lisation par exemple des tâches qui s'exécutent sur les processeurs etc. C'est un peu l'étape intermédiaire entre les bibliothèques classiques et les ateliers de programmation. Un point très important concernant les ateliers de programmation est de fournir un contact avec des experts. Pour les méthodes de Krylov, on montre que pour une méthode choisie et une machine parallèle donnée il y a une taille de sous espace qui conduit aux performances optimales.

Messaoud YOUCEF-OUALI : quelles sont les bibliothèques les plus performantes ?

Serge PETITON : c'est très difficile à dire. En fait tout dépend de la machine. Les complexités des algorithmes utilisés dans les différentes bibliothèques sont à peu prêt identiques. Deux se détachent des autres d'un point de vue global d'après les utilisateurs, ce sont ScaLAPACK et PETSc. Une bibliothèque comme P-SPARSLIB est beaucoup utilisée mais par une classe plus restreinte de personnes pour l'instant.

Les questions étant épuisées, Michel COSNARD remercie Serge PETITON pour sa présentation et propose aux membres de l'assemblée de prendre un pause.

$10^6 \times 10^6$ si les valeurs propres sont bien séparées. Cela dépend surtout du spectre de la matrice. Néanmoins, cela demande des temps de calcul énorme et des méthodes encore du niveau de la recherche.

Le transparent SP-19 est consacré à la conclusion de l'exposé de Serge PETITON. Il existe de nombreuses bibliothèques parallèles accessibles pour les applications bien conditionnées et de tailles normales (in core). Elles sont souvent basées sur ScaLAPACK et MPI avec les PBLAS 1, 2, et 3 et les BLAS optimisées par les constructeurs. (IBM, CRAY, Intel...) ou par des distributeurs de logiciels (NAG...). Un point important est qu'actuellement, chaque bibliothèque définit un modèle de programmation parallèle qui est fixe. Or sur certaines nouvelles machines, les processeurs sont hautement vectoriels et les bibliothèques ne savent pas gérer cette possibilité. Serge PETITON pense que dans l'avenir on trouvera de plus en plus de bibliothèques multi-domaines où le domaine de programmation sera vu comme un objet ou au moins comme un paramètre important. L'avenir est aux ateliers de programmation scientifique. Ils devront fournir plus que des toolkits. Il faudra avoir à la fois une approche orientée objet, des maillages adaptatifs, de l'out of core, permettre plusieurs modèles de programmation, avoir des outils de visualisation, de l'interactivité et surtout des liens avec des experts capables de paramètrer la taille des sous espaces pour certaines méthodes de Krylov par exemple.

Le transparent SP-20 est consacré à des directions de recherche sur lesquels Serge PETITON travaille actuellement dans le cadre du projet KrylovKit. Serge PETITON invite les personnes intéressées par ce projet à le contacter. Ce sera un atelier de programmation uniquement basé sur les méthodes de Krylov. Ces méthodes sont de plus en plus utilisées que ce soit pour les systèmes linéaires ou pour les valeurs propres. Pour les cas ne nécessitant pas ces méthodes, ScaLAPACK devrait de toutes manières fournir des méthodes. Devront être disponibles les méthode d'Arnoldi, d'Arnoldi hybride, d'Arnoldi par blocs (qui n'est pas encore implémentée et qui peut être très importante), FGMRES (utile quand on a plusieurs seconds membres), QMR, BiCGStab, et des gradients conjugués, le tout pour des problèmes de très grandes tailles. Le but est d'intégrer tous les résultats récents de la recherche sur l'algèbre linéaire parallèle. Elle devrait permettre le passage au MIMD et au multi-domaines. Ce projet a été commencé en 1996 par une phase de spécifications. Des codes existent qui tournent sur différentes machines parallèles. Cette recherche est réalisée en collaboration avec un industriel, Adulis, qui possède entre autre un centre d'expertise objet.

Paul FEAUTRIER : quelles sont vraiment les différences entre un toolkit et une bibliothèque?

Serge PETITON : Les exemples de bibliothèques classiques sont LINPACK, ScaLAPACK... C'est à dire un ensemble de routines qui sont des boites noires. Un toolkit fournit en plus des outils de *profiling* capables par exemple de visualiser les portions de codes les plus coûteuses en temps CPU, des outils de visua-

Le transparent SP-16 présenté par Serge PETITON est consacré à la bibliothèque P-SPARSLIB. La représentation de matrices creuses utilisée est optimisée pour les produits matrice-vecteur. Quand on a des matrices creuses en algèbre linéaire on a souvent recourt à des méthodes itératives et donc, souvent, le produit matrice-vecteur est très utilisé. Il est important de noter que tous les algorithmes utilisés reposent sur des méthodes de KRYLOV itératives.

