MINFO0402

COMPTE RENDU DE TP1

Corentin MACHET
Quentin JUILLIARD
Licence 2e Année, Informatique – URCA

SOMMAIRE

I. PREAMBULE 2

II. RESOLUTION DE MATRICES TRIANGULAIRES 2

1. HYPOTHESES 2
2. FONCTION RESOUINF 2
3. FONCTION RESOUSUP 3
4. RESULTATS 3

III. RESOLUTION D'UN SYSTEME MATRICIEL 10

1. HYPOTHESES 10
2. FONCTION REDUC 10
3. FONCTION GAUSS 10
4. RESULTATS 11

I. PREAMBULE

Ce travail pratique tend à inculquer les manipulations algorithmiques de base via le langage Scilab. Il cherche à mettre en évidence les aspects mathématiques implicites des opérateurs disponibles nativement au sein de l'interpréteur. Ici, nous nous proposons d'étudier les résolutions matricielles, la réduction et la remontée de Gauss (Cf : exercices 4 et 5). L'exercice 6 fera l'objet d'une partie du CRTP2 comme stipulé dans le sujet.

A l'issue de ce travail, vous disposerez des fichiers suivants :

- A_MACHET_CORENTIN.txt et A_JUILLIARD_QUENTIN.txt
- ALIRE.txt (détail des fichiers de l'archive)
- Fonction.sci, Exercice4.sce et Exercice5.sce (fonctions et scripts des programmes)
- Le présent compte rendu

Les fichiers sources feront l'objet de commentaires.

NB : Comme les fonctions des exercices 4 et 5 ont été réalisé à la suite des 3 premiers exercices du TP, le fichier source Fonction.sci contiendra également les éléments de réponses des exercices précédents.

II. RESOLUTION DE MATRICES TRIANGULAIRES

1. HYPOTHESES

Il s'agit ici de proposer une solution à l'exercice 4 : résoudre des équations du type $Ax=b$. On admet que A est une matrice M(n,n) inversible et b un vecteur colonne à n composantes tel que n soit un entier donné.

Par convention, pour la fonction RESOUINF(A,b,n), A sera triangulaire inférieure, et pour la fonction RESOUSUP(A,b,n), A sera triangulaire supérieure.

Les coefficients diagonaux seront, par définition, non-nul.

2. FONCTION RESOUINF

```
// Résoudre x pour A une matrice carré et b un vecteur de taille n tq Ax = b
// Hypothèse : A est triangulaire inférieure et inversible
function X = RESOUINF(A,b,n)
    X = zeros(1, n) // initialisation vecteur nul
    X(1) = b(1)/A(1,1) // calcul du premier terme (qui va nous permettre de calculer les autres)
    for i = 2:n
        tmp = 0 // variable temporaire
        for j = 1:(i-1)
            tmp = tmp + A(i,j)*X(j)
        end
        X(i) = (b(i) - tmp)/A(i,i)
    end
endfunction
```

A chaque tour de la boucle i, nous conservons d'abord la somme des termes précédents de la ligne, que nous pouvons calculer grâce à la solution du tour i-1 (c'est d'ailleurs pour cela que X[1] est calculé en dehors de la boucle, puisque par définition, il n'a pas de solution précédente). Comme la variable temporaire tmp contient cette somme, il ne reste plus qu'à résoudre tmp+A[i,i].X[i]=b[i].

3. FONCTION RESOUSUP

```
// Résoudre x pour A une matrice carré et b un vecteur de taille n tq Ax = b
// Hypothèse : A est triangulaire supérieure et inversible
function X = RESOUSUP(A,b,n)
    X = zeros(1, n) // initialisation vecteur nul
    X(n) = b(n)/A(n,n) // calcul du dernier terme (qui va nous permettre de calculer les autres en remontant)
    for i = (n-1):-1:1
        tmp = 0 // variable temporaire
        for j = (i+1):n
            tmp = tmp + A(i,j)*X(j)
        end
        X(i) = (b(i) - tmp)/A(i,i)
    end
endfunction
```

De la même manière que pour RESOUINF, il s'agit de calculer la somme des éléments précédents avant de résoudre l'équation. La seule différence est de procéder par « remontée » et non plus par « descente », puisqu'ici c'est la ligne n qui n'a qu'un seul inconnu (c'est donc par elle qu'il faut commencer).

