DEVOIR 2

Analyse numérique pour ingénieur — Automne 2018
MAT-2910

Consignes

— Ce devoir doit obligatoirement être réalisé avec le logiciel Matlab.
— Vous ne pouvez pas obtenir d'aide du CDA pour ce devoir.
— Il n'y a pas de rapport à fournir. Il y a au total 6 programmes Matlab (.m) à créer et fournir : un fichier script (*devoir2 .m*) et 5 fonctions Matlab (*choleski.m*, *remontee.m*, *descente.m*, *resochol.m* et *resolution.m*).
Le fichier script fait appel aux fonctions. Lorsqu'on lance le fichier script, il doit produire les 2 figures demandées. Si les figures n'apparaissent pas, l'équipe ou l'étudiant(e) se verra attribuer la note 0.
— Des points sont alloués à la qualité (lisibilité) des figures qui doivent être produites par le script Matlab demandé. Il est donc recommandé de donner des titres (ou légendes) aux figures, ainsi que d'indiquer ce qu'il y a en abscisse ou en ordonnée. L'utilisation de différentes couleurs pour plusieurs courbes sur une même figure est recommandée.
— Les fichiers Matlab demandés doivent être compressés en 1 seul fichier qui doit porter le nom de l'auteur ou les noms des auteurs, et qui doit être déposé dans la **boîte de dépôt** sur le site du cours : www.portaildescours.ulaval.ca
— Le devoir peut être réalisé en équipe d'au plus deux personnes. Si l'équipe est constituée de 2 personnes, il ne doit pas s'agir d'une équipe ayant déjà été formée pour réaliser le devoir 1, sinon la note sera 0.
— Tout plagiat, même partiel, sera sanctionné (voir plan de cours).
— Il vous incombe de régler les conflits 1 semaine avant la date de remise des programmes. À moins d'1 semaine de la date de remise, les équipes **ne pourront plus être modifiées**.

Remise

— Date de remise : voir sur le site du cours (www.portaildescours.ulaval.ca).
— Lieu de remise : boîte de dépôt sur le site du cours (www.portaildescours.ulaval.ca)
— **Tout travail remis après la date et l'heure d'échéance se verra pénalisé de 2% pour chaque 15 minutes de retard pour un maximum 20% pour le premièr jour. Par la suite, une pénalité de 10% par jour de retard sera imposée.**

Support

— 2 séances d'information sont prévues, pour les lieux et heures voir sur le site du cours (www.portaildescours.ulaval.ca).

But du travail :

Ce travail traite de la résolution numérique des problèmes aux limites en dimension un. Pour plus de détails, on pourra consulter le chapitre 7 du manuel du cours.

On veut résoudre l'équation différentielle linéaire du deuxième ordre

$$\begin{array}{lcll} -y''(x) & = & r(x), & x \in [0,\, 1], \\ y(0) & = & 0, & \\ y(1) & = & 0. & \end{array} \tag{1}$$

Un tel problème est appelé problème aux limites, car la fonction inconnue y doit satisfaire les conditions aux limites $y(0) = 0$ et $y(1) = 0$, posées aux limites de l'intervalle $[0,\, 1]$.

Pour approcher la solution du problème (1), on subdivise l'intervalle $[0, 1]$ en N sous-intervalles de même longueur

$$h = \frac{1}{N}.$$

On définit les noeuds de la subdivision par

$$x_i = i\ h, \qquad i = 0, 1, \ldots, N.$$

Au lieu de chercher une fonction $y(x)$ qui satisfait (1), on cherche un vecteur $\vec{y} = (y_0, y_1, \ldots, y_N)$ dont la ième composante y_i donne une approximation de la valeur exacte $y(x_i)$ au ième noeud x_i. Bien sûr, on impose dès le départ

$$y_0 = 0, \quad y_N = 0.$$

Il nous reste à déterminer $N-1$ approximations $y_1, \ldots, y_{N-1}$. Pour ce faire, on construit un système linéaire en remplaçant dans (1) x par chacun des noeuds **intérieurs** x_i, $y(x_i)$ par l'inconnue y_i et la dérivée seconde par la formule de différence finie centrée d'ordre 2 :

$$y''(x_i) \simeq \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} \simeq \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \tag{2}$$

On obtient un système linéaire de $N-1$ équations et $N-1$ inconnues $y_1, \ldots, y_{N-1}$ qui s'écrit

$$-\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = r_i, \qquad \forall i = 1, \ldots, N$$

où on a utilisé la notation $r_i = r(x_i)$.

