

# Etude d'un oscillateur linéaire amorti à un degré de liberté

## 1.1 Euler explicite

### 1.1.a

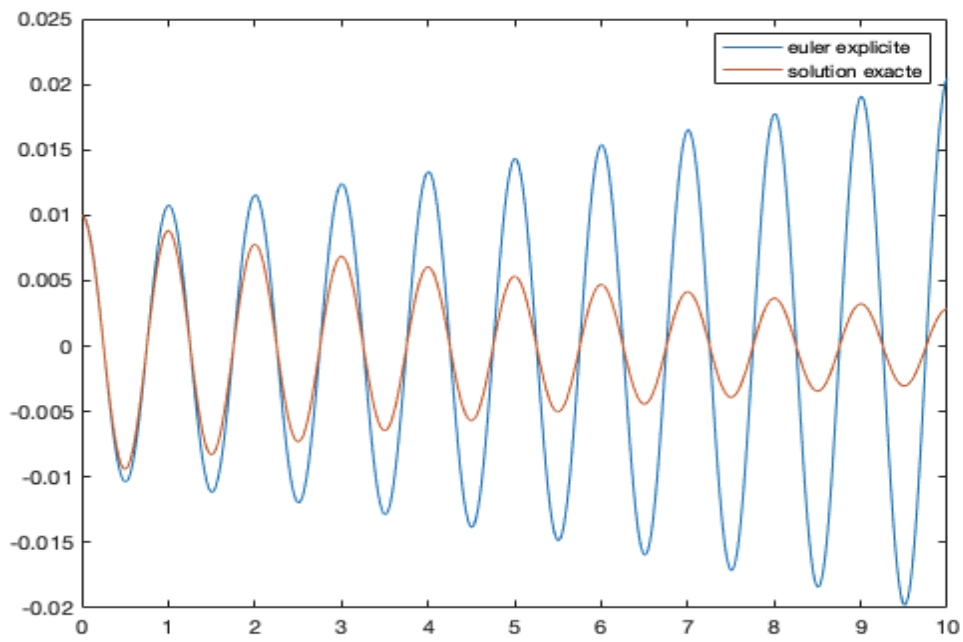
Quand on choisit  $dt > 2\varepsilon/\omega_0$  ( $dt=0.01$ ) :

```
clear all;
T0 = 1;
w0 = 2*pi/T0;
eps = 0.02;
Tc = 2*eps/w0;
x0 = 0.01;
Dx0 = 0;
omega = w0*sqrt(1-eps^2);
```

### 1.1

```
%on choisit dt>Tc
n = 1000;
dt = 10*T0/n;
t = 0:dt:10*T0;
A1 = [1,dt;-w0^2*dt,1-2*eps*w0*dt];
U1(:,1) = [x0;Dx0];
for j = 1:length(t)-1
    U1(:,j+1) = A1*U1(:,j);
end
plot(t,U1(1,:))
hold on;
plot(t,exp(-eps*w0*t).*(x0*cos(omega*t)+(eps*w0*x0+Dx0)/omega.*sin(omega*t)));
legend('euler explicite','solution exacte')
```

On peut obtenir le résultat comme ci-dessous :



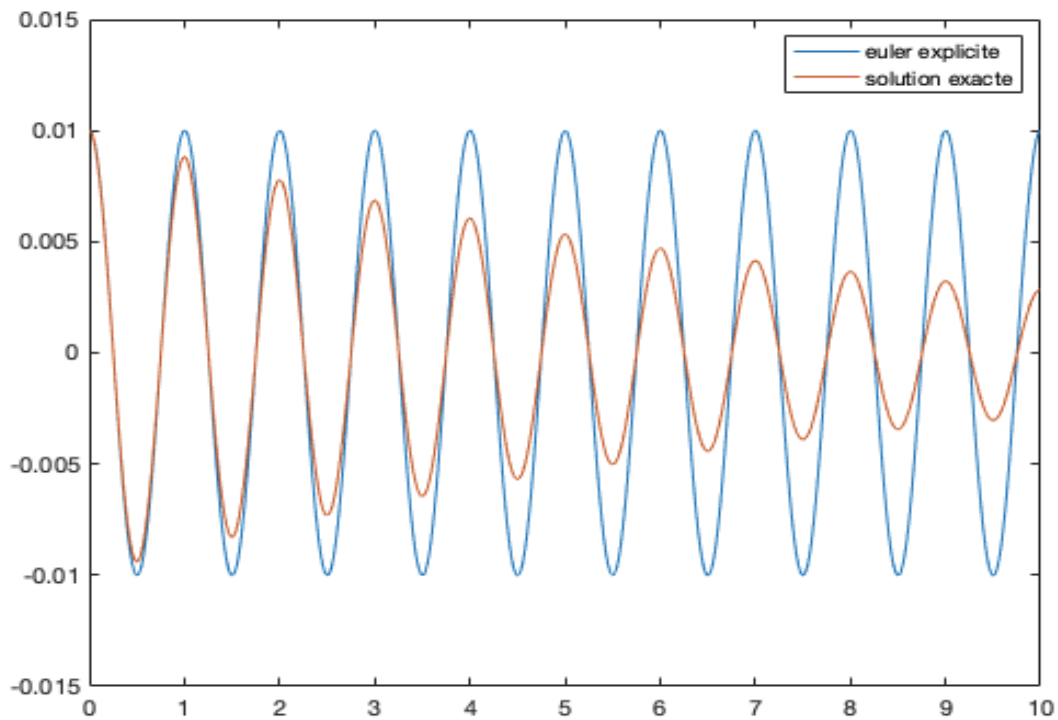
On peut trouver que quand le pas de temps est assez grand (supérieur à  $2\varepsilon/\omega_0$ ), la solution d'Euler explicite est croissante, alors que la solution exacte est amortie.

## 1.1.b

Quand on choisit  $dt = 2\varepsilon/\omega_0$  :

```
%on choisit dt=Tc
dt = Tc;
t = 0:dt:10*T0;
A1 = [1,dt;-w0^2*dt,1-2*eps*w0*dt];
U1(:,1) = [x0;Dx0];
for j = 1:length(t)-1
    U1(:,j+1) = A1*U1(:,j);
end
plot(t,U1(1,:))
hold on;
plot(t,exp(-eps*w0*t).*(x0*cos(omega*t)+(eps*w0*x0+Dx0)/omega.*sin(omega*t)));
legend('euler explicite','solution exacte')
```

On peut obtenir le résultat comme ci-dessous :

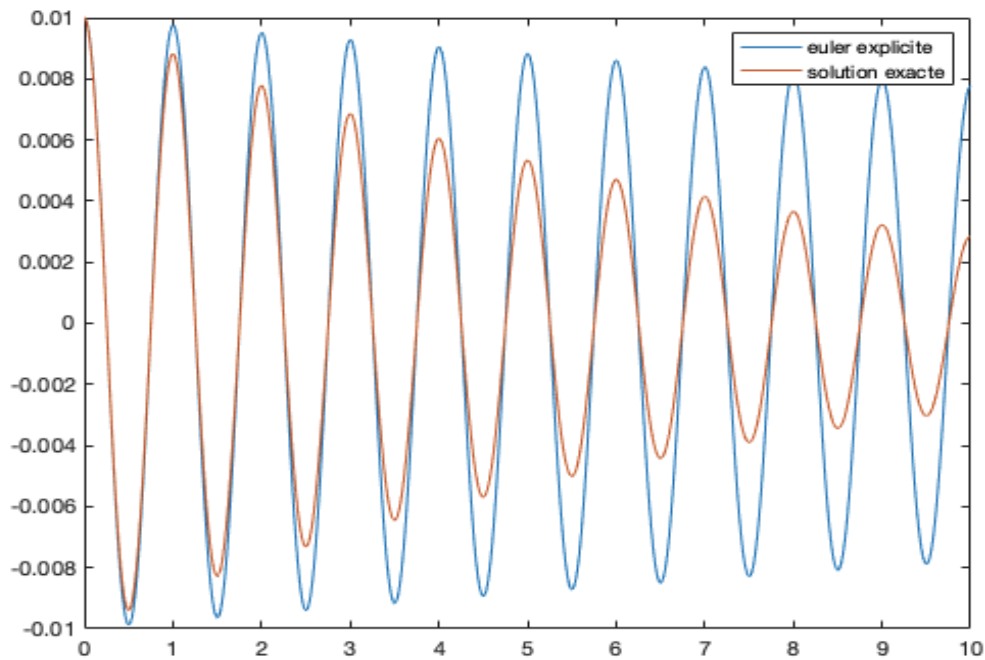


D'après la figure, quand le pas de temps égale  $2\varepsilon/\omega_0$ , la solution d'Euler explicite est l'oscillation sans amortissement, alors que la solution exacte est amortie.

