DIET est en cours de développement, suite à l'ARC OURAGAN, qui a permis le développement de SCILAB //. Ainsi nous développerons une interface pour pouvoir accéder à ces techniques de résolution sur des serveurs de calcul parallèles distants directement depuis SCILAB. Dans ce contexte, pour les méthodes itératives, nous considérerons la version parallèle de GMRES décrite ci-dessus. Pour les méthodes directes, nous considérerons les solveurs MUMPS, et PaStiX, décrits précédemment, ainsi que SuperLU (développé à l'Université de Berkeley [13, 28, 5, 6]).
4. L'étude, le développement et l'analyse de stratégies de placement statique et de séquencement dynamique adaptées à des architectures de type grappes de nœuds à mémoire partagée dont la charge variable doit être prise en compte sera aussi étudiée sur les solveurs directs.

3.4 Algorithmes de graphes et réordonnancements

Il s'agit là d'un domaine où les fonctionnalités de SCILAB doivent être étendues. Le réordonnancement des matrices creuses permet d'obtenir des gains substantiels tant en place mémoire qu'en temps de calcul. L'efficacité d'une méthode de réordonnancement dépend de différents facteurs (type et implémentation du solveur, type de problème traité, etc.). De tels algorithmes sont utiles aussi bien pour des opérations de base comme les produits de matrices (par exemple, la méthode probabiliste d'Edith Cohen pour prévoir la structure d'un produit de matrices), que pour les méthodes directes (par exemple, la renumérotation du graphe de la matrice pour réduire le remplissage) ou pour d'autres usages (par exemple, le partitionnement de graphes pour dégager du parallélisme).

Nous considérerons les algorithmes suivants, afin de les rendre accessible depuis SCILAB:

- des algorithmes de réordonnancement pour des méthodes directes: il existe des algorithmes classiques dont les plus récentes sont des approches de type (column) minimum degree [3, 12], minimum fill, dissections emboitées, approches hybrides comme celles fournies par ON-METIS [25], PORD [35], ainsi que SCOTCH + HAMD [32] (actuellement utilisé par PaStiX – voir plus haut – et en cours d'intégration dans MUMPS),
- des algorithmes de réordonnancement non symétrique prenant éventuellement en compte les valeurs numériques de type *weighted bipartite matching* [14],
- des algorithmes pour la factorisation symbolique avec prédiction du nombre d'éléments par lignes/colonnes et la construction/la visualisation d'un arbre d'élimination,
- des algorithmes de partitionnement de graphes tels METIS [25] ou SCOTCH [30],
- pour les systèmes très non symétriques, la décomposition de Dulmage-Mendelsohn (composantes fortement connexes et bipartite matching) et les graphes avec fusion de ligne (pour prédire les resources dans le cas de systèmes non symétriques).

3.5 Validation sur des problèmes de grande taille

Les méthodes de résolution intégrées dans SCILAB seront validées sur des problèmes issus de contextes industriels.

- Un ensemble de problèmes tests sont disponibles sous forme de matrices dans les librairies Rutherford-Boeing ou PARASOL et pourront permettre de valider les techniques développées.
- Le projet ALADIN étudie la simulation de problèmes d'écoulement et de dispersion dans les milieux poreux, en collaboration avec l'Institut de Mécanique des Fluides et des Solides de Strasbourg. Cette application servira à tester les méthodes de résolution de systèmes linéaires symétriques indéfinis ou non symétriques.
- Le projet CORIDA s'intéresse à des équations aux dérivées partielles modélisant le processus de solidification d'alliages métalliques. Le modèle est 2D (+1D en temps) et est d'importance industrielle [26, 33, 27]. Pour mener à bien cette application, nous proposons d'intégrer dans SCILAB un mailleur existant qui permette l'adaptation selon une carte de tailles (plusieurs mailleurs sont disponibles gratuitement dans le cadre de travaux de recherche, en particulier les mailleurs développés par le projet Gamma). Nous proposons également de développer et d'intégrer dans Scilab des briques de base permettant de résoudre des problèmes d'équations aux dérivées partielles par la méthode des éléments finis. Ces travaux pourront constituer un premier pas vers une boite à outils plus générale pour les équations aux dérivées partielles.

4 Identité et rôle précis des participants

Le tableau ci-dessous établit la liste des onze équipes partenaires de l'action, ainsi que les membres des équipes qui y participeront.

