MT94/P26/TD0 - Initiation à Scilab

Utilisez le livret "Initiation à Scilab" qui vous a été distribué lorsque vous aurez besoin d'aide technique sur un sujet particulier (vous pouvez aussi utiliser les commandes `doc` et `apropos`). La séance commencera avec un Scilab "classique" sur votre machine perso, mais à un moment adéquat nous basculerons sur le JupyterHub de MT94 :

`https://jupyterhub-mt94.utc.fr`

Le login est celui que vous utilisez habituellement pour vous identifier sur les différents services de l'UTC, par contre le mot de passe est choisi à la première connexion (n'utilisez pas votre mot de passe habituel).

1 Algèbre linéaire

1. Résoudre avec Scilab le système linéaire

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= -1, \\ 4x_1 + 5x_2 + 6x_3 &= 0, \\ 7x_1 - 8x_2 &= 2, \end{aligned}$$

   en l'écrivant tout d'abord sous la forme $Ax = b$. Vérifier que $x = A^{-1}b$ donne un résultat numériquement différent. On pourra aussi vérifier que ni AA^{-1} ni $A^{-1}A$ ne donnent exactement la matrice identité. Quelle est la méthode la plus rapide pour obtenir la solution du système d'équations (donner la complexité en fonction du nombre d'inconnues) ?
2. Calculer les valeurs propres et les vecteurs propres de la matrice A (question précédente) en utilisant la fonction `spec`. Vérifier que les égalités $A = PDP^{-1}$ et $D = P^{-1}AP$ sont vérifiées à la précision machine. Recommencer avec les matrices $A + A^\top$, $A^\top A$ et vérifiez certains résultats attendus.

2 Polynômes

1. Calculer les racines du polynôme (fonctions `poly`, `roots`)

$$p_5(z) = z^5 - 1$$

2. Calculer les valeurs de $p_5(x_n)$ pour $x_n = n/10$, $n = 0 \dots 10$ (fonction `horner`).

3 Variables aléatoires

1. Générer 1000 réalisations d'une variable aléatoire uniforme sur $[0,1]$ (fonction `rand`). Vérifiez en augmentant n que la moyenne empirique (fonction `mean`) converge vers la bonne valeur.
2. Générer 1000 réalisations d'une variable aléatoire normale centrée et réduite (fonction `rand`). Vérifiez en augmentant n que la moyenne empirique (fonction `mean`) et l'écart-type empirique (fonction `std`) convergent vers les bonnes valeurs.
3. Ecrire quelques lignes de Scilab permettant d'estimer π avec la méthode de Monte Carlo.

4 Scripts et fonctions

1. On considère la suite récurrente (u_n) définie par $u_0 = u_1 = 1$ et pour tout n

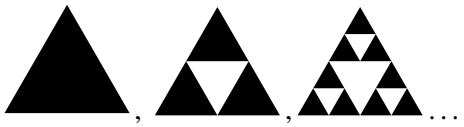
$$u_{n+2} = u_{n+1} + u_n.$$

   Ecrire un programme Scilab montrant que u_{n+1}/u_n converge vers le nombre d'or.
2. Ecrire une fonction Scilab calculant $n!$ de manière récursive.

5 Courbes et surfaces

1. Représenter les graphes des fonctions $x \to x^n$ pour x $[0,1]$ et n=$1,2,3,4,5$ sur la même figure.
2. Représentez dans $\mathbb{R}^3$ les graphes des fonctions de deux variables suivantes, définies sur $[-1,1]\times[-1,1]$ (fonctions `meshgrid`, `surf`) :
   (a) $(x,y) \to x^2+y^2$
   (b) $(x,y) \to x^2-y^2$
   (c) $(x,y) \to |p_5(x+iy)|$ où p_5 a été défini à la section 2.
3. Représentez la sphère unité en utilisant son paramétrage en coordonnées sphériques
4. Représentez un tore (`https://fr.wikipedia.org/wiki/Tore`)

6 Triangle de Sierpinski



On considère le domaine T_0 de $\mathbb{R}^2$ constitué par l'intérieur (ici rempli en noir) d'un triangle équilatéral dont la base est parallèle à l'axe de abscisses, par exemple le triangle (A,B,C) où $A(0,0)$, $B(1,0)$, $C(0.5,\sqrt{3}/2)$. On note $T_1 = F(T_0)$ le domaine obtenu en privant T_0 de l'intérieur du triangle (D,E,F) où D, E et F sont les milieux respectifs des segments AB, BC, AC. Ensuite, pour tout n, T_{n+1} est le domaine obtenu en appliquant F à tous les triangles pleins de T_n. Par définition, le triangle de Sierpinski est défini comme

$$\mathscr{T} = \lim_{n\to\infty} T_n$$

Cet objet présente des tas de propriété bizarres, je vous laisse vous documenter sur le sujet ! Bien évidemment on ne pourra représenter que T_n pour une valeur finie de n que l'on choisira raisonnable au regard de nos possibilités de représentation dans Scilab. Votre travail est surtout de déterminer T_n sous la forme de tous les sous-ensembles de 3 points constituant chaque triangle plein à représenter. On vous aidera à question 3. pour la partie graphique.

1. Ecrire une fonction Scilab de la forme

   ```
   function triangles = F(A,B,C,n)
   ...
   endfunction
   ```

   acceptant en entrée les coordonnées de trois points dans les vecteurs colonnes `A,B,C` et renvoyant dans `triangles` une matrice à deux lignes contenant sur chaque colonne les coordonnées (x,y) des points successifs de tous les triangles pleins de T_n.
2. Après avoir vérifié que `F(A,B,C,n)` renvoie bien ce qu'il faut pour $n=0$ et $n=1$, utilisez une valeur de n plus grande, mais au préalable faites un petit calcul pour estimer la taille des triangles en fonction de n et en déduire jusqu'où ne pas aller trop loin !
3. Ajoutez en dessous de votre programme le code Scilab suivant qui vous permettra de visualiser vos triangles (vous pouvez le copier-coller directement à partir du pdf). Ici on suppose que `tri` contient le résultat de l'appel à votre fonction `F()` :

   ```
   clf
   x=matrix(tri(1,:),3,-1);
   y=matrix(tri(2,:),3,-1);
   plot3d(x,y,0*x);
   set(gca(),"view","2d")
   ```

   Ne vous étonnez pas si on utilise la 3d pour visualiser nos triangles ici cela va plus vite comme ça en Scilab !