Projet Maths pour l'info

—IMAC 1—

Equation de la chaleur

L'équation de la chaleur décrit le mode de propagation de la chaleur par conduction. Prenons une pièce pourvue d'un radiateur et d'une fenêtre, admettons qu'au temps t, la pièce ait une température uniforme, comment la chaleur va-t-elle se propager?

Ce projet est à réaliser en langage C et en binôme. Le rapport devra être déposé au secrétariat avant le 15/01/2003 . Une version électronique comprenant le programme et le rapport devra être envoyée au chargé de TD.

1 Position du problème

Nous allons nous intéresser au traitement numérique de l'équation de la chaleur :

$$\frac{\partial \theta}{\partial t} - \frac{K}{\rho c} \cdot \nabla \theta = \frac{W}{c}$$

soit en 2 dimensions

$$\frac{\partial \theta}{\partial t} - \frac{K}{\rho c} \cdot \left(\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} \right) = \frac{W}{c}$$

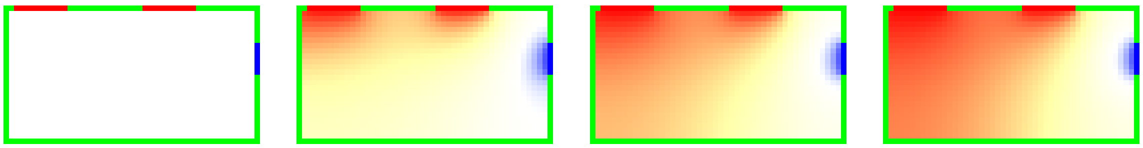
avec :

- $\theta(x, y)$: la température au point (x,y).
- K : la conductivité thermique du matériau considéré.
- ρ : la densité du matériau considéré.
- c : la capacité calorifique du matériau considéré.
- W : la quantité de chaleur fournie (pour une source).

Cette équation nécessite de spécifier les conditions initiales.

2 Travail demandé

Le but du projet est d'implémenter une application permettant de visualiser sur des images la propagation de la chaleur dans une scène donnée. Le programme doit lire un fichier contenant toutes les informations sur la scène (taille, localisation et température des sources, ...), calculer sur plusieurs itérations la propagation de la chaleur en utilisant 2 méthohdes différentes et la sauvegarger sous forme d'images au format *.ppm*.



Initialement, la pièce est à une température uniforme.
Sur les 3 images de droite, on voit la chaleur se propager.

3 Résolution numérique de l'équation

Une difficulté des méthodes numériques consiste à discrétiser un problème continu. C'est le cas ici, où il faut effectuer une discrétisation en espace et en temps. Pour cela, On se donne un pas spacial h et un pas temporel k. Le pas spacial nous indique que nous allons chercher à calculer les valeurs des températures aux points de coordonnées $(x, y) = (x_0+i.h, y_0+j.h)$. De même, le pas temporel nous indique que les solutions seront estimées aux temps $t = t_0 + n.k$. Dans la suite, nous noterons $\theta^n_{i,j}$ la température au temps $t_0 + n.k$ du point de coordonnées $(x_0 + i.h, y_0 + j.h)$.

3.1 méthode explicite

La dérivée temporelle peut être approchée par une méthode de dérivée "avant" du type :

$$\frac{\partial \theta}{\partial t} \simeq \frac{\theta^{n+1}_{i,j} - \theta^n_{i,j}}{k}$$

De même, la dérivée spatiale peut être approchée par une méthode "centrée" du type :

$$\frac{\partial^2 \theta}{\partial x^2} \simeq \frac{\theta^n_{i-1,j} - 2.\theta^n_{i,j} + \theta^n_{i+1,j}}{h^2} \qquad \text{pour la derivee selon } x$$

$$\frac{\partial^2 \theta}{\partial y^2} \simeq \frac{\theta^n_{i,j-1} - 2.\theta^n_{i,j} + \theta^n_{i,j+1}}{h^2} \qquad \text{pour la derivee selon } y$$

