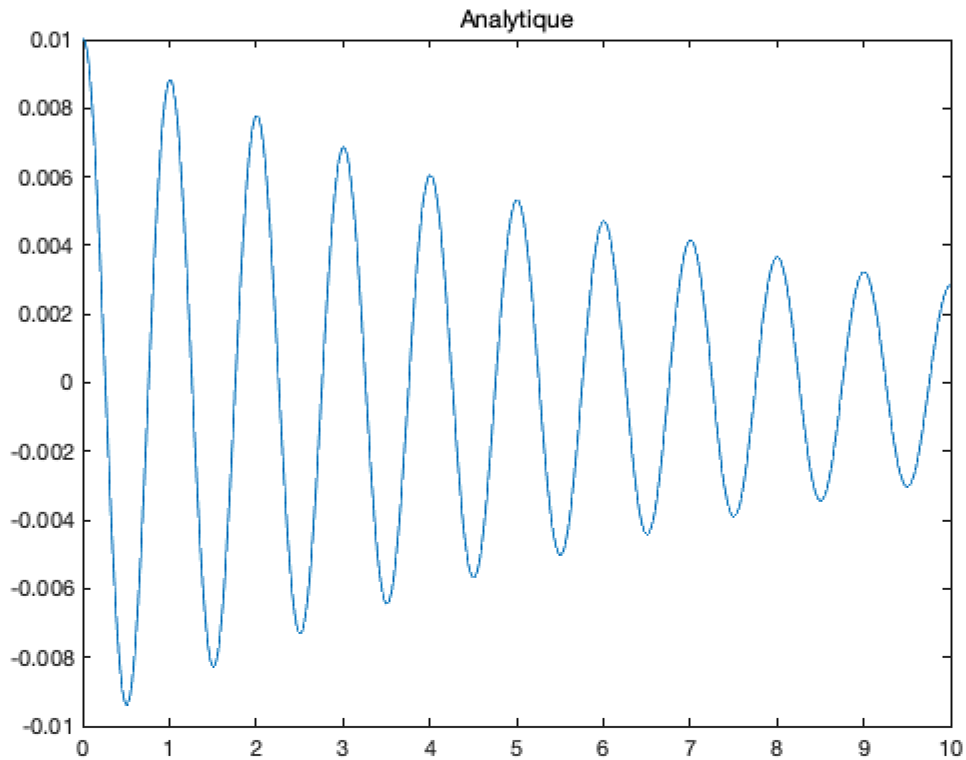
Etude d’un oscillateur lineaire amorti a un degre de liberte

Table of Contents

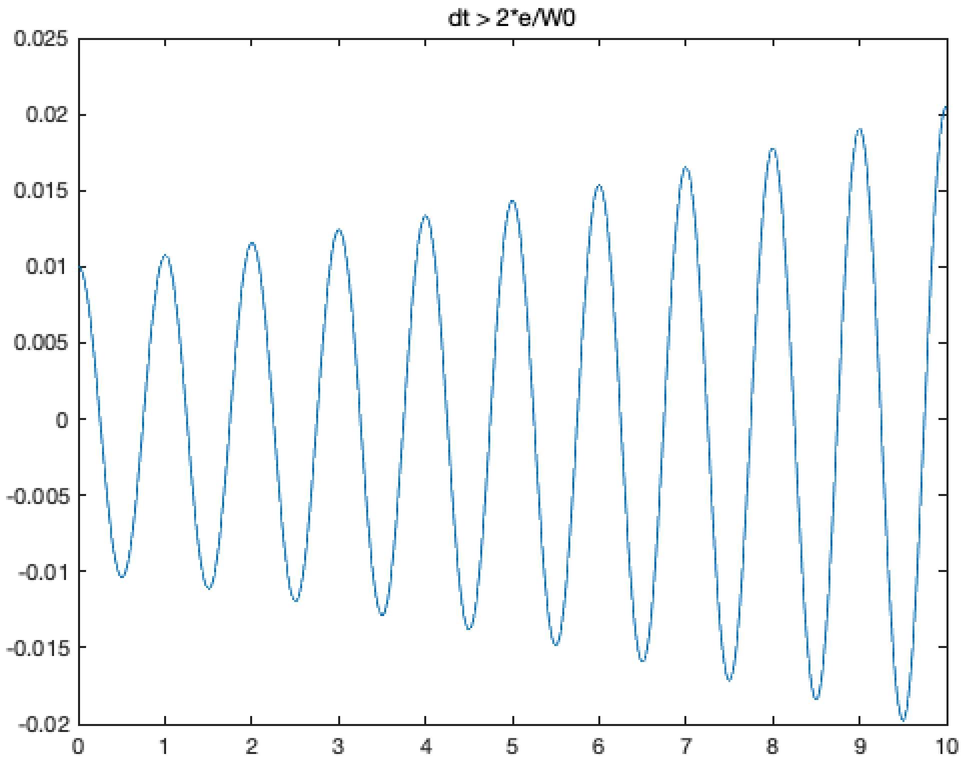
Préparation

```
T0 = 1;
W0 = 2*pi/T0;
e = 0.02;
m = 1;
b = 2*e*W0*m;
X0 = 0.01;
dX0 = 0;
Ft = 0;
omiga = W0*(1-e^2)^0.5;
x = [];
x(1) = X0;
n = 1;
for t = 0:0.01:10*T0
    n = n + 1;
    x(n) = exp(-e*W0*t)*(X0*cos(omiga*t) + (e*W0*X0 + dX0)/
omiga*sin(omiga*t));
end
t = linspace(0,10*T0,n);
plot(t,x);
title('Analytique');
```



