M1 IM/MPA - EDP et Différences Finies.

TP1 : Approximation de l'équation $-u" + cu = f$ par différences finies (4 séances).

I Conditions de Dirichlet

On cherche à approcher numériquement la solution du problème

$$\begin{cases} -u'' + cu = f & \text{sur }]a,b[, \\ u(a) = u_a \\ u(b) = u_b, \end{cases} \tag{1}$$

avec $c \geq 0$ et f deux fonctions continues données et $a, b, u_a, u_b \in \mathbb{R}$ définissant les conditions de Dirichlet.

Pour une subdivision uniforme de $[a,b]$ en $N+2$ points $X = (x_i)_{0 \leq i \leq N+1}$, de pas $h = (b-a)/(N+1) > 0$, on notera $U_h^{ex} = (u(x_i))_{0 \leq i \leq N+1}$ le vecteur des valeurs de la solution (exacte) de (1) aux points de la subvision.

De plus, on notera $U_h = (u_i)_{0 \leq i \leq N+1}$ les $N+2$ valeurs de l'approximation numérique de u obtenues par la méthode des différences finies.

Exercice 1

1. **Fonction ResolutionDH (Dirichlet homogène)** Créer une fonction *ResolutionDH* qui prend en entrée les paramètres N, a, b, c et f et qui renvoie le maillage X ainsi que l'approximation $U_h \in \mathbb{R}^{N+2}$ de la solution u de (1) sur $[a,b]$ avec conditions de Dirichlet *homogènes* ($u_a = u_b = 0$) obtenues grâce à la discrétisation de la dérivée seconde vue en cours.

   Indic 1 : Attention, pour une subdivision de $[a,b]$ en $N+2$ points, le système à résoudre sera de dimension N (de la forme $A_h u_h = b_h$ avec $u_h \in \mathbb{R}^N$) car la valeur de la solution sur le bord est déjà connue, on a alors $U_h = (u_0, u_h, u_{N+1}) \in \mathbb{R}^{N+2}$.

   Indic 2 : Pour résoudre un système linéaire vous pouvez utiliser la fonction `linalg.solve` *du package NumPy.*

2. **Validation 1/2** Dans cette question, on considère $c = c_1$ et $f = f_1$, avec c_1, f_1 définies sur $[0,1]$ par

   $$f_1(x) = (\pi^2 + 2x)\sin(\pi x) \text{ et } c_1(x) = 2x.$$

   — a) Vérifier (sur feuille) que $u : [0,1] \to \mathbb{R}, x \mapsto \sin \pi x$ est bien solution de (1) pour $a = 0$, $b = 1$ et $u_a = u_b = 0$.
   — b) Le principe du maximum est-il vérifié ?
   — c) Définir dans python les fonctions python $f1$, $c1$ et $u1$ correspondantes.

3. **Validation 2/2** On considère toujours $c = c_1$ et $f = f_1$ sur $[0,1]$. Sur un même graphe, représenter la solution approchée (pour $N = 100$) et la solution exacte (soignez l'affichage en ajoutant au moins un titre, un label pour les axes et une légende).

4. **Dirichlet non homogènes** Créer une fonction *ResolutionD* qui fait la même chose que la fonction *ResolutionDH* mais qui prend en plus deux arguments u_a et u_b et qui résout le problème (1) pour des conditions de bord de Dirichlet quelconques.

   Indication : On modifiera le vecteur $b_h \in \mathbb{R}^N$ (du schéma $A_h u_h = b_h$ pour les conditions de Dirichlet homogène) pour faire apparaitre l'influence de $u_0 = u_a$ et $u_{N+1} = u_b$ dans le calcul de u_1 et u_N. Par exemple, sur le bords $x = a$, on peut remarquer qu'on a la relation

$$\frac{2u_1 - u_2 - u_a}{h^2} + c(x_1)u_1 = f(x_1) \Leftrightarrow \frac{2u_1 - u_2}{h^2} + c(x_1)u_1 = f(x_1) + \frac{u_a}{h^2}.$$

   De même, on a

$$\frac{2u_N - u_b - u_N}{h^2} + c(x_N)u_1 = f(x_N) \Leftrightarrow \frac{2u_N - u_{N-1}}{h^2} + c(x_N)u_N = f(x_N) + \frac{u_b}{h^2}.$$

5. **Choisissez votre problème**
   — a) Choisissez, à votre goût, $a < b \in \mathbb{R}$, u_2 (une fonction deux fois dérivable, pas nécessairement positive, mais ne prenez pas un polynôme de degré ≤ 3) et c_2 (une fonction positive continue).
   — b) En déduire les valeurs de u aux bords (u_a, u_b) et la fonction f_2 telle que u_2 est solution de (1) pour $c = c_2$ et $f = f_2$.
   — c) Tracer sur le même graphe u_2 et son approximation numérique (obtenue avec *ResolutionD*).

