M1 IM/MPA : TP EDP et Différences Finies.

TP2 2 : Schémas itératifs - Euler et Crank-Nicolson pour les EDOs et l'équation de la chaleur

I EDO

La première partie de ce TP porte sur l'implémentation des schémas itératifs d'Euler explicite/implicite et de Crank-Nicolson pour la résolution numérique d'EDO. Nous considérerons ici le cas de l'EDO autonome linéaire du TP0 :

$$\begin{cases} y'(t) = -ky(t) + r, \\ y(t_0) = y_0. \end{cases} \tag{1}$$

sur l'intervalle de temps $[t_0, t_f]$, avec $t_0 \in \mathbb{R}$ et $t_f > t_0$, $k > 0$, $r \in \mathbb{R}$, $y_0 \in \mathbb{R}$.

On cherche à obtenir une approximation $(y_n)_{0\leq n\leq N}$ des valeurs $y(t_n)_{0\leq n\leq N}$ prises par la solution exacte y du problème (1) aux instants $t_i = t_0 + i\Delta_t$, où Δ_t est le *pas de temps* $\Delta_t = (t_f - t_0)/N$. On notera h ou Δ_x pour le pas d'espace.

I.1 Résolution numérique

1. *Programmation de la méthode d'Euler explicite* .

a) (Voir TP0) Écrire (sur feuille) le schéma d'Euler explicite pour cette équation (relation satisfaite par y_{n+1} et y_n) puis créer une fonction *EulerExplicite* qui prend comme arguments k, r, t_0, t_f, y_0 et N et qui renvoie le vecteur $(t_n)_{0\leq n\leq N}$ et le vecteur $(y_n)_{0\leq n\leq N}$ des valeurs approchées de la solution y de (1) aux temps $(t_n)_{n\in[0,N]}$ par la méthode d'Euler explicite.

b) On fixe $k = 150$, $r = 0$, $y_0 = 1$ et $[t_0, t_f] = [0, 1]$. La solution exacte de (1) est donc $y(t) = \exp(-150t)$. Tracer sur un même graphe la solution exacte et l'approximation obtenue avec Euler explicite : on testera les quatre valeurs de N telles que $\Delta_t = \frac{1}{50}, \frac{1}{70}, \frac{1}{90}$ et $\frac{1}{160}$.

c) Tracer, en échelle *log-log*, le graphe de $\max_{0\leq n\leq N}(|y_n - y(t_n)|)$ (la norme infinie de l'erreur) en fonction du pas Δ_t pour des valeurs de Δ_t faible (par exemple entre 1/1500 et 1/1000). En déduire l'ordre de convergence du schéma.

2. *Programmation de la méthode d'Euler implicite* .

Reprendre les questions **1.a)**, **1.b)** et **1.c)** en remplaçant Explicite par Implicite.

3. *Programmation de la méthode de Crank-Nicolson* .

Reprendre les questions **1.a)**, **1.b)** et **1.c)** en remplaçant Euler Explicite par Crank Nicolson à pas constant [1].

1. On rappelle que pour l'équation $y'(t) = f(t, y(t))$, ce schéma s'écrit

$$y_{n+1} = \frac{\Delta_t}{2}\left(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\right)$$

I.2 Analyse théorique

On considère l'équation (1) en posant, pour simplifier, $t_0 = 0$. Alors la solution exacte est

$$y(t) = (y_0 - r/k)e^{-kt} + r/k.$$

En particulier, elle converge vers r/k lorsque $t \to +\infty$.

a) On note $\delta = (y_0 - r/k)$. On peut montrer que $\forall n \geq 0,\ (y_n - r/k) = \delta(1 - \Delta_t k)^n$. En déduire une explication au phénomène observé à la question **1.b)** de la partie I.1.

b) De même, on montrerait que pour Euler Implicite on a

$$(y_n - r/k) = \delta/(1 + k\Delta_t)^n$$

et pour Crank-Nicolson

$$(y_n - r/k) = \delta \times \left(\frac{1 - \Delta_t k/2}{1 + \Delta_t k/2}\right)^n.$$

En déduire une explication au phénomène observé aux questions **2.b)** et **3.b)** de la partie I.1.

