Université de Lorraine - UFR MIM - 2015/2016
Cours MATLAB

MATLAB 6

Interpolation

J-P. Croisille

1- Interpolation spline (1)

1) Taper le programme suivant, qui trace la fonction $x \mapsto \ln(x)$ sur $]0,1]$, ainsi que $N = 4$ points sur cette courbe.

```
%% INTERPOLATION SPLINE DE LA FONCTION X |--> LOG(X) SUR [0.01,1.01]
%%
a=0.01 ; b=1.01 ; dx=0.01; % EXTREMITES DE L'INTERVALLE, PAS D'ESPACE
x = a:dx:b;  % TABLEAU DES ABSCISSES POUR X--> LOG(X)
y = log(x);  % TABLEAU DES ORDONNEES POUR X-->LOG(X)
figure(1); plot(x,y,'b-'); hold on; % TRACE DE LA FONCTION X--> LOG(X) EN BLEU
n=5;  dx=(b-a)/(n-1);
xx = a:dx:b; yy = log(xx); % CHOIX DE 4 POINTS SUR LE GRAPHE DE X-->LOG(X)
plot(xx,yy,'ro'); hold on;
```

2) Tracer sur le même graphique la spline qui interpole les quatre valeurs par `yyy = spline(xx,yy,x);`.
3) Recommencer la même opération avec $N = 10$ points, puis $N = 20$ points. Qu'observe-t-on ?
4) Pour une valeur donnée $\bar{x} \in]0,1]$, calculer l'erreur entre $\ln(\bar{x})$ et l'approximation par la spline dans chacun des trois cas.

2- Interpolation spline (2)

1) Editer le programme suivant:

```
%% INTERPOLATION SPLINE DE LA FONCTION X |--> SIN(2*PI*X) SUR [0,1]
%%
clear all; % VIDE LA MEMOIRE
a=0 ; b=2*pi ; np=500; dx=(b-a)/(np-1);
x = a:dx:b;
nn=4; %% FREQUENCE DES OSCILLATIONS
y = sin(nn*x);
figure(1);
n=10; dx=(b-a)/(n-1); %% NOMBRE DE POINTS D'INTERPOLATION SUR LA SPLINE.
xx = a:dx:b;
yy = sin(nn*xx);
yyy = spline(xx,yy,x);
plot(x,y,xx,yy,'ro',x,yyy,'g'); grid on;
```

```
title ('bleu: fonction exacte, vert: interpolee spline cubique');
```

.

Que fait ce programme ?
2) Augmenter progressivement le nombre de points sur la spline en changeant la valeur de `n`. Que se passe-t-il ?
3) Fixer ensuite `n=10` et augmenter progressivement la valeur de `nn`, la fréquence ds oscillations. Que se passe-t-il ? Pourquoi ?

3- Interpolation de Lagrange

1) Editer la fonction `matlab` suivante qui évalue le polynome de Lagrange au point `y`.

```
function ev = evalue(f,x,y)
% INPUT: X[1:N] : TABLEAU DES ABSISSES
%        F[1:N] : TABLEAU DES MESURES
%             Y : VALEUR OU LE POLYNOME DE LAGRANGE EST EVALUE.
% OUTPUT:     EV : EVALUATION DU POLYNOME DE LAGRANGE AU POINT Y.

n=size(x,2);
c=newton(x,f,n);
pn=c(n);
for k=(n-1):-1:1
   pn=pn*(y-x(k))+c(k);
end
ev = pn;
```

. (Utiliser le programme `newton.m`).

2) Editer la fonction `main.m` suivante:

```
function main(x)
t=-5:.01:5; % ABSISSES POUR LE TRACE DE LA FONCTION DE RUNGE
f=1./(1+x.^2); % FONCTION DE RUNGE
for i=1:size(t,2)
    ordonnee(i)=evalue(f,x,t(i));
end,
ff=1./(1+t.^2);
plot(t,ff,'r',t,ordonnee,'go')
title('rouge=fonction de Runge, vert=interpolee de Lagrange');
grid;
```

(On utilisera le programme `evalue.m`.)

3) On considère la fonction de Runge sur l'intervalle $x \in [-5,5] \mapsto 1/(1+x^2)$. Editer le script suivant qui trace la fonction de Runge et le polynôme interpolé de Lagrange dans deux cas. Visualiser les fichiers postscript créés.

```
%% SCRIPT POUR TESTER L'INTERPOLEE DE LAGRANGE AVEC LA FONCTION DE
%RUNGE
% 1- POINTS EQUIDISTANTS, DE -5 A 5, PAS 2.
figure(1);
x=-5:2:5;
main(x);
print -depsc fig1.ps; % IMPRESSION DANS LE FICHIER POSTSCRIPT 'FIG1.PS'
% 2- POINTS EQUIDISTANTS, DE -5 A 5, PAS 1.
figure(2);
```

```
x=-5:1:5;
main(x);
print -depsc fig12.ps; % IMPRESSION DANS LE FICHIER POSTSCRIPT 'FIG2.PS'
```

4) Continuer le script avec les cas suivants:

- Points d'interpolation de -5 à 5 avec incrément 1/2.
- Points d'interpolation `x=[-5 -4.5 -4 -3.5 -3 -1 1 3 5];`.
- Points d'interpolation donnés par les 4 points de Tchebychef:
  `n=4;x=pi*[0:n]; x=pi-x/(n+1);x=5*cos(x);`
- Même chose avec `n=10`, puis `n=20`.

Qu'observe-t-on ?