6. **Erreur du schéma** On s'intéresse à l'erreur en norme L^2 et en norme infinie :

   - $\|U_h - U_h^{ex}\|_{h,\infty} = \max_{i\in\{0,\ldots,N+1\}} |u_i - u(x_i)|$, norme infinie discrète.
   - $\|U_h - U_h^{ex}\|_{h,2} = \sqrt{h}||U_h - U_h^{ex}||_2 = \sqrt{h}\left(\sum_{i=0}^{N+1} |u_i - u(x_i)|^2\right)^{\frac{1}{2}}$, norme L^2 discrète.

   Notons le coefficient $\sqrt{h} = 1/\sqrt{N+1}$ dans la norme euclidienne, qui permet de tenir compte du pas, et donc du nombre d'éléments dans la somme (quand N augmente, les erreurs ponctuelles $|u_i - u(x_i)|$ diminuent mais le nombre d'éléments à sommer augmente !).
   De plus, si on note $h = (b-a)/(N+1)$, $v \in L^2(a,b)$ $V = (v_0, v_1, \ldots v_{N+1}) \in \mathbb{R}^{N+2}$ où $v_i := v(a + ih)$, on a

$$\begin{aligned} \|v\|_{L^2} &:= (\int_a^b v(x)^2 dx)^{1/2} = \lim_{h\to 0}(\sum_0^{N+1} v_i^2 * h)^{1/2} = \lim_{h\to 0}\sqrt{h}(\sum_0^{N} v_i^2)^{1/2} \\ &= \lim_{h\to 0}\sqrt{h}\|U_h\|_2 = \lim_{h\to 0}\|U_h\|_{h,2} \end{aligned} \qquad (2)$$

   où la première norme est dans l'espace des fonctions L^2 et la seconde dans $\mathbb{R}^n$.

   Pour chacune de ces normes discrètes, représenter sur un graphe l'erreur en fonction du du pas h (on pourra par exemple faire varier N entre 50 et 1000 par pas de 10).

7. **Ordre du schéma**

— a) Pour chacune de ces normes discrètes, représenter sur un graphe le *logarithme* de cette erreur en fonction du *logarithme* du pas h. *On pourra, au choix, utiliser* `matplotlib.pyplot.loglog` *ou la fonction* `log` *de NumPy*

— b) En déduire l'ordre de convergence du schéma [1]. Pour estimer la pente, on peut calculer son taux d'accroissement. On pourra également utiliser la fonction `np.polyfit(x,y,d)` qui renvoie les coefficients (dans l'ordre des degrés décroissant) du polynôme de degré d (ici $d = 1$) minimisant la distance aux données au sens des moindres carrés (norme euclidienne).

8. **[Question optionnelle :] Solution dans** $\mathbb{R}_3[X]$ Répéter les questions 5 et 6 en faisant le choix d'une fonction $u = u3$ qui soit un polynôme de degré ≤ 3. Que constate-t-on ? Proposer une explication. *Indic : une erreur inférieur à* 10^{-10} *peut être considérée comme nulle (erreurs d'arrondis).*

II Conditions mixtes de Dirichlet et Neumann

On considère le même problème que dans la partie I mais avec une condition de **Neumann** sur le bord droit :

$$\begin{cases} -u'' + cu = f \quad \text{sur }]a,b[, \\ u(a) = u_a \\ u'(b) = v_b, \end{cases} \tag{3}$$

avec $c \geq 0$ et f deux fonctions continues données et $a, b, u_a, v_b \in \mathbb{R}$ définissant les conditions de bord.

Exercice 2

1. **Neumann** Créer une fonction *ResolutionN* qui fait la même chose que la fonction *ResolutionD* de l'exercice 1 mais avec une conditions de Dirichlet sur le bord $x = a$ et de Neumann en $x = b$ ($u(a) = u_a$, $u'(b) = v_b$).

