9 novembre 2017

Les objectifs de ce mini-stage sont les suivants :

1. Acquérir la maîtrise de **Scilab**, qui est un outil de calcul numérique.
2. Acquérir la maîtrise de **LaTeX**, qui est un outil de traitement de texte, utilisé par les scientifiques (utile notamment pour les formules mathématiques).
3. Apprendre à rédiger un compte-rendu de TP, à partir de quelques exercices.

Tous les documents pour ce TP sont disponibles sur chamilo dans le dossier :

ENSIMAG3MMAI/Documents/TP Scilab-Latex

La partie 1 de ce sujet vous propose une introduction rapide à Scilab. Prenez le temps de répondre aux questions et de feuilleter le polycopié. Familiarisez-vous ensuite à LaTeX avec la partie 2, le polycopié et le fichier exemple fourni enonce.tex. Enfin, faites les exercices de la partie TP, que vous rédigerez dans un compte-rendu. Consignes pour le compte-rendu :

— Travailler en binôme.
— Rendre une archive .tar.gz par Teide, et contenant le compte-rendu au format pdf ainsi que le code au format sce. Votre compte-rendu donnera lieu à un bonus/malus pour la note d'analyse.
— Ne pas mettre le code dans le compte-rendu.
— Être concis : quelques pages doivent suffire !
— Se relire, soigner les figures et les formules.

Date limite de rendu : 22 novembre 2017

Bon courage à tous, et n'hésitez pas à faire appel aux encadrants aussi souvent que nécessaire !

1 Prise en main de Scilab

1.1 Introduction

Pour lancer le logiciel Scilab il faut taper dans un terminal `scilab &`. Une fenêtre de commandes s'ouvre. Après avoir lancé Scilab, vous pouvez tester les commandes suivantes :

— `help` pour ouvrir l'aide en Scilab,
— `help mot-clé` pour obtenir la description de la fonction `mot-clé`,
— `apropos mot-clé` pour obtenir la liste des pages d'aide contenant `mot-clé`,
— `quit` pour sortir de Scilab.

Question 1 *Donner la description de la commande* `spec`*. Trouver la commande Scilab qui permet de définir la matrice identité.*

Les commandes `clear`, `clc` et `clf` permettent d'effacer respectivement les données mises en mémoire, l'écran de commandes et les figures. Elles doivent être exécutées régulièrement pour éviter les erreurs et libérer la mémoire.

1.2 Exécuter sous Scilab

Les commandes Scilab peuvent être tapées directement en ligne, par exemple :

```
--> A = ones(3,4)
```

ou bien, écrites dans un fichier de commandes "`*.sce`". Nous vous conseillons d'utiliser un éditeur de texte qui offre la coloration syntaxique pour le language Scilab ; par exemple, *SciNotes*, accessible depuis le menu "applications" de Scilab. Pour cela,

1. ouvrir un fichier intitulé par exemple `test.sce`, comportant les lignes suivantes :

   ```
   // Ceci est un commentaire
   clc; clf; clear;
   A = ones(3, 4)
   ```

2. sous Scilab, taper :

   ```
   --> exec("test.sce")
   ```

Si l'on veut définir une fonction, par exemple la fonction *carré*, on peut le faire soit dans notre script `tests.sce`, soit dans un autre fichier, par exemple `carre.sce` :

1. Ouvrir un fichier intitulé par exemple `carre.sce`, comportant les instructions suivantes :

   ```
   //Fonction carre
   function d = carre(x)
      d = x .* x
   endfunction
   ```

2. Sous Scilab, charger le fichier `carre.sce` :

   ```
   --> exec("carre.sce")
   ```

La fonction *carré* est maintenant définie sous Scilab :

```
--> x = [0, 1, 2, 3, 4]
--> carre(x)
```

1.3 Vecteurs et matrices

Scilab est un logiciel de calcul *numérique*, l'essentiel des objets que l'on manipule sont des matrices de nombre.

