Méthodes Numériques

Université A/Mira de Béjaia
Faculté de Technologie
Département ST2

Mai 2015

TTP CORRIGÉ

EXERCICE1: METHODE DE DICHOTOMIE

NB : il est strictement interdit de changer les noms de variables ou de fonctions donnés.

Soit la fonction $h(x) = x^3 - 4x + 1$ définie sur R .

1.5pt 1- **Ecrire** une fonction Matlab qui retourne *h(x)*, puis compléter le tableau suivant:

x	***-1,66***	***-1***	***1***	***2***	***2,57***
h(x)	3.0657	1	-2	1	7.6946

1pt 2- **Selon ce tableau**, combien la fonction *h(x)* a-t-elle de racines **au minimum** dans l'intervalle [-3 3] ?
Réponse : h(x) possède 2 racines au minimum dans l'intervalle [-3 3]

1pt 3- **Tracer** le graphe de *h(x)* dans l'intervalle [-3 3].
Quel est le nombre de racines de *h(x)* **selon ce graphe**?.
Réponse : h(x) possède 3 racines dans l'intervalle [-3 3]

1.5pt 4- **Donner** des valeurs approximatives des racines de *h(x)* en utilisant son graphe.
Réponse: -2.1 , 0.26, 1.86

2pt 5- **Ecrire** un script pour calculer la racine approchée de $h(x)$ dans l'intervalle [a b] par la méthode de dichotomie (**données à lire: a , b et la précision p**).

1pt 6- **Exécuter** le programme pour trouver la racine négative : on prend **p=10^{-5} , a et b à choisir.**
Valeurs choisies : a=-3 , b= -2
Résultat : -2.1149

PROGRAMMES :

la fonction :

```
function y=h(x)
y=x^3-4*x+1 ;
```

le script :

```
a=input('donner a');
b=input('donner b');
p=input('donner la précision');
while abs(a-b)>p
    x=(a+b)/2;
    if h(a)*h(x)<0
        b=x;
    else
        a=x;
    end
end
x
```

TTP CORRIGÉ

EXERCICE2: METHODE DE DICHOTOMIE

Soit la fonction $d(x) = x^2(x^2 - 4) + 1$ définie sur R .

1.5pt 1- **Ecrire** une fonction Matlab qui retourne *d(x),* puis compléter le tableau suivant:

x	***-1,5***	***-0,4***	***0***	***2,44***	***3***
d(x)	-2.9375	0.3856	1	12.6310	46

1pt 2- **Selon ce tableau**, combien la fonction *d(x)* a-t-elle de racines **au minimum** dans l'intervalle [-3 3] ?
Réponse : d(x) possède UNE racine au minimum dans l'intervalle [-3 3]

1pt 3- **Tracer** le graphe de *d(x)* dans l'intervalle [-3 +3].
Quel est le nombre de racines de *d(x)* **selon ce graphe**?.
Réponse : d(x) possède 4 racines dans l'intervalle [-3 3]

1.5pt 4- **Donner** des valeurs approximatives des racines de *d(x)* en utilisant son graphe.
Réponse: -1.9 , -0.5, 0.5, 1.9

2pt 5- **Ecrire** un script pour calculer la racine approchée de *d*(*x*) dans l'intervalle [a b] par la méthode de dichotomie (**données à lire: a , b et la précision p**).

1pt 6- **Exécuter** le programme pour trouver la racine minimale : on prend **p=10**$^{-5}$, **a et b à choisir.**
Valeurs choisies : a=-2 , b= -1
Résultat : -1.9318

PROGRAMMES :

la fonction :

```
function y=d(x)
y=x^2*(x^2-4)+1;
```

le script :

```
a=input('donner a');
b=input('donner b');
p=input('donner la précision');
while abs(a-b)>p
    x=(a+b)/2;
    if d(a)*d(x)<0
        b=x;
    else
        a=x;
    end
end
x
```

TTP CORRIGE

NOM : .... GROUPE : ....