Équipe	Institut ou organisation	Membres
Projet ALADIN	INRIA-Rennes	E. Canot, J. Erhel, F. Guyomarc'h B. Philippe
ALiENoR et avant-projet ScAlApplix	LaBRI UMR 5800 INRIA-FutUR	P. Hénon F. Pellegrini P. Ramet J. Roman
APO	ENSEEIHT-IRIT UMR 5505	P. Amestoy
Projet CORIDA	INRIA-Lorraine	B. Pinçon J.-F. Scheid
Projet METALAU	INRIA-Rocquencourt	F. Delebecque C. Gomez M. Goursat J.-P. Quadrat S. Steer
Projet REMAP	INRIA-Rhône-Alpes	E. Caron, F. Desprez, J-Y. L'Excellent
Projet RESEDAS	INRIA-Lorraine	E. Jeannot
Heudiasyc (UMR 6599) Roberval (UMR 6066)	Université de Technologie de Compiègne Université de Technologie de Compiègne	J-P. Boufflet C. Denis P. Breitkopf M. Vayssade
	Université de Beyrouth	D. Mezher
	Xerox Palo Alto Research Center	J. Gilbert

De plus, cette ARC s'inscrit naturellement dans la continuité de projets et d'actions de recherche avec nos partenaires français du CERFACS ainsi que nos partenaires internationaux (Y. Saad, I. Duff, J. Koster, etc), qui pourront être consultés.

4.1 Projet ALADIN et université de Beyrouth

Le projet ALADIN est le coordinateur de l'action (responsable J. Erhel).

Le projet ALADIN intégrera le logiciel SCILIN, qu'il complètera par de nouvelles méthodes et de nouveaux préconditionnements. Il intégrera également, avec le projet CORIDA, le logiciel ARPACK pour le calcul de valeurs propres. Il développera, en collaboration avec D. Mezher de l'université de Beyrouth, un logiciel pour la factorisation QR de matrices creuses ainsi qu'un logiciel parallèle pour la méthode GMRES. Enfin, le projet ALADIN développera l'application décrite au paragraphe 3.5.

4.2 Équipes ALiENor et APO, Projet ReMaP

Le projet ReMaP a travaillé sur SCILAB // (SCILAB parallèle) en participant précédemment à l'ARC OURAGAN et s'intéresse particulièrement au Grid Computing. Nous interfacerons des solveurs parallèles creux (MUMPS, PaStiX, SuperLU, ...) dans SCILAB, dans une approche calcul sur la grille.

Le projet ReMaP, les équipes ALiENor et APO poursuivront des travaux sur les solveurs MUMPS et PaStiX. Pour le contexte grille, nous nous intéresserons aux aspects séquencement statique et dynamique sur des serveurs hétérogènes.

D'autre part, des techniques classiques de réordonnancement seront étudiées pour les méthodes directes qui pourront contribuer au paquetage de l'Université de Technologie de Compiègne (voir 4.6).

4.3 Projet CORIDA

Le projet CORIDA intégrera dans SCILAB les solveurs creux UMFPACK [11] pour matrices non symétriques et SNMF de la librairie TAUCS [3] (Sivan Toledo). De manière optionelle, des paramètres supplémentaires (comme par exemple le choix du réordonnancement) pourront être définis pour contrôler le comportement de ces solveurs. Le projet CORIDA intégrera, avec le projet ALADIN, le logiciel ARPACK. L'application décrite au paragraphe 3.5 servira à démarrer le développement d'outils pour la résolution d'équations aux dérivées partielles.

4.4 Projet METALAU

L'expérience du projet METALAU dans le développement de SCILAB aidera à des choix consistants de spécifications (types et interfaces) par rapport à l'existant. D'autre part, le projet METALAU jouera un rôle moteur dans l'étape de validation.

4.5 Projet RESEDAS

Le projet RESEDAS met au point une interface Netsolve/SCILAB pour le calcul creux. A ce titre il travaille sur la conversion de format de matrices creuses (SCILAB, CSR, CSC, ...). L'interfaçage avec netsolve pourra servir de point de départ pour les intégrations mentionnées précédemment puisque NetSolve permet déjà d'accéder aussi bien à des solveurs itératifs (PETSc, Aztec, ITPACK) que directs (MA28 ou SuperLU).

4.6 Équipes Heudiasyc et Roberval, UTC

Il s'avère nécessaire de disposer d'un paquetage de méthodes de réordonnancement pour SCILAB pour compléter son environnement de développement d'applications (voir paragraphe 3.4). Ce qui permettra à l'utilisateur d'expérimenter puis de choisir une méthode en adéquation avec sa problématique.

Nous intégrerons dans SCILAB différentes méthodes de réordonnancement de matrices creuses. Nous disposons actuellement des codes C des méthodes [3] [10] [15] [18] [17] [29] [36].