Finalement, l'équation de la chaleur devient :

$$\frac{\partial \theta}{\partial t} - \frac{K}{\rho c} \cdot \left(\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} \right) \simeq \frac{\theta^{n+1}_{i,j} - \theta^n_{i,j}}{k} - \frac{K}{\rho c} \cdot \left(\frac{\theta^n_{i-1,j} - 2.\theta^n_{i,j} + \theta^n_{i+1,j}}{h^2} + \frac{\theta^n_{i,j-1} - 2.\theta^n_{i,j} + \theta^n_{i,j+1}}{h^2} \right)$$

Il faut donc résoudre une équation du type :

$$\theta_{i,j}^{n+1} - \theta_{i,j}^{n} - \frac{k.K}{\rho c h^2} \cdot \left(\theta_{i-1,j}^{n} - 2.\theta_{i,j}^{n} + \theta_{i+1,j}^{n} + \theta_{i,j-1}^{n} - 2.\theta_{i,j}^{n} + \theta_{i,j+1}^{n}\right) = \beta(i,j,n)$$

avec

$$\beta(i,j,n) = \frac{kW}{c}$$

Le terme $\beta(i,j,n)$ correspond à l'énergie apportée au système (à la pièce). Cette énergie n'est pas fournie à toute la pièce dans sa globalité mais seulement localement autour des sources. Il en résulte que $\beta(i,j,n) = 0$ si le point (i,j) ne correspond pas à une source. Pour simplifier, nous poserons $\beta(i,j,n) = 0$ pour tous les points de la pièce et nous ne recalculerons pas les températures des sources de chaleur et supposerons qu'elles sont constantes. Il n'en serait pas de même s'il on voulait prendre en compte l'influence de l'air environnantsur les sources. Nous reviendrons dessus dans la suite.

- Montrer que l'équation précédente peut se mettre sous la forme :

$$\theta_{i,j}^{n+1} = \theta_{i,j}^{n} + \alpha \cdot \left(\theta_{i-1,j}^{n} + \theta_{i+1,j}^{n} + \theta_{i,j-1}^{n} + \theta_{i,j+1}^{n} - 4.\theta_{i,j}^{n}\right) + \beta(i,j,n)$$

- Expliquer graphiquement ce que représente cette équation.

3.2 méthode implicite

Les méthodes explicites sont parfois instables et il est souvent préférable d'utiliser des méthodes implicites.

La dérivée temporelle n'est plus approchée par une méthode de dérivée "avant" mais par une méthode de dérivée arrière :

$$\frac{\partial \theta}{\partial t} \simeq \frac{\theta_{i,j}^{n} - \theta_{i,j}^{n-1}}{k}$$

La dérivée spatiale reste approchée par une méthode "centrée" :

$$\frac{\partial^2 \theta}{\partial x^2} \simeq \frac{\theta_{i-1,j}^{n} - 2.\theta_{i,j}^{n} + \theta_{i+1,j}^{n}}{h^2} \qquad \text{pour la derivee selon } x$$

$$\frac{\partial^2 \theta}{\partial y^2} \simeq \frac{\theta_{i,j-1}^{n} - 2.\theta_{i,j}^{n} + \theta_{i,j+1}^{n}}{h^2} \qquad \text{pour la derivee selon } y$$

La différence apparait quand on rassemble les deux formules. En effet, la dérivée temporelle et la dérivée spatiale sont appliquées au temps $n+1$. On obtient :

$$\frac{\partial \theta}{\partial t} - \frac{K}{\rho c} \cdot \left(\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2}\right) \simeq \frac{\theta_{i,j}^{n+1} - \theta_{i,j}^{n}}{k} - \frac{K}{\rho c} \cdot \left(\frac{\theta_{i-1,j}^{n+1} - 2.\theta_{i,j}^{n+1} + \theta_{i+1,j}^{n+1}}{h^2} + \frac{\theta_{i,j-1}^{n+1} - 2.\theta_{i,j}^{n+1} + \theta_{i,j+1}^{n+1}}{h^2}\right)$$

