Calcul Formel et Sciences de la Matière – Février 2006
Durée 3 heures – Documents et calculatrices interdits

Prenez le temps de lire ce sujet. Les questions sont indépendantes pour la plupart. L'énoncé est un peu long mais il est noté sur 22 points.

1 Méthode de Gram-Schmidt [11pts]

Dans tout le problème les vecteurs sont de dimension n à coefficients dans le corps $\mathbb{R}$. Le procédé d'orthonormalisation de Gram-Schmidt est une méthode qui permet de construire une famille orthonormale de k vecteurs $(q_i)_{1\leq i\leq k}$ à partir d'une famille **libre** de vecteurs $(v_i)_{1\leq i\leq k}$. La famille se construit de manière incrémentale, c'est à dire qu'à l'étape j, on construit q_j en utilisant les valeurs des vecteurs q_h pour $h < j$. Ce travail sera abordé dans la deuxième partie du problème.

En troisième partie, une application de ce procédé à la méthode des moindres carrés sera traitée.

1.1 Famille Orthonormale

On rappelle qu'une famille de vecteurs $(q_i)_{1\leq i\leq k}$ est orthonormale ssi

$$\forall i \forall j \qquad 1 \leq i < j \leq k \Longrightarrow {}^t q_i.q_j = 0$$

où ${}^t v$ désigne le transposé du vecteur v et

$$\forall i \qquad 1 \leq i \leq k \Longrightarrow {}^t q_i.q_i = 1$$

On suppose que le paquetage `LinearAlgebra` est chargé en mémoire. On rappelle que la fonction `Transpose` permet de calculer la transposée d'une matrice et d'un vecteur. On rappelle aussi que lorsqu'on multiplie un vecteur ligne u par un vecteur colonne v, (dont le nombre de lignes de v est égal au nombre de colonnes de u) on obtient un nombre.

On donne le code Maple suivant :

```
# L désigne une liste de vecteurs de même dimension
est_orthonormale:=proc(L)
  local res, i, j , k;
  k:=nops(L);
  res:=true;
  for i from 1 to k-1 do
    for j from i+1 to k do
       if Transpose(L[i]).L[j]<>0 then
         res:=false;
       fi;
    od;
    if Transpose(L[i]).L[i]<>1 then
      res:=false;
    fi;
  od;
  if Transpose(L[k]).L[k]<>1 then
    res:=false;
  fi;
  res;
end:
```

On donne les vecteurs suivants $V1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ et $V2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

question 1 [1pt] . Quelles sont les instructions `Maple` permettant d'affecter aux variables `V1` et `V2` les valeurs correspondantes ?

Réponse:

```
V1:=<1 , -1>; V2:=<1 , 1>;
```

fin Réponse.

question 2 [1pt] . Que vaut alors l'expression suivante `est_orthonormale([V1,V2]);` ?

Réponse:

`false` car la famille n'est pas orthonormale (elle est seulement orthogonale).

fin Réponse.

question 3 [2pt] . La procédure `est_orthonormale` comporte une maladresse, laquelle ? Proposez une amélioration.

Réponse:

Il se peut que dès le premier test, on trouve une valeur différente de 0. À partir de cet instant, on connait le résultat, mais on va continuer à faire des calculs inutiles. Pour y remédier, il suffit de remplacer les boucles *pour* par des boucles *tant que* et d'adapter légérement le critère d'arret.

```
# L désigne une liste de vecteurs de même dimension
est_orthonormale:=proc(L)
  local res, i, j , k;
  k:=nops(L);
  res:=true;
  i:=1;
  while res and (i<=k-1) do
    j:=i+1;
    while res  and (j<=k) do
       if Transpose(L[i]).L[j]<>0 then
         res:=false;
       fi;
       j:=j+1;
    od;
    if Transpose(L[i]).L[i]<>1 then
      res:=false;
    fi;
    i:=i+1;
  od;
  if Transpose(L[k]).L[k]<>1 then
    res:=false;
  fi;
  res;
end:
```

fin Réponse.