EXERCICE1: METHODE DU POINT FIXE

<u>NB</u> : il est strictement interdit de changer les noms de variables ou de fonctions donnés.

l'équation $x^3 + x - 1 = 0$ admet une solution unique dans l'intervalle [-3 3]. Cette équation peut s'écrire sous les 2 formes suivantes :

$$x = 1/(1+x^2) \quad , \qquad on\ pose\ \ g_a(x) = 1/(1+x^2)$$
$$x = 1 - x^3 \quad , \qquad on\ pose\ \ g_b(x) = 1 - x^3$$

3pt

1- **Tracer** les graphes de $g_a(x)$, $g_b(x)$, $g_a'(x)$ et $g_b'(x)$ dans l'intervalle [-3 3] pour étudier les propriétés de contractance et stabilité des 2 fonctions puis **compléter** le tableau suivant:

	Contractante?	***Stable?***
$g_a(x)$	☐ Oui ☐ Non Explication : K=max(g_a'(t))≈0.65<1	☐ Oui ☐ Non Explication : ga([-3 3])=[0.1 1] ⊂ [-3 3]
$g_b(x)$	☐ Oui ☐ Non Explication : K=max(g_b'(t))≈26>1	☐ Oui ☐ Non Explication : gb([-3 3])=[-26 28] ⊄ [-3 3]

1pt

2- **Tracer** le graphe de $x^3 + x - 1 = 0$ sur l'intervalle [-3 3]. **Donner** une valeur approximative de la racine :
<u>Réponse</u>: 0.68

3pt

3- En utilisant la fonction stable et contractante parmi $g_a(x)$ et $g_b(x)$, **écrire** un script permettant de calculer la racine approchée de $x^3 + x - 1 = 0$ par la méthode du point fixe (**<u>données à affecter</u>: la valeur initiale x_0 et la précision p**).

1pt

4- **Exécuter** le programme pour trouver la racine: on prend **p=10^{-7} , x_0 au choix.**
<u>Valeur choisie</u> : x_0= 0.5
<u>Résultat</u> : xr = 0.6823

<u>PROGRAMME :</u>

<u>la fonction :</u>

```
function y=ga(x)
y= 1/(1+x^2);
```

<u>le script :</u>

```
x0=0.5;
xr=ga(x0);
while abs (xr-x0)>p
    x0=xr;
    xr=ga(x0);
end
xr
```

TEST DE TP

NOM : .... GROUPE : ....

EXERCICE2: METHODE DU POINT FIXE

NB : il est strictement interdit de changer les noms de variables ou de fonctions donnés.

Soit l'équation $t^3 + t - 1 = 0$ qui peut s'écrire sous la forme :

$$t = 1/(1 + t^2) \quad , \qquad on\ pose\ \ g_t(t) = 1/(1 + t^2)$$

1pt 5- **Tracer** dans la même fenêtre graphique, les fonctions $y_t(t) = t$ et $g_t(t) = 1/(1 + t^2)$, sur l'intervalle [-2 5]. A l'aide de ces graphes, donner une approximation de la racine de $t^3 + t - 1 = 0$.

Réponse : 0.68

3pt 6- **Tracer** les graphes de $g_t(x)$ et $g_t'(x)$ dans l'intervalle [-2 5] pour étudier les propriétés de contractance et stabilité de $g_t(x)$puis **compléter** le tableau suivant:

	$g_t(t)$ ***Contractante?***	$g_t(t)$ ***Stable?***
Dans l'intervalle [-2 5]	☐ Oui ☐ Non Explication : $K = \max(g_t'(t)) \approx 0.65 < 1$	☐ Oui ☐ Non Explication : $gt([-2\ 5]) = [\ 0.03 \quad 0.2\ ] \subset [-2\ 5]$
Dans l'intervalle [0 1]	☐ Oui ☐ Non Explication : $K = \max(g_t'(t)) \approx 0.65 < 1$	☐ Oui ☐ Non Explication : $gt([0\ 1]) = [0.5 \quad 1] \subset [0\ 1]$

3pt 7- **Ecrire** un script permettant de calculer la racine approchée de $t^3 + t - 1 = 0$ par la méthode du point fixe (**données à lire: la valeur initiale t_0 et le nombre d'itérations N**).

1pt 8- **Exécuter** le programme pour trouver la racine: on prend **N=30**, **t_0 au choix.**

Valeur choisie : $t_0 = 0.5$

Résultat : tr = 0.6823

PROGRAMME :

la fonction :

```
function y=gt(t)
y= 1/(1+t^2);
```

le script :

```
t0=input ('donner t0');
N=input('donner le nombre d''itérations');
tr=gb(t0);
for i=1:N
    t0=tr;
    tr=gt(t0);
end
tr
```

Méthodes Numériques

Université A/Mira de Béjaia
Faculté de Technologie
Département ST2

Mai 2015

TTP CORRIGE

NOM : .... GROUPE : ....

EXERCICE1: METHODE DE GAUSS

NB : il est strictement interdit de changer les noms de variables ou de fonctions donnés.

