Mécanique Numérique DM4

Cécilia Li Sha

Sujet 1. Étude d’un oscillateur linéaire amorti à un degré de liberté

1.1 (a)-1.1(c)Résolution avec Euler explicite

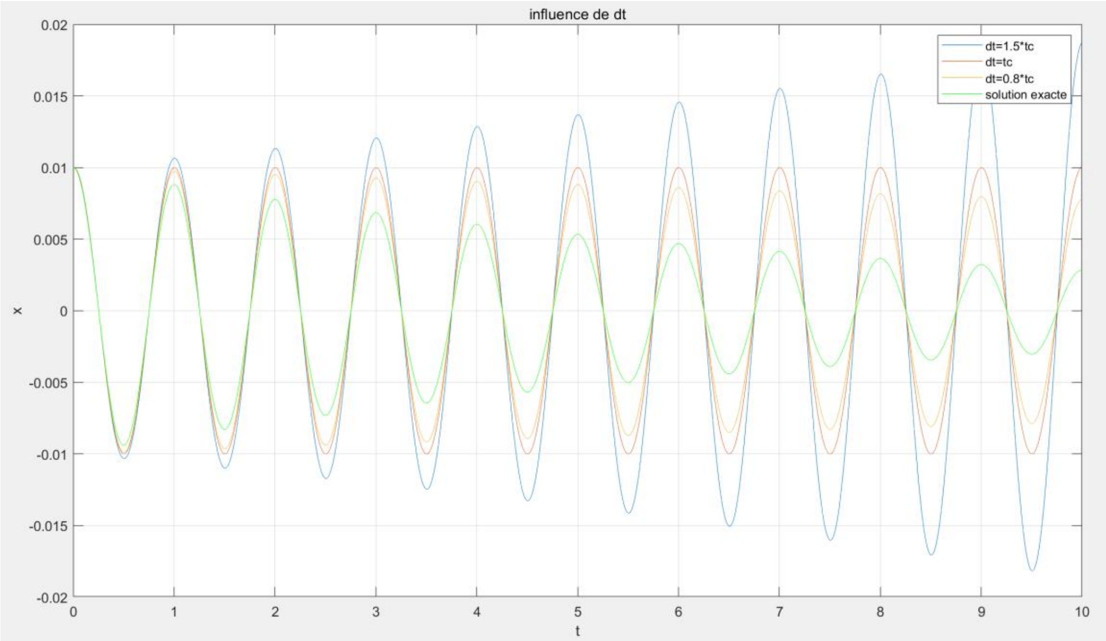
1.1 (a)-1.1(c)

Script Matlab

```
T0=1;q0=0.01;dq0=0;w0=2*pi;e=0.02;
tc=2*e/w0;%tc=2*e/w0
np1=size(t1,1);
%euler explicite avec dt different
for dt1=[1.5*tc,tc,0.8*tc]
    t1=(0:dt1:10*T0)';
    q1b=zeros(np1,1);
    dq1b=zeros(np1,1);
    ddq1b=zeros(np1,1);
    q1b(1)=q0;
    dq1b(1)=dq0;
    ddq1b(1)=-2*e*w0*dq1b(1)-w0^2*q1b(1);
    for i=2:np1
        q1b(i)=q1b(i-1)+dt1*dq1b(i-1);
        dq1b(i)=dq1b(i-1)+dt1*ddq1b(i-1);
        ddq1b(i)=-2*e*w0*dq1b(i)-w0^2*q1b(i);
    end
    plot(t1,q1b)
    hold on
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=0:0.01:10;
q=subs(q,t);
plot(t,q,'g')

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

```
title('influence de dt');
legend('dt=1.5*tc','dt=tc','dt=0.8*tc','solution exacte')
```

On obtient le dessin dessous avec 4 lignes dedans:

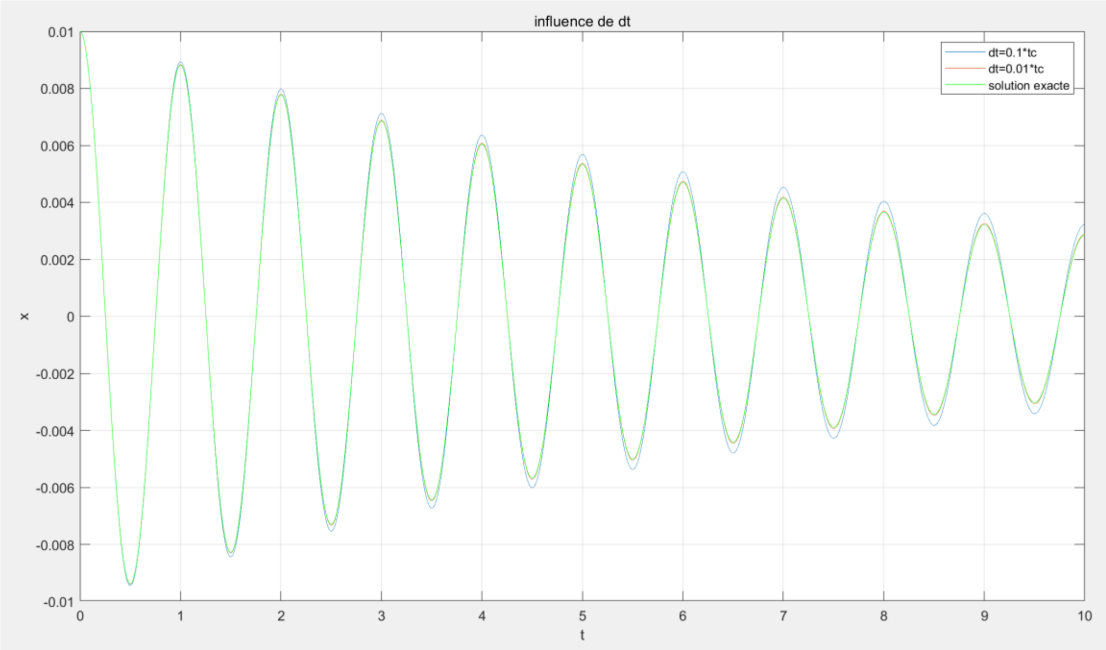


On remarque tc=2*e/w0, la solution diverge quand dt est plus grand que tc(dt>tc), converge quand dt est plus petit que tc(dt<tc), et oscille quand dt est égale à tc(dt=tc).

1.1 (d)

--Dans notre sujet, le solution exacte est convergente, donc il faut que dt soit plus petit que tc(dt<tc), qui permet d'étudier la précision.

--on essaie 2 valeur de dt, dt=0.1*tc et dt=0.01*tc, et on obtient : si dt=0.1*tc, on a une précision suffisante.



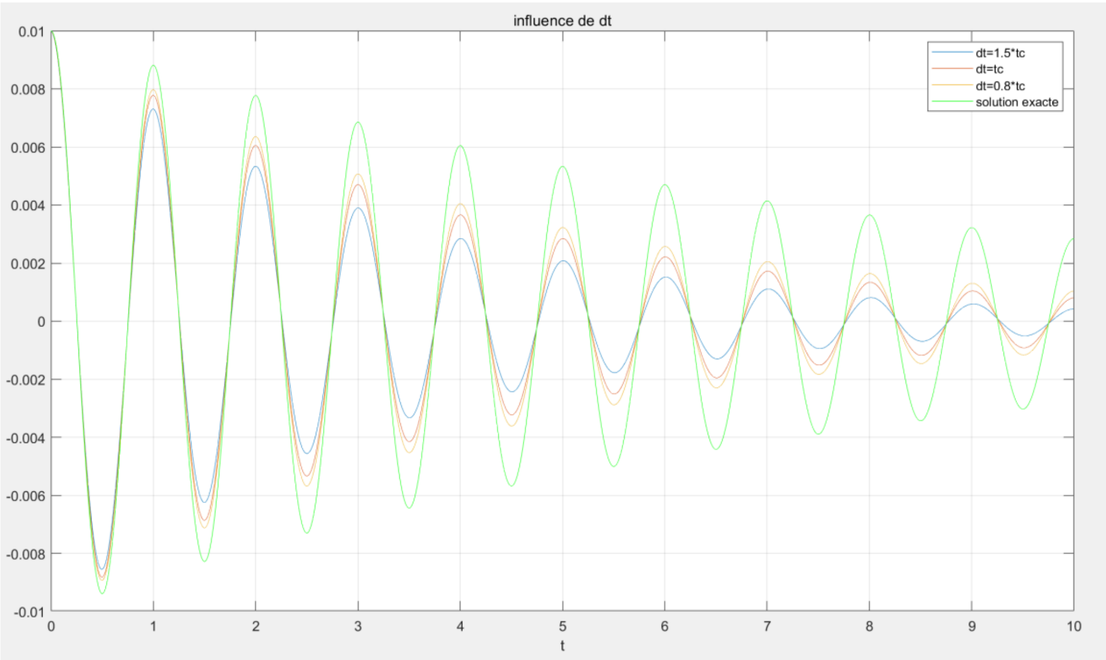
1.2 Résolution avec Euler implicite

Script Matlab

