



TP 6

Fichiers de fonction (Matlab)



Il existe de nombreuses fonctions prédéfinies en Matlab, mais il arrivera forcément un moment où vous voudrez utiliser une fonction qui n'est pas définie. Heureusement, il est possible de définir ses propres fonctions et de s'en servir exactement comme les fonctions préexistantes.

function [paramètres de sorties] = nom_fonction (paramètres d'entrées)

des opérations

endfunction

Ces fonctions peuvent prendre divers types d'arguments (scalaire, matrice, etc...) et peuvent réaliser des opérations mathématiques plus ou moins complexes.

Exemple 01

// définition de la fonction polyf

function résultat=polyf(x)

 résultat=x^3-3*x^2+1

endfunction

Remarque : l'enregistrement du script il prend le même nom de la fonction comme dans notre exemple *polyf.m*

Application : Dans la fenêtre de commande (Matlab), on taper polyf(5) pour avoir le résultat

```
Console Scilab 2023.1.0
--> exec polyf.sce ← Ecrire exec polyf.sce c'est pour définir polyf
--> // definition de la fonction polyf
--> function résultat=polyf(x)
-->     résultat=x^3-3*x^2+1
--> endfunction
--> polyf(5) ← Pour avoir le résultat de la fonction
ans =
51.
```

Exemple 02

Soit la fonction suivante : $z = x^3 + 2x - y^2 + \sqrt{y}$, avec l'éditeur de Matlab, créer un fichier nommé fooot.m

The screenshot shows the MATLAB environment with two editor windows and a command window.

Editor - C:\Users\hadef\Documents\MATLAB\fooot.m

```
function [z]=fooot(x,y)
z=x^3+2*x^2-2*y^2+sqrt(y)
disp(z)
end
```

Editor - C:\Users\hadef\Documents\MATLAB\fooot1.m

```
x= input('entrer la valeur de x= ');
y= input('entrer la valeur de y= ');
[z]=fooot(x,y)
```

Command Window

```
>> fooot1
entrer la valeur de x= 2
entrer la valeur de y= 3
-0.2679

z =
-0.2679
```

A green arrow points from the line `[z]=fooot(x,y)` in the `fooot1.m` editor to the line `[z]=fooot(x,y)` in the command window output. Another green arrow points from the line `z=x^3+2*x^2-2*y^2+sqrt(y)` in the `fooot.m` editor to the value `-0.2679` in the command window output.

Exemple 03

Résoudre l'équation second ordre $ax^2 + bx + c = 0$ dans les cas : $\Delta > 0$, $\Delta = 0$ et $\Delta < 0$ sous Scilab en utilisant les fichiers de fonction.

The image displays three MATLAB code editors side-by-side:

- Left Editor:** Shows the main script `equ2ordre.m` with the following code:

```
1 //clc;clear;
2 //Résolution de l'équation de 2eme ordres
3 disp('entrer la valeur de a, b, c')
4 a=input('entrer la valeur de a=')
5 b=input('entrer la valeur de b=')
6 c=input('entrer la valeur de c=')
7 delta=b^2-(4*a*c);
8 if delta>0 then
9     ...//x1=(-b+sqrt(delta))/2*a
10    ...//x2=(-b-sqrt(delta))/2*a
11    [x1,x2]=solution1(a,b,delta);
12    disp(x1)
13    disp(x2)
14 elseif delta==0
15     ...//x0=(-b^2)/(2*a)
16     [x0]=solution2(a,b);
17     disp(x0)
18 else
19     ...disp('pas de solution')
20 end
21
```
- Middle Editor:** Shows the function `solution1.m` with the following code:

```
1 function [x1,x2]=solution1(a,b,delta)
2     x1=(-b+sqrt(delta))/2*a;
3     x2=(-b-sqrt(delta))/2*a;
4 endfunction
5
```
- Right Editor:** Shows the function `solution2.m` with the following code:

```
1 function [x0]=solution2(a,b)
2     x0=(-b^2)/(2*a);
3 endfunction
4
```