Soit le système d'équations linéaires: $Ax = b$, de n inconnus. la méthode de Gauss permet de résoudre ce système en le modifiant d'abord en $x = g$, avec G une matrice triangulaire supérieure. Les deux étapes de cette méthode sont décrites par les algorithmes suivants:

Algorithme 1	*Algorithme 2*

pour s allant de 1 à $n-1$
  pour i allant de $s+1$ à n
    $K \leftarrow A_{i,s}/A_{s,s}$;
    pour m allant de s à n
      $A_{i,m} \leftarrow A_{i,m} - K \times A_{s,m}$
    fin
    $b_i \leftarrow b_i - K \times b_s$
  fin
fin
	pour s allant de n à 1
	  $x_s \leftarrow g_s/G_{s,s}$;
	  pour m allant de 1 à $s-1$
	    $g_m \leftarrow g_m - G_{m,s} \times x_s$
	  fin
	fin

7pt 1- **Ecrire** un script qui permet de résoudre le système d'équation $Ax = b$ par la méthode de Gauss. (**données à lire: la matrice A et le vecteur b**).

1pt 2- **Exécuter** le script pour trouver la solution du système suivant :

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix} \times \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{Bmatrix}$$

Réponse : [0 3 -3 0]'

PROGRAMME :

```
clear all
A=input('donner A');
b=input('donner b');
n=size(A,1);
for s=1:n-1
    for i=s+1:n
        K=A(i,s)/A(s,s);
        for m=s:n
            A(i,m)=A(i,m)-K*A(s,m);
        end
        b(i)=b(i)-K*b(s)
    end
end
G=A;
g=b;
% calcul de x
for s=n:-1:1
    x(s)=g(s)/G(s,s);
    for m=1:s-1;
        g(m)=g(m)-G(m,s)*x(s)
    end
end
x
```

TTP CORRIGE

NOM : .... GROUPE : ....

EXERCICE2: METHODE DE GAUSS

NB : il est strictement interdit de changer les noms de variables ou de fonctions donnés.

Soit le système d'équations linéaires: $Ax = b$, de n inconnus. la méthode de Gauss permet de résoudre ce système en le modifiant d'abord en $x = g$, avec G une matrice triangulaire supérieure. Les étapes de cette méthode sont décrites par les algorithmes suivants:

Algorithme 1	*Algorithme 2*

pour s allant de 1 à $n-1$
  pour i allant de $s+1$ à n
    $K \leftarrow A_{i,s}/A_{s,s}$;
    pour m allant de s à n
      $A_{i,m} \leftarrow A_{i,m} - K \times A_{s,m}$
    fin
    $b_i \leftarrow b_i - K \times b_s$
  fin
fin
	pour s allant de n à 1
	  $x_s \leftarrow g_s/G_{s,s}$;
	  pour m allant de 1 à $s-1$
	    $g_m \leftarrow g_m - G_{m,s} \times x_s$
	  fin
	fin

4.5pt 1- **Ecrire** la fonction **Modify** qui retourne G et g

1.5pt 2- En utilisant la fonction Modify, **écrire** le système suivant sous la forme $Gx = g$:

Réponse :

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ 0 & 1.5 & -1 & 0 \\ 0 & 0 & 1.33 & -1 \\ 0 & 0 & 0 & 1.25 \end{bmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1.5 \\ 2 \\ 2.5 \end{pmatrix}$$

1pt 3- En utilisant G, **vérifier** si le vecteur $x = \begin{Bmatrix} 1 \\ 1 \\ 3 \\ 4 \end{Bmatrix}$ est une solution du système précédent.

Réponse : G*x=[9 19 15 4]' $\neq g$

1pt 4- **Trouver** la solution du système précédent en utilisant la division gauche.

Réponse : A\b = [2 3 3 2]'

PROGRAMME :

```
function [G,g]=Modify(A,b)
n=size(A,1);
for s=1:n-1
    for i=s+1:n
        K=A(i,s)/A(s,s);
        for m=s:n
            A(i,m)=A(i,m)-K*A(s,m);
        end
        b(i)=b(i)-K*b(s)
    end
end
G=A;
g=b;
```

TTP CORRIGE

NOM : ………………………………………… GROUPE : ……………………..

EXERCICE1: METHODE DE CHOLESKY

NB : il est strictement interdit de changer les noms de variables ou de fonctions donnés.