### 1.1.c

Quand on choisit  $dt = 0.8 * 2\varepsilon / \omega_0$  :

```
%on choisit dt=0.8*Tc
dt = 0.8*Tc;
t = 0:dt:10*T0;
A1 = [1,dt;-w0^2*dt,1-2*eps*w0*dt];
U1(:,1) = [x0;Dx0];
for j = 1:length(t)-1
    U1(:,j+1) = A1*U1(:,j);
end
plot(t,U1(1,:))
hold on;
plot(t,exp(-eps*w0*t).*(x0*cos(omega*t)+(eps*w0*x0+Dx0)/omega.*sin(omega*t)));
legend('euler explicite','solution exacte')
```



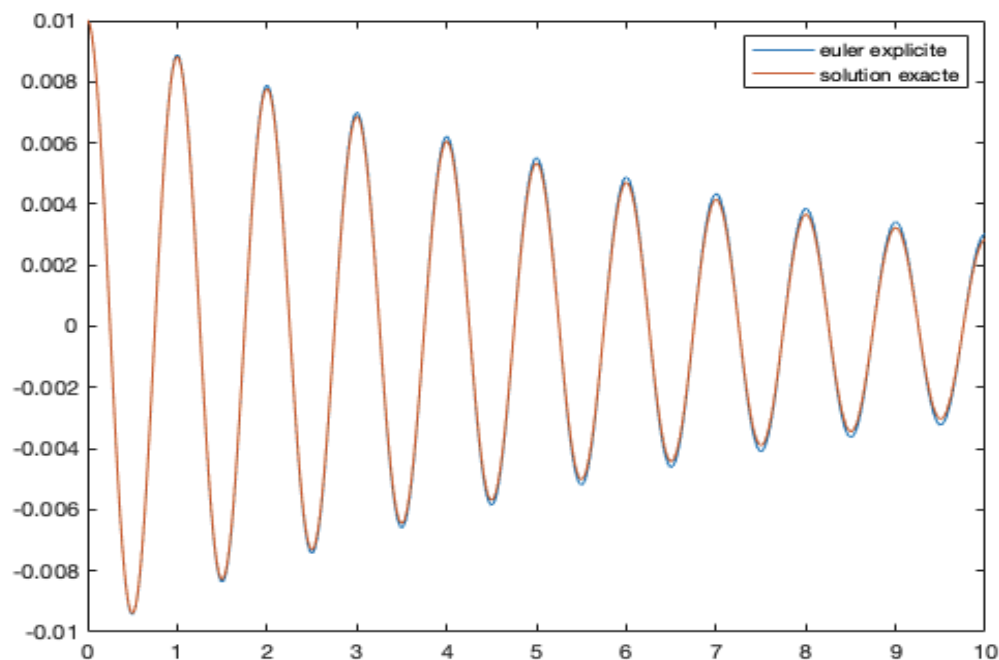
On peut voir que quand le pas de temps est petit à  $2\varepsilon / \omega_0$ , la solution d'Euler explicite est amortie comme la solution exacte.

### 1.1.d

- Les critères permettant d'étudier la précision de la solution sont: la cohérence de croissance, la propagation des erreurs numériques, etc.
- Quand on met le rapport  $dt/(2\varepsilon/\omega_0)$  inférieur à 0.05, la solution calculée présente une précision suffisante.

```
%on choisit dt=0.8*Tc
dt = 0.05*Tc;
t = 0:dt:10*T0;
A1 = [1,dt;-w0^2*dt,1-2*eps*w0*dt];
%calculer le module des valeurs propres
vp = eig(A1);
module_vp = abs(vp(1))
U1(:,1) = [x0;Dx0];
for j = 1:length(t)-1
    U1(:,j+1) = A1*U1(:,j);
end
plot(t,U1(1,:))
hold on;
plot(t,exp(-eps*w0*t).*(x0*cos(omega*t)+(eps*w0*x0+Dx0)/omega.*sin(omega*t)));
legend('euler explicite','solution exacte')
```

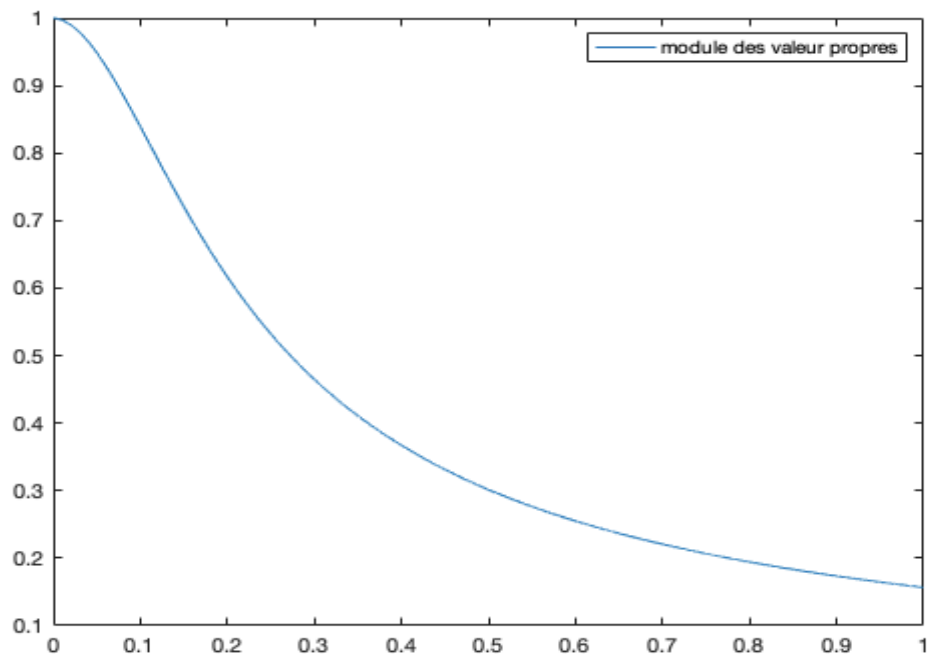
```
module_vp =
    1.0000
```



## 1.2 Euler implicite

Si on choisit la méthode d'Euler implicite, la solution calculée est toujours amortie. Et on calcule le module des valeurs propres de la matrice d'amplification :

```
clf;
%déterminer la matrice d'amplification A2
B = [1,-dt;w0^2*dt,1+2*eps*w0*dt];
A2 = inv(B);
dt_test = 0:0.0001:1;
%trouver le pas de temps critique
for i = 1:length(dt_test)
    B_test(1,i) = [1,-dt_test(i);w0^2*dt_test(i),1+2*eps*w0*dt_test(i)];
    A_test(1,i) = inv(B_test(1,i));
    vpA_test(:,i) = eig(A_test(1,i));
    module_test(i) = sqrt(real(vpA_test(1,i))^2+imag(vpA_test(1,i))^2);
end
plot(dt_test,module_test);
legend('module des valeur propres');
```



