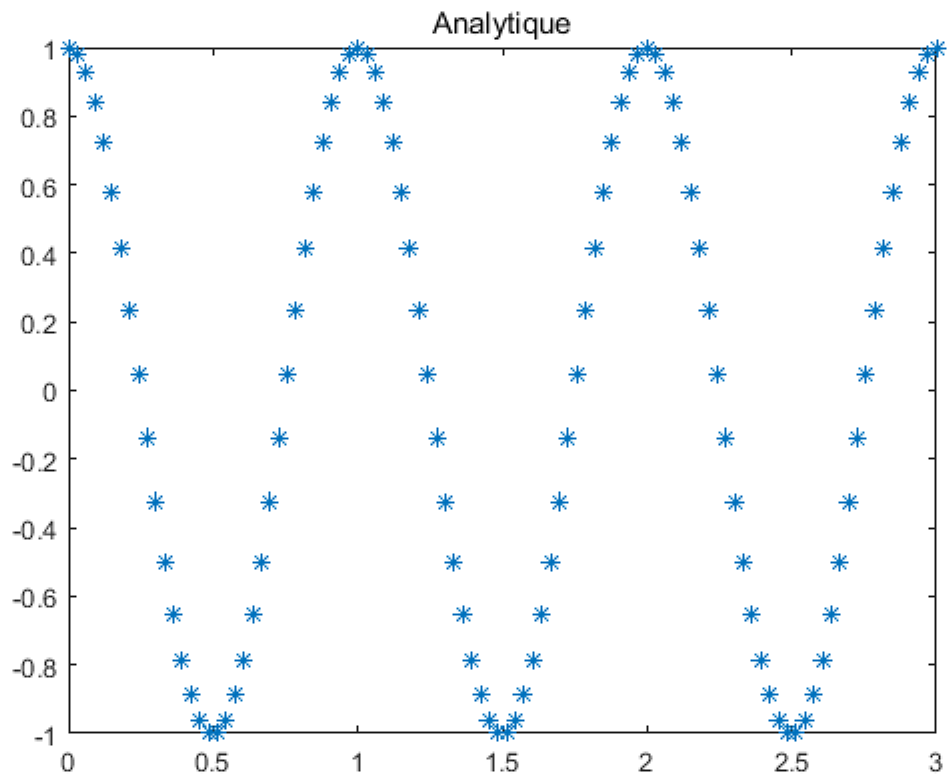
Table of Contents

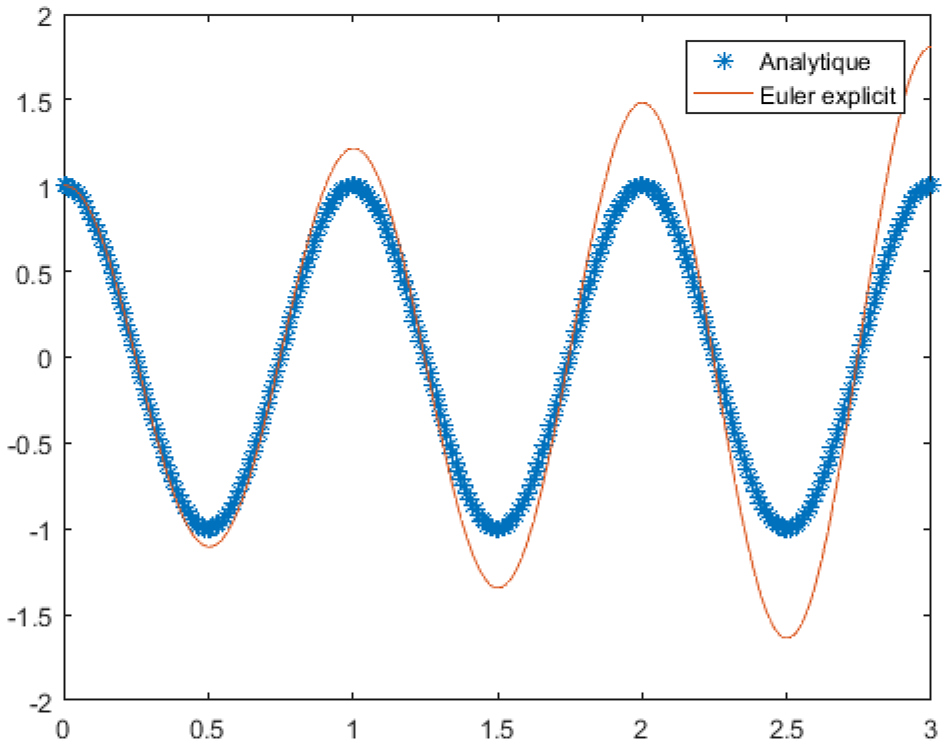
Q1

```
W0 = 2*pi;
q0 = 1;
Dq0 = 0;
T0 = 3;
%%q = a*sin(W0*t) + b * cos(W0*t)
%on obtenir a = 0 b = 1 en fonction de CDI
t = linspace(0,T0,100);
q = cos(W0*t);
plot(t,q,'*');
title('Analytique');
%E est une constant
E = W0^2/2;
```



