Rapport MINFO0402

Exercice 4

Résoudre des matrices triangulaires inversibles inférieures, puis supérieures (correspondant à A) de taille (n,n) par rapport au vecteur colonne b de même taille que A (n). X correspond à la solution.

Fonctions utilisées :

[X]=RESOUINF(A, b, n) : Prend en paramètre la matrice A, triangulaire inversible inférieure, le vecteur colonne b, et la taille commune n, puis retourne X la solution à l'équation AX = b

[X]=RESOUSUP(A, b, n) : Prend en paramètre la matrice A, triangulaire inversible supérieure, le vecteur colonne b, et la taille commune n, puis retourne X la solution à l'équation AX = b

Méthode de parcours

Les matrices possèdent respectivement les formes suivantes

$$A = (a_{i,j}) = \begin{bmatrix} a_{1,1} & 0 & \cdots & \cdots & 0 \\ a_{2,1} & a_{2,2} & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & 0 \\ a_{n,1} & a_{n,2} & \cdots & \cdots & a_{n,n} \end{bmatrix}$$

$$A = (a_{i,j}) = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & \cdots & a_{1,n} \\ 0 & a_{2,2} & & & a_{2,n} \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & a_{n,n} \end{pmatrix}$$

Source : Wikipédia

Notre méthode de parcours devra donc être différent. En effet, l'initialisation de X, pour une matrice triangulaire inférieure, devra commencer à la ligne 1 (a(1,1)*X(1) = b(1)) tandis que ce sera l'inverse pour une matrice triangulaire supérieure (on commence à la ligne n avec a(n,n)*X(n) = b(N)).

Le Script vérifie ici si la fonction donne le bon résultat en utilisant les outils de scilab, permettant eux aussi d'obtenir le résultat de RESOUINF et RESOUSUP.

```
n  =

  5.
interval  =

  10.

A :

  9.   1.   1.   1.   3.
  1.   7.   1.   5.   3.
  4.   9.   5.   8.   10.
  4.   6.   0.   5.   2.
  7.   4.   9.   8.   8.

b :

  5.
  0.
  2.
  5.
  7.

X (RESOUINF) :

  0.5555556
 -0.0793651
  0.0984127
  0.6507937
 -0.3329365

X (RESOUSUP) :

  0.5125397
 -0.4978571
 -2.39
  0.65
  0.875

Vérification X (RESOUINF) :

  0.5555556
 -0.0793651
  0.0984127
  0.6507937
 -0.3329365

Vérification X (RESOUSUP) :

  0.5125397
 -0.4978571
 -2.39
  0.65
  0.875
```

Exercice 5

Appliquer une réduction de Gauss pour transformer une matrice A inversible (transformation supposée sans permutation) en une matrice triangulaire supérieure afin de résoudre AX = b via RESOUSUP précédemment étudié.

Fonctions utilisées :

[X]=RESOUSUP(A, b, n) : Prend en paramètre la matrice A, triangulaire inversible supérieure, le vecteur colonne b, et la taille commune n, puis retourne X la solution à l'équation AX = b

[A, b] = REDUC(A, b, n) : Prend en paramètre la matrice A inversible, le vecteur colonne b, et la taille commune n, puis retourne U et Y (ici A et b pour des raisons pratiques), formes adaptées de A et b.

[X]=GAUSS(A, b, n) : Prend en paramètre la matrice A inversible, le vecteur colonne b et la taille commune n, puis retourne X la solution de AX = b, après avoir passé A et b à REDUC, puis RESOUSUP.

Méthode de parcours :

Notre matrice est supposée sans permutation. On peut donc appliquer un calcul des coefficients sur chaque ligne, puis parcourir les lignes suivantes (de k+1 à la fin n) en transformant les valeurs de A et b via le coefficient de la ligne k.

La fonction est vérifiée en calculant directement le résultat via l'outil \ de Scilab.

Première capture d'écran :

Affichage des valeurs de A et b, ainsi que des constantes définies dans le script n la taille et interval la définition de l'invervalle des valeurs.

