

Master Mathématiques Appliquées

FREEFEM++ (TP 1)

Faculté Polydisciplinaire à Larache

Réalisé par : ABARHANE MOUAD

Encadré par : Pr. CHAKIR TAJANI

Année universitaire : 2022/2023

1 Suite de Fibonacci

- Code FreeFem++ :
- Gnuplot :

2 Génération aléatoire

- les fonctions Random sur FreeFem++:
- Code FreeFem++:
- Matrice Toeplitz :
- Matrice d'Hilbert :
- Matrice Magique :

3 Résolution $Ax = b$

- La méthode de Cholesky :
- Décomposition LU :

4 Interpolation de Lagrange :

5 Intégration numérique :

- Méthode de Simpson :
- Méthode de Trapèze :

Fibonacci.edp X

C: > Users > mouad > OneDrive > Desktop > Fibonacci.edp

```

1  int u0= 0,u1=1,n,i,un;
2  cout << "Donner le nombre " << endl;
3  cin >> n;
4  int [int]v=(n);
5  v(0)=u0;
6  v(1)=u1;
7  cout << "le terme u(0)=" << u0 << endl;
8  cout << "le terme u(1)=" << u1 << endl;
9  for(i=2;i<n;i++){
10  un=u0+u1;
11  u0=u1;
12  u1=un;
13  cout << "le terme u(" << i << ")=" << un << endl;
14  v(i)=un;
15  }
16  cout << "le vecteur v de dimension n=" << v << endl;
17  {
18  ofstream fout ("data_Tp.dat");
19  int u0=0 , u1=1 , n,i,un;
20  cout << "donner le nombre n " << endl;
21  cin >> n;
22  int [int]v(n);
23  v[0]=u0;
24  v[1]=u1;

```

```
20  cout << "donner le nombre n " << endl;
21  cin >> n;
22  int [int]v(n);
23  v[0]=u0;
24  v[1]=u1;
25  fout << v(0)<< ""<<u0<<endl;
26  fout << v(1)<< ""<<u1<<endl;
27  int sum =1;
28  for (i=2;i<n;i++)
29  {
30  un=u0+u1;
31  u0=u1;
32  u1=un;
33  sum+= un ;
34  v[i]=un ;
35  fout <<v(i)<<""<<un<<endl;
36  }
37  cout <<"la somme est"<<sum<<endl;
38  }
```

FreeFem++-cs *

File Edit Compute Graphics Options Help

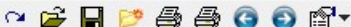
le terme $u(68)=1073992269$
le terme $u(69)=-188547518$
le terme $u(70)=885444751$
le terme $u(71)=696897233$
le terme $u(72)=1582341984$
le terme $u(73)=-2015728079$
le terme $u(74)=-433386095$
le terme $u(75)=1845853122$
le terme $u(76)=1412467027$
le terme $u(77)=-1036647147$
le terme $u(78)=375819880$
le terme $u(79)=-660827267$
le terme $u(80)=-285007387$
le terme $u(81)=-945834654$
le terme $u(82)=-1230842041$
le terme $u(83)=2118290601$
le terme $u(84)=887448560$
le terme $u(85)=-1289228135$
le terme $u(86)=-401779575$
le terme $u(87)=-1691007710$

```
cout << "donner le nombre n " << endl;
```



gnuplot

File Plot Expressions Functions General Axes Chart Styles 3D Help



G N U P L O T

Version 5.4 patchlevel 5 last modified 2022-09-28

Copyright (C) 1986-1993, 1998, 2004, 2007-2022

Thomas Williams, Colin Kelley and many others

gnuplot home: <http://www.gnuplot.info>

faq, bugs, etc: type "help FAQ"

immediate help: type "help" (plot window: hit 'h')

Terminal type is now 'windows'

Options are '0 color solid butt enhanced standalone'

gnuplot> cd 'C:\Users\mouad\OneDrive\Desktop\Tp Freefem ++'