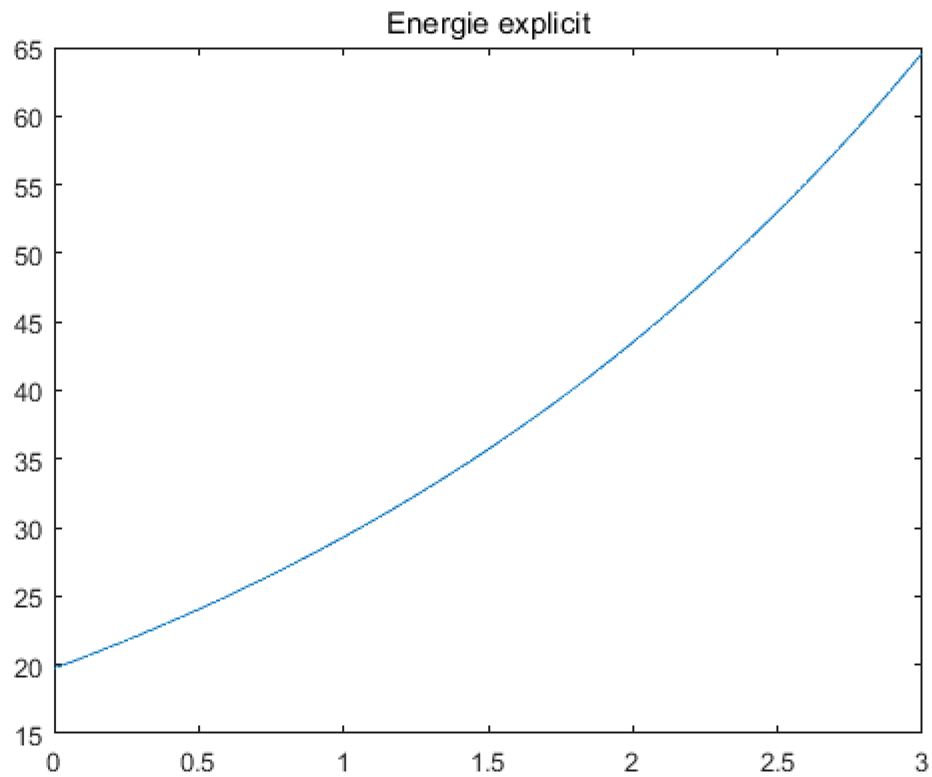
Q2

```
%2.1 remplacer D2q avec -W0^2*q on peut obtenir le resultat.
%2.2 et 2.3
dt = 0.01; %ou 0.1 0.01
A = [1,dt;-W0^2*dt,1];
Q1 = [];%Q1 est q
Q2 = [];%Q2 est dq
Q1(1)=1;
Q2(1)=0;
Q = [1;0];
n = 1;
E2 = [];
E2(1) = 1/2*(Q2(1)^2 + W0^2 * Q1(1)^2);
for t = 0:dt:T0
    n = n + 1;
    Q = A * Q;
    Q1(n) = Q(1,1);
    Q2(n) = Q(2,1);
    E2(n) = 1/2*(Q2(n)^2 + W0^2 * Q1(n)^2);
end
t2 = linspace(0,T0,n);
q = cos(W0*t2);
plot(t2,q,'*',t2,Q1);
legend('Analytique','Euler explicit')
```



