Michel GAO Minghui 15241017 ZY192410

Oscillateur non linéaire à un degré de liberté

1. Résolution avec un schéma de NEWMARK explicite

On a $\gamma = 0.5$ et $\beta = 0$

Donc on peut simplifier les relation (4) et (5) de l'exercice :

$$q_{i+1} = q_i + \Delta t \dot{q}_i + 0.5 \Delta t^2 \ddot{q}_i$$

$$\dot{q}_{i+1} = \dot{q}_i + 0.5 \Delta t \ddot{q}_i + 0.5 \Delta t \ddot{q}_{i+1}$$

On a aussi la relation :

$$\ddot{q} + \omega_0^2 q (1 + a q^2) = 0$$

Je note $f(q,t) = -\omega_0^2 q(1 + a q^2)$:

$$\ddot{q} = f(q,t)$$

1.1

Si q_i $\dot{q}_i$ $\ddot{q}_i$ sont quantités connues :

$$q_{i+1} = q_i + \Delta t \dot{q}_i + 0.5 \Delta t^2 f(q_i, t)$$

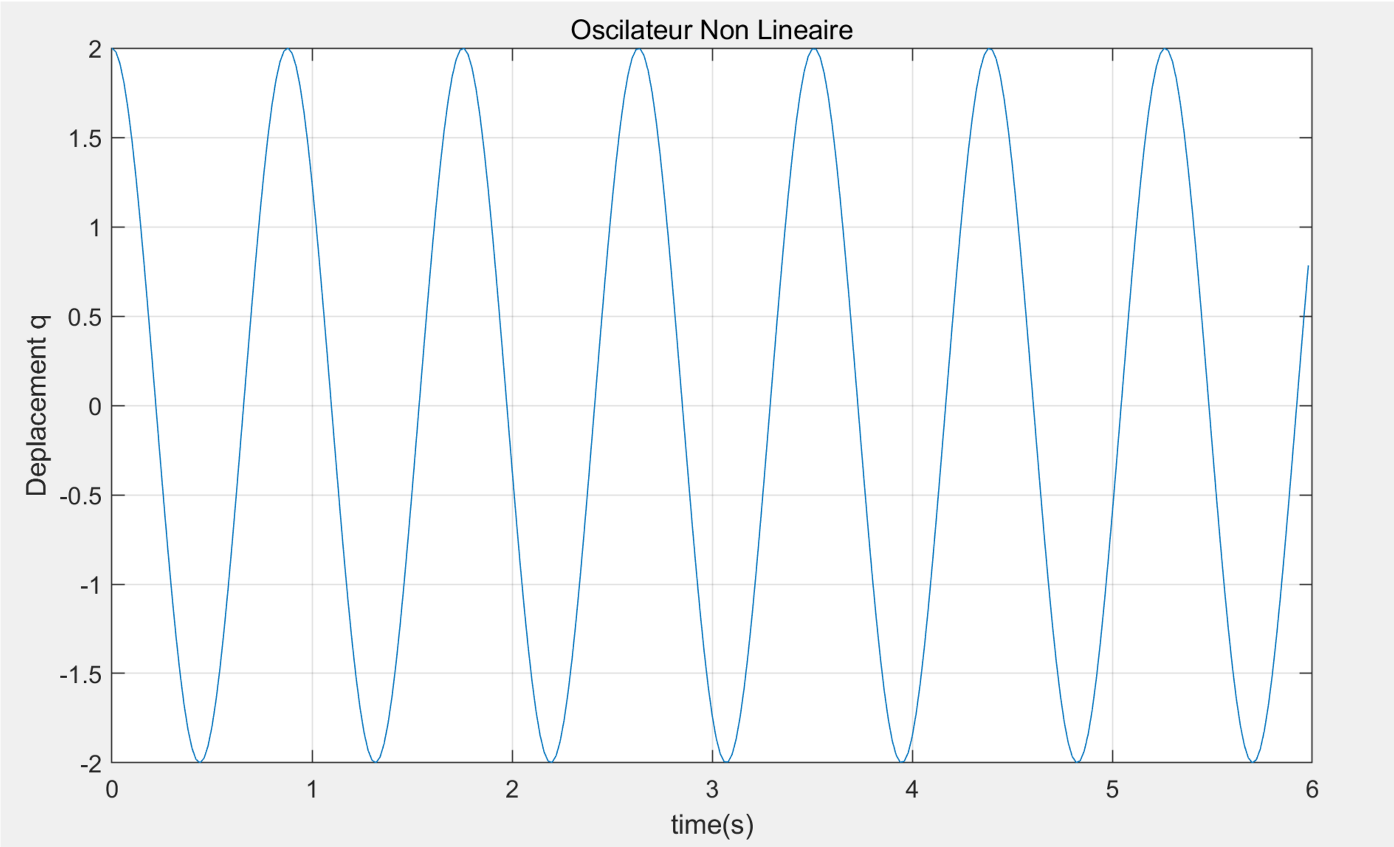
$$\dot{q}_{i+1} = \dot{q}_i + 0.5 \Delta t f(q_i, t) + 0.5 \Delta t\, f\left(q_{i+1}, t + \Delta t\right)$$