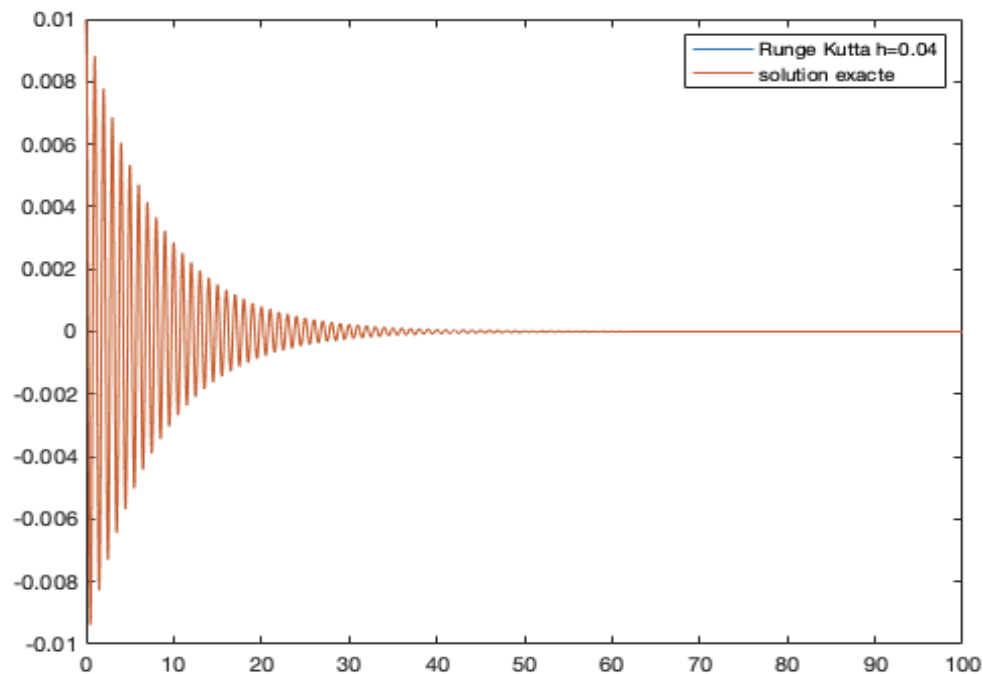
Le module est toujours inférieur à 1, donc il n'y a pas de pas de temps critique.

## 1.3 Runge-Kutta

### 1.3.a

Quand  $h = 0.04$  :

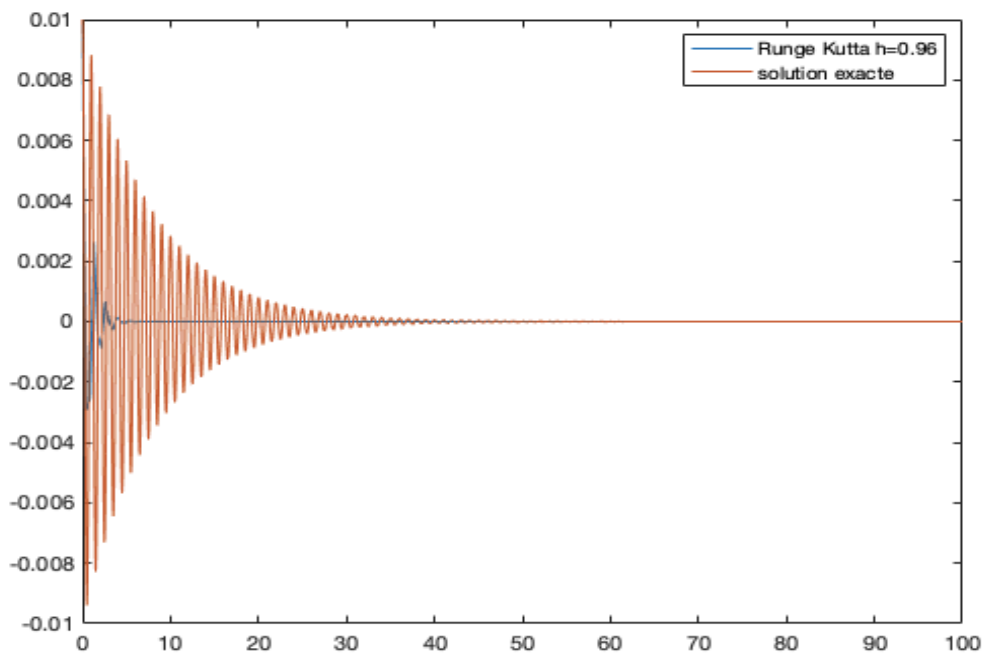
```
clf;
%quand h=0.04
h = 0.04;
dt_rk = h*2*sqrt(2)/w0;
t_rk = 0:dt_rk:100*T0;
C = [0,1;-w0^2,-2*eps*w0];
Urk(:,1) = [x0;Dx0];
for i = 1:length(t_rk)-1
    k1 = C*Urk(:,i);
    k2 = C*(Urk(:,i)+1/2*dt_rk*k1);
    k3 = C*(Urk(:,i)+1/2*dt_rk*k2);
    k4 = C*(Urk(:,i)+dt_rk*k3);
    K = 1/6*(k1+2*k2+2*k3+k4);
    Urk(:,i+1) = Urk(:,i)+K*dt_rk;
end
%comparer les figures
plot(t_rk,Urk(1,:))
hold on;
plot(t,exp(-eps*w0*t).*(x0*cos(omega*t)+(eps*w0*x0+Dx0)/omega.*sin(omega*t)));
legend('Runge Kutta h=0.04','solution exacte');
```



Le résultat est très proche à la solution exacte, stable et précis.

Quand  $h = 0.96$  :

```
clf;
%quand h=0.96
h = 0.96;
dt_rk = h*2*sqrt(2)/w0;
t_rk = 0:dt_rk:100*T0;
C = [0,1;-w0^2,-2*eps*w0];
Urk(:,1) = [x0;Dx0];
for i = 1:length(t_rk)-1
    k1 = C*Urk(:,i);
    k2 = C*(Urk(:,i)+1/2*dt_rk*k1);
    k3 = C*(Urk(:,i)+1/2*dt_rk*k2);
    k4 = C*(Urk(:,i)+dt_rk*k3);
    K = 1/6*(k1+2*k2+2*k3+k4);
    Urk(:,i+1) = Urk(:,i)+K*dt_rk;
end
%comparer les figures
plot(t_rk,Urk(1,:))
hold on;
plot(t,exp(-eps*w0*t).*(x0*cos(omega*t)+(eps*w0*x0+Dx0)/omega.*sin(omega*t)));
legend('Runge Kutta h=0.96','solution exacte');
```

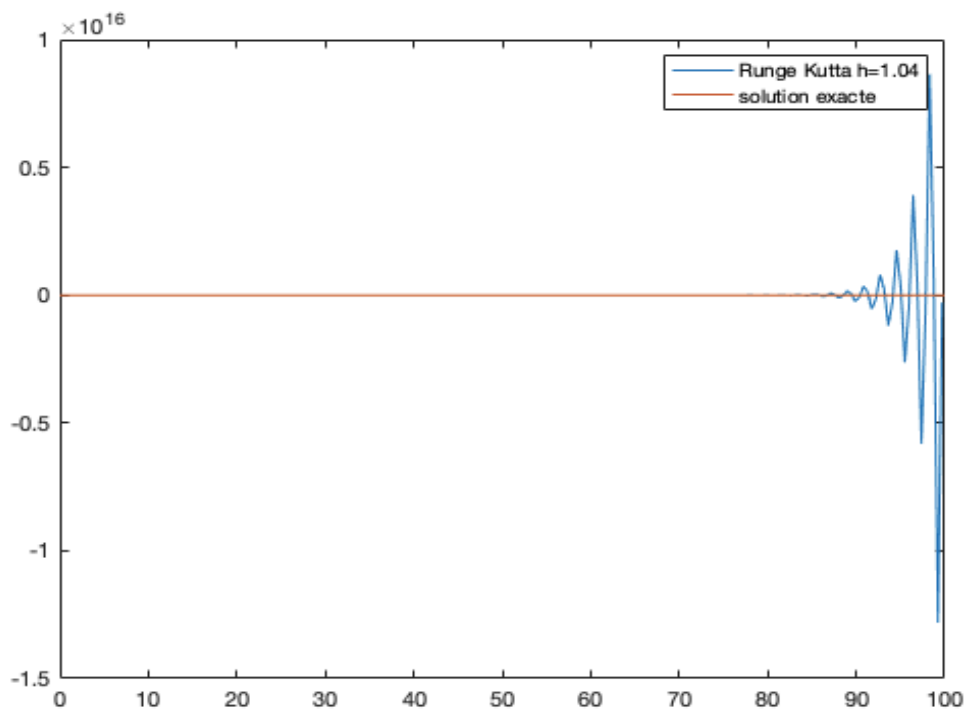


Le résultat est stable mais pas précis de la solution exacte.