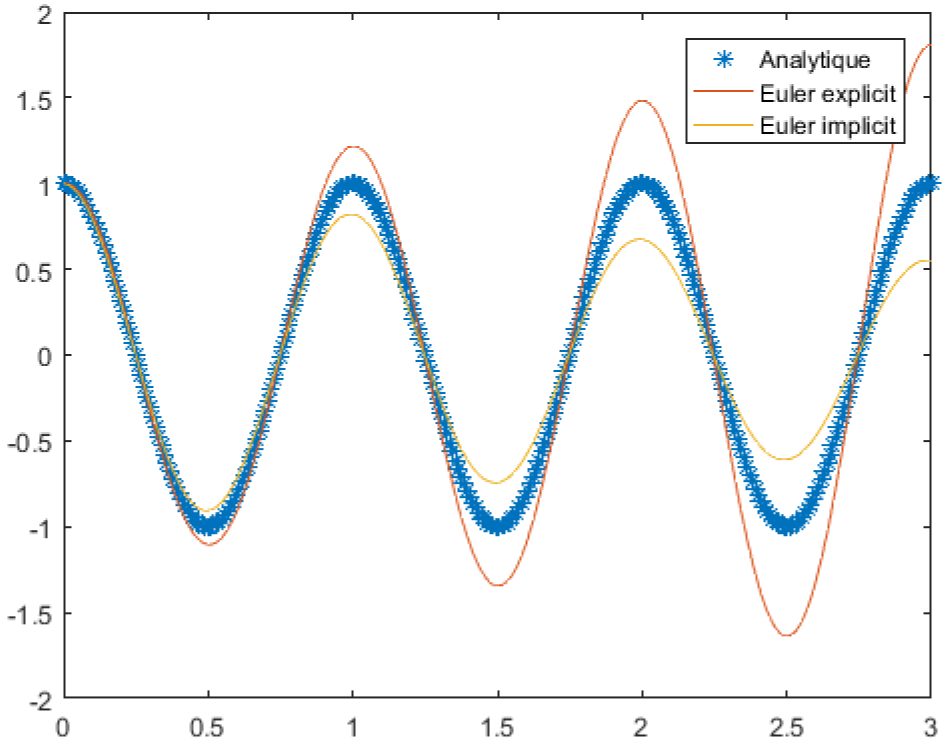
2.4

```
%E2 augment moins vite si dt est plus petit.
plot(t2,E2);
title('Energie explicit')

%2.5
[V,D] = eig(A);
%dt = 0.01
% D =
%    1.0000 + 0.0628i   0.0000 + 0.0000i
%    0.0000 + 0.0000i   1.0000 - 0.0628i
%dt = 0.001
% D =
%    1.0000 + 0.0063i   0.0000 + 0.0000i
%    0.0000 + 0.0000i   1.0000 - 0.0063i
%les valeurs absolues des valeurs propres sont plus grandes si dt est
 plus
%grandes et l'énergie diverge plus vite
```