```
T0=1;q0=0.01;dq0=0;w0=2*pi;e=0.02;
omega=w0*(1-e1^2)^(0.5);
tc=2*e/w0;%tc=2*e/w0

%euler explicite avec dt different
for dt2=[1.5*tc,tc,0.8*tc]
    t1=(0:dt2:10*T0)';
    np1=size(t1,1);
    q1b=zeros(np1,1);
    A=[1,-dt2;dt2*(w0^2),1+2*e1*w0*dt2];
    A=inv(A);
    Q=[q0;dq0];

    for i=1:np1
        q1b(i)=Q(1);
        Q=A*Q;
    end
    plot(t1,q1b)
    hold on
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=0:0.01:10;
q=subs(q,t);
plot(t,q,'g')

grid on;
xlabel('t');
ylabel('x');
title('influence de dt');
legend('dt=1.5*tc','dt=tc','dt=0.8*tc','solution exacte')
```

On obtient le dessin dessous avec 4 lignes dedans:



On remarque tc=2*e/w0, la solution obtenue par Euler implicite est toujours convergente n'Importe quelle dt, donc pour Euler Implicite, il n'existe pas le pas de temps critique.

1.3 Résolution avec Runge Kutta

1.3(a)

Il faut d'abord définir la fonction cal_f :

```
function[dUc]=cal_f(Uc,tc,e1,w0)
dUc=zeros(2,1);
dUc(1)=Uc(2);
dUc(2)=-2*e1*w0*Uc(2)-(w0^2)*Uc(1);
end
```

Et on calcule en utilisant Runge Kutta avec h différent :

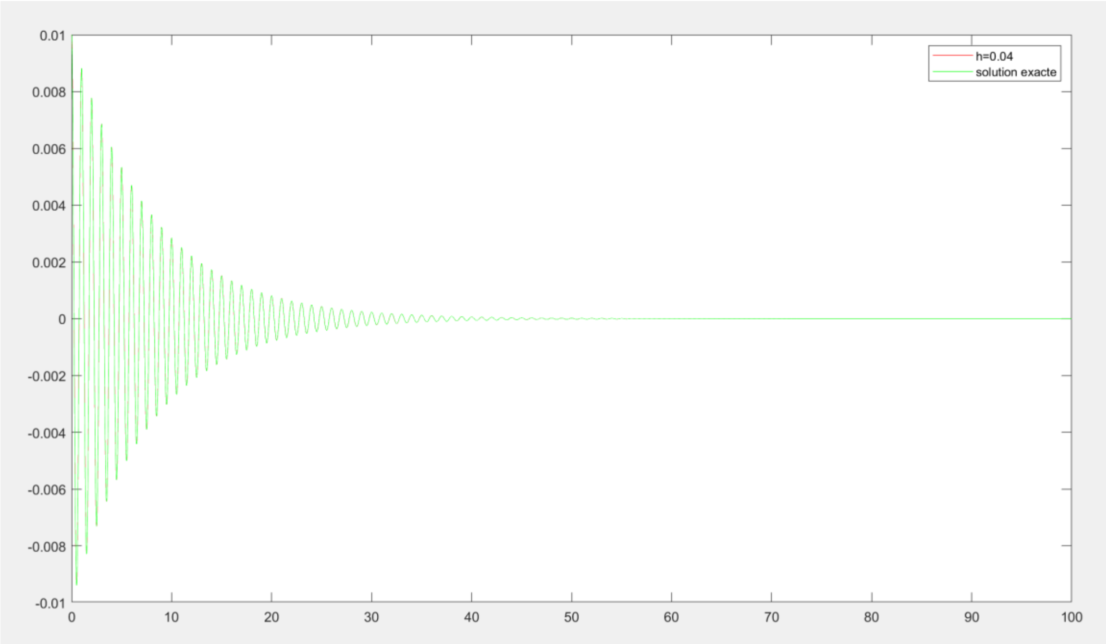
```
T0=1;q0=0.01;dq0=0;w0=2*pi;e=0.02;