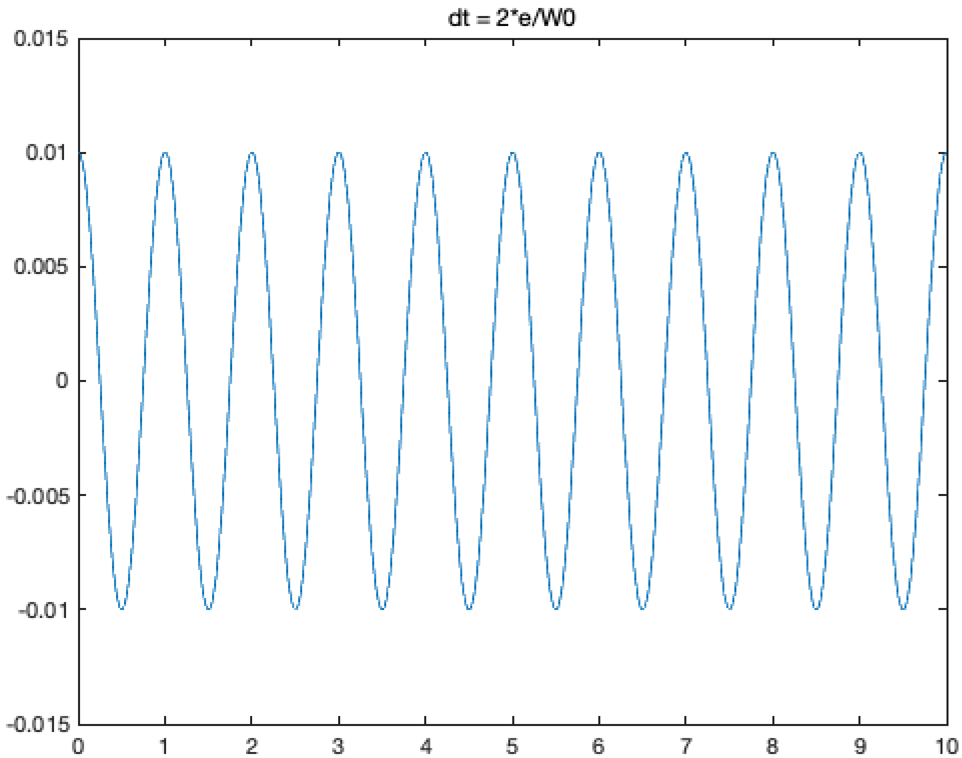
Q1.1a

```
t_1 = 2*e/W0;%0.0064
dt = 0.01;%>t_1
A = [1,dt;-dt*W0^2,1-2*dt*e*W0];
X = [X0;dX0];
x_ex = [];
dx_ex = [];
x_ex(1) = X0;
dx_ex(1) = dX0;
n = 1;
for t = 0:dt:10*T0
    n = n + 1;
    X = A*X;
    x_ex(n) = X(1,1);
    dx_ex(n) = X(2,1);
end
t = linspace(0,10*T0,n);
plot(t,x_ex);
title('dt > 2*e/W0');
%on peut voir que x diverge;
```