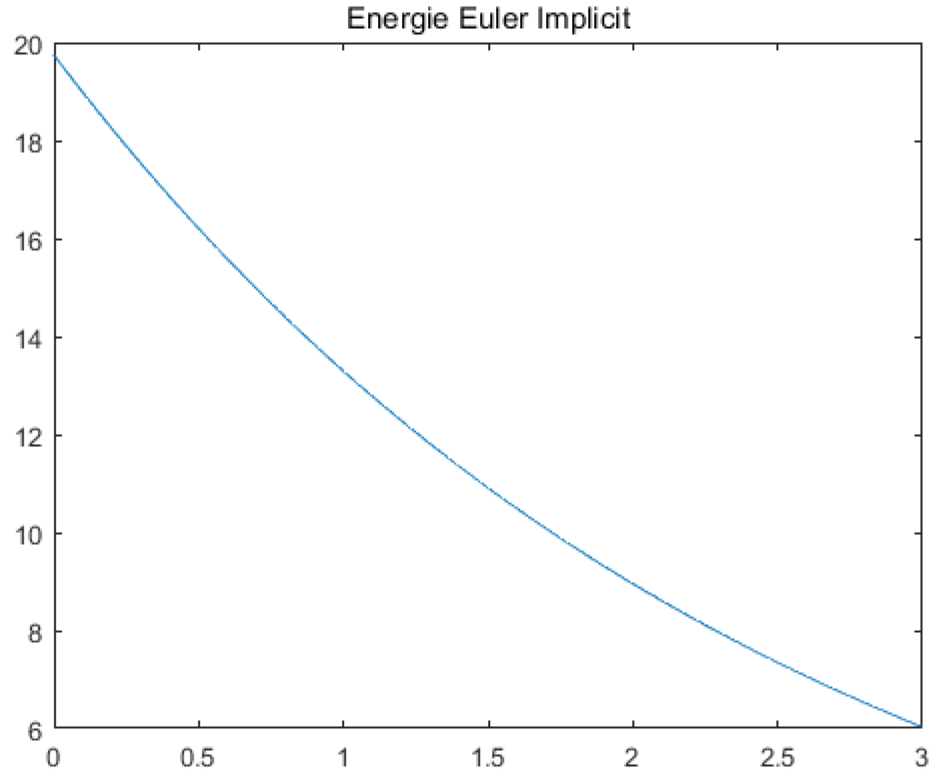
Q3

```
A2 = [1,dt;-W0^2*dt,1]/((dt*W0)^2+1);
Q1_im = [];
Q2_im = [];
Q1_im(1)=1;
Q2_im(1)=0;
Q_im = [1;0];
n_im = 1;
E2_im = [];
E2_im(1) = 1/2*(Q2_im(1)^2 + W0^2 * Q1_im(1)^2);
for t = 0:dt:T0
    n_im = n_im + 1;
    Q_im = A2 * Q_im;
    Q1_im(n_im) = Q_im(1,1);
    Q2_im(n_im) = Q_im(2,1);
    E2_im(n_im) = 1/2*(Q2_im(n_im)^2 + W0^2 * Q1_im(n_im)^2);
end
t2 = linspace(0,T0,n_im);
plot(t2,q,'*',t2,Q1,t2,Q1_im);
legend('Analytique','Euler explicit','Euler implicit')
%on peut voir que euler implicit converge ver 0 et euler explicit
 diverge.
```