%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)
```

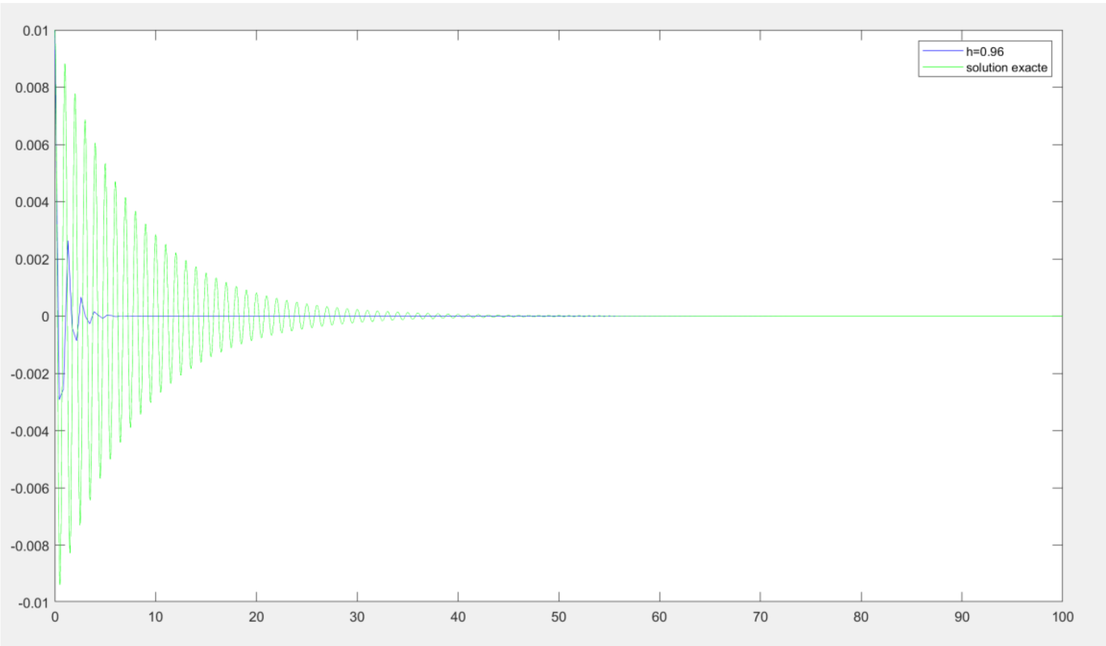
```
        plot(t1,q1b,'r')
        hold on
        plot(t,q,'g')
        legend('h=0.04','solution exacte')
    elseif h==0.96
        figure(2)
        plot(t1,q1b,'b')
        hold on
        plot(t,q,'g')
        legend('h=0.96','solution exacte')
    elseif h==1.04
        figure(3)
        plot(t1,q1b,'y')
        hold on
        plot(t,q,'g')
        legend('h=1.04','solution exacte')
    end
    hold on
end

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

```
ylabel('x');
```

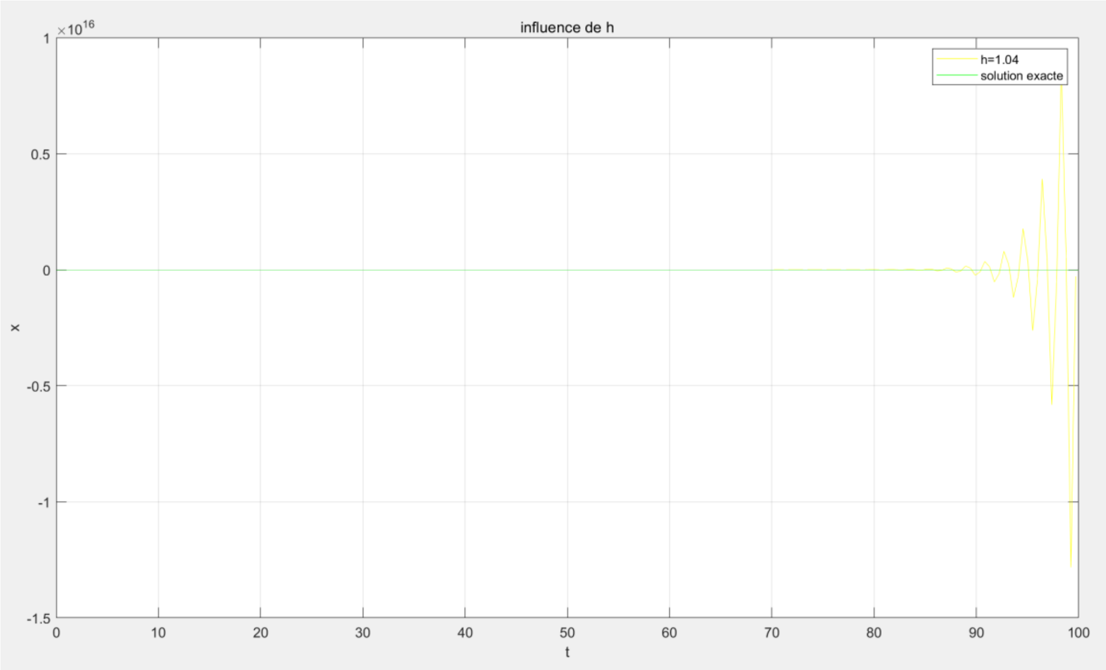
Pour h=0.04, résultat est convergent, précis et stable, de plus, proche de la solution exacte.



Pour h=0.96, résultat est convergent et stable, mais il y a une grande decalage entre le résultat et la solution exacte.



Pour h=1.04, le résultat est divergente, en revanche la solution exacte est convergente, donc n'est pas stable, ni précis, cette façon ne marche pas si h=1.04.



Donc il exsite un hc, si h est plus petit que hc, le résultat est convergente.

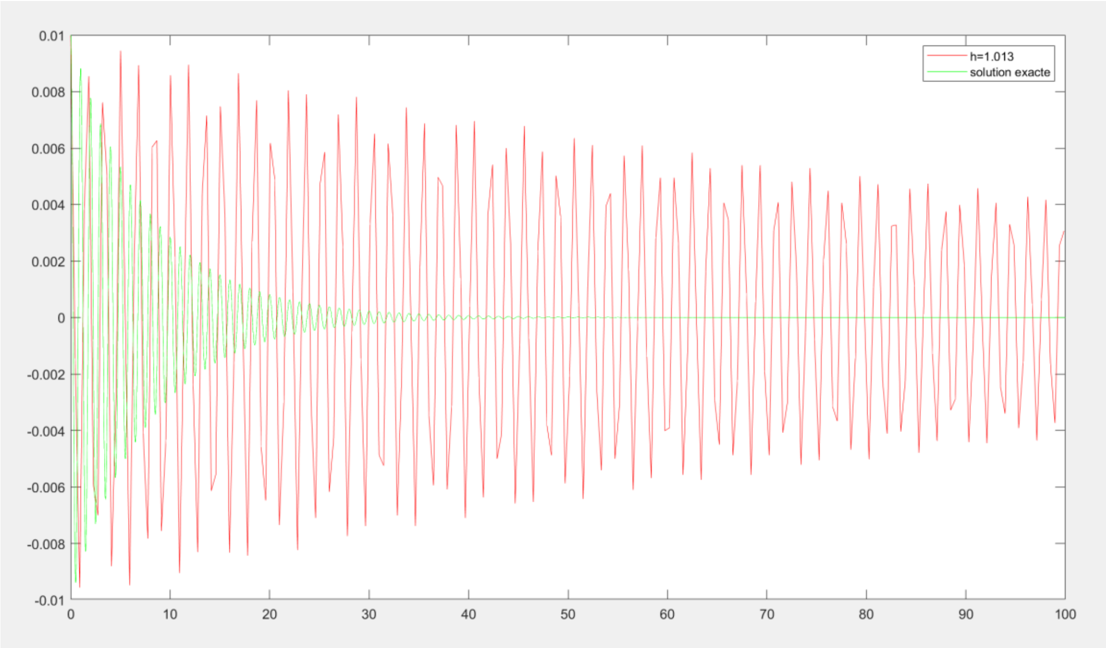
1.3.b En fait on essaie plusieurs fois, et finalement on obtient que hmax=1.014, hmin=1.013, hc est entre hmax et hmin.