Quand  $h= 1.04$  :

```
clf;
%quand h=1.04
h = 1.04;
dt_rk = h*2*sqrt(2)/w0;
t_rk = 0:dt_rk:100*T0;
C = [0,1;-w0^2,-2*eps*w0];
Urk(:,1) = [x0;Dx0];
for i = 1:length(t_rk)-1
    k1 = C*Urk(:,i);
    k2 = C*(Urk(:,i)+1/2*dt_rk*k1);
    k3 = C*(Urk(:,i)+1/2*dt_rk*k2);
    k4 = C*(Urk(:,i)+dt_rk*k3);
    K = 1/6*(k1+2*k2+2*k3+k4);
    Urk(:,i+1) = Urk(:,i)+K*dt_rk;
end
%comparer les figures
plot(t_rk,Urk(1,:))
hold on;
plot(t,exp(-eps*w0*t).*(x0*cos(omega*t)+(eps*w0*x0+Dx0)/omega.*sin(omega*t)));
legend('Runge Kutta h=1.04','solution exacte');
```

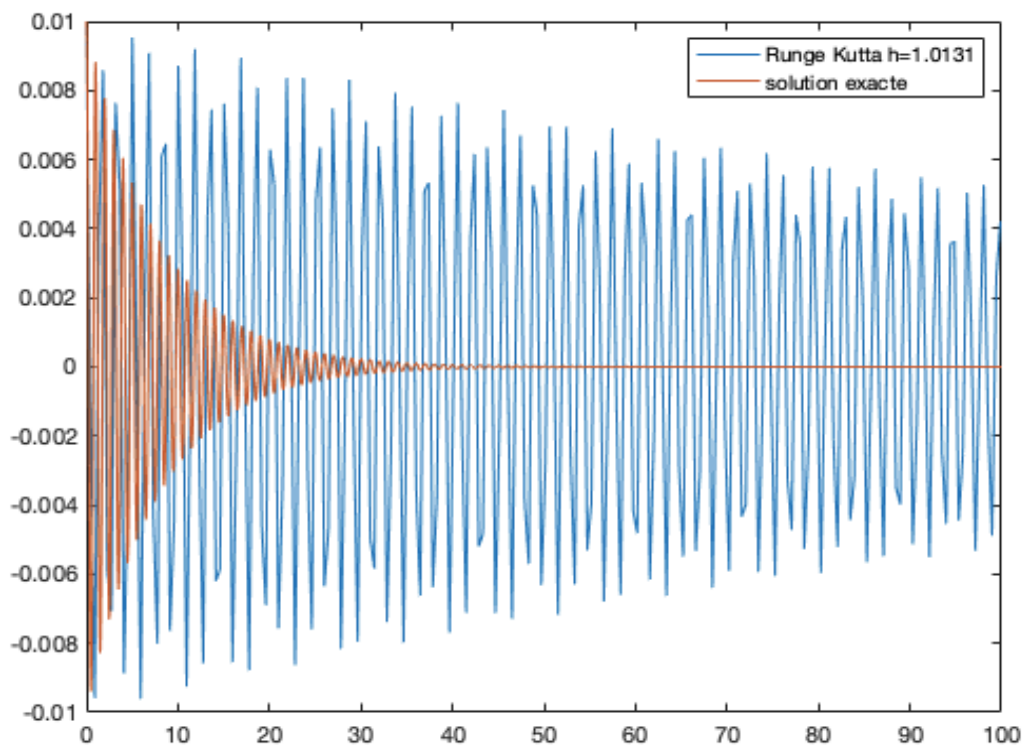


Le résultat n'est pas stable ni précis de la solution exacte.

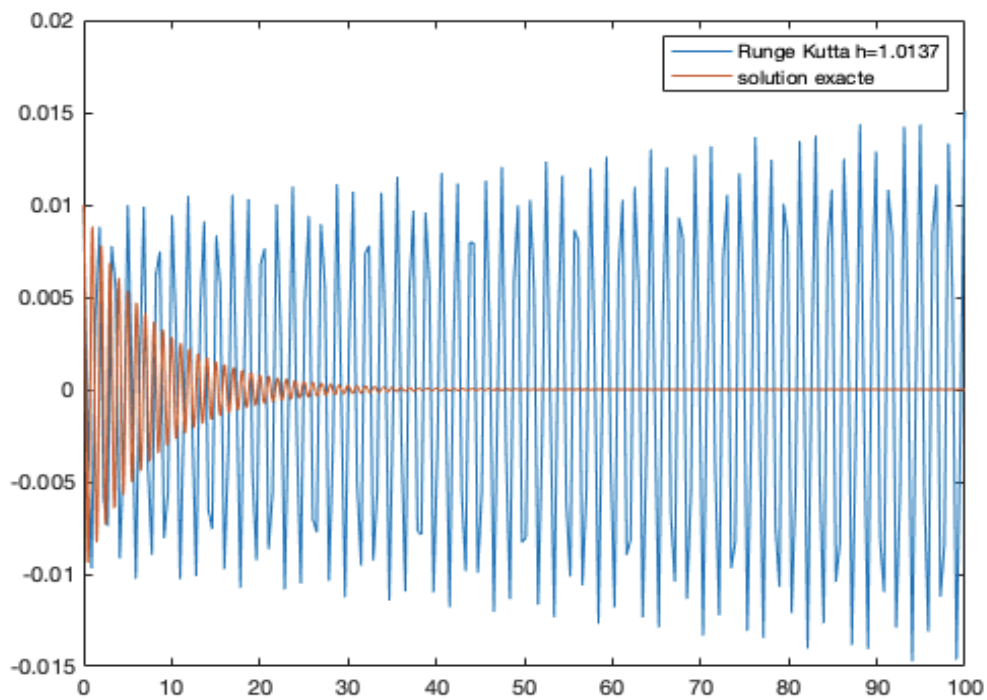
### 1.3.b

On utilise la méthode de dichotomie, et puis on peut confirmer  $h_c$  est dans le intervalle à partir de la question 1.3.a :  $1.0131 < h_c < 1.0137$

```
clf;
%quand h=1.013125
%h = 1.01375
h = 1.013125;
dt_rk = h*2*sqrt(2)/w0;
t_rk = 0:dt_rk:100*T0;
C = [0,1;-w0^2,-2*eps*w0];
Urk(:,1) = [x0;Dx0];
for i = 1:length(t_rk)-1
    k1 = C*Urk(:,i);
    k2 = C*(Urk(:,i)+1/2*dt_rk*k1);
    k3 = C*(Urk(:,i)+1/2*dt_rk*k2);
    k4 = C*(Urk(:,i)+dt_rk*k3);
    K = 1/6*(k1+2*k2+2*k3+k4);
    Urk(:,i+1) = Urk(:,i)+K*dt_rk;
end
%comparer les figures
plot(t_rk,Urk(1,:))
hold on;
plot(t,exp(-eps*w0*t).*(x0*cos(omega*t)+(eps*w0*x0+Dx0)/omega.*sin(omega*t)));
legend('Runge Kutta h=1.0131','solution exacte');
```



Quand  $h = 1.0131$ , la solution du schéma Runge-Kutta est amortie et stable.



