



Prof. Michel Daydé
Toulouse le 28 Février 2018

Directeur de l'IRIT

Objet : Rapport sur la thèse de Xinzhe Wu

La thèse de Xinzhe Wu porte sur l'implantation des méthodes de Krylov ainsi que sur les environnements de programmation multi-niveaux dans la perspective des futures machines à l'Exaflops avec un nombre d'unités de calcul et une hétérogénéité croissants (utilisation de GPUs ou autres accélérateurs). Dans ces architectures, les synchronisations, les communications globales voire la garantie d'une certaine tolérance aux fautes deviennent un vrai challenge à la fois du point de vue algorithmique et du point de vue de la gestion et de la programmation de ces calculateurs.

Les méthodes de Krylov sont communément utilisées pour la résolution des systèmes linéaires creux de très grande taille ou des problèmes aux valeurs propres. La première partie de la thèse se concentre sur l'implantation d'un générateur parallèle de matrices de tests pour les problèmes de valeurs propres, qui passe à l'échelle, afin de tester les méthodes itératives sur les calculateurs parallèles. La seconde partie porte sur l'étude de l'implantation et de la performance des méthodes de Krylov en termes de convergence, préconditionnement, ..., pour arriver à proposer une canevas global de conception de ces méthodes dans la perspective de l'exascale permettant de gérer le volume des communications, la tolérance aux fautes et la réutilisabilité du code. Enfin la possibilité d'utiliser un environnement d'exécution de graphe de tâches (YML) pour implanter ces méthodes est abordée.

L'introduction permet de décrire les motivations et les divers apports de la thèse.

Le chapitre 2 fait un historique des calculateurs parallèles depuis les années 90 et un excellent état de l'art de évolutions que l'on constate actuellement sur la route vers l'exascale avec l'accroissement du nombre de cœurs et l'utilisation d'accélérateurs accroissant l'hétérogénéité des architectures et donc la difficulté à les programmer efficacement. Le survol des modèles de programmation parallèle effectué dans le même chapitre est très didactique et synthétique tout en couvrant tous les points importants. Ensuite, l'auteur décrit un certain nombre des difficultés induites par l'exascale : hétérogénéité croissante, architecture potentielle de ces machines ainsi que leur programmation.

Le chapitre 3 porte sur les méthodes de Krylov. Il commence par un survol des méthodes itératives pour les matrices non-hermitiennes de grande taille où l'exploitation de la structure creuse est importante. L'auteur s'attache plus particulière

à GMRES avec toutes ses améliorations / optimisations : restart, troncature, déflation, ... La convergence de GMRES et son lien avec le spectre des matrices est étudiée ainsi que le préconditionnement de ces méthodes (Jacobi, SOR, SSOR, ILU, multigrille, préconditionneurs polynomiaux, ...). L'auteur détaille ensuite la méthode de Arnoldi et ses variantes (ERAM, IRAM, Krylov-Schur) pour les problèmes aux valeurs propres. La parallélisation des méthodes de Krylov sur les supercalculateurs actuels, l'identification des principales difficultés ainsi que les bibliothèques existantes sont alors abordées. Le chapitre se conclut par la liste de quelques-unes des difficultés qui apparaissent lors du portage vers des machines exaflopiques.

Le chapitre 4 se consacre au développement du générateur de matrices creuses avec un spectre donné. C'est un outil indispensable pour les spécialistes du domaine. Ce chapitre conclut le travail déjà abordé dans au moins deux thèses précédentes avec une approche originale et une implantation parallèle pour CPU et GPU permettant d'engendrer les matrices directement dans la mémoire locale des processeurs. Les expérimentations démontrent la pertinence et l'efficacité de l'approche ainsi que la précision du spectre obtenu. Le tout se conclut par un logiciel public doté d'interfaces vers C, PETSc, ... Un vrai travail d'utilité publique.

Le chapitre 5 décrit UCGLE, une méthode parallèle et asynchrone pour la résolution des systèmes creux non-Hermitien sur les plateformes de calcul de grande taille. UCGLE vise à remplir plusieurs objectifs : communications optimisées, asynchronisme, exploitation de plusieurs niveaux de parallélisme et tolérance aux fautes ainsi que réutilisation. UCGLE est basé sur une approche « Unit and Conquer » qui fait collaborer plusieurs méthodes itératives pour accélérer la convergence. L'implantation de la méthode est effectuée à l'aide de trois composants : GMRES avec restart, un préconditionneur polynomial aux moindres carrés et ERAM pour approcher les valeurs propres dominantes. Le workflow de chaque composant est étudié ainsi que les paramètres déterminants de chaque composant. Le « manager » de UCGLE est ensuite décrit ainsi que la réalisation des communications. Les expérimentations riches et très complètes permettent d'illustrer l'influence des divers paramètres que ce soit dans GMRES ou dans le préconditionneur. La comparaison avec diverses variantes classiques de GMRES démontre l'efficacité de la méthode et ses bonnes performances sur CPU et GPU, sa tolérance aux fautes, l'impact du spectre des valeurs propres sur la convergence (et identifie bien les cas qui posent problème) et enfin ses bonnes propriétés de passage à l'échelle qui en font un candidat intéressant pour les futures plateformes à l'exascale.