On affiche ensuite le résultat de la réduction de A et b (U et Y) par REDUC

Deuxième capture d'écran :

On affiche le résultat de RESOUSUP sur U et Y (A et b réduits).

```
n  =

  5.
interval  =

  10.

A :

  3.   7.   3.   8.   2.
  1.   2.   9.   6.   9.
  8.   9.   1.   4.   3.
  0.   7.   5.   7.   7.
  4.   2.   7.   1.   0.

b :

  2.
  10.
  9.
  1.
  9.

A réduit :

  3.   7.           3.      8.           2.
  0.  -0.3333333    8.      3.3333333    8.3333333
  0.   0.        -239.   -114.        -244.
  0.   0.           0.    -5.5188285    5.3807531
  0.   0.           0.      0.          -9.8498863

b réduit :

  2.
  9.3333333
 -267.
  3.7322176
 -6.0576194
```

```
X (Par la fonction) :

  1.734452
 -0.7710899
  0.5258621
 -0.0766626
  0.6149938

X (Par Scilab) :

  1.734452
 -0.7710899
  0.5258621
 -0.0766626
  0.6149938
```

Exercice 6

Résoudre MX = d avec M une matrice tridiagonale composée des valeurs (diagonales) a, b et c, considérés ici comme des tableaux de dimension 1. a(1) et c(n) sont initialisés par le script mais non utilisés (plus pratique que réduire la taille de ces deux tableaux). On sait que :

$$\begin{cases} x_i = e_i x_{i+1} + f_i & 1 \le i \le n-1 \\ x_n = f_n \end{cases}$$

Où l'on donnera les expressions récurentes définissant les e_i, $(1 \le i \le n-1)$ et les f_i, $(1 \le i \le n)$.

On considérera donc les tableaux f et e.

Pour les tailles, M est de taille (n,n) et a, b, c, d, e, f et x sont de taille (n).

<u>Fonctions utilisées :</u>

[x]=RESOUTRI(a, b, c , d, n) : Prend en paramètre la matrice M sous forme de ses 3 diagonales a, b et c, tableaux monodimensionnels de taille n, elle aussi passée en paramètre. d correspondant au deuxième membre de l'équation. Retour X, la solution de MX = d.

f=PRODMATTAB(a, b, c , x, n) : Prend en paramètre la matrice M sous forme de ses 3 diagonales a, b et c, tableaux monodimensionnels de taille n, elle aussi passée en paramètre. Multiplie M par le vecteur colonne x et renvoie f, le résultat.

<u>Méthode de parcours :</u>

Par application de la formule de l'énoncé, on a :

x(1) = -c(1)/b(1) * x(2) + d(1)/b(1) soit x(1) = e(1) * x(2) + f(1)

x(i) = (-c(i) /(a(i)*e(i-1)+b(i))) * x(i+1) + (d(i) - a(i)*f(i-1))/ (a(i)*e(i-1)+b(i)) soit x(i) = e(i) * x(i+1) + f(i)

x(n) = (d(n)-a(n)*f(n-1))/((a(n)*e(n-1))+b(n)) soit soit x(1) = e(1) * x(2) + f(1)

On parcourt donc une première fois notre matrice de 1 à n pour calculer e et f (traitement spécial pour 1, donc 2 à n techniquement), puis, de n à 1 (idem, traitement spécial pour n donc de n-1 à 1) pour définir x (nécessitant la valeur au rang i+1 sauf pour x(n)).

Le calcul de la deuxième fonction est une version simplifiée du calcul classique, puisque l'on a ici 3 tableaux et au maximum 3 valeurs non nulles par ligne.

```
n  =

  7.
interval  =

  10.

a :

  7.
  10.
  9.
  7.
  8.
  8.
  10.

b :

  2.
  5.
  7.
  4.
  5.
  8.
  3.

c :

  9.
  2.
  9.
  9.
  10.
  1.
  8.

d :

  9.
  7.
  6.
  8.
  1.
  5.
  0.
```

```
X :

  0.8917506
  0.8018332
 -2.9633362
  2.1696505
  2.2294168
 -2.7504288
  9.1680961

Norme de MX-D (vérifications). Satisfaisant si très proche de 0:

  4.351D-15
```