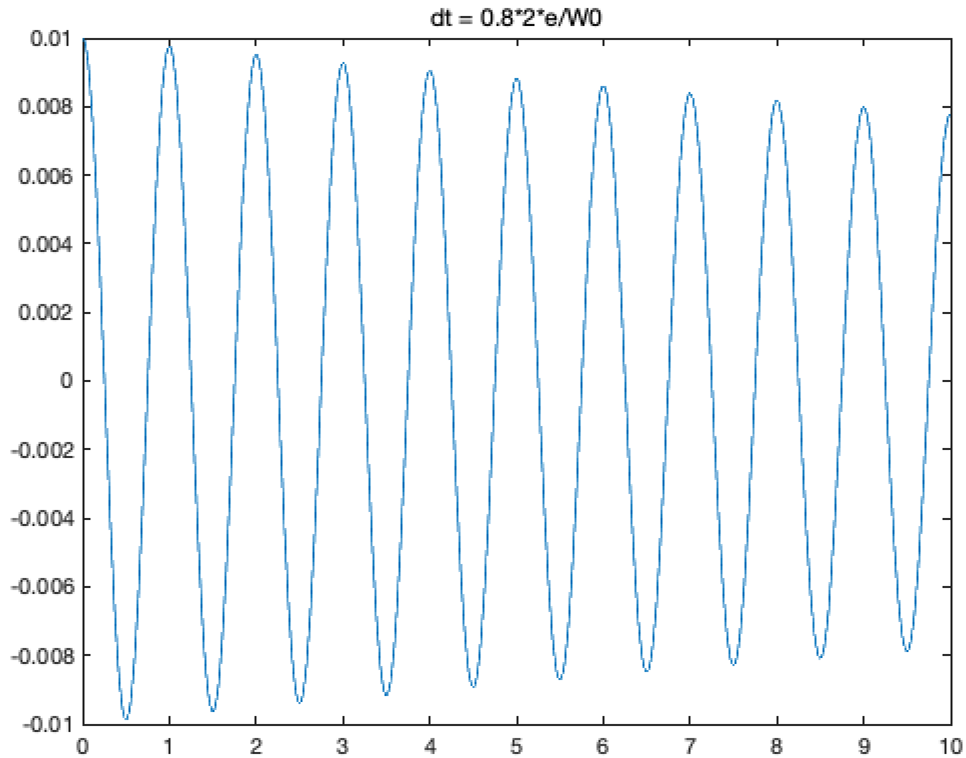
Q1.1b

```
dt = t_1;
A = [1,dt;-dt*W0^2,1-2*dt*e*W0];
X = [X0;dX0];
x_ex = [];
dx_ex = [];
x_ex(1) = X0;
dx_ex(1) = dX0;
n = 1;
for t = 0:dt:10*T0
    n = n + 1;
    X = A*X;
    x_ex(n) = X(1,1);
    dx_ex(n) = X(2,1);
end
t = linspace(0,10*T0,n);
plot(t,x_ex);
title('dt = 2*e/W0');
% x est sinusoidale, ni converge ni diverge.
```



Q1.1c

```
dt = 0.8*t_1;
A = [1,dt;-dt*W0^2,1-2*dt*e*W0];
X = [X0;dX0];
x_ex = [];
dx_ex = [];
x_ex(1) = X0;
dx_ex(1) = dX0;
n = 1;
for t = 0:dt:10*T0
    n = n + 1;
    X = A*X;
    x_ex(n) = X(1,1);
    dx_ex(n) = X(2,1);
end
t = linspace(0,10*T0,n);
plot(t,x_ex);
title('dt = 0.8*2*e/W0');
% x converge.
```