   Indication : Comme dans l'exercice 1, u_0 *est connu et* u_{N+1} *entièrement déterminé par* u_N *donc la dimension du système à résoudre est* N. *On modifiera la matrice* $A_h \in M_N(\mathbb{R})$ *et le vecteur* $b_h \in \mathbb{R}^N$ *(du schéma* $A_h u_h = b_h$*) pour faire intervenir la condition de Dirichlet en* $x = a$ *(Cf exercice 1) et la discrétisation de la dérivée sur le bord* $x = b$. *En particulier, au point* x_N *on a les relation*

$$\begin{cases} \frac{2u_N - u_{N-1} - u_{N+1}}{h^2} + c(x_N)u_N = f_N \\ \frac{u_{N+1} - u_N}{h} = v_b \end{cases} \tag{4}$$

---

1. **Ordre de convergence et représentation graphique :** On dit qu'un schéma converge à l'ordre p pour la norme $|| \ ||$ si p est le plus grand entier pour lequel il existe une constante $C > 0$ indépendante de h telle que

$$||U_h - U_h^{ex}|| \leq Ch^p.$$

En pratique, on a $||U_h - U_h^{ex}|| \approx Ch^p$ et ainsi

$$\ln(||U_h - U_h^{ex}||) \approx \ln(C) + p\ln(h).$$

L'ordre de convergence p de la méthode peut ainsi être déduit à partir de la pente de la droite obtenue en traçant l'erreur $||U_h - U_h^{ex}||$ en fonction de h en échelle log-log.

En isolant u_{N+1} dans la seconde équation et en l'injectant dans la première, il vient

$$\begin{cases} \frac{u_N - u_{N-1}}{h^2} + c(x_N)u_N = f_N + \frac{v_b}{h} \\ u_{N+1} = u_N + hv_b \end{cases} \tag{5}$$

2. **Validation** Répéter les questions 5 et 7 de l'exercice 1 avec votre fonction *resolutionN*. Que constate-t-on ? Commenter.
3. **Schéma d'ordre 2** Pour retrouver un schéma d'ordre 2, on va utiliser la technique du *point fantôme*[2], qui permet d'utiliser l'approximation centrée de la dérivée simple en $x = b$, cette approximation étant d'ordre 2 :

$$u'(x) = \frac{u(x+h) - u(x-h)}{2h} + \mathcal{O}(h^2)$$

.

Méthode du point fantôme :

— On crée un point artificiel $x_{N+2} = b + h$

— Au point x_{N+1}, le schéma satisfait la condition de Neumann **et** la relation $u'' + cu = f$. Cette dernière est discrétisée par

$$\frac{2u_{N+1} - u_N - u_{N+2}}{h^2} + c(x_{N+1})u_{N+1} = f(x_{N+1})$$

— La condition de Neumann en $x = b$ s'écrit, par approximation centrée de la dérivée, $(u_{N+2} - u_N)/2h = v_b \Leftrightarrow u_{N+2} = u_N + 2hv_b$.

— En injectant dans la relation précédente, il vient

$$\frac{2u_{N+1} - 2u_N}{h^2} + c(x_{N+1})u_{N+1} = f(x_{N+1}) + \frac{2v_b}{h}.$$

— Ce système peut s'écrire, sous forme matricielle, comme un système de taille $N+1$ (identique au système utilisé dans l'exercice 1 pour les lignes 1 à $N-1$) de la forme $\tilde{A}_h u_h = \tilde{b_h}$ avec

$$A_h = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & \dots & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & \dots & \dots & 0 & -2 & 2 \end{pmatrix} + diag(C) \text{ et } b_h = \begin{pmatrix} f(x_1) + u_a/h^2 \\ f(x_2) \\ \vdots \\ f(x_N) \\ f(x_{N+1}) + \frac{2v_b}{h} \end{pmatrix}$$

où $diag(C)$ est la matrice de diagonale $c(x_1), \dots, c(x_{N+1})$.

4. En repartant du code de votre fonction *ResolutionN*, construire une fonction *ResolutionN2* qui implémente la méthode du point fantôme.
5. Reprendre la question 2 et vérifier que le schéma converge à l'ordre 2.

2. Voir par exemple https ://stordeux.perso.univ-pau.fr/COURS/AN1.pdf

III Conditions de Dirichlet en 2D

On cherche à approcher numériquement la solution $u : [0,1] \times [0,1] \to \mathbb{R}$ de l'équation $-\Delta u + cu = f$, sur $]0,1[\times]0,1[$ avec des conditions aux bords de Dirichlet homogènes. : $u(0,\cdot) = u(1,\cdot) = u(\cdot,0) = u(\cdot,1) = 0$. On rappelle que

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}.$$

On suppose que c et f sont continue, que c est positive et $u \in \mathcal{C}^3$. On admet l'existence d'une unique solution u à ce problème.