- Montrer que l'équation précédente peut se mettre sous la forme :

$$\theta_{i,j}^{n+1} - \alpha\left(\theta_{i-1,j}^{n+1} + \theta_{i+1,j}^{n+1} + \theta_{i,j-1}^{n+1} + \theta_{i,j+1}^{n+1} - 4.\theta_{i,j}^{n+1}\right) = \beta(i,j,n) + \theta_{i,j}^{n}$$

- Cette fois, le calcul de $\theta_{i,j}^{n+1}$ est plus compliqué. Nous sommes face à un système d'équations linéaires. Il faudra le résoudre à chaque itérations. Vérifier que le système à résoudre est de la forme :

$$\begin{pmatrix} A & B & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ B & A & B & 0 & \ddots & & & \vdots \\ 0 & B & A & B & 0 & \ddots & & \vdots \\ 0 & 0 & B & A & B & 0 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & 0 & B & A & B & 0 \\ \vdots & & & & 0 & B & A & B \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & B & A \end{pmatrix} . \begin{pmatrix} \theta_{0,0}^{n+1} \\ \theta_{1,0}^{n+1} \\ \vdots \\ \theta_{N,0}^{n+1} \\ \theta_{0,1}^{n+1} \\ \theta_{1,1}^{n+1} \\ \vdots \\ \theta_{i,j}^{n+1} \\ \vdots \\ \vdots \\ \vdots \\ \theta_{N,M}^{n+1} \end{pmatrix} = \beta + \begin{pmatrix} \theta_{0,0}^{n} \\ \theta_{1,0}^{n} \\ \vdots \\ \theta_{N,0}^{n} \\ \theta_{0,1}^{n} \\ \vdots \\ \vdots \\ \theta_{i,j}^{n} \\ \vdots \\ \vdots \\ \vdots \\ \theta_{N,M}^{n} \end{pmatrix}$$

avec

$$A = \begin{pmatrix} 4\alpha+1 & -\alpha & 0 & \cdots & 0 \\ -\alpha & 4\alpha+1 & -\alpha & \ddots & \vdots \\ 0 & -\alpha & 4\alpha+1 & -\alpha & 0 \\ \vdots & \ddots & -\alpha & 4\alpha+1 & -\alpha \\ 0 & \cdots & 0 & -\alpha & 4\alpha+1 \end{pmatrix}$$

et

$$B = \begin{pmatrix} -\alpha & 0 & \cdots & 0 \\ 0 & -\alpha & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\alpha \end{pmatrix}$$

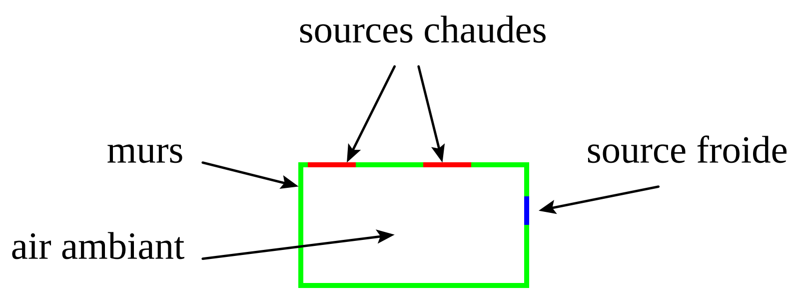
Ce système satisfait-il les conditions nécessaires pour être résolu avec les méthodes de Jacobi et Gauss-Seidel ?
La matrice principale est dite "creuse" (constituée majoritairement de 0). Trouver une adaptation de la méthode de Jacobi (ou Gauss-Seidel au choix) pour éviter une grande quantité de calculs inutiles. Quelle est la complèxité de votre méthode.

