

```

1 display('Dans ce fichier n''est programmee que la méthode de Jacobi')
2 Methode = input(' choisir la méthode de résolution')
3 %vous repondez: 'MGS'ou 'Jacobi'
4 %Exemple 1:A=[2 -1 1;2 2 2;-1 -1 2], b=[-1; 4; 5]
5 %Exemple 2: A=[-1 2;2 -1], b=[-2;5]
6 %Exemple 3: A=[-2 1;1 2 ], b=[1;3]
7 %Exemple 4:
8 A=[5 2; 2 2], b=[-1; 5]
9 [n,m]= size(A);
10 switch Methode
11 case {'Jacobi'}
12 display('Resolution avec la methode de Jacobi')
13 CMJ=0;
14 for i =1:n
15 if sum((abs(A(i,:)))-2*abs(A(i,i)) >= 0
16 display(['La lere ligne ds laquelle la diagonale n''est pas strict. dom i=',num2str
(i)])
17 display('Ce qui implique que A n''est pas a diagonale strictement dominante')
18 display('On va alors calculer le rayon spectral de BJ')
19 CMJ=1;
20 break %on quitte la boucle "for" sur les lignes de A
21 end
22
23 end
24 if CMJ== 0
25 display('A est a diagonale strictement dominante')
26 display('On n''a pas besoin de calculer le rayon spectral')
27 end
28 D=diag(diag(A));
29 E=-tril(A,-1);
30 F=-triu(A,+1);
31 BJ=(inv(D))*(E+F) %la matrice itérative pour la formule de Jacobi
32 if CMJ == 1% A n'est pas a diagonale strictement dominante.
33 ValPBJ= eig(BJ); %calcul de toutes les valeurs propres de BJ
34 rhoJ=norm(ValPBJ,Inf); %rhoJ= rayon spectral de BJ
35 if rhoJ >=1
36 display(['Le rayon spectral de BJ =',num2str(rhoJ)] )
37 display('La méthode de Jacobi ne converge pas')
38 return %on stop les calculs et on quitte le pg
39 end
40 display(['Le rayon spectral de BJ =', num2str(rhoJ),' < 1 et donc (MJ) converge'] )
41 end
42 dJ=inv(D)*b
43 xJ=zeros(n,1); %x^(1) solution initiale
44 %Les solutions seront stockées dans une matrice
45 %SOLJ. Le nombre de lignes est celui des composantes
46 %du vecteur x^(k) et celui des colonnes correspond au nombre d'itérations.
47 SOLJ(1:n,1) = xJ;
48 tol=0.01; % tolérance pour stopper les calculs
49 %initialisation de la différence entre 2 sols consécutives en norme
50 % ca représente l'erreur
51 difJ=1;
52 %initialisaton du nombre d'itérations
53 k=0;
54 while difJ > tol % on peut mettre d'autres tests d'arret

```

```
55 k = k + 1;
56 Y=BJ*xJ + dJ; % Y= x^(k+1), xj=x^(k)
57 difJ= norm(Y-xJ);%||x^(k+1) - x^(k)|| l'erreur a l'iteration k
58 xJ=Y;
59 %les erreurs sont stockees par iteration ds le vecteur NJ
60 NJ(k)= difJ;
61
62 SOLJ(1:n,k+1) = xJ;%x^(k+1) est stockee ds la colonne k+1 de la matrice SOLJ
63 end
64 display('*****')
65 display(' Toutes les solutions obtenues: ')
66 display('*****')
67 SOLJ
68 display('*****')
69 display('le nombre d'iterations:')
70 display('*****')
71 k
72 display('*****')
73 display('l'erreur entre 2 sol. consecutives evaluee en norme')
74 display('*****')
75
76 disp('||solJ(k+1)- solJ{k}|| =');
77 NJ
78 display('*****')
79 display('diffJ=||A*x^(m)-b||, x^(m) est la derniere solution obtenue =')
80 %diffJ=|| A*x^(m) -b||, x^(m) est la derniere solution calculee
81 display('*****')
82 solJ=xJ
83 solexacte=inv(A)*b
84 diffJ= norm(A*xJ-b)
85
86 case {'MGS'}
87 display('Resolution avec la methode de Gauss-Seidel')
88 otherwise
89 %Si a la demande de input.on ne repond pas
90 %par 'MGS'pour La methode de Gauss-Seidel ou
91 %'Jacobi' pour la methode de Jacobi , on aura la remarque suivante
92 warning('choisir MGS ou Jacobi')
93 end
94
```