$$\ddot{q}_{i+1} = f\left(q_{i+1}, t + \Delta t\right)$$

1.2

Le code de Matlab :

```
omega0 = 2*pi;
a = 0.1;
interval = 6;
pas = 0.02;
steps = floor(interval/pas);
time_list = 0:pas:(interval-pas);
q = zeros(1,steps);
dq = zeros(1, steps);
q(1)=2;
dq(1)=0;
for i = 1:(steps-1)
    q(i+1) = q(i) + pas *dq(i) + 0.5*pas^2*(-omega^2*q(i)*(1+a*q(i)^2));
    dq(i+1) = dq(i) + 0.5*pas*(-omega^2*q(i)*(1+a*q(i)^2)) + 0.5*pas*(-omega^2*q(i+1)*(1+a*q(i+1)^2));
end
plot(time_list, q)
grid on;
xlabel('time(s)')
ylabel('Deplacement q')
```



1.3 $\Delta t = 0.02s$

Pour $t = 0s$:

$$q(t) = 2$$

Pout $t = \Delta t$:

$$q(t) = 1.9779$$

Pour $t = 2\Delta t$:

$$q(t) = 1.9123$$

Pour $t = T_0$:

$$q(t) = 0.7837$$

2. Résolution avec un schéma de NEWMARK implicite

On a $\gamma = 0.5$ et $\beta = 0.25$

2.1 On cherche à minimiser le résidu ϵ

2.2

A partir des relation (4) et (5), on obtient les relations suivantes :

$$\Delta q_{i+1} = 0.25\Delta t^2 \Delta \ddot{q}_{i+1}$$

$$\Delta \dot{q}_{i+1} = 0.5\Delta t \Delta \ddot{q}_{i+1}$$

Il nous faut transformer la formule suivante pour obtenir la correction :

$$\ddot{q} + \omega_0^2 q(1 + aq^2) = 0$$

Je note :

$$f(\ddot{q}, q) = \ddot{q} + \omega_0 q(1 + aq^2)$$

$$\Downarrow$$

$$f(\ddot{q}, q) = 0$$

On remplace q par $q_{i+1}^* + \Delta q_{i+1}$, et $\ddot{q}$ par $\ddot{q}_{i+1}^* + \Delta \ddot{q}_{i+1}$:

$$f\left(\ddot{q}_{i+1}^* + \Delta \ddot{q}_{i+1}, q_{i+1}^* + \Delta q_{i+1}\right) = 0$$

$$\Downarrow$$

$$\ddot{q}_{i+1}^* + \Delta \ddot{q}_{i+1} + \omega_0^2\left(q_{i+1}^* + \Delta q_{i+1}\right)\left(1 + a\left(q_{i+1}^* + \Delta q_{i+1}\right)^2\right) = 0$$

On obtient finalement que la correction appliquée à $\ddot{q}_{j+1}^*$ est :

$$\Delta \ddot{q}_{n+1} = -\frac{f(\ddot{q}_{i+1}^*, q_{i+1}^*)}{\frac{\partial f(\ddot{q}_{i+1}^*, q_{i+1}^*)}{\partial \ddot{q}_{i+1}^*} + 0.25\frac{\partial f(\ddot{q}_{i+1}^*, q_{i+1}^*)}{\partial q_{n+1}^*}\Delta t^2}$$

Dans notre cas :

