

TPs EDP

Travaux Pratiques N° 4

Méthodes des différences finies Equation de la chaleur 1D instationnaire

Version du 4 novembre 2024

1 Conditions de Dirichlet

On souhaite résoudre numériquement le problème aux limites (*boundary value problem* ou B.V.P.) suivant

$$\frac{\partial u}{\partial t}(t, x) - \nu \frac{\partial^2 u}{\partial x^2}(t, x) = f(t, x), \quad \forall (t, x) \in]t_0; t_0 + T[\times]a; b[, \quad (1.1)$$

$$u(t_0, x) = u_0(x), \quad \forall x \in [a; b], \quad (1.2)$$

$$u(t, a) = g_a(t), \quad \forall t \in [t_0; t_0 + T], \quad (1.3)$$

$$u(t, b) = g_b(t), \quad \forall t \in [t_0; t_0 + T]. \quad (1.4)$$

avec ν un réel strictement positif, $t_0 \in \mathbb{R}$, $T > 0$, $(a, b) \in \mathbb{R}^2$, $a < b$. Bien évidemment des conditions de compatibilité sont imposées entre les conditions aux limites et la condition initiale.

On note t^n , $n \in \llbracket 0, N_t \rrbracket$ et x_i , $i \in \llbracket 0, N_x \rrbracket$ les discrétisations régulières des intervalles $[t_0; t_0 + T]$ et $[a; b]$ avec N_t pas de discrétisation en temps et N_x pas de discrétisation en espace.

On souhaite implémenter deux schémas de résolution de (1.1) :

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \nu \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} = f_i^{n+1}. \quad (1.5)$$

et

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \nu \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} = f_i^n. \quad (1.6)$$

où $\Delta t = T/N_t$, $\Delta x = (b - a)/N_x$, $f_i^n = f(t^n, x_i)$ et $u_i^n \approx u(t^n, x_i)$.

On rappelle que le premier schéma est le **schéma d'Euler implicite** et le second le **schéma d'Euler explicite**. Le schéma d'Euler implicite est inconditionnellement stable et le schéma d'Euler explicite est stable sous la condition de C.F.L.

$$\nu \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}.$$

On note, $\forall n \in \llbracket 0, N_t \rrbracket$, $\mathbf{U}^n$ les vecteurs de dimension $N_x + 1$, de composantes $\mathbf{U}_i^n = u_{i-1}^n$, $\forall i \in \llbracket 1, N_x + 1 \rrbracket$.

Q. 1 [Maths]

Pour chaque schéma, écrire sur feuille et **de manière détaillée**, la façon dont il a été obtenu ainsi que la discrétisation du B.V.P. (1.1) à (1.4) et l'envoyer **par mail** à l'adresse cuvelier@math.univ-paris13.fr ou sur **Discord** en message privé.

2 Implémentation et validation

On étudie ce problème aux limites avec les données $t_0 = 0$, $T = 2$, $a = 0$, $b = 2\pi$, $\nu = 2$, $k = 5$

$$f(t, x) = -k \sin(kt) \cos(x) + \nu \cos(kt) \cos(x),$$

$$u_0(x) = \cos(kt_0) \cos(x),$$

$$g_a(t) = \cos(kt) \cos(a),$$

$$g_b(t) = \cos(kt) \cos(b).$$

Dans ce cas, la solution exacte est donnée par $u_{\text{ex}}(t, x) = \cos(kt) \cos(x)$.

Q. 2 [Maths]

Proposer un autre exemple non trivialement similaire avec une solution exacte (non identiquement nulle!).

Q. 3

- a. Pour résoudre le B.V.P. par un schéma d'**Euler implicite**, le programme `heat1d.implicit_DirichletCL` est fourni, ainsi que les fonctions `fsftx1d.eval*`, `fsftx1d.norm` et `fsftx1d.plot` (voir l'archive fournie : **TP4.tar.gz**). Dans les codes fournis, il manque le fichier `+head1d/+implicit/Euler_DirichletCL.m` correspondant à la fonction :

$$[t,x,u]=\text{heat1d.implicit.Euler_DirichletCL}(BVP,Nt,Nx)$$

résolvant le B.V.P. par un schéma d'Euler implicite avec

- BVP : structure, définie dans `heat1d.implicit_DirichletCL`, contenant l'ensemble des données du B.V.P. à résoudre.
- Nt : nombre de pas de discrétisation en temps,
- Nx : nombre de pas de discrétisation en espace,
- t : discrétisation en temps (dimension Nt+1), $t(n) = t^{n-1}$, $\forall n \in \llbracket 1, N_t + 1 \rrbracket$,
- x : discrétisation en espace (dimension Nx+1), $x(i) = x_{i-1}$, $\forall i \in \llbracket 1, N_x + 1 \rrbracket$,
- u : $u(i,n)$ solution approchée au temps $t(n)$ et point $x(i)$ (dimension (Nx+1,Nt+1)).