Le transparent SP-17 est consacré à PETSc. C'est une bibliothèque orientée objet. On retrouve les BLAS, LAPACK et MPI. Le transparent illustre la hiérarchie des niveaux d'abstraction de la bibliothèque. Au centre de la bibliothèque se trouve un module dédié aux méthodes de sous espaces de KRYLOV. Ces méthodes sont de plus en plus utilisées car elles sont très bien adaptées aux systèmes linéaires et calcul de valeurs propres non symétriques pour de grosses matrices très nombreuses dans le cadre des applications qui sont exécutées sur des machines parallèles. C'est une bibliothèque très bien réalisée au niveau de la programmation. Elle permet par exemple à un utilisateur d'emcapsuler son propre solveur à l'intérieur de la bibliothèque. En contrepartie, c'est une bibliothèque relativement lourde à installer.

Avec le transparent SP-18, Serge PETITON propose de montrer comment les bibliothèques classiques sont en train de se transformer en *toolkits* pour le calcul scientifique parallèle. La majorité des bibliothèques actuelles sont orientées passage de message. De plus en plus de bibliothèques considèrent le cas des matrices creuses en ayant recourt aux méthodes de Krylov. Malheureusement, très peu proposent le calcul de problèmes de valeurs propres non Hermitiennes avec les dernières méthodes d'accélération. Les pré conditionnements parallèles de qualité font encore défaut. Aucune bibliothèque ne prend en compte le *out of core* qui est l'expression anglaise caractérisant les problèmes qui ne peuvent pas entrer entièrement dans la mémoire de la machine. Or de plus en plus d'industriels et de chercheurs ont besoin de gérer des données qui ne tiennent pas sur les ordinateurs les plus puissants, ou l'accès à ces machines ne leur est pas autorisé. La plupart d'entre eux résolvent le problème en écrivant eux même la gestion du "out of core", comme à l'Aérospatiale par exemple. Il n'existe pas de résolution complète de grands problèmes de valeurs propres non Hermitiennes comme on en trouve en chimie. Or ce type de calcul était proposé dans les bibliothèques scalaires. Les méthodes parallèles actuelles ne permettent pas d'obtenir toutes les valeurs propres mais celles de plus grandes parties réelles.

Paul FEAUTRIER : est-ce vraiment intéressant d'obtenir de telles valeurs ? Quelle est la précision des valeurs obtenues ?

Serge PETITON : des chimistes en ont besoin par exemple, pour résoudre des problèmes de changements de niveaux d'énergie. Et si on utilise les techniques adéquates, on peut espérer une bonne précision. On peut avoir de gros problèmes pour calculer 3, 4 valeurs propres si elles sont de modules proches. C'est par exemple le cas de la méthode QR de EISPACK, même si vous avez une matrice 100×100. Par contre, vous pouvez espérer traiter une matrice creuse

etc. Les communications sont, comme en MPI, de 3 types : non bloquantes, localement bloquantes ou globalement bloquantes. Serge PETITON présente le transparent SP-11 pour illustrer la programmation avec ScaLAPACK. Classiquement, la première routine appelée est celle qui permet au processus d'obtenir des informations telles que le nombre de processeurs et son numéro de processus (BLACS-PINFO). Ensuite est appelée la routine permettant de générer la grille de processeurs virtuelle. Serge PETITON ne souhaite pas entrer trop dans les détails mais plutôt montrer les grandes lignes de la programmation en ScaLAPACK : de nombreux appels de procédure avec une mnémotechnique très forte.

Serge PETITON présente ensuite le transparent SP-12 dédié aux PBLAS. Les PBLAS sont des versions parallèles de BLACS. Comme pour les BLACS, les noms des routines suivent une mnémotechnique très forte. Ils s'écrivent sous la forme Pxyyzz, où x représente la précision souhaitée (S, D, C ou Z), yy représente le type de matrice, zz représente le code. En général, la distribution des matrices est *bloc cyclic* sur la grille virtuelle 2D des processeurs. Pour chaque matrice distribuée, on trouve un descripteur contenant la taille de la matrice, le nombre de blocs distribués, les coordonnées du processeur ayant le premier élément de la matrice, le contexte et la dimension la plus importante lors de la répartition. On se ramène donc de coordonnées sous la forme (m,n) à une adresse de processeur et à une adresse dans la mémoire locale du processeur.