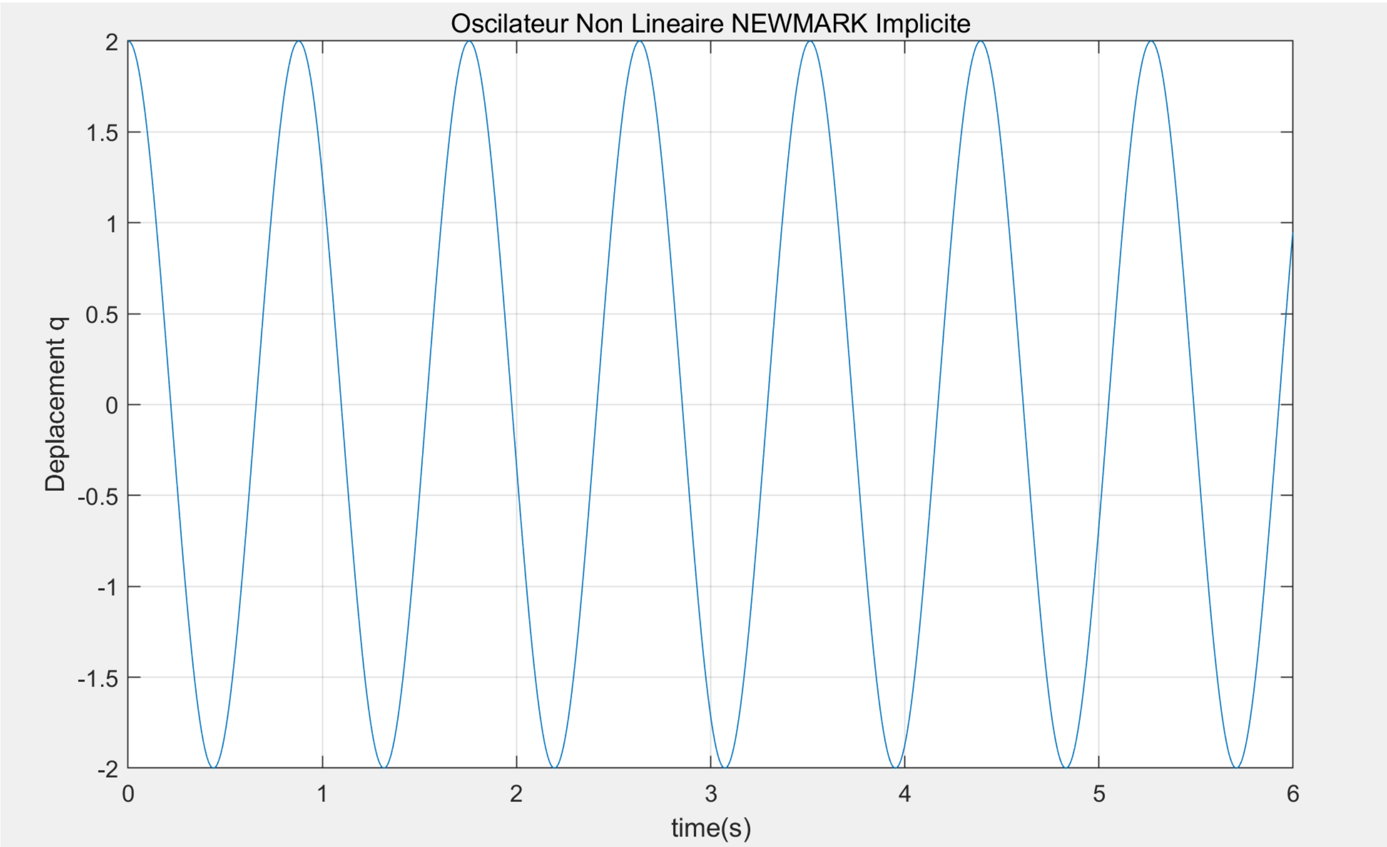
$$\Delta \ddot{q}_{n+1} = -\frac{\ddot{q}_{i+1}^* + \omega_0^2 q_{i+1}^*\left(1 + a{q_{i+1}^*}^2\right)}{1 + 0.25\omega_0^2\left(1 + 3a{q_{i+1}^*}^2\right)\Delta t^2}$$

