

Equation de la chaleur

1 Informations pratiques

Ce projet est à effectuer en binômes. Dès qu'un binôme sera constitué, il enverra un mail à vincent.torri@gmail.com pour vérification.

Le code rendu sera écrit en C++ pour le développement. **Tout projet ne compilant pas se verra attribuer la note de 0.** Mieux vaut rendre un projet incomplet mais compilant, qu'un projet ne compilant pas. Si plusieurs binômes ont des codes trop similaires, leur note sera divisée par le nombre de binômes impliqués.

Le code devra être indenté de manière uniforme. La définition des méthodes et des fonctions ne devront pas dépasser un nombre de lignes raisonnable (au plus 50 lignes)

Le code devra être commenté en utilisant la syntaxe de Doxygen.

Le rapport sera écrit en \LaTeX et inclura les choix, les problèmes techniques qui se posent et les solutions trouvées (la conception (dont un diagramme UML complet) et réalisation). Le soin apporté à la grammaire et à l'orthographe sera largement pris en compte.

Le projet sera envoyé sous forme d'archive tar.xz (archive d'un répertoire ayant pour nom *NOM1_NOM2* et contenant les fichiers) ayant pour nom *NOM1_NOM2_projet_chaleur.tar.xz* avant le Dimanche 17 janvier 2021 à minuit à l'adresse vincent.torri@gmail.com. Le non respect du nom de l'archive sera pénalisé par 5 points en moins. L'archive contiendra le rapport en \LaTeX PDF, ainsi que les fichiers C++. **Tout projet rendu en retard se verra attribuer la note de 0.** Donc ne pas attendre le dernier moment pour l'envoyer. Un mail de confirmation sera envoyé.

2 Sujet

Le but de ce projet est de simuler l'évolution de la température dans un matériau. L'équation aux dérivées partielles simulant la propagation de la

chaleur s'appelle (à juste titre) l'équation de la chaleur :

$$\frac{\partial u}{\partial t} = \frac{\lambda}{\rho c} \Delta u + \frac{F}{\rho c}, \quad (1)$$

avec $u : \mathbb{R}_+ \times [0, L]^d \rightarrow \mathbb{R}_+$, $u(t, x)$ la température **en degré Kelvin** du matériau étudié au temps $t \in \mathbb{R}_+$ et en $x \in [0, L]^d$. Δ est le Laplacien sur $[0, L]^d$. $F : \mathbb{R}_+ \times [0, L]^d \rightarrow \mathbb{R}_+$ est la source de chaleur appliquée sous l'objet (F peut simuler une flamme ou une source de froid). Les quantités λ , ρ et c sont des grandeurs physiques, respectivement la conductivité thermique, la masse volumique et la chaleur massique du matériau constituant l'objet. Leur valeur dans les unités internationales pour les matériaux considérés dans ce projet est donnée dans le tableau ci-dessous.

	λ (W/(m.K))	ρ (kg/m ³)	c (J/(kg.K))
cuivre	389	8940	380
fer	80.2	7874	440
verre	1.2	2530	840
polystyrène	0.1	1040	1200

(2)

La température initiale est une température uniforme de 13 degrés Celsius, c'est-à-dire la valeur de $u(0, x)$, pour tout x .

Le projet consistera en la résolution de l'équation (1) en utilisant la méthode des différences finies, dans le cas implicite, en considérant 2 objets : une barre infiniment mince ($d = 1$) de longueur L et une plaque infiniment mince ($d = 2$).

Les conditions au bord, détaillées plus tard, sont mixtes :

- Pour la barre : condition de Neumann en $x = 0$ et de Dirichlet en $x = L$.
- Pour la plaque : condition de Neumann en $x = 0$ et $y = 0$, et de Dirichlet en $x = L$ et $y = L$.

Interprétation des conditions au bord

- La condition de Dirichlet dit : on impose sur le bord de l'objet une température (par exemple, on veut que le bord à droite de la barre soit toujours de 13 degrés Celsius).
- La condition de Neumann dit : la température s'échappe vers le milieu extérieur (l'air ambiant, par exemple)

Ce genre de conditions mixtes peut se rencontrer dans la vie courante : le freezer d'un réfrigérateur peut être modélisé par un parallélépipède rectangledont le côté supérieur a une température fixe (0 degré Celsius par exemple,

un condition de Dirichlet), tandis que le froid s'échappe par les autres côtés pour refroidir l'intérieur du réfrigérateur (conditions de Neumann).

