Hackathon

Sujet 2 - Schéma différences finies

Context et modalités d'évaluation

L'objectif de ce sujet est d'étudier la résolution numérique de l'équation de la chaleur instationnaire.

Le problème à résoudre peut se formuler de la manière suivante :

$$\frac{\partial T}{\partial t} - \Delta(D(x,y)\cdot T(x,y)) = 0$$

$$T(x=0,y,t)=0,\forall t$$
$$T(x,y=0,t)=0,\forall t$$
$$T(x=L,y,t)=0,\forall t$$
$$T(x,y=L,t)=0,\forall t$$

$$T(x,y,0) = \frac{1}{2} + \sin(2\pi x) * \sin(2\pi y) - \frac{1}{2}\cos(2\pi x) * \cos(2\pi y)\ \forall x$$

La démarche pour résoudre ce problème est d'utiliser la méthode des différences finies. Le principe est le suivant :

1. On introduit une discrétisation en espace (grille régulière)

$x_0 = 0 < x_1 = \Delta x < x_2 = x_1 + \Delta x < ... < x_{i+1} = x_i + \Delta x < ... < x_{N-1} = 1. y_0 = 0 < y_1 = \Delta y < y_2 = y_1 + \Delta y < ... < y_{i+1} = y_i + \Delta y < ... < y_{N-1} = 1.$

2. A l'aide de cette discrétisation en espace nous pouvons approximer les dérivées spatiales de la manière suivante :

$$\left.\frac{\partial T}{\partial x}\right|_{x=x_i,y=y_j} = \frac{T_{i+1,j}-T_{i,j}}{\Delta x}\ ;\ \left.\frac{\partial^2 T}{\partial x^2}\right|_{x=x_i,y=y_j} = \frac{T_{i+1,j}-2\cdot T_{i,j}+T_{i-1,j}}{\Delta x^2}$$

$$\left.\frac{\partial T}{\partial y}\right|_{x=x_i,y=y_j} = \frac{T_{i,j+1}-T_{i,j}}{\Delta y}\ ;\ \left.\frac{\partial^2 T}{\partial y^2}\right|_{x=x_i,y=y_j} = \frac{T_{i,j+1}-2\cdot T_{i,j}+T_{i,j-1}}{\Delta y^2}$$

Où $T_{i,j} = T(x_i, y_j)$.

En introduisant alors cette discrétisation nous pouvons réécrire le problème initiale sous la forme matricielle suivante

$$\frac{\partial}{\partial t}\{T\} + [K]\cdot\{T\} = \{0\}$$

Avec $\{T\} \in \mathbb{R}^{N_x N_y}$ le vecteur global des température en chaque point de la discrétisation et $[K] \in \mathbb{R}^{N_x N_y \times N_x N_y}$ la matrice de conductivité thermique.

3. On introduit alors une discrétisation en temps :

$$t^0 = 0 < t^1 = \Delta t < t^2 = t^1 + \Delta t < ... < t^{k+1} = t^k + \Delta t < ... < t^{N_t - 1} = t_final$$

Nous pouvons alors discrétiser le terme de dérivée temporelle de la manière suivante en notant $\{T\}^{(k)} = \{T\}(t = t^k)$

$$\frac{1}{\Delta t}\left(\{T\}^{(k+1)} - \{T\}^{(k)}\right) = -[K]\cdot\{T\}^{(??)}$$

La question apparaissant alors est prend-on (k) ou $(k+1)$ pour le terme de droite ? Les deux sont possibles est suivant le choix deux méthodes différentes en découlent

Euler Explicite : on prend (k) ce qui nous donne la relation :

$$\{T\}^{(k+1)} = \{T\}^{(k)} - \Delta t\,[K] \cdot \{T\}^{(k)}$$

Euler Implicite : on prend $(k+1)$ ce qui nous donne la relation :

$$([1] + \Delta t\,[K])\,\{T\}^{(k+1)} = \{T\}^{(k)}$$

Pour le moment nous prendrons pour la suite du sujet les valeurs numériques suivantes :

$$D(x,y) = 1\ \forall (x,y),\ x \in [0,1],\ y \in [0,1],\ t \in [0,0.5]$$

Question 1 (Pour les C++ seulement):

Développer une classe *Matrix* permettant de représenter un objet de type matrice. Et implémenter les fonctionnalités nécessaires à la réalisation du projet à savoir :

1. Somme/différence de deux matrices
2. Produit matriciel
3. Produit d'une matrice par un scalaire

Il est fortement recommandé que vous mettiez en place quelques tests pour vérifier que vos opérations matricielles sont correctes.

Question 2 :

Mettre en place le code permettant de résoudre le problème de thermique instationnaire en utilisant le schéma Euler explicite. Attention au pas de temps que vous prenez !

Exporter les résultats dans un fichier texte et proposer un script Python permettant de tracer l'évolution de la température en fonction de x et t.

Question 3 :

Pour la résolution du problème de thermique vous allez avoir besoin de résoudre un système linéaire. Il vous faut donc implémenter une méthode de résolution d'un système linéaire. Vous pouvez choisir la méthode que vous souhaitez mais si vous n'avez pas de préférence je vous conseille d'implémenter la méthode du gradient conjugué.

Là encore il est fortement conseillé de mettre en place des tests pour vérifier le bon fonctionnement de la méthode.

Question 4 :

Mettre en place le code permettant de résoudre le problème de thermique instationnaire en utilisant le schéma **Euler Implicite**.

Exporter les résultats dans un fichier texte et proposer un script Python permettant de tracer l'évolution de la température en fonction de x et t.

Question 5 :

Considérons maintenant un milieu hétérogène, c'est-à-dire un $D(x)$ qui n'est plus constant. Pour cela générer un vecteur de D_i aléatoires dont les valeurs sont comprises entre 0.5 et 1.5. Refaire les résolutions des questions 2 et 4 (si elles ont bien été faites dès le début c'est instantanné ;)). **Attention** avec un D hétérogène la matrice K n'est plus symmétrique donc la résolution ne peut plus se faire à l'aide d'un gradient conjugué ...

Question 6 :

Procéder à des mesures de performance de votre code en fonction de la discrétisation spatiale. Comment évolue le temps de calcul ?

Pour réduire ce temps de résolution une piste envisageable est de tirer parti de la stucture de la matrice K. En effet cette dernière est pleine de zéro. Pour optimiser cela nous devons utiliser une matrice creuse. Proposer une nouvelle implémentation de la classe Matrix en faisant du stockage creux et les opérations associées. Pour les plus malins vous avez le droit pour cette question de vous accrocher à une librairie externe ;)

Question 3 (pour les Python seulement):

En utilisant une librairie graphique (celle que vous voulez) faire la visualisation en direct de l'évolution du champ de température en fonction de l'espace et du temps.