1.2 la décomposition QR

Pour représenter la famille de vecteurs, on peut utiliser une matrice, dans laquelle les vecteurs de la famille $(v_i)_{1\leq i\leq k}$ seront rangés en colonne. A est une matrice à n lignes et k colonnes. On suppose que les k colonnes de A sont linéairement indépendantes, c'est à dire que A est de rang k. On va appliquer le procédé d'orthonormalisation de Schmidt pour fabriquer deux matrices nommées Q et R. La matrice Q a les mêmes dimensions que la matrice A et possède la propriété[1] ${}^tQ.Q = I_k$ où I_k désigne la matrice Identité de dimensions $k\times k$, la matrice R est une matrice carrée triangulaire supérieure inversible de dimension $k\times k$.

Dans la suite, on note A_{ij} le coefficient situé à la ligne i et à la colonne j de la matrice A

```
decomposition_QR:=proc( A )
pour j allant de 1 a k faire
  pour h allant de 1 a n faire
     V_h := A_hj
  fin_pour
  pour i allant de 1 a j-1 faire
     R_ij := 0
```

[1]c'est ainsi que se traduit matriciellement le fait que la famille est orthonormée

```
      pour h allant de 1 a n faire
        R_ij := R_ij + Q_hi * A_hi
      fin_pour
      pour h allant de 1 a n faire
        V_h := V_h - R_ij * Q_hi
      fin_pour
  fin_pour
  R_jj:= 0
  pour h allant de 1 a n faire
        R_jj := R_jj + V_h^2
  fin_pour
  R_jj:=sqrt(R_jj)
  pour h allant de 1 a n faire
        Q_hj := V_h/R_jj
  fin_pour
fin_pour
retourner la liste formée des matrices Q et R
```

question 4 [2pts] . Coder en Maple la fonction `decompositionQR`

Réponse:

```
decompositionQR:=proc(A)
local V,Q,R,j,i,h,k,n;
  n:=RowDimension(A);
  k:=ColumnDimension(A);
  V:=Vector[column](n);
  Q:=Matrix(n,k);
  R:=Matrix(k,k);
  for j from 1 to k do
    for h from 1 to n do
      V[h]:=A[h,j];
    od;
    for i from 1 to j-1 do
      R[i,j]:=0;
      for h from 1 to n do
         R[i,j]:=R[i,j]+Q[h,i]*A[h,i];
      od;
      for h from 1 to n do
         V[h]:=V[h]-R[i,j]*Q[h,i];
      od;
    od;
    R[j,j]:=0;
    for h from 1 to n do
       R[j,j]:=R[j,j]+V[h]^2;
    od;
    R[j,j]:=sqrt(R[j,j]);
    for h from 1 to n do
      Q[h,j]:=V[h]/R[j,j];
    od;
  od;
  [Q,R];
end;
```

fin Réponse.

question 5 [1pt] . Si la matrice A n'est pas de rang k, il y aura alors à une étape du calcul l'un des r_{jj} qui sera nul. Dans ce cas là, on décide d'interrompre le calcul grâce à l'instruction `error(gram_schmidt,"famille_liée")` qui permet d'interrompre un brutalement le calcul en produisant un message d'errreur. Indiquez les modifications à apporter dans le code.

Réponse:

il suffit d'ajouter avant la derniere boucle pour, les lignes suivantes :

```
if R[j,j]=0 then
  error(gram_schmidt,"famille_liée");
fi;
```

fin Réponse.

1.3 Application à la méthode des moindres carrés

On suppose que l'équation $Ax = b$ n'a pas de solution. mais on souhaite trouver une solution approchée par la méthode des moindres carrés.

question 6 [1pt] . Rappeler la méthode vue en cours.

Réponse:

On multiplie les deux membres de l'équation par ${}^tA$. On obtient ensuite le système ${}^tA.Ax = {}^tA.b$ qui admet une unique solution et qu'on peut résoudre par la méthode de Gauss. la solution trouvée, est l'approximation souhaitée.

fin Réponse.