4. RESULTATS

```
--> exec("Exercice4.sce")
--> // Chargement des fonctions nécessaires
--> pathname = get_absolute_file_path("Exercice4.sce");
--> exec(pathname+'\Fonction.sci',-1);
--> // Initialisation des variables globales
--> min = 5;
--> max = 20;
--> n = round(rand()*(max - min + 1)) + min // min <= n <= max
 n  =

   18.
```

On choisit aléatoirement les dimensions de notre matrice

```
--> // Création des matrices...
--> I3 = [1,0,0; 0,1,0; 0,0,1] // matrice identité 3,3
 I3  =

   1.   0.   0.
   0.   1.   0.
   0.   0.   1.
```

La matrice identité sera utilisée pour un test trivial

```
--> A1_INF = [1,0,0; 1,2,0; -5,-8,-8] // matrice triangulaire inférieure inversible 3,3
 A1_INF  =
```

```
   1.   0.   0.
   1.   2.   0.
  -5.  -8.  -8.

--> A1_SUP = [2,3,4; 0,1,1; 0,0,5] // matrice triangulaire supérieure inversible 3,3
 A1_SUP  =

   2.   3.   4.
   0.   1.   1.
   0.   0.   5.

--> A2_INF = zeros(n,n);
--> for i = 1:n
-->     for j = 1:i
-->         A2_INF(i,j) = round(rand()*50)+1;
-->     end
--> end
--> A2_SUP = zeros(n,n);
--> for i = 1:n
-->     for j = n:-1:i
-->         A2_SUP(i,j) = round(rand()*50)+1;
-->     end
--> end
--> // ... et des vecteurs colonnes associés
--> b1 = [1;2;3]
 b1  =

   1.
   2.
   3.

--> b2 = round(100*rand(n,1));
--> // Affichage des matrices aléatoires
--> disp(A2_INF);

         column 1 to 8

   28.   0.    0.    0.    0.    0.    0.    0.
   12.   44.   0.    0.    0.    0.    0.    0.
   2.    36.   9.    0.    0.    0.    0.    0.
   43.   22.   18.   32.   0.    0.    0.    0.
   39.   47.   31.   29.   44.   0.    0.    0.
   29.   26.   38.   47.   37.   27.   0.    0.
   47.   33.   23.   34.   15.   30.   3.    0.
   48.   49.   29.   30.   17.   51.   41.   44.
   26.   6.    36.   44.   1.    41.   31.   7.
   27.   11.   40.   40.   36.   22.   18.   33.
   16.   10.   27.   7.    50.   35.   39.   24.
   6.    36.   17.   46.   42.   24.   22.   45.
   49.   25.   9.    4.    46.   48.   33.   38.
   31.   13.   27.   31.   21.   6.    27.   5.
```

```
   49.   48.   37.   40.   39.   50.   9.    26.
   47.   48.   51.   8.    41.   45.   37.   29.
   22.   43.   23.   32.   46.   18.   47.   19.
   38.   36.   17.   32.   38.   23.   12.   45.

         column 9 to 16

   0.    0.    0.    0.    0.    0.    0.    0.
   0.    0.    0.    0.    0.    0.    0.    0.
   0.    0.    0.    0.    0.    0.    0.    0.
   0.    0.    0.    0.    0.    0.    0.    0.
   0.    0.    0.    0.    0.    0.    0.    0.
   0.    0.    0.    0.    0.    0.    0.    0.
   0.    0.    0.    0.    0.    0.    0.    0.
   0.    0.    0.    0.    0.    0.    0.    0.
   41.   0.    0.    0.    0.    0.    0.    0.
   2.    46.   0.    0.    0.    0.    0.    0.
   38.   25.   38.   0.    0.    0.    0.    0.
   3.    28.   9.    4.    0.    0.    0.    0.
   10.   47.   42.   14.   13.   0.    0.    0.
   18.   45.   10.   10.   8.    34.   0.    0.
   38.   21.   26.   40.   18.   45.   11.   0.
   42.   5.    25.   2.    50.   40.   32.   40.
   12.   21.   15.   27.   21.   14.   16.   2.
   48.   19.   22.   29.   37.   44.   16.   49.

         column 17 to 18

   0.    0.
   0.    0.
   0.    0.
   0.    0.
   0.    0.
   0.    0.
   0.    0.
   0.    0.
   0.    0.
   0.    0.
   0.    0.
   0.    0.
   0.    0.
   0.    0.
   0.    0.
   0.    0.
   24.   0.
   6.    4.

--> disp(A2_SUP);

         column 1 to 8
```

