Table of Contents

Etude d''un oscillateur lineaire amorti a un degre de liberte

```
T0 = 1;
epsilon = 0.02;
w0 = 2*pi/T0;
t = 2*epsilon/w0
omiga = w0*sqrt(1-epsilon^2);
x =@(y) exp(-epsilon*w0*y)*(0.01*cos(omiga*y)+(epsilon*w0*0.01+0)/
omiga*sin(omiga*y));
```

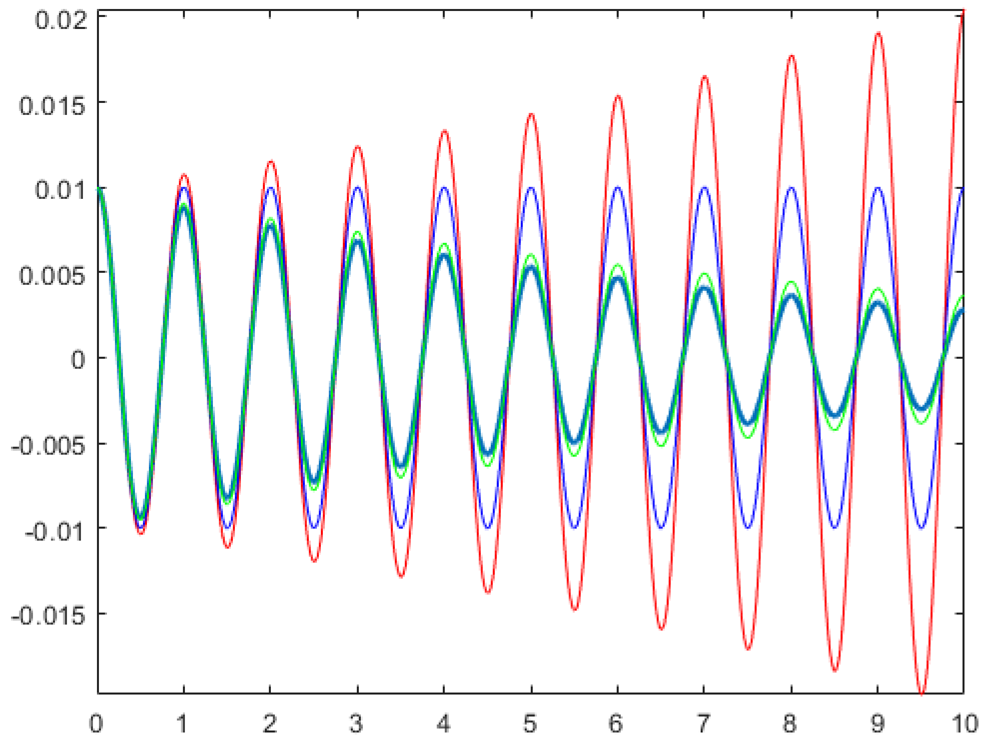
```
t =

    0.0064
```

question1.1

1.1a EULER explicite

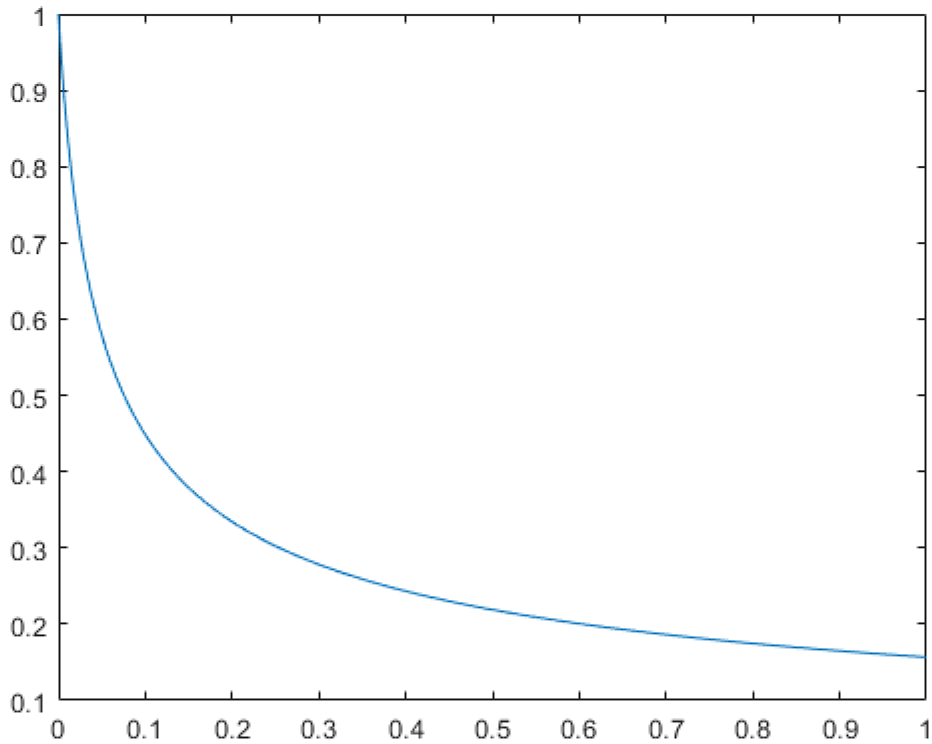
```
deltat = 0.01
x1 = [0.01];
dx1 = [0];
for i=1:(10/deltat)
    x1(i+1) = x1(i)+deltat*dx1(i);
    dx1(i+1) = dx1(i)+deltat*(-w0^2*x1(i)-2*epsilon*w0*dx1(i));
end
figure(1);
fplot(x,[0,10*T0],'linewidth',2);
hold on;

plot([0:deltat:10],x1,'r')

%%1.1b EULER explicite
deltat = t
x2 = [0.01];
dx2 = [0];
for i=1:(10/deltat)
    x2(i+1) = x2(i)+deltat*dx2(i);
    dx2(i+1) = dx2(i)+deltat*(-w0^2*x2(i)-2*epsilon*w0*dx2(i));
```

```
end

plot([0:deltat:10],x2,'b')

%%1.1c EULER explicite