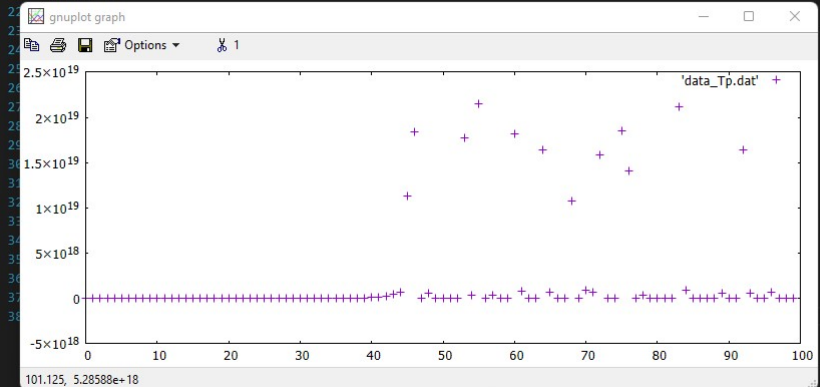
gnuplot> plot 'data_Tp.dat'

gnuplot>

```

20 cout << "donner le nombre n " << endl;
21 cin >> n;

```



```

load "ffrandom" // pour charger la bibliothèque aléatoire.
srandomdev () // permet de donner une nouvelle génération
| | | // pour chaque exécution du programme.
random() // pour générer un nombre réel ou entier aleatoire.
randint32() // pour générer un entier aléatoire.
randint32()%k // pour générer un entier aléatoire entre 1-k et k-1.
abs(randint32()%k) // pour générer un entier positif aléatoire entre 0 et k-1.
randreal1() // pour générer un nombre réel aléatoire dans l'intervalle [0,1]
randreal1 ()*k // pour générer un nombre réel aléatoire dans l'intervalle [0,k]
floor() // donne la partie entière

```



```
1  load "lapack"
2  load "fflapack"
3  load "ffrandom"
4  srandomdev();
5  int a,b;
6  real c,d,e,r,f;
7  a = randint31();
8  cout <<"a="<<a<<endl;
9  b = randint32();
10 cout <<"b="<<b<<endl;
11 c = randreal1();
12 cout <<"c="<<c<<endl;
13 d = randreal2();
14 cout <<"d="<<d<<endl;
15 e = randres53();
16 cout <<"e="<<e<<endl;
17 r = randreal3();
18 cout <<"r="<<r<<endl;
19 f = abs(randint32()%2);
20 cout <<"f=" << f <<endl;
```

Matrice Toeplitz :

```

1  int n;
2  int i;
3  int j;
4  cout << "Donner la taille de la matrice "<< endl;
5  cin >> n ;
6  int [int] l=(n);
7  int [int] c=(n);
8  real [int,int]Toeplitz(n,n);
9  for(i=0;i<n;i++){
10 cout << "Donner l(" << i<<")" << endl ;
11 cin>> l(i) ;
12 }
13 cout << "Le vecteur est " <<"l=" << l << endl ;
14 c(0)=l(0);
15 for(j=1;j<n;j++){
16 cout << "Donner c("<< j << ") "<<endl ;
17 cin >> c(j) ;
18 }
19 cout << "Le vecteur est " << "c=" << c <<endl;
20 Toeplitz=0;
21 cout << "La matrice Toeplitz est " << "Toeplitz =" << Toeplitz<< endl ;
22     for(i=0;i<n;j++){
23         for(j=0;j<n;j++){
24             if(i==0){

```

Matrice Toeplitz :

```

15  for(j=1;j<n;j++){
16  cout << "Donner c("<< j << ")"<<endl ;
17  cin >> c(j) ;
18  }
19  cout << "Le vecteur est " << "c=" << c <<endl;
20  Toeplitz=0;
21  cout << "La matrice Toeplitz est " << "Toeplitz =" << Toeplitz<< endl ;
22  for(i=0;i<n;j++){
23      for(j=0;j<n;j++){
24          if(i==0){
25              Toeplitz(j,i)=1(j);
26              Toeplitz(i,j)=c(j);
27          }
28          if(i<n-1 && j<n-1){
29              Toeplitz(i+1,j+1)=Toeplitz(i,j);
30          }
31      }
32  }
33  cout << "La matric Toeplitz est " << "Toeplitz=" <<Toeplitz << endl;

```

```

1  int n;
2  int i;
3  int j;
4  cout << "Donner la taille de la matrice "<< endl;
5  cin >> n ;
6  real [int,int]Hilbert(n,n);
7  Hilbert=0;
8  for(i=0;i<Hilbert.n;i++)
9      for(j=0;j<Hilbert.n;j++)
10         Hilbert(i,j)=1./((i+1)+(j+1)-1);
11
12  cout << "La matric Hilbert est " << "Hilbert=" <<Hilbert << endl;

