INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Calcul Scientifique Haute performance

ECTS
3 crédits

Volume horaire

Présentation

Description

* méthode d'arnoldi pour construire les espaces de Krylov, solveur gmres,
* solveur de lanzcos, lien avec le gradient conjugué
* déflation et extraction d'informations spectrales (deflated CG, vecteurs de Ritz), séquence de systèmes linéaires, techniques pour le calcul des valeurs/vecteurs propres
* méthodes de décomposition de domaine (de Schwarz, de P.-L. Lions), étude théorique 1D et 2D, application au calcul parallèle
* méthodes multigrille géométrique, propriété de lissage, schéma V et FMG
* équivalence entre matrice creuse et graphe. Concept de remplissage et dépendances entre inconnus d'un problème creux (arbre d'élimination). Permutations de matrices permettant de réduire le remplissage. Complexité opératoire. Exploitation du parallélisme dans la factorisation de matrices creuses,
* architecture des supercalculateurs modernes (processeurs multi-coeurs, multiprocesseurs ou noeuds SMP et NUMA, accélérateurs GPU, réseaux d'interconnexion). Classification de Flynn. Hiérarchie des mémoires caches. Principes de localité spatiale et temporelle. Programmation parallèle en OpenMP: régions parallèles, directives de synchronisation, boucles parallèles, situations de compétition et tâches. Programmation parallèle en MPI: communications point à point et collectives, opérations de réduction et éventuellement optimisations. Analyse des performances: loi d'Amdahl, scalabilité forte et faible, modèle roof-line et calcul du chemin critique.

Objectifs

A la fin de ce module, l'étudiant devra avoir compris et pourra expliquer (principaux concepts) :

* le principe des méthodes de Krylov, leur utilisation pour résoudre des systèmes linéaires ou calculer des valeurs et vecteurs propres,
* la notion de préconditionnement, la construction et l'utilisation des préconditionneurs,
* théorie et concepts de base des méthodes directes pour systèmes linéaires creux. Complexité opératoire et parallélisme des méthodes directes,
* les notions de base de l'architecture des calculateurs parallèles, les modelés de programmation pour systèmes à mémoire partagée (OpenMP) et distribuée (MPI) et les concepts et méthodes de base pour analyser les performances d'un algorithme ou code parallèle (loi d'Amdahl, hiérarchie de la mémoire cache, principes de localité spatiale et temporelle, modèle roofline, calcul du chemin critique et scalabilité forte et faible).

L'étudiant devra être capable de :

*d'évaluer les coût (flops/mémoire) des différentes méthodes présentées,
* d'analyser l'influence des préconditionneurs,
* d'utiliser des langages haut-niveau pour la discrétisation d'équations aux dérivées partielles,
* de coder les solveurs, de paralléliser des codes simples suivant le standard le plus adéquat ainsi que de les exécuter sur les ressources appropriées,
* d'analyser l'efficacité d'une méthode vis à vis de la complexité opératoire, du temps de calcul et de

l'empreinte mémoire utilisée dans une perspective de calcul haute performance.

Pré-requis nécéssaires

* cours d'Algèbre Linéaire ou de Calcul Scientifique, notamment les méthodes de factorisation LU ou Cholesky
* Notions de base de l'architecture des calculateurs et des langages de programmation impératifs

Évaluation

L'évaluation des acquis d'apprentissage est réalisée en continu tout le long du semestre. En fonction des enseignements, elle peut prendre différentes formes : examen écrit, oral, compte-rendu, rapport écrit, évaluation par les pairs...

Infos pratiques

Lieu(x)

Toulouse