Mécanique Numérique DM4

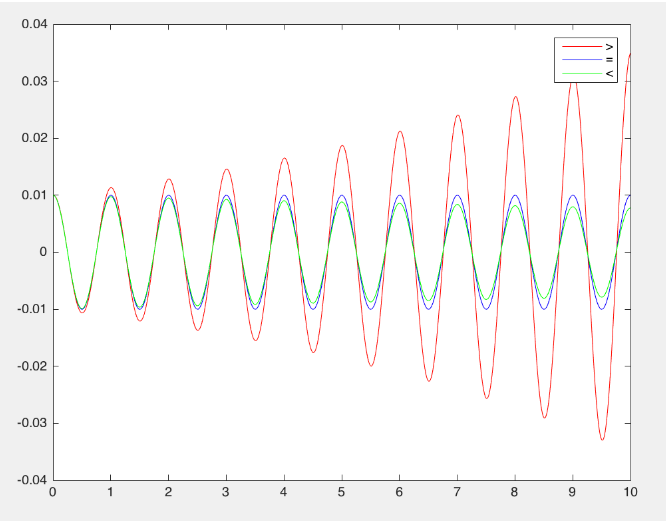
I. Etude d'un oscillateur linéaire amorti à un degré de liberté

1.1

On peut choisir 3 dt différents（dt=4*epsilon/w0，2*epsilon/w0，1.6*epsilon/w0）pour trouver la différence. Et on les met dans un même graph.

```
T0=10;q0=0.01;dq0=0;w0=2*pi;,epsilon=0.02;omega=w0*(1-epsilon^2)^(1/2)
for dt1=[4*epsilon/w0,2*epsilon/w0,1.6*epsilon/w0 ]
    t1=(0:dt1:T0)';
    np1=size(t1,1);
    q1=zeros(np1,1);
    dq1=zeros(np1,1);
    ddq1=zeros(np1,1);
    q1(1)=q0;
    dq1(1)=dq0;
    ddq1(1)=-2*epsilon*w0*dq1(1)-w0^2*q1(1);
    for inc=2:np1
        q1(inc)=q1(inc-1)+dt1*dq1(inc-1);
        dq1(inc)=dq1(inc-1)+dt1*ddq1(inc-1);
        ddq1(inc)=-2*epsilon*w0*dq1(inc)-w0^2*q1(inc)
    end
    if dt1==4*epsilon/w0
        plot(t1,q1,'r')
        hold on
    elseif dt1==2*epsilon/w0
        plot(t1,q1,'b')
        hold on
    else
        plot(t1,q1,'g')
        hold on
    end
end;
legend('>','=','<')
```

Alors on obtient:

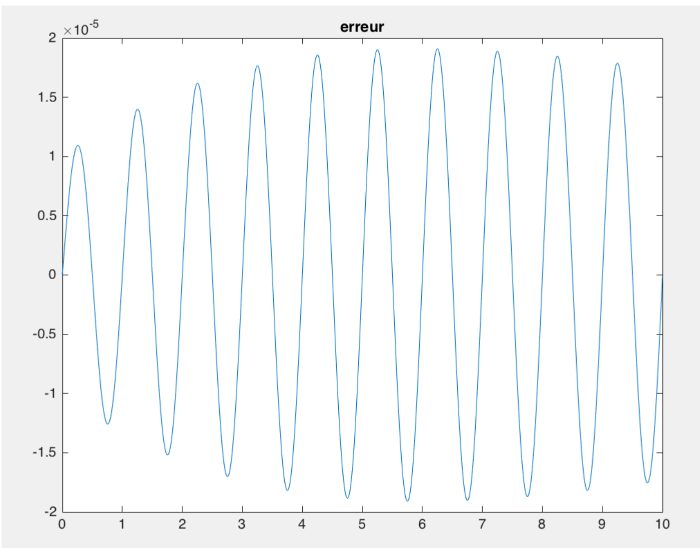


on trouve que pour 2epsilon/w0 > ou < dt, l'image sont divergents; pour 2epsilon/w0=dt, l'image converge.

