

# Calcul scientifique parallèle sur GPU

26 novembre 2008

## 1 Contexte

### 1.1 Évolution des architectures de calcul scientifique

Avec la stagnation de la puissance par processeur, des gains de puissance ne peuvent être réalisés que par la multiplication des processeurs résolvant un même problème en coopérant (parallélisation massive). Dans les années 90 et 2000 cette multiplication s'est faite principalement selon un modèle à mémoire distribuée. Depuis quelques années on "découvre" le potentiel pour le calcul scientifique des processeurs de cartes graphiques (Graphical Processing Unit, GPU). Ces processeurs rassemblent sur une même puce plusieurs dizaines voire centaines de coeurs de calcul strictement synchronisés (modèle SIMD) et beaucoup moins complexes qu'un processeur superscalaire de type x86-64. Des noeuds de ce type seront installés en 2009 dans le supercalculateur civil du CEA (CCRT).

### 1.2 Systèmes-modèles envisagés

La modélisation du temps et du climat est une grosse consommatrice de puissance de calcul. Néanmoins les GPUs ne peuvent exécuter tels quels ces codes numériques "complexes". Il est nécessaire d'évaluer le potentiel de ces architectures sur des systèmes de complexité intermédiaire représentatifs du type de calcul réalisés par ces modèles, tout en restant beaucoup plus manipulables. Cette approche est celle qui a conduit dans les années 1960 à l'adoption des méthodes numériques encore en cours aujourd'hui. Dans ce cadre, l'équation-modèle de référence est l'équation de Saint-Venant, décrivant l'évolution d'une couche de fluide homogène incompressible soumise à la gravité et la force de Coriolis :

$$\begin{aligned}\rho(\partial_t u - fv + u\partial_x u + v\partial_y u) + \partial_x p &= 0 \\ \rho(\partial_t v + fu + u\partial_x v + v\partial_y v) + \partial_y p &= 0 \\ \partial_t p + \partial_x(up) + \partial_y(vp) &= 0\end{aligned}$$

Ici  $(u, v)$  sont les deux composantes du vent/courant,  $p$  la pression au sol/fond, égale au poids de la colonne d'air/eau,  $\rho$  est la masse volumique de l'air/eau (constante) et  $f$  est le paramètre de Coriolis (constant).

Plusieurs variantes peuvent être envisagées sur cette base. La plus simple est la version linéarisée autour de  $p = p_0$  ( $\simeq$ équation des ondes), qui décrit par exemple la propagation d'un tsunami dans un bassin océanique. Pour l'atmosphère on utilisera plutôt un modèle à deux couches. On pourra se référer au poly du cours MEC433 et au contrôle 2008 pour plus de discussion sur le contenu physique de ces équations et les équations à  $N$  couches.

## 2 Projet

### 2.1 Environnement de travail

Le travail se basera sur un code existant en langage C résolvant les équations d'évolution précédentes selon la discrétisation spatiale de Sadourny (différences finies centrées d'ordre 2 sur grilles régulières décalées) et une discrétisation temporelle de type saute-mouton ou Runge-Kutta. Il se fera en principe en environnement Linux.

Le travail de développement se fera en CUDA, une extension de C ajoutant quelques mots-clés permettant d'indiquer quelles routines doivent s'exécuter sur le GPU et dans quelle type de mémoire résident les variables/tableaux manipulés.

## 2.2 Travail demandé

L'exécution d'un pas de temps peut se décomposer en succession de tâches élémentaires simples représentant chacune une dizaine de lignes de codes :

- opérations purement locales : multiplication, addition de deux tableaux, p.ex. :

$$a_{i,j} := b_{i,j} + c_{i,j}$$

- opérations faisant intervenir les voisins : moyenne ou différence entre points de grille immédiatement voisins, p.ex. :

$$a_{i,j} := b_{i,j} + b_{i,j+1}$$

Ces opérations seront implémentées en CUDA. Des tests unitaires seront réalisés pour vérifier qu'elles donnent les mêmes résultats que l'implémentation CPU. Il s'agira ensuite d'analyser/améliorer la performance à la fois en termes de calcul (Gflops) et de débit mémoire (Go/s), ce dernier étant généralement le facteur limitant pour ce type de calcul. Par conséquent il faudra probablement dans un deuxième temps regrouper plusieurs opérations élémentaires afin d'améliorer le ratio calcul/transfert et s'approcher de la performance théorique du matériel (800 GFlops selon le constructeur).

## 2.3 Cas-tests

académique :

- réponse impulsionnelle des équations linéarisées
- interaction de tourbillons co-rotatifs ou contra-rotatifs (une couche)
- turbulence en déclin (une couche)

Un peu moins académique :

- réfraction d'une onde de tsunami dans un bassin (équations linéarisées avec bathymétrie réaliste)
- formation des dépressions (deux couches - assez académique quand même)