Université de Franche-Comté
Licence de Mathématiques, Semestre 6, 2018-19

Discrétisation des EDPs
Feuille de TP n^{o}2
Méthodes des différences finies

On considère le problème aux limites

$$(\mathrm{P})\quad \begin{cases} -u'' + cu = f \text{ dans }]0,1[\\ u(0) = 0 \\ \alpha u(1) + \beta u'(1) = 0 \end{cases}$$

où $f, c \in \mathscr{C}([0,1];\mathbb{R})$ avec $c \geq 0$ et $\alpha, \beta \in \mathbb{R}$ sont donnés.

Pour $N \in \mathbb{N}$ donné, on note $(x_0, \ldots, x_{N+1})$ la subdivision uniforme de pas h de $[0,1]$ et u_k désigne une approximation de $u(x_k)$ pour tout $k \in [\![0, N+1]\!]$. On note $\mathbf{U}_h = (u_k)_{k=1}^{N}$ et $\overline{\mathbf{U}}_h = (u_k)_{k=0}^{N+1}$.

Consignes-conseils

> Toutes les fonctions seront écrites dans le même fichier `ApproximationsEDP.sci`.
> Vous pourrez faire un script (fichier avec extension ”`.sce`”) par question.
> Les graphiques devront tous contenir un titre et des légendes.

1. *Approximation de l'équation*
   On approche l'équation $-u'' + cu = f$ dans $]0,1[$ par le schéma aux différences finies centré d'ordre 2 et on met le système linéaire ainsi obtenu sous la forme matricielle
   $$\mathbf{A}_h \mathbf{U}_h = \mathbf{b}_h \tag{1}$$
   (a) Rappeler les expressions de $\mathbf{A}_h$ et $\mathbf{b}_h$ et préciser leurs tailles.
   (b) Construire la matrice $\mathbf{A}_h$ de deux façons différentes :
      - en utilisant une ou plusieurs boucles `for` ;
      - en utilisant les commandes `diag` et `eye` pour créer une matrice tridiagonale.

      Vérifier que les deux matrices ainsi construire coïncident (utiliser la commande `disp` pour l'affichage).
   (c) Utiliser Scilab pour montrer que la matrice $\mathbf{A}_h$ est symétrique et définie positive.
   (d) Construire le second membre $\mathbf{b}_h$.

2. *Prise en compte des conditions aux limites*
   La condition limite en $x = 1$ peut être de trois types :
   † Dirichlet : $(\alpha, \beta) = (1, 0)$ † Neumann : $(\alpha, \beta) = (0, 1)$ † Robin : $(\alpha, \beta) = (1, 1)$.
   (a) *Dirichlet* : quelle méthode suggérez vous pour approcher les conditions $u(0) = 0$ et $u(1) = 0$?
   (b) *Neumann* : proposer 2 méthodes pour approcher la condition $u'(1) = 0$.
   (c) *Robin* : proposer 2 méthodes pour approcher la condition $u(1) + u'(1) = 0$.
   (d) Utiliser la command `input` pour que l'utilisateur puisse choisir la condition limite en $x = 1$ et construire, à partir de $\mathbf{A}_h$ et $\mathbf{b}_h$ assemblés précédement, la matrice $\overline{\mathbf{A}}_h$ et le vecteur $\overline{\mathbf{b}}_h$ tq
   $$\overline{\mathbf{A}}_h \overline{\mathbf{U}}_h = \overline{\mathbf{b}}_h \tag{2}$$
   prenne en compte la condition $u(0) = 0$ et celle choisie par l'utilisateur en $x = 1$ (on pourra utiliser lla commande `select...case`)

3. *Validation*
   (a) Quelle méthode suggérez vous pour valider votre programme ?
   (b) Effectuer cette validation en l'illustrant par des graphiques (utiliser la commande \ pour résoudre (2)).

4. *Erreur de convergence*
   Etudier dans les différents cas et en l'illustrant par des graphiques, l'erreur de convergence en fonction de N (on pourra utiliser la commande `regress`).