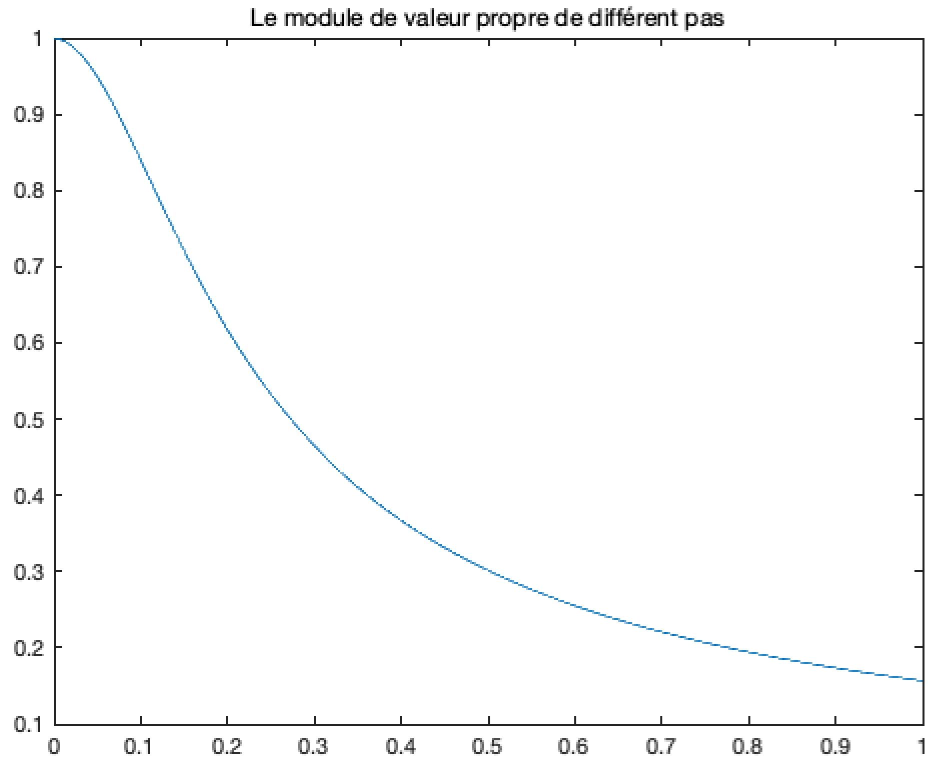


Q1.1d

```
%le rapport de dt/(2*e/W0) est un critere de la solution, et le
 rapport
%doit etre plus petit que 1 pour que la solution soit precis
```

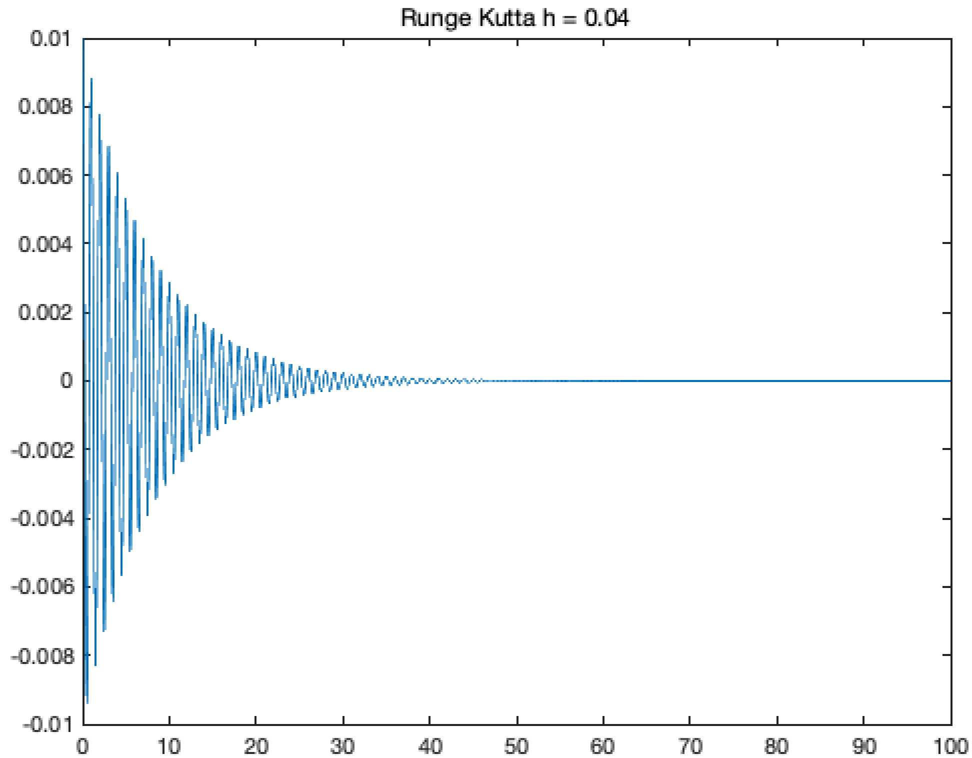
Q1.2

```
syms pas;
A_im = [1+2*pas*e*W0,pas;-pas*W0^2,1]/(1 + 2*pas*e*W0 + pas^2*W0^2);

vp = [];
for pas = linspace(0, 1, 1001)
vp = [vp, max(abs(eig(eval(A_im))))];
end

t = linspace(0, 1, 1001);
plot(t,vp);
title('Le module de valeur propre de différent pas');
%x converge, si le pas est plus petit, x converge moins vite, et le
 pas augment, x converge plus vite.
```