La façon la plus simple de définir une matrice $n \times m$ en Scilab est d'entrer au clavier la liste de ses éléments, par exemple

```
--> A = [a11, a12, a13; a21, a22, a23]
```

Opérations élémentaires A tester sur des exemples !

```
--> A(k,:)  // k-ième ligne de la matrice A
--> A + B   // somme
--> A * B   // produit
--> A .* B  // produit terme à terme
--> A^2     // équivalent à A * A
--> A.^2    // équivalent à A .* A
--> det(A)  // déterminant de A
--> A'      // transposée de A
--> inv(A)  // inverse de A
```

Question 2 *Entrer sous Scilab la matrice :*

$$A = \begin{pmatrix} 1 & 0 & 3 & 1 \\ 1 & 2 & 0 & 1 \\ 0 & 1 & 3 & 0 \end{pmatrix}.$$

1. *Quelle est la commande donnant la taille de la matrice A ?*
2. *Extraire la première ligne, la dernière colonne et l'élément à la deuxième ligne, troisième colonne de A.*
3. *Extraire la diagonale, les parties triangulaires supérieures et inférieures de A.*

Question 3 (Matrices particulières)

1. *En utilisant les fonctions* `ones` *et* `diag`*, définir la matrice identité* 10×10.
2. *Définir la matrice tridiagonale d'ordre* 10 *suivante en utilisant ces commandes :*

$$C = \begin{pmatrix} 2 & -1 & & \mathbf{0} \\ 1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ \mathbf{0} & & 1 & 2 \end{pmatrix}.$$

1.4 Fonctions

1.4.1 Fonctions échantillonnées (= discrétisées)

Une fonction peut être définie par rapport à une discrétisation de la variable x, ainsi :

```
--> x = 0:0.1:1
```

correspond à une discrétisation par pas de 0.1, de l'intervalle $[0, 1]$, soit 11 valeurs. On définit des fonctions sur cette grille discrète, par exemple :

```
--> y = sin(2 * %pi * x) + cos(%pi * x)  // somme de deux sinusoïdes
--> z = x.^2                             // parabole
```

Remarque : Les fonctions prédéfinies (comme `cos` ou `sin`) agissent terme à terme sur les vecteurs, et même sur les matrices. Ainsi on n'a pas besoin de faire de boucles. En règle générale on évite les boucles autant que possible, car elles ralentissent l'exécution.

1.4.2 Tracé de courbes

Pour tracer une courbe $y = x^2$ sur l'intervalle $[a, b]$:

```
--> n = ...              // nombre de points de discrétisation
--> dx = (b - a)/(n - 1) // pas de la discrétisation
--> x = a:dx:b;          // x est échantillonné entre a et b avec un pas de dx
--> plot(x, x.^2)
```

Remarques :

— On pourra également utiliser la commande `x=linspace(a,b,n)` pour définir la grille discrète.

— La commande `plot` est une fonction de Matlab qui a ensuite été intégrée à Scilab. On peut aussi utiliser la fonction originale de Scilab `plot2d` décrite dans le poly.

Pour varier le trait ou la couleur

```
--> plot(x, f(x), "r-")
```

Dans la chaîne de caractère (troisième argument), on donne une lettre imposant la couleur (`"r"` pour red, `"b"` pour blue, `"g"` pour green...) et un symbole pour le trait (`"-"` pour un trait continu, `"--"` pour des tirets, `"-."` pour des pointillés...). Pour plus de détails, `help LineSpec`.

Pour tracer plusieurs courbes, rajouter un titre, des axes, une légende

```
--> x = 0:0.1:10
--> plot(x, cos(x), "r", x, sin(x), "b")
--> xtitle("Graphe de la fonction sin")  // titre
--> legend("cos","sin")                  // légende
--> xlabel("temps"); ylabel("f(t)");     // noms des axes
```

Pour tracer plusieurs graphes dans une fenêtre

La commande `subplot(n, m, p)` placée avant chaque tracé de courbe, subdivise la fenêtre du graphe en une matrice $n \times m$ de sous-fenêtres et sélectionne la p-ième pour dessiner le graphe courant : l'élément (i, j) de la matrice correspond au graphe numéro $(i-1)n + j$.

Pour créer une nouvelle figure

Taper `figure()` par exemple pour créer une nouvelle fenêtre.