On se propose maintenant d'utiliser le résultat du travail de la deuxième partie.

Voici un algorithme permettant d'obtenir x

```
Calculer la décomposition QR de A
Calculer le vecteur V = tQb
Résoudre le système Rx = V
```

question 7 [2pts] . Donner la code `Maple` d'une procédure nommée `remontee` paramétrée par R supposée triangulaire supérieure et V un vecteur et qui permet de résoudre le système $Rx = V$. Comme le système est déjà sous forme triangulaire, il suffit de calculer x_i en commençant par la fin. la valeur de x_i est donnée par la formule suivante

$$x_i = \frac{v_i - \sum_{j=i+1}^{n} R_{ij} v_j}{R_{ii}}$$

Réponse:

la matrice R est triangulaire supérieure. La résolution du système est alors très simple. Il suffit d'appliquer l'algorithme de remontée.

```
remontee:=proc(R,V)
local k,x,i,j,s;
k:=RowDimension(R);
x:=Vector[column](k);
for i from k to 1 by -1 do
  s:=V[i];
  for j from i+1 to k do
    s:=s-R[i,j]*x[j];
  od;
  x[i]:=s/R[i,i];
od;
x;
```

fin Réponse.

question 8 [1pt] . Réaliser une procédure `moindre_carre` paramètrée par A et b qui implante l'algorithme proposé. On réutilisera les fonctions définies précédement même si on ne les a pas écrites.

Réponse:

```
moindre_carre:=proc(A,b);
local l,Q,R,x,V;
  l:=decompositionQR(A);
  Q:=l[1];
  R:=l[2];
  V:=Transpose(Q).b
  x:=remontee(R,V);
  x;
end ;
```

fin Réponse.

2 Calcul approché de racine cubique [5pts]

Voici le code d'une fonction `Maple` permettant de calculer la racine cubique d'un nombre a avec une précision eps donnée.

```
racine_cubique_approchee:=proc(a,eps)
local f,g,u,u1;
  f:=x->x^3-a;
  g:=x->3*x^2;
  u:=a;
  u1:=u-f(u)/g(u);
  while abs(f(u))>eps or abs(u-u1)>eps do
    u:=u1;
    u1:=u-f(u)/g(u);
  od;
  u;
end;
```

question 9 [1pt] . quelle est le nom de la méthode mise en œuvre dans la procédure.

Réponse:

Il s'agit de la méthode de Newton.

fin Réponse.

l'appel suivant donnee un résultat en nombre exact

```
racine_cubique_approchee(2,0.001);
                                   2935497269576521
                                   ----------------
                                   2329904227757400
```

question 10 [1pt] . Comment faire, sur l'exemple précédent, pour avoir la valeur approchée de représentée par un nombre inexact ?

Réponse:

```
evalf(racine_cubique_approchee(2,0.001));
         1.259921861
```

fin Réponse.

question 11 [1pt] . On souhaite obtenir davantage de chiffre après la virgule comment procéder ?

Réponse:

Il suffit de changer la valeur de la variable globale `Digits` qui permet de fixer le nombre de chiffres significatifs utilisés pendant les calculs sur les nombres inexacts. par exemple :

```
Digits:=30;
```

fin Réponse.

On a fixé le nombre de chiffres significatifs à 100. Puis on a réalisé un tableau qui donne pour chaque itération, les valeurs de u, et le nombre de chiffres exacts.

nombre d'itérations	valeur approchée	nombre de chiffres exacts
1	1.50000000000000000000000000000000000000000000...	1
2	1.29629629629629629629629629629629629629629629...	2
3	1.26093222474174855127236079617031997984378936759...	3
4	1.25992186056592625032709872756596163029388789602...	7
5	1.25992104989539477442883227583580694702571309648...	13
6	1.25992104989487316476721082322559990978283014226...	25
7	1.25992104989487316476721060727822835057025146470...	50
8	1.25992104989487316476721060727822835057025146470...	99

TAB. 1 – Tableau vitesse de convergence

question 12 [1pt] . Par observation de ce tableau, quel semble être l'ordre de la méthode ?