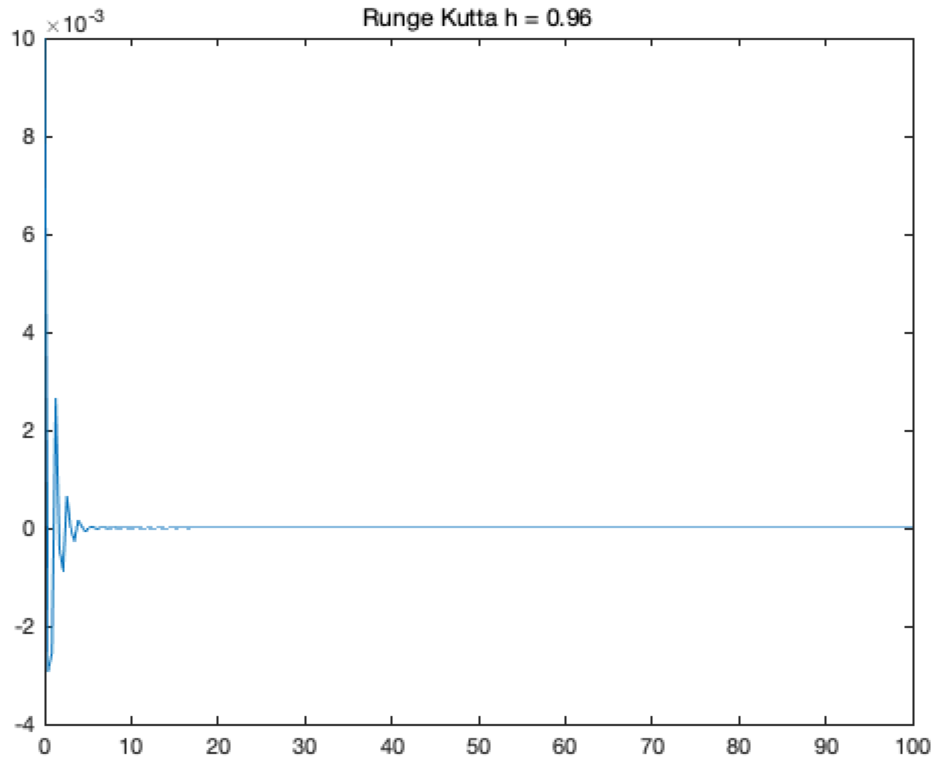
Q1.3a h = 0.04

```
h = 0.04;
dt = h*2*2^0.5/W0;
M = [0,1;-W0^2,-2*e*W0];
X = [X0;dX0];
x_rg = [];
dx_rg = [];
x_rg(1) = X0;
dx_rg(1) = dX0;
n = 1;
for t = 0:dt:100*T0
    n = n + 1;
    k1 = M * X;
    k2 = M * (X + k1 * dt/2);
    k3 = M * (X + k2 * dt/2);
    k4 = M * (X + k3 * dt);
    K = (k1 + 2*k2 + 2*k3 + k4)/6;
    X = X + K *dt;
    x_rg(n) = X(1,1);
    dx_rg(n) = X(2,1);
end
t = linspace(0,100*T0,n);
plot(t,x_rg);
title('Runge Kutta h = 0.04')
```



