

TP2 MPI

Michel Donnet

November 8, 2023

Contents

1	Explication du code	2
1.1	Grid.cpp & Grid.h	2
1.2	writer.cpp & writer.h	3

1 Explication du code

Mon code est composé d'un fichier main.cpp, d'un fichier Grid.cpp avec son header Grid.h, et d'un fichier writer.cpp avec son header writer.h. Un makefile est présent pour compiler le tout.

J'ai décidé de créer une grille à 1 dimension pour mon code, afin de faciliter les transferts de donnée entre processeurs... Dans mon vecteur 1 dimension se trouve toutes les données de la grille 2D utilisée pour résoudre l'équation de chaleur.

La grille sur laquelle on va appliquer l'équation de chaleur est carrée, et le code ne fonctionne pas si le nombre de processeurs utilisés ne divise pas le nombre de colonnes de la grille.

1.1 Grid.cpp & Grid.h

On a défini dans Grid.cpp les fonctions dont la signature apparait dans Grid.h, et qui sont utilisées pour calculer l'équation de chaleur... On a:

```
#ifndef __GRID_H__
#define __GRID_H__

#include <vector>

using namespace std;
using std::vector;

// Allow us to create a grid
vector<double> create_grid(int size);

// Print our 1D vector as if it was a 2D vector with shape size * size
void print_grid(vector<double> grid, int size);

// Applies heat equation on the n eme element of our grid...
// We indicate to the function that the size of our imaginary 2D grid is size, so we can
// access to elements in our grid like if the grid was a 2D grid
double compute_element(int n, vector<double> grid, int size);

// Return the first column of a vector given, which represent part of grid with shape size
vector<double> first_column(vector<double> grid, int size);
// Return the last column of a vector given, which represent part of grid with shape size
vector<double> last_column(vector<double> grid, int size);
```

```
#endif
```

Comme j'ai mis une documentation au dessus de chaque fonction pour indiquer ce que celle-ci fait, je vais vous épargner une répétition inappropriée. . .

J'ai également défini dans Grid.cpp une fonction permettant de convertir l'index d'un tableau 1D en tableau 2D pour une taille de tableau 2D donnée (sachant que le tableau 2D est carré).

Ainsi, j'ai modifié la fonction *compute_element* pour que celle ci puisse fonctionner pour calculer l'équation de chaleur sur un vecteur 1 dimension, donc j'ai dû décomposer l'index n en l'index i et l'index j pour accéder à mon vecteur comme s'il était avec 2 dimensions, et j'ai ensuite appliquée la formule donnée dans le code à disposition.

Les fonctions *first_column* et *last_column* sont utiles pour la décomposition du problème entre les processeurs..

1.2 writer.cpp & writer.h

Ici, j'ai repris la fonction writer.h donnée. Cependant, j'ai simplement mis sa signature dans le dossier .h , et tout ce qui était dans le code initial writer.h se trouve dans le fichier writer.cpp, moins le '#pragma once' car on n'inclus plus le fichier dans d'autres fichiers, mais seulement le header du fichier.

J'ai également modifié la fonction donnée en donnant non pas un vecteur 2D, mais un vecteur 1D. Pour cela, j'ai dû changer la signature de la fonction (en remplaçant le vecteur de vecteur par un vecteur simple), et la ligne

```
double value = ((data[iY][iX] - minval) / (maxval - minval));

// Changed to 1d vector:
double value = ((data[iY + iX * size] - minval) / (maxval - minval));
```