# MÉTHODES NUMÉRIQUES

*Laboratoire 9*

**MODÉLISATION DE DONNÉES PAR DES COURBES D'INTERPOLATION**

**DIFFÉRENCE ENTRE RÉGRESSION ET INTERPOLATION**
**==========================================**

```
clear all
% Les données
x=0:10;
y=sin(x);
% Il génère un ensemble plus fin de données indépendantes
xi=0:0.25:10;

% Interpolation linéaire
yi=interp1(x,y,xi);
   plot(x,y,'m*',xi,yi,'m')
   hold on
   grid
   text(0.5,0.25,' Interpolation linéaire’)

% Interpolation par intervalles, de type "spline"
yi=spline(x,y,xi);
   plot(xi,yi,'g')
   text(2.5,0.75,' Interpolation spline "spline"')

% Régression polynomial
p=polyfit(x,y,5);   % génère des coefficients polynomiaux
yi=polyval(p,xi);   % calcule les valeurs du polynôme en xi
   plot(xi,yi,'--c')
   text(1.25,1.2,' Régression polynomiale')
```

**DIFFÉRENTES COURBES D'INTERPOLATION ET LE DANGER D'EXTRAPOLATION**
**=========================================================**

```
clear all
% Données initiales connues
x=1:0.5:10;
y=log(x);
   figure(1)
   plot(x,y,'r*')
   hold on
   text(2,-1,'* Données initiales complètes')

% Modélisez les données de l'ensemble ci-dessous avec un polynôme
% d'ordre 4, utilisant la méthode de Newton des differences
%divisées
xi=1:5; yi=log(xi);
b0=yi(1);
   for i=1:(length(xi)-1),
      dif1(i)=(yi(i+1)-yi(i))/(xi(i+1)-xi(i));
   end
b1=dif1(1);

   for i=1:(length(dif1)-1),
      dif2(i)=(dif1(i+1)-dif1(i))/(xi(i+2)-xi(i));
   end
b2=dif2(1);
   for i=1:(length(dif2)-1),
      dif3(i)=(dif2(i+1)-dif2(i))/(xi(i+3)-xi(i));
```

```
   end
b3=dif3(1);
   for i=1:(length(dif3)-1),
      dif4(i)=(dif3(i+1)-dif3(i))/(xi(i+4)-xi(i));
   end
b4=dif4(1);

ynewton=b0+b1.*(x-xi(1))+b2.*(x-xi(1)).*(x-xi(2))+...
        b3.*(x-xi(1)).*(x-xi(2)).*(x-xi(3))+...
        b4.*(x-xi(1)).*(x-xi(2)).*(x-xi(3)).*(x-xi(4));
plot(x,ynewton,'g')
text(2,-2,'- vert - Newton divisé différent / ensemble réduit')

% Modéliser les données ci-dessus avec un polynôme d'ordre 4,
% en utilisant la méthode polynomiale de Lagrange
ylagrange=0;
   for i=1:length(xi),
      LL=[];
      for j=1:length(xi),
         if i~=j
          LL=[LL; (x-xi(j))./(xi(i)-xi(j))];
         end
      end
      L=prod(LL);
      ylagrange=ylagrange+yi(i).*L;
   end
plot(x,ylagrange,'m+')
text(2,-3,'+ violet - Polynômes Lagrange/ensemble réduit')

% Modéliser les données de l'ensemble réduit avec un polynôme
%d'ordre 4, utilisant le polynôme général d'interpolation

Z=[1 1 1 1 1; xi; xi.^2; xi.^3; xi.^4]'; % Matrice des coefficients
C=yi;              % Vecteur de termes libre
A=inv(Z)*C';       % Les coefficients du polynôme du 4e degré

% Comparer A avec P, obtenu par la commande "polyfit" (trouver
% l'ordre inverse des coefficients)
P=polyfit(xi,yi,4);

% Restaurez le polynôme pour l'ensemble de données complet:
p=polyfit(x,y,4);  % génère des coefficients polynomiaux
yp=polyval(p,x);   % calcule les valeurs du polynôme en xi
plot(x,yp,'--c')
text(2,-4,'-- bleu - Polynôme général d'interpolation/ensemble
%complet')

% ou d'autres courbes d'interpolation
ys1=interp1(x,y,x,'linear');     % interpolation linéaire
ys2=interp1(x,y,x,'nearest');    % interpolation à travers le point
                                 % le plus proche
ys3=interp1(x,y,x,'cubic');      % interpolaire cubique
ys4=interp1(x,y,x,'spline');     %spline cubique interpolaire
figure(2)
plot(x,y,'r*',x,ys1,x,ys2,x,ys3,x,ys4)
```