Q1.3a h = 0.96

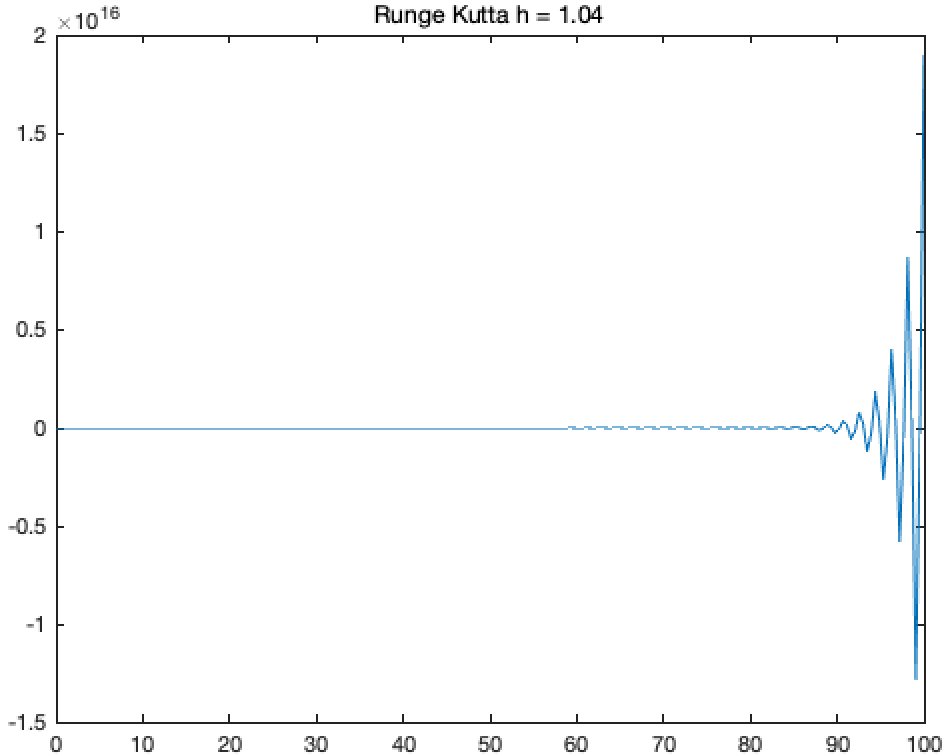
```
h = 0.96;
dt = h*2*2^0.5/W0;
M = [0,1;-W0^2,-2*e*W0];
X = [X0;dX0];
x_rg = [];
dx_rg = [];
x_rg(1) = X0;
dx_rg(1) = dX0;
n = 1;
for t = 0:dt:100*T0
    n = n + 1;
    k1 = M * X;
    k2 = M * (X + k1 * dt/2);
    k3 = M * (X + k2 * dt/2);
    k4 = M * (X + k3 * dt);
    K = (k1 + 2*k2 + 2*k3 + k4)/6;
    X = X + K *dt;
    x_rg(n) = X(1,1);
    dx_rg(n) = X(2,1);
end
t = linspace(0,100*T0,n);
plot(t,x_rg);
title('Runge Kutta h = 0.96')
```



Q1.3a h =1.04

```
h = 1.04;
dt = h*2*2^0.5/W0;
M = [0,1;-W0^2,-2*e*W0];
X = [X0;dX0];
x_rg = [];
dx_rg = [];
x_rg(1) = X0;
dx_rg(1) = dX0;
n = 1;
for t = 0:dt:100*T0
    n = n + 1;
    k1 = M * X;
    k2 = M * (X + k1 * dt/2);
    k3 = M * (X + k2 * dt/2);
    k4 = M * (X + k3 * dt);
    K = (k1 + 2*k2 + 2*k3 + k4)/6;
    X = X + K *dt;
    x_rg(n) = X(1,1);
    dx_rg(n) = X(2,1);
end
t = linspace(0,100*T0,n);
plot(t,x_rg);
title('Runge Kutta h = 1.04');
```

```
%la stabilite de x depend de h, h augment, x est moins stable.
% h depasse un valeur critique, x diverge.
```



Q1.3b

```
hc = 1.0135;
%hmax = 1.0138 diverge un peu
%hmin = 1.0132 converge un peu
tc = hc * 2*2^0.5/W0; % tc = 0.4562
```