Pour exporter une figure

Dans la fenêtre graphique à exporter, cliquer sur le menu `File`, puis `Export`. Dans la fenêtre qui s'ouvre alors :

1. choisir l'extension du fichier image : choisir de préférence `pdf`,
2. choisir entre `Color` et `Black & White`,
3. choisir l'orientation. Attention, par défaut, l'image est en landscape (format paysage) !
4. entrer le nom du fichier image : `myfig` par exemple, sans l'extension.

Question 4 *Tracer les courbes de la fonction sin sur l'intervalle* $[0, 2\pi]$ *pour 6 points de discrétisation, puis 21 points de discrétisation (en une autre couleur). Ajouter un titre et une légende.*

2 Prise en main de Latex

Un fichier Latex est repérable par son extension ".tex". Un exemple de fichier Latex est donné sur chamilo dans le dossier TP Scilab-Latex.

Editer et compiler un fichier Latex

1. Pour éditer et modifier ce fichier, il suffit d'utiliser votre éditeur de texte préféré, par exemple :

   ```
   $ emacs enonce.tex &
   ```

2. Pour compiler ce fichier, taper la commande :

   ```
   $ pdflatex enonce.tex
   ```

3. Si la compilation s'est exécutée sans erreurs, un fichier "enonce.pdf" a été créé, à visualiser par exemple avec :

   ```
   $ evince enonce.pdf &
   ```

NB : Il est conseillé de garder la fenètre avec le pdf active (en insérant un & à la fin de la commande). Aprés chaque modification puis compilation du fichier tex, le fichier pdf est mis à jour et rafraîchi automatiquement.

Structure d'un fichier Latex

Le langage Latex comprend deux types d'éléments (en plus du texte normal) : des commandes et des environnements.

Une **commande** est identifiable par le préfixe \, et peut comprendre des paramètres indiqués entre **{ }** et des options indiquées entre **[ ]**.

Un **environnement** se déclare à l'aide de deux commandes `\begin` et `\end`. Le contenu compris entre ces deux balises aura une mise en forme, ou comprendra des commandes spécifiques à l'environnement.

La première ligne d'un fichier Latex est toujours la déclaration de sa classe.

Question 5 *Repérez la classe à laquelle appartient cet énoncé. Changez cette classe pour* `book` *et observez les changements sur le document.*

S'en suivent, dans un ordre quelconque :
- import des packages utilisés (encodage du document, langue, environnements spécifiques...)
- définition personnalisée de commandes, d'environnements
- informations relatives au document : date, auteur, titre

Tous ces éléments donnent des informations importantes pour la compilation du fichier, mais ne vont rien afficher directement sur le document pdf généré.

Contenu du document
C'est l'environnement `document` qui va générer le contenu du document pdf. Dans cet environnement, en plus du texte classique, on peut intégrer (entre autre) des commandes spécifiant :
- la structure de document (sections)
- l'organisation et la mise en forme du texte
- l'affichage et le référencement de figures (contenant par exemple des images, ou des tableaux)
- du contenu spécialisé (par exemple mathématiques, chimie ou encore musique)

Certaines commandes permettent aussi de générer automatiquement un titre, une table des matières, ou une page de références bibliographique.

Question 6 *Essayez d'inclure une image dans le document .tex. Pour cela on utilisera la commande* `\includegraphics` *dans l'environnement* `figure`.

Ecrire des mathématiques
Il existe deux principales manières d'inclure du contenu mathématique dans un document Latex : au milieu de texte, en encadrant le contenu mathématique par des balises $, ou alors sur une ligne séparée avec l'environnement `equation`.

Mise en page
La mise en page du fichier se fait de manière automatique. En particulier les sections que vous aurez indiqué seront automatiquement numérotées, de même que les figures, les équations et théorèmes. Le placement des figure est aussi décidé par Latex, ce qui peut donner parfois des résultats inattendus. Deux solutions pour remédier à une figure éloignée du texte qui lui fait référence :
- ajouter l'option `[h]` ou `[h!]` à l'environnement `figure` pour forcer la position de la figure par rapport au texte
- laisser la figure se placer à un endroit différent du document, et ajouter une référence que l'on pourra citer dans le texte. Pour cela, il faut utiliser la commande `\label` dans l'environnement `figure`, pour associer un mot clé à votre figure. Dans votre texte, vous pouvez faire référence à cette figure en utilisant la commande `\ref` suivi du mot clé que vous lui avez attribué. Ceci fonctionne aussi pour tout ce qui est numéroté dans la structure générale du document : sections, équations, théorèmes..