4.	18.	3.	30.	2.	12.	29.	7.
0.	17.	49.	45.	35.	50.	8.	25.
0.	0.	23.	30.	43.	4.	4.	38.
0.	0.	0.	41.	28.	26.	25.	22.
0.	0.	0.	0.	7.	28.	20.	3.
0.	0.	0.	0.	0.	3.	44.	23.
0.	0.	0.	0.	0.	0.	4.	27.
0.	0.	0.	0.	0.	0.	0.	47.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.

column 9 to 16

50.	47.	16.	18.	21.	28.	18.	41.
22.	15.	31.	39.	6.	16.	7.	19.
33.	30.	42.	5.	9.	26.	33.	32.
8.	35.	49.	12.	34.	23.	4.	50.
10.	23.	17.	5.	17.	6.	9.	35.
31.	15.	25.	46.	25.	39.	46.	7.
11.	43.	22.	16.	22.	47.	30.	4.
47.	5.	16.	49.	5.	16.	12.	2.
3.	49.	2.	33.	3.	9.	34.	9.
0.	11.	3.	40.	38.	25.	3.	17.
0.	0.	22.	22.	44.	30.	49.	19.
0.	0.	0.	24.	45.	14.	43.	23.
0.	0.	0.	0.	6.	37.	47.	20.
0.	0.	0.	0.	0.	27.	17.	9.
0.	0.	0.	0.	0.	0.	5.	47.
0.	0.	0.	0.	0.	0.	0.	39.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.

column 17 to 18

42.	35.
11.	9.
10.	30.
46.	16.
45.	16.
36.	49.
14.	43.
17.	25.
12.	41.

```
  14.   40.
  47.   10.
  48.   32.
  38.   47.
  14.   29.
  30.   48.
  1.    41.
  47.   48.
  0.    7.

--> disp(b2);

  18.
  66.
  28.
  10.
  54.
  36.
  63.
  1.
  52.
  79.
  97.
  0.
  84.
  45.
  70.
  2.
  84.
  54.

--> // FONCTION RESOUINF(A,b,n)
--> RESOUINF(I3,b1,3) // reponse triviale = b1
 ans  =

  1.   2.   3.
```

Correct car b1 = (1,2,3)

```
--> XI1 = RESOUINF(A1_INF,b1,3)
 XI1  =

  1.   0.5  -1.5

--> A1_INF*XI1' // résultat attendu = b1
 ans  =

  1.
  2.
  3.
```

*Pour prouver que le résultat est solution, il suffit de comparer A1_INF*XI1 à b1. Comme c'est égal, le résultat est correct.*

```
--> XI2 = RESOUINF(A2_INF,b2,n)
 XI2  =
```

```
        column 1 to 4

  0.6428571   1.3246753  -2.3304473  -0.1511769

        column 5 to 8

  0.9840189   1.5618223  -4.6010727   1.5821539

        column 9 to 12

  4.4981706   2.1339896  -1.2954462  -28.836692

        column 13 to 16

  25.022021   2.5261966   32.574981  -61.552198

        column 17 to 18

 -3.1478366   505.53079

--> A2_INF*XI2' // résultat attendu = b2
 ans  =

  18.
  66.
  28.
  10.000000
  54.000000
  36.000000
  63.
  1.0000000
  52.000000
  79.000000
  97.000000
 -2.132D-14
  84.
  45.000000
  70.
  2.0000000
  84.
  54.000000

--> // FONCTION RESOUSUP(A,b,n)
--> RESOUSUP(I3,b1,3) // reponse triviale = b1
 ans  =

  1.   2.   3.

--> XS1 = RESOUSUP(A1_SUP,b1,3)
 XS1  =
```

