Interpolation de Lagrange : mise en évidence du phénomène de Runge

Grégory Vial

5 mars 2003

Le but de ces pages est de présenter quelques programmes matlab qui permettent de calculer le polynôme d'interpolation de Lagrange d'une fonction. On met en évidence le *phénomène de Runge* (pour la théorie, voir [2], par exemple) dans le cas des points équidistants, et la convergence pour les abscisses de Tchebytchev.

En dimension 1

Soit f une fonction régulière définie sur un intervalle $I \subset \mathbb{R}$. On note (x_i) des points de I, $0 \le i \le n$ et p_n le polynôme d'interpolation de f aux abscisses x_i. L'erreur d'interpolation est donnée par (voir [1], par exemple) :

$$\forall\, x \in I \quad \exists\, \xi_x \in I \quad f(x) - p_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\,\pi_n(x),$$

où $\pi_n(x) = (x - x_0)\cdots(x - x_n)$.

Si f est une fonction analytique, alors on peut estimer la croissance des dérivées $f^{(n+1)}(\xi_x)$ en fonction de n et montrer que p_n converge vers f dans un intervalle dont la largeur dépend du rayon de convergence (voir [2]). Quand le rayon de convergence est infini, il y a convergence uniforme sur I tout entier.

La fonction `lagr.m` qui suit calcule le poynôme d'interpolation de la fonction exponentielle aux abscisses `x` ; les graphes de la figure 1 sont obtenus avec le script `main.m`.

```
function valp=lagr(valf,x,t)
% évaluation en t du polynôme de
% Lagrange de f aux abscisses x.
% x doit être colonne
A=fliplr(vander(x));
p=A\valf;
valp=polyval(flipud(p),t);
    ----------------
function y=f(x)
y=exp(x)
    ----------------
a=0;b=2;
aa=-1;bb=3;n=4;
x=linspace(a,b,n+1);
t=linspace(aa,bb,100);
valp=lagr(f(x'),x',t);
plot(t,valp,t,f(t),x,f(x),'ro')
legend('p_n','exponentielle'
  ,'pts d''interp.')
title('Polynôme de Lagrange de
  l''exponentielle pour n=4')
nn=5:20;

err=[];
t=linspace(a,b,100);
for n=nn;
  x=linspace(a,b,n+1);
  valp=lagr(f(x'),x',t);
  err=[err,norm(valp-f(t),'inf')];
end
plot(nn,err)
title('Erreur d''interpolation')
```

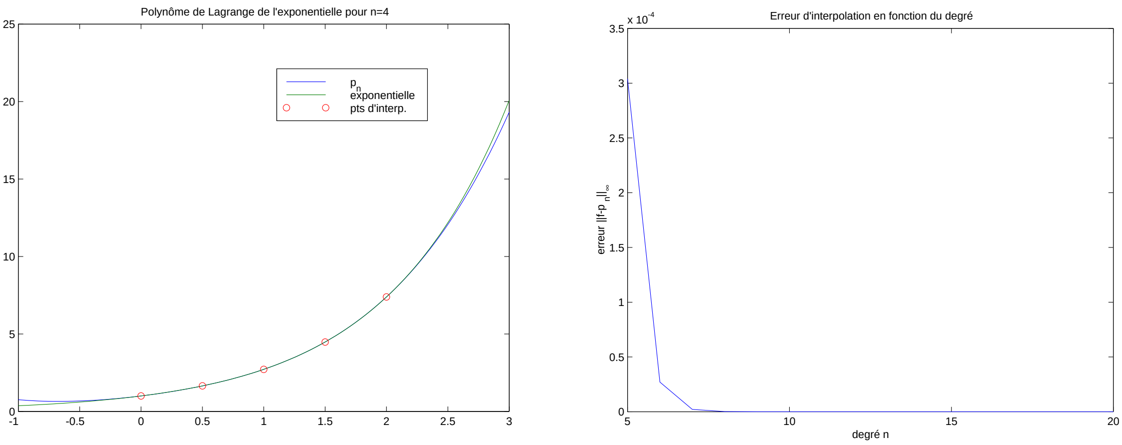

FIG. 1 – Convergence du polynôme d'interpolation vers l'exponentielle.

On a ainsi mis en évidence la convergence uniforme du polynôme d'interpolation de Lagrange vers la fonction exponentielle sur l'intervalle $[0, 2]$.