En même temps, si  $h = 1.0137$ , la solution du schéma Runge-Kutta n'est

plus stable. Donc on peut conclure:  $dt = h_c \times \frac{2\sqrt{2}}{\omega_0}$  où

$$1.0131 < h_c < 1.0137.$$

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

## 1. Newmark explicite, $\gamma=0.5$ $\beta=0$

1.1

La matrice d'amplification est :

$$\begin{aligned}
 q_{n+1} &= q_n + \Delta t \dot{q}_n + \Delta t^2 (0.5 - \beta) \ddot{q}_n + \Delta t^2 \beta \ddot{q}_{n+1} \\
 \dot{q}_{n+1} &= \dot{q}_n + \Delta t (1 - \gamma) \ddot{q}_n + \Delta t \gamma \ddot{q}_{n+1} \\
 \text{et } \ddot{q}_{n+1} &= -\omega_0^2 q_{n+1}
 \end{aligned}$$

$$\Rightarrow \begin{bmatrix} 1 + \beta \Delta t^2 \omega_0^2 & 0 \\ \gamma \Delta t \omega_0^2 & 1 \end{bmatrix} \begin{bmatrix} q_{n+1} \\ \dot{q}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 - \Delta t^2 (0.5 - \beta) \omega_0^2 & \Delta t \\ -(1 - \gamma) \Delta t \omega_0^2 & 1 \end{bmatrix} \begin{bmatrix} q_n \\ \dot{q}_n \end{bmatrix}$$

$$[B] \begin{bmatrix} q_{n+1} \\ \dot{q}_{n+1} \end{bmatrix} = [C] \begin{bmatrix} q_n \\ \dot{q}_n \end{bmatrix}$$

la matrice d'amplification  $A = [B]^{-1} \times [C]$

$$A = \begin{bmatrix} 1 - \frac{\omega_0^2 \Delta t^2}{2(1 + \beta \omega_0^2 \Delta t^2)} & \frac{\Delta t}{1 + \beta \omega_0^2 \Delta t^2} \\ -\omega_0^2 \Delta t \left[ \frac{\gamma \omega_0^2 \Delta t^2}{2(1 + \beta \omega_0^2 \Delta t^2)} + 1 \right] & 1 - \frac{\gamma \omega_0^2 \Delta t^2}{1 + \beta \omega_0^2 \Delta t^2} \end{bmatrix}$$

## 1.2

On va déterminer le pas de temps critique à partir de cette matrice

d'amplification. Comme on a étudié pendant le cours, quand gamma

egale 0.5, il faut que  $(\gamma + \frac{1}{2})^2 - 4\beta \leq \frac{4}{\omega_0^2 \Delta t^2}$ . On a déjà  $\gamma = 0.5$  et

$\beta = 0$ , donc le pas de temps critique est :

```
clear all;
m = 2;
a = 0.5;
g = 9.81;
F0 = 20;
%calculer w0
w0 = sqrt(g/a);
w = 2*pi;
theta10 = 0;
theta20 = 0;
Dtheta10 = -1.31519275;
Dtheta20 = -1.85996342;
gamma1 = 0.5;
beta1 = 0;
```

```
Tc = sqrt(4/(w0^2*((gamma1+1/2)^2-4*beta1)))
```

```
Tc =
```

```
0.4515
```

1.3

$$ma^2 \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \ddot{q}_0 + mga \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} q_0 = 0$$

1.4

En présence d'une force dépendant du temps, on a :

$$q_{n+1} = q_n + \Delta t \dot{q}_n + \Delta t^2 (0.5 - \beta) \ddot{q}_n + \Delta t^2 \beta \ddot{q}_{n+1}$$

$$\dot{q}_{n+1} = \dot{q}_n + \Delta t (1 - \gamma) \ddot{q}_n + \Delta t \gamma \ddot{q}_{n+1}$$

$$ma^2 \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \ddot{q}_n + mga \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} q_n = F_0 \sin \omega t \begin{vmatrix} a \\ \frac{a}{\sqrt{2}} \end{vmatrix}$$

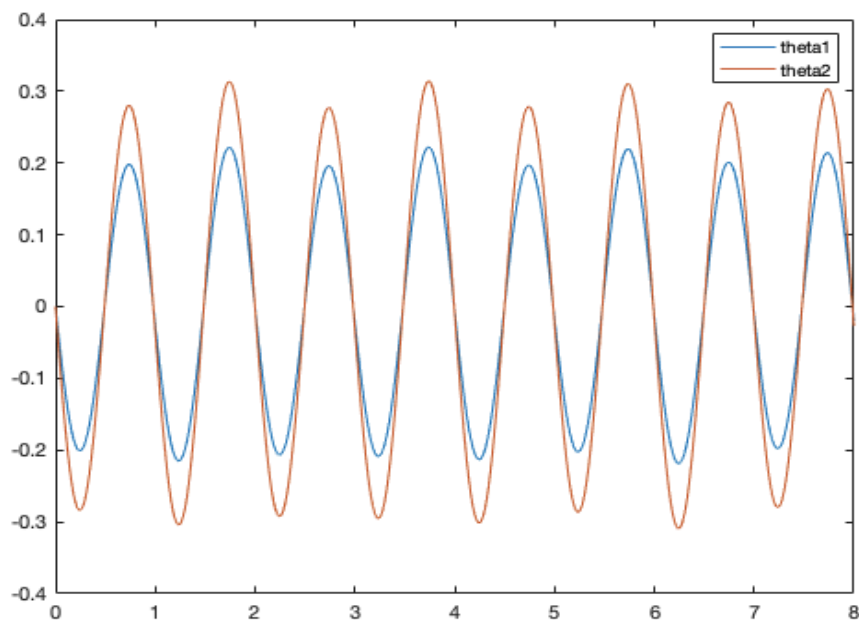
## 1.5

```
%D2q+omega0*omega0*q = f*sin(w*t)
omega02 = inv([2,1;1,1])*m*g*a*[2,0;0,1]/(m*a^2);
f = F0*inv([2,1;1,1])*[a;a/sqrt(2)]/(m*a^2);

dt = 0.02;
t = 0:dt:T0;
B = [[1,0;0,1]+beta1*dt^2*omega02,0*[1,0;0,1];gamma1*dt*omega02,[1,0;0,1]];
C = [[1,0;0,1)-(0.5-beta1)*dt^2*omega02,dt*[1,0;0,1];-(1-gamma1)*dt*omega02,[1,0;0,1]];
A = inv(B)*C;
%calculer les valeurs propres
vp = eig(A);
re = [real(vp(2));real(vp(4))];
im = [imag(vp(2));imag(vp(4))];
mo = [abs(vp(2));abs(vp(4))];

U(:,1) = [q0;Dq0];
for i = 1:length(t)-1
    F = [dt^2*(0.5-beta1)*f*sin(w*i*dt);dt*(1-gamma1)*f*sin(w*i*dt)];
    U(:,i+1) = A*U(:,i)+inv(B)*F;
end
plot(t,U(1,:));
hold on;
plot(t,U(2,:));
legend('theta1','theta2')
```

Le résultat est comme ci-dessous :



## 1.6

Pour la valeur de t egale à 0s :

```
%quand t = 0s  
q = U(1:2,1)  
Dq = U(3:4,1)  
D2q = f*sin(w*25*dt)-omega02*q
```

```
q =  
  
    0  
    0  
  
Dq =  
  
-1.3152  
-1.8600  
  
D2q =  
  
1.0e-14 *  
  
0.0717  
0.1015
```

Pour la valeur de t egale à dt :

```
%quand t = dt  
q = U(1:2,2)  
Dq = U(3:4,2)  
D2q = f*sin(w*25*dt)-omega02*q
```

```
q =  
  
-0.0262  
-0.0370  
  
Dq =  
  
-1.2975  
-1.8349  
  
D2q =  
  
0.3006  
0.4251
```



Pour la valeur de t egale à 2\*dt :

```
%quand t = 2*dt  
q = U(1:2,3)  
Dq = U(3:4,3)  
D2q = f*sin(w*25*dt)-omega02*q
```

```
q =  
  
-0.0518  
-0.0732
```

```
Dq =  
  
-1.2594  
-1.7811
```

```
D2q =  
  
0.5948  
0.8412
```

Pour la valeur de t egale à 0.5s :

```
%quand t = 0.5s  
q = U(1:2,26)  
Dq = U(3:4,26)  
D2q = f*sin(w*25*dt)-omega02*q
```

```
q =  
  
0.0108  
0.0152
```

```
Dq =  
  
1.2688  
1.7943
```

```
D2q =  
  
-0.1237  
-0.1750
```