Pour étudier la précision, on peut étudier l'erreur.

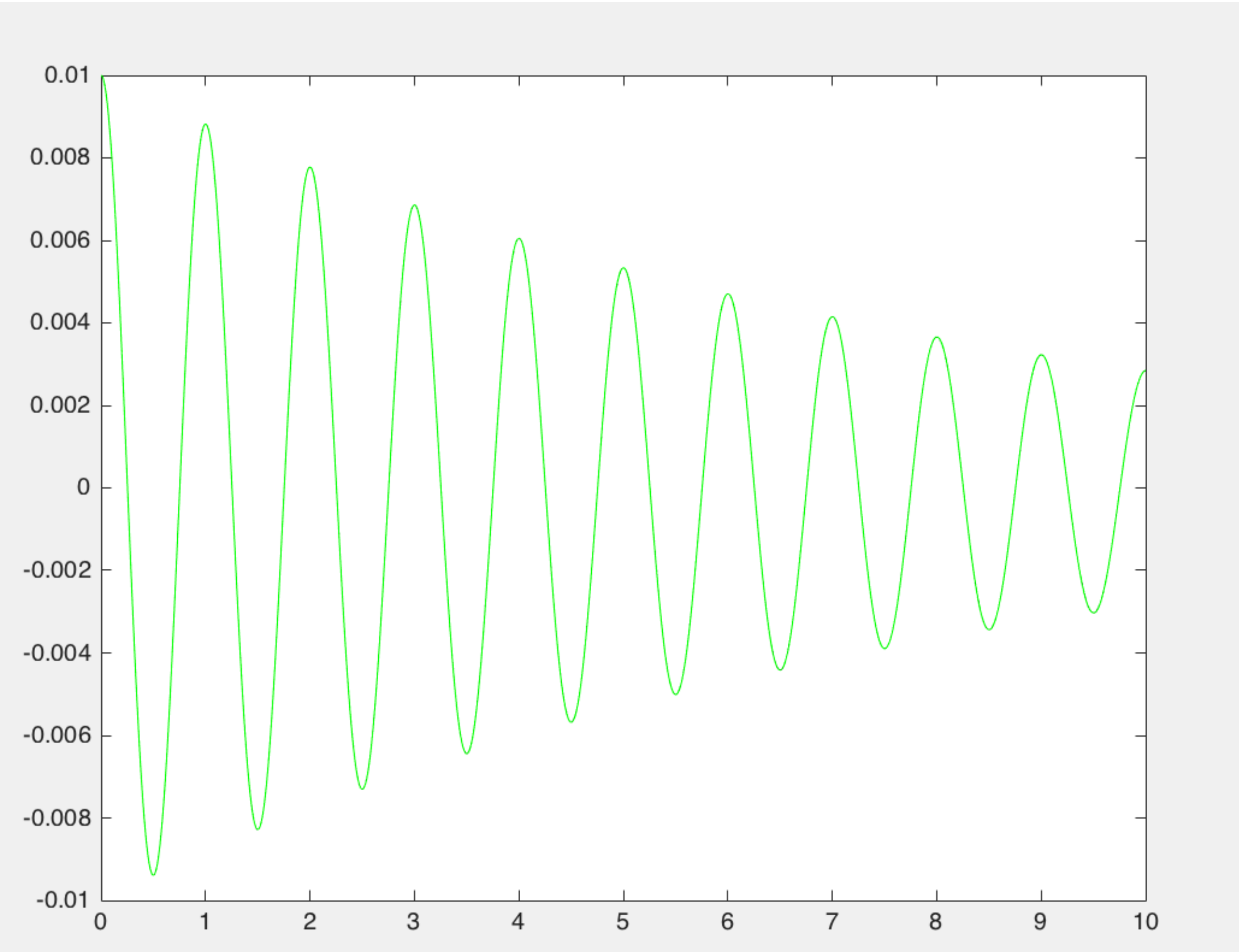
```
w0=2*pi;
ep=0.02;
omega=w0*(1-ep^2)^(0.5);
dt=0.05*2*ep/w0;
T0=1;
q0=0.01;
dq0=0;

A=[1,0;dt*(w0^2),1+2*dt*ep*w0];
B=[1,dt;0,1];
C=inv(A)*B
U=[q0;dq0];
Y=[];
for inc=1:(10*T0/dt)
    Y(inc)=U(1);
    U=C*U;
end
L=linspace(0,10*T0,10*T0/dt);
Y0=(exp(-ep*w0*L)).*(q0*cos(omega*L)+(ep*w0*q0+dq0)*(sin(omega*L))/omega);
plot(L,(Y-Y0));
title('erreur');
```

