



## Analyse numérique (XMS1ME090) – 2026-2027 – TP2

### Méthode des différences finies

### Equation de la chaleur

On considère le problème aux limites suivant :

$$\begin{cases} \frac{\partial u}{\partial t}(x, t) - \frac{\partial^2 u}{\partial x^2}(x, t) = f(x, t), \quad \forall x \in ]0, 1[, \quad \forall t \in ]0, T[ \\ u(0, t) = u_g, \quad \forall t \in [0, T] \\ u(1, t) = u_d, \quad \forall t \in [0, T] \\ u(x, 0) = u_0(x), \quad \forall x \in [0, 1] \end{cases}$$

qui modélise la diffusion thermique dans une tige métallique de longueur  $L = 1$ , sur laquelle est appliquée une source de chaleur représentée par la fonction  $f(x, t)$ . L'inconnue  $u(x, t)$  représente la température en  $x$  à l'instant  $t$ . On suppose que la température initiale, à  $t = 0$ , est donnée par  $u_0(x)$  sur  $[0, 1]$ , et que la température est maintenue constante aux deux extrémités,  $u_g$  en  $x = 0$ , et  $u_d$  en  $x = 1$ , valeurs compatibles avec  $u_0(x)$ , c'est-à-dire telles que  $u_0(0) = u_g$  et  $u_0(1) = u_d$ .

Dans ce TP, on souhaite tester deux schémas aux différences finies pour approcher une solution de ce problème pour les données suivantes :

$$\begin{cases} f(x, t) = 0, \\ u_0(x) = \sin(\pi x) + \frac{1}{4} \sin(10\pi x), \\ u_g = 0, \quad u_d = 0. \end{cases}$$

La solution exacte correspondante vaut :

$$u(x, t) = \sin(\pi x) \exp(-\pi^2 t) + \frac{1}{4} \sin(10\pi x) \exp(-100\pi^2 t)$$

On considère un maillage espace-temps de  $[0, 1] \times [0, T]$ , de pas  $h = \frac{1}{N+1}$  en espace avec les points discrets  $x_i = ih$ ,  $i = 0, \dots, N+1$ , et de pas  $\Delta t = \frac{T}{M+1}$  en temps avec les instants

discrets  $t_j = j\Delta t$ ,  $j = 0, \dots, M + 1$ .

Pour  $j = 0, \dots, M + 1$ , on note  $U^{(j)}$  le vecteur de  $\mathbb{R}^N$  des valeurs  $u_i^{(j)}$ ,  $i = 1, \dots, N$  qu'on va déterminer pour approcher les valeurs de la solution exacte  $u(x_i, t_j)$ .

Le premier schéma aux différences finies qu'on étudie est le schéma explicite :

$$U^{(j+1)} = U^{(j)} - \Delta t A_h U^{(j)} + \Delta t C^{(j)}, \quad j = 0, \dots, M \quad (1)$$

et le second schéma est le schéma implicite :

$$U^{(j)} = U^{(j-1)} - \Delta t A_h U^{(j)} + \Delta t C^{(j)}, \quad j = 1, \dots, M + 1 \quad (2)$$

où  $A_h$  et  $C^{(j)}$  désignent la matrice carrée de dimension  $N$  et le vecteur de longueur  $N$  suivants :

$$A_h = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & 0 \\ & -1 & 2 & -1 & \\ & & \ddots & \ddots & \ddots \\ 0 & & & \ddots & \ddots & -1 \\ & & & & -1 & 2 \end{pmatrix} \quad \text{et} \quad C^{(j)} = \begin{pmatrix} f(x_1, t_j) + \frac{u_g}{h^2} \\ f(x_2, t_j) \\ \vdots \\ f(x_{N-1}, t_j) \\ f(x_N, t_j) + \frac{u_d}{h^2} \end{pmatrix}$$