## 2. Newmark implicite, gamma=0.5 beta=0.25

2.1

La matrice d'amplification est :

$$\begin{aligned}q_{n+1} &= q_n + \Delta t \dot{q}_n + \Delta t^2 (0.5 - \beta) \ddot{q}_n + \Delta t^2 \beta \ddot{q}_{n+1} \\ \dot{q}_{n+1} &= \dot{q}_n + \Delta t (1 - \gamma) \ddot{q}_n + \Delta t \gamma \ddot{q}_{n+1} \\ \text{et } \ddot{q}_{n+1} &= -\omega_0^2 q_{n+1}\end{aligned}$$

$$\Rightarrow \begin{bmatrix} 1 + \beta \Delta t^2 \omega_0^2 & 0 \\ \gamma \Delta t \omega_0^2 & 1 \end{bmatrix} \begin{bmatrix} q_{n+1} \\ \dot{q}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 - \Delta t^2 (0.5 - \beta) \omega_0^2 & \Delta t \\ -(1 - \gamma) \Delta t \omega_0^2 & 1 \end{bmatrix} \begin{bmatrix} q_n \\ \dot{q}_n \end{bmatrix}$$
$$[B] \begin{bmatrix} q_{n+1} \\ \dot{q}_{n+1} \end{bmatrix} = [C] \begin{bmatrix} q_n \\ \dot{q}_n \end{bmatrix}$$

la matrice d'amplification  $A = [B]^{-1} \times [C]$

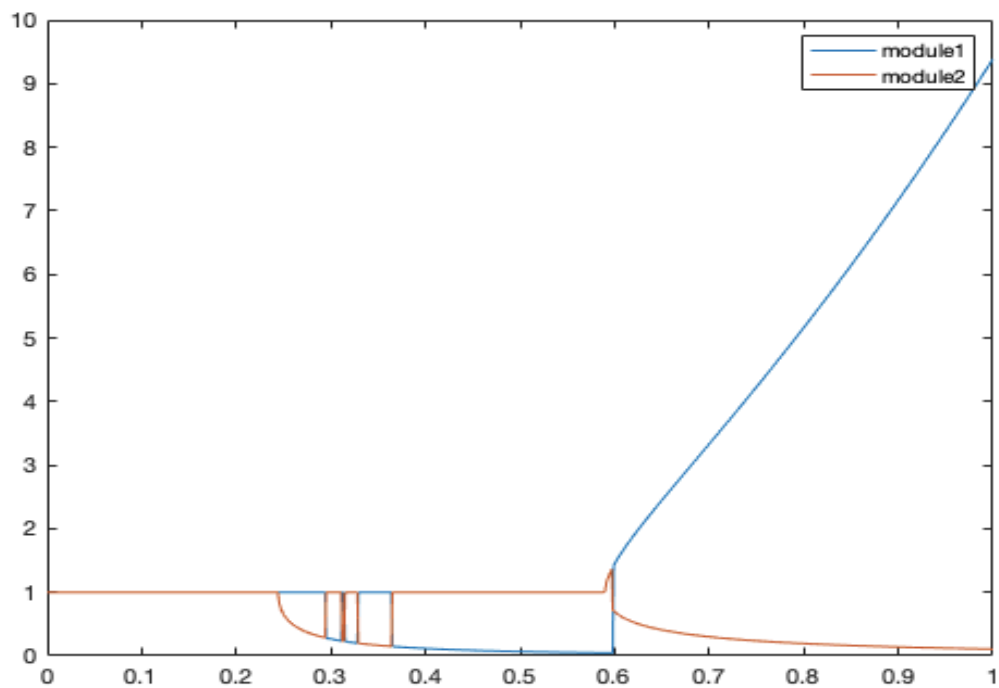
$$A = \begin{bmatrix} 1 - \frac{\omega_0^2 \Delta t^2}{2(1 + \beta \omega_0^2 \Delta t^2)} & \frac{\Delta t}{1 + \beta \omega_0^2 \Delta t^2} \\ -\omega_0^2 \Delta t \left[ \frac{\gamma \omega_0^2 \Delta t^2}{2(1 + \beta \omega_0^2 \Delta t^2)} + 1 \right] & 1 - \frac{\gamma \omega_0^2 \Delta t^2}{1 + \beta \omega_0^2 \Delta t^2} \end{bmatrix}$$

## 2.2

```
clf;
dt2 = 0:0.001:1;
for j = 1:length(dt2)
    B2(1,j) = [[1,0;0,1]+beta1*dt2(j)^2*omega02,0*[1,0;0,1];gamma1*dt2(j)*omega02,[1,0;0,1]];
    C2(1,j) = [[1,0;0,1]-(0.5-beta1)*dt2(j)^2*omega02,dt2(j)*[1,0;0,1];-(1-gamma1)*dt2(j)*omega02,[1,0;0,1]];
    A2(1,j) = inv(B2(1,j))*C2(1,j);
    %calculer les valeurs propres
    vp2(:,j) = eig(A2(1,j));
    re2(:,j) = [real(vp2(2,j));real(vp2(4,j))];
    im2(:,j) = [imag(vp2(2,j));imag(vp2(4,j))];

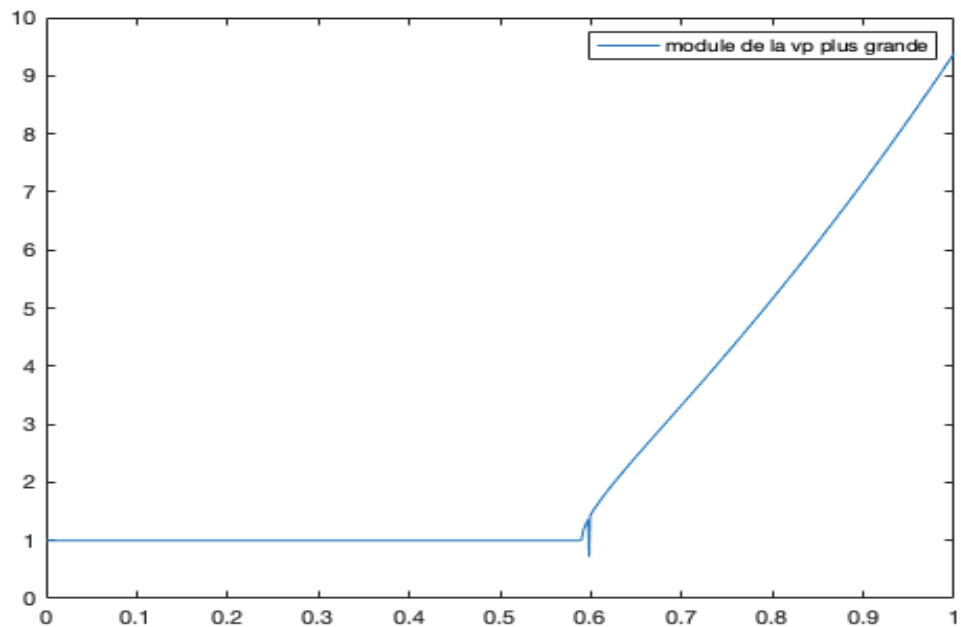
    mo2(:,j) = [abs(vp2(2,j));abs(vp2(4,j))];
end
plot(dt2,mo2(1,:));
hold on;
plot(dt2,mo2(2,:));
legend('module1','module2');
%legend('module de la vp plus grande');
```

On présente les deux modules dans un même graphique :



Et puis on choisit le plus grande valeur propre :

```
clf;
dt2 = 0:0.001:1;
for j = 1:length(dt2)
    B2(1,j) = [[1,0;0,1]+beta1*dt2(j)^2*omega02,0*[1,0;0,1];gamma1*dt2(j)*omega02,[1,0;0,1]];
    C2(1,j) = [[1,0;0,1)-(0.5-beta1)*dt2(j)^2*omega02,dt2(j)*[1,0;0,1];-(1-gamma1)*dt2(j)*omega02,[1,0;0,1]];
    A2(1,j) = inv(B2(1,j))*C2(1,j);
    %calculer les valeurs propres
    vp2(:,j) = eig(A2(1,j));
    re2(:,j) = [real(vp2(2,j));real(vp2(4,j))];
    im2(:,j) = [imag(vp2(2,j));imag(vp2(4,j))];
    %choisir la plus grande valeur propre
    mo2(j) = max(abs(vp2(2,j)),abs(vp2(4,j)));
end
plot(dt2,mo2);
legend('module de la vp plus grande');
```



2.3

$$ma^2 \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \ddot{q}_0 + mga \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} q_0 = 0$$

2.4

En présence d'une force dépendant du temps, on a :

$$q_{n+1} = q_n + \Delta t \dot{q}_n + \Delta t^2 (0.5 - \beta) \ddot{q}_n + \Delta t^2 \beta \ddot{q}_{n+1}$$