3.3 condition initiales, conditions aux limites

Ce genre de méthode nécessite de spécifier des condition initiales. Dans notre cas, il s'agit de la tempréature de la pièce, de celle des sources de chaleurs (radiateurs, fenêtres, ...).
Il est aussi nécessaire de connaitre conditions aux limites. En effets, les points situés sur le bord de la scène n'ont pas de voisin du coté extérieur, il faut donc leur affecter un traitement spécial. Pour plus de simplicité, nous conviendrons que la scène est une pièce close (ie elle est entourée de murs) et que les murs n'échangent pas d'énergie avec l'air environnant. Il est possible de substituer une partie des murs par des source de chaleurs chaudes (des radiateurs) ou froides (des fenêtres). Nous conviendrons donc que l'air environnant ne peut pas influer sur la température de ces sources.
En résumé, il existe donc trois sortes d'éléments :

- l'air environnant (qui intéragit avec lui-même).
- diverses sources de chaleurs (qui agissent sur l'air uniquement).
- les murs (qui n'interagissent pas avec les autres éléments).

Voici un exemple de scène :



Les murs entourent la pièce.
Il sont parfois substitués par un radiateur ou une fenêtre.

Les murs n'ont pas de temprature propore. Pour calculer la température du point (i,j) voisin d'un mur, il est possible de considérer que, durant ce calcul, le mur est à la même température que le point.

4 Format des fichiers décrivants les scènes

Vous allez devoir manipuler 2 sortes de fichiers, les fichiers décrivant la scène, que nous noterons "fichier.*sc*" et les fichiers image *ppm*.

4.1 fichier.*sc*

Ces fichiers permettent de décrire une scène, notamment ses dimensions et la disposition des sources de chaleur. Voici plus en détails les informations contenus dans ces fichiers :

- La *largeur* et la *hauteur* (en nombre de pixels).
- Le pas spatial h et le pas temporel k.
- La conductivité, la densité et la capacité calorifique.
- Le nombre *nbSources* de températures à spécifier (au moins celle de l'air environnant et celle d'une source) Il y aura moins de 10 temératures différentes.
- Les températures proprement dites, **en commençant par celle de l'air environnant**.
- Le plan de la scène.

Le plan de la scène est composé d'éléments. Un élément est soit un mur, soit de l'air, soit une source. A chaque éléments est associé un numéro : on numérote les éléments dans l'ordre d'énumération des températures.

- Le numéro 0 correspond donc à l'air.
- Le numéro 1 à la première source de chaleur.
- Le numéro 2 (s'il existe) à la seconde source de chaleur.
- ...
- Le numéro *nbSources* correspond à un mur, aucune température n'est associée à ce numéro.

On discrétise un plan en le découpant de telle sorte qu'il fasse en largeur et en hauteur les dimensions spécifiées plus haut. Chaque élément de surface correspond à un élément. Pour composer un plan, on écrit les numéros les uns à la suite des autres en lisant ligne par ligne (de gauche à doite) en commençant par le haut.

par exemple:

```
largeur 6
hauteur 4
h 0.001
k 0.001
conductivite 0.025
densite 0.1245
capacite 1004.0

source 2
20
40
221122000022000022222222
```

La température de l'air ambiant sera à $20°c$ et sera numéroté 0.
La seule source de chaleur sera à $40°c$ et sera numérotée 1.
Les murs seront donc numérotés 2.

Voici le plan avec une source de chaleur en haut :

221122
200002
200002
222222

Voici la façon de l'écrire dans le fichier.*sc*, tous les éléments sur la même ligne :

221122000022000022222222

4.2 fichier.*ppm*

Il s'agit d'un format de fichier reconnu par divers éditeurs d'images comme *gimp* ou *ee* (electric eye). Il en existe plusieurs variantes : couleurs/noir et blanc, ASCII/binaire. Le format le mieux adapté à notre problème est le format couleurs en binaire. Le fichier commencera donc avec un en-tête correspondant.

Un pixel en couleur est défini par 3 nombres : un pour "la quantité" de rouge (R), un pour "la quantité" de vert (V) et un pour "la quantité" de bleu (B), noté RVB (RGB en anglais). Nous coderons ces quantités par des *unsigned char*.Un *unsigned char* est codé sur 8 bits, soit un octet. Ces nombres vont donc de 0 à 255.

