PERSENDA Edgar

HARDY Corentin

Projet Programmation Concurrente

Polytech'Nice Sophia - SI4

22 février 2016

Introduction

Ce rapport traite de la version 1.1 du projet de Programmation Concurrente. Ce projet consiste à programmer un simulateur de transfert de chaleur par conduction sur un objet plan. On prend comme hypothèse que les cases centrales sont à une température fixe et plus élevée que le reste de la plaque. De même, les cases sur le bord du plan sont à une température fixe. La température sur le reste de la plaque est la même que sur le bord avant la simulation.

La taille des tableaux traités va de $2^4 * 2^4$ à $2^{13} * 2^{13}$.

Il n'y aucun threads dans cette version, ils seront implémentés dans la version suivante.

Détail de l'algorithme

Constantes

On définit :

- h le pas dans l'espace
- k le pas dans le temps
- D la diffusion thermique de notre objet plan

La formule suivante est celle que l'on utilise pour notre algorithme :

$$T_{x,t+k} = \frac{T_{x+h,t} + (H-2)T_{x,t} + T_{x-h,t}}{H} \quad \text{où} \quad H = \frac{h^2}{kD}$$

Elle est valide pour un maillage fin, avec $H \geq 2$. On prend par hypothèse $H = 6$. Si on utilise cette formule pour former une matrice carrée de taille trois, on trouve la matrice suivante :

1/36	4/36	1/36
4/36	16/36	4/36
1/36	4/36	1/36

On peut retrouver cette matrice en appliquant notre formule de diffusion thermique dans un objet plan modélisé par 9 cases, ou seule la case centrale a une température supérieure aux autres à un temps t0. Les cases ont à t0 une température nulle. On retrouve cette matrice après une itération.

L'algorithme

Initialisation de deux tableaux à 2 dimensions, un qui représente le tableau des températures avec les bords, l'autre qui sert a stocker le résultat de la diffusion.
Initialisation du centre du tableau à la température haute et le reste à la température faible.

Pour un itérateur i de 0 au nombre d'application voulu :

On parcourt l'ensemble du premier tableau à l'exception des bords :

À chaque case, on modifie les 8 cases adjacentes du second tableau selon la valeur de la case actuelle, ainsi que la case actuelle. On applique les coefficients de la matrice définie précédemment pour les calculs.

On recopie le tableau de résultat dans le premier tableau et on réinitialise le tableau de résultat.
On réinitialise les cases du bord, ainsi que les cases centrales du premier tableau.

Détails

La réinitialisation des bords se fait en parcourant la première ligne du tableau et la dernière afin de les remettre à leurs valeurs initiales. De même pour la première et dernière colonne. Enfin, on réinitialise les valeurs du centre (entre $2^{n-1} – 2^{n-4} + 1$ et $2^{n-1} + 2^{n-4} + 1$ en ligne et colonne).

Notre algorithme est en $O(kn^2)$, avec k le nombre d'itérations et n la taille du tableau. La taille mémoire est de deux fois la taille d'un tableau (soit $2*(n+1)^2$).

