

Nom: SAH  
Matricule: X140

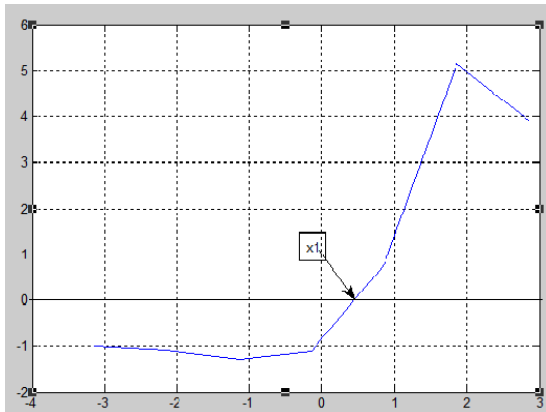
Prénom: DIHIA  
Groupe: 05

### Exercice1:

Séparons les racines des équations suivantes :

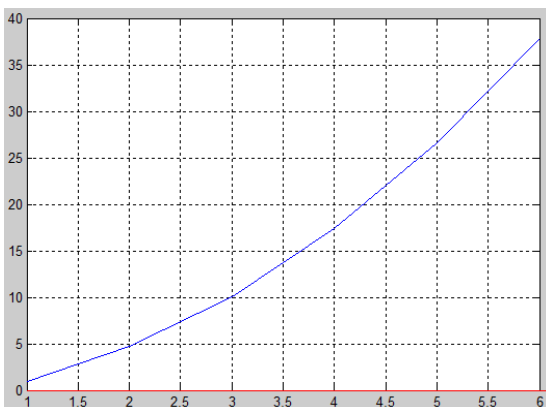
1)  $e^x \sin(x) - 1 = 0$  dans  $[-\pi, \pi]$

```
>> x=[-pi:pi];  
>> f=inline('exp(x).*sin(x)-1','x')  
f =  
    Inline function:  
    f(x) = exp(x).*sin(x)-1  
>> plot(x,f(x))  
>> grid
```



2)  $x^2 + \log(x) = 0$  dans  $\mathbb{R}^+$

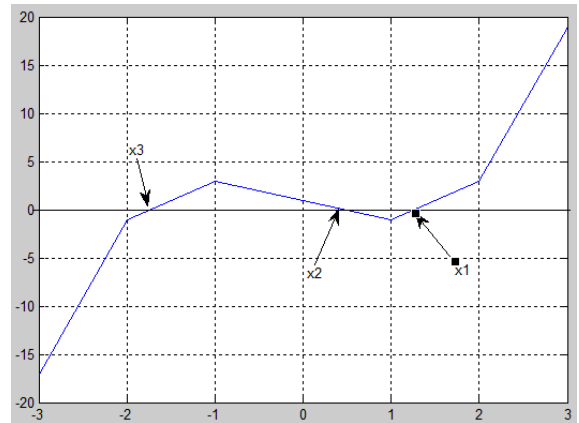
```
>> x=[1:6];  
>> f=inline('x.^2+log(x)','x')  
f =  
    Inline function:  
    f(x) = x.^2+log(x)  
>> plot(x,f(x))  
>> grid
```



3)  $x^3 - 3x + 1 = 0$  dans  $[-3, 3]$

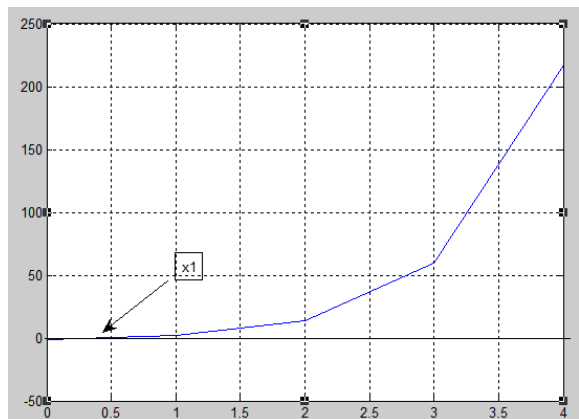
```
>> x=[-3:3];  
>> f=inline('x.^3-3.*x+1','x')  
f =  
    Inline function:  
    f(x) = x.^3-3.*x+1  
>> plot(x,f(x))  
>> grid
```