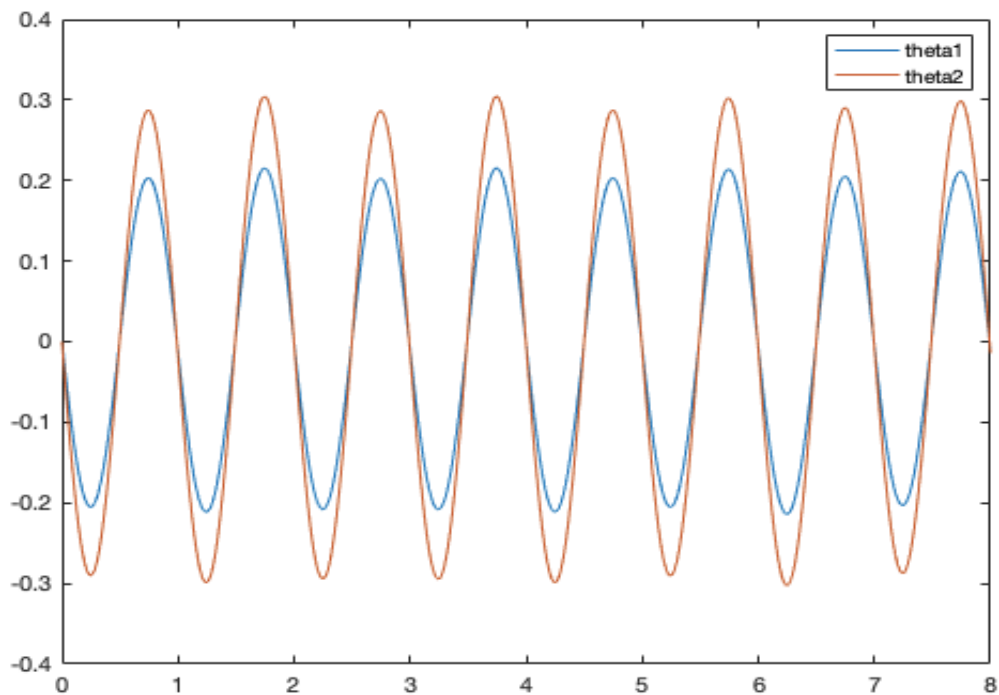
$$\dot{q}_{n+1} = \dot{q}_n + \Delta t (1 - \gamma) \ddot{q}_n + \Delta t \gamma \ddot{q}_{n+1}$$

$$ma^2 \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \ddot{q}_n + mga \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} q_n = F_0 \sin \omega t \begin{vmatrix} a \\ \frac{a}{\sqrt{2}} \end{vmatrix}$$

## 2.5

```
%D2q+omega0*omega0*q = f*sin(w*t)
omega02 = inv([2,1;1,1])*m*g*a*[2,0;0,1]/(m*a^2);
f = F0*inv([2,1;1,1])*[a;a/sqrt(2)]/(m*a^2);

dt = 0.02;
t = 0:dt:T0;
B = [[1,0;0,1]+beta2*dt^2*omega02,0*[1,0;0,1];gamma2*dt*omega02,[1,0;0,1]];
C = [[1,0;0,1]-(0.5-beta2)*dt^2*omega02,dt*[1,0;0,1];-(1-gamma2)*dt*omega02,[1,0;0,1]];
A = inv(B)*C;
U(:,1) = [q0;Dq0];
for i = 1:length(t)-1
    F = [dt^2*(0.5-beta2)*f*sin(w*i*dt);dt*(1-gamma2)*f*sin(w*i*dt)];
    U(:,i+1) = A*U(:,i)+inv(B)*F;
end
plot(t,U(1,:));
hold on;
plot(t,U(2,:));
legend('theta1','theta2');
```



## 2.6

Pour la valeur de t egale à 0s :

```
%quand t = 0s  
q = U(1:2,1)  
Dq = U(3:4,1)  
D2q = f*sin(w*25*dt)-omega02*q
```

q =

```
0  
0
```

Dq =

```
-1.3152  
-1.8600
```

D2q =

```
1.0e-14 *  
0.0717  
0.1015
```

Pour la valeur de t egale à dt :

```
%quand t = dt  
q = U(1:2,2)  
Dq = U(3:4,2)  
D2q = f*sin(w*25*dt)-omega02*q
```

q =

```
-0.0262  
-0.0371
```

Dq =

```
-1.2975  
-1.8349
```

D2q =

```
0.3011  
0.4259
```

Pour la valeur de t egale à 2\*dt :

```
%quand t = 2*dt  
q = U(1:2,3)  
Dq = U(3:4,3)  
D2q = f*sin(w*25*dt)-omega02*q
```

```
q =  
  
-0.0519  
-0.0734
```

```
Dq =  
  
-1.2594  
-1.7810
```

```
D2q =  
  
0.5967  
0.8438
```

Pour la valeur de t egale à 0.5s :

```
%quand t = 0.5s  
q = U(1:2,26)  
Dq = U(3:4,26)  
D2q = f*sin(w*25*dt)-omega02*q
```

```
q =  
  
0.0051  
0.0071
```

```
Dq =  
  
1.2910  
1.8258
```

```
D2q =  
  
-0.0580  
-0.0821
```



# Etude d'un oscillateur non linéaire à un degré de liberté

## 1. Newmark explicite

### 1.1

$$\begin{aligned}q_{j+1} &= q_j + \Delta t \dot{q}_j + \Delta t^2 (\omega_0^2 (0.5 - \beta)) \ddot{q}_j + \beta \Delta t^2 \ddot{q}_{j+1} \\ \dot{q}_{j+1} &= \dot{q}_j + \Delta t (1 - \gamma) \ddot{q}_j + \gamma \Delta t \ddot{q}_{j+1} \\ \ddot{q}_j + \omega_0^2 q_j (1 + a q_j^2) &= 0\end{aligned}$$

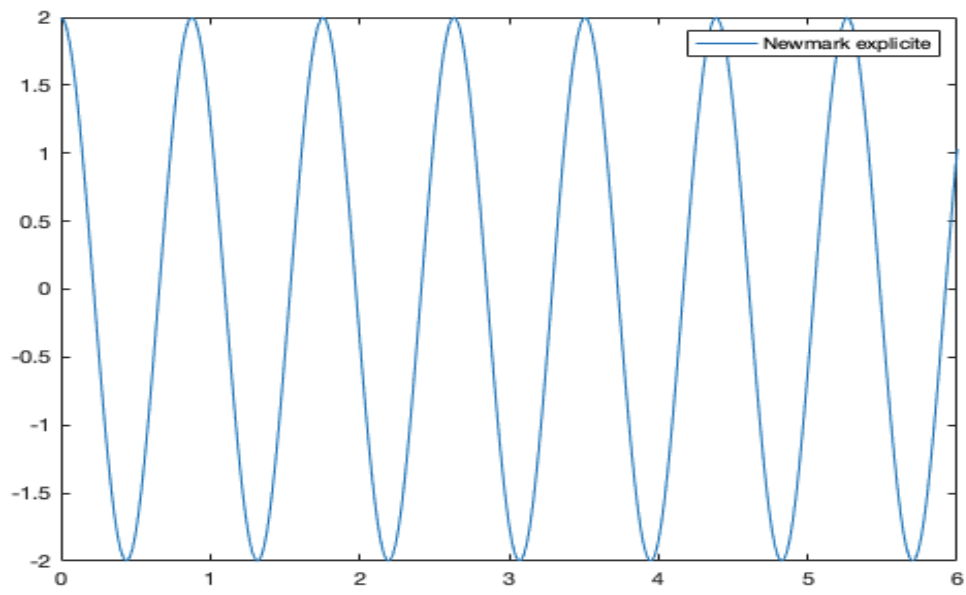
### 1.2

```
w0 = 2*pi;
a = 0.1;
q0 = 2;
Dq0 = 0;
gamma1 = 0.5;
beta1 = 0;
q(1) = q0;
Dq(1) = Dq0;
T0 = 6;
```

### 1.2

```
dt = 0.02;
t = 0:dt:T0;
for i = 1:length(t)-1
    D2q(i) = -w0^2*q(i)*(1+a*q(i)*q(i));
    q(i+1) = q(i)+dt*Dq(i)+dt^2*(0.5-beta1)*D2q(i);
    Dq(i+1) = Dq(i)+dt*(1-gamma1)*D2q(i)+gamma1*dt*(-w0^2*q(i+1)*(1+a*q(i+1)*q(i+1)));
end
plot(t,q);
legend('Newmark explicite');
```

Le résultat est comme ci-dessous :



1.3

### 1.3

```
qchoisit = [q(1),q(2),q(3),q(length(t))]
```

```
qchoisit =
```

```
2.0000    1.9779    1.9123    1.0329
```

## 2. Newmark implicite

2.1

On doit chercher à minimiser la norme du résidu.