```

Matrice Magique :

```

1  int n;
2  int i;
3  int j;
4  cout << "Donner la taille de la matrice "<< endl;
5  cin >> n ;
6  real [int,int] Magique(n,n);
7  Magique=0;
8  i=n/2;
9  j=n-1;
10 Magique(i,j)=1;
11 for (int s = 2 ; s< n^2+1 ; ++s){
12     i=i-1;
13     j=j+1;
14     if (i== -1 & j==n){
15         i=0;
16         j=n-2;
17     }
18     if (i== -1 & j !=n){
19         i=n-1;
20     }
21     else { if (j==n & i!=-1){
22         j=0;
23     }
24     else { if (Magique(i,j)!=0){
25         i=i-1;

```

Matrice Magique :

```

14     if (i==-1 & j==n){
15         |   i=0;
16         |   j=n-2;
17     }
18     if (i==-1 & j !=n){
19         |   i=n-1;
20     }
21     else { if (j==n & i!=-1){
22         |   j=0;
23     }
24     else { if (Magique(i,j)!=0){
25         |   i=i+1;
26         |   j=j-2;
27         |   }
28     }
29 }
30 Magique(i,j)=s;
31 }
32 cout <<"Magique"<<Magique<<endl;
33 cout <<"i"<<i<<endl;
34 cout <<"j"<<j<<endl;

```

La méthode de Cholesky :

```

1  int n=4;
2  real[int,int] Cholesky(n,n);
3  int i,j,k;
4  real s;
5  real[int,int] L(n,n);
6  real[int,int] l(n,n);
7  real[int] x(n),b(n),y(n);
8  b=[1,2,4,5];
9  Cholesky=[[1,1,1,1] ,
10 [1,5,5,5] , [1,5,14,14] , [1,5,14,15]];
11 L=0;
12 L(0,0)=sqrt(Cholesky(0,0));
13 for (i=1;i<n;i++){
14     L(i,0)=Cholesky(i,0)/L(0,0);
15 }
16 for(j=1;j<n;j++){
17     s=0;
18     for (k=0;k<=j-1;k++){
19         s=s+L(j,k)^2;
20     }
21     L(j,j)=sqrt (Cholesky(j,j)-s);
22     for (i=j;i<n;i++){
23         s=0;
24         for (k=0;k<=j-1;k++){

```

La méthode de Cholesky :

```

16  for(j=1;j<n;j++){
17      s=0;
18      for (k=0;k<=j-1;k++){
19          s=s+L(j,k)^2;
20      }
21      L(j,j)=sqrt (Cholesky(j,j)-s);
22      for (i=j;i<n;i++){
23          s=0;
24          for (k=0;k<=j-1;k++){
25              s=s+L(i,k)*L(j,k);
26          }
27      L(i,j)=(Cholesky(i,j)-s)/L(j,j);
28      }
29  }
30  for (i=0;i < n;i++)
31  for (j=0 ;j < n;j++)
32  l(i,j)=L(j,i);
33  cout << "La factorisation de Cholesky est comme suit "<< endl ;
34  cout <<"L="<<endl ;
35  cout <<L<<endl;
36  cout <<"l="<<endl ;
37  cout <<l<<endl;

```


Décomposition LU :

```

1  int n,i,j,k;
2  real d ;
3  cout << "Donner la taille de la matrice " << endl;
4  cin >>n;
5  real [int,int] A(n,n);
6  real [int,int] L(n,n);
7  real [int,int] U(n,n);
8  L=0;
9  for(i=0;i<n;i++) L(i,i)=1;
10 int [int] Y=(n); Y=0;
11 int [int] X=(n); X=0;
12 int [int] B=(n); B=0;
13     for (i=0;i<n;i++){
14         for (j=0;j<n;j++){
15             cout <<"Donner A("<<i+1<<","<<j+1<<")"<<endl;
16             cin >> A(i,j);
17         }
18     }
19 cout << "A="<<A<<endl;
20     for(i=0;i<n;i++){
21         cout <<"Donner B("<<i+1<<")"<<endl;
22         cin >> B(i);
23     }
24 cout << "B="<< B<< endl;
25 U=A;