## Série N°2 Analyse Numérique (TP)



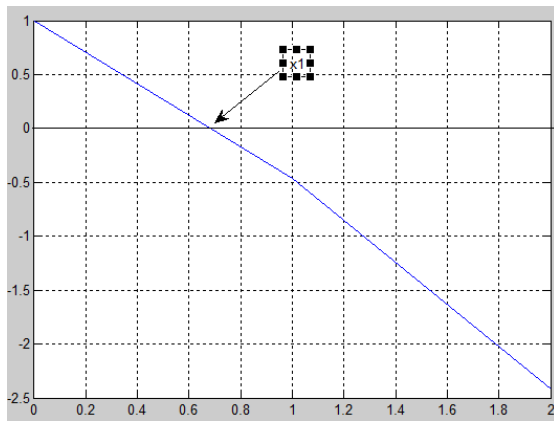
4)  $xe^x - 1 = 0$  dans  $[0, 4]$

```
>> x=[0:4];  
>> f=inline('x.*exp(x)-1','x')  
f =  
    Inline function:  
    f(x) = x.*exp(x)-1  
>> plot(x,f(x))  
>> grid
```



5)  $\cos(x) - x = 0$  dans  $[0, 2]$

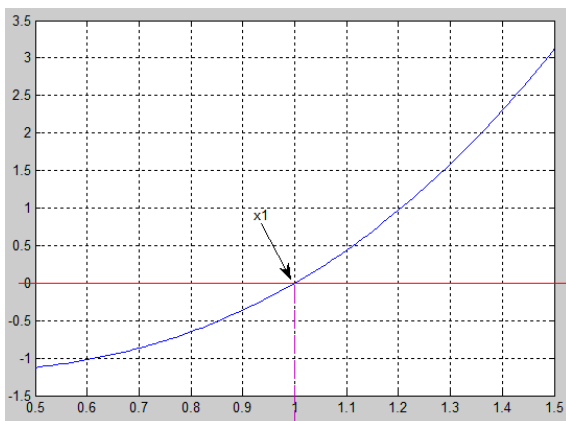
```
>> x=[0:2];  
>> f=inline('cos(x)-x','x')  
f =  
    Inline function:  
    f(x) = cos(x)-x  
>> plot(x,f(x))  
>> grid
```



## Exercice2 :

1) Localisation des racines de l'équation  $f(x)$  et vérifions qu'elle admet une racine séparée dans  $S \in [1/2, 3/2]$

```
>> x=[1/2:0.01:3/2];
>> f=inline('x.^3+x.^2-x-1','x')
f =
    Inline function:
    f(x) = x.^3+x.^2-x-1
>> plot(x,f(x))
>> grid
```



2) Le programme qui donne une solution approximée de S par la méthode de Dichotomie : (M-file)

```
1 a=1/2;
2 b=3/2;
3 f=inline('x.^3+x.^2-x-1','x');
4 a(1)=a; b(1)=b;
5 for i=1:n
6     x(i) = ((a(i)+b(i))/2);
7     if f(x(i)) == 0
8         break
9     elseif f(x(i)) * f(a(i)) < 0
10        a(i+1)=a(i);
11        b(i+1)=x(i);
12    else
13        a(i+1)=x(i);
14        b(i+1)=b(i);
15    end
16 end
17 x
```