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

%Runge Kutta avec h different
for h=[1.013 1.014]
    dt3=h*2*sqrt(2)/w0;
    t1=(0:dt3:100*T0)';
    np1=size(t1,1);
    q1b=zeros(np1,1);
```

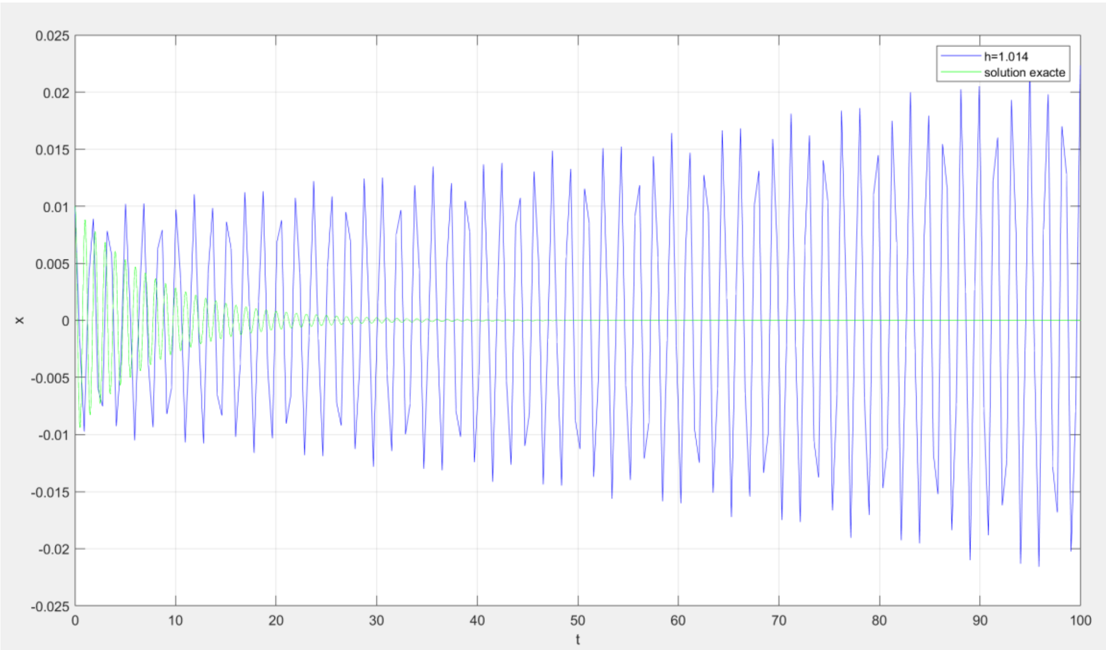
```
dq1b=zeros(np1,1);
q1b(1)=q0;
dq1b(1)=dq0;
Q=[q0;dq0];
for i=2:np1
    tc=t1(i-1);
    xc=Q;
    k1=cal_f(xc,tc,e,2*pi);
    xc=Q+k1*dt3/2;
    k2=cal_f(xc,tc+dt3/2,e,2*pi);
    xc=Q+k2*dt3/2;
    k3=cal_f(xc,tc+dt3/2,e,2*pi);
    xc=Q+k3*dt3;
    k4=cal_f(xc,tc+dt3,e,2*pi);
    dq=(k1+2*k2+2*k3+k4)/6;
    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==1.013
        figure(1)
        plot(t1,q1b,'r')
        hold on
        plot(t,q,'g')
        legend('h=1.013','solution exacte')
    elseif h==1.014
        figure(2)
        plot(t1,q1b,'b')
        hold on
        plot(t,q,'g')
        legend('h=1.014','solution exacte')
    end
    hold on
end
grid on;
xlabel('t');
ylabel('x');
```

h=1.013, convergent



h=1.014,divergent



Donc on obtient que hmax=1.014， hmin=1.013, hc est entre hmax et hmin.

Sujet2 Etude d'un double pendule avec l'hypothèse des petits mouvements

1.1