Pour illustrer l'utilisation des PBLAS, Serge PETITON présente le transparent SP-13 qui contient un exemple réalisé par Antoine PETITET qui finit actuellement sa thèse dans l'équipe de Jack DONGARRA et fait partie des personnes qui ont travaillé au développement de ScaLAPACK. On retrouve certaines des initialisations des BLACS comme BLACS-PINFO. La routine DESCINIT permet d'initialiser les descripteurs pour chacune des matrices. Par la suite, lors d'appels à des primitives BLACS, on passe la matrice à travers son descripteur. La programmation résultante n'est pas très lisible contrairement à une programmation data-parallèle type HPF.

Le transparent SP-14 illustre la hiérarchie de la bibliothèque ScaLAPACK. Au niveau global se trouvent ScaLAPACK évidement, et les PBLAS. Au niveau local, on utilise les BLACS pour les communications, les BLAS sur chaque processeur, LAPACK et les routines de passage de messages.

PARPACK (Parallel ARnoldi PACKage) est une sous partie de ScaLAPACK (voir le transparent SP-15) et correspond à la version parallèle de ARPACK vue précédemment. Elle implémente la méthode de SORENSEN en Fortran77 et en utilisant MPI. Cette méthode (et de nombreuses autres, avec l'apparition des méthodes de Krylov) comporte plusieurs phases de calcul dont la plus exigeante en termes de calculs est la phase dite de projection représentée par les équations 1, 2, 3 et 4 du transparent SP-15. Les calculs de Ritz, qui sont une autre phase de la méthode, sont faits de manière redondante sur les processeurs en utilisant LAPACK.

lisation et une approche orientée objet très poussée. Elle fournit des méthodes pour les équations aux dérivées partielles. Elle est basée sur BLAS, LAPACK et MPI. Cette bibliothèque fournit un *profiler* qui permet d'avoir un profil très précis des calculs qui ont été faits. Des *drivers* sont aussi fournis (d'autres bibliothèques en fournissent aussi). Les *drivers* sont des successions d'appels à des routines d'une bibliothèque qui permettent de résoudre un problème précis. TMSSL est la nouvelle bibliothèque de Thinking Machine Corp. Elle n'est pour l'instant pas vraiment opérationnelle mais semble sur la papier très intéressante.

Messaoud YOUCEF-OUALI : TMSSL est-elle une bibliothèque propriétaire ?

Serge PETITON : non, elle est basée sur l'environnement global work qui devrait lui permettre de fonctionner sur différents types de machines. Mais pour l'instant TMSSL reste à l'état embryonnaire. Surtout, la majorité de l'équipe, parmi lesquels les responsables du projet, qui avait réalisé CMSSL ayant quitté Thinking Machine, on peut douter de la capacité de la nouvelle équipe à réaliser un produit aussi performant, d'autant plus que cette équipe semble moins ouverte aux collaborations universitaires.

La compagnie Scientific Computing Associate propose un package nommé PCG-PACK de gradient conjugué préconditionné. Elle est représentative de nombreuses autres sociétés qui développent des packages spécialisés et qui proposent des services de suivi et de maintenance qui ne sont pas disponibles quand on utilise des produits purement universitaires et gratuits comme ScaLAPACK. La bibliothèque P-SPARLIB est la nouvelle bibliothèque en cours de développement à l'université de Minneapolis par l'équipe de Youcef SAAD. C'est, au moins sur la papier, une des bibliothèques les plus complètes actuellement. La bibliothèque NAG est basée sur ScaLAPACK et utilise PVM et MPI. De nombreuses autres bibliothèques sont disponibles de part le monde, telle PUL développée à Edinburgh, PCG++ à Austin ou encore PIM à Rio do sud et l'université du Kent. Des sociétés comme IBM et CRAY proposent leur propres implémentations de ScaLAPACK basées sur des bibliothèques de passage de message propriétaires optimisées pour leurs machines. METIS est une bibliothèque de partitionnement de graphes et de matrices creuses développée à Minneapolis par l'équipe de G. KARYPIS.