2.2

l'expression de la correction  $\bar{a}$   $\ddot{q}_{j+1}^*$  est :

$$\Delta \ddot{q}_{j+1} + \ddot{q}_{j+1}^*$$

avec :

$$\Delta \ddot{q}_{j+1} + \Delta q_{j+1} \times \omega_0^2 \cos q_{j+1}^*$$

$$= - (\ddot{q}_{j+1}^* + \omega_0^2 \sin q_{j+1}^*)$$

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

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

## 2.3

```
gamma2 = 0.5;
beta2 = 0.25;
%on prend la précision égale 0.0001
eps = 0.0001;

dt = 0.02;
t = 0:dt:T0;
q(1) = q0;
Dq(1) = Dq0;
D2q(1) = -w0^2*q(1)*(1+a*q(1)*q(1));

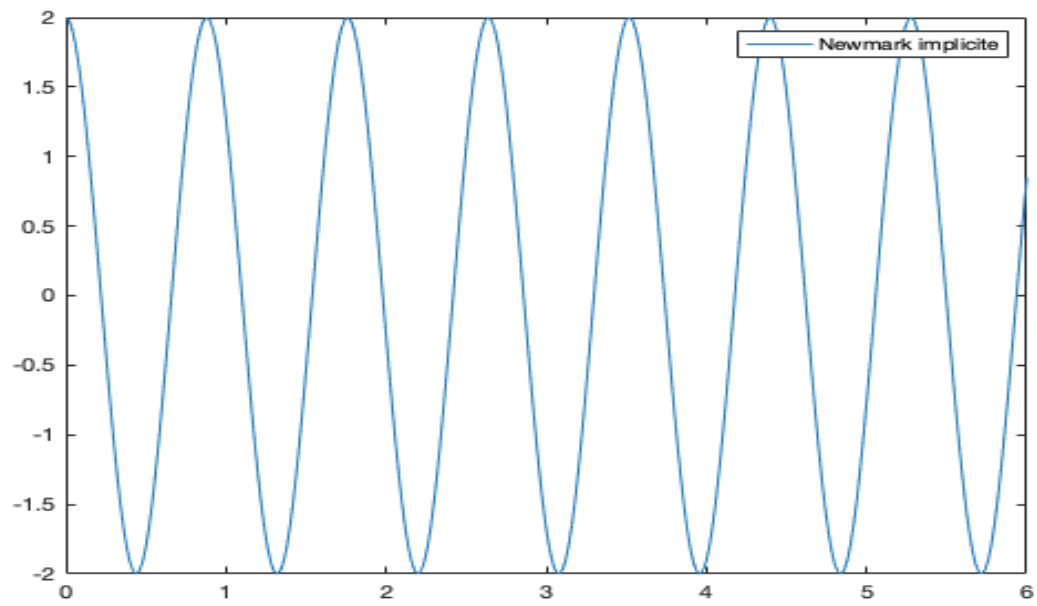
qe(1) = q(1);
Dqe(1) = Dq(1);
D2qe(1) = D2q(1);

for j = 1:length(t)-1
    qe(j+1) = q(j)+dt*Dq(j)+dt^2*(0.5-beta2)*D2q(j);
    Dqe(j+1) = Dq(j)+dt*(1-gamma2)*D2q(j);
    D2qe(j+1) = 0;
    f = D2qe(j+1)+w0^2*qe(j+1)*(1+a*qe(j+1)*qe(j+1));
    %comparer le résidu et la précision
    while abs(f)>=eps
        deltaD2qe = -(D2qe(j+1)+w0^2*qe(j+1)*(1+a*qe(j+1)*qe(j+1)))/(1+w0^2*beta2*dt^2*(1+3*a*qe(j+1)*qe(j+1)));
        deltaqe = beta2*dt^2*deltaD2qe;
        deltaDqe = gamma2*dt*deltaD2qe;

        %les corrections
        qe(j+1) = qe(j+1)+deltaqe;
        Dqe(j+1) = Dqe(j+1)+deltaDqe;
        D2qe(j+1) = D2qe(j+1)+deltaD2qe;
        f = D2qe(j+1)+w0^2*qe(j+1)*(1+a*qe(j+1)*qe(j+1));
    end
    q(j+1) = qe(j+1);
    Dq(j+1) = Dqe(j+1);
    D2q(j+1) = D2qe(j+1);
end

clf;
plot(t,q)
legend('Newmark implicite');
```

Et le résultat est comme ci-dessous :



2.4

## 2.4

```
qchoisit = [q(1),q(2),q(3),q(length(t))]
```

```
qchoisit =
```

```
2.0000    1.9781    1.9131    0.8485
```

### 3. Energie mécanique

#### 3.1

L'énergie mécanique pour cet oscillateur non linéaire est définie comme la somme de l'énergie cinétique et l'énergie potentiel.

L'énergie cinétique:  $0.5*m*v^2$

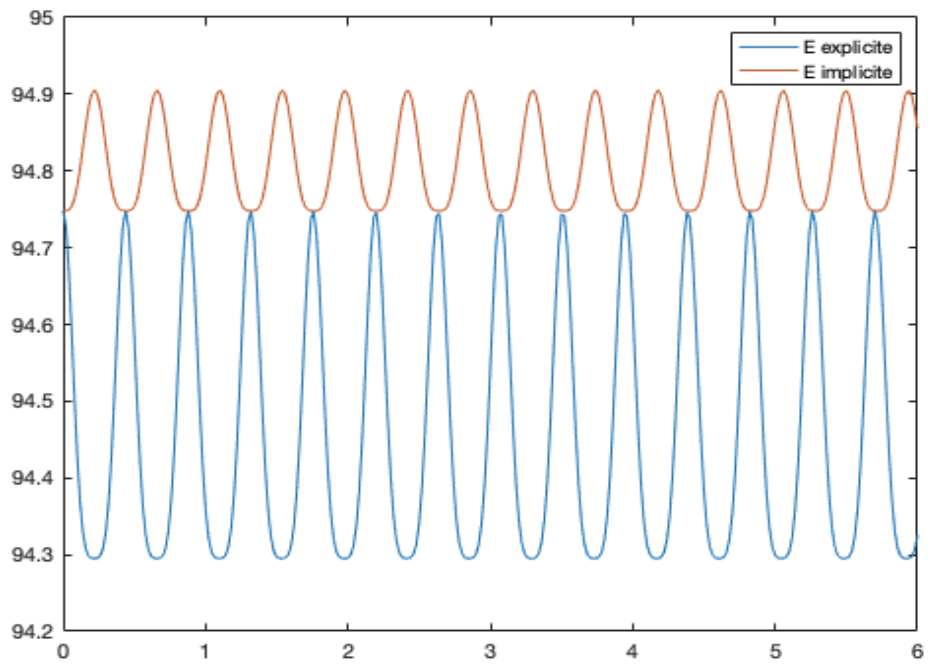
L'énergie potentiel:  $0.5*k*q^2+0.25*k*a*q^4$

#### 3.2

```
for i = 1:length(t)
    %pour le schéma explicite
    Ecex(i) = 0.5*Dq1(i)^2;
    Epex(i) = 0.5*w0^2*q1(i)^2+0.25*w0^2*a*q1(i)^4;
    Eex(i) = Ecex(i)+Epex(i);

    %pour le schéma implicite
    Ecim(i) = 0.5*Dq(i)^2;
    Epim(i) = 0.5*w0^2*q(i)^2+0.25*w0^2*a*q(i)^4;
    Eim(i) = Ecim(i)+Epim(i);
end
clf;
plot(t,Eex)
hold on;
plot(t,Eim)
legend('E explicite','E implicite')
```

La figure des deux energies sont comme ci-dessous :



### 3.3

On peut observer que l'énergie calculé par Newmark implicite est plus stable que cela par Newmark explicite .