



MDI224

Interpolation par splines cubiques

Travaux Pratique 2 - Deuxième semestre de 2011

PROFESSEUR: ROLAND BADEAU

ANTHONY CLERBOUT CASIER: 234
TIAGO CHEDRAUOI SILVA CASIER: 214

Décembre 15, 2011

Table des matières

1	Résolution du système linéaire	3
1.1	Méthode du gradient à pas constant	3
1.1.1	Implémentation	3
1.2	Méthode du gradient à pas optimal	3
1.2.1	Implémentation	3
1.3	Méthode du gradient conjugué	4
1.3.1	Implémentation	4

Table des figures

1.1.1 Implémentation

Méthode du gradient à pas constant

```

1 % % % % % % % % % % % % % % % %
2 % 19/01/11
3 % Chedraoui Silva,Tiago
4 % Casier: 214
5 % CLERBOUT, Anthony
6 % Casier: 234
7 % TP2: interpolation par splines
8 % cubiques partie II
9 % Description: Gradient a pas constant
10 % % % % % % % % % % % % % % % %
11
12 function x = mygradient(A,b,x0,beta,eps
    )
13
14 % Entree
15 % A : matrice
16 % b : vecteur
17 % x0: vecteur d'initialisation
18 % beta: le pas
19 % esp: critere de convergence

```

```

20
21 xn=[1 3 2 0 1]'; % rand vector
22 x=x0;
23 x=[x xn];
24 i=1;
25
26 while (norm (x(:,i+1) - x(:,i))>eps),
27
28     % Pour la fonction
29     %  $J=1/2*xT*A*x-xTb$ ;
30     % Le gradient est donne par
31     %  $g=Ax-b$ ;
32
33     g = A*xn-b;
34
35     xn = x(:,i+1)-beta*g;
36     x = [x xn];
37     i++;
38
39 end;

```

1.2.1 Implémentation

Méthode du gradient à pas optimal

```

1 % % % % % % % % % % % % % %
2 % 19/12/11
3 % Chedraoui Silva,Tiago
4 % Casier: 214
5 % CLERBOUT, Anthony
6 % Casier: 234
7 % TP2: interpolation par splines
8 % cubiques partie II
9 % Description: Gradient a pas constant
10 % % % % % % % % % % % % % % % %
11
12 function x = mygradient(A,b,x0,eps)
13
14 % Entree
15 % A : matrice
16 % b : vecteur

```

```

17 % x0: vecteur d'initialisation
18 % esp: critere de convergence
19
20 xn=[1 3 2 0 1]'; % rand vector
21 x=x0;
22 x=[x xn];
23 i=1;
24
25 while (norm (x(:,i+1) - x(:,i))>eps),
26
27     % Pour la fonction
28     % J=1/2*xT*A*x-xTb;
29     % Le gradient est donne par
30     % g=Ax-b;
31
32     g = A*xn-b;
33

```

```

34  beta = g'*g/(g'*A*g);
35
36  xn = x(:,i+1)-beta*g;
37  x = [x xn];
38  i++;

```

```

39
40 end;

```

1.3 Méthode du gradient conjugué

1.3.1 Implémentation

Pour voir la méthode du gradient conjugué, on a fait le code suivant :

Méthode du gradient conjugué

```

1  % % % % % % % % % % % % % % % %
2  % 19/12/11
3  % Chedraoui Silva,Tiago
4  % Casier: 214
5  % CLERBOUT, Anthony
6  % Casier: 234
7  % TP2: interpolation par splines
8  % cubiques partie II
9  % Description: Gradient a pas constant
10 % % % % % % % % % % % % % % % %
11
12 function x = gradient_conjugué(A,b,x0,
    eps)
13
14 xn = [1 3 2 0 1]'; % rand vector
15 x = x0;
16 x = [x xn];
17 i = 1;

```

```

18
19 g = A*xn-b;
20 w = g;
21
22 while (norm (x(:,i+1) - x(:,i))>eps),
23
24     beta = g'*w/(w'*A*w);
25
26     xn = x(:,i+1)-beta*g;
27     x = [x xn];
28
29     g = A*xn-b;
30     a = -g'*A*w/(w'*A*w);
31     w=g+a*w;
32
33     i++;
34
35 end

```