Fiche 4 : Définitions de fonctions

I Fonction build in (Rappel)

Il existe un certain nombre de fonctions déjà définies dans Scilab. Nous allons revoir les fonctions classiques vues en cours dans l'année. On suppose par la suite qu'on a assigné une valeur à la variable x.

Définition 1 (Puissance)**.**

Pour obtenir x^n, on tape `x^n`.

Définition 2 (Racine carrée)**.**

Pour obtenir $\sqrt{x}$, on tape `sqrt(x)`.

Définition 3 (Valeur absolue)**.**

Pour obtenir $|x|$, on tape `abs(x)`.

Définition 4 (Partie entière)**.**

Pour obtenir $\lfloor x \rfloor$, on tape `floor(x)`.

Définition 5 (Logarithme)**.**

Pour obtenir $\ln(x)$, on tape `log(x)`.

Définition 6 (Exponentielle)**.**

Pour obtenir e^x, on tape `exp(x)`.

II Définir une fonction

Dans Scilab, il est également possible de définir une fonction à partir d'instructions (de commandes, de boucles et/ou des fonctions build in).

Une fois définies, on utilisera ces fonctions de la même façon que les fonctions prédéfinies.

Définition 7. Pour définir une fonction, il faut tout d'abord lui donner un NOM, attention, ce nom doit être différent des commandes classiques de Scilab. Par exemple, appelons notre fonction f.

Il faudra alors définir la valeur de $f(x)$ à partir de la valeur de x, pour ce faire, il y a pleins de façons. Mais généralement, on utilisera une formule à partir des fonctions build in.

```
function y= NOM(x)
    Instructions
endfunction
```

ATTENTION. N'oubliez pas le `endfunction` !

ATTENTION. Dans un contexte plus professionnel, si vous êtes amenés à programmer, je vous déconseille d'appeler les fonctions f. Donnez leur des noms plus évocateurs qui donnent plus d'informations sur ce qu'elles font, même si le nom est plus long. Cela vous prendra un peu plus de temps pour écrire, mais à partir de la $15e$ fonction vous serez heureux de ne pas avoir tout appelé avec des lettres uniques.

Exemple. On souhaite définir la fonction f qui à un réel x associe le réel $e^{-x} + \ln(1+x)$.

La commande pour définir cette fonction est alors

```
function y=f(x)
    y=exp(-x)+log(1+x)
endfunction
```

Méthode. Vous pouvez alors demander à Scilab de vous renvoyer la valeur de f en un point donné `t` en tapant : `f(t)`.

ATTENTION. Je ne peux que vous conseiller de vérifier que vous avez bien codé votre fonction en essayant certaines valeurs clés.

Exemple. Par exemple, en tapant `f(0)`, Scilab doit renvoyer 1.

Méthode. Vous pouvez alors tracer le graphe de la fonction ainsi définie avec la commande plot2d.

Exemple. Pour tracer le graphe de la courbe f entre 0 et 1,

```
x=[0 :0.01 :1]
y=f(x)
plot2d(x,y)
```

remarque. On peut écrire un programme pour calculer la valeur de la fonction. La valeur de la fonction $f(x)$ est la valeur que vaut y lorsque le programme arrive à endfunction

Exemple. On veut rédiger un programme qui définit la fonction sumgeom qui pour un x renvoie la somme partielle de la série géométrique : $\sum_{k=0}^{40} x^k$.

```
n=40
function y=sumgeom(x)
    y=1
    for k=1 :n
        y=y+x^k
    end
endfunction
```

remarque. On peut également définir une fonction sur des entiers.

Exemple. On veut définir la fonction qui a un entier n associe la valeur $n!$

```
function y=factorielle(n)
    y=1
    for k=1 :n
        y=y*k
    end
endfunction
```

EXERCICE 1. Rédigez un programme qui définit la fonction sumexp qui pour un x renvoie la somme partielle de la série exponentielle : $\sum_{k=0}^{40} \frac{x^k}{k!}$.

Vous pouvez utiliser la fonction factorielle définie précédemment.

III Algorithme de dichotomie

Soit une fonction f admettant une racine α ($f(\alpha) = 0$).

Le but de la recherche par dichotomie est de trouver une valeur approchée de α.

Pour ce faire, on considère deux réels a et b, tels que $f(a)$ et $f(b)$ sont de signes opposés. On peut supposer pour simplifier que $f(a) < 0$ et $f(b) > 0$.

Si f est continue, par le théorème des valeurs intermédiaires, f s'annule entre a et b.

Pour simplifier, on considère que f ne s'annule qu'en α entre a et b.

On regarde alors le réel m au milieu entre a et b, on a $m = \frac{a+b}{2}$.

Alors, si $f(m) > 0$, alors, par le théorème des valeurs intermédiaires, on trouve que α est entre a et m, sinon α est entre m et b.

On peut alors reitérer l'algorithme sur le segment contenant α.

L'intérêt est qu'on a alors divisé par deux la taille du segment. En itérant l'algorithme, on va alors réduire le segment autour de α et on aboutira à une valeur approchée de α.

Pour résumer, pour une fonction f définie sous Scilab et des réels a et b tels que $f(a)$ et $f(b)$ soient de signes contraires, on aboutit à l'algorithme suivant.

```
while abs(b-a)>10^(-6)
    m=(a+b)/2
    if f(m)f(a)>0 then
        a=m
        else
        b=m
    end
end
disp(m)
```

Cet algorithme fournit une valeur approchée de f à 10^{-6} près.

EXERCICE 2. Reprenez le DM de février et refaites la partie III en utilisant la commande function.