Ecole des Mines de Nancy
2014-2015
Mathématiques: TCS11

Analyse numérique: Test 2 - sujet B
(durée du test: 2h)

- *Le polycopié du cours et les notes de TD sont autorisés à l'exclusion de tout autre document. Vous pouvez réutiliser vos fonctions Matlab créées en TD.*
- *Durant la première heure, on travaillera uniquement sur feuille: les ordinateurs resteront éteints. On écrira sur la feuille les algorithmes ou fonctions Matlab qui seront codés dans la suite. Durant la seconde heure, seul Matlab sera ouvert sur votre session: en particulier tout accès à Internet (navigateur, mail...) est interdit.*
- *A la fin de la séance, vous rendez votre copie et vous envoyez à l'adresse notée au tableau vos fonctions Matlab. Il serait judicieux et apprécié par le correcteur de rassembler vos fonctions de manière à ce qu'on puisse les exécuter très facilement (par exemple en créant un fichier principal).*
- *Les quatre exercices sont indépendants et peuvent être faits dans un ordre quelconque.*

Exercice 1:

On considère les matrices

$$A_1 := \begin{pmatrix} -3 & 3 & -6 \\ -4 & 7 & -8 \\ 5 & 7 & -9 \end{pmatrix}$$

et

$$A_2 := \begin{pmatrix} 7 & 6 & 9 \\ 4 & 5 & -4 \\ -7 & -3 & 8 \end{pmatrix}.$$

Soient le vecteur unité $\mathbf{x} := (1,1,1)^T$ et les vecteurs second membres associés $\mathbf{b}_j := A_j\mathbf{x}$, $j = 1, 2$.

1) Est ce que la méthode de Gauss-Seidel converge pour la résolution du système linéaire $A_1\mathbf{x} = \mathbf{b}_1$? Donner une preuve mathématique.

2) Est ce que la méthode de Jacobi converge pour la résolution du système linéaire $A_2\mathbf{x} = \mathbf{b}_2$? Donner une preuve mathématique.

3) Est ce que la méthode de Gauss-Seidel converge pour la résolution du système linéaire $A_2\mathbf{x} = \mathbf{b}_2$? Donner une preuve mathématique.
4) Que peut-on déduire quant à la comparaison des méthodes de Jacobi et de Gauss-Seidel pour résoudre $A_2\mathbf{x} = \mathbf{b}_2$?

Exercice 2:

On considère le système non linéaire suivant

$$\begin{cases} e^{x_1^2+x_2^2} - 1 = 0, \\ e^{x_1^2-x_2^2} - 1 = 0. \end{cases} \tag{1}$$

1) Calculer l'unique solution $\mathbf{x}^*$ de ce système.

2) Ecrire une fonction Matlab appelée `Newton_Systeme` qui calcule la solution du système non linéaire précédent par la méthode de Newton. Vous préciserez notamment les arguments en entrée

- point initial $\mathbf{x}_0$,
- tolérance ε sur le critère d'arrêt: $||\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}|| \leq \varepsilon$,
- nombre maximal d'itérations $k^{\max}$,

et de sortie

- matrice des itérés successifs
$$[\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, ..., \mathbf{x}^{(k)}],$$
- nombre d'itérations à convergence (ou divergence) k^{iter}.

3) Ecrire un script Matlab `exo2` où vous testerez la fonction `Newton_Systeme` en prenant les valeurs $\mathbf{x}_0 := (0,0)^T$, $k^{\max} = 500$ et $\varepsilon := 10^{-10}$. Quel résultat obtenez-vous?

4) Que se passe t-il si la donnée initiale est $\mathbf{x}_0 := (20,20)^T$?

Exercice 3:

On souhaite résoudre le système dynamique

$$\begin{cases} y'(t) = f(t,y), t > 0, \\ y(t=0) = y_0. \end{cases} \tag{2}$$

Une possibilité consiste à utiliser une méthode multi-pas du type

$$y_{n+1} := y_n + \frac{h}{2}(3f_n - f_{n-1}), n \geq 1,$$

avec $y_n = y(t_n)$, h le pas de la méthode et $f_n = f(t_n, y_n)$.

1) Comment faire pour initialiser la méthode?

2) Ecrire une fonction Matlab `Adams` permettant de résoudre un équation différentielle ordinaire donnée. Préciser les arguments de la fonction afin qu'elle puisse prendre en compte une fonction f générale.

3) Ecrire un script Matlab `exo3` qui permette de tester cette fonction sur un exemple de votre choix.

Exercice 4:

On considère l'équation différentielle

$$\begin{cases} -y'' - 3y' - 2y = \sin x \\ y'(0) = 0 \quad y(1) = 2. \end{cases} \tag{3}$$

1) a) Déterminer toutes les solutions de l'équation différentielle $-y'' - 3y' - 2y = 0$.
b) Chercher une solution particulière de l'équation différentielle $-y'' - 3y' - 2y = \sin x$ sous la forme $y(x) = a\cos x + b\sin x$.
c) En déduire la solution de l'équation (3).

2) On veut utiliser une méthode de différences finies pour résoudre cette équation. Écrire le système linéaire qu'on doit résoudre.

3) Écrire le programme Matlab pour résoudre cette équation différentielle (on pourra utiliser la résolution de systèmes linéaires préprogammée dans Matlab). On mettra en entrée n le nombre de points de la discrétisation. Tracer sur un même graphique la solution approchée et la solution exacte sur l'intervalle $[0,1]$.

4) Évaluer l'erreur entre la solution approchée et la solution exacte pour plusieurs valeurs de n, nombre de points choisis pour la discrétisation. Est-ce en accord avec la théorie?