**Nous vous conseillons de ne pas utiliser `np.matrix()` et d'utiliser `np.linspace()` pour le vecteur de discrétisation spatiale et `np.dot()` pour le produit matrice / vecteur. La seule utilisation de `linalg` que nous vous conseillons c'est le `la.solve()`.**

1. Vérifier que dans le cas particulier étudié dans le TP on a  $C^j$  qui vaut le vecteur nul pour tout  $j$ .
2. Ecrire un script Python qui effectue les étapes suivantes :
  - (a) Calcul de la subdivision uniforme de  $N + 1$  intervalles de  $[0, 1]$ , de pas  $h = \frac{1}{N+1}$ ,
  - (b) Choix du pas  $dt$  et du temps final  $T$ ,
  - (c) Calcul de la matrice  $A_h$ ,
  - (d) Calcul itératif des vecteurs  $U^{(j)}$  par le schéma choisi. Pour représenter les vecteurs  $U^{(j)}$  aux instants successifs  $t_j$ , on utilisera deux vecteurs, un pour  $U^{(j)}$  et un pour  $U^{(j+1)}$  qu'on mettra à jour à chaque itération.
  - (e) Calcul de l'erreur max :

$$erreur = \max_{i,j} |u(x_i, t_j) - u_h(x_i, t_j)|$$

- (f) Tracé du graphe de la fonction  $u_h(x, T)$  pour  $x \in [0, 1]$  à  $t = T$ .

3. Valider votre code à l'aide des informations données ci-dessous. Dans le cas  $N = 10$ ,  $dt = 0.0001$  et la méthode **explicite** voici les vecteurs  $U^{(0)}$ ,  $U^{(1)}$  et  $U^{(2)}$  (sans les termes de bord) ainsi que les erreurs à l'itération  $j$  à obtenir (attention tous les chiffres significatifs ne sont pas donnés!) :

```

1 U^(0) =
2 [0.3521657 0.40548061 0.94468697 0.682224 1.2372768 0.74236608
3 1.13703999 0.56681218 0.67580102 0.21129942]
4 U^(1) =
5 [0.3485496 0.4113599 0.93498677 0.69211594 1.22457224 0.75313006
6 1.12536468 0.5750307 0.66886179 0.21436316]
7 err^(1) = 0.01152821
8 U^(2) =
9 [0.34509216 0.41693578 0.92571215 0.7014974 1.21242507 0.76333854
10 1.1142016 0.5828251 0.662227 0.2172688 ]
11 err^(2) = 0.02142717
12 U^(3) =
13 [0.34178585 0.42222267 0.91684295 0.71039262 1.2008089 0.77301793
14 1.1035265 0.59021552 0.65588224 0.22002384]
15 err^(3) = 0.02987590

```

Si le temps final correspond à  $3 \times dt$  alors l'erreur maximale à calculer est évidemment : 0.02987590. Avec les mêmes valeurs mais le schéma **implicite**, on obtient :

```

1 U^(0) =
2 [0.3521657 0.40548061 0.94468697 0.682224 1.2372768 0.74236608
3 1.13703999 0.56681218 0.67580102 0.21129942]
4 U^(1) =
5 [0.34870108 0.41107025 0.93539311 0.6916286 1.22510444 0.75259976
6 1.12585376 0.57462581 0.66915247 0.21421223]
7 err^(1) = 0.012060404
8 U^(2) =
9 [0.3453811 0.41638338 0.92648723 0.70056798 1.21344022 0.7623272
10 1.11513451 0.58205291 0.66278147 0.21698094]
11 err^(2) = 0.022442312
12 U^(3) =
13 [0.34219921 0.42143254 0.9179518 0.70906323 1.20226117 0.77157135
14 1.10486112 0.58911102 0.65667547 0.2196121 ]
15 err^(3) = 0.031328176

```

Évidemment le vecteur  $U^{(0)}$  est le même puisque c'est la condition initiale. Dans ce cas l'erreur maximale serait 0.031328176 si le temps final correspond à  $3 \times dt$ .

4. Exécuter ce script pour faire des tests numériques en utilisant la feuille de résultats.