Vous pouvez maintenant passer à la partie suivante du TP, et utiliser le poly Latex au fur et à mesure des besoins que vous aurez.

3 Travaux pratiques - Compte-rendu

Les exercices de cette partie sont à rédiger dans le compte-rendu Latex. Pour vous aider dans l'écriture des formules mathématiques, vous pouvez récupérer le source du TP (le .tex) sur la page web du stage.

3.1 Sensibilisation à l'arithmétique machine

Scilab, comme la plupart de langages de calcul, utilise la norme IEEE-754 pour le stockage des nombres en mémoire. Sans rentrer dans les détails, un nombre réel est représenté par un nombre flottant

$$x = 0.b_1 b_2 \dots b_m \, 10^e.$$

La partie $0.b_1b_2\dots b_m$ s'appelle la mantisse, e est l'exposant. Comme la mantisse n'a qu'un nombre m fixé de chiffres significatifs, Scilab ne pourra pas distinguer 2 réels au-delà de cette précision. A titre indicatif, en Scilab la mantisse est codée sur 52 bits, l'exposant sur 11 bits et le signe sur 1, ce qui fait qu'un réel occupe 64 bits (8 octets) en mémoire.

En Scilab, on ne peut donc pas calculer un nombre avec une précision arbitraire. La constante `%eps` est la distance entre le nombre 1 et le flottant machine qui lui est immédiatement supérieur : on l'appelle le zéro machine, il vaut $2^{-52} \approx 2.22 * 10^{-16}$. Pour mieux comprendre, essayez les commandes suivantes :

```
--> format(20)
--> 1 + %eps
--> 1 + 0.5 * %eps
```

Exercice 1 *Exécuter dans Scilab les commandes suivantes. Expliquer et commenter les résultats.*

```
--> x = 1e30
--> y = 1e-8
--> z = ((y + x) - x) / y
--> w = (y + (x - x)) / y
```

Exercice 2 *On considère la fonction f définie de la manière suivante : pour $x \in [0,4]$ on calcule*

— $y = \sqrt{\sqrt{\sqrt{\dots\sqrt{x}}}}$ *(128 fois)*

— *puis* $f(x) = ((\dots((y^2)^2)^2\dots)^2)^2$ *(128 fois)*

Tracer la courbe de f en fonction de x. Que constatez-vous ? Expliquer le résultat.

Exercice 3 *On cherche à calculer la valeur de l'intégrale*

$$I_{20} = \int_0^1 x^{20} e^x dx$$

1. *Donner une formule de récurrence pour calculer cette intégrale, puis évaluer I_{20} à partir de I_0*
2. *Évaluer I_{20} en utilisant le développement en série de e^x.*
3. *Conclusions ?*

Exercice 4 *Ici, on va calculer directement une approximation de I_{20} en utilisant la méthode des rectangles. Écrire une fonction* `I = rectangle(n)` *qui calcule I_{20} en utilisant n points. Vérifier que, pour n suffisamment grand, on retrouve bien le même résultat qu'à la question 3.2.*

3.2 Étude du phénomène de Gibbs

Le phénomène de Gibbs est un phénomène oscillatoire qui s'observe graphiquement quand on approche une fonction par sa série de Fourier. L'exercice qui suit vous fait d'abord calculer la série de Fourier d'une fonction "simple", puis tracer cette série tronquée aux premiers termes.