Le phénomène de Runge

On s'intéresse maintenant à la fonction f définie sur $[-5, 5]$ par

$$f(x) = \frac{1}{1 + x^2}.$$

Le script suivant permet de construire les courbes de la figure 2.

```
a=-5;b=5;
aa=-1;bb=3;
t=linspace(a,b,100);
nn=5:20;
errU=[];errT=[];
for n=nn;
  xU=linspace(a,b,n+1);
  xT=(a+b)/2+(b-a)/2*cos(pi/(n+1)*(1/2+[0:n]));
  valp=lagr(f(xU'),xU',t);
  valq=lagr(f(xT'),xT',t);
  errU=[errU,norm(valp-f(t),'inf')];
  errT=[errT,norm(valq-f(t),'inf')];
end
plot(nn,errU)
title('Erreur d''interpolation aux abscisses équidistantes')
xlabel('degré n')
ylabel('erreur ||f-p_n||_\infty')
figure
plot(nn,errT)
title('Erreur d''interpolation aux abscisses de Tchebytchev')
xlabel('degré n')
ylabel('erreur ||f-p_n||_\infty')
```

On observe la convergence uniforme du polynôme d'interpolation aux abscisses de Tchebytchev (graphe de droite) et la divergence pour les points uniformes.

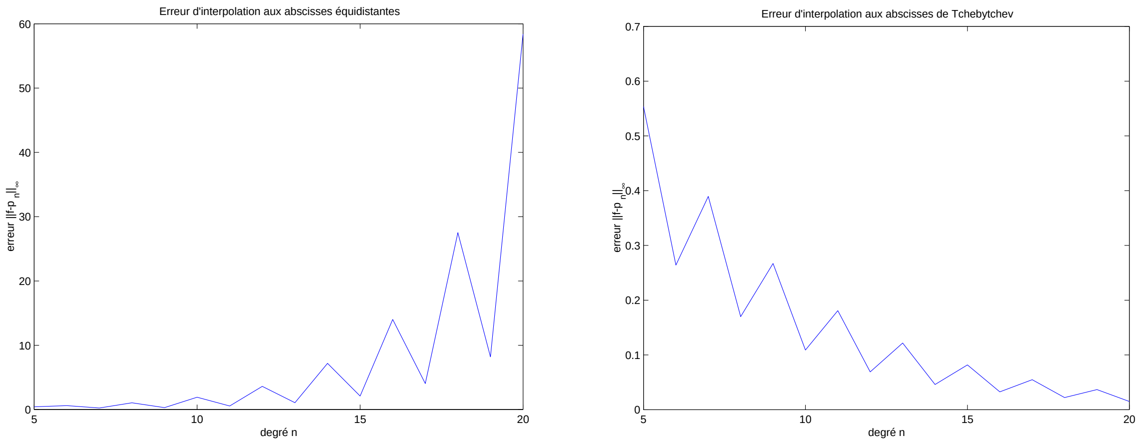

FIG. 2 – Erreur d'interpolation en fonction du degré.

Le programme suivant fournit une simulation intéressante pour montrer l'évolution du polynôme d'interpolation en fonction du degré, à la fois pour les abscisses équidistantes et de Tchebytchev.

```
a=-5;b=5;
t=linspace(a,b,100);
nn=5:20;
figure
subplot(2,1,1)
axis([-5,5,-1,2])
hold on
plot(t,f(t))
subplot(2,1,2)
axis([-5,5,-1,2])
hold on
plot(t,f(t))
for n=nn;
  xU=linspace(a,b,n+1);
  xT=(a+b)/2+(b-a)/2
    *cos(pi/(n+1)*(1/2+[0:n]));
  valp=lagr(f(xU'),xU',t);
  valq=lagr(f(xT'),xT',t);
  pause
  clf
  subplot(2,1,1)
  axis([-5,5,-1,2])
  hold on
  plot(t,f(t))
  plot(t,valp,'r')
  title('Abscisses uniformes')
  text(0,0,['n=',num2str(n)])
  text(-1,1.5,['||f-p_n||_\infty='
    ,num2str(norm(f(t)-valp,'inf'))]);
  subplot(2,1,2)
  axis([-5,5,-1,2])
  hold on
  plot(t,f(t))
  plot(t,valq,'r')
  title('Abscisses de Tchebytchev')
  text(0,0,['n=',num2str(n)])
  text(-1,1.5,['||f-p_n||_\infty='
    ,num2str(norm(f(t)-valq,'inf'))]);
end
```

La figure 3 donne l'état de la fenêtre graphique à la fin de l'exécution. Quelques points concernant la programmation méritent d'être soulignés :

- la commande `subplot` sert à découper la fenêtre graphique ;
- la commande `axis` permet de conserver la même échelle pour tous les graphes ;
- la commande `num2str` transforme un nombre en chaîne de caractère.