Ecrire cette fonction.

- b. Utiliser les commandes `tic` et `toc` pour comparer les temps d'exécution de la fonction `head1d.implicit.Euler_DirichletCL` entre un stockage plein ou creux de la matrice.
- c. Que pourrait-on faire, au niveau algorithmique, pour améliorer les performances de ce code ?
- d. Implémenter ces améliorations et vérifier s'il y a un gain de performances.

Q. 4

- a. Pour résoudre le B.V.P. par un schéma d'**Euler explicite**, le programme `heat1d.explicit_DirichletCL` est fourni, ainsi que les fonctions `fsftx1d.eval`, `fsftx1d.norm` et `fsftx1d.plot` (voir fichier **TP4.tar.gz**). Dans les codes fournis, il manque le fichier `+head1d/+explicit/Euler_DirichletCL.m` correspondant à la fonction :

$$[t,x,u]=\text{head1d.explicit.Euler_DirichletCL}(EDP,Nt,Nx)$$

résolvant le B.V.P. par un schéma d'Euler explicite. Ecrire cette fonction, sachant que ses paramètres sont identiques à ceux de la fonction `head1d.implicit.Euler_DirichletCL`.

- b. Dans le programme `heat1d.explicit_DirichletCL`, changer le paramètre Nt de 2100 à 2000. Que se passe-t-il lors de la résolution ? Quel est le résultat théorique qui intervient ?
- c. [Facultatif] Ecrire un programme permettant de mettre en lumière ce phénomène.

3 Application

Comme application possible nous allons résoudre numériquement par le schéma implicite d'Euler le problème de conduction thermique dans une barre de longueur $L = 6$:

$$\frac{\partial u}{\partial t}(t,x) - \nu \frac{\partial^2 u}{\partial x^2}(t,x) = 0, \quad \forall (t,x) \in]0;T] \times]0;L[, \quad (3.1)$$

$$u(t_0,x) = u_0(x), \quad \forall x \in [0;L], \quad (3.2)$$

$$u(t,0) = u_g(t), \quad \forall t \in [0;T], \quad (3.3)$$

$$u(t,L) = u_d(t), \quad \forall t \in [0;T]. \quad (3.4)$$

avec $T = 10$, $u_0(x) = 10, \forall x \in [0, L]$

$$u_g(t) = \begin{cases} 10 + 90t, & \forall t \in [0, 1] \\ 100, & \forall t > 1 \end{cases}$$

$$u_d(t) = \begin{cases} 10 + 70t, & \forall t \in [0, 1] \\ 80, & \forall t > 1 \end{cases}$$

Q. 5

- a. Ecrire le programme `heat1d.bar01` permettant de résoudre ce problème par le schéma d'Euler implicite.
- b. Exécuter ce programme pour différentes valeurs de ν (par exemple $\nu = 0.1$, $\nu = 1$ et $\nu = 3$). Qu'observe-t-on ?

*. `fsftx1d` abréviation de *functionnal space of functions in time and 1d in space*

4 Conditions mixtes

On souhaite résoudre numériquement le B.V.P. suivant

$$\frac{\partial u}{\partial t}(t, x) - \nu \frac{\partial^2 u}{\partial x^2}(t, x) = f(t, x), \quad \forall (t, x) \in]t_0; t_0 + T] \times]a; b[, \quad (4.1)$$

$$u(t_0, x) = u_0(x), \quad \forall x \in [a; b], \quad (4.2)$$

$$\delta_a u(t, a) + \mu_a \frac{\partial u}{\partial x}(t, a) = g_a(t), \quad \forall t \in [t_0; t_0 + T], \quad (4.3)$$

$$\delta_b u(t, b) + \mu_b \frac{\partial u}{\partial x}(t, b) = g_b(t), \quad \forall t \in [t_0; t_0 + T], \quad (4.4)$$

avec ν un réel strictement positif, $t_0 \in \mathbb{R}$, $T > 0$, $(a, b) \in \mathbb{R}^2$, $a < b$, $(\delta_a, \delta_b) \in \llbracket 0, 1 \rrbracket^2$. Pour s'assurer de l'existence et l'unicité d'une solution à ce problème, on prend

$$\begin{cases} \mu_a \leq 0 & \text{si } \delta_a = 1, \\ \mu_a < 0 & \text{si } \delta_a = 0, \end{cases} \quad \text{et} \quad \begin{cases} \mu_b \geq 0 & \text{si } \delta_b = 1, \\ \mu_b > 0 & \text{si } \delta_b = 0. \end{cases} \quad (4.5)$$

Q. 6

Reprendre les questions 1, 3 et 4 pour le problème (4.1)-(4.4). Lors de la programmation, les conditions aux limites seront intégrées à la structure BVP.