3.4

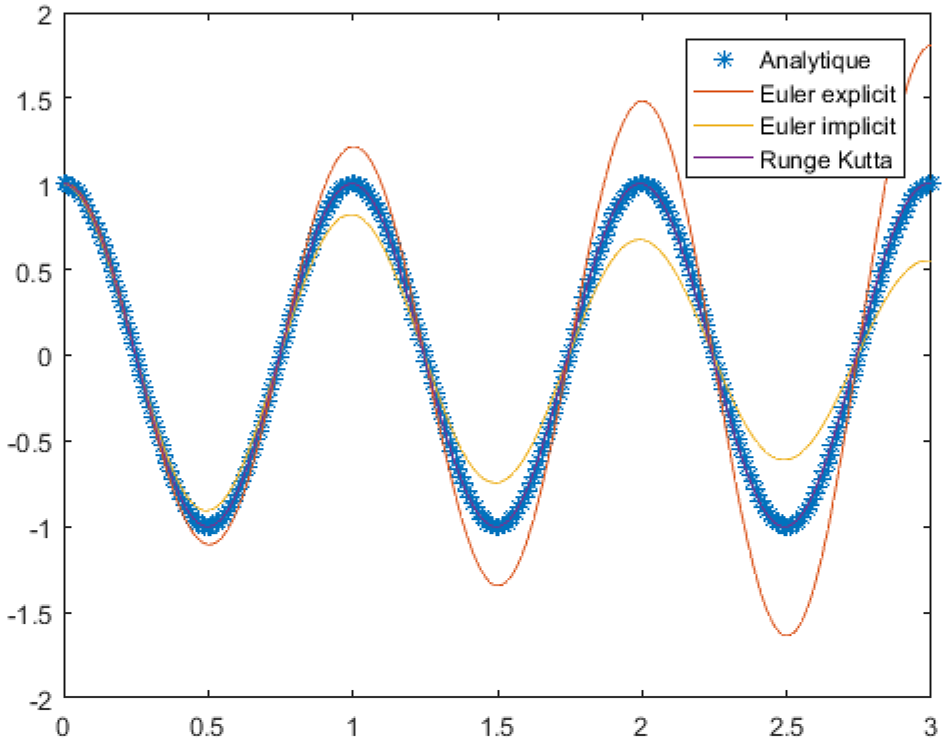
```
plot(t2,E2_im);
title('Energie Euler Implicit');
%l'énergie converge moins vite si dt est plus petie.
%3.5
[V2,D2] = eig(A2);
% dt = 0.01
% D2 =
%
%    0.9961 + 0.0626i   0.0000 + 0.0000i
%    0.0000 + 0.0000i   0.9961 - 0.0626i
%dt = 0.001
% D2 =
%
%    1.0000 + 0.0063i   0.0000 + 0.0000i
%    0.0000 + 0.0000i   1.0000 - 0.0063i
%on peut voir que les valeurs propres sont plus près de 1 si dt est
 plus
%petit.
```



Q4

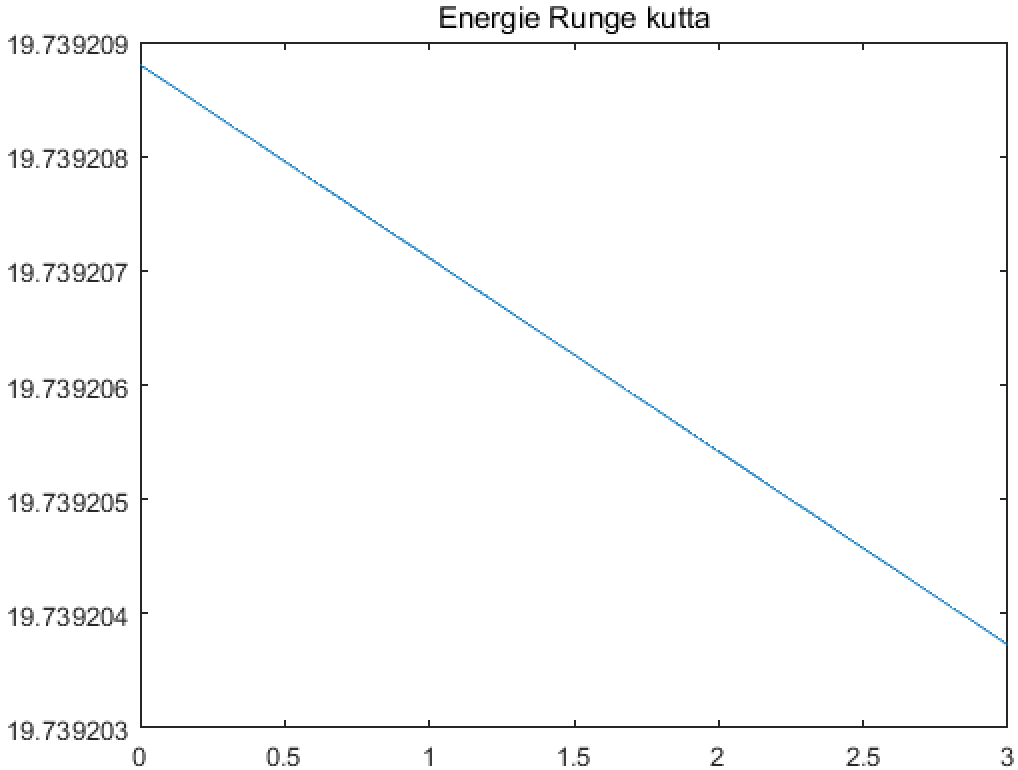
```
%4.1 x = q dx = dq
x = 1;
dx = 0;
X = [x;dx];
M = [0 , 1; -W0^2, 0];
dX = M * X;

%4.2
q_rg = [];
q_rg(1) = 1;
dq_rg = [];
dq_rg(1) = 0;
n = 1;
E_Rk = [];
E_Rk(1) = 1/2*(dq_rg(1)^2 + W0^2 * q_rg(1)^2);
for t = 0:dt:T0
    n = n + 1;
    k1 = M * X;
    k2 = M * (X + k1 * dt/2);
    k3 = M * (X + k2 * dt/2);
    k4 = M * (X + k3 * dt);
    K = (k1 + 2*k2 + 2*k3 + k4)/6;
    X = X + K *dt;
```

```
    q_rg(n) = X(1,1);
    dq_rg(n) = X(2,1);
    E_Rk(n) = 1/2*(dq_rg(n)^2 + W0^2 * q_rg(n)^2);
end
plot(t2,q,'*',t2,Q1,t2,Q1_im,t2,q_rg);
legend('Analytique','Euler explicit','Euler implicit','Runge Kutta')
```



