Devoir Surveillé 02

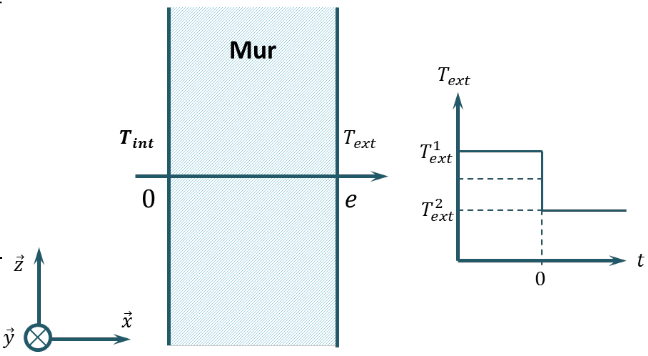
Le 02 février 2018– 2h00

Informatique

Vous trouverez à la fin de ce sujet une annexe contenant des informations concernant les programmes à écrire en `python`

1 Mise en situation

On étudie les transferts thermiques dans le mur d'une maison. La température à l'intérieur de la maison est constante dans le temps et égale à $T_{\text{int}} = 20°\text{C}$. Aux temps négatifs $t < 0$, la température extérieure est égale à $T_{\text{ext},1} = 10°\text{C}$. À $t = 0$, elle chute brusquement à $T_{\text{ext},2} = -10°C$ et elle reste égale à cette valeur aux temps positifs ($t > 0$). On souhaite étudier l'évolution du profil de température dans le mur au cours du temps.
Le mur a une épaisseur $e = 40$ cm. Les propriétés physiques du mur sont constantes : conductivité thermique $\lambda = 1,65$ W.m^{-1}.K^{-1}, capacité thermique massique : $c_p = 1\,000$ J.kg^{-1}.K^{-1}, masse volumique : $\rho = 2\,150$ kg.m^{-3}.



On suppose que les longueurs L_y et L_z suivant $\overrightarrow{y}$ et $\overrightarrow{z}$ sont très grandes devant l'épaisseur e. En conséquence, on suppose que la température T dans le mur ne dépend que du temps t et de la coordonnée x. La coordonnée x variera de 0 à e.

Objectif L'objectif est de déterminer l'évolution du flux thermique dans le mur au cours du temps. Pour cela, on s'appuiera sur la résolution d'une équation différentielle en utilisant un schéma explicite puis implicite.

1.1 Équation gouvernant la température

En l'absence de source d'énergie, l'équation régissant le transport de la chaleur s'exprime ainsi :

$$\rho \mathbf{c_p} \frac{\partial \mathbf{T}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{t})}{\partial \mathbf{t}} = \lambda \mathbf{\Delta T}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{t}) \tag{1}$$

On rappelle la définition du Laplacien d'une fonction suffisamment régulière $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$.

Question 1 *En utilisant les hypothèses dimensionnelles, donner l'équation de la chaleur simplifiée.*

1.2 Conditions aux limites

On envisage plusieurs types de conditions aux limites :
- **cas 1 :** la température est imposée aux limites du système;
- **cas 2 :** la paroi extérieure est isolée par un matériau de très faible conductivité.

Question 2 *Traduire chacune de ces conditions aux limites sur la fonction $T(x,t)$ et/ou sa dérivée.*

Dans la suite, seul le premier cas sera étudié.

Question 3 *Résoudre l'équation de la chaleur simplifiée* ***en régime permanent*** *dans les conditions suivantes :*
- ***conditions 1 :*** *pour un instant particulier négatif $t_1 < 0$;*
- ***conditions 2 :*** *pour un instant particulier positif $t_2 > 0$, très longtemps après la variation de température extérieure quand le régime permanent est de nouveau établi dans le mur.*

Question 4 *Quelle est la nature des profils $T(x)$ obtenus (en régime permanent) à ces deux instants ? Tracer à la main les deux profils sur un même graphique sur la copie.*