```

Décomposition LU :

```

18     }
19     cout << "A="<<A<<endl;
20     for(i=0;i<n;i++){
21         cout <<"Donner B("&<<i+1<<")"<<endl;
22         cin >> B(i);
23     }
24     cout << "B="<< B<< endl;
25     U=A;
26     for (k=0;k<=n-1;k++){
27         L(k+1:n-1,k)=U(k+1:n-1,k)*(1/U(k,k));
28         for (j=k+1;j<=n-1;j++){
29             U(j,k:n-1)=U(j,k:n-1)-(L(j,k)*U(k,k:n-1));
30         }
31     }
32     for(i=0;i<n;i++){
33         for(j=0;j<n;j++){
34             if(i>j)U(i,j)=0;
35         }
36     }
37     cout << "L="<<L<<"U="<<U<<endl;
38

```

```
1  int i,j,n; real X,Lg=0,L;
2  cout << "Veuillez saisir le nombre des points : "<< endl;
3  cin >> n;
4  real [int] x(n+1),y(n+1);
5  for(int i=1;i<=n;i++){
6      cout << "Donner l'abscisse x("<<i<<")="<<endl;
7      cin>>x(i);
8      cout << "Donner l'abscisse y("<<i<<")="<<endl;
9      cin>>y(i);
10     cout<<endl;
11 }
12 cout << "Entrer le point d'interpolation : " << endl;
13 cin >> X;
14 for(int i=1;i<=n;i++){
15     L=1;
16     for(int j=1;j<=n;j++){
17         if(i!=j){
18             L*=(X-x(j))/(x(i)-x(j));
19         }
20     }
21     Lg =Lg +L*y[i];
22 }
23 cout << "L'interpolation de la valeur " <<X<< "est"<< Lg<< endl;
```

```
1  int n,i; real a,b,I2,h;
2  cout << "entrer la valeur de a "<< endl;
3  cin>>a;
4  cout << "entrer la valeur de b "<< endl;
5  cin>>b;
6  cin>> b;
7  if(a>b){
8  cout<<"erreue"<<endl;
9  }
10 else {
11 cout << "Entrer le nombre de subdivision n " << endl;
12 cin>> n;
13 cout << endl;
14 func real f(real x){
15 return cos(x);
16 }
17 h=(b-a)/n;
18 int k ;
19 I2=(f(a)+f(b))*h/3;
20 for(k=1;k<=n-1;k++){
21     if (k%2==0){
22         I2+=(f(a+(k*h)))*(2*h)/3;
23     }
24 else
```

```
14 func real f(real x){
15     return cos(x);
16 }
17 h=(b-a)/n;
18 int k ;
19 I2=(f(a)+f(b))*h/3;
20 for(k=1;k<=n-1;k++){
21     if (k%2==0){
22         I2+=(f(a+(k*h)))*(2*h)/3;
23     }
24     else
25     {
26         I2+=(f(a+(k*h)))*(4*h)/3;
27     }
28 }
29 cout << "le nombre de subdivision est n=" << n<< endl;
30 cout << endl;
31 cout << "l'integrale par la methode de Simpson sur ["<< a<< ", "<< b<< "]" egale "<< I2<< endl;
32 cout << endl ;
33 }
```

```
1  int n,i; real a,b,I1,h;
2  cout << "entrer la valeur de a "<< endl;
3  cin>>a;
4  cout << "entrer la valeur de b "<< endl;
5  cin>>b;
6  cin>> b;
7  if(a>b){
8  cout<<"erreue"<<endl;
9  }
10 else {
11 cout << "Entrer le nombre de subdivision n " << endl;
12 cin>> n;
13 cout << endl;
14 func real f(real x){
15 return cos(x);
16 }
17 I1=(f(a)+f(b))*(h/2);
18 h=(b-a)/n;
19 for(i=1;i<n;i++){
20 I1+=f(a+(i*h))*h;
21 }
22 cout << "le nombre de subdivision est n=" << n<< endl;
23 cout << endl;
24 cout << "l'integrale par la methode de Trapeze sur ["<< a<< ", "<<b<<"] egale "<< I1<< endl;
25 cout << endl ;
26 f
```