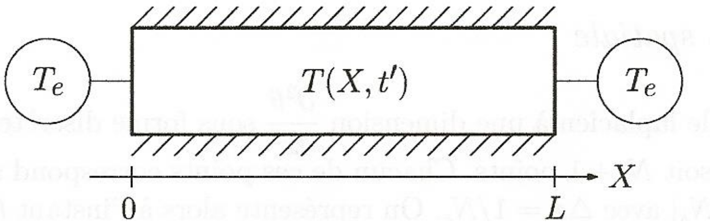
TD Python : Résolution de l'équation de diffusion thermique par la méthode d'Euler explicite

<u>Capacité numérique</u> : à l'aide d'un langage de programmation, résoudre l'équation de la diffusion thermique à une dimension par une méthode des différences finies dérivée de la méthode d'Euler explicite de résolution des équations différentielles ordinaires.

Objectifs du TP
— Savoir adimensionner une équation physique (CE de MPSI)
— Comprendre le passage du problème continu au problème discret
— Cerner les limites de l'algorithme utilisé

I. Problème 1D

On étudie l'établissement du régime stationnaire dans une tige homogène en cuivre de longueur L = 1m, de capacité thermique c = 0,38kJ.kg^{-1}.K^{-1}, de conductivité thermique λ(Cu) = 400W.m^{-1}K^{-1} et de masse volumique ρ = 8,9.10^{3}kg.m^{-3}.
Les parois latérales de la tige sont calorifugées.
A t' = 0 la température de la tige est uniforme de température T_e = 25°C. A partir de t'= 0 on impose à l'extrémité X = 0 une température T_1 = 0°C alors que l'extrémité X = L est maintenue à la température T_e.



On rappelle que l'application du premier principe de la Thermodynamique sur une tranche de barreau d'épaisseur dX conduit à l'équation de diffusion : $\frac{\partial T}{\partial t'} = K\frac{\partial^2 T}{\partial X^2}$ où K est la diffusivité thermique.

1) Adimensionnement du problème :

On introduit l'écart de température au thermostat, adimensionné : $\theta(X, t') = \frac{T(X,t')-T_e}{T_1-T_e}$
Montrer que l'équation de diffusion peut s'écrire sous la forme adimensionnée suivante :

$$\frac{\partial\theta(x,t)}{\partial t} = \frac{\partial^2\theta(x,t)}{\partial x^2}$$

en posant x =X/L (position adimensionnée) et $t = t'/\tau$ (temps adimensionné), où τ est un temps caractéristique du système que l'on exprimera en fonction de L et K.
Calculer numériquement la diffusivité thermique K de la barre et le temps τ.

2) Discrétisation du problème :

- Discrétisation spatiale et temporelle :
  - On découpe la longueur adimensionnée x=X/L=1 de la barre en N_x intervalles de longueur $\Delta x=1/N_x$.
  - On discrétise le temps adimensionné en N_t intervalles de durées $\Delta t=1/N_t$.

A chaque temps $t_i = i.\Delta t$ on va calculer les températures le long de la barre en chaque point $x_n = n.\Delta x$: $\theta_n^i = \theta(x_n,i.\Delta t)$.

- De l'équation différentielle à l'équation aux différences finies :

En utilisant des développements de Taylor-Young de $\theta(x, t)$ au voisinage de x_{n+1} et x_{n-1}, montrer que l'on a la relation : $\frac{\partial^2\theta}{\partial x^2}(x_n,t) \approx \frac{1}{(\Delta x)^2}[\theta(x_{n+1},t) + \theta(x_{n-1},t) - 2\theta(x_n,t)]$
Justifier le nom de la méthode numérique associée : *méthode de somme des différences finies centrées*.

De même montrer que $\frac{\partial \theta}{\partial t}(x_n, t) \approx \frac{1}{\Delta t}[\theta(x_n, t+\Delta t) - \theta(x_n, t)]$

En déduire que l'équation de diffusion thermique peut être discrétisée sous la forme :

$$\theta_n^{i+1} = \theta_n^{i} + (\theta_{n+1}^{i} + \theta_{n-1}^{i} - 2\theta_n^{i}).\frac{\Delta t}{\Delta x^2}$$

3) Programme Python :

Importer **matplotlib.pyplot** renommé **plt** pour les représentations graphiques.
Importer **numpy** renommé np.

On choisit N_x=100 et N_t=100000.
Créer deux tableaux des températures : TT0 pour les températures adimensionnées à t, et TT pour les températures adimensionnées à t+Δt. Initialiser ces températures (créer donc des tableaux de Nx+1 zéros et affecter la valeur +1 en x=0).

Ecrire un programme Python qui calcule les températures en tout point de la barre jusqu'au temps t_{max}'=τ/10 (donc t_{max} =1/10) *(par exemple avec une boucle while).*

Tracer la vraie température $T=T_e+(T_1-T_e)$*TT en fonction de la position au temps final t_{max}'=τ/10.

Modifier le programme pour qu'il trace la température pour t'_{max} = τ/10, τ/5 et τ/2 sur le même graphe.