Soit le système d'équations linéaires: $Ax = b$. La méthode de Cholesky permet de résoudre ce système en 3 étapes décrites par les algorithmes suivants:

Algorithme 1	***Algorithme 2***
pour s allant de 1 à n $L_{s,s} \leftarrow \sqrt{A_{s,s} - \sum_{k=1}^{s-1} L_{s,k}^2}$ **pour** j allant de $s+1$ à n $L_{j,s} \leftarrow \frac{1}{L_{s,s}}\left(A_{s,j} - \sum_{k=1}^{s-1} L_{s,k}L_{j,k}\right)$ **fin** **fin**	$y_1 \leftarrow b_1/L_{1,1}$; **pour** s allant de 2 à n $y_s \leftarrow \frac{1}{L_{s,s}}\left(b_s - \sum_{k=1}^{s-1} L_{s,k}y_k\right)$; **fin** ***Algorithme 3*** $x_n \leftarrow y_n/L_{n,n}^t$; **pour** s allant de $n-1$ à 1 $x_s \leftarrow \frac{1}{L_{s,s}^t}\left(y_s - \sum_{k=s+1}^{n} L_{s,k}^t\, x_k\right)$; **fin**

5pt 1- **Ecrire** la fonction ***Findx*** qui calcule la solution à partir de L et b.

1.5pt 2- **Résoudre** le système suivant en utilisant la fonction ***Findx***:

Réponse : L=[1 0 0 0 ; 1 1 0 0 ;1 2 1 0; 1 3 3 1] ; b=[2 1 1 1]' ;
x= findx(L,b') = [5 -6 4 -1]'

1.5pt 3- **Retrouver** la solution du système précédent à partir de A et b en utilisant la division gauche.

Réponse A=L*L' , x=A\b= [5 -6 4 -1]'

1	1	1	1
1	2	3	4
1	3	6	10
1	4	10	20

PROGRAMME :

```
function x=findx(L,b)
n=size(L,1); y(1)=b(1)/L(1,1);
for s=2:n
    k=1:s-1;
    y(s)=(b(s)-sum(L(s,k).*y(k)))/L(s,s);
end
Lt=L';
x(n)=y(n)/Lt(n,n);
for s=n-1:-1:1
    k=s+1:n;
    x(s)=(y(s)-sum(Lt(s,k).*x(k)))/Lt(s,s);
end
x=x';
```

Méthodes Numériques

Université A/Mira de Béjaia
Faculté de Technologie
Département ST2

Mai 2015

TTP CORRIE

NOM : .... GROUPE : ....

EXERCICE2: METHODE DE CHOLESKY

NB : il est strictement interdit de changer les noms de variables ou de fonctions donnés.

Soit le système d'équations linéaires: $Ax = b$. La méthode de Cholesky permet de résoudre ce système en 3 étapes décrites par les algorithmes suivants:

Algorithme 1

pour s allant de 1 à n

$$L_{s,s} \leftarrow \sqrt{A_{s,s} - \sum_{k=1}^{s-1} L_{s,k}^2}$$

pour j allant de $s+1$ à n

$$L_{j,s} \leftarrow \frac{1}{L_{s,s}}\left(A_{s,j} - \sum_{k=1}^{s-1} L_{s,k} L_{j,k}\right)$$

fin

fin

Algorithme 2

$y_1 \leftarrow b_1 / L_{1,1}$;

pour s allant de 2 à n

$$y_s \leftarrow \frac{1}{L_{s,s}}\left(b_s - \sum_{k=1}^{s-1} L_{s,k} y_k\right);$$

fin

Algorithme 3

$x_n \leftarrow y_n / L_{n,n}^t$;

pour s allant de $n-1$ à 1

$$x_s \leftarrow \frac{1}{L_{s,s}^t}\left(y_s - \sum_{k=s+1}^{n} L_{s,k}^t\, x_k\right);$$

fin

5pts 1- **Ecrire** la fonction ***decomposeA*** qui retourne la matrice L et L^t à partir de A.

1.5pt 2- **Ecrire** le système $Ax = b$ suivant sous la forme $L\,L^t x = b$:

Réponse A=[1 1 1 1; 1 2 3 4; 1 3 6 10; 1 4 10 20] ;[L,Lt]=decomposeA(A) ;

```
L = 1    0    0    0
    1    1    0    0
    1    2    1    0
    1    3    3    1
```

1.5pt 3- **Retrouver** la solution en utilisant la division gauche.

Réponse : y=L\b== [2 -1 1 -1]', x=Lt\b= [5 -6 4 -1]'

PROGRAMME :

```
function [L , Lt]=decomposeA(A)
n=size(A, 1);
L=zeros(n);
for s= 1:n
    k= 1:s- 1;
    L(s,s)=sqrt(A(s,s)-sum(L(s,k).^2));
        for j=s+ 1:n
            L(j,s)=(A(s,j)-sum(L(s,k).*L(j,k)))/L(s,s);
        end
end
Lt=L';
```