Institut de Recherche
en Informatique de Toulouse

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 librairies existantes sont alors abordées. Le chapitre se conclut par la liste de quelque 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 a 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 en 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.

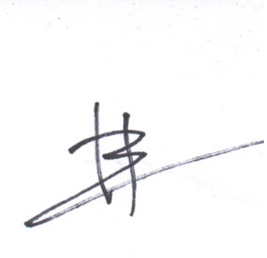
Logiquement le chapitre 8 prolonge l'effort d'implantation des méthodes de Krylov aux futures architectures en s'intéressant à l'auto-tuning. Le nombre important de paramètres qui conditionnent le comportement et l'efficacité de UCGLE, ainsi que montré dans les chapitres précédents, incite à développer 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.



Michel DAYDE,

Directeur de l'IRIT

(This is an English translation of thesis report in French issued by Prof. Michel Daydé)

Prof. Michel Daydé

February 28, 2018, Toulouse

Director of IRIT

Subject: Report on Xinzhe Wu's thesis

Xinzhe Wu's thesis focuses on the implementation of Krylov methods as well as multi-level programming environments in the perspective of future Exaflops machines with a growing number of computing units and heterogeneity (use of GPUs or other accelerators). In these architectures, synchronizations, global communications or the guarantee of a certain fault tolerance become a real challenge both from an algorithmic point of view and from the point of view of the management and programming of these computers.

Krylov methods are commonly used for solving very large sparse linear systems or eigenvalue problems. The first part of the thesis focuses on the implementation of a parallel generator of test matrices for eigenvalue problems, which is scaled up, in order to test the iterative methods on parallel computers. The second part deals with the study of the implementation and performance of Krylov methods in terms of convergence, preconditioning, ..., in order to propose a global framework for the design of these methods in the perspective of the exascale to manage the volume of communications, fault tolerance and code reusability. Finally, the possibility of using a task graph execution (YML) environment to implement these methods is discussed.

The introduction allows to describe the motivations and various contributions of the thesis.

Chapter 2 makes a history of parallel calculators since the 90s and an excellent state of the art evolutions that are currently observed on the road to the exascale with the increase in the number of cores and the use of accelerators increasing the heterogeneity of architectures and therefore the difficulty to program them effectively. The overview of the parallel programming models carried out in the same chapter is very didactic and synthetic while covering all the important points. Then, the author describes a number of the difficulties induced by the exascale: increasing heterogeneity, potential architecture of these machines as well as their programming.

Chapter 3 deals with Krylov methods. It starts with an overview of the iterative methods for large non-Hermitian matrices where the exploitation of the sparse structure is important. The author focuses more specifically on GMRES with all its improvements / optimizations: restart, truncation, deflation, ... The convergence of GMRES and its link with the spectrum of the matrices is studied as well as the preconditioning of these methods (Jacobi, SOR, SSOR, ILU, multigrid, polynomial preconditioners, ...). The author then details the method of Arnoldi and its variants (ERAM, IRAM, Krylov-Schur) for eigenvalue problems. The parallelization of Krylov methods on current supercomputers, the identification of the main difficulties as well as existing libraries are then discussed. The chapter concludes with a list of some of the difficulties that arise when porting to exaflops machines.

Chapter 4 is devoted to the development of the sparse matrix generator with a given spectrum. It is an indispensable tool for specialists in the field. This chapter concludes the work already addressed in at least two previous theses with an original approach and a parallel implementation for CPU and GPU allowing to generate the matrices directly in the local memory of the processors. The experiments demonstrate the relevance and effectiveness of the approach as well as the accuracy of the spectrum obtained. It all ends with a public software with interfaces to C, PETSc, ... A real work of public utility.

