Exercice1

Oscillateur conservatif à un degré de liberté
Solution analytique

```
%% Analytique
close all;
clc;
clear;

% Initialisation & Configuration
dt = 0.01;
Tt = 3;
w0 = 2*pi;
te = 0:dt:Tt;
[mp,np] = size(te);
q0 = 1;
dq0 = 0;
U_a = [q0;dq0];
for ind = 2:np
    U_a(1,ind) = q0*cos(w0*(ind-1)*dt)+dq0*sin(w0*(ind-1)*dt)/w0;
    U_a(2,ind) = -q0*w0*sin(w0*(ind-1)*dt) + dq0*cos(w0*(ind-1)*dt);
end
energe_a = 0.5*(U_a(2,:).^2 + w0^2*(U_a(1,:).^2));

% Visualisation
figure();
plot(te,U_a(1,:),'--g','Linewidth',2);
figure();
plot(te,energe_a,'--g','Linewidth',2);
```

Figure 1

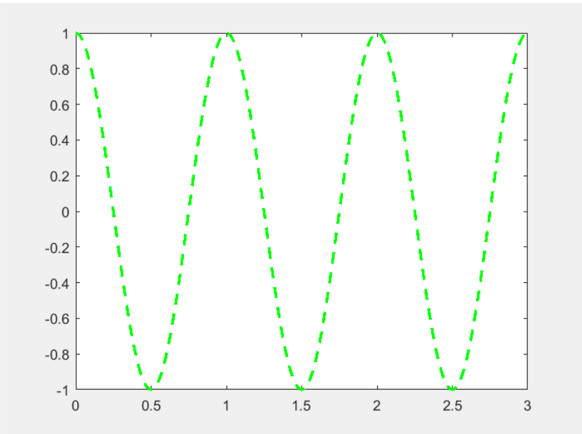
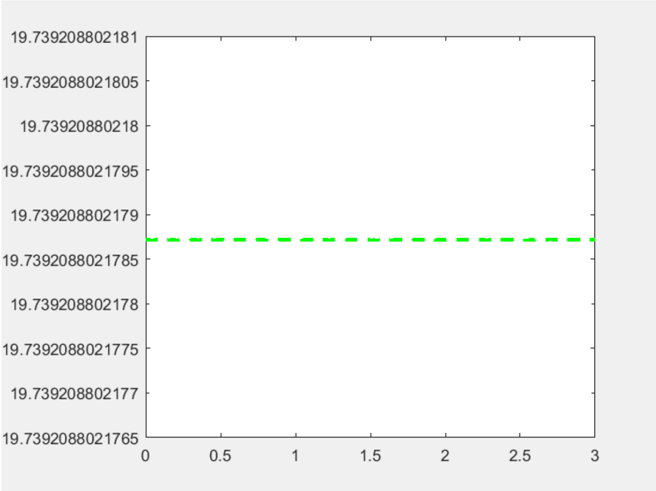


Figure 2



E* est une constante.

Exercice2

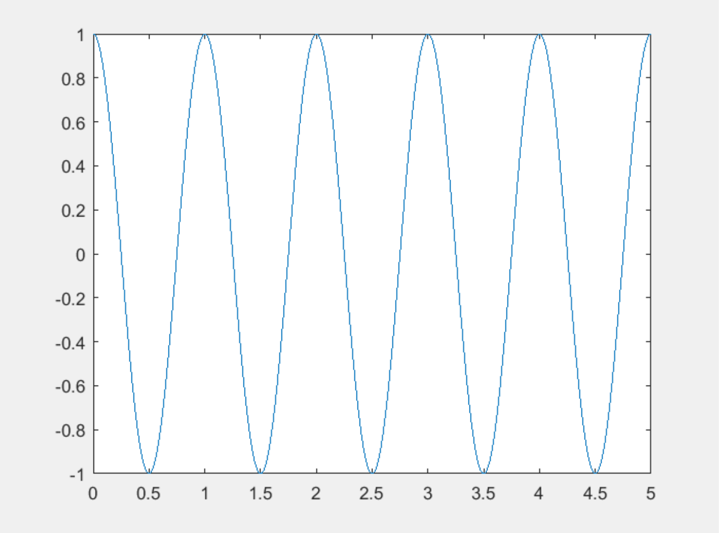
Oscillateur conservatif à un degré de liberté
Solution par Euler Explicite

```
% schema euler explicite
% a:direct b:matrice d'amplification
% a
% initialisation
dte = 0.01;
te = 0:dte:5;
w0 = 2*pi;
[mp,np] = size(te);
q = zeros(np,1);
dq = zeros(np,1);
ddq = zeros(np,1);
energe = zeros(np,1);
q(1) = 1;
dq(1) = 0;
tic;
ind = 1;
for t=te
    q(ind) = q0 * cos(w0*t) + dq0/w0 * sin(w0 * t);
    dq(ind) = -w0 * q0 * sin(w0 * t) + dq0 * cos(w0 * t);
    ddq(ind) = -w0^2 * q(ind);
    energe(ind) = 0.5 * (dq(ind)*dq(ind) + w0^2 * (q(ind)^2));
    ind = ind+1;
end
```

```
toc;

plot(te,q);
```