Avec l'image



1.2

```
clear all
close all
w0=2*pi;
ep=0.02;
omega=w0*(1-ep^2)^(0.5);
T0=1;
q0=0.01;
dq0=0;
for dt2=[0.001*ep/w0]
    A=[1,-dt2;dt2*(w0^2),1+2*ep*w0*dt2];
    A=inv(A);
    U=[q0;dq0];
    Y=[];
    for inc=1:(10*T0/dt2)
        Y(inc)=U(1);
        U=A*U;
    end
    L=linspace(0,10*T0,10*T0/dt2);
    plot(L,Y,'k')
        hold on
        Y0=(exp(-ep*w0*L)).*(q0*cos(omega*L)+(ep*w0*q0+dq0)*(sin(omega*L))/omega);
        plot(L,Y0,'g')
    end
```



Je ne pense pas qu'il existe un pas de temps critique parce que les solutions convergent toujours.

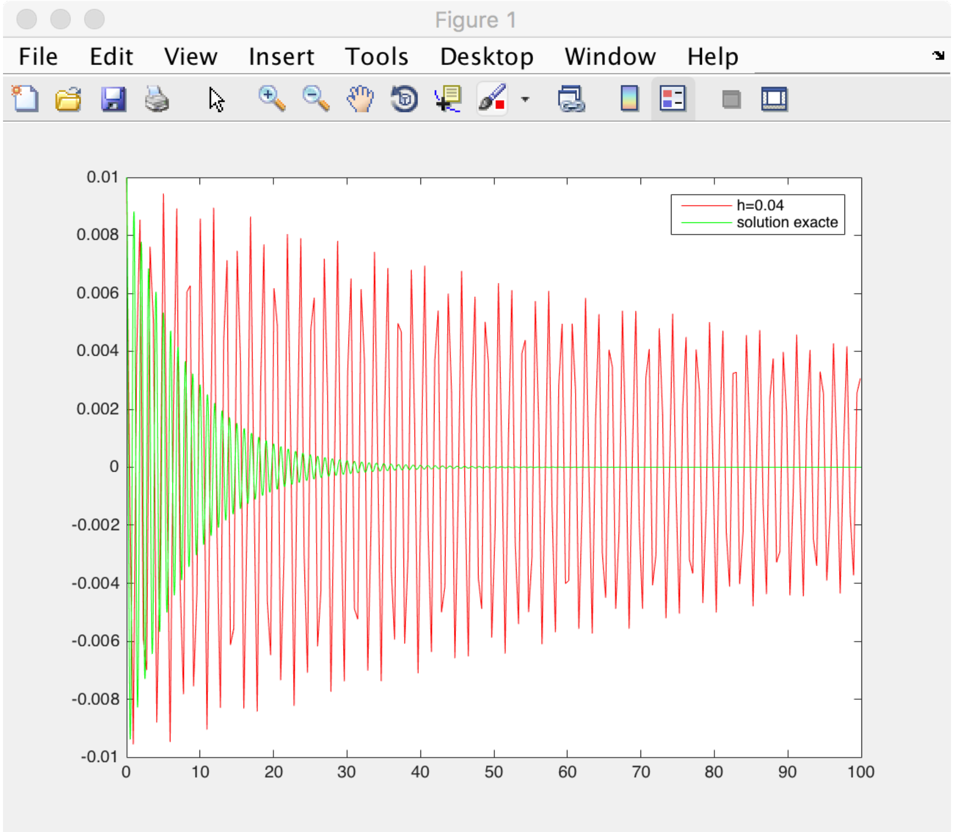
1.3

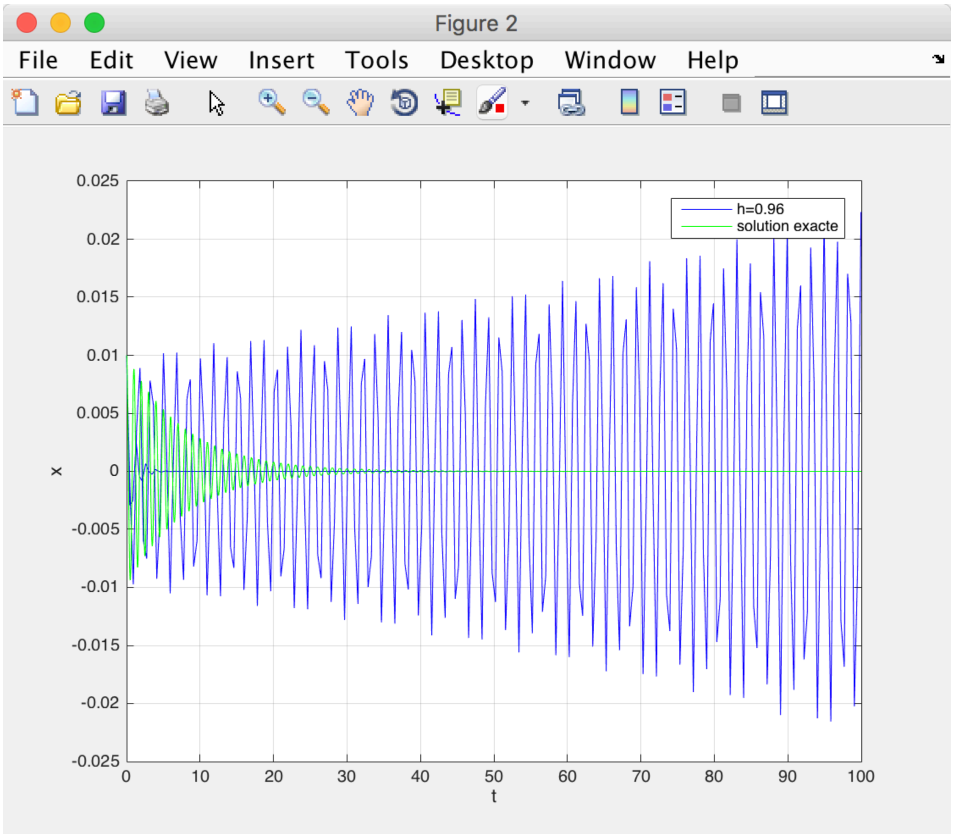
En utilisant Runge Kutta, on peut déduire le script.