2.3

Le code de Matlab :

```
close all;
omega = 2*pi;
a = 0.1;
interval = 6;
pas = 0.002;
steps = floor(interval/pas);
time_list = 0:pas:(interval-pas);
q = zeros(1,steps);
```

```
dq = zeros(1, steps);
ddq = zeros(1, steps);
q(1)=2;
dq(1)=0;
ddq(1)=-omega^2*q(1)*(1+a*q(1)^2);
epsilon=0.000001; % la precision
for i = 1:(steps-1)
    prediction_ddq = 0;
    prediction_dq = dq(i) + pas*0.5*ddq(i);
    prediction_q = q(i) + pas*dq(i)+ pas^2*0.25*ddq(i);
   while abs(prediction_ddq + omega^2*prediction_q*(1+a*prediction_q^2)) > epsilon
       corr_ddq = -(prediction_ddq+omega^2*prediction_q*(1+a*prediction_q^2))/(1+0.25*omega^2*pas^2*(1+3*a*prediction_q^2));
       corr_dq = 0.5*pas*corr_ddq;
       corr_q = 0.25*pas^2*corr_ddq;
       prediction_ddq = prediction_ddq + corr_ddq;
       prediction_dq = prediction_dq + corr_dq;
       prediction_q = prediction_q + corr_q;
   end
   ddq(i+1) = prediction_ddq;
   dq(i+1) = prediction_dq;
   q(i+1) = prediction_q;
end
plot(time_list, q)
grid on;
xlabel('time(s)')
ylabel('Deplacement q')
title('Oscilateur Non Lineaire NEWMARK Implicite')
```



2.4 $\Delta t = 0.02s$

J'ai fixé une précision de $\epsilon = 0.000001$, mais évidemment, pour de différentes précisions, on obtiendra de différents résultats.

Pour $t = 0s$:

$$q(t) = 2$$

Pour $t = \Delta t$:

$$q(t) = 1.9781$$

Pour $t = \Delta t$:

$$q(t) = 1.9131$$

Pour $t = T_0$:

$$q(t) = 0.5893$$

3. Energie mécanique

3.1 L'énergie mécanique pour cet oscillateur non linéaire est définie par la somme de l'énergie cinétique E_c et l'énergie potentielle E_p

$$E_{mec} = E_c + E_p$$

Pour l'énergie cinétique, on a :

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

Pour l'énergie potentielle, on a :

$$E_p(t) = \int_{q(t=0)}^{q(t)} \vec{F}_{m\rightarrow ressort}(q)\ dq$$

3.2 Je pose $k = (2\pi)^2$ et $m = 1\ kg$

Le code de Matlab :