On peut aussi récrire le système sous forme matricielle

$$
\begin{pmatrix}
2 & -1 & 0 & \dots & 0 \\
-1 & 2 & -1 & \ddots & \vdots \\
0 & -1 & 2 & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots & -1 \\
0 & \dots & 0 & -1 & 2
\end{pmatrix}
\begin{pmatrix}
y_1 \\ y_2 \\ \vdots \\ y_{N-1}
\end{pmatrix}
= h^2
\begin{pmatrix}
r_1 \\ r_2 \\ \vdots \\ r_{N-1}
\end{pmatrix}
\tag{3}
$$

Notons que la matrice a une structure bande, plus précisément *tridiagonale*. De plus, elle est symétrique et on peut montrer qu'elle est définie positive (preuves que de telles matrices peuvent apparaître dans certains problémes !). On va donc résoudre le système linéaire, par la méthode de Choleski et en prenant en compte la structure bande de la matrice, afin bien sûr de faire des économies de calcul.

Question 1

a) Écrire une fonction Matlab intitulée *choleski.m*, qui aura trois arguments (entrées) : une matrice A, supposée symétrique et définie positive, le nombre n de lignes, ainsi qu'un entier m désignant sa largeur de bande (dans l'exemple de l'équation (3) on a $m = 2$). Elle aura comme unique sortie une matrice L triangulaire inférieure. Cette fonction construira la matrice L de la factorisation de Choleski en prenant en compte la structure bande de la matrice A (et de L).

b) Écrire une fonction Matlab intitulée *remontee.m* qui aura quatre arguments (entrées) : une matrice A, supposée triangulaire supérieure, le nombre n de lignes, un vecteur b, ainsi qu'un entier m désignant la largeur de bande supérieure de A. Elle aura comme unique sortie un vecteur x, solution de l'équation $Ax = b$, calculée en faisant une remontée triangulaire prenant en compte la largeur de bande de A.

c) Écrire une fonction Matlab intitulée *descente.m* qui aura quatre arguments : une matrice A, supposée triangulaire inférieure, le nombre n de lignes, un vecteur b, ainsi qu'un entier m désignant la largeur de bande inférieure de A. Elle aura comme unique sortie un vecteur x, solution de l'équation $Ax = b$, calculée en faisant une descente triangulaire prenant en compte la largeur de bande de A.

d) Écrire une fonction Matlab intitulée *resochol.m* qui aura quatre arguments : une matrice A, supposée symétrique définie positive, le nombre n de lignes, un vecteur b, ainsi qu'un entier m désignant la largeur de bande inférieure (ou supérieure) de A. Elle aura comme unique sortie un vecteur x, solution de l'équation $Ax = b$, calculée en faisant appel aux 3 fonctions précédentes (et donc en faisant une factorisation de Choleski), et prenant en compte la structure bande de A.

Question 2.

a) Écrire une fonction Matlab intitulée *resolution.m*, qui aura deux arguments (entrées) : une fonction $r(x)$ et le paramètre N désignant le nombre de sous-intervalles de $[0, 1]$. Elle aura comme unique sortie le vecteur $\vec{y} = (y_0, y_1, \ldots, y_N)$ obtenu en résolvant (3) et en prenant $y_0 = 0$ et $y_N = 0$. Bien sûr la résolution de (3) se fera en faisant appel à la fonction *resochol.m*.

b) Créer un fichier script Matlab intitulé *devoir2.m* . Donner vous une solution $y(x)$, et en déduire le second membre $r(x)$ de (1) correspondant. Il faut bien sûr que $y(0) = y(1) = 0$. Ensuite calculer les approximations de $y(x)$ pour 2 différentes valeurs de h en faisant appel à *resolution.m* et tracer (**Figure 1**) les approximations ensembles avec le graphe de $y(x)$ sur $[0, 1]$. Les valeurs de h doivent être telles que l'on puisse distinguer les différentes approximations entre elles, et qu'on puisse les distinguer de $y(x)$. Utiliser la commande Matlab *legend* pour bien indiquer quelles sont les 3 courbes et indiquer quelles valeurs de h ont été utilisées pour chacune des 2 courbes des approximations. Dans le titre de la figure (commande Matlab *title*) doit figurer l'expression de $y(x)$.

c) Toujours dans le script *devoir2.m*, calculer différentes approximations de $y(x)$ pour différentes valeurs de h $(10^{-1}, 10^{-2}, 10^{-3}, \cdots)$ et tracer (**Figure 2**) la courbe d'erreur $E(h)$ en fonction de h en échelle logarithmique (commande *loglog*), où

$$E(h) = \max_{i=1,\ldots,N-1} |y_i - y(x_i)|.$$

Avec l'échelle logarithmique on pourra "lire" l'ordre de la méthode, c'est-à-dire la coefficient p tel que l'erreur se comporte comme $E(h) \approx Ch^p$, puisque p sera la pente de la courbe (droite) ainsi obtenue.