Réponse:

Le nombre de chiffres exacts semble doubler à chaque itération, cela laisse à penser qu'il s'agit d'une méthode d'ordre 2

fin Réponse.

question 13 [1pt] . En considérant l'équation résolue par la fonction `racine_cubique_approchee`, il existe un résultat du cours permet de valider votre observation, lequel ? Vérifier qu'on se trouve bien dans le cadre d'application de cette propriété.

Réponse:

la méthode est d'ordre deux lorsque la racine est simple. Ici le seul cas ou la racine serait multiple, est $a = 0$ ce qui n'est pas le cas, (remarque, ici, dans le cas où a vaut 0 la suite est directement initialisée avec la valeur 0 est par conséquent elle est constante, et on a immédiatement la réponse)

fin Réponse.

3 Une équation différentielle ordinaire [6pts]

3.1 Analyse qualitative

On considère l'équation différentielle suivante :

$$(E) \qquad x' = 4(x-1)(x-2)(x+1)$$

question 14 [1pt] . Est-ce une équation différentielle autonome ? Justifier.

Réponse:

Il s'agit d'une équation différentielle ordinaire autonôme, car elle est du type $x' = f(x,t)$ où on a posé $f(x,t) = 4(x-1)(x-2)(x+1)$. On constate qu'en fait $f(x,t)$ ne dépend pas de t

fin Réponse.

question 15 [1pt] . Quelles sont les solutions constantes ?

Réponse:

On résout $f(x) = 0$, on obtient trois solutions constantes $x = -1$, $x = 1$, et $x = 2$

fin Réponse.

question 16 [1pt] . Dessiner la ligne de phase.

question 17 [1pt] . Discuter la nature des points fixes.

Réponse:

Soit on observe la ligne de phase, soit on calcule $f'(x)$ pour chacuns des points fixes.

$$f'(x) = 4(x-2)(x+1) + 4(x-1)(x+1) + 4(x-1)(x-2)$$

- $f'(-1) = 24 > 0$ on en déduit que -1 est un point fixe répulsif.
- $f'(1) = -8 < 0$ on en déduit que 1 est un point fixe attractif.
- $f'(2) = 12 > 0$ on en déduit que 2 est un point fixe répulsif.

fin Réponse.

3.2 Méthode d'Euler

On donne en plus comme condition initiale $x(0) = 0$

question 18 [1pt] . Intégrer par la méthode d'Euler l'équation différentielle (E) sur $[0, 1]$ en faisant $N = 1$ pas

Réponse:

$h = \frac{b-a}{N} = 1$ comme N vaut 1, on doit calculer t_0, t_1, x_0, x_1.

On a $\forall n \in \{0, 1\} \quad t_n = a + nh$ donc $t_0 = 0$ et $t_1 = 1$. On doit avoir $x_0 = x(0)$ donc $x_0 = 0$. Il ne reste plus qu'à calculer x_1. C'est à dire

$$x_1 = x_0 + hf(x_0) = 8$$

fin Réponse.

question 19 [1pt] . Intégrer par la méthode d'Euler l'équation différentielle (E) sur $[0, 1]$ en faisant $N = 2$ pas

Réponse:

$h = \frac{b-a}{N} = 1$ comme N vaut 2, on doit calculer $t_0, t_1, t_2, x_0, x_1, x_2$.

On a $\forall n \in \{0, 1\} \quad t_n = a + nh$ donc $t_0 = 0$, $t_1 = \frac{1}{2}$, et $t_2 = 1$. On doit avoir $x_0 = x(0)$ donc $x_0 = 0$. Il ne reste plus qu'à calculer x_1 et x_2. C'est à dire

$$x_1 = x_0 + hf(x_0) = 4$$

$$x_2 = x_1 + hf(x_1) = 64$$

fin Réponse.