Serge PETITON propose de reprendre quelques unes de ces bibliothèques en détails et commence par ScaLAPACK en présentant le transparent SP-10. La programmation de ScaLAPACK est SPMD, comme beaucoup d'autres bibliothèques, et utilise les PBLAS 1, 2 et 3 et les BLACS. Les BLACS permettent des communications point à point, des diffusions, des opérations de combinaisons et de nombreuses autres routines de passage de messages. En général, on opère sur une grille virtuelle 2D. On trouve 2 classes principales de matrice : les matrices rectangulaires et les matrices trapézoïdales. On peut ensuite définir si les directions sont les lignes, les colonnes ou tous les processeurs pour les *scoped operations*. Un contexte joue le rôle du communicateur en MPI. De nombreuses primitives permettent de créer les grilles virtuelles, créer les contextes

L'ensemble de la bibliothèque étant basé sur les BLAS, quand celles-ci sont vectorisées, on arrive à utiliser la puissance vectorielle de la machine à peu de frais. Les BLAS vectorielles ont souvent été optimisées par les constructeurs pour leurs machines. Ceci a permis de réutiliser les codes développés en LINPACK pour des machines scalaires et de les mettre en oeuvre sur des machines vectorielles à moindre coût. Ensuite sont apparues des versions où la totalité de certains algorithmes a été reprise et vectorisée. Une des premières bibliothèques complètement vectorielles et générales, c'est à dire dépassant le cadre des systèmes linéaires, a été MAGEV développée par Control Data. Elle se basait sur les primitives Q8 qui étaient un équivalent des BLAS micro-codées pour les machines de Control Data. On trouvait entre autres des calculs de valeurs propres pour des matrices symétriques et l'ensemble des routines pour systèmes linéaires. Parallèlement, la bibliothèque EIVPACK développée à Control Data France traitait les problèmes de valeurs propres non Hermitiennes et où l'on voit apparaître les premières méthodes de Krylov, telle que la méthode d'Arnoldi pour résoudre des problèmes de valeurs propres et aussi une des premières versions de la méthode QR pour les valeurs propres en vectoriel. Pour le cas creux en général, on avait souvent recourt aux Gather/Scatter qui permettaient de gérer les indirections liées aux systèmes creux. Certaines machines, telles les CRAY X-MP, possédaient des accélérateurs matériels spécialement développés à cet effet.

Avec le transparent SP-8, Serge PETITON propose de parler des bibliothèques parallèles. La première, mais qui n'est pas très répandue, est PLAPACK développée par R. VAN DE GEIJN à l'université du Texas à Austin. Elle introduisit l'approche orientée objet dans les bibliothèques et utilise les premières versions des BLAS parallèles. La plus populaire des bibliothèques parallèles est ScaLAPACK (SCAlable LAPACK). Elle est issue d'un projet de la DARPA de réalisation d'une bibliothèque *scalable* pour l'algèbre linéaire dense ou creuse pour les machines massivement parallèles. Souvent, le terme ScaLAPACK ne désigne qu'une partie du projet de la DARPA : la version parallèle de LAPACK basée sur les BLAS parallèles. En fait, le projet total contient aussi PARPAK, PARPRE et CAPSS. Les BLAS parallèles sont elles-mêmes basées sur des bibliothèques parallèles générales standard telle PVM ou MPI. De nombreux constructeurs ont ensuite optimisé les implémentations de ScaLAPACK en utilisant leur propres bibliothèques de communication de messages (CMMD pour Thinking Machine, NX pour Intel). La bibliothèque CMSSL est une des rares bibliothèques data parallèle. Elle est due à Thinking Machine. On y trouve une méthode d'Arnoldi implicite ainsi que des calculs sur matrices creuses utilisant le format COO data parallèle. Il est à noté que ce format, s'il permet d'exprimer très simplement la matrice en programmation data parallèle, ne fournit pas les meilleures performances pures. Une autre des particularités de cette bibliothèque était, sous certaines conditions, de pouvoir compiler les communications pour les optimiser. La bibliothèque PETSc (voir le transparent SP-9), dont la version 2.0 est toute récente, est développée à Argone National Laboratory par l'équipe de Bill GROPP. C'est plus qu'une bibliothèque dans le sens usuel du terme: en plus des sous-programmes de calcul numériques on trouve des outils de visua-

bliothèque a été développée à Oak Ridge National Laboratory par DONGARRA entre autres. ARPACK est apparue plus récemment. Elle est basée sur une version implicite d'Arnoldi introduite par Danny Sorensen de l'université de Rice. Elle utilise aussi les BLAS 1, 2 et 3 et LAPACK. ARPACK et LAPACK sont diffusées actuellement en version parallèle. La dernière bibliothèque scalaire présentée par Serge PETITON est ITPACK qui était basée sur les méthodes itératives et développée dans un milieu plus industriel. Serge PETITON précise que de nombreuses autres bibliothèques existent, par exemple celles développées par CRAY pour des domaines plus spécifiques ou utilisant des méthodes très classiques.

