Ecole des Mines de Nancy
2016-2017
Mathématiques

Analyse numérique: Test 2 - sujet A

(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. 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. Chaque fonction Matlab que vous créez sera nommée exoN-Nom.m où N est le numéro de l'exercice et Nom votre nom de famille. Attention: on pourra enlever des points si cette consigne n'est pas respectée!*
- *Les quatre exercices sont indépendants et peuvent être faits dans un ordre quelconque. Quand la question est précédée de* **t** *il s'agit d'une question théorique, quand elle est précédée de* **m** *il s'agit d'une question Matlab.*

Exercice 1:
On veut résoudre l'équation différentielle

$$\begin{cases} y''(t) = y'(t) - ty(t) + 1 \\ y(0) = 0 \,,\; y'(0) = 1\,. \end{cases} \tag{1}$$

t 1) Mettre cette équation sous la forme d'un système $Y'(t) = F(t, Y(t))$, $Y(0) = Y_0$.
t 2) On veut résoudre l'équation (1) par la méthode de Runge-Kutta d'ordre 2 avec un pas constant h. Calculer Y_1 (valeur approchée de $Y(h)$).
m 3) Programmer en Matlab la méthode de Runge-Kutta d'ordre 2 avec le pas h pour résoudre (1) sur l'intervalle $[0, 2]$ et tracer la courbe de la solution aprochée ainsi obtenue ($t \mapsto y(t)$).

Exercice 2:
Soit A_n la matrice tridiagonale $n \times n$ définie par

$$A_n = \begin{pmatrix} 4 & -1 & 0 & \dots & 0 \\ -1 & 4 & -1 & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -1 & 4 & -1 \\ 0 & \dots & 0 & -1 & 4 \end{pmatrix}.$$

t 1) Montrer que pour tout entier $n \geq 2$, les méthodes de Jacobi et Gauss-Seidel (pour résoudre un système linéaire $A_nX = b$) vont converger (*si vous ne savez pas le montrer pour tout n, faites-le pour $n = 2$*).

m 2) Programmer la méthode de Jacobi: on écrira une fonction Matlab qui prend en entrée l'entier n et un vecteur b et qui envoie en sortie la solution X de $A_nX = b$ ainsi que le nombre d'itérations. On prendra une précision $eps = 1E-6$ et on écrira un test d'arrêt faisant intervenir le résidu $\|A_nX - b\|$. Donnez le résultat que vous obtenez pour $n = 10$ et $b = ones(10, 1)$
t 3) Montrer que, pour tout entier n, A_n est symétrique définie positive, et que son conditionnement pour la norme $\|.\|_2$ vérifie $cond_2(A_n) \leq 3$.

Exercice 3:
Soit b un nombre réel positif. On considère l'algorithme de point fixe

$$\begin{cases} x_{n+1} = \dfrac{3}{4}\,x_n + \dfrac{b}{x_n^3} \\ x_0 > 0 \text{ donné}\,. \end{cases} \tag{2}$$

t 1) Si cet algorithme converge, vers quel x^* converge-t-il?
t 2) Si on prend x_0 proche de x^*, est-ce que l'algorithme va converger?
t 3) Plus précisément, si on prend $b = 7$ et $x_0 = 2$, montrer que l'algorithme va converger.
m 4) Programmer en Matlab l'algorithme. On écrira une fonction qui prend en entrée b et x_0 et qui renvoie en sortie x^* et le nombre d'itérations. On expliquera quel(s) test(s) d'arrêt on utilise.

Exercice 4:
Certaines compagnies aériennes autorisent à embarquer un bagage cabine dont *la somme des trois dimensions* est plus petite que 120 cm.
t 1) En supposant que notre bagage a la forme d'un parallélépipède (ou pavé) de dimensions a, b, c et qu'on cherche à maximiser son volume, écrire le problème d'optimisation avec contraintes qu'on souhaite résoudre.
t 2) Montrer l'existence d'une solution à ce problème d'optimisation.
t 3) Résoudre ce problème en écrivant les conditions d'optimalité de Kuhn-Tucker.
m 4) Retrouver le résultat en utilisant la fonction *fmincon* de Matlab (indiquer la syntaxe utilisée et les fonctions créées pour utiliser fmincon).