II Équation de la chaleur

La seconde partie de ce TP porte sur l'implémentation des schémas itératifs d'Euler explicite/implicite et de Crank-Nicolson pour la résolution numérique de l'équation de la chaleur en dimension 1. On considère l'équation

$$\partial_t u - \mu \partial_{xx} u = 0$$

pour $t \in I = [0, 0.5]$ et $x \in J = [0, L]$, avec conditions aux bords de type Dirichlet homogènes ($\forall t \in I,\ u(t, 0) = u(t, L) = 0$) et condition initiale $u(0, .) = u_0(.)$ où u_0 est une fonction donnée.

Cette équation modélise la diffusion de la chaleur, de distribution initiale u_0, dans une barre unidimensionelle de longueur L ayant un coefficient de conduction μ et dont les extrémités sont maintenues à température nulle.

Notations : On note (t_n, x_j), $0 \leq n \leq N$, $0 \leq j \leq M + 1$ les points de la discrétisation de $I \times J$ en une grille régulière de taille $(N + 1) \times (M + 2)$. On note Δt et Δx les pas de temps et d'espace. On note u_j^n l'approximation de $u(t_n, x_j)$ obtenue grâce à la méthode des différences finies. Compte tenu des conditions de bords et initiale, les inconnues sont les u_j^n pour $1 \leq n \leq N$ et $1 \leq j \leq M$.

a) Préparer (à la main) le schéma numérique obtenu en utilisant le schéma d'Euler explicite en temps et l'approximation centrée du laplacien en espace. On exprimera u_j^{n+1} en fonction de u_j^n, u_{j-1}^n et u_{j+1}^n. On pourra noter $\lambda = \mu \frac{\Delta t}{\Delta x^2}$.

.

b) Ré-écrire ce schéma sous la forme $U^{n+1} = AU^n$ où A est une matrice de taille M et

$$U^n = \begin{pmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_{M-1}^n \\ u_M^n \end{pmatrix}$$

Remarque : Vous avez vu/verrez que ce schéma est stable si et seulement si $\mu\Delta t/(\Delta x)^2 < 1/2$, c'est la condition de Courant, Friedrichs et Lewy (CFL).

c) Créer une fonction *ResolutionExp* qui prend en argument N, M et mu, tf et L. Cette fonction met en œuvre ce schéma sur l'intervalle de temps $I = [0, tf]$ et l'intervalle d'espace $J = [0, L]$ avec $u_0(x) := \sin(\pi x)$ (L sera supposé entier pour que les conditions de bords et initiale soient cohérentes) et renvoie la discrétisation T de I en $N+1$ points, la discrétisation X de J en $M+2$ points, et la matrice $U \in M_{M+2,N+1}(\mathbb{R})$ contenant les u_j^n (avec $u_0^n = u_{M+1}^n = 0$ et $u_j^0 = u_0(x_j)$ pour tout n, j).

d) Tracer la solution numérique obtenue avec les paramètres suivants :

— $tf = 0.5$, $L = 1$, $\mu = 1$, $N = 110$, $M = 10$.

Pour cela on pourrait utiliser la fonction *pcolormesh* (tracé 2D) du package matplotlib.pyplot (tracé en 2D), mais on lui préférera la fonction *plot_surface* (tracé en 3D), dont l'aide est disponible ici

`https://matplotlib.org/stable/gallery/mplot3d/surface3d.html`

Attention, sur les machines du PV, il pourra être nécessaire d'ajouter

```
from mpl_toolkits.mplot3d import Axes3D
```

e) Même question avec $N = 50$. Commenter.

f) Lorsque $\mu = 1$ et $L = 1$, la solution exacte est donnée par $u : (t, x) \mapsto \exp(-\pi^2 t)\sin(\pi x)$. Avec la fonction *plot_surface*, tracer la différence entre la solution solution exacte et son approximation obtenue pour $M = 100$, $N = 30000$, $tf = 0.5$.

g) Reprendre les questions **a)-f)** avec le schéma d'Euler implicite en temps et l'approximation centrée du laplacien en espace. Pour la question **b)** on a cette fois une relation du type $AU^{n+1} = U^n$ (on pourrait montrer que A est définie positive, donc inversible). Normalement votre code relatif aux questions **d)-f)** est directement réutilisable.

h) Reprendre les questions **a)-f)** avec le schéma de Crank-Nicolson. Cette fois, à la question **b)** on a une relation du type $A_2 U^{n+1} = A_1 U^n$. Normalement votre code relatif aux questions **d)-f)** reste réutilisable.