3 Premier cas : la barre

L'équation devient :

$$\left\{ \begin{array}{lcl} \frac{\partial u}{\partial t} & = & \frac{\lambda}{\rho c} \frac{\partial^2 u}{\partial x^2} + \frac{F}{\rho c}, \quad \forall (t, x) \in [0, t_{\max}] \times [0, L], \\ u(0, x) & = & u_0, \quad \forall x \in [0, L], \\ \frac{\partial u}{\partial x}(t, 0) & = & 0, \quad \forall t \in [0, t_{\max}], \\ u(t, L) & = & u_0, \quad \forall t \in [0, t_{\max}], \end{array} \right. \quad (3)$$

On regarde la propagation de la chaleur jusqu'au temps t_{\max} et pour une barre de longueur L . A $t = 0$, la barre a une température uniforme (constante en x) u_0 . La source de chaleur F simule deux ajouts de chaleur, de température f et $\frac{3}{4}f$:

$$\begin{aligned} F(x) &= t_{\max} f^2, \text{ sur } \left[\frac{L}{10}, \frac{2L}{10} \right], \\ F(x) &= \frac{3}{4} t_{\max} f^2, \text{ sur } \left[\frac{5L}{10}, \frac{6L}{10} \right], \\ F(x) &= 0 \text{ sinon.} \end{aligned}$$

Application numérique : $L = 1\text{m}$, $t_{\max} = 16\text{s}$, $u_0 = 13$ degrés Celsius et $f = 80$ degrés Celsius. On discrétisera l'intervalle en espace et en temps avec 1001 points et on utilisera un schéma aux différences finies implicite.

ATTENTION : la valeur de $u(t, x)$ calculée est en degré Kelvin.

travail Ecrire un programme en C++ correctement structuré qui résoud (3) sans l'aide de bibliothèque extérieure à C++ pour tous les matériaux du tableau 2. Puis afficher une animation (affichage de la solution approchée de $u(t, \cdot)$ pour 100 valeurs de t réparties uniformément sur $[0, t_{\max}]$) en utilisant le programme Scilab), pour tous les matériaux (On écrira dans un fichier les valeurs de la solution approchée sous forme de colonne et on utilisera la fonction **read** de Scilab pour les stocker dans un vecteur).

Optimisation : utiliser les threads C++11 *std : :thread* pour accélérer les calculs. Effectuer des mesures à l'aide de *std : :chrono* pour mesurer le gain. Ne pas hésiter à choisir un nombre plus grand de points.

4 Deuxième cas : la plaque

L'équation devient :

$$\left\{ \begin{array}{lcl} \frac{\partial u}{\partial t} & = & \frac{\lambda}{\rho c} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \frac{F}{\rho c}, \quad \forall (t, x, y) \in [0, t_{\max}] \times [0, L]^2, \\ u(0, x, y) & = & u_0, \quad \forall (x, y) \in [0, L]^2, \\ \frac{\partial u}{\partial x}(t, 0, y) & = & 0, \quad \forall (t, y) \in [0, t_{\max}] \times [0, L], \\ \frac{\partial u}{\partial y}(t, x, 0) & = & 0, \quad \forall (t, x) \in [0, t_{\max}] \times [0, L], \\ u(t, L, y) & = & u_0, \quad \forall (t, y) \in [0, t_{\max}] \times [0, L], \\ u(t, x, L) & = & u_0, \quad \forall (t, x) \in [0, t_{\max}] \times [0, L]. \end{array} \right. \quad (4)$$

La source de chaleur F devient :

$$\begin{aligned} F(x, y) &= t_{\max} f^2, \text{ sur } \left[\frac{L}{6}, \frac{2L}{6} \right] \times \left[\frac{L}{6}, \frac{2L}{6} \right], \\ F(x, y) &= t_{\max} f^2, \text{ sur } \left[\frac{4L}{6}, \frac{5L}{6} \right] \times \left[\frac{L}{6}, \frac{2L}{6} \right], \\ F(x, y) &= t_{\max} f^2, \text{ sur } \left[\frac{L}{6}, \frac{2L}{6} \right] \times \left[\frac{4L}{6}, \frac{5L}{6} \right], \\ F(x, y) &= t_{\max} f^2, \text{ sur } \left[\frac{4L}{6}, \frac{5L}{6} \right] \times \left[\frac{4L}{6}, \frac{5L}{6} \right], \\ F(x, y) &= 0 \text{ sinon.} \end{aligned}$$

Application numérique : $L = 1\text{m}$, $t_{\max} = 16\text{s}$, $u_0 = 13$ degrés Celsius et $f = 80$ degrés Celsius. On discrétisera l'intervalle en espace (dans les deux directions x et y) et en temps avec 1001 points et on utilisera un schéma aux différences finies implicite.

ATTENTION : la valeur de $u(t, x)$ est en degré Kelvin.

travail Ecrire un programme en C++ correctement structuré qui résoud (4) sans l'aide de bibliothèque extérieure à C++ pour tous les matériaux du tableau 2. Puis afficher une animation (affichage de la solution approchée de $u(t, \cdot, \cdot)$ pour 100 valeurs de t réparties uniformément sur $[0, t_{\max}]$) en utilisant le programme Scilab), pour tous les matériaux (On écrira dans un fichier les valeurs de la solution approchée sous forme de tableau et on utilisera la fonction **read** de Scilab pour les stocker dans une matrice).

Optimisation : utiliser les threads C++11 *std::thread* pour accélérer les calculs. Effectuer des mesures à l'aide de *std::chrono* pour mesurer le gain. Ne pas hésiter à choisir un nombre plus grand de points.