```
syms m a g F0 w beta gamma dt n

I = [1, 0; 0, 1];
A1 = [2, 1; 1, 1];
A2 = [2, 0; 0, 1];
A3 = [a; a / sqrt(2)];
% D'apres le sujet, m * a * a * A1 * d2q +
m * g * a * A2 * q = F0 * sin(w * t) * A3
% q = [theta1; theta2] et dq = [dtheta1;
dtheta2]
A4 = - inv(A1) * g / a * A2;
A5 = inv(A1) * F0 / m / a / a * A3;
% dq = A4 * q + A5 * sin(w * t)
A6 = I - dt * dt * beta * A4;
A7 = I + dt * dt * (0.5 - beta) * A4;
A8 = I * dt;
A9 = dt * dt * (0.5 - beta) * A5 * sin(w *
n * dt) + dt * dt * beta * A5 * sin(w * (n
+ 1) * dt);
```

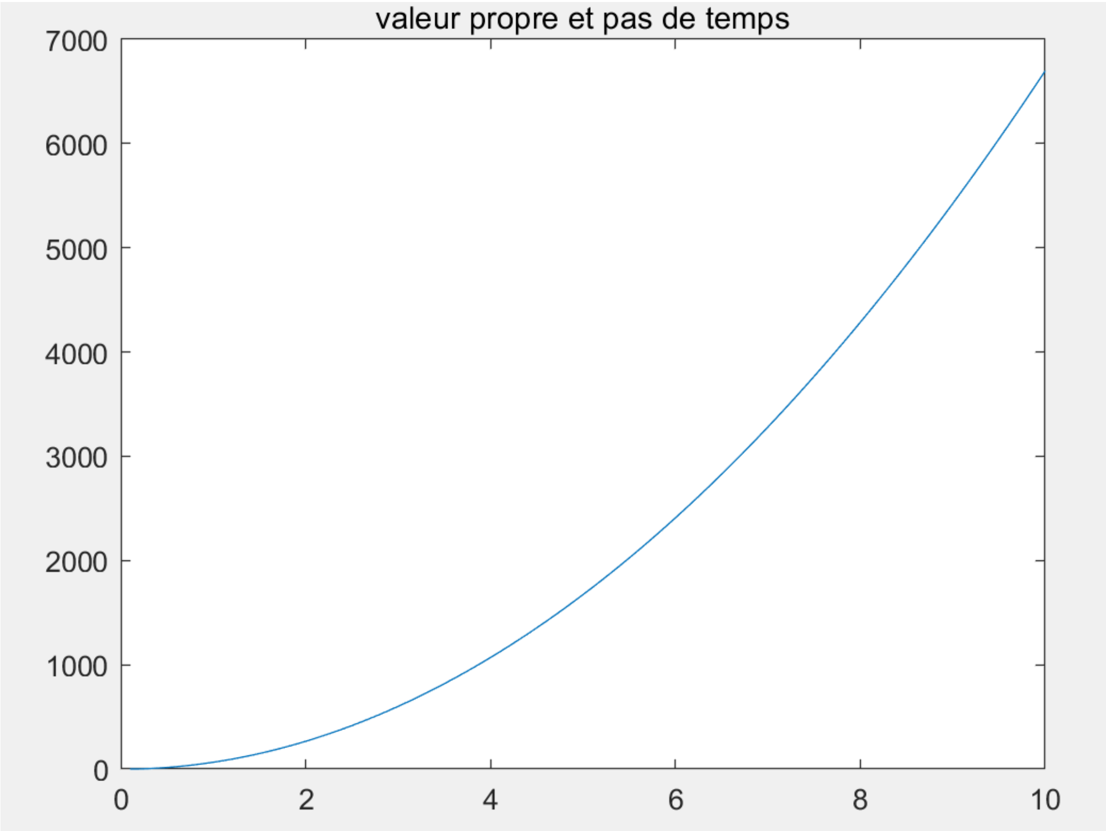
```
% A6 * qn1 = A7 * qn + A8 * dqn + A9
A10 = - dt * gamma * A4;
A11 = I;
A12 = dt * (1 - gamma) * A4;
A13 = I;
A14 = dt * (1 - gamma) * A5 * sin(w * n *
dt) + dt * gamma * A5 * sin(w * (n + 1) *
dt);
% A10 * qn1 + A11 * dqn1 = A12 * qn + A13 *
dqn + A14
A15 = [A6, 0 * I; A10, A11];
A16 = [A7, A8; A12, A13];
A17 = [A9; A14];
% Comme U = [q; dq], alors on peut trouver
A15 * dUn = A16 * Un + A17
A = inv(A15) * A16;
B = inv(A15) * A17;
```

Et on obtient A et B très compliqués.

1.2

```
m = 2;a = 0.5;g = 9.81;F0 = 20;w = 2 *
pi;beta = 0;gamma = 0.5;
lambda = [];
```

```
for dt = 0.1:0.1:10
lambda = [lambda, max(abs(eig(eval(A))))];
end
dt = 0.1:0.1:10;
plot(dt, lambda);
title('valeur propre et pas de temps');
```



1.3

```
theta1 = 0;
theta2 = 0;
dtheta1 = - 1.31519275;
dtheta2 = - 1.85996342;
```

```
q = [theta1; theta2];
dq = [dtheta1; dtheta2];
d2q = eval(A4) * q0;
```

1.4

Les relations sont comme suivant : on onte Q = [q; dq], et

Qn1 = A * Qn + B, d2q = A4 * q + A5 * sin(w * t).

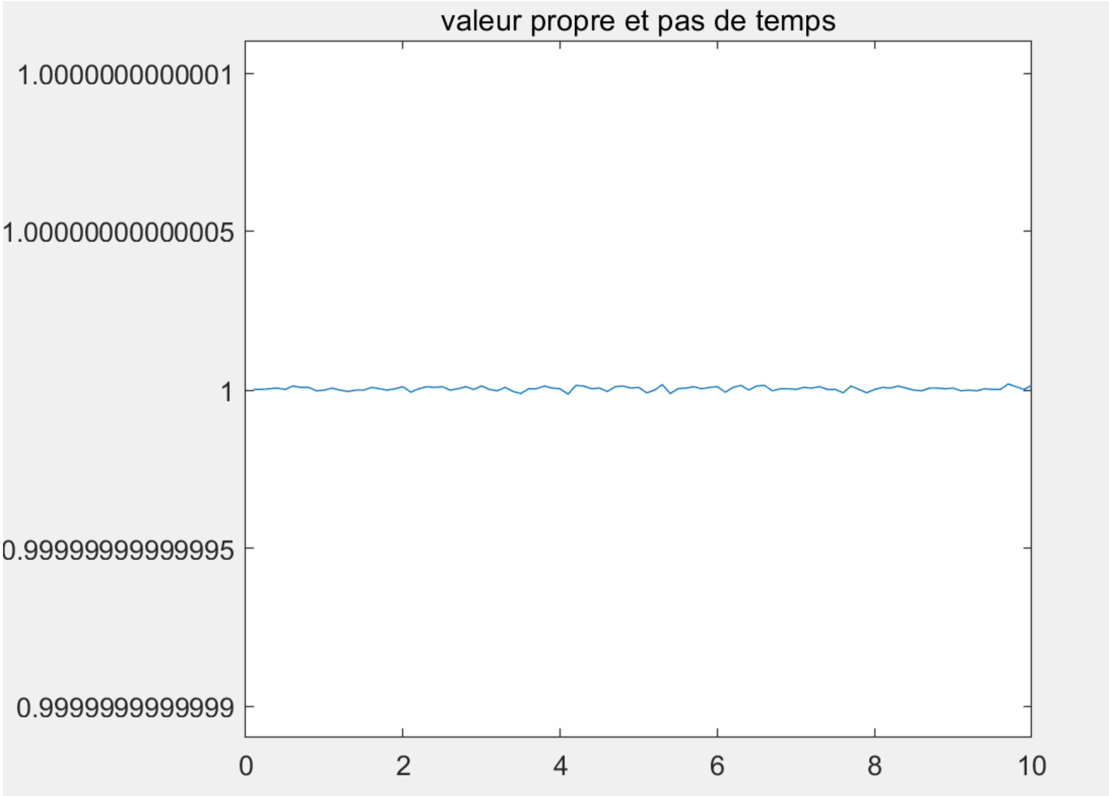
1.5

1.6

2.1C’est le même que 1.1

2.2