2 Résolution numérique : méthode des différences finies

On cherche à résoudre numériquement l'équation aux dérivées partielles :

$$\alpha \frac{\partial \mathbf{T(x,t)}}{\partial \mathbf{t}} = \frac{\partial^2 \mathbf{T(x,t)}}{\partial \mathbf{x^2}} \quad \text{avec } \alpha \text{ constante.} \tag{2}$$

Les conditions aux limites sont les suivantes :

- $T(0,t) = T_{\text{int}}$ pour $t > 0$;
- $T(e,t) = T_{\text{ext},2}$ pour $t > 0$;
- $T(x,0) = ax + b$ pour $x \in [0,e]$.

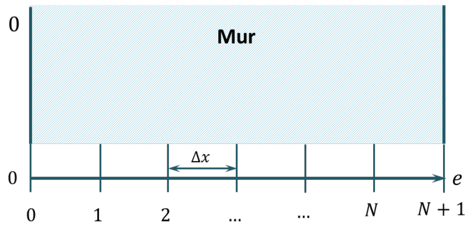
Question 5 *Quelle est l'expression de α en fonction des paramètres physiques du mur ?*

Question 6 *Exprimer a et b en fonction de $T_{int}, T_{ext,1}$ et e.*

La partie suivante permettra de déterminer une solution de l'équation aux dérivées partielles en utilisant la méthode des différences finies.

2.1 Discrétisation dans l'espace et dans le temps

On divise l'intervalle $[0,e]$, représentant l'épaisseur du mur, en $N+2$ points, numérotés de 0 à $N+1$, régulièrement espacés de Δx. Cette division est appelée « discrétisation ». La distance Δx est appelée le « pas d'espace ». A l'intérieur du mur (frontières intérieure et extérieure exclues) se trouvent donc N points. On cherche à obtenir la température en ces points particuliers à chaque instant.



Question 7 *Donner l'expression de Δx en fonction de N et de l'épaisseur du mur e.*

Question 8 *Donner l'abscisse x_i du i^e point en fonction de i et Δx, sachant que $x_0 = 0$ et $x_{N+1} = e$.*

Le temps est discrétisé en *ItMax* intervalles de durée Δt et on ne s'intéresse au profil de température qu'aux instants particuliers $t_k = k \cdot \Delta t$. L'intervalle élémentaire de temps Δt est appelé le « pas de temps ».
Deux méthodes de résolutions sont proposées :

- méthode utilisant un schéma explicite ;
- méthode utilisant un schéma implicite.

2.2 Méthode utilisant un schéma explicite

Objectif Déterminer le schéma explicite permettant la résolution de l'équation de la chaleur.

On donne le développement limité à l'ordre 3 de $T(x+\Delta x, t)$ et $T(x-\Delta x, t)$:

$$T(x+\Delta x,t) = T(x,t) + \frac{\partial T(x,t)}{\partial x}\Delta x + \frac{1}{2}\frac{\partial^2 T(x,t)}{\partial x^2}(\Delta x)^2 + \frac{1}{6}\frac{\partial^3 T(x,t)}{\partial x^3}(\Delta x)^3 + o\left(\Delta x^3\right)$$

$$T(x-\Delta x,t) = T(x,t) - \frac{\partial T(x,t)}{\partial x}\Delta x + \frac{1}{2}\frac{\partial^2 T(x,t)}{\partial x^2}(\Delta x)^2 - \frac{1}{6}\frac{\partial^3 T(x,t)}{\partial x^3}(\Delta x)^3 + o\left(\Delta x^3\right)$$

Question 9 *En déduire une expression approchée à l'ordre 1 de $\left[\frac{\partial^2 T(x,t)}{\partial x^2}\right]_{x,t}$ (dérivée partielle spatiale seconde de T évaluée au point x à l'instant t) en fonction de $T(x+\Delta x,t)$, $T(x-\Delta x,t)$ et $T(x,t)$ et Δx.*

On note T_i^k la température $T(x_i, t_k)$, évaluée au point d'abscisse x_i à l'instant t_k. De même, on note $T_{i+1}^k = T(x_i + \Delta x, t_k)$ et $T_{i-1}^k = T(x_i - \Delta x, t_k)$.