Rappel sur les séries de Fourier Soit $f : \mathbb{R} \mapsto \mathbb{C}$ une fonction T periodique, la série de Fourier de f s'écrit comme une combinaison linéaire de fonctions sinusoïdales :

$$f(t) = \sum_{n=-\infty}^{\infty} c_n(f) \exp\left(2i\pi \frac{n}{T} t\right), \tag{1}$$

où les coefficients $c_n(f)$ de f sont appelé *coefficients de Fourier* et sont définis comme suit :

$$c_n(f) = \frac{1}{T} \int_{-T/2}^{T/2} f(t) \exp\left(-2i\pi \frac{n}{T} t\right) dt. \tag{2}$$

Dans le cas de fonctions $f : \mathbb{R} \mapsto \mathbb{R}$, quelques simplifications peuvent être faites, la série de Fourier peut s'écrire comme suit :

$$f(t) = a_0(f) + \sum_{n=1}^{\infty} a_n(f) \cos\left(2\pi \frac{n}{T} t\right) + b_n(f) \sin\left(2\pi \frac{n}{T} t\right), \tag{3}$$

où les coefficients $a_n(f)$, sont définis comme suit :

$$\begin{cases} a_0(f) = \dfrac{1}{T} \displaystyle\int_{-T/2}^{T/2} f(t) dt \\ a_n(f) = \dfrac{2}{T} \displaystyle\int_{-T/2}^{T/2} f(t) \cos\left(2\pi \frac{n}{T} t\right) dt \\ b_n(f) = \dfrac{2}{T} \displaystyle\int_{-T/2}^{T/2} f(t) \sin\left(2\pi \frac{n}{T} t\right) dt \end{cases}. \tag{4}$$

Exercice 5 *(théorique) Soit la fonction 1-périodique f, définie sur $[-\frac{1}{2}, \frac{1}{2}[$ par*

$$f(x) = \begin{cases} -1 & \textit{pour} \quad -\frac{1}{2} \leq x \leq 0, \\ 1 & \textit{pour} \quad 0 < x < \frac{1}{2}. \end{cases} \tag{5}$$

Calculer la série de Fourier de f et expliquer en quels points x la formule ci-dessous est valide :

$$f(x) = \frac{4}{\pi} \sum_{n=0}^{+\infty} \frac{\sin(2(2n+1)\pi x)}{2n+1}. \tag{6}$$

Exercice 6 *(pratique) Écrire un programme Scilab traçant la série de Fourier tronquée de la fonction f (formule (5)). Concrètement, reprendre la formule (6) et calculer la somme partielle des* `(Ntermes+1)`*-premiers termes, pour une valeur de* `Ntermes` *entrée au clavier :*

```
Ntermes = input('Entrer le nombre de termes')
```

Pour la boucle de contrôle :

```
for i = 0:Ntermes
    S = S + ...
end
```

Quel phénomène observez-vous sur le graphe ? Comment varie-t-il avec `Ntermes` *? Que pouvez-vous en conclure sur la convergence de la série de Fourier ?*

3.3 Théorème de Gerschgörin

(petit) Rappel sur les valeurs propres et vecteurs propres Soit A une matrice, les valeurs propres λ et les vecteurs propres v de A respectent l'égalité :

$$(A - \lambda I)v, \tag{7}$$

où I est la matrice identité.

Exercice 7 *Dans cet exercice, nous nous intéressons au théorème suivant :*

Soit A une matrice carrée d'ordre N. Les valeurs propres de A appartiennent à l'union des N disques D_k du plan complexe, soit : $\lambda \in \bigcup_{k=1}^{N} D_k$, où D_k, appelé disque de Gerschgörin est défini par :

$$D_k = \left\{ z \in \mathbb{C} : |z - a_{kk}| \leq \Lambda_k = \sum_{j=1, j\neq k}^{N} |a_{kj}| \right\}.$$

1. *Démontrer ce théorème.*
2. *Écrire un programme Scilab permettant de visualiser les disques de Gerschgörin dans le plan complexe (pour tracer un cercle tracer deux demi-cercles en utilisant la commande* `plot2d`*).*
3. *Vérifier que les valeurs propres de la matrice A suivante sont toutes dans l'union des disques de Gerschgörin de A :*

$$A = \begin{pmatrix} 1+i & i & 2 \\ 3 & 2+i & 1 \\ 1 & i & 6 \end{pmatrix}.$$

4. *On appelle matrice à diagonale strictement dominante, une matrice telle que*

$$\forall i \quad \sum_{k \neq i} |a_{ik}| < |a_{ii}|.$$

Démontrer, en utilisant les disques de Gerschgörin que toute matrice à diagonale strictement dominante est inversible.