5 Schéma de Crank-Nicolson (facultatif)

Le schéma de Crank-Nicolson pour la discrétisation de l'équation de la chaleur (4.1) est donné par

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} - \frac{\nu}{2} \left(\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} \right) = \frac{1}{2} (f_i^n + f_i^{n+1}). \quad (5.1)$$

Ce schéma est d'ordre 2 en temps et en espace et il est inconditionnellement stable.

Q. 7 [facultatif]

A vous d'écrire un programme permettant de résoudre le problème (4.1)-(4.4) en utilisant le schéma (5.1). Il faudra bien évidemment proposer une méthodologie de tests/validations.

6 Problème de Cauchy

L'objectif ici est de montrer que le B.V.P. (4.1) à (4.4) peut s'écrire, après discrétisation en espace uniquement, sous la forme d'un problème de Cauchy :

Trouver $\mathbf{U} : [t_0, t_0 + T] \longrightarrow \mathbb{R}^m$ tq

$$\frac{d\mathbf{U}}{dt}(t) = \mathcal{F}(t, \mathbf{U}(t)), \quad \forall t \in [t_0, t_0 + T], \quad (6.1)$$

$$\mathbf{U}(t_0) = \mathbf{U}^0 \in \mathbb{R}^m \quad (6.2)$$

avec $\mathcal{F} : [t_0, t_0 + T] \times \mathbb{R}^m \longrightarrow \mathbb{R}^m$.

On note $x_i = a + ih$, $i \in \llbracket 0, N_x \rrbracket$ la discrétisation régulière en $(N_x + 1)$ points de $[a, b]$ et on pose

$$\alpha_0 \stackrel{\text{def}}{=} \delta_a - 3\frac{\mu_a}{2h}, \quad \alpha_1 \stackrel{\text{def}}{=} 4\frac{\mu_a}{2h}, \quad \alpha_2 \stackrel{\text{def}}{=} -\frac{\mu_a}{2h} \quad (6.3)$$

$$\beta_0 \stackrel{\text{def}}{=} \delta_b + 3\frac{\mu_b}{2h}, \quad \beta_1 \stackrel{\text{def}}{=} -4\frac{\mu_b}{2h}, \quad \beta_2 \stackrel{\text{def}}{=} \frac{\mu_b}{2h}. \quad (6.4)$$

Soient $m = N_x - 1$, $\mathbb{A} \in \mathcal{M}_m(\mathbb{R})$ la matrice définie par

$$\mathbb{A} \stackrel{\text{def}}{=} -\frac{\nu}{h^2} \begin{pmatrix} 2 + \frac{\alpha_1}{\alpha_0} & -1 + \frac{\alpha_2}{\alpha_0} & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 + \frac{\beta_2}{\beta_0} & 2 + \frac{\beta_1}{\beta_0} \end{pmatrix},$$

et $\mathbf{b}$ l'application définie par

$$\begin{aligned} \mathbf{b} &: [t_0, t_0 + T] \longrightarrow \mathbb{R}^m \\ t &\longmapsto \mathbf{b}(t) = \begin{pmatrix} \frac{\nu}{\alpha_0 h^2} g_a(t) + f(t, x_1) \\ f(t, x_2) \\ \vdots \\ f(t, x_{m-1}) \\ \frac{\nu}{\beta_0 h^2} g_b(t) + f(t, x_m) \end{pmatrix} \end{aligned}$$

Q. 8 [Maths]

Après discrétisation en espace uniquement, et, en utilisant des approximations des dérivées spatiales à l'ordre 2, montrer que le B.V.P. (4.1) à (4.4) s'écrit sous la forme d'un problème de Cauchy (6.1)-(6.2) avec $m = N_x - 1$, $\mathbf{U}^0 = (u_0(x_1), \dots, u_0(x_m))^t$ et

$$\begin{aligned} \mathcal{F} &: [t_0, t_0 + T] \times \mathbb{R}^m \longrightarrow \mathbb{R}^m \\ (t, \mathbf{Y}) &\longmapsto \mathbb{A}\mathbf{Y} + \mathbf{b}(t) \end{aligned} \quad (6.5)$$