3) L'approximation de S à  $0.5 \times 10^{-5}$  près :

```
1 clc,clear all;
2 a=1/2;
3 b=3/2;
4 e=0.5*10^(-5);
5 f=inline('x.^3+x.^2-x-1','x')
6 function x= dichotomie(a,b,f,e)
7     a(1)=a; b(1)=b;
8     n=fix(log((b-a)/e)/log(2))+1;
9     for i=1:n
10        x(i) = ((a(i)+b(i))/2);
11        if f(x(i)) == 0
12            break
13        elseif f(x(i)) * f(a(i)) < 0
14            a(i+1)=a(i);
15            b(i+1)=x(i);
16        else
17            a(i+1)=x(i);
18            b(i+1)=b(i);
19        end
20    end
21    x
```

4) On écritons le programme précédent sous forme d'une fonction Matlab:

Dans M-file :

```
1 function x= dichotomie(a,b,f,e)
2     a(1)=a; b(1)=b;
3     n=fix(log((b-a)/e)/log(2))+1;
4     for i=1:n
5         x(i) = ((a(i)+b(i))/2);
6         if f(x(i)) == 0
7             break
8         elseif f(x(i)) * f(a(i)) < 0
9             a(i+1)=a(i);
10            b(i+1)=x(i);
11        else
12            a(i+1)=x(i);
13            b(i+1)=b(i);
14        end
15    end
```

Sur Matlab :

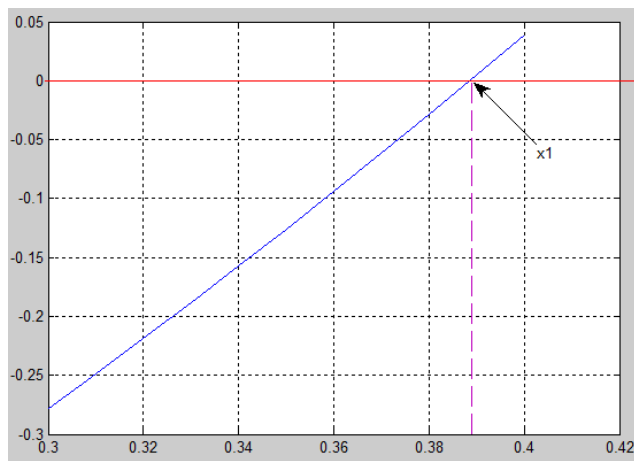
```
Command Window
>> x=dichotomie(1/2,3/2,inline('x.^3+x.^2-x-1','x'),0.5*10^(-5))

x =

    1
```

## Exercice3 :

1) Montrons graphiquement que  $f(x)=0$  admet une racines positive  $S \in [0.3,0.4]$  :



|        |        |
|--------|--------|
| 0.4000 | 0.3887 |
|--------|--------|

2)Le programme qui donne une solution approximée de S par la méthode de Newton :

Dans un M-file :

```

1 - clc,clear all
2 - x0=0.4;
3 - n=10;
4 - x=[0.3:0.01:0.4];
5 - f=inline('x.^2.*exp(x)+2.*x-1','x');
6 - df=inline('(x.^2+2.*x).*exp(x)+2','x');
7 - x(1)=x0;
8 - for i=1:n
9 -     x(i+1)=x(i)-f(x(i))/df(x(i));
10 - end;
11 - x

```

Sur Matlab :

|                                           |
|-------------------------------------------|
| x =                                       |
| 0.4000 0.3887 0.3886 0.3886 0.3886 0.3886 |
| 0.3886 0.3886 0.3886 0.3886 0.3886        |

3)Approchons la racine S à  $0,5 \cdot 10^{-1}$ :

Dans un M-file :

```

1 - clc,clear all
2 - x0=0.4;
3 - e=0.5*10^(-1);
4 - f=inline('x.^2.*exp(x)+2.*x-1','x');
5 - df=inline('(x.^2+2.*x).*exp(x)+2','x');
6 - x(1)=x0;
7 - n=1;
8 - for i=1:n
9 -     x(i+1)=x(i)-f(x(i))/df(x(i));
10 -     if (0.95*(x(i+1)-x(i))^2)<e
11 -         break
12 -     else
13 -         n=n+1;
14 -     end;
15 - end;
16 - x

```

Sur Matlab :

|     |
|-----|
| x = |
|-----|