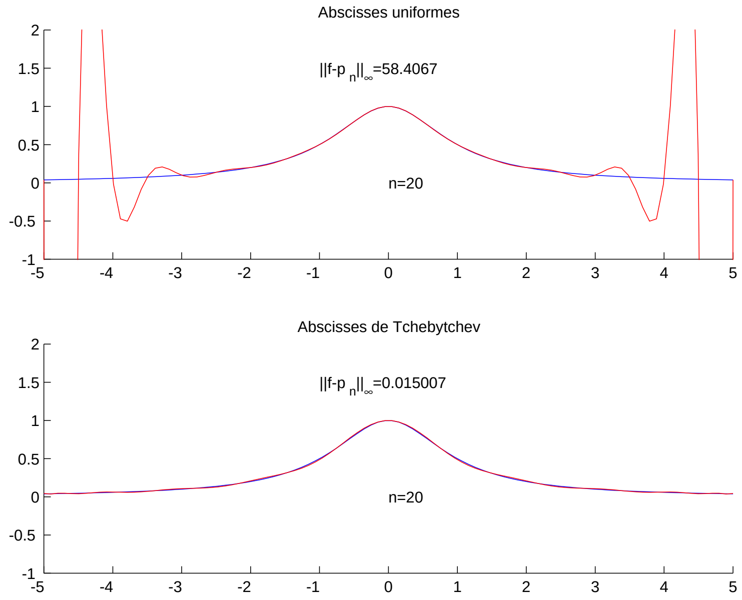

Fig. 3 – Le phénomène de Runge.

En dimension 2

On pourrait penser que le phénomène de Runge n'apparaît que pour la fonction f considérée au paragraphe précédent. Bien au contraire, il se manifeste dans presque toutes les applications pratiques.

Considérons le problème suivant : étant donnés $n+1$ points $M_i = (x_i, y_i)$ dans le plan $\mathbb{R}^2$, on cherche à tracer une courbe lisse qui les relie. Les nombres x_i peuvent être vus comme les évaluations d'une fonction x en i (de même y_i) ; on peut utiliser une interpolation pour définir les fonctions x et y sur l'intervalle continu $[1, n+1]$ et construire ainsi une courbe paramétrée qui relie les points M_i.

```
% On clique avec le bouton
% de droite le dernier point.
figure
plot([0,1,1,0],[0,0,1,1],'k')
hold on
button=1;
xx=[];yy=[];
while (button~=3)
  [x,y,button]=ginput(1);
  plot(x,y,'ro')
  xx=[xx;x];
  yy=[yy;y];
end
n=length(xx);
% abscisses équidistantes
t=linspace(1,n,100);
xt=lagr(xx,1:n,t);
yt=lagr(yy,1:n,t);
% abscisses de Tchebytchev
t=linspace(cos(pi/(2*n))
  ,cos(pi/n*(1/2+n-1)),100);
tch=cos(pi/(n)*(1/2+[0:n-1]));
xtt=lagr(xx,tch,t);
ytt=lagr(yy,tch,t);
% Tracé
clf
plot(xx,yy,'ro')
hold on
plot(xt,yt,'b',xtt,ytt,'g--')
legend('Points d''interpolation'
  ,'Abscisses uniformes'
  ,'Abscisses de Tchebytchev')
```

La figure 4 présente une réalisation du script ci-dessus.

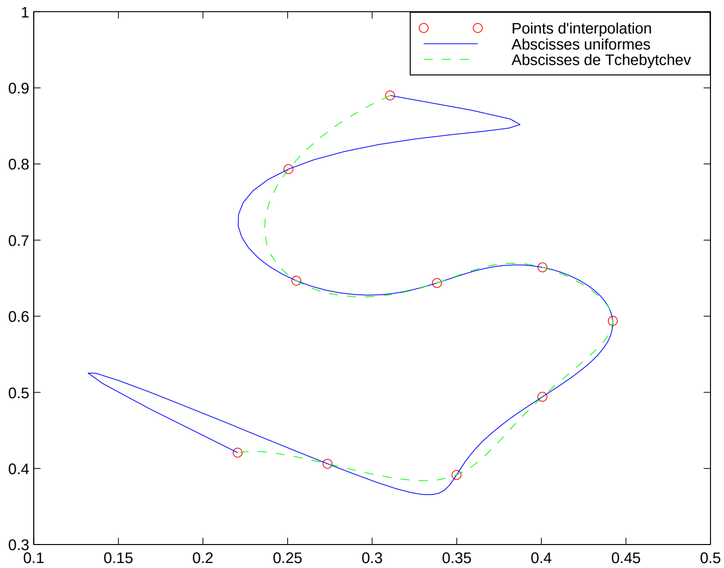

FIG. 4 – Le phénomène de Runge en dimension 2.

On observe, comme en dimension 1, que le choix des points de Tchebychev permet d'obtenir un résultat satisfaisant alors que les abscisses uniformes conduisent à une méthode divergente. Dans le cas bidimensionnel présenté ici, ces abscisses interviennent dans le paramétrage, qui est choisi de façon arbitraire : il est donc indiqué d'utiliser les points de Tchebytchev.

Références

[1] M. CROUZEIX, A. L. MIGNOT. *Analyse numérique des équations différentielles*. Collection mathématiques appliquées pour la maîtrise. Masson, Paris 1984.

[2] J.-P. DEMAILLY. *Analyse numérique et équations différentielles*. PUG, Grenoble 1996.