

MAT-19961 CALCUL MATRICIEL EN GÉNIE

Solutions - Devoir #3

2.6.2

$$\text{On trouve } \mathbf{x}^{(1)} = \begin{bmatrix} 2.5 \\ 5.375 \end{bmatrix}, \mathbf{x}^{(2)} = \begin{bmatrix} 3.0375 \\ 5.0625 \end{bmatrix}, \mathbf{x}^{(3)} = \begin{bmatrix} 3.0063 \\ 4.9953 \end{bmatrix}, \dots, \mathbf{x}^{(5)} = \begin{bmatrix} 2.9999 \\ 5.0001 \end{bmatrix}.$$

2.6.6

$$\text{On trouve } \mathbf{x}^{(1)} = \begin{bmatrix} 2.5 \\ 5.0625 \end{bmatrix}, \mathbf{x}^{(2)} = \begin{bmatrix} 3.0063 \\ 4.9992 \end{bmatrix}, \dots, \mathbf{x}^{(4)} = \begin{bmatrix} 3.0000 \\ 5.0000 \end{bmatrix}.$$

Note: Pour le système des numéros 2.6.2 et 2.6.6, la méthode de Gauss-Seidel converge plus vite que la méthode de Jacobi.

2.6.10

a) La matrice est strictement dominante en diagonale car $|3| > |-2|$ et $|2| < |3|$.

b) La matrice n'est pas strictement dominante en diagonale. Prenons la 2e ligne. $|3| + |-4| > |6|$.

2.6.14

La méthode de Gauss-Seidel produit

$$\mathbf{x}^{(18)} = \begin{bmatrix} 10.000 \\ 13.000 \\ -15.000 \end{bmatrix}$$

Ce résultat est précis à trois chiffres significatifs après le point. Notons qu'en arrondissant $\mathbf{x}^{(17)}$ à trois chiffres après le point, on obtient le bon résultat.

$$\mathbf{x}^{(17)} = \begin{bmatrix} 9.9991 \\ 12.9995 \\ -14.9997 \end{bmatrix}$$

Problème de Matlab

```
function y=jacobi(A, b, x0, k)

% Solution d'un système d'équations linéaires Ax = b
% par la méthode de Jacobi
% Dx_(k+1) = Nx_k + b
% D: matrice diagonale contenant la diagonale de A
% N = D - A
% Conditions initiales: x0
% La fonction calcule k itérations

D=diag(diag(A));
N=D-A;
x=x0;

% boucle "for"

for i=1:k,    x = D\(N*x + b);
end

x' % affiche le résultat sur une ligne pour sauver de l'espace
```

```
function y=seidel(A, b, x0, k)

% Solution d'un système d'équations linéaires Ax = b
% par la méthode de Gauss-Seidel
% Mx_(k+1) = Nx_k + b
% M: matrice triangulaire inférieure de A
% N = M - A
% Conditions initiales: x0
% La fonction calcule k itérations

M=tril(A);
N=M-A;

x=x0;

% boucle "for"

for i=1:k,
    x = M\(N*x + b);
    k
    x'
end

x' % affiche le résultat sur une ligne pour sauver de l'espace
```

Note: On utilise l'instruction “\” pour résoudre le système diagonal ou le système triangulaire. En pratique, il faudrait utiliser un sous-programme pour résoudre de tels systèmes. Les deux fonctions doivent être sauveés dans deux fichiers, jacobi.m et seidel.m, dans votre répertoire principal ou dans un sous-répertoire nommé “Matlab”.

Exemples d'utilisation

```
>> A
```

```
A =
```

```
    50    -1     0
     1  -100     2
     0     2    50
```

```
>> b
```

```
b =
```

```
    149
   -101
    -98
```

```
>> x0
```

```
x0 =
```

```
    0
    0
    0
```

```
>> jacobi(A,b,x0,2)
```

```
ans =
```

```
    3.0002    1.0006   -2.0004
```

```
>> seidel(A,b,x0,2)
```

```
ans =
```

```
    3.0008    1.0000   -2.0000
```

```
>>
```