Question 10 *Déduire de la question précédente que $\left[\frac{\partial^2 T(x,t)}{\partial x^2}\right]_{x_i,t_k} \approx \frac{T_{i+1}^k - 2T_i^k + T_{i-1}^k}{\Delta x^2}$ (dérivée partielle seconde de T évaluée en x_i à l'instant t_k).*

La dérivée partielle temporelle de l'équation différentielle est maintenant approchée grâce à un développement limité.

En utilisant le même raisonnement en réalisant un développement limité de la fonction $t \mapsto T(x,t)$ à l'ordre 1, on obtiendrait l'équation suivante valable en chaque point d'abscisse x_i et à chaque instant t_k :

$$\left[\frac{\partial T(x,t)}{\partial t}\right]_{x_i,t_k} = \frac{T_i^{k+1} - T_i^k}{\Delta t}.$$

Question 11 *En utilisant les questions précédentes, montrer que l'équation différentielle peut se mettre sous la forme :*

$$T_i^{k+1} = r\,T_{i-1}^k + (1-2r)\,T_i^k + r\,T_{i+1}^k.$$

r sera explicité en fonction de Δx, Δt et α.

Dans toute la suite on considère que l'on a défini r et que l'on peut l'utiliser directement.

L'équation précédente est appelée schéma numérique explicite. Si on connaît la température en tous les points x_1, x_2, ..., x_N à l'instant t_k on peut calculer grâce à elle la température en tous les points à l'instant t_{k+1}.

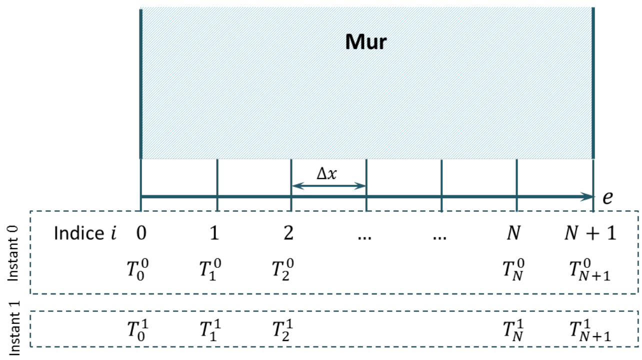
Question 12 *L'équation est-elle valable dans tout le domaine, c'est-à-dire pour toute valeur de i, $0 \le i \le N+1$? Que valent T_0^k et T_{N+1}^k ?*

Question 13 *Montrer que pour tout instant k, le problème peut se mettre sous la forme matricielle suivante :*

$$T^{k+1} = M \cdot T^k + r\,V \quad \text{avec} \quad T^k = \begin{pmatrix} T_1^k \\ T_2^k \\ \dots \\ T_{N-1}^k \\ T_N^k \end{pmatrix}.$$

avec M une matrice carrée $N \times N$, V un vecteur de taille N que l'on explicitera.

Ainsi, à chaque pas de temps k, on calculera un vecteur T^k contenant la température à chaque abscisse i du mur.



Question 14 *Expliciter succinctement comment déterminer la température dans le mur à chaque instant.*

Une annexe est fournie en fin de sujet vous rappelant quelques fonction de la bibliothèque **Numpy**.

Question 15 *On donne* `T0` *le vecteur température à l'instant $k=0$. Écrire la fonction* `euler_explicite(M,T0,V,k)` *retournant le vecteur de température T^k à l'instant k. Cette fonction sera définie de manière* ***récursive****.*

2.3 Méthode utilisant un schéma implicite

Objectif Déterminer une méthode permettant de résoudre l'équation de la chaleur à partir du schéma implicite donné.