Figure




```
% euler explicite b
dt = 0.05;
te = (0:dt:10);
w0 = 2*pi;
[mp,np] = size(te);
np = np - 1;
q = zeros(np,1);
dq = zeros(np,1);
ddq = zeros(np,1);
U = [q';dq'];
U(:,1) = [1;0];
ind = 1;
A = [1 dt;-w0^2*dt 1];
for t = 1:np;
    U(:,ind+1) = A * U(:,ind);
    ind = ind + 1;
end
figure();
plot(te,U(1,:));
figure();
plot(U(1,:),U(2,:));
```

Figure 1

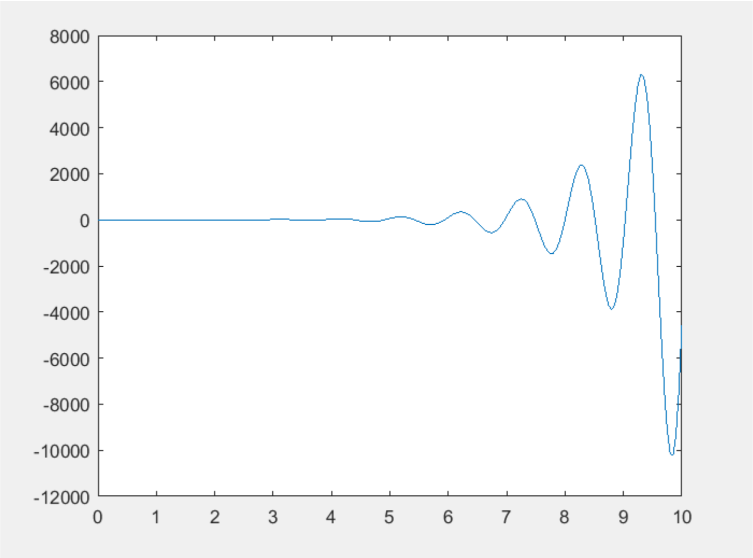
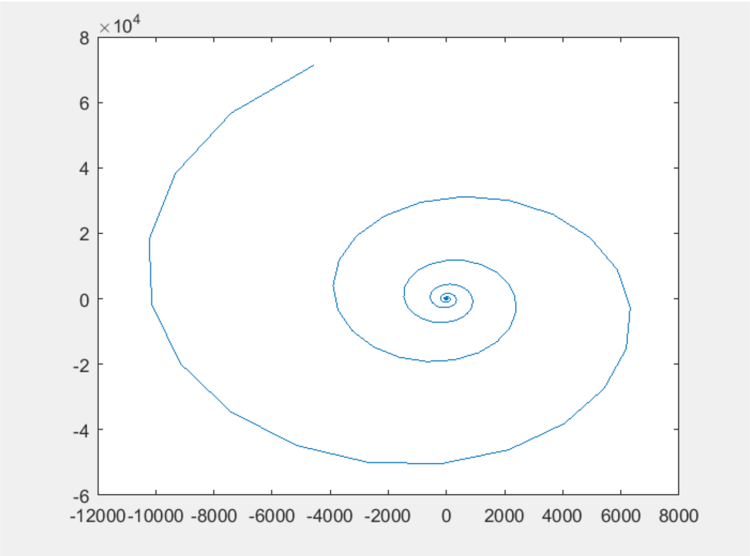