Chapter 5 describes UCGLE, a parallel and asynchronous method for solving non-Hermitian hollow systems on large computing platforms. UCGLE aims to fulfill several objectives: optimized communications, asynchronism, exploitation of several levels of parallelism and fault tolerance as well as reuse. UCGLE is based on a "Unit and Conquer" approach that collaborates on several iterative methods to accelerate convergence. The implementation of the method is carried out using three components: GMRES with restart, a least squares polynomial preconditioner and ERAM to approach the dominant eigenvalues. The workflow of each component is studied as well as the determining parameters of each component. The "manager" of UCGLE is then described as well as the realization of the communications. The rich and comprehensive experiments can illustrate the influence of various parameters in GMRES or the preconditioner. The comparison with various classical variants of GMRES demonstrates the efficiency of the method and its good performances on CPU and GPU, its tolerance to faults, the impact of the eigenvalue spectrum on the convergence (and identifies well the problematic cases) and finally its good scaling properties that make it an interesting candidate for future exascale platforms.

Chapter 6 examines the extension of UCGLE to solving a sequence of linear systems with the same operator and different right-hand sides. This case can be found in important applications. The eigenvalues obtained during the solving the previous linear systems are reused, improved on the fly and exploited to build the starting point of solving the following systems. The adaptation of UCGLE to this context is carefully described in particular the numerical adaptations. In the experiments carried out, the comparison with several variants of GMRES shows that it improves the convergence and significantly reduces the computation time.

An identical approach is used in Chapter 7 to extend UCGLE to solve linear systems with several right-hand sides. The division of the linear system with several second members in several subsets allows a better localization of the calculations, a reduction of the global communications and an improvement of the performance by launching several "block GMRES" at the same time. This extension of UCGLE, called m-UCGLE, therefore uses "block GMRES" and a block version of the least squares polynomial method. The new manager, capable of allocating multiple compute components at a time, managing asynchronous communications as well as recovery and fault tolerance is then described. Performance evaluation and comparison with other approaches is very much in favor of m-UCGLE for computing times, convergence, scalability and fault tolerance.

Logically, chapter 8 extends the effort to implement Krylov methods to future architectures by focusing on auto-tuning. The large number of parameters that condition the behavior and efficiency of UCGLE, as shown in previous chapters, encourages the development of heuristics to manage them in a transparent way for the user. After an overview of the approaches used conventionally for auto-tuning, the author, among the multiple parameters of UCGLE chooses to focus on adjusting the size of the Krylov subspaces and the number of applications of the polynomial preconditioner. Taking into account a greater number of parameters would have been desirable but in view of the complexity of the task related to the management of the two parameters retained, one understands the difficulty of the development of heuristics. The strategy is carefully described and the experiments demonstrate its relevance and effectiveness.

Chapter 9 provides the adaptation required by YML to support the implementation of a "Unite and Conquer" approach as UCGLE i.e. the ability to have a task manager capable of handling fault tolerance, balancing of the load, the asynchronous communications and the heterogeneity of the calculators mixing CPU, GPU and other accelerators. After a description of YML and its ability to interface with XMP to support multi-level parallelism, the limitations of YML to implement UCGLE are identified: essentially the asynchronism of communications and the convergence of iterative methods in a context of parallelism multilevel. Solutions are suggested: go to dynamic graphs within YML and develop the output of parallel branches, which requires an optimization of the YML scheduler. Unfortunately, in the limited time allotted to the thesis, it was obviously not possible to carry out the subsequent developments and tests.

The conclusion summarizes all the work done and gives some ideas for future research: development of an optimized version in CUDA of the matrix generator for GPU (the one presented in the thesis is based on PETSc), development of a version of UCGLE for matrices where some of the eigenvalues have a positive real part while the others have a negative real part, evolution of YML to implement "Unite and Conquer" methods such as UCGLE.

The thesis has given rise to a very significant number of publications in international conferences (with a journal in submission) and workshops.

In view of the quality of the manuscript, considering all the work done and its importance as well as the large volume of work required, we recommend that Xinzhe Wu be allowed to support his thesis.

Michel Daydé

(Signature)

Director of IRIT