

UE : Dimensionnement et évaluation des architectures (I5AISE51)

Sujet de TD

Institut National des Sciences Appliquées de Toulouse
P.-E. Hladik, pehladik@insa-toulouse.fr

TD 3 : kernel multidimension

Version bêta (8 janvier 2021)

1 Ensemble de Julia en CUDA

Objectif 1.1

— réaliser un kernel multidimension

(1.1) Travail à faire : Ensemble de Julia

Les ensembles de Julia est un exemple d'ensembles fractals. Considérant deux nombres complexes, c et z_0 , l'ensemble de Julia correspondant est la frontière de l'ensemble des valeurs initiales z_0 pour lesquelles la suite

$$z_{n+1} = z_n^2 + c$$

est bornée.

Nous allons représenter cet ensemble par une image où chaque pixel sera considéré comme un nombre complexe z_0 dans un plan où 0 est le centre de l'image. Si la suite (z_n) est bornée pour le pixel z_0 le pixel prend la couleur rouge, sinon il prend la couleur noir.

Un code séquentiel permettant de représenter l'ensemble de Julia est déjà produit (pour $c = -0.8 + 0.156i$). Le but est de passer ce code sur un GPU.

1. Récupérer l'archive TD3.
2. Compilez le code avec `nvcc -o julia julia_bmp.cu`
3. Exécutez le code et observez la belle image produite.
4. Modifier le code pour l'exécuter sur le GPU et observez les performances!

(1.1) Comment faire : Données

Avant toute chose il faut penser à :

- allouer les espaces mémoire de travail sur le *device*,
- faire les copies de l'*host* vers le *device* (si besoin),
- récupérer à la fin les données du *device* vers l'*host*.

(1.2) Comment faire : `__device__`

N'oubliez pas que pour appeler une fonction sur le GPU il faut qu'elle soit déclarée comme `__device__`, même les méthodes d'un objet...