4.4

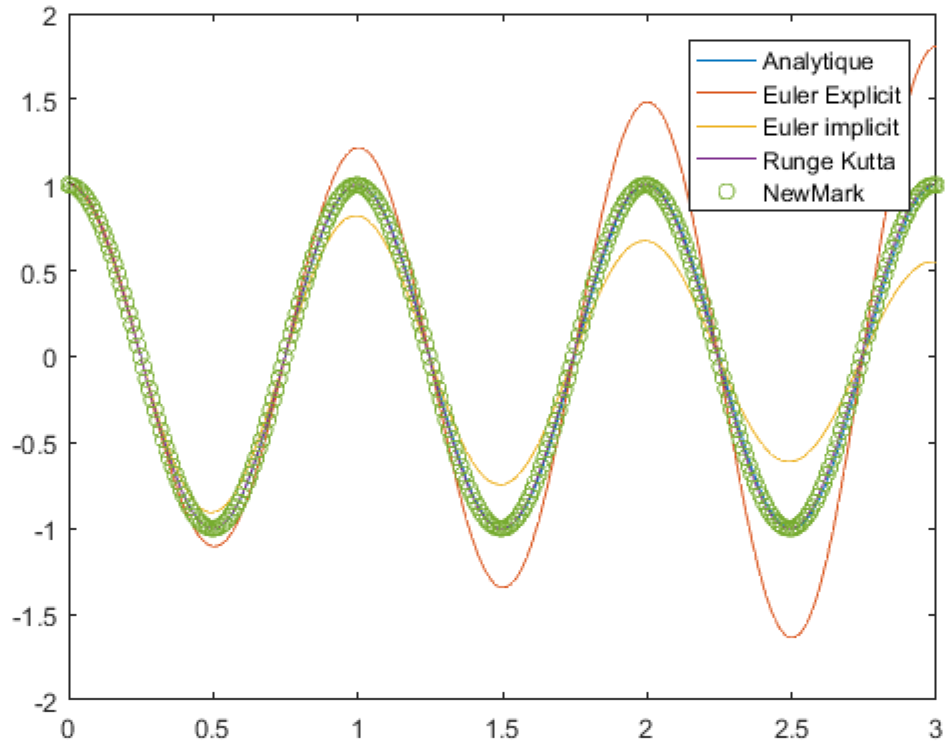
```
%l'énergie est plus stable que méthode Euler explicit et Euler
 implicit.
plot(t2,E_Rk);
title('Energie Runge kutta')
```