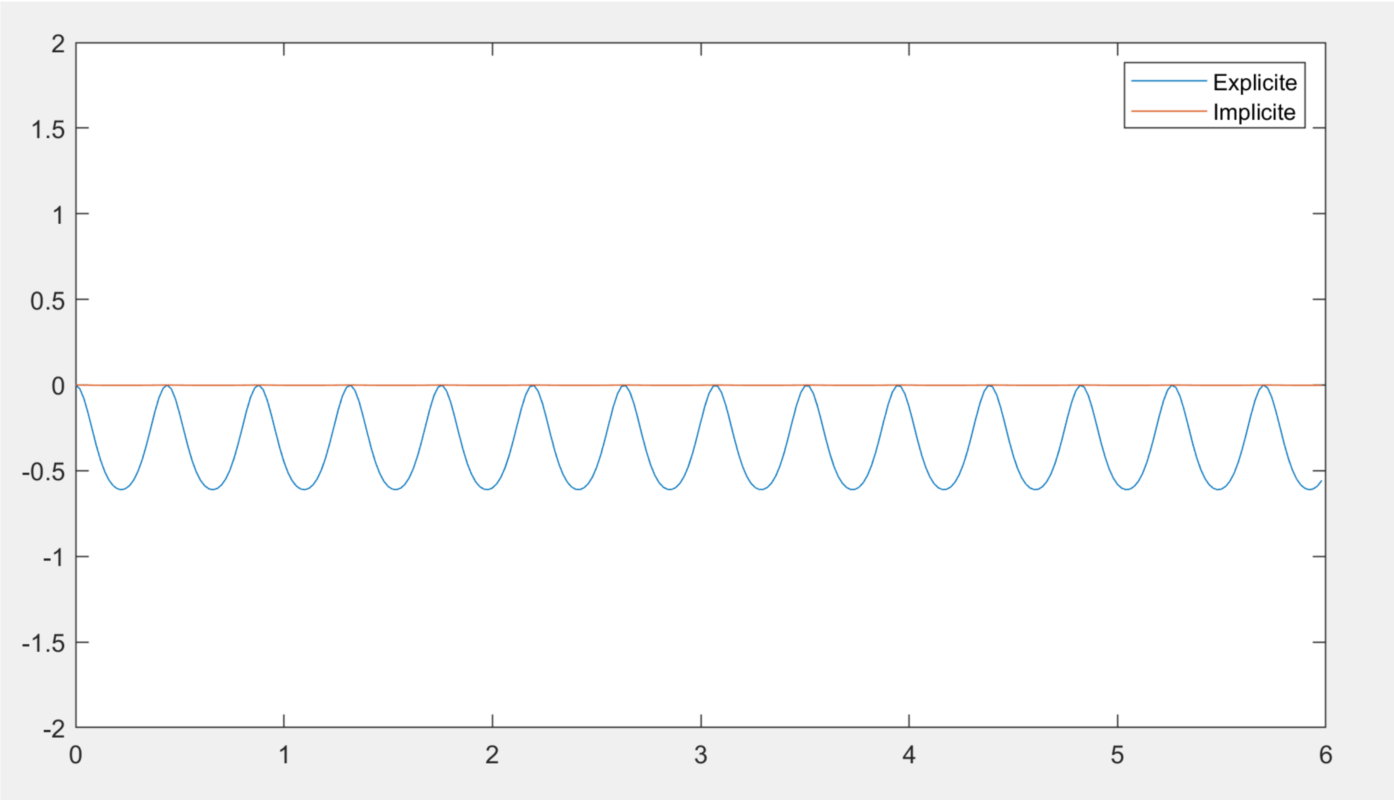
```
clear all;
close all;
omega = 2*pi;
a = 0.1;
interval = 6;
pas = 0.02;
steps = floor(interval/pas);
time_list = 0:pas:(interval-pas);
k = (2*pi)^2;
m = 1;

% NEWMARK explicite
q = zeros(1,steps);
dq = zeros(1, steps);
q(1)=2;
dq(1)=0;
for i = 1:(steps-1)
```

```
    q(i+1) = q(i) + pas *dq(i) + 0.5*pas^2*(-omega^2*q(i)*(1+a*q(i)^2));
    dq(i+1) = dq(i) + 0.5*pas*(-omega^2*q(i)*(1+a*q(i)^2)) + 0.5*pas*(-omega^2*q(i+1)*(1+a*q(i+1)^2));
end
temp_q = q(2:300);
temp_q(300) = 0;
dX = temp_q - q;
F0 = k.*q.*(1+ a.*q.^2);
temp_F = F0(2:300);
temp_F(300) = 0;
F = (F0 + temp_F)/2; % valeur centrale
dEp = F.*dX;
temp_Ep = cumsum(dEp);
Ep(1)=0;
Ep(2:300) = temp_Ep(1:299); % Un peu de dephasage
Ec = 0.5*dq.^2;
Emec_explicite = Ep + Ec;

% implicite
q = zeros(1,steps);
dq = zeros(1, steps);
ddq = zeros(1, steps);
q(1)=2;
dq(1)=0;
ddq(1)=-omega^2*q(1)*(1+a*q(1)^2);
epsilon=0.01; % la precision
for i = 1:(steps-1)
    prediction_ddq = 0;
    prediction_dq = dq(i) + pas*0.5*ddq(i);
    prediction_q = q(i) + pas*dq(i)+ pas^2*0.25*ddq(i);
   while abs(prediction_ddq + omega^2*prediction_q*(1+a*prediction_q^2)) > epsilon
       corr_ddq = -
(prediction_ddq+omega^2*prediction_q*(1+a*prediction_q^2))/(1+0.25*omega^2*pas^2*(1+3*a*prediction_q^2));
       corr_dq = 0.5*pas*corr_ddq;
       corr_q = 0.25*pas^2*corr_ddq;
       prediction_ddq = prediction_ddq + corr_ddq;
       prediction_dq = prediction_dq + corr_dq;
       prediction_q = prediction_q + corr_q;
   end
   ddq(i+1) = prediction_ddq;
   dq(i+1) = prediction_dq;
   q(i+1) = prediction_q;
end
temp_q = q(2:300);
temp_q(300) = 0;
dX = temp_q - q;
F0 = k.*q.*(1+ a.*q.^2);
temp_F = F0(2:300);
temp_F(300) = 0;
F = (F0 + temp_F)/2; % valeur centrale
```

```
dEp = F.*dX;
temp_Ep = cumsum(dEp);
Ep(1)=0;
Ep(2:300) = temp_Ep(1:299); % Un peu de dephasage
Ec = 0.5*dq.^2;
Emec_implicite = Ep + Ec;
plot(time_list,Emec_explicite,time_list,Emec_implicite)
legend('Explicite', 'Implicite')
ylim([-2,2])
```



3.3 $\Delta t = 0.02s$

Les deux énergies mécanique ne sont pas synchronisées.