deltat = t*0.2
x3 = [0.01];
dx3 = [0];
for i=1:(10/deltat)
    x3(i+1) = x3(i)+deltat*dx3(i);
    dx3(i+1) = dx3(i)+deltat*(-w0^2*x3(i)-2*epsilon*w0*dx3(i));
end

plot([0:deltat:10],x3,'g')
hold off;

%%1.1d
%plus petit le rapport deltat*w0/2*epsilon, plus precis le resultet
 est
%0.2 est bon


deltat =

    0.0100

##: ##############################################
#####

deltat =

    0.0064


deltat =

    0.0013
```



question 1.2

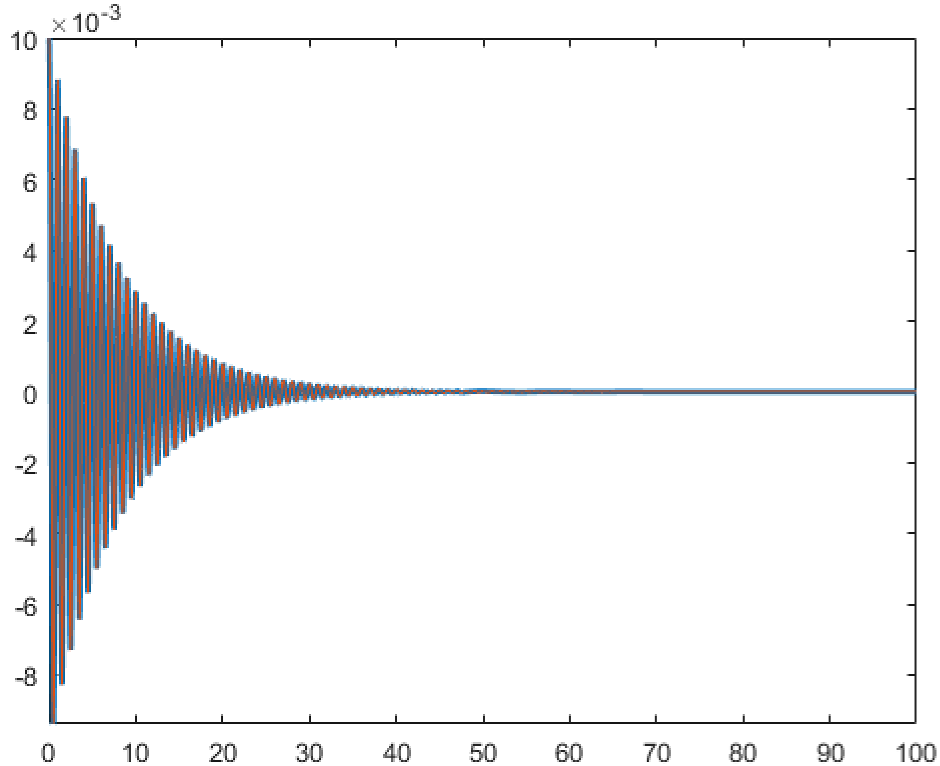
```
% on peut trouver que (1-deltat*[0,1;-w0^2,-2epsilon*w0])X(j+1)=X(j)
% Donc on doit trouver les valeurs propres de [0,1;-w0^2,-2epsilon*w0]
syms dt
Ma=inv([1,-dt;w0^2,1+2*epsilon*w0*dt]);
eigmax=[];
for dt=0:0.001:1
    eigmax=[eigmax,max(abs(eig(eval(Ma))))];
end
dt=0:0.001:1;
figure(5);
plot(dt,eigmax)
% [x,y]=eig(Ma)
% eq=abs(y(1,1))-1
% eq2=abs(y(2,2))-1
% critiquet=vpasolve(eq2,dt)
%pas de temps critique est 0
```