Synthèse des mesures

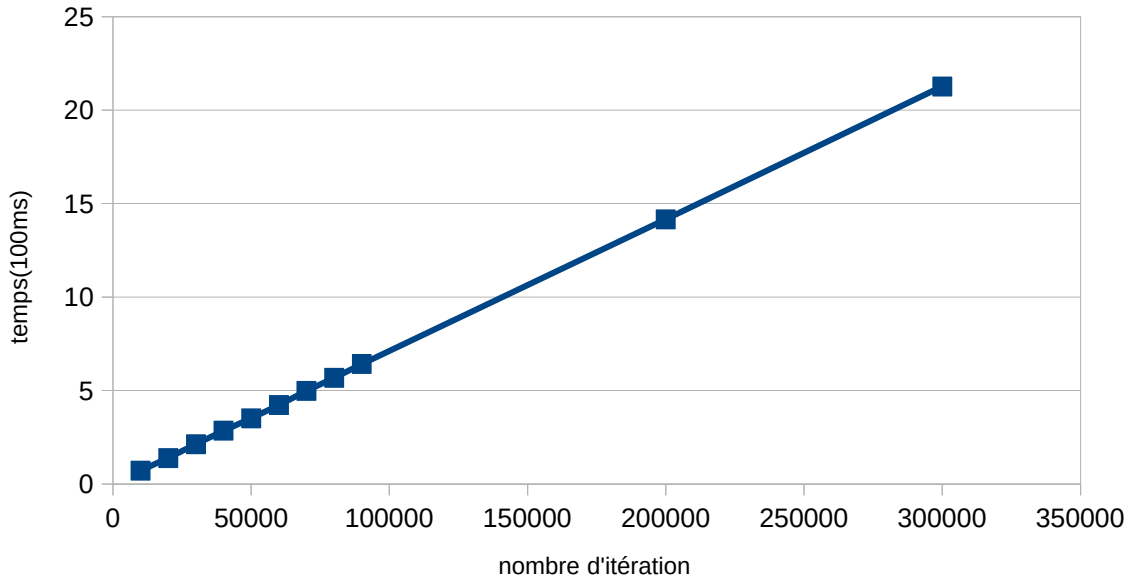

Illustration 1: Courbe du temps d'exécution en fonction du nombre d'itérations

L'illustration 1 représente le temps d'exécution en fonction du nombre d'itérations. Pour une simplification de lecture, les valeurs sont de 10000 à 100000, avec un pas de 10000, puis 200000 et 300000. Cette courbe est bien linéaire. Ces mesures sont faites avec un tableau de taille $2^4 * 2^4$.

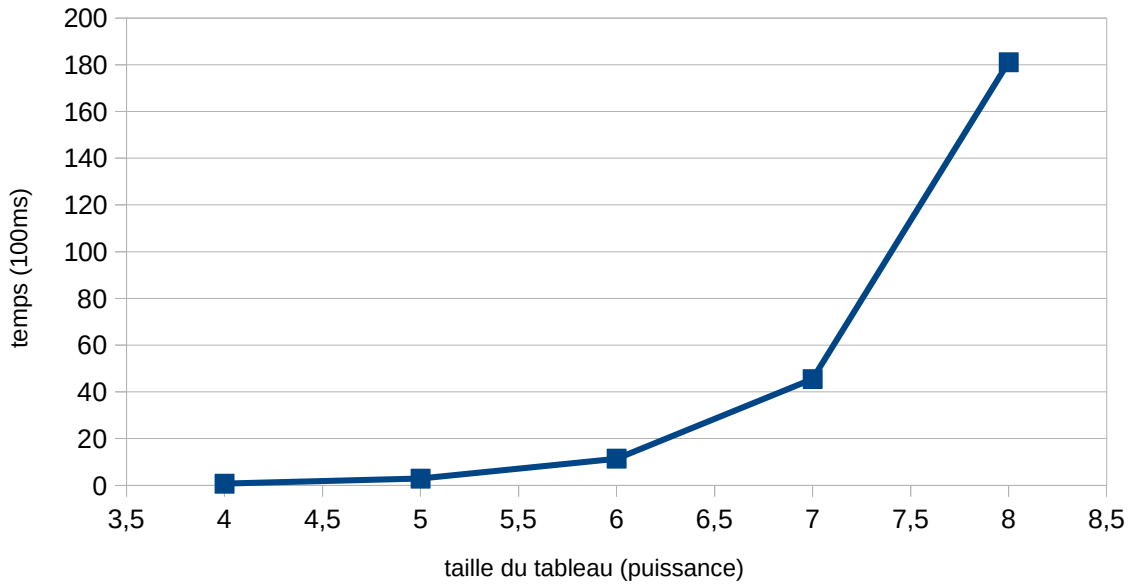

Illustration 2: Courbe du temps d'exécution en fonction de la puissance de la taille du tableau

L'illustration 2 représente le temps d'exécution en fonction de la puissance de la taille du tableau. Sa

forme est bien celle que l'on attendait. En effet, comme il s'agit de puissance de deux, la taille du tableau est multipliée par quatre pour chaque pas.

Les résultats précédents sont des moyennes de dix exécutions de l'algorithme, auquel on a enlevé le temps maximal et minimal d'exécution. Le temps utilisé est le temps CPU consommé.

Conclusion

L'algorithme que nous avons implémenté paraît satisfaisant. Le temps d'exécution CPU consommé évolue normalement en fonction du nombre d'itération ou de la taille du tableau. Les résultats produits par l'algorithme sont aussi cohérents.

Source

La formule de $T_{x,\ t+k}$ est une citation du site suivant :

http://robert.mellet.pagesperso-orange.fr/diff/diff_01.htm