

# Programmation concurrente

Version 1.1 : programme itératif

Aurélien COLOMBET (ca309567)

Quentin SALMERON (sq000508)

22 février 2016

## 1. Introduction

Dans le cadre de ce projet, nous simulons un modèle très simple du phénomène de transfert de chaleur par conduction. Nous partons d'une plaque d'une taille définie (par les paramètres d'exécution du programme), avec une température de base, un contour à température froide fixe, et un centre à température chaude, fixe aussi.

## 2. Hypothèses et calcul des constantes

Notre programme prend en entrée divers paramètres qui lui permettent de définir des données nécessaires à la simulation. Une des données les plus importantes pour commencer est la définition de la taille de la plaque. Nous obtenons le nombre de lignes (et de colonnes) en faisant le calcul  $2^{N+4}$  (N compris entre 0 et 9), obtenant ainsi une plaque de taille minimale  $2^4 * 2^4 (= 16 * 16)$  et de taille maximale  $2^{13} * 2^{13} (=8192 * 8192)$ .

A partir de la taille obtenue, nous pouvons calculer les deux constantes utilisées lors de notre programme, nécessaires afin de trouver le centre de la plaque (la partie chauffante). Le simulateur calcule l'emplacement du coin supérieur gauche  $2^{N+4-1} - 2^{N+4-4}$  et celui du coin inférieur droit  $2^{N+4-1} + 2^{N+4-4}$ . La zone située entre ces deux points est la zone chauffante.

## 3. Algorithme

### a. Description des structures de données

Afin de représenter la plaque, nous avons choisi d'utiliser un tableau à deux dimensions (une matrice). Pour les besoins de notre algorithme, nous utilisons deux tableaux à deux dimensions, un tableau temporaire permettant d'effectuer les calculs intermédiaires, et un tableau contenant le résultat final de chaque itération (ce tableau ne contient que des valeurs correctes à une itération donnée).

## b. Description de l'algorithme

L'algorithme de diffusion de chaleur se déroule en trois étapes, les deux premières étant le calcul. Tout d'abord, nous parcourons la matrice de référence et remplissons la matrice temporaire en effectuant le calcul :

$$\left(\frac{1}{6} * cellule\ gauche\right) + \left(\frac{4}{6} * cellule\ milieu\right) + \left(\frac{1}{6} * cellule\ droite\right)$$

Nous parcourons maintenant la matrice temporaire (sur laquelle nous venons d'écrire), pour écrire sur la matrice de référence avec le même algorithme que précédemment, mais cette fois-ci à la verticale :

$$\left(\frac{1}{6} * cellule\ dessus\right) + \left(\frac{4}{6} * cellule\ milieu\right) + \left(\frac{1}{6} * cellule\ dessous\right)$$

Enfin, nous remplissons à nouveau la zone centrale avec la température chaude de départ (la partie centrale étant la partie chauffante, sa valeur change durant les étapes de diffusion, mais doit redevenir à la température chaude une fois les calculs effectués).

## 4. Conclusion

### a. Résultats obtenus

Par défaut, le programme exécute les options : -i 10000 -e 012345 -s 024 -t 13 -m. A notre avancée, cela correspond à effectuer 10 000 itérations (option 'i') sur une plaque de 16\*16 ( $2^{0+4}$ ), puis sur une plaque de 64\*64 ( $2^{2+4}$ ), puis sur une plaque de 256\*256 ( $2^{4+4}$ ), et pour chacune des plaques, effectuer dix fois ces itérations pour faire une moyenne de consommation du CPU (option 'm').

La consommation du CPU correspond au temps pris pour appliquer l'algorithme, divisé par le nombre de tics d'horloge du CPU par seconde. En lançant notre programme avec les options citées précédemment, nous obtenons un temps de 0 tic d'horloge CPU pour une plaque de taille 16\*16, un temps de 1 tic d'horloge CPU pour une plaque de taille 64\*64, et un temps de 22 tics d'horloge CPU pour une plaque de taille 256\*256.