En utilisant un schéma d'Euler implicite, on montre que l'équation $\alpha \dfrac{\partial T(x,t)}{\partial t} = \dfrac{\partial^2 T(x,t)}{\partial x^2}$ peut se mettre sous la forme suivante :

$$T_i^k = -r\,T_{i-1}^{k+1} + (1+2r)\,T_i^{k+1} - r\,T_{i+1}^{k+1}.$$

La température à l'instant t_k est exprimée en fonction de la température à l'instant ultérieur t_{k+1}. Le système d'équation peut être écrit sous la forme matricielle :

$$\mathbf{MT^{k+1}} = \mathbf{T^k} + \mathbf{rV}. \qquad (3)$$

avec :

$$M = \begin{pmatrix} 1+2r & -r & 0 & 0 & 0 & \dots & 0 \\ -r & 1+2r & -r & 0 & 0 & \dots & 0 \\ 0 & -r & 1+2r & -r & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & -r & 1+2r \end{pmatrix} \quad V = \begin{pmatrix} T_{\text{int}} \\ 0 \\ \vdots \\ 0 \\ T_{\text{ext}} \end{pmatrix} \quad T^k = \begin{pmatrix} T_1^k \\ T_2^k \\ \vdots \\ T_{N-1}^k \\ T_N^k \end{pmatrix}.$$

Question 16 *Pour obtenir T^{k+1} en fonction de T^k, il est nécessaire d'inverser le système matriciel à chaque pas de temps. Donner le nom d'un algorithme permettant de faire cela et donner sa complexité.*

Objectif La matrice M étant particulière (tridiagonale) , nous allons utiliser un algorithme plus performant pour résoudre ce système matriciel tridiagonal : l'algorithme de Thomas. Le système que l'on cherche à résoudre est le suivant :

$$MU = D$$

M est une matrice tridiagonale de dimension $N \times N$, c'est à dire que tous les éléments sont nuls sauf les diagonales principales, supérieures et inférieures.

On a donc :

$$M = \begin{pmatrix} b_0 & c_0 & & & & \\ a_1 & b_1 & c_1 & & 0 & \\ & a_2 & b_2 & c_2 & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & a_{N-2} & b_{N-2} & c_{N-2} \\ & & & & a_{N-1} & b_{N-1} \end{pmatrix} \quad U = \begin{pmatrix} u_0 \\ \vdots \\ \vdots \\ \vdots \\ u_{N-1} \end{pmatrix} \quad D = \begin{pmatrix} d_0 \\ \vdots \\ \vdots \\ \vdots \\ d_{N-1} \end{pmatrix}.$$

Dans cet algorithme, on calcule les coefficients suivants :

$$c'_0 = \frac{c_0}{b_0} \quad c'_i = \frac{c_i}{b_i - a_i c'_{i-1}} \quad \text{pour} \quad i = 1, 2, \dots, N-2.$$

$$d'_0 = \frac{d_0}{b_0} \quad d'_i = \frac{d_i - a_i d'_{i-1}}{b_i - a_i c'_{i-1}} \quad \text{pour} \quad i = 1, 2, \dots, N-1.$$

Les inconnues $u_0, u_1, \dots, u_{N-1}$ sont alors obtenues par les formules :

$$u_{N-1} = d'_{N-1} \quad u_i = d'_i - c'_i u_{i+1} \quad \text{pour} \quad i = N-2, N-3, \dots, 1, 0.$$

Question 17 *En utilisant l'algorithme de Thomas, écrire une fonction* `CalcTkp1(M,D)` *qui retourne le vecteur U, solution du système matriciel, à partir de la matrice M et du vecteur D.*

On pourra commencer par créer des vecteurs contenant des zéros : A, B, C, Cp ,Dp (pour C' et D') et U.
Puis on pourra définir A, B et C à partir de la matrice M et ensuite calculer Cp et Dp puis U.