```
m = 2;a = 0.5;g = 9.81;F0 = 20;w =
2 * pi;beta = 0.25;gamma = 0.5;
lambda = [];
for dt = 0.1:0.1:10
lambda = [lambda,
max(abs(eig(eval(A))))];
end
dt = 0.1:0.1:10;
plot(dt, lambda);
title('valeur propre et pas de
temps');
```



Le valeur propre ne change pas, elle ne dépend pas de pas de temps.

2.3

Les relations sont comme suivant : on onte Q = [q; dq], et

Qn1 = A * Qn + B, d2q = A4 * q + A5 * sin(w * t).

En fait, ce sont les même que celles de 1.4, seulement le beta change.

2.4

Qn1 = A * Qn + B

2.5

2.6

Sujet 3 Oscillateur nonlinéaire à un degré de liberté

1.1 Démonstration

1.1. t_j — q_j $\dot{q}_j$ $\ddot{q}_j$

t_{j+1} — q_{j+1} $\dot{q}_{j+1}$ $\ddot{q}_{j+1}$

D'après le cas, on a

$$\vec{F}_{ressort\to m} = -kq(1+aq^2)\vec{x_0}$$

$$m\ddot{q} = F_{ressort\to m}$$

$$m\ddot{q} = -kq(1+aq^2)$$

$$\ddot{q} + \frac{k}{m} q(1+aq^2) = 0$$

or $w_0^2 = \frac{k}{m}$

d'où $\ddot{q} + w_0^2 q(1+aq^2) = 0$

D'après le schéme de Newmark avec $\gamma = 0.5$ $\beta = 0$, si on a q_j, $\dot{q}_j$, $\ddot{q}_j$

$$\Rightarrow \begin{cases} q_{j+1} = q_j + \Delta t \dot{q}_j + \Delta t^2 (0.5) \ddot{q}_j \\ \dot{q}_{j+1} = \dot{q}_j + \Delta t \cdot 0.5 \ddot{q}_j + 0.5 \Delta t \ddot{q}_{j+1} \\ \ddot{q}_{j+1} = -\frac{k}{m} q_{j+1}(1 + a\, q_{j+1}^2) \\ \quad = -w_0^2 q_{j+1}(1 + a \cdot q_{j+1}^2) \end{cases}$$

donc on peut déterminer q_{j+1}, $\dot{q}_{j+1}$, $\ddot{q}_{j+1}$ en l'instant t_{j+1}.

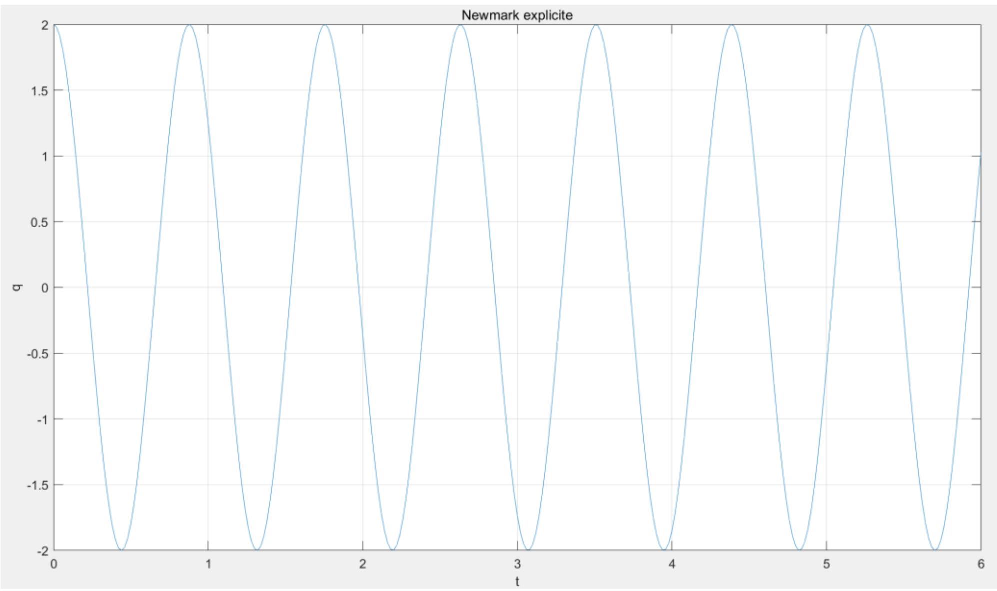
1.2 et 1.3

Script Matlab

```
q0=2;dq0=0;w0=2*pi;T0=6;dt=0.02;a=0.1;

t=(0:dt:T0)';

nb=size(t,1);

q1b=zeros(nb,1);

dq1b=zeros(nb,1);

ddq1b=zeros(nb,1);

q1b(1)=q0;

dq1b(1)=dq0;
```

```
ddq1b(1)=-w0*w0*q1b(1)*(1+a*q1b(1)*q1b(1));
for i=2:nb
    q1b(i)=q1b(i-1)+dt*dq1b(i-
1)+0.5*dt*dt*ddq1b(i-1);
    ddq1b(i)=-
w0*w0*q1b(i)*(1+a*q1b(i)*q1b(i));
    dq1b(i)=dq1b(i-1)+0.5*dt*ddq1b(i-
1)+0.5*dt*ddq1b(i);
end
plot(t,q1b),hold on
grid on;
xlabel('t');
ylabel('q');
title('Newmark explicite')
```

Et on obtient le schémas suivant :



On print le valeur de q à l'instant différant :

