Diffusion Thermique

Ce TP est consacré à la résolution approchée de l'équation de la diffusion thermique.

1 Résolution en une dimension sans source par la méthode des éléments finis ; schéma explicite

On considérera dans cette partie l'évolution de la température en fonction du temps pour un fil conducteur unidimensionnel : la température de ce fil sera donc une fonction de deux variables : $T(z,t)$.

Le fil conducteur électrique est initialement dans son régime stationnaire. Il est brutalement exposé à un refroidissement à 300K à ses bornes.

On va s'intéresser dans un premier temps à l'équation de la diffusion thermique en une dimension d'espace et sans source (D est appelé la diffusivité thermique du fil conducteur) :

$$\frac{\partial T}{\partial t} = D\,\frac{\partial^2 T}{\partial x^2} \qquad (E_0)$$

,

on considèrera la condition initiale suivante :

$$\forall x,\ T(x,0) = 300 + 100x/L \qquad (C_1)$$

et la condition aux limites suivante :

$$\forall t > 0,\ T(0,t) = T(L,t) = 300 \qquad (C_2)$$

La méthode des différences finies est une technique courante de recherche de solutions approchées d'équations aux dérivées partielles qui consiste à résoudre un système de relations (schéma numérique) liant les valeurs des fonctions inconnues en certains points suffisamment proches les uns des autres.

Ainsi, on va chercher à discrétiser notre problème. Nous ferons l'hypothèse, qui est celle utilisée pour la méthode d'Euler de résolution d'une équation différentielle, à savoir que la dérivée d'une fonction en z peut être approchée par l'accroissement de cette fonction entre z et $z + \Delta z$, où Δz désigne notre pas de discrétisation (pour la variable z).

Les valeurs de la température à un l'instant initial seront stockées dans une liste T_j^0 de longueur N. Il y aura donc N pas de discrétisation en espace. Les valeurs de la température à l'instant $t = \Delta t$ seront stockées dans une liste T_j^1, qui aura également comme longueur N. De même, les valeurs de la température à l'instant $t = n\Delta t$ seront stockées dans une liste T_j^n.

Question 1. Écrire une fonction `PasEspace(L,N)` qui renvoie le pas d'espace Δx en fonction de la longueur L sur laquelle on résout le problème et du nombre N de pas souhaités.

Le pas de temps Δt ne peut pas être choisi indépendamment du pas d'espace, pour des questions de stabilité de schéma numérique. On supposera donc que l'on dispose d'une variable globale DeltaT qui stocke la valeur de Δt. Aux fins d'applications numériques, la valeur de DeltaT sera précisée dans les questions où l'on en aura besoin.

Question 2. Montrer que l'on a dans le cadre de l'approximation des différences finies :

$$\frac{\partial T}{\partial t} \approx \frac{T_j^{n+1} - T_j^n}{\Delta t}$$

et que :

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{j+1}^n - 2T_j^n + T_{j-1}^n}{\Delta x^2}$$

Question 3. En déduire la relation de récurrence suivante pour $j \neq 0$ et $j \neq N-1$:

$$T_j^{n+1} = T_j^n + c\,(T_{j-1}^n - 2\,T_j^n + T_{j+1}^n)\,, \qquad \text{avec} \quad c = \frac{\Delta t\, D}{\Delta x^2}$$

.

Question 4. Comment s'écrit la relation de récurrence précédente lorsque $j = 0$ et lorsque $j = N-1$?

Question 5. Écrire une fonction `initialisation(L,N)` qui renvoie la liste comprenant les valeurs de T_j^0 respectant la condition initiale (C_1) de l'équation aux dérivées partielles (E_0).

Question 6. Écrire une fonction `transition(T,c)` qui prend en argument c et une liste T, contenant les valeurs de T^n pour un n quelconque, et renvoie la liste TP, constituée des éléments de T^{n+1}.

Question 7. Écrire une suite de commandes qui permet d'avoir une représentation graphique du profil en température toutes les 10 minutes. On prendra DeltaT = 1s ; N = 100 et $D = 1,2 \times 10^{-4}\ m^2.s^{-1}$. La température sera calculée jusqu'à un temps de 5 heures.

2 Résolution en deux dimensions

On s'intéresse désormais au problème de la diffusion thermique sans source avec deux dimensions d'espace. Celle-ci a pour équation :

$$\frac{\partial T}{\partial t} = D\,\Delta T \qquad (E_2)$$

Question 8. En s'inspirant de la résolution de ce problème en une dimension, montrer que le schéma suivant permet de résoudre numériquement cette équation :

$$T_{i,j}^{n+1} = T_{i,j}^n + \Delta t\, D\,\left[\frac{T_{i-1,j}^n - 2\,T_{i,j}^n + T_{i+1,j}^n}{\Delta x^2} + \frac{T_{i,j-1}^n - 2\,T_{i,j}^n + T_{i,j+1}^n}{\Delta y^2}\right]$$

Nous nous intéressons désormais au problème de la diffusion thermique autour d'une source rectangulaire de chaleur après son extinction. On résout ainsi le problème précédent sur le domaine d'espace carré $[-0,2;0,2] \times [-0,2;0,2]$.