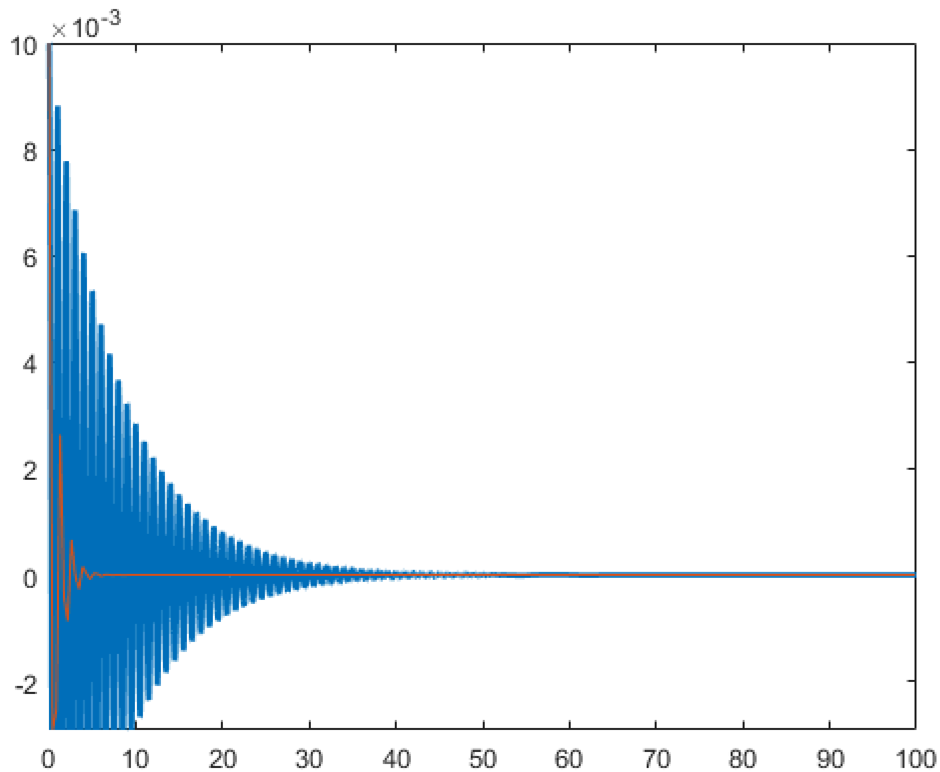


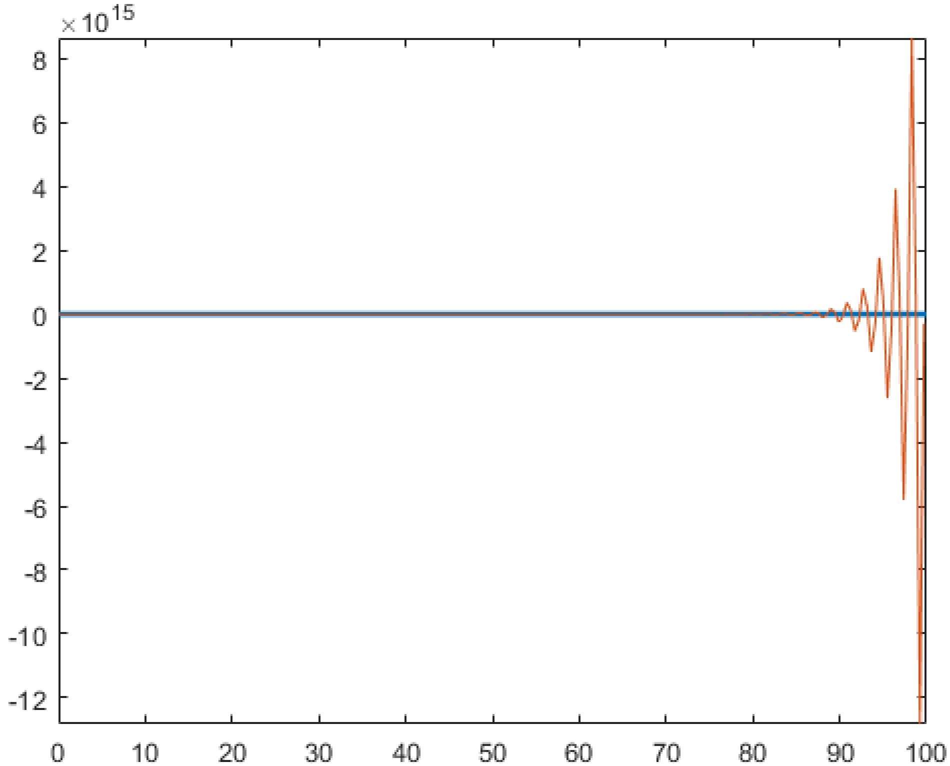
question 1.3

```
% 1.3.a
% h=0.04
% selon l'equation 1, [Dq;D2q]=[0,1;-w0^2,-2*epsilon*w0][q;Dq]
% deltat = 0.04*2*sqrt(2)/w0;
x4 = [0.01];
dx4 = [0];
A = [0,1;-w0^2,-2*epsilon*w0];
for i=1:100/deltat
    k1 = A*[x4(i);dx4(i)];
    k2 = A*([x4(i);dx4(i)]+deltat/2.*k1);
    k3 = A*([x4(i);dx4(i)]+deltat/2.*k2);
    k4 = A*([x4(i);dx4(i)]+deltat.*k3);
    K = (k1+2*k2+2*k3+k4)./6;
    x4 = [x4,x4(i)+deltat*K(1)];
    dx4 = [dx4,dx4(i)+deltat*K(2)];
end
figure(2);
fplot(x,[0,100*T0],'linewidth',2)
hold on;
plot([0:deltat:100],x4)
hold off;

% % h=0.96
```

```
deltat = 0.96*2*sqrt(2)/w0;
x5 = [0.01];
dx5 = [0];
A = [0,1;-w0^2,-2*epsilon*w0];
for i=1:100/deltat
    k1 = A*[x5(i);dx5(i)];
    k2 = A*([x5(i);dx5(i)]+deltat/2.*k1);
    k3 = A*([x5(i);dx5(i)]+deltat/2.*k2);
    k4 = A*([x5(i);dx5(i)]+deltat.*k3);
    K = (k1+2*k2+2*k3+k4)./6;
    x5 = [x5,x5(i)+deltat*K(1)];
    dx5 = [dx5,dx5(i)+deltat*K(2)];
end
figure(3);