Les bibliothèques prenant en compte les matrices creuses sont présentées sur le transparent SP-5. Ces matrices sont très répandues avec les maillages, les méthodes à éléments finis... Une des premières bibliothèques a été le Yale Sparse Package diffusée dès 70. C'était une bibliothèque complète d'algèbre linéaire pour les matrices creuses. Elle imposa aussi la représentation des matrices creuses sous le format CSR (Compress Sparse Row). Sous le format CSR, tous les éléments non nuls de la matrice sont stockés dans un tableau à une dimension de taille nnz (Number of Non Zero). Un autre tableau contient les indices des colonnes des éléments non nuls pour un ordre de ligne donné. Un dernier tableau de taille n contient les indices des indices de colonne des premiers éléments de ligne en suivant l'ordre choisi. C'est devenu un format très classique. La bibliothèque ELLPACK est dédiée aux méthodes à éléments finis et utilise un format particulier de stockage des matrices creuses dans un tableau à 2 dimensions dont une est compressée. Puis est apparue la bibliothèque SPARSKIT, bibliothèque scalaire pour calcul sur matrice creuse développée à la NASA et à l'université de Minneapolis par Y. SAAD. Elle utilise aussi le format CSR et surtout on trouve un ensemble d'outils de manipulations de matrices qui offrent à l'utilisateur le choix entre transformer ses matrices au format CSR avant les appels aux routines de la bibliothèque (de nombreuses routines de conversion sont fournies) ou travailler directement en format CSR en utilisant les outils de manipulation et de visualisation de matrices fournis. La bibliothèque fournit en particulier des conversions depuis COO vers CSR. Le format COO est simple : tous les éléments non nuls sont stockés avec leurs coordonnées i et j. Il est à remarquer que toutes ces bibliothèques (denses ou creuses) ont à chaque fois été introduites et développées par des numériciens très réputés dans leur domaines de recherche et ont été diffusées dans le monde entier. Pour en finir avec les bibliothèques scalaires, Serge PETITON présente le transparent SP-6 qui lui a été fourni par Scientific Computing Associates pour illustrer comment un industriel présente sa bibliothèque. Cette bibliothèque, SMPACK, propose en particulier une élimination de Gauss pour systèmes creux pour des stations de travail ou des mainframes avec des versions par bloc, qui peuvent être optimisées pour des machines parallèles à mémoire partagée.

Avec le transparent SP-7, Serge PETITON propose de passer aux bibliothèques vectorielles et d'en présenter une liste évidement partielle. La première bibliothèque à avoir été vectorisée fut LINPACK où les BLAS étaient vectorielles.

recourt à des bibliothèques spécifiques qui sont peu répandues et très ciblées pour chacune des méthodes. Leur énumération aurait été fastidieuse. La plupart du temps on utilise des méthodes implicites, surtout pour les problèmes très compliqués qui justifient l'utilisation de machines parallèles, et on se ramène à des problèmes d'algèbre linéaire. Les méthodes utilisées requièrent plusieurs pas de temps et à chaque pas de temps, il faut résoudre un problème d'algèbre linéaire. Les bibliothèques d'algèbre linéaire sont donc les plus importantes dans le domaine des bibliothèques de calcul et si les bibliothèques numériques plus spécialisées ont le mérite d'exister, elles sont bien moins répandues et portables que les bibliothèques d'algèbre linéaire généralistes.