[3] voir http://www.tau.ac.il/ stoledo/taucs/

Ce paquetage pourra être complémentaire ou bien ensuite étendu avec d'autres algorithmes mentionnés dans le paragraphe 3.4.

D'autre part, pour augmenter les possibilités d'échanges de données (cf. paragraphe 3.1), nous implémenterons des procédures d'import/export de données vers/depuis le logiciel de CAO IDEAS.

4.7 Xerox Palo Alto Research Center

L'expérience de John Gilbert concernant MATLAB et les retours sur les besoins des utilisateurs serviront à la phase de spécification. Son rôle sera de fixer les priorités à faire resortir: efficacité versus facilité d'utilisation. D'autre part, John Gilbert apportera au projet son expérience avec MATLAB et sa vision globale du domaine. Il participera en particulier aux travaux concernant les graphes.

5 Financement demandé et destination envisagée des crédits

Le financement est chiffré est Kilo-Euros.

Pour la première année, le financement global s'élève à 47 KEuros et se répartit comme suit :

- post-doc, 6 mois, avec Aladin : 20 KEuros
- stage de DEA, 4 mois, avec Alienor et APO : 5 KEuros
- stage de DEA, 4 mois, avec UTC : 5 KEuros
- missions : 9 KEuros
- workshop : 8 KEuros

Pour la seconde année, le budget demandé est de 63 KEuros et servira aux financements suivants :

- post-doc, 6 mois, avec Remap : 20 KEuros
- post-doc, 6 mois, avec Corida : 20 KEuros
- stage de DEA, 4 mois, avec Metalau : 5 KEuros
- missions : 9 KEuros
- workshop final : 9 KEuros

Soit un budget total demandé de 110 KEuros.

Références

[1] P. Amestoy, T. Davis, and I. Duff. An approximate minimum degree ordering algorithm. *SIAM Journal of Matrix Analysis and Applications*, 17(4):886–905, 1996.

[2] P. Amestoy, I. Duff, and C. Puglisi. Multifrontal qr factorization in a multiprocessor environment. *Numerical Linear Algebra with Applications*, 3:275–300, 1996.

[3] P. R. Amestoy, T. A. Davis, and I. S. Duff. An approximate minimum degree ordering algorithm. *SIAM Journal on Matrix Analysis and Applications*, 17:886–905, 1996.

[4] P. R. Amestoy, I. S. Duff, and J.-Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Comput. Methods Appl. Mech. Eng.*, 184:501–520, 2000.

[5] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and X. S. Li. Analysis and comparison of two general sparse solvers for distributed memory computers. ACM Transactions on Mathematical Software, 2001. to appear.

[6] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and X. S. Li. Impact of the implementation of mpi point-to-point communications on the performance of two general sparse solvers. Technical Report RT/APO/01/4, ENSEEIHT-IRIT, Oct 2001. Submitted to *Parallel Computing*.

[7] Patrick R. Amestoy, Iain S. Duff, Jacko Koster, and Jean-Yves L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

[8] R. Barret, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the solution of linear systems: building blocks for iterative methods - 2nd edition.* SIAM / netlib, Philadelphia, PA, 1994.

[9] A. Björck. *Numerical Methods for Least Squares Problems.* SIAM, 1996.

[10] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proc. 24th National Conference of the Association for Computing Machinery*, pages 157–172, 1969. New Jersey.

[11] T. A. Davis and I. S. Duff. An unsymmetric-pattern multifrontal method for sparse LU factorization. *SIAM Journal on Matrix Analysis and Applications*, 18:140–158, 1997.

[12] T. A. Davis, J. R. Gilbert, S. I. Larimore, and E. G.-Y. Ng. A column approximate minimum degree ordering algorithm. Technical Report TR-00-005, Computer and Information Sciences Department, University of Florida, Gainesville, FL, 2000.

[13] James W. Demmel, Stanley C. Eisenstat, John R. Gilbert, Xiaoye S. Li, and Joseph W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM Journal on Matrix Analysis and Applications*, 20(3):720–755, 1999.

[14] I. S. Duff and J. Koster. The design and use of algorithms for permuting large entries to the diagonal of sparse matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(4):889–901, 1999.

[15] I. S. Duff, J. K. Reid, and J. A. Scott. The use of profile reduction algorithms with a frontal code. *Int. J. for Num. Meth. in Eng.*, 28:2555–2568, 1989.

[16] J. Erhel. A parallel GMRES version for general sparse matrices. *Electronic Transactions on Numerical Analysis*, 3:160–176, 1995.