```
q1b(1)     q(0s)=2
q1b(2)     q(dt)=1.9779
q1b(3)     q(2*dt)=1.9123
q1b(end)  q(T0)=1.0329
```

2.1

On cherche à minimiser la valeur de l'équation qu'on obtient par équation (2) : ddq1b+w0*w0*q1b*(1+a*q1b*q1b), on veut que cette valeur être égale à 0.

2.2

Calcul de la Correction

$$\Delta q_{j+1} = \beta \cdot \Delta t^2 \cdot \Delta \ddot{q}_{j+1}$$

$$\Delta \dot{q}_{j+1} = \gamma \Delta t \cdot \Delta \ddot{q}_{j+1}$$

$$\ddot{q}^*_{j+1} + \Delta \ddot{q}_{j+1} + w_0^2 (q^*_{j+1} + \Delta q_{j+1}) \times (1 + a \cdot (q^*_{j+1} + \Delta q_{j+1})^2) = 0$$

$$\boxed{\Delta \ddot{q}_{j+1} + w_0^2 \Delta q_{j+1} (1 + a (q^*_{j+1} + \Delta q_{j+1})^2) + w_0^2 q^*_{j+1} \times a \times (q^*_{j+1} + \Delta q_{j+1})^2} = \boxed{-(\ddot{q}^*_{j+1} + w_0^2 \cdot q^*_{j+1})}$$

avec $\Delta q_{j+1} = \beta \Delta t^2 \Delta \ddot{q}_{j+1}$

donc $\Delta \ddot{q}_{j+1} = - \dfrac{\ddot{q}^*_{j+1} + w_0^2 q^*_{j+1} \cdot (1 + a \cdot q^{*2}_{j+1})}{1 + \beta \cdot \Delta t^2 \cdot (3a \cdot w_0^2 q^{*2}_{j+1} + w_0^2)}$

2.3 et 2.4

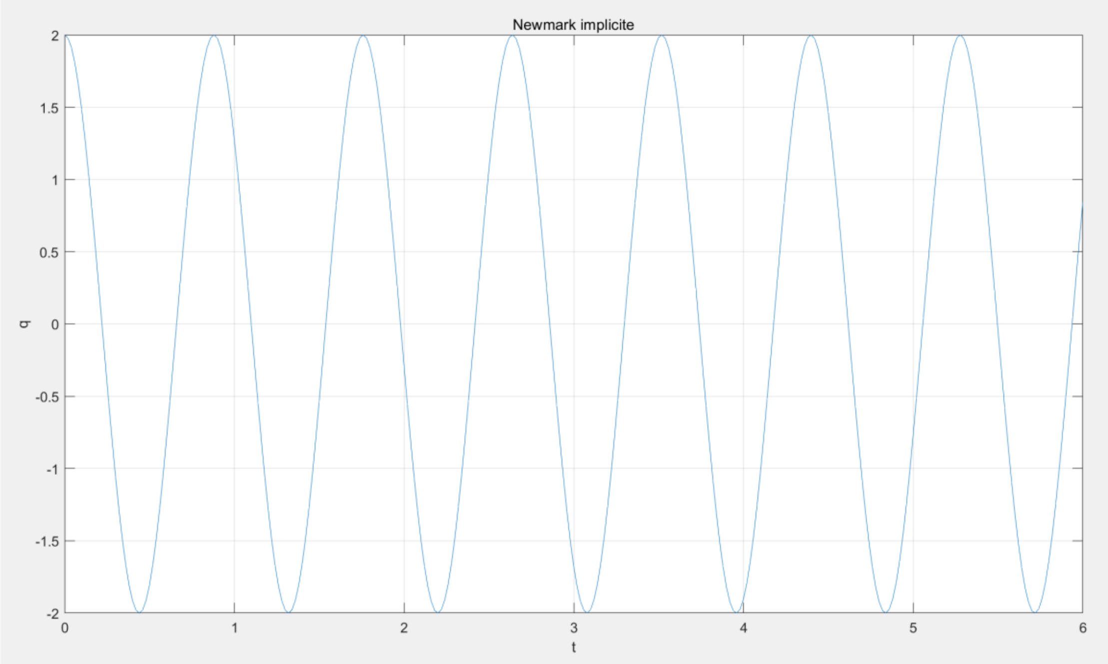
Script Matlab

```
q0=2;dq0=0;w0=2*pi;T0=6;dt=0.02;a=0.1;
r=0.5;b=0.25;
t=(0:dt:T0)';
nb=size(t,1);
q1b=zeros(nb,1);
```

```
dq1b=zeros(nb,1);
ddq1b=zeros(nb,1);
d_q1b=zeros(nb,1);
d_dq1b=zeros(nb,1);
d_ddq1b=zeros(nb,1);
q1b(1)=q0;
dq1b(1)=dq0;
ddq1b(1)=-
w0*w0*q1b0(1)*(1+a*q1b(1)*q1b(1));
for i=2:nb
    ddq1b(i)=0;
    dq1b(i)=dq1b(i-1)+(1-r)*dt*ddq1b(i-1);
    q1b(i)=q1b(i-1)+dt*dq1b(i-1)+(0.5-
b)*dt*dt*ddq1b(i-1);
    d_ddq1b(i)=(-
(ddq1b(i)+w0*w0*q1b(i)*(1+a*q1b(i)*q1b(i)))
)/(1+b*dt*dt*(3*a*w0*w0*q1b(i)*q1b(i)+w0*w0
));
    d_q1b(i)=b*dt*dt*d_ddq1b(i);
    d_dq1b(i)=r*dt*d_ddq1b(i);
    q1b(i)=q1b(i)+d_q1b(i);
    dq1b(i)=dq1b(i)+d_dq1b(i);
```

```
    ddq1b(i)=ddq1b(i)+d_ddq1b(i);
end
plot(t,q1b),hold on
grid on;
xlabel('t');
ylabel('q');
title('Newmark implicite')
```

On obtient



On print le valeur de q à l'instant différant :

```
q1b(1)    q(0s)=2
q1b(2)    q(dt)=1.9781
q1b(3)    q(2*dt)=1.9131
q1b(end)  q(T0)=0.8478
```

3.1

Calcul de l'énergie

Pour l'Energie.

$$E = E_c + E_p$$

$$E_c = \frac{1}{2} m \dot{q}^2$$

$$E_p = \left| \int F \cdot dq \right|$$

Or $F = -kq(1+aq^2)$

$$E_p = \left| \int -kq(1+aq^2)\, dq \right|$$

$$= \int (kq + kaq^3)\, dq$$

$$= \frac{k}{2} q^2 + \frac{ka}{4} q^4$$

donc.

$$E = E_c + E_p$$

$$= \frac{1}{2} m \dot{q}^2 + \frac{k}{2} q^2 + \frac{ak}{4} q^4$$

Or, on ne sait pas k, et m, on peut calculer E/m.

$$E/m = \frac{1}{2} \dot{q}^2 + \frac{1}{2} \underbrace{\frac{k}{m}}_{= w_0^2} q^2 + \frac{a}{4} \frac{k}{m} q^4$$

avec w_0 connu

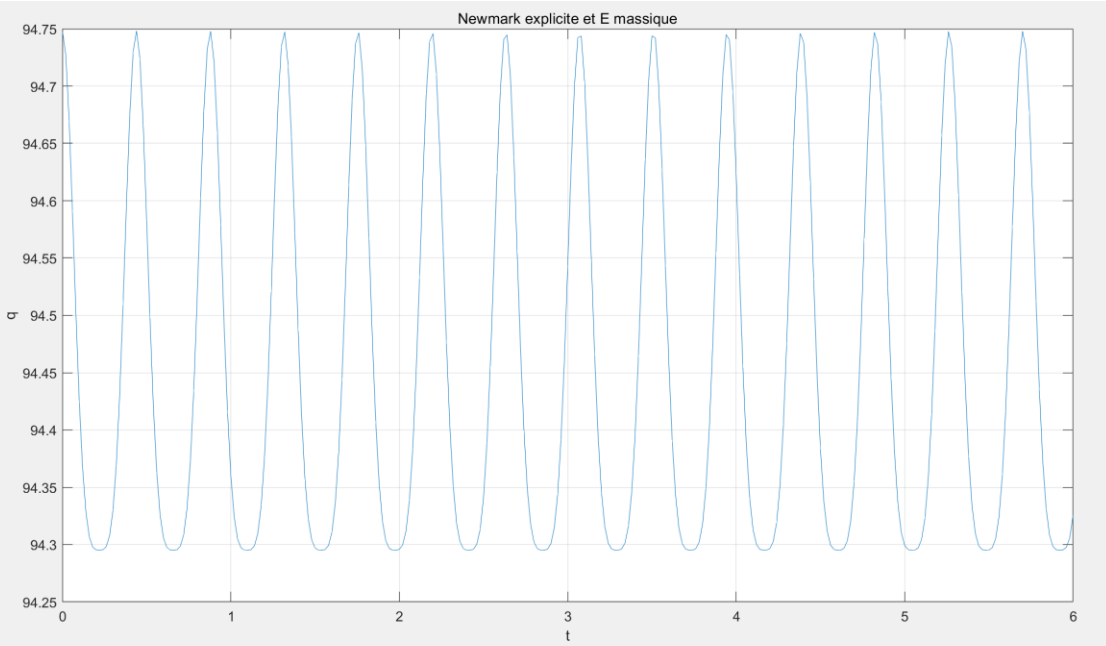
3.2 et 3.3

Pour Newmark explicit

Script