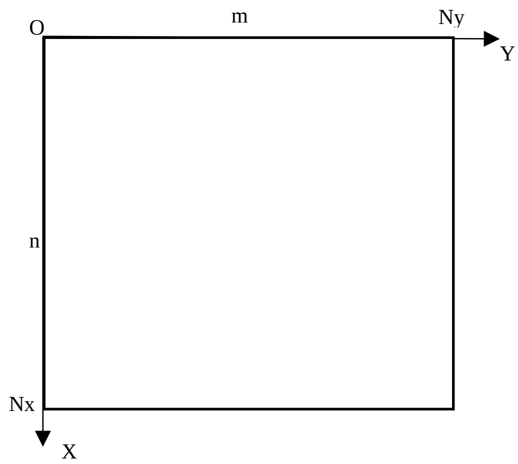
4) Convergence de la méthode :

- Convergence de la méthode : cette méthode ne converge que si $\left[\frac{\Delta t}{\Delta x^2}\right] \leq \frac{1}{2}$.
- Observer la divergence de la méthode pour un pas temporel trop grand ou un pas spatial trop petit. (*On pourra installer un « break » dans la boucle while si la valeur maximale de TT dépasse 1 en valeur absolue).*
  Par exemple on testera pour N_x=100, la valeur limite de N_t.

II. Problème 2D

1) Modélisation 2D

a) Discrétisation de l'espace 2D



On étudie une plaque métallique en cuivre rectangulaire de 1m de large et 1m de long.
A t'=0 la température de la plaque est uniforme de température T_e=25°C. A partir de t'=0 on impose sur le côté X=0 et sur le côté Y=0 une température T_1=0°C alors que les côtés X=L et Y=L sont maintenus à la température T_e.

Le problème étudié est donc un problème 2D dans le plan Oxy.

On introduira à nouveau les variables adimensionnées x=X/L, y=Y/L, t=t'/τ et θ.

L'espace sera décrit par un maillage de dimensions N_x=20, N_y=20 donc $\Delta x=1/N_x$ et $\Delta y=1/N_y$.
Pour parcourir l'axe des x on fera varier l'indice n des lignes du tableau (noté TT) des températures adimensionnées θ.
Pour parcourir l'axe des y on fera varier l'indice m des colonnes du tableau TT.

b) Discrétisation de l'équation de diffusion 2D

Montrer par la formule de Taylor à l'ordre 2 que :

$$\frac{\partial^2\theta(x,y)}{\partial x^2} \approx \frac{\theta(x+\Delta x,y)+\theta(x-\Delta x,y)-2\theta(x,y)}{(\Delta x)^2}$$

$$\frac{\partial^2\theta(x,y)}{\partial y^2} \approx \frac{\theta(x,y+\Delta y)+\theta(x,y-\Delta y)-2\theta(x,y)}{(\Delta y)^2}$$

En déduire que l'équation de la chaleur peut être discrétisée sous la forme :

$$\theta_{n,m}^{i+1} = \theta_{n,m}^{i} + (\theta_{n+1,m}^{i} + \theta_{n-1,m}^{i} - 2\theta_{n,m}^{i}).\frac{\Delta t}{\Delta x^2} + (\theta_{n,m+1}^{i} + \theta_{n,m-1}^{i} - 2\theta_{n,m}^{i}).\frac{\Delta t}{\Delta y^2}$$

c) Conditions initiales et aux limites spatiales

Il est nécessaire de définir des conditions initiales (on choisira $T=T_e$ partout donc $\theta=0$) et des conditions aux limites spatiales du domaine.
Les conditions aux limites sont essentiellement de deux types :
-DIRICHLET : C'est la valeur de la fonction elle-même (la température ici) qui est connue en tout point des frontières du domaine de calcul.
- NEUMANN : Ce sont les valeurs des dérivées partielles (variations) de la fonction (ce qui correspondrait au flux thermique surfacique ici) qui sont connues en tout point des frontières du domaine de calcul.

On choisit ici :
Les conditions initiales :

$\theta(n,m) = 0$ pour tout n et tout m à $t = 0$

Les conditions aux limites :

$\theta(n=0,m) = +1$; $\theta(n=Nx-1,m) = 0$ pour tout m quel que soit t

et $\theta(n,m=0) = +1$; $\theta(n,m=Ny-1) = 0$ pour tout n quel que soit t

2) Programme Python

Importer **matplotlib.pyplot** renommée **plt** pour les représentations graphiques.
Importer **numpy** renommé **np**.

a) Initialisation des tableaux

Créer deux tableaux TT et TT0 de dimensions (Nx,Ny) qui représentent les températures adimensionnées. Initialiser toutes les valeurs à 0.
Introduire les conditions aux limites dans les tableaux TT et TT0.

b) Itérations

- On choisit $N_t=100000$ et $N_x=20$, $N_y=20$. Vérifier que ces valeurs vérifient le critère de convergence.
- Balayage de « l'intérieur » du tableau avec affectation de la nouvelle valeur de TT[n,m] en utilisant les TT0[n,m] calculés à l'itération précédentes
- Copie de TT dans TT0
- Poursuivre l'itération tant que $t < t_{max}=1/100$

c) Tracés

Tracé du tableau T des vraies températures $T=T_e+(T_1-T_e)*TT$:

```
plt.imshow(T)
plt.colorbar()
plt.show()
```