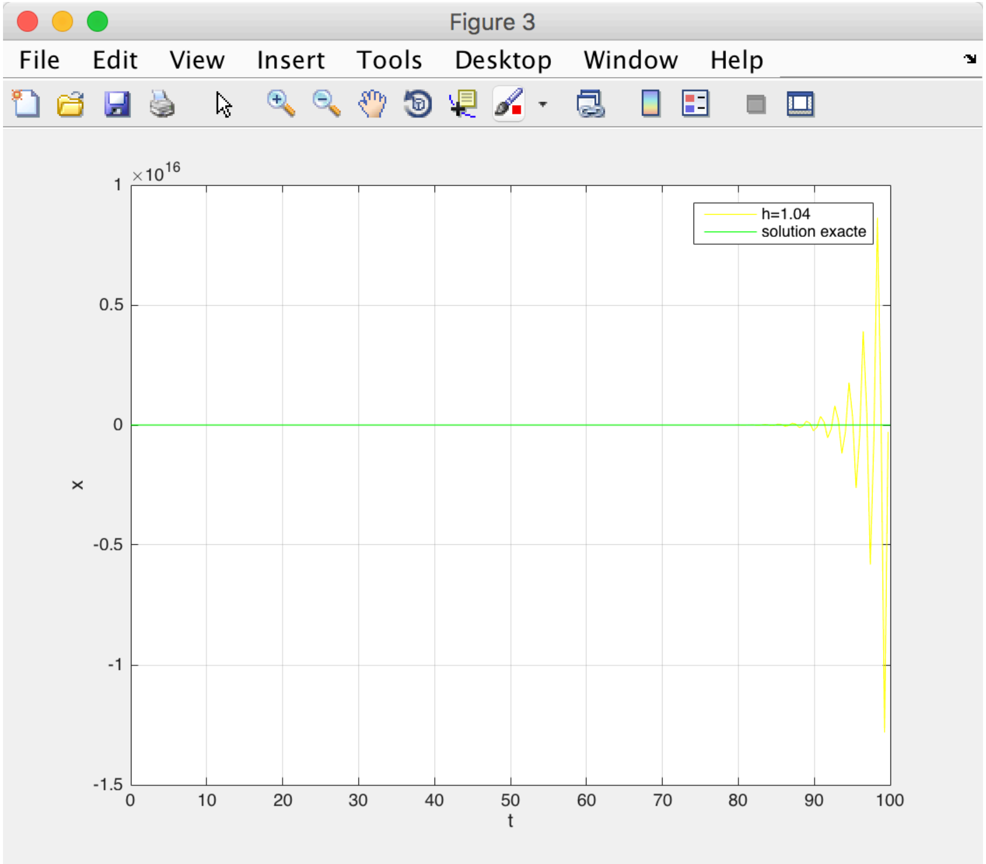
```
T0=1;q0=0.01;dq0=0;w0=2*pi;e=0.02;



%Runge Kutta avec h different
for h=[0.04,0.96,1.04]
   dt3=h*2*sqrt(2)/w0;
   t1=(0:dt3:100*T0)';
   np1=size(t1,1);
   q1b=zeros(np1,1);
   dq1b=zeros(np1,1);
   q1b(1)=q0;
   dq1b(1)=dq0;
   Q=[q0;dq0];
   for i=2:np1
      tc=t1(i-1);
      xc=Q;
      k1=cal_f(xc,tc,e,2*pi);
      xc=Q+k1*dt3/2;
      k2=cal_f(xc,tc+dt3/2,e,2*pi);
      xc=Q+k2*dt3/2;
      k3=cal_f(xc,tc+dt3/2,e,2*pi);
      xc=Q+k3*dt3;
      k4=cal_f(xc,tc+dt3,e,2*pi);
      dq=(k1+2*k2+2*k3+k4)/6;
      Q=Q+dq*dt3;
      q1b(i)=Q(1);
      dq1b(i)=Q(2);
   end
   %solution exacte
```

```
    syms t
    q=dsolve('D2x+2*0.02*2*pi*Dx+(2*pi)^2*x=0','x(0)=0.01 ','Dx(0)=0');
    t=0:0.01:100;
    q=subs(q,t);
    if h==0.04
        figure(1)
        plot(t1,q1b,'r')
        hold on
        plot(t,q,'g')
        legend('h=0.04','solution exacte')
    elseif h==0.96
        figure(2)
        plot(t1,q1b,'b')
        hold on
        plot(t,q,'g')
        legend('h=0.96','solution exacte')
    elseif h==1.04
        figure(3)
        plot(t1,q1b,'y')
        hold on
        plot(t,q,'g')
        legend('h=1.04','solution exacte')
    end
    hold on
end

grid on;
xlabel('t');
ylabel('x');
```

et les schémas sont ci-dessous :







On sait que quand h<hc, la solution doit être convergente, si h>hc, la solution doit être divergente. Donc hc doit être entre 1.04 et 0.96