```
q0=2;dq0=0;w0=2*pi;T0=6;dt=0.02;a=0.1;

t=(0:dt:T0)';

nb=size(t,1);

q1b=zeros(nb,1);

dq1b=zeros(nb,1);

ddq1b=zeros(nb,1);

E_massique=zeros(nb,1);

q1b(1)=q0;
```

```
dq1b(1)=dq0;
ddq1b(1)=-w0*w0*q1b(1)*(1+a*q1b(1)*q1b(1));
E_massique(1)=1/2*dq1b(1)*dq1b(1)+1/2*w0*w0
*q1b(1)^2+a/4*w0*w0*q1b(1)^4;%E_massique
for i=2:nb
    q1b(i)=q1b(i-1)+dt*dq1b(i-
1)+0.5*dt*dt*ddq1b(i-1);
    ddq1b(i)=-
w0*w0*q1b(i)*(1+a*q1b(i)*q1b(i));
    dq1b(i)=dq1b(i-1)+0.5*dt*ddq1b(i-
1)+0.5*dt*ddq1b(i);

E_massique(i)=1/2*dq1b(i)*dq1b(i)+1/2*w0*w0
*q1b(i)^2+a/4*w0*w0*q1b(i)^4;%E_massique
end
plot(t,E_massique),hold on
grid on;
xlabel('t');
ylabel('q');
title('Newmark explicite et E massique')
```



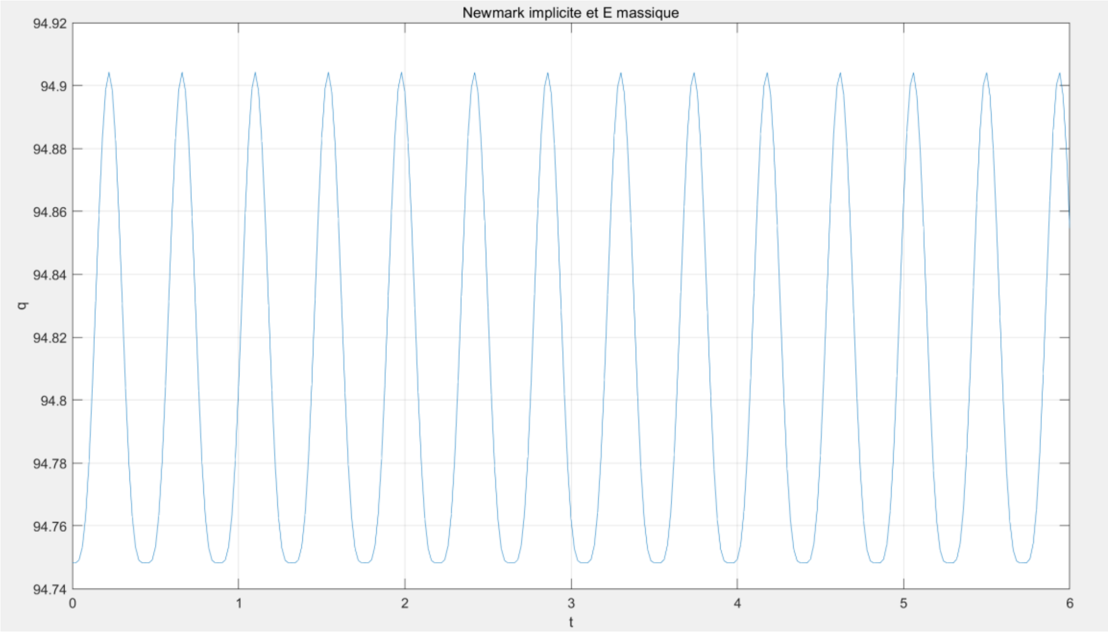
Pour Newmark implicit

Script

```
q0=2;dq0=0;w0=2*pi;T0=6;dt=0.02;a=0.1;

r=0.5;b=0.25;

t=(0:dt:T0)';

nb=size(t,1);

q1b=zeros(nb,1);

dq1b=zeros(nb,1);

ddq1b=zeros(nb,1);

d_q1b=zeros(nb,1);

d_dq1b=zeros(nb,1);

d_ddq1b=zeros(nb,1);

E_massique=zeros(nb,1);

q1b(1)=q0;
```

```
dq1b(1)=dq0;
ddq1b(1)=-w0*w0*q1b(1)*(1+a*q1b(1)*q1b(1));
E_massique(1)=1/2*dq1b(1)*dq1b(1)+1/2*w0*w0
*q1b(1)^2+a/4*w0*w0*q1b(1)^4;%E_massique
for i=2:nb
    ddq1b(i)=0;
    dq1b(i)=dq1b(i-1)+(1-r)*dt*ddq1b(i-1);
    q1b(i)=q1b(i-1)+dt*dq1b(i-1)+(0.5-
b)*dt*dt*ddq1b(i-1);
    d_ddq1b(i)=(-
(ddq1b(i)+w0*w0*q1b(i)*(1+a*q1b(i)*q1b(i)))
)/(1+b*dt*dt*(3*a*w0*w0*q1b(i)*q1b(i)+w0*w0
));
    d_q1b(i)=b*dt*dt*d_ddq1b(i);
    d_dq1b(i)=r*dt*d_ddq1b(i);
    q1b(i)=q1b(i)+d_q1b(i);
    dq1b(i)=dq1b(i)+d_dq1b(i);
    ddq1b(i)=ddq1b(i)+d_ddq1b(i);

E_massique(i)=1/2*dq1b(i)*dq1b(i)+1/2*w0*w0
*q1b(i)^2+a/4*w0*w0*q1b(i)^4;%E_massique
end
```

```
plot(t,E_massique),hold on
grid on;
xlabel('t');
ylabel('q');
title('Newmark implicite et E massique')
```



Si on met les 2 ensemble