La matrice $\mathbb{A}$ est une **matrice tridiagonale de Jacobi** et on admet le résultat suivant :

Proposition 6.1

La matrice $\mathbb{A} \in \mathcal{M}_m(\mathbb{R})$ définie en (6) admet m valeurs propres réelles, distinctes deux à deux. De plus, en notant $\lambda_i(\mathbb{A})$, $i \in \llbracket 1, m \rrbracket$ ces valeurs propres ordonnées par ordre décroissant, on a

$$-4\frac{\nu}{h^2} < \lambda_m(\mathbb{A}) < \dots < \lambda_1(\mathbb{A}) < 0. \quad (6.6)$$

avec

$$\lambda_m(\mathbb{A}) < \frac{\nu}{h^2} \left(-4 + \left(\frac{\pi}{m-1} \right)^2 \right) \quad \text{et} \quad -\frac{\nu}{h^2} \left(\frac{\pi}{m-1} \right)^2 < \lambda_1(\mathbb{A}) \quad (6.7)$$

6.1 Schéma d'Euler progressif

Le schéma d'Euler progressif (ou explicite) pour la résolution d'un problème de Cauchy s'écrit

$$\mathbf{Y}^{[n+1]} = \mathbf{Y}^{[n]} + h_t \mathcal{F}(t^n, \mathbf{Y}^{[n]}),$$

et il est d'ordre 1.

Q. 9 [Maths]

a. Montrer qu'avec $\mathcal{F}$ donné par (6.5), on a

$$\mathbf{Y}^{[n+1]} = \mathbb{B}\mathbf{Y}^{[n]} + \mathbf{c} \quad (6.8)$$

où l'on explicitera $\mathbb{B} \in \mathcal{M}_m(\mathbb{R})$ et $\mathbf{c} \in \mathbb{R}^m$.

b. Comparer ce schéma avec le schéma d'Euler explicite obtenu en Q. 1

c. En utilisant la Proposition 6.1, montrer que

$$\nu \frac{h_t}{h^2} < \frac{1}{2} \quad \Rightarrow \quad \rho(\mathbb{B}) < 1. \quad (6.9)$$

Q. 10

- Ecrire une fonction `ode.euler` permettant de résoudre numériquement un problème de Cauchy quelconque par le schéma d'Euler progressif.
- Ecrire un programme résolvant numériquement le problème de Cauchy associé au B.V.P. (4.1) à (4.4) et décrit en **Q. 8**. Ce programme devra utiliser la fonction `ode.euler`.
- Ecrire un programme permettant de calculer les 10 plus grandes valeurs propres en module de la matrice d'itération $\mathbb{B}$ lorsque la condition de C.F.L est vérifiée et lorsqu'elle ne l'est pas. On utilisera la fonction `eigs`.

6.2 Schéma de Runge-Kutta d'ordre 2

Un schéma de Runge-Kutta d'ordre 2 pour la résolution d'un problème de Cauchy s'écrit

$$\mathbf{Y}^{[n+1]} = \mathbf{Y}^{[n]} + h_t \mathcal{F} \left(t^n + \frac{h_t}{2}, \mathbf{Y}^{[n]} + \frac{h_t}{2} \mathcal{F}(t^n, \mathbf{Y}^{[n]}) \right),$$

Q. 11 [Maths]

- Montrer qu'avec $\mathcal{F}$ donné par (6.5), on a

$$\mathbf{Y}^{[n+1]} = \mathbb{B} \mathbf{Y}^{[n]} + \mathbf{c} \quad (6.10)$$

où l'on explicitera $\mathbb{B} \in \mathcal{M}_m(\mathbb{R})$ et $\mathbf{c} \in \mathbb{R}^m$.

- En utilisant la Proposition 6.1, montrer que

$$\nu \frac{h_t}{h^2} < \frac{1}{2} \Rightarrow \rho(\mathbb{B}) < 1. \quad (6.11)$$

Q. 12

- Ecrire une fonction `ode.RK2` permettant de résoudre numériquement un problème de Cauchy quelconque par le schéma de Runge-Kutta d'ordre 2 proposé.
- Ecrire un programme résolvant numériquement le problème de Cauchy associé au B.V.P. (4.1) à (4.4) et décrit en **Q. 8**. Ce programme devra utiliser la fonction `ode.RK2`.
- Ecrire un programme permettant de calculer les 10 plus grandes valeurs propres en module de la matrice d'itération $\mathbb{B}$ lorsque la condition de C.F.L est vérifiée et lorsqu'elle ne l'est pas. On utilisera la fonction `eigs`.