[17] J. A. George and J. W. Liu. An automatic nested dissection algorithm for irregular finite element problems. *SIAM Journal of Numerical Analysis*, 15:1053–1069, 1978.

[18] N. E. Gibbs, W.G. Poole Jr, and P. K. Stockmeyer. An algorithm for reducing the bandwidth and the profile of a sparse matrix. *SIAM J. Numer. Anal.*, 13(2):236–250, 1976.

[19] D. Goudin, P. Hénon, F. Pellegrini, P. Ramet, J. Roman, and J.-J. Pesqué. Parallel sparse linear algebra and application to structural mechanics. *Numerical Algorithms, Baltzer Science Publisher*, 24:371–391, 2000.

[20] P. Hénon. *Distribution des Données et Régulation Statique des Calculs et des Communications pour la Résolution de Grands Systèmes Linéaires Creux par Méthode Directe*. PhD thesis, LaBRI, Université Bordeaux I, Talence, France, November 2001.

[21] P. Hénon, P. Ramet, and J. Roman. A Mapping and Scheduling Algorithm for Parallel Sparse Fan-In Numerical Factorization. In *Proceedings of EuroPAR'99, Toulouse, France*, number 1685 in Lecture Notes in Computer Science, pages 1059–1067. Springer Verlag, September 1999.

[22] P. Hénon, P. Ramet, and J. Roman. PaStiX: A Parallel Sparse Direct Solver Based on a Static Scheduling for Mixed 1D/2D Block Distributions. In *Proceedings of Irregular'2000, Cancun, Mexique*, number 1800 in Lecture Notes in Computer Science, pages 519–525. Springer Verlag, May 2000.

[23] P. Hénon, P. Ramet, and J. Roman. PaStiX: A Parallel Direct Solver for Sparse SPD Matrices based on Efficient Static Scheduling and Memory Managment. In *Tenth SIAM Conference on Parallel Processing for Scientific Computing, Portsmouth, Virginie, USA*, March 2001.

[24] P. Hénon, P. Ramet, and J. Roman. PaStiX: A High-Performance Parallel Direct Solver for Sparse Symmetric Definite Systems. *Parallel Computing*, 28(2):301–321, January 2002.

[25] G. Karypis and V. Kumar. METIS – *Unstructured Graph Partitioning and Sparse Matrix Ordering System – Version 2.0*. University of Minnesota, June 1995.

[26] D. Kessler, O. Krueger, J. Rappaz, and J.F. Scheid. A phase-field model for the isothermal solidification of a binary alloy. *Computer Assisted Mechanics and Engineering Science*, (7):279–288, 2000.

[27] O. Krueger, M. Picasso, and J.F. Scheid. A posteriori error estimates and adaptive finite elements for a nonlinear parabolic problem related to solidification. *submitted to Applied Numerical Mathematics.*

[28] X. S. Li and J. W. Demmel. A scalable sparse direct solver using static pivoting. In *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing*, San Antonio, Texas, March 22–24 1999.

[29] S. Negre, J. P. Boufflet, J. Carlier, and P. Breitkopf. Improving the finite elements ordering for the frontal solver. *Revue Européenne des Eléments Finis*, 9(8):917–940, 2000.

[30] F. Pellegrini. SCOTCH 3.4 User's guide. Technical Report RR 1264-01, LaBRI, Université Bordeaux I, November 2001.

[31] F. Pellegrini, J. Roman, and P. R. Amestoy. Hybridizing nested dissection and halo approximate minimum degree for efficient sparse matrix ordering. *Concurrency: Practice and Experience*, 12:69–84, 2000. Preliminary version published in *Proceedings of Irregular'99*, LNCS 1586, 986–995.

[32] F. Pellegrini, J. Roman, and P. R. Amestoy. Hybridizing nested dissection and halo approximate minimum degree for efficient sparse matrix ordering. *Concurrency: Practice and Experience*, 12:69–84, 2000. Preliminary version published in *Proceedings of Irregular'99*, LNCS 1586, 986–995.

[33] J. Rappaz and J.F. Scheid. Existence of solutions to a phase-field model for the isothermal solidification process of binary alloys. *Math. Meth. Appl. Sci.*, 23(6):491–513, 2000.

[34] Y. Saad. *Iterative methods for sparse linear systems.* PWS Publishing Company, 1996.

[35] Y. Schulze. Towards a tighter coupling of bottom-up and top-down sparse matrix ordering methods. *to appear in BIT*, 1999.

[36] S. W. Sloan. An algorithm for profile and wavefront reduction of sparse matrices. *Int. J. for Num. Meth. in Eng.*, 23:239–251, 1986.