Nous prendrons une discrétisation de 101 pas selon l'axe des abscisses et de 101 pas selon l'axe des ordonnées. La température à un instant sera ainsi codée dans une matrice 101×101

On suppose à l'instant initial que la température vaut 400 K si $(x,y) \in [-0,1;0,1] \times [-0,12;0,12]$ et vaut 300 K sinon.

Question 9. Écrire une fonction `Initial2d()` qui renvoie la matrice contenant les températures initiales. On commencera par chercher à quelle condition sur les lignes et les colonnes les coefficients de cette matrice valent 400 et à quelle condition ils valent 300.

Question 10. Écrire une fonction `transition(T,D,DeltaT,DeltaX,DeltaY)` qui renvoie la température après une itération du schéma numérique. Nous supposerons que nous avons une température de 300K à la frontière de la matrice.

Question 11. Écrire alors une fonction `température(n)` qui renvoie la température après n secondes. On prendra D=0,00012 et DeltaT=0,01 (on pourra observer que les solutions sont instables numériquement si on choisit DeltaT=0,1 dans la question suivante).

Question 12. Nous désirons faire une représentation en couleurs de la solution obtenue après un temps donné.

Pour cela, commencer par créer des **tableaux numpy** `XX` et `YY` des abscisses et ordonnées, équiréparties en 101 points entre -0,2 et 0,2.

Question 13. On donne le code suivant :

```
T=np.array(temperature(50))
plt.ion()
plt.figure()
p=plt.pcolormesh(XX,YY,T, shading='flat')
plt.colorbar(p)
plt.axis('image')
plt.draw()
```

Le tester et l'adapter pour avoir une représentation graphique toutes les secondes pendant une période de 10 secondes dans une nouvelle figure.

3 Schéma implicite en une dimension

Le schéma explicite (2) ne converge que si le pas de temps Δt est suffisamment faible par rapport au pas d'espace Δx. Ceci est caractérisé en informatique par le nombre de Courant (et la condition de Courant–Friedrichs–Lewy)

On définit $C_o = \dfrac{v\Delta t}{\Delta x}$ avec :

- v : vitesse dans la direction x
- Δt : intervalle temporel
- Δx : intervalle dimensionnel

Pratiquement, si C_0 est inférieur à un seuil, on observe une instabilité de calcul, erreur d'approximation dans des calculs numériques, grandissant rapidement au fur et à mesure des calculs. Si la dimension de la grille est inférieure à la distance parcourue dans l'intervalle de pas de temps par l'onde la plus rapide que permet l'équation, l'erreur grandit et envahit la solution physique.

Si l'on souhaite effectuer un calcul pour un temps physique long, beaucoup d'itérations seront nécessaires et le temps de calcul sera très long. C'est pourquoi on préfère d'autres types de schémas appelés schémas implicites.

Dans cette partie, la dérivée partielle seconde par rapport à x de la température apparaissant dans l'équation (1) est évaluée au point d'abscisse x_i et à l'instant k+1 :

$$\frac{\partial^2 T}{\partial x^2}(x,t) \approx \frac{\partial^2 T}{\partial x^2}(x_i, t_{k+1})$$

et la dérivée partielle par rapport à t est évaluée au point d'abscisse x_i et à l'instant k :

$$\frac{\partial T}{\partial t}(x,t) \approx \frac{\partial T}{\partial t}(x_i, t_k)$$

Question 14. Montrer que l'équation obtenue de la diffusion thermique peut être mise sous la forme

$$T_i^k = -cT_{i-1}^{k+1} + (1+2c)T_i^{k+1} - cT_{i+1}^{k+1}\,, \qquad \text{avec} \quad c = \frac{\Delta t\, D}{\Delta x^2}$$

.

Cette équation est appelée schéma implicite car la température à l'instant est exprimée en fonction de la température à l'instant ultérieur

Question 15. Mettre ce schéma sous forme matricielle.

Question 16. Effectuez l'inversion du système obtenu à chaque itération avec la méthode du pivot de Gauss partiel.

Question 17. Le pivot de Gauss est en $\mathcal{O}(n^3)$. Pour parer au problème de la complexité, l'algorithme de Thomas existe pour les matrices tridiagonales. Lire :

`https://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm`

et implémenter l'algorithme proposé. Conclure.