Q5

```
beta = 0.25;
gama = 0.5;
B = [1 + beta*dt^2*W0^2,0;gama*dt*W0^2,1];
C = [1-(0.5-beta)*dt^2*W0^2,dt;-(1-gama)*dt*W0^2,1];
T = inv(B);
A = T * C;

q_NM = [];
dq_NM = [];
Q_NM = [1;0];
q_NM(1) = 1;
dq_NM(1) = 0;
E_NM = [];
E_NM(1) = 1/2*(dq_NM(1)^2 + W0^2 * q_NM(1)^2);
n = 1;
for t = 0:dt:T0
    n = n + 1;
    Q_NM = A * Q_NM;
    q_NM(n) = Q_NM(1,1);
    dq_NM(n) = Q_NM(2,1);
    E_NM(n) = 1/2*(dq_NM(n)^2 + W0^2 * q_NM(n)^2);
end
t2 = linspace(0,T0,n);
```

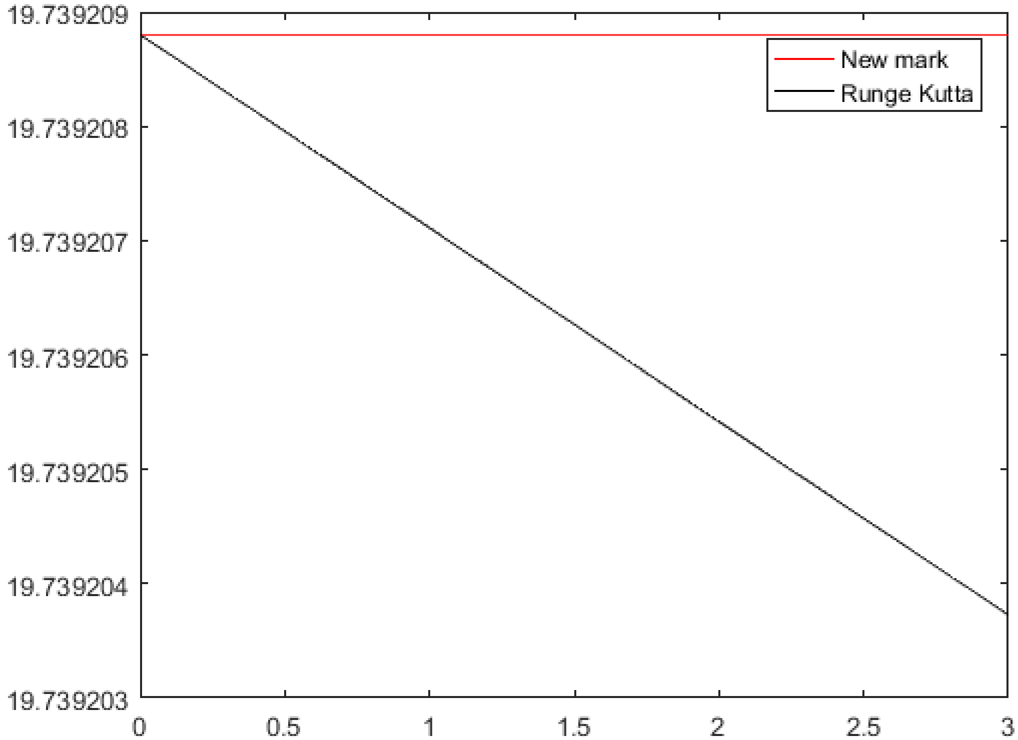
```
plot(t2,q,t2,Q1,t2,Q1_im,t2,q_rg,t2,q_NM,'o');
legend('Analytique','Euler Explicit','Euler implicit','Runge
 Kutta','NewMark');
```