Serge PETITON présente dans un premier temps les bibliothèques scalaires qui sont souvent à l'origine des bibliothèques parallèles actuelles (voir les transparents SP-4 et SP-5). La première a été la bibliothèque LINPACK, dès 1974. C'est la première apparition des BLAS (Basic Linear Algebra Sub-routine) autorisée par l'apparition de la programmation structurée et des langages procéduraux(souvent Fortran, d'où le terme sub-routine qui deviendra par la suite sub-programm). On trouve la fameuse routine SAXPY qui permet, en simple précision (S), le produit d'une matrice (A) par un vecteur (X) plus (P) un autre vecteur (Y). Rapidement, les constructeurs ont proposé des implémentations spécialisées, voir câblées, des BLAS sur leurs machines pour offrir plus de performances aux utilisateurs. C'est aussi la première apparition de MACHEPS (Epsilon Machine) utilisé pour les tests de convergence numérique. Les méthodes implémentés sont principalement les méthodes directes pour des matrices denses ou des matrices bandes. Cette bibliothèque a été développée par DONGARRA, BUNCH, MOLER et STEWART avec l'aide du numéricien WILKINSON. Vers la même époque, EISPACK est apparue, c'est un peu le pendant *valeur propre* de LINPACK. Elle permet le calcul de vecteurs et de valeurs propres pour des matrices denses essentiellement en utilisant la méthode QR. Il n'y avait pas de méthode itérative de type Krylov qui sont à l'heure actuelle omniprésentes dans les bibliothèques parallèles aussi bien pour la résolution de systèmes linéaires que pour le calcul de vecteurs et valeurs propres. Cette bibliothèque a été développée entre autres par GARBOW, BOYLE, DONGARRA et MOLER. Ces deux bibliothèques ont connu un succès remarquable vers la fin des années 70. Elles ont été diffusées dans le monde entier et utilisées de manière intensive tout en étant gratuites. Puis est apparue LAPACK, implémentée en Fortran. On trouve aussi maintenant une version C (CLAPACK) et une version C++ (LAPACK++). Ces versions sont moins complètes que la version Fortan. D'un point de vue fonctionnel, LAPACK est la réunion de LINPACK et de EISPACK à laquelle ont été rajoutées des méthodes numériques nouvelles des années 70. On trouve aussi des versions par bloc d'algorithmes classiques, c'est à dire des versions orientées vers les machines parallèles à mémoire partagée. Les BLAS sont aussi présentes dans les versions 1, 2 et 3. Elles ont été repensées et spécifiées de manière très rigoureuse pour définir quelles étaient les opérations à fournir, quels devaient être leurs paramètres, leur noms etc. Les BLAS 1 concernent les opérations entre vecteurs. Les BLAS 2 font intervenir des matrices et des vecteurs. Les BLAS 3 concernent les opérations entre matrices. Cette bi-

©OFTA, transcrit par Jean-Christophe Mignot

Observatoire Français des Techniques Avancées

Groupe *Ordinateurs Massivement Parallèles*

COMPTE RENDU (partiel) DE LA TREIZIÈME RÉUNION

Jeudi 24 octobre 1996

Exposé de Serge PETITON (Université de Lille 1, LIFL): "Bibliothèques pour le calcul numérique parallèle ; synthèses et perspectives". Discussion

Serge PETITON commence son exposé en présentant le transparent SP-1. Il est actuellement professeur à l'université de Lille I et travaille parallèlement au projet ASCI (Laboratoire d'Applications Scientifiques pour le Calcul Intensif) dirigé par M. Pierre LALLEMAND. Parler des bibliothèques pour le calcul numérique parallèle est un exercice de style difficile, il faut éviter de faire un catalogue des librairies disponibles qui lasserait, il ne faut pas entrer trop dans le détails des algorithmes numériques utilisés. Serge PETITON listera donc les bibliothèques les plus connues en indiquant les points importants et invite les membres de l'assemblée à ne pas hésiter à l'interrompre pour de plus amples informations lors de son exposé. Les transparents de l'exposé de Serge PETITON sont disponibles sur le site WEB http://www.lifl.fr/~petiton.

Le transparent SP-2 présente le plan de l'exposé. Serge PETITON fera le point sur l'existant, les avantages et les inconvénients. Il parlera des progrès possibles et des directions de recherche et développements. Après l'introduction, Serge PETITON fera un historique des bibliothèques pour le calcul numérique. Il présentera ensuite l'orientation des bibliothèques les plus récentes et comment elles pourraient évoluer. La partie suivante concernera quelques problèmes qui ne sont pas traités par les bibliothèques actuelles et qui devront l'être. Il terminera par une conclusion et une présentation des perspectives et prévisions pour les bibliothèques pour le calcul numérique parallèle.

Serge PETITON parlera essentiellement des bibliothèques d'algèbre linéaire (voir le transparent SP-3). Après la phase de modélisation, on se ramène à des méthodes explicites ou implicites. Si des méthodes explicites sont utilisées, on a