

Rapport projet simulateur

Programmation Concurrente

La continuation de la propagation de chaleur de $t=0$ à $t=10$

0.	110,20,20,20,20,20,20-20,20,20-20
1.	110,22,20,20,20,20,20-20,20,20-20
2.	110,25,20,20,20,20,20-20,20,20-20
3.	110,27,20,20,20,20,20-20,20,20-20
4.	110,29,20,20,20,20,20-20,20,20-20
5.	110,31,20,20,20,20,20-20,20,20-20
6.	110,33,21,20,20,20,20-20,20,20-20
7.	110,35,21,20,20,20,20-20,20,20-20
8.	110,37,21,20,20,20,20-20,20,20-20
9.	110,39,22,20,20,20,20-20,20,20-20
10.	110,40,22,20,20,20,20-20,20,20-20

Description en pseudo code de chacun des algorithmes implémentés

Algorithme de passage à la température suivante pour un slot (un slot représente 4 cm de mur). Application de la formule suivante :

$$T(x,t+1) = T(x,t) + C*(T(x+1,t) + T(x-1,t))$$

Ce calcul est fait par la fonction *nextTemperature(numberOfSlot, time)*.

En prenant la position du slot (endroit dans le mur) concerné ainsi que la position dans le temps désirée cette méthode renvoie une température en *Double*.

La méthode *nextIteration()* s'occupe du passage d'une itération à une autre itération.

Pseudo code:

```

    boucle for ( j = 1; j < nombreDeSlot-1; j++) {
        //liste des temperatures
        nouvelleIteration.add(j,nextTemperature(j, tempsCourant));
    }
    // savoir si la face interne du mur a changer de temperature
    [...]
    //Ajout de la nouvelle iteration à la Map de stockage
    this.evolution.put(this.evolution.size(), nouvelleIteration);

```

Le code est très simplifié et incomplet mais ce pseudo code résume ce que fait la méthode *nextIteration()*.

Le temps d'exécution pour 100.000 cycles de simulation

Afin de mesurer le temps d'exécution selon les solutions. Il est évident que pour mesurer le temps d'exécution toute trace d'affichage doit être retirée. Vous devez expliquer dans votre rendu comment vous avez mesuré le temps d'exécution.

J'ai mesuré le temps d'exécution en stockant le `System.nanoTime()` avant et après l'exécution de la boucle principale de calcul. La différence de ces 2 valeurs donne le temps d'exécution.

Temps pour 100000 iterations : 0.33031482s

Le numéro de l'itération qui voit la face interne du mur (celle qui ne reçoit pas le soleil) changer de température

C'est à l'itération numéro 2086 que la face interne du mur change de température.

Calcul des différentes constantes.

Pour calculer C :

$$C = (\text{conductivitéThermique} * dt) / (\text{masseVolumique} * \text{ChaleurMassiqueSpécifique} * dx^2)$$

où dt est le temps entre 2 calculs et dx l'épaisseur d'une tranche (d'un slot)

C est donc fortement dépendant du matériaux mais ne change pas pour un matériau donné.

Cette constante peut donc être calculée au début pour les différentes matières.

Combien faut-il de pas d'itération pour simuler une année complète ?

$$1 \text{ an} = 365 \text{ jours} = 8760 \text{ heures} = 525\,600 \text{ minutes} = 31\,536\,000 \text{ secondes}$$

Dans notre cas, nous effectuons 1 itération toutes les 1 secondes. C'est notre pas. nous aurons donc : **31 536 000 itérations sur une année complète.**