5.1.3

```
plot(t2,E_NM,'r',t2,E_Rk,'black');
legend('New mark','Runge Kutta');
%NEWMARK l'energie est plus stable que les autres methode.

%5.1.4
[V3,D3] = eig(A);
%dt = 0.01
% D =
%
%    0.9980 + 0.0628i   0.0000 + 0.0000i
%    0.0000 + 0.0000i   0.9980 - 0.0628i
%dt = 0.001
% D =
%
%    1.0000 + 0.0063i   0.0000 + 0.0000i
%    0.0000 + 0.0000i   1.0000 - 0.0063i
%dt = 0.1
% D =
%
%    0.8203 + 0.5719i   0.0000 + 0.0000i
```

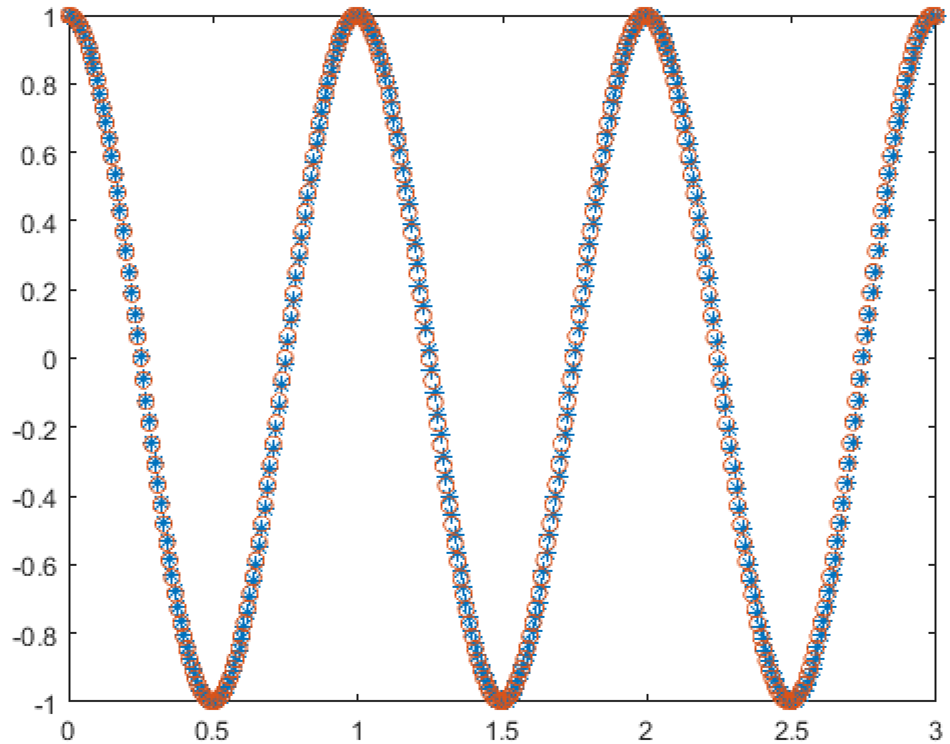
```
%    0.0000 + 0.0000i   0.8203 - 0.5719i
% les valeur propres ne changent pas beaucoup en fonction du dt.
```



5.2

```
beta = 0;
gama = 0.5;
B = [1 + beta*dt^2*W0^2,0;gama*dt*W0^2,1];
C = [1-(0.5-beta)*dt^2*W0^2,dt;-(1-gama)*dt*W0^2,1];
T = inv(B);
A = T * C;

q_NM = [];
dq_NM = [];
Q_NM = [1;0];
q_NM(1) = 1;
dq_NM(1) = 0;
E_NM = [];
E_NM(1) = 1/2*(dq_NM(1)^2 + W0^2 * q_NM(1)^2);
n = 1;
for t = 0:dt:T0
    n = n + 1;
    Q_NM = A * Q_NM;
    q_NM(n) = Q_NM(1,1);
    dq_NM(n) = Q_NM(2,1);
    E_NM(n) = 1/2*(dq_NM(n)^2 + W0^2 * q_NM(n)^2);
```

```
end
t2 = linspace(0,T0,n);
t = linspace(0,T0,n);
q = cos(W0*t);
plot(t,q,'*');
hold on;
plot(t2,q_NM,'o');
```



Published with MATLAB® R2016b