On choisit une discrétisation du pavé $[0,1]\times[0,1]$ à l'aide d'une grille uniforme $(x_i, y_j)_{(i,j)\in\{0,\ldots,N+1\}^2}$ de pas $h > 0$. On cherche $(u_{i,j})_{(i,j)\in\{0,\ldots,N+1\}^2}$, $(N+2)^2$ valeurs telles que pour tout $(i,j) \in \{0,...,N+1\}^2$, $u_{i,j}$ est une "bonne approximation" de $u(x_i, y_j)$.

Exercice 3 1. Démontrer l'identité

$$\Delta u = \frac{u(x+\eta,y) + u(x-\eta,y) + u(x,y+\eta) + u(x,y-\eta) - 4u(x,y)}{\eta^2} + O(\eta^2)$$

2. En s'inspirant de la question 1 et du schéma pour la dimension 1 d'espace, donner un schéma numérique permettant de calculer $(u_{i,j})_{(i,j)\in\{0,\ldots,N+1\}^2}$; c'est à dire une relation satisfaite par les inconnues $(u_{i,j})_{(i,j)\in\{0,\ldots,N+1\}^2}$ (on ne demande pas de mettre le schéma sous forme matricielle).
3. En utilisant la notation :

$$u_h^{(j)} = \begin{pmatrix} u_{1,j} \\ \vdots \\ u_{N,j} \end{pmatrix}, \; j \in \{1,...,N\}$$

et

$$u_h = \begin{pmatrix} u_h^{(1)} \\ \vdots \\ u_h^{(N)} \end{pmatrix} \in \mathbb{R}^{N^2},$$

écrire le schéma obtenu sous forme matricielle $A_h U_h = b_h$.
Indication : on pourra écrire b_h et A_h sous la forme

$$b_h = \begin{pmatrix} F_h^{(1)} \\ \vdots \\ F_h^{(N)} \end{pmatrix} \;\; avec \; F_h^{(i)} = \begin{pmatrix} f(x_1, y_i) \\ \vdots \\ f(x_N, y_i) \end{pmatrix} \;\; et \;\; A_h = \frac{1}{h^2} \begin{pmatrix} \tilde{A} & -I & & & \\ -I & \tilde{A} & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & \tilde{A} & -I \\ & & & -I & \tilde{A} \end{pmatrix} + diag(C_h)$$

avec $\tilde{A}$ et $diag(C_h) \in M_{N,N}(\mathbb{R})$ à déterminer, et I la matrice identité de $M_{N,N}(\mathbb{R})$

4. Soit $u : [0,1] \times [0,1] \to \mathbb{R}, (x,y) \mapsto \sin(4\pi x)\sin(3\pi y)$ et $c(x,y) = e^x e^y$. Calculer Δu. Pour quel second membre f l'équation $-\Delta u + cu = f$ est-elle satisfaite ?
5. Implémenter une fonction DirichletH2D qui prend en argument N, c et f et qui renvoie les vecteurs $X = (x_i)_{0 \leq i \leq N+1}$ et $Y = (y_i)_{0 \leq i \leq N+1}$ ainsi qu'une matrice (un tableau) $U \in M_{N+2 \times N+2}(\mathbb{R})$ contenant les $(u_{i,j})_{0 \leq i,j \leq N+1}$.
Indication : pour créer la matrice U, on pourra commencer par trouver $u_h \in \mathbb{R}^{N\times N}$, solution de $A_h u_h = b_h$ puis réorganiser u_h en une matrice de $M_{N\times N}(\mathbb{R})$ (par exemple avec la méthode reshape*). Enfin, créera la matrice U qui possède des 0 sur ses bords et dont les valeurs intérieures sont données par la matrice de u_h*
6. Considérer l'équation de la question 4 et tracer l'évolution de l'erreur en norme infinie en fonction du pas h en échelle log-log. En déduire l'ordre de convergence de la méthode.
7. Pour $N = 50$ et pour les fonctions c et f de la question 4, tracer la solution numérique de l'équation $-\Delta u + cu = f$ sur $[0,1] \times [0,1]$ avec conditions de Dirichlet homogènes. 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` Vous devriez obtenir la figure suivante