Question 18 *Donner la complexité de l'algorithme et comparer à celui de la question 16. On prendra en compte uniquement les tests, les affectations (même de tableaux) et les opérations élémentaires (+, -, *, /) qui compte chacune pour « un ».*

3 Résolution de l'équation différentielle implicite

L'objectif est de calculer la température en chaque point au cours du temps. Parmi les variables d'entrée se trouvera un vecteur T_0 de dimension N, défini en dehors de la fonction, contenant les valeurs de la température aux points de discrétisation à l'instant initial. Au sein de la fonction, un algorithme calculera itérativement la température avec un nombre maximal d'itérations ItMax. En sortie de la fonction, on récupérera le nombre d'itérations réellement effectuées, `nbIter` et une matrice `T_tous_k`, de dimensions $N \times Itmax$. Chaque colonne de cette matrice contient le vecteur T^k dont les éléments sont les valeurs de la température aux N points x_1, ..., x_N, points à l'intérieur du mur à l'instant k :

$$T^k = \begin{pmatrix} T_1^k \\ T_2^k \\ \dots \\ T_{N-1}^k \\ T_N^k \end{pmatrix} \quad T\_tous\_k = \begin{pmatrix} T_1^1 & T_1^2 & \cdots & T_1^{ItMax-1} & T_1^{ItMax} \\ T_2^1 & T_2^2 & \cdots & T_2^{ItMax-1} & T_2^{ItMax} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ T_{N-1}^1 & T_{N-1}^2 & \cdots & T_{N-1}^{ItMax-1} & T_{N-1}^{ItMax} \\ T_N^1 & T_N^2 & \cdots & T_N^{ItMax-1} & T_N^{ItMax} \end{pmatrix}.$$

On souhaite arrêter le calcul lorsque la température ne varie presque plus dans le temps. Dans ce but, on évaluera la norme de $T^k - T^{k-1}$ à chaque itération.

Définition On définit la norme d'un vecteur v par :

$$||v|| = \sqrt{\sum_{i=1}^{N} v_i^2} \quad \text{avec} \quad v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ \vdots \\ v_n \end{pmatrix}.$$

Question 19 *Écrire une fonction* `calc_norme` *qui calcule la norme d'un vecteur.*

Question 20 *Écrire la fonction* `solution(M,T,V)`*, d'arguments une matrice M tridiagonale et deux vecteurs T et V et retournant le vecteur U tels que $MU = T + rV$*. On utilisera la fonction `CalcTkp1`.

Question 21 *Affecter la valeur 2000 à* `ItMax`*. Créer la matrice* `T_tous_k` *de dimensions $N \times ItMax$ en la remplissant de zéros.*

Question 22 *D'après les questions précédentes la température T dans le mur vérifie la condition initiale $T(x,0) = ax + b$ où $a = \dfrac{T_{ext,1} - T_{int}}{e}$ et $b = T_{int}$.*
Écrire la fonction `T0(Tint,Text,N)` *d'arguments la température intérieure T_{int}, la température extérieure T_{ext} et l'entier N et retournant le vecteur T^0, de taille N, contenant les températures en chaque point du mur lorsque $t = 0$.*

Question 23 *Écrire la suite d'instructions qui affecte à la première colonne de* `T_tous_k` *le vecteur des valeurs initiales T^0*. On utilisera la fonction `T0`.