fplot(x,[0,100*T0],'linewidth',2)
hold on;
plot([0:deltat:100],x5)
hold off;

% h=1.04
deltat = 1.04*2*sqrt(2)/w0;
x6 = [0.01];
dx6 = [0];
A = [0,1;-w0^2,-2*epsilon*w0];
for i=1:100/deltat
    k1 = A*[x6(i);dx6(i)];
    k2 = A*([x6(i);dx6(i)]+deltat/2.*k1);
    k3 = A*([x6(i);dx6(i)]+deltat/2.*k2);
    k4 = A*([x6(i);dx6(i)]+deltat.*k3);
    K = (k1+2*k2+2*k3+k4)./6;
    x6 = [x6,x6(i)+deltat*K(1)];
    dx6 = [dx6,dx6(i)+deltat*K(2)];
end
figure(4);
fplot(x,[0,100*T0],'linewidth',2)
hold on;
plot([0:deltat:100],x6)
hold off;

% quand h=0.04, la figure est stable et precis
% quand h=0.96, la figure n'est pas precis mais elle est stable
% quand h=0.04, la figure n'est pas stable ou precis

##: ##############################################
#####
##: ##############################################
#####
##: ##############################################
#####
```


×10^-3
10
8
6
4
2
0
-2
-4
-6
-8
0
10
20
30
40
50
60
70
80
90
100


×10^-3
10
8
6
4
2
0
-2
0
10
20
30
40
50
60
70
80
90
100



Published with MATLAB® R2019b