Même démarche que précédemment

Remarque : il y a des petites disparités dues aux arrondis des calculs effectués par Scilab (notamment pour des valeurs nulles ou des fractions à infinies décimales)

```
  -2.8   1.4   0.6

--> A1_SUP*XS1' // résultat attendu = b1
 ans  =

   1.
   2.
   3.

--> XS2 = RESOUSUP(A2_SUP,b2,n)
 XS2  =

         column 1 to 4

  -7227625.5   2376457.7  -800874.66  -369153.39

         column 5 to 8

   702559.18  -184525.33   12248.356  -2122.574

         column 9 to 12

   1878.4606  -318.83348   36.867922   269.50061

         column 13 to 16

  -176.99653  -32.794544   50.772753  -7.9024238

         column 17 to 18

  -6.0911854   7.7142857

--> A2_SUP*XS2' // résultat attendu = b2
 ans  =

   18.000000
   66.000000
   28.000000
   10.000000
   54.000000
   36.000000
   63.000000
   1.0000000
   52.000000
   79.000000
   97.000000
   2.842D-14
   84.000000
   45.000000
   70.000000
```

2.0000000
84.000000
54.

III. RESOLUTION D'UN SYSTEME MATRICIEL

1. HYPOTHESES

Il s'agit à présent de proposer une solution à l'exercice 5 : adapter l'exercice précédent pour les matrices non-triangulaires, i.e. réaliser une réduction de Gauss, puis une remontée de Gauss (= RESOUSUP). Nous admettrons que pour toute équation $Ax=b$, A est une matrice carré inversible, puis triangulaire supérieure suite à la réduction. X est solution, obtenable sans permutation de ligne.

Toutes autres conditions concernant les matrices triangulaires restent inchangées.

2. FONCTION REDUC

```
// Réduction de Gauss sur un matrice A tq Ax = b
// Hypothèse : A est carré et inversible sans permutation de ligne ni pivot nul
function [A,b] = REDUC(A,b,n)
    for j = 1:n
        for i = (j+1):n
            // on détermine d'abord le pivot A(j,j)
            tmp = A(i,j)/A(j,j) // puis on calcule le facteur annulant le coefficient A(i,j) de la ligne que l'on traite
            for k = j:n
                // on applique le coefficient à chaque élément de la ligne...
                A(i,k) = A(i,k) - A(j,k)*tmp
            end
            b(i) = b(i) - b(j)*tmp // ...y compris au résultat de l'équation associée
        end
    end
endfunction
```

On conserve la valeur du facteur pour les transformations de chaque ligne, annulant les coefficients de la colonne traitée. Il est important de préserver tmp jusqu'à la fin du traitement de la colonne car la valeur de A[i,j] est modifiée par la suite.

Chaque coefficient de chaque ligne est ensuite traité par combinaison linéaire.

3. FONCTION GAUSS

```
// Résolution d'une matrice inversible tq Ax = b
// Hypothèse : on admet que les conditions d'utilisation des fonctions précédentes sont respectées
function X = GAUSS(A,b,n)
    // Réduction de Gauss
    [S,y] = REDUC(A,b,n)
    // Remontée de Gauss
    X = RESOUSUP(S,y,n)
endfunction
```

Ici, on solutionne l'équation par la méthode de Gauss (réduction + remontée). C'est pourquoi on utilise les fonctions précédemment déclarées.

