Les élèves sont conduits à expérimenter les méthodes d'interpolation et de régression pour modéliser l'évolution d'une valeur boursière de 1996 à nos jours.

Les points de mesures qui serviront à cette étude correspondent au cours affiché au 1^er^ janvier de chaque année. Ces valeurs sont consignées dans les tableaux [année] et [valeur] du fichier ANTP3.m.

Extrait de http://bourse.latribune.fr/: Evolution du CAC40 depuis 1996.

I. INTERPOLATION POLYNOMIALE

SPLINES CUBIQUES DE MATLAB

Q1. Le programme ANTP3.m ci-contre permet d'interpoler de nos observations par la méthode des splines cubiques.

- Rappeler le principe de cette méthode.
- Identifier les variables utilisées dans le programme.
- Exécuter ce programme et commenter l'allure de la courbe obtenue.

```
% ------------------- SPLINES DE MATLAB -
for x=1:(12*(N-1))
Ps(x) = interp1(mois,valeur,x,'spline');
temp(x)=x-1;
end;
figure(1);
subplot(221); %axis([0 12*13 0 8]);
plot(temp,Ps,'r',mois,valeur,'*'); grid;
title('Splines cubiques de Matlab');
```

METHODE DE LAGRANGE

Q2. La trame de l'algorithme de l'interpolation de Lagrange est présentée ci-dessous.

- Rappeler la méthode de Lagrange et compléter ce programme.
- Comparer le résultat de cette interpolation avec celle des splines cubiques de MATLAB.
- Estimer sans faire de calcul, un encadrement de l'ordre du polynôme de Lagrange.
- Justifier les phénomènes de « bords » sur la représentation graphique de $P_L(x)$.

METHODE DE NEWTON

Q3. Réaliser le même travail avec la méthode de Newton. Quelle est la fonction des deux grandes boucles for ?

```
% ---------------- LAGRANGE --------------------------
l=zeros(1,N);
for x=1:(12*(N-1))
    somme=0;
    for i=1:N
        produit=1;
        for j=1:N
            if (j~=i)        % Test i<>j
            produit=produit*(x-....(j))/(....(i)-mois(
            end;
            l(i)=produit;
        end;
        somme=somme+valeur(i)*l(i);
    end;
    Pl(x)=somme;   temp(x)=x;
end;

subplot(222);
plot(temp,Ps,'r',temp,Pl,'g',mois,valeur,'*');
title('Splines cubiques / Lagrange');
grid;
```

```
% ---------------- NEWTON ----------------------

a=......;    %-- Calcul des coeff par diff. div.
for k=1:(N-1)
    for i=k+1:N
        a(i)=(a(i)-a(k))/(mois(i)-....(k));
    end;
end;
a

for x=1:(12*(N-1))   % ---  evaluation pour x
    Pn(x)=a(N);
    for i=1:(N-1)
        Pn(x)=a(N-i)+(x-mois(N-i))*Pn(x);
    end;
end;

subplot(223);
plot(temp,Ps,'r',temp,Pn,'b',mois,valeur,'*');
title('Splines cubiques / Newton');
grid;
```

II. APPROXIMATION POLYNOMIALE : REGRESSION

PHASE DE TRIANGULATION

Q4. Nous esquissons dans la fenêtre ci-contre la modélisation de nos mesures au moyen d'une approximation polynomiale.

- Compléter de programme.
- Etablir (ou rappeler) l'expression théorique de la matrice A et du vecteur b.
- Quelle est la variable qui représente le degré du polynôme d'approximation ?
- Quel est le tableau qui consigne les coefficient de ce polynôme ?
- Exécuter le programme pour plusieurs valeur du degré de ce polynôme (1<degré<12).
- Relever la représentation graphique de ce polynôme et commenter les éventuels Warning générés dans la fenêtre de commande de MATLAB.
- Quelle est l'approximation qui offre la meilleure précision ?

```
% ---------------- APPROX --------------------------

m=4; a=zeros(m+1,1);
A=zeros(m+1,m+1); b=zeros(m+1,1); s=zeros(2*m+1,1);

for i=1:N
    tempo=1;
    for j=1:m+1
        b(j)=b(j)+valeur(i)*tempo;
        tempo=tempo*mois(i);
    end;
    tempo=1;
    for j=1:2*m+1
        s(j)=s(j)+tempo;
        tempo=tempo*mois(i);
    end;
end;

for i=1:m+1
    for j=1:m+1
        A(i,j)=s(i+j-1);
    end;
end;
a=inv(.)*.

for x=1:(12*(N-1))
    Pa(x)=0;
    for k=0:m
        Pa(x)=..................;
    end;
end;

subplot(224); axis([0 12*13 0 8]);
plot(temp,Pa,'r',mois,valeur,'*'); grid;
title('Approximation polynomiale');
```