Université Chahid Hamma Lakhdar d'El Oued

Faculté : Technologie
Département : [redacted]

Année universitaire : 2019/2020

TP 01

But TP:

Nous allons étudier rapidement quelques commandes et fonctions de bases concernant

- la manipulation des vecteurs
- la manipulation des matrices
- les représentations graphiques

1- Vecteur

Effectuer les commandes Matlab suivantes, recopier le résultat affiché et commenter les résultats obtenus :

Commandes	résultats	commentaires
» $x = [1\ 2\ 3\ 4]$ » $y = [-1,\ 0,\ -6,\ 2]$		
» $z = y'$		
» $u = 3*x-y+2$		
» **size**(x) » **size**(y) » **size**(z) » **size**(u)		
» $x - z$		
» $y./x$ » $x./y$		
» $x * x'$		
» $x' * x$		
» x/y » $(x * y')/(y * y')$		
» $x.\hat{}2$ » sin(x)		
» who		
» whos		

2- Matrices

Commandes	résultats	commentaires
» A = [4 -1 2 0 ; 1 5 1 -2 ; -2 2 6 1 ; 1 0 -3 5]		
» $A * x$ » $A * x'$		
» B = inv(A) » $B * A$		
» $B * x'$		
» size(A), size(B), size(x), size(x')		
» $A(1, 1)$ » $A(1, 1:2)$ » $A(1:2, :)$		
» A^2 » A.^2		

3- Représentation graphique

Effectuer les commandes Matlab suivantes et commenter les résultats obtenus :

Commandes	résultats	commentaires
» x=-1 :0.1 :1		
» y=sin(x)		
» **plot**(x,y)		
» **title**('exemple de graphique 2D')		
» **xlabel**('axe x')		
» **ylabel**('axe y')		
» **grid on**		
» **hold on**		
» **plot**(x,cos(x),'+r')		

TP N°2.
Méthodes numériques (Résolution de l'équation $f(x)=0$)
La Méthode de bissection (de dichotomie)

La méthode de bissection (de dichotomie) est, en mathématiques, un algorithme de recherche d'un zéro d'une fonction f continue sur l'intervalle $[a,\ b]$ qui consiste à répéter des partages d'un intervalle en deux parties ($[a,\ m]$ et $[m,\ b]$ où $m = (a+b)/2$) puis à sélectionner le sous-intervalle dans lequel existe un zéro de la fonction.

I. L'algorithme de la méthode de bissection

Cette méthode est utilisée pour calculer **les zéros** d'une fonction continue $f\,(\Re \Leftrightarrow \Re)$ dans un intervalle $[a,b]$:

1. ***Si*** $f(a).f(b) > 0$, alors la méthode termine (**pas de zéros**).
2. ***Sinon si*** $f(a).f(b) < 0$, on sait alors qu'il existe au moins un **zéro** $\overline{x}$ de f dans l'intervalle $[a,b]$.
3. ***Tant que*** $|b-a| > tol$:
   a) Soit $m=\frac{(a+b)}{2}$;
   b) ***Si*** $f(m) = 0$, alors $\overline{x} = m$ et la méthode **termine.**
   c) ***Sinon si*** $f(a)\ .\ f(m) < 0$ on pose
   $b = c;$
   d) ***Sinon si*** $f(m)\ .\ f(b) < 0$, on pose
   $a = m;$
   fin si
   affichez : (***$\overline{x}$ et $f(\overline{x})$ pour chaque itération***)
   fin tant que

fin si

II. Application

1. Implémenter la méthode de **la bissection** sous forme d'une fonction de **Matlab**:

```
function [xzero, iter, err, fzero] = bissection(a, b, tol);
```

Avec

les arguments d'entrée	les arguments de sortie
a, b : les limites de l'intervalle $[a,\ b]$. ***tol*** : L'erreur tolérée par le résultat.	***xzero*** : La racine trouvée par la méthode. ***iter*** : Le nombre d'itérations effectuées ***err*** : Valeur absolue de l'erreur finale (***abs***(***a-b***) finale). ***fzero*** : : l'image de zéro à trouver dans la fonction

2. Appliquer la fonction produite pour trouver le zéro de la fonction suivante :
   $f(x) = x^3 + x^2 - 3x - 3$ dans l'intervalle $[1, 2]$.
3. Produire le tableau des valeurs progressives ($x, f(x)$) jusqu'au zero.
4. Tracer cette fonction en indiquant dans le graphe le zéro trouvé.

RM

Pour définir une fonction courte il est possible d'utiliser les fonctions *inline* et les fonctions anonymes.
La création d'une fonction inline suit la syntaxe suivante :

nom_fonction = inline('expression mathematique de la fonction')

Par exemple pour définir la fonction : $2x^3-4x^2+x-3$, on écrit :

>> f= inline ('2*x^3-4*x^2+x-3','x') Ou **>> f= inline ('2*x^3-4*x^2+x-3')**

Pour calculer f(3) on écrit :

>> f(3)

ans = 18

```
function [xzero,iter,err,fzero]=bissection(a,b,tol);
%la fonction de dichotomie
%-------------------------
f=inline('x^3+x^2-3*x-3');% définition la fonction f(x):
a0=a;b0=b;% enregistrement les limités initiale pour la partie graphique
iter=0;
fa=f(a);
fb=f(b);
if fa*fb>0
    disp('pas de zero dans cet intervalle');
elseif fa*fb<0
    while abs(b-a)>tol
        m=(a+b)/2;
        fzero=f(m);
        if fzero ==0
            a=m; b=m;
            err=(a+b)/2;
        elseif f(a)*fzero<0
            b=m;
        elseif f(b)*fzero<0
            a=m;
        end
        str=['zero=',num2str(m),'fzero=',num2str(fzero)];
        disp(str)
        iter=iter+1;
    end
end
%le zero et leur image:
xzero=(a+b)/2;
fzero=f(xzero);
err=abs(b-a);
%la partie graphique:
fplot(f,[a0 b0]);
xlabel('x')
ylabel('la fonction f(x)')
text(xzero,fzero,'le zero calculé');
hold on
plot(xzero,fzero,'o')
grid on
```

Exécution :

```
>>[ xzero,iter,err,fzero]=bissection(1,2,1e-5);
```

Les questions :

Considérant l'équation : $f(x) = x^3+x^2-3x-3$

1. Dessinez la courbe de $f(x)$ sur l'intervalle [-2, 2], puis trouvez des intervalles convenables pour appliquer la méthode de bissection.

2. Pour chaque intervalle (un pour chaque racine), appliquez la fonction Matlab 'bissection.m' sur $f(x)$, en considérant : $tol = 0.001$.

Solution :

Dessiner la courbe

>> f = inline ('x^3+x^2-3*x-3','x'); >> fplot (f,[-2,2]), grid on

Choisir les intervalles : (Chaque intervalle [a,b] est choisie tel que f(a).f(b)<0)

Il existe 3 racines, donc on choisit 3 intervalles

$I_1 = [-2, -1.5]$ Pour la première racine — >> [Xzero, N, E, fzero] = bissection (-2,-1.5,0.001)

$I_2 = [-1.5, -0.5]$ Pour la deuxième racine — >> [Xzero, N, E, fzero] = bissection (-1.5,-0.5,0.001)

$I_3 = [1.5, 2]$ Pour la troisième racine — >> [Xzero, N, E, fzero] = bissection (1.5,2,0.00000001)