Figure 2



Si $\lambda < 1$, inconditionnellement stable

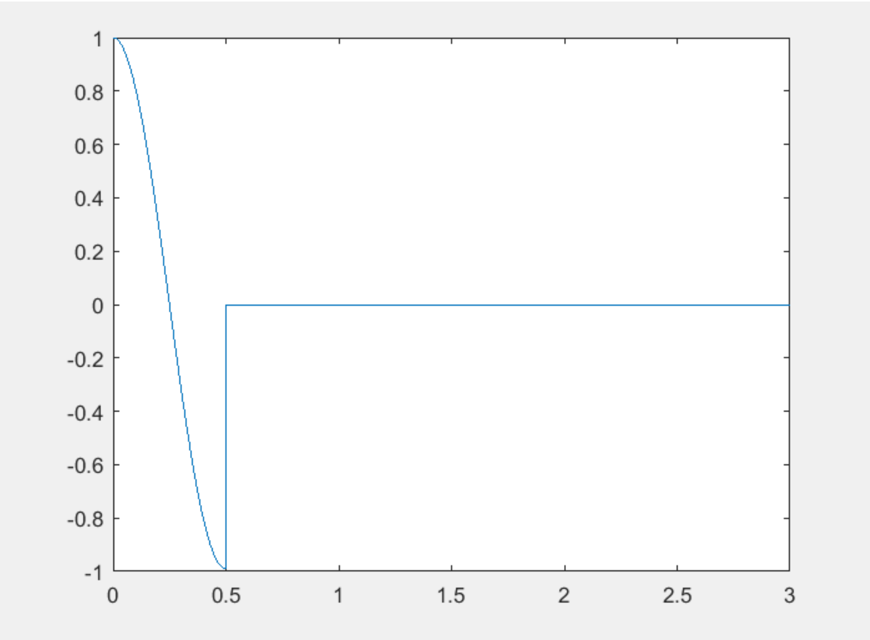
Exercice3

Oscillateur conservatif à un degré de liberté
Solution avec Euler Implicite, avec Runge Kutta.

```
% euler implicite q(j+1)=q(j)+dt*dq(j+1)
dte = 0.001;
te = 0:dte:3;
w0 = 2*pi;
[mp,np] = size(te);
q = zeros(np,1);
```

```
dq = zeros(np,1);
ddq = zeros(np,1);
energe = zeros(np,1);
q(1) = 1;
dq(1) = 0;
tic;
ind = 1;
for t=1:500
    q(ind+1) = (q(ind)+dte*dq(ind))/(1+dte^2*w0^2);
    ddq(ind+1) = -w0^2 * q(ind+1);
    dq(ind+1) = dq(ind) + dte*ddq(ind+1);
    energe(ind) = 0.5 * (dq(ind)*dq(ind) + w0^2 * (q(ind)^2));
    ind = ind+1;
end
toc;
figure();
plot(te,q');
```

Figure

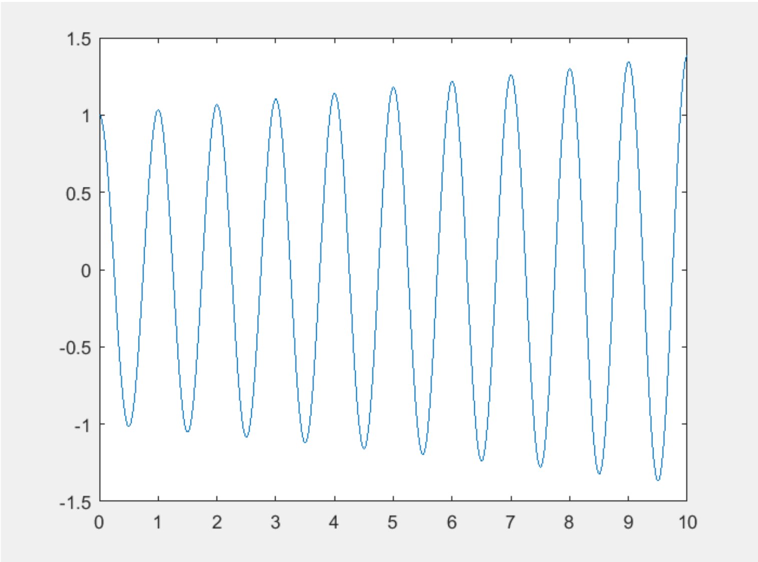



```
% methode Runge Kutta
close all;
clear;
clc;
w0 = 2 * pi;
dt = 0.01;
te = 0:dt:10;
```

```
[mp,np] = size(te);
q = zeros(np,1);
dq = zeros(np,1);
ddq = zeros(np,1);
energe = zeros(np,1);
q(1) = 1;
dq(1) = 0;
for n = 1:np-1
    k1 = dq(n);
    dk1 = -w0^2*q(n);
    k2 = dq(n) - 0.5*w0^2*dt*q(n);
    dk2 = -w0^2*q(n) - 0.5*w0^2*dt*dq(n);
    k3 = dq(n) * (1-w0^2*dt^2*0.25)-0.5*dt*w0^2*q(n);
    dk3 = -w0^2 * q(n)* (1-w0^2*dt^2*0.25)-0.5*dt*w0^2*dq(n);
    k4 = dq(n) * (1-dt^2*w0^2*0.25) + (dt^3*w0^4/8 - dt*w0^2/2)*q(n);
    dk4 = -w0^2 * q(n) * (1-dt^2*w0^2*0.25) + (dt^3*w0^4/8 - dt*w0^2/2)*dq(n);
    K = (k1+2*k2+2*k3+k4)/6;
    dK = (dk1+2*dk2+2*dk3+dk4)/6;
    q(n+1) = q(n) + K*dt;
    dq(n+1) = dq(n) + dt*dK;
end

figure();
plot(te,q');
% equation
t0 = 0;
tfinal = 20;
x0 = [0 0.25]';
```

Figure



Si $\lambda < 1$, inconditionnellement stable

Si $\lambda(\Delta t) < 1$, depend de Δt, alors pas de temps critique.