4. RESULTATS

```
--> exec("Exercice5.sce")
--> // Chargement des fonctions nécessaires
--> pathname = get_absolute_file_path("Exercice5.sce");
--> exec(pathname+'\Fonction.sci',-1);
--> // Initialisation des variables globales
--> min = 5;
--> max = 20;
--> n = round(rand()*(max - min + 1)) + min // min <= n <= max
 n  =

   11.

--> // Création des matrices...
--> A1_SUP = [2,3,4; 0,1,1; 0,0,5] // matrice triangulaire supérieure inversible 3,3
 A1_SUP  =

   2.   3.   4.
   0.   1.   1.
   0.   0.   5.

--> A1 =  [1,1,2; 1,2,1; -5,-8,-8] // matrice inversible 3,3
 A1  =

   1.   1.   2.
   1.   2.   1.
  -5.  -8.  -8.

--> A2 = zeros(n,n);
--> for i = 1:n
-->     for j = 1:n
-->         A2(i,j) = round(rand()*50)+1;
-->     end
--> end
--> // ... et des vecteurs colonnes associés
--> b1 = [1;1;1]
 b1  =

   1.
   1.
   1.

--> b2 = round(100*rand(n,1));
--> // Affichage des matrices aléatoires
--> disp(A2);

         column 1 to 8

   13.   42.   26.   24.   51.   25.   32.   37.
   8.    40.   1.    16.   2.    17.   23.   31.
```

```
  17.   9.    46.   22.   20.   50.   7.    42.
  36.   48.   30.   13.   41.   33.   51.   23.
  6.    5.    21.   18.   51.   22.   8.    12.
  2.    45.   16.   33.   20.   43.   26.   11.
  6.    20.   4.    11.   20.   22.   27.   33.
  14.   29.   29.   41.   48.   46.   38.   15.
  44.   31.   8.    36.   10.   10.   5.    50.
  1.    19.   40.   4.    20.   21.   13.   47.
  18.   32.   6.    38.   15.   18.   37.   3.

         column 9 to 11

  15.   4.    46.
  28.   44.   14.
  29.   37.   32.
  11.   51.   10.
  1.    42.   11.
  21.   30.   12.
  49.   12.   27.
  21.   10.   9.
  8.    50.   32.
  30.   33.   14.
  46.   38.   28.

--> disp(b2);

  83.
  30.
  76.
  35.
  96.
  73.
  68.
  1.
  94.
  65.
  70.

--> // FONCTION REDUC(A,b,n)
--> REDUC(A1_SUP,b1,3) // reponse triviale = A1_SUP, b1
 ans  =

  2.   3.   4.
  0.   1.   1.
  0.   0.   5.
```

Trivialement, la réduction d'une matrice triangulaire supérieure est elle-même

```
--> REDUC(A1,b1,3)
 ans  =

  1.   1.   2.
  0.   1.  -1.
```

```
   0.   0.  -1.

--> REDUC(A2,b2,n)
 ans  =

        column 1 to 4

   13.   42.          26.          24.
   0.    14.153846  -15.          1.2307692
   0.    0.         -36.668478  -5.3913043
   0.    0.          0.         -30.702979
   0.    0.          0.          0.
   0.    0.          0.          0.
   0.    0.          0.          0.
   0.    0.          0.          0.
   0.    0.          0.          0.
   0.   -1.776D-15   0.          0.
   0.    0.          0.          1.776D-15

        column 5 to 8

   51.          25.          32.          37.
  -29.384615   1.6153846   3.3076923   8.2307692
  -142.03261   22.548913  -24.11413    20.320652
   201.04269  -98.778568   53.574478  -103.13161
   81.320242  -20.987965   16.563931  -30.712778
   8.882D-16   10.433971   7.7169491  -46.428543
   0.          0.          3.7011848   70.31358
   0.          0.          0.         -663.63466
   0.          0.          0.          0.
   0.          0.          0.          0.
   2.842D-14   0.          0.          0.

        column 9 to 11

   15.          4.          46.
   18.769231   41.538462  -14.307692
   70.282609   166.54348  -74.576087
  -159.21076  -279.15918   46.213428
  -45.963722  -28.659556   1.6135465
  -21.725476  -6.7606451  -36.345478
   63.863594  -14.989244   61.080385
  -569.23596  -26.098736  -541.6801
   27.436348  -247.22274   108.61681
   0.          181.36996  -94.110931
   0.          0.          98.183741

--> // FONCTION GAUSS(A,b,n)
--> X = GAUSS(A1,b1,3)
 X  =
```

```
   19.  -6.  -6.

-->  A1*X' // résultat attendu = b1
 ans  =

   1.
   1.
   1.

-->  X2 = GAUSS(A2,b2,n)
 X2  =

         column 1 to 4

   3.9494084    6.0491948    0.5049492   -2.9677717

         column 5 to 8

   4.3429369   -0.2270908   -9.9201206   -3.5149174

         column 9 to 11

   5.4536942   -0.2943868   -0.288157

-->  A2*X2' // résultat attendu = b2
 ans  =

   83.000000
   30.000000
   76.000000
   35.000000
   96.000000
   73.000000
   68.000000
   1.0000000
   94.000000
   65.000000
   70.000000
```

Comme X doit être solution de l'équation Ax=b, nous vérifions que le résultat de Ax soit b (ici, b1 = (1,1,1)).

Même démarche que précédemment