INSA

AROM-GM-3

Année universitaire 2016–2017
20/10/2016
Durée : 2h00

EXAMEN : INTRODUCTION AUX LOGICIELS MATHEMATIQUES

Seuls les notes et les programmes de TP ainsi que les documents distribués en TP sont autorisés.

Exercice 1 `Logiciel MATLAB`, *Approximation des dérivées :*

Soit f une fonction dérivable sur $[a, b]$ à valeurs dans $\mathbb{R}$. On cherche des valeurs approchées de $f'(x)$ sur un échantillonnage.

Soit $n \in \mathbb{N}$, $n \geq 3$. On pose $h = \dfrac{b-a}{n-1}$, $x_k = a + (k-1)h,\ k = 1, \dots, n$ et $\boldsymbol{Y} = [f(x_1), \dots, f(x_n)]^T$. Une approximation de $\boldsymbol{D}$ du vecteur $[f'(x_1), \dots, f'(x_n)]^T$ est donnée en résolvant le système

$$\boldsymbol{AD} = \frac{1}{h}\boldsymbol{BY}$$

où $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n \times n}$ sont définis par

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & -1 & 0 & & \\ 1 & 4 & 1 & 0 & & \\ 0 & 1 & 4 & 1 & \ddots & \\ & \ddots & \ddots & \ddots & \ddots & 0 \\ & & 0 & 1 & 4 & 1 \\ & & 0 & 1 & 0 & -1 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} -2 & 4 & -2 & 0 & & \\ -3 & 0 & 3 & 0 & & \\ 0 & -3 & 0 & 3 & \ddots & \\ & \ddots & \ddots & \ddots & \ddots & 0 \\ & & 0 & -3 & 0 & 3 \\ & & 0 & -2 & 4 & -2 \end{bmatrix}$$

1. Ecrire un fichier "function" `f.m` qui contiendra la fonction f, par exemple $f(x) = \cosh(x) = \frac{e^x + e^{-x}}{2}$.

```
>> f([0.5 1]')
ans =
    1.1276
    1.5431
```

2. Ecrire un fichier "script" `VI1.m` ("VI" sont vos initiales...), qui étant donnés l'entier n construit et affiche la matice $\boldsymbol{A}$. Tester avec $n = 5$. *On pourra utiliser les fonctions* `diag et ones`*, commencer par construire la matrice $n \times n$ avec des 4 sur la diagonale et des 1 sur les sur et sous diagonales, puis corriger les lignes 1 et n.*

```
>> VI1
A =
     1     0    -1     0     0
     1     4     1     0     0
     0     1     4     1     0
     0     0     1     4     1
     0     0     1     0    -1
```

3. Compléter le programme précédent en ajoutant la construction de $\boldsymbol{B}$. Sauvegarde sous `VI2.m`. Test avec $n = 5$.

```
>> VI2
B =
    -2     4    -2     0     0
    -3     0     3     0     0
     0    -3     0     3     0
     0     0    -3     0     3
     0     0    -2     4    -2
```

4. Compléter le programme précédent en ajoutant le calcul de $\boldsymbol{Y}$, du second membre $\frac{1}{h}\boldsymbol{B} \times \boldsymbol{Y}$ et enfin le calcul des dérivées approchées $\boldsymbol{D}$ puis l'affichage de $\boldsymbol{D}$. Sauvegarde sous `VI3m`. Test avec $n = 5$, $a = 0$, $b = 2$ et la fonction $f(x) = \cosh(x)$.

```
>> VI3
D =
    0.0214
    0.5161
    1.1727
    2.1419
    3.5745
```

5. Compléter le programme précédent en ajoutant le graphe de la dérivée approchée et de la dérivée exacte. Sauvegarde sous `VI4.m`. Mêmes éléments test. *Pour $f(x) = \cosh(x)$, la dérivée exacte est $f'(x) = \sinh(x)$...*

6. Compléter le programme précédent en ajoutant l'erreur $err = \max\limits_{k=1,\ldots,n} |f'(x_k) - d_k|$. Sauvegarde sous `VI5.m`. Mêmes éléments test.

7. Etudier l'erreur quand n varie, en partant d'un tableau $tabn = [20 : 10 : 200]$. En particulier, déterminer numériquement l'ordre de la méthode. Sauvegarde sous `VI6.m`. Test avec $a = 0$, $b = 2$ et la fonction $f(x) = \cosh(x)$. On affichera le tableau d'erreur, le graphe de $\log(err)$ en fonction de $\log(n)$ et le calcul de la pente de la droite de régression à l'aide de `polyfit`.

8. *Approximation des dérivées secondes :*

   Soit $f \in C^2([a,b])$. On cherche les valeurs approchées de $f''(x) = (f')'(x)$. Comme précédemment, à partir de $n \in \mathbb{N}$, $n \geq 3$ et d'un échantillonnage uniforme, $\boldsymbol{D}_2$, approximation de $[f''(x_1), \ldots, f''(x_n)]^T$ est donnée en résolvant successivement les systèmes suivants :

$$\boldsymbol{A}\boldsymbol{D}_1 = \frac{1}{h}\boldsymbol{B}Y,\ \boldsymbol{A}\boldsymbol{D}_2 = \frac{1}{h}\boldsymbol{B}\boldsymbol{D}_1$$

   où $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$ sont définies précédemment.