Revenons au format *ppm*. Le premier nombre correspond au "numéro de variante" choisi, dans notre cas *P6*. Sur la ligne suivante, les deux nombres correspondent à la *largeur* et la *hauteur*. Enfin, le dernier nombre correspont à la valeur maximale que l'on puisse rencontrer (ici 255). Des commentaires peuvent être insérés dans l'en-tête, une ligne de commentaires commence par un #.
Chaque pixel sera écrit dans le fichier en procédant ligne par ligne, de gauche à doite, en commençant par le haut. Les pixels seront écrits sous forme de 3 octets écrits les uns à la suite des autres (RVB).

exemple:

```
P6
128 256
255
^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y
^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y
^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y
^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y
^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y^@y
^@y^@y^@y ...
```

---

5 mise en oeuvre du programme C

5.1 structures de données

3 structures sont nécessaires.

- une structure *temperature* contenant 2 tableaux servant à stocker les températures.
- une structure *source* contenant les informations concernant les températures des diverses sources de chaleur ainsi que leur position.
- une structure *scene* contenant un élément de type *temperature*, un élement de type *source* et toutes les constantes relatives à l'équation à résoudre.

Voici les trois structures à utiliser.

```
typedef struct temperature{
  int larg,haut;
  double *T1;          // tableau 2D des temperatures
  double *T2;          // tableau 2D des temperatures
  double *courant;     // pointeur sur le tableau courant
  double *precedent;   // pointeur sur le tableau precedent
}temperature;
```

```
typedef struct source{
  int larg,haut;
  unsigned char *sourceType;  // plan de la scene
  double *Tsource;            // temperature de fenetres, radiateurs, ...
  int nbSources;              // nb de source differentes
}source;
```

```
typedef struct scene{
  temperature T;
  source src;
  double h, k, conductivite, densite, capacite, constante;
  int numeroImage;   // concerne le nom des fichiers ppm
}scene;
```

Le stockage des matrices (2D) s'effectue dans des tableaux 1D, notament pour les tableaux T1, T2 et Tsource. *courant* et *precedent* sont des pointeurs sur les tableaux. Pour passer d'une itération à l'autre, on fait un "swap" sur les pointeurs plutôt qu'une recopie des tableaux.

Les images *ppm* génèrées se nomment $image000.pm$, $image001.pm$, $image002.pm$, etc ... Il est donc nécessaire d'utiliser un compteur $numeroImage$.

5.2 quelques directions

Dans un premier temps, il faut réaliser les fonctions de lecture des fichiers *.sc* et d'écriture. Il est nécessaire de faire une fonction donnant une couleur à partir d'une température. Une température "froide" sera plutôt bleue, une température "chaude" sera plutôt rouge, la température ambiante initiale sera blanche. Vous devrez enfin coder pour chaque mode de résolution (explicite et implicite) une fonction calculant la température en chacun des points de la scène pour une itération.
Ce devoir est axé sur deux parties : la méthode explicite et la méthode implicite. La méthode implicite est plus délicate, ne la commencez que quand la méthode explicite fonctionne correctement.

Il est très fortement recommandé de faire un fichier .c et un fichier .h pour chaque structure, un fichier supplémentaire contenant le programme principal et un Makefile pour compiler le tout. Vous utiliserez les notations de l'énoncé.

6 options

Il est conseillé de rajouter à votre programme quelques options. En voici quelques unes

- Pouvoir spécifier le fichier *.sc* en tapant son nom derière le nom de l'éxécutable :
  $>$ a.out fichier1.sc
- Faire vos propre scènes.
- Gérer l'influence de l'air sur une source de chaleur. Pour cela, il est recommandé de considérer les sources comme de l'air recevant de l'énergie. Si la case (i, j) représente une source, cette case est traitée comme une case normale à la diffénce près que le terme $\beta(i, j)$, représentant la quantité d'énergie reçue par cette case, est non nul. Il faudra en outre veiller à ce que les sources ne soient plus sur le bord de la scène, pour cela il suffit de rajouter une épaisseur de mur.

Vous pourez laisser libre cours à votre imagination, toute amélioration sera la bienvenue.

Bon courage.