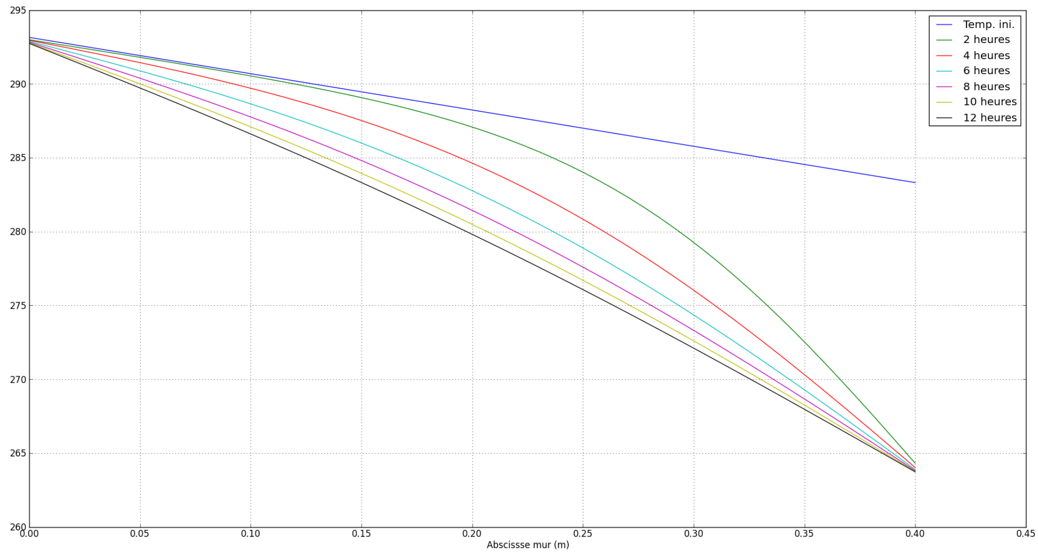
Question 24 *Écrire les instructions permettant de définir M et le vecteur V qui interviennent dans l'équation 3.*

Question 25 *Donner la suite d'instructions permettant de calculer le profil de la température à l'instant $k = 1$ ($t = \Delta t$). C'est à dire le vecteur T^1* . On utilisera la fonction `solution`.

Question 26 *Écrire la fonction* `euler_implicite(M,T0,V)` *d'argument une matrice M et deux vecteurs $T0$ et V et renvoyant la matrice* `T_tous_k`.
Cette boucle sera interrompue lorsque la norme du vecteur $T^k - T^{k-1}$ deviendra inférieure à 10^{-2} ou lorsque le nombre d'itérations atteindra la valeur ItMax (prévoir deux cas). Utiliser pour cela les fonctions `calc_norme` *et* `solution` *définies précédemment.*

4 Analyse des résultats

12 heures sont nécessaires pour atteindre le régime permanent. En utilisant le schéma implicite, on souhaite afficher la courbe de température en fonction de l'abscisse du mur. Le graphe attendu est le suivant :



Question 27 *Le pas de discrétisation temporel est de 30 secondes. Les résultats de la simulation sont stockés dans la matrice* `T_tous_k` *définie précédemment. Écrire les instructions permettant de tracer le réseau de courbes précédentes.*

Les résultats de la simulation sont codés dans un fichier texte codé en ASCII. L'écriture des nombres est limitée à 10 caractères (signe et virgule inclus). Le mur est discrétisé en 100 abscisses. Le pas de discrétisation temporel est de 30 secondes.

Question 28 *Quelle sera la taille du fichier texte généré ?*

5 Annexe

Dans tout le problème, on suppose que la bibliothèque **Numpy** a été chargée.
Tous les vecteurs et matrices intervenant seront des tableaux de numpy (array).
Par exemple, on peut définir des vecteurs par :
$U = array([1,2,3])$ et $V = array([1,0,1])$.
On a comme pour des listes : $U[0]=1, U[1]=2, V[1]=0$...

On peut définir une matrice par $M = array([[1,1,2],[1,-1,0],[0,1,3]])$ c'est la matrice $M = \begin{pmatrix} 1 & 1 & 2 \\ 1 & -1 & 0 \\ 0 & 1 & 3 \end{pmatrix}$.

De même , par exemple, on a $M[0,2]=2$.
On peut additionner deux vecteurs : $U+V = array([2,2,4])$, multiplier par un scalaire : $3*U = array([2,6,9])$.
Pour effectuer le produit scalaire on fait **vdot(u,v)**.
Pour effectuer le calcul de MU on fait **dot(M,U)**.
On peut aussi définir un vecteur de taille N contenant des zéros par l'instruction $A =$**zeros([N])** et une matrice de taille $i \times j$ contenant des zéros par $M =$**zeros([i,j])** .