   Ecrire un fichier `VI7.m` qui, étant donné l'entier n, construit les matrices, calcule $\boldsymbol{D}_2$, trace le graphe des dérivées secondes approchées et exactes, et calcule l'erreur $err = \max\limits_{i=1,\ldots,n} |f''(x_i) - d_{2i}|$. Afficher `err`. Test avec $f(x) = \cosh(x)$, $n = 5$, $a = 0$, $b = 2$.

9. Etudier et tracer l'erreur quand n varie en partant d'un tableau `tabn`$= [20 : 10 : 200]$. Donner l'ordre de la méthode en commentaire. Affichage de $\log(err)$ en fonction de $\log(n)$. Sauvegarde sous `VI8.m`. Mêmes éléments test.

Exercice 2 `Logiciel R`

- *Votre code R sera enregistré dans un seul script sous un nom de la forme* `R_NOM_PRENOM.r`.
- *Les questions 1, 2 et 3 sont indépendantes.*

Pour commencer, copier-coller les fichiers `JO2016_NbAthlete.txt` et `JO2016_Medaille.csv` situé dans le répertoire public `X:\GM-345\GM 3A\IntroLogiciels\TPnote` dans votre répertoire de travail.

1. **Jeux Olympiques (JO) de Rio**

   Le jeu de données contenu dans le fichier `JO2016_NbAthlete.txt` porte sur le nombre d'athlètes engagés aux JO par pays. Pour chaque pays, l'enregistrement comprend 3 variables :

   - `Continent` : continent auquel appartient le pays,
   - `Pays` : nom du pays,
   - `NbAthlete` : nombre d'athlètes du pays engagés aux JO de Rio.

   Le jeu de données contenu dans le fichier `JO2016_Medaille.csv` donne le bilan des médailles pour les 15 meilleurs pays. Pour chaque pays, vous disposez de 5 variables :

   - `Rang` : Rang du pays dans le classement,
   - `Pays` : Nom du pays,
   - `Or` : Nombre de médailles d'or obtenues par les athlètes du pays,
   - `Argent` : Nombre de médailles d'argent obtenues par les athlètes du pays,
   - `Bronze` : Nombre de médailles de bronze obtenues par les athlètes du pays.

   (a) Lire les fichiers `JO2016_NbAthlete.txt` et `JO2016_Medaille.csv` et stocker leur contenu dans des data-frames nommés respectivement `athlete` et `medaille`.

   (b) Supprimer du tableau `athlete` les lignes pour lesquelles la variable `Continent` est manquante.

   (c) Tracer le diagramme en bâtons du nombre d'athlètes engagés par continent. Vous y ferez figurer un titre.

   (d) Construire un vecteur `solo` contenant la liste des pays n'ayant engagé qu'un seul athlète.

   (e) Dans le tableau `medaille`, ajouter une colonne nommée `Total` qui donne le nombre total de médailles par pays.

   (f) Fusionner les deux tableaux `athlete` et `medaille` dans un nouveau tableau nommé `tab` où vous ne conserverez que les 15 pays ayant obtenu le plus de médailles.

   (g) On se propose de faire un nouveau classement des 15 pays du tableau `tab` en fonction du nombre de médailles obtenues par athlète engagé. Créer une variable `med_par_ath` donnant le nombre de médailles par athlète engagé, puis trier votre tableau par ordre décroissant de cette variable pour obtenir le nouveau classement.

2. **Test statistique du khi-deux**

   Soit $\mathbf{N} = (n_{ij})$ une matrice réelle non nulle de taille $p \times q$ $(p, q > 1)$. On note :

   - $n_{i\bullet}$ la somme des termes de la $i^{\text{ème}}$ ligne,
   - $n_{\bullet j}$ la somme des termes de la $j^{\text{ème}}$ colonne,
   - n la somme des termes de la matrice.

(a) Écrire une fonction `statistique` qui prend comme argument la matrice `N` et qui retourne la quantité suivante :

$$T = \sum_{i=1}^{p} \sum_{j=1}^{q} \frac{\left(n_{ij} - \frac{n_{i\bullet} n_{\bullet j}}{n}\right)^2}{\frac{n_{i\bullet} n_{\bullet j}}{n}}.$$

La quantité `T` s'appelle la statistique du khi-deux.

(b) Écrire une fonction `test_khi2` qui prend comme arguments une matrice `N` et un réel `r` compris strictement entre 0 et 1 et qui rend la chaîne de caractères `H0` si la statistique `T` est inférieure au quantile d'ordre `r` de la loi du khi-deux à $(p-1) \times (q-1)$ degrés de liberté $\chi^2_{(p-1)(q-1)}$ et la chaîne de caractères `H1` sinon (Vous utiliserez la fonction `qchisq`).

(c) Créer la matrice `E` ci-dessous et lui appliquer la fonction `test_khi2` en prenant l'ordre `r` égal à 0,95.

$$E = \begin{pmatrix} 8 & 8 & 10 & 6 & 11 & 16 \\ 5 & 2 & 11 & 2 & 17 & 23 \end{pmatrix}$$

3. **Racines d'un polynôme de degré 2**

Soit $a \in \mathbb{R}^*$ et $b, c \in \mathbb{R}$. On considère le polynôme de degré 2 $P(X) = aX^2 + bX + c$. Les racines du polynôme P sont les solutions de l'équation $P(X) = 0$.

Écrire une fonction :

- qui prend comme arguments les coefficients `a`, `b` et `c` du polynôme P,
- qui affiche un message pour prévenir l'utilisateur si le polynôme n'admet pas de racines réelles,
- qui rend les racines du polynôme s'il en existe,
- qui trace un graphique du type suivant pour illustrer la position de la (des) racine(s) s'il y en a.