TD-TP3 Méthodes Numériques (MN)
Polytech Grenoble, RICM3

Mars 2016

1 Résolution de systèmes linéaires particuliers

L'objectif est d'étudier et d'implémentaires des méthodes de résolution de systèmes ($Ax = b$) particuliers. b est un vecteur et x le vecteur solution. Nous nous intéresserons à des matrices A qui seront diagonales ou triangulaires (inférieur ou supérieur). Les matrices sont carrées (NxN). Les éléments de vos matrices et vecteurs sont des flottants (4 octets). Votre valeur de N sera choisie pour que ces vecteurs et matrices tiennent complétement dans le cache L3 de votre processeur. Vos codes seront compilés avec l'option d'optimisation **-O2**.

Vous évaluerez les performances de vos fontcions en utilisant l'outil *perf* disponible sous Linux. Si vous ne disposez pas de *perf* sur votre machine et que l'outil ne s'installe pas correctement, un accès à un serveur de calcul disposant de cet outil vous sera fourni.

Chaque fonction de cette fiche sera testée séparement dans un programme. Pour chaque fonction et programme, vous fournirez le temps d'exécution donné par *perf*, le nombre d'instuctions exécutées, le nombre de cycles processeurs, le taux de défaut de cache et le taux d'échec sur des branchements.

Voici donc les différents points à développer.

- Résolution du système avec une matrice A diagonale (tous les éléments non nuls sont sur la diagonale de la matrice A. (fonction *diag* du cours). Vous implémenterez en C et évaluerez avec *perf* la fonction *diag*.
- Résolution du système avec une matrice triangulaire inférieur (fonction tri_inf du cours). Tous les éléments non nuls sont sous la diagonale de la matrice L. Vous implémenterez en C et évaluerez avec *perf* la fonction tri_inf.
- Résolution du système avec une matrice triangulaire supérieur (fonction tri_sup du cours). Tous les éléments non nuls sont au dessus la diagonale de la matrice U Vous implémenterez en C et évaluerez avec *perf* la fonction tri_sup.
- Les matrices triangulaires n'ont que N^2 éléments nons nuls. Quelle structure de données proposez-vous pour représenter de manière plus optimale les matrices triangulaires. Donnez les nouvelles fonctions tri_inf et tri_sup utilisant ces structures de données.

2 Vectorisation des calculs

Les processeurs disposent d'unités de calcul vectoriel et de registres vectoriels. Un cours a été consacré aux principes de la vectorisation des calculs (cours sur la vectorisation). Ces opérations vectorielles sont accessibles au travers d'*intrinsics*. La liste complètes des *intrinsics* est disponible sur le site de la société Intel à l'adresse suivante : `https://software.intel.com/sites/landingpage/IntrinsicsGuide/#`. Consultez le fichier */proc/cpuinfo* pour connaître les versions de vectorisation disponibles sur votre processeur (flags).

— Vectorisez le produit scalaire de deux vecteurs développé dans votre premier TP. Vous vous appuierez sur le code fourni à la page 39 des slides du cours.
— Vectorisez la fonction tri_inf développée dans la section précédente. Evaluez avec $perf$ cette fonction tri_inf.
— Vectorisez la fonction tri_sup développée dans la section précédente.
— Le produit de deux matrices triangulaires supérieures (ou inférieures) donne comme résultat une matrice triangulaire supérieure (ou inférieure). Donnez la version vectorisée de la multiplication de matrice triangulaire. Ajoutez-y de l'OpenMP. Evaluez avec l'outil $perf$.