

Travaux dirigés sur ordinateur n°3 :
Introduction à Scilab (suite 2) (↔ Q 2)

Question 1 : Écrire ses propres fonctions scilab Scilab permet de définir de nouvelles fonctions utilisateurs, La syntaxe est la suivante :

```
fonction [<arguments de retour>]=<nom_de_la_fonction>(<arguments d'entrée>),  
.....instructions.....  
endfonction
```

Ensuite pour appeler la fonction, il suffit de taper le nom de la fonction avec entre parenthèses les arguments désirés.

Regarder attentivement puis tester l'exemple suivant

```
fonction y=f(x)  
    y=x^2-1  
endfonction  
f(2)  
z=f(100)
```

une fonction ne modifie pas les valeurs des variables définies en dehors de la fonction.

1. Écrire une fonction SCILAB `g` qui calcule la fonction numérique définie par $g(t) = \frac{\ln t}{1+t^2}$. La tester.
2. Écrire une fonction SCILAB `norme` qui calcule la norme euclidienne d'un élément de \mathbb{R}^2 , elle prendra en argument un vecteur ayant deux coordonnées et renverra sa norme. La tester.
3. Écrire une fonction SCILAB `polaire` qui calcule les coordonnées polaires d'un élément de $\mathbb{R}^2 \setminus (\{0\} \times \mathbb{R}^-)$ elle prendra en argument un vecteur ayant deux coordonnées (x, y) et renverra le vecteur

$$\left(\sqrt{x^2 + y^2}, 2\text{Arctan} \left(\frac{y}{x + \sqrt{x^2 + y^2}} \right) \right)$$

La tester.

4. Écrire une fonction SCILAB `fact` qui calcule les factorielles, elle prendra en argument un entier n et renverra la valeur de $n!$. Pour cela on utilisera dans la définition de la fonction une boucle `for`. La tester.
5. Écrire une fonction SCILAB `facto` qui calcule les factorielles de façon récursive, dans les instructions de la fonction on utilisera un `if` et on utilisera le fait que $0! = 1$ et que $\sinon n! = (n - 1)! * n$. La tester.

Question 2 :

1. Aller sur la page personnel d'Alexandre Mizrahi (par exemple taper alexandre mizrahi dans google.fr), ou aller sur le site de l'UCP puis **établissement** puis **pages personnelles**.
2. Aller dans **Des TP en Scilab**, ouvrir le fichier `exemple.sce` du TP3
3. Sélectionnez le (ctrl A), copiez le (ctrl C), puis dans une fenêtre de l'éditeur scilab, collez le (ctrl V), sauvez ce fichier dans votre répertoire (Documents). Vous pouvez l'utiliser. On aurait aussi pu sauvegarder le fichier dans son répertoire et venir le lire avec l'éditeur de scilab.
4. Répondre aux questions suivantes, concernant ce petit fichier :
 - (a) Que fait `f(3)` que fait `g(3)` ?
 - (b) Pourquoi `g([4, 7])` n'est pas défini, mais `f([4, 7])` l'est ?
 - (c) Quelle est la différence entre `f([4, 7; 1, 3])` et `g([4, 7; 1, 3])` ?
 - (d) Combien de nombre se trouve dans la matrice ligne `U` ?

- (e) Que contiennent les variables V et W ?
- (f) Que fait la commande `plot2d(U, V)` ?
- (g) Comment tracer une approximation de la dérivée de f à l'aide de W ?
- (h) Que contient la variable T ?

Question 3 : Dichotomie Il s'agit d'écrire une fonction scilab qui prend 4 arguments : a, b, f, e , où f est une fonction définie sur $[a, b]$, et e (e pour erreur) est un réel positif et qui renvoie un réel se trouvant à une distance inférieur à e d'une racine de f . On commence par vérifier que $f(a)$ et $f(b)$ sont de signes différents, ensuite on calcule le nombres d'itérations nécessaires puis on utilise une boucle for...
Tester le programme avec $f(x) = x^2 - 1$, $a = 0$, $b = 2$, et $e = 10^{-6}$.