Le chapitre 6 étudie l'extension de UCGLE à la résolution d'une suite de systèmes linéaires avec le même opérateur et des second membres différents. On trouve ce cas de figure dans des applications importantes. Les valeurs propres obtenues lors de la résolution des précédents systèmes linéaires sont réutilisées, améliorées à la volée et exploitées pour construire le point de départ de la résolution des systèmes suivants. L'adaptation de UCGLE à ce contexte est soigneusement décrite en particulier les adaptations numériques. Dans les expérimentations réalisées, la comparaison avec plusieurs variantes de GMRES montre qu'elle améliore la convergence et réduit significativement les temps de calcul.

Une démarche identique est utilisée dans le chapitre 7 pour étendre UCGLE à la résolution de systèmes linéaires avec plusieurs seconds membres. La division du système linéaire avec plusieurs second membres en plusieurs sous-ensembles permet une meilleure localisation des calculs, une réduction des communications globales et une amélioration de la performance en lançant plusieurs « block GMRES » à la fois. Cette extension de UCGLE, appelée m-UCGLE, utilise donc « block GMRES » et une

version par bloc de la méthode polynomiale aux moindres carrés. Le nouveau manager, capable d'allouer de multiples composants de calcul à la fois, de gérer les communications asynchrones ainsi que la reprise et la tolérance aux fautes est ensuite décrit. L'évaluation des performances et la comparaison avec d'autres approches est très en faveur de m-UCGLE que ce soit pour les temps de calcul, la convergence, le passage à l'échelle et la tolérance aux fautes.


Logiquement le chapitre 8 prolonge l'effort d'implantation des méthodes de Krylov aux futures architectures en s'intéressant à l'auto-tuning. Le nombre important de paramètres qui conditionnent le comportement et l'efficacité de UCGLE, ainsi que montré dans les chapitres précédents, incite à développer des heuristiques permettant de les gérer de façon transparente pour l'utilisateur. Après un survol des approches utilisées classiquement pour l'auto-tuning, l'auteur, parmi les multiples paramètres de UCGLE choisit de se concentrer sur le réglage de la taille des sous-espaces de Krylov et le nombre d'application du préconditionneur polynomial. La prise en compte d'un nombre plus élevé de paramètres aurait été souhaitable mais au vu de la complexité de la tâche liée à la gestion des deux paramètres retenus, on comprend toute la difficulté de la mise au point d'heuristiques. La stratégie est décrite avec soin et les expérimentations démontrent sa pertinence et son efficacité.

Le chapitre 9 porte l'adaptation requise par YML pour supporter l'implantation d'une approche « Unite and Conquer » comme UCGLE c-à-d la capacité avoir un gestionnaire de tâches capable de gérer la tolérance aux fautes, l'équilibrage de la charge, les communications asynchrones et l'hétérogénéité des calculateurs mêlant CPU, GPU et autres accélérateurs. Après une description de YML et de sa capacité à s'interfacer avec XMP pour supporter un parallélisme multi-niveaux, les limitations de YML pour implanter UCGLE sont identifiées : essentiellement l'asynchronisme des communications et la convergence des méthodes itératives dans un contexte de parallélisme multi-niveaux. Des solutions sont suggérées : aller vers des graphes dynamiques au sein de YML et développer la sortie des branches parallèles, ce qui nécessite une optimisation du scheduler de YML. Malheureusement, dans le temps limité imparti à la thèse, il n'a visiblement pas été possible de réaliser les développements et les tests subséquents.

La conclusion résume à la fois l'ensemble du travail effectué et donne quelques pistes de recherches futures : développement d'une version optimisée en CUDA du générateur de matrices pour GPU (celle présentée dans la thèse repose sur PETSc), développement d'une version de UCGLE pour les matrices où certaines des valeurs propres ont une partie réelle positive alors que les autres ont une partie réelle négative, évolution de YML pour implanter des méthodes « Unite and Conquer » telles que UCGLE.

La thèse a donné lieu à un nombre très significatifs de publications dans des conférences internationales (avec un journal en soumission) et des workshops.

Au vu de la qualité du manuscrit, en considérant l'ensemble des travaux réalisés et leur importance ainsi et le volume important de travail requis, nous recommandons que l'on autorise Xinzhe Wu à soutenir sa thèse.

A handwritten signature in black ink, consisting of a stylized 'M' followed by a horizontal line extending to the right.

Michel DAYDE,
Directeur de l'IRIT