Published with MATLAB® R2018a

Etude d’un double pendule avec l’hypothèse des petits mouvements

Table of Contents

Q1.1

```
syms m;
syms a;
syms g;
syms F0;
syms w;
syms beta;
syms gamma;
syms dt;
syms n;
I = [1, 0; 0, 1];

% On a m * a * a * M1 * d2q + m * g * a * M2 * q = F0 * sin(w * t) *
 M3 avec
M1 = [2, 1; 1, 1];
M2 = [2, 0; 0, 1];
M3 = [a; a / sqrt(2)];
% q = [theta1; theta2] et d2q = [d2theta1; d2theta2]
% Alors, on peut trouver d2q = M4 * q + M5 * sin(w * t) avec
M4 = - inv(M1) * g / a * M2;
M5 = inv(M1) * F0 / m / a / a * M3;
% En utilisant les relation (2) et (3), on a
% M6 * qn1 = M7 * qn + M8 * dqn + M9 avec
M6 = I - dt * dt * beta * M4;
M7 = I + dt * dt * (0.5 - beta) * M4;
M8 = I * dt;
M9 = dt * dt * (0.5 - beta) * M5 * sin(w * n * dt) + dt * dt * beta *
 M5 * sin(w * (n + 1) * dt);
% Et M10 * qn1 + M11 * dqn1 = M12 * qn + M13 * dqn + M14 avec
M10 = - dt * gamma * M4;
```

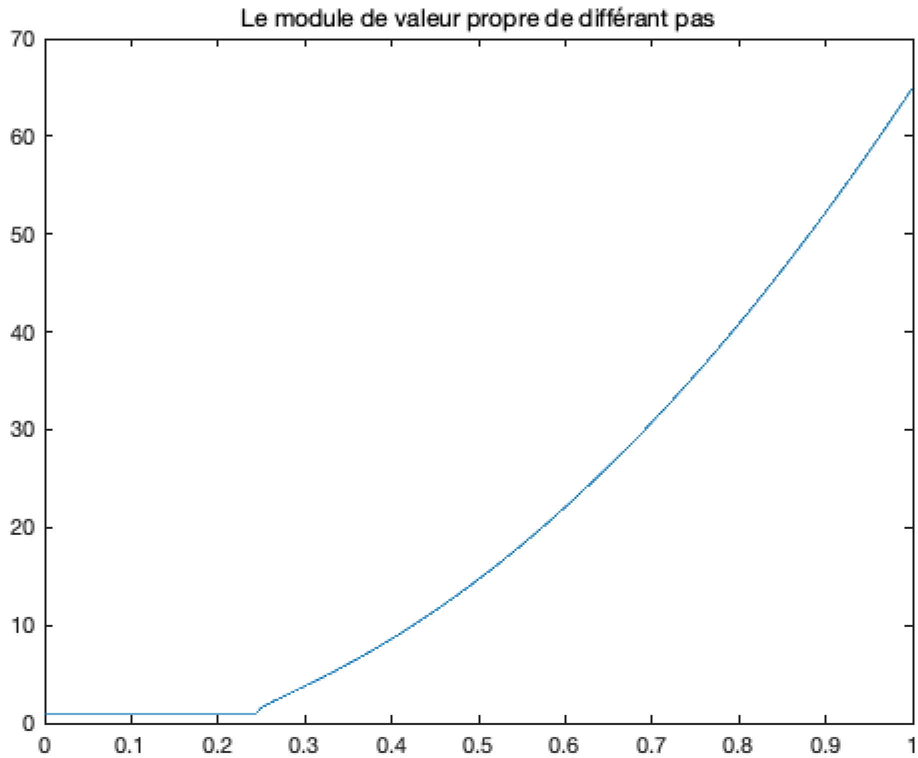
```
M11 = I;
M12 = dt * (1 - gamma) * M4;
M13 = I;
M14 = dt * (1 - gamma) * M5 * sin(w * n * dt) + dt * gamma * M5 *
 sin(w * (n + 1) * dt);
% Soit U = [q; dq], alors on peut trouver M15 * Un1 = M16 * Un + M17
 avec
M15 = [M6, 0 * I; M10, M11];
M16 = [M7, M8; M12, M13];
M17 = [M9; M14];
% Alors, on a Un1 = A * Un + B avec
A = inv(M15) * M16;
B = inv(M15) * M17;
% Alors on peut recevoir le résultat par matlab:
% A =
% [                                                   (((2*g*(beta
 - 1/2)*dt^2)/a + 1)*(a^2 + 2*beta*g*a*dt^2))/(a^2 + 4*a*beta*dt^2*g
 + 2*beta^2*dt^4*g^2) - (2*beta*dt^4*g^2*(beta - 1/2))/(a^2 +
 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2),
              (a*beta*dt^2*g*((2*g*(beta - 1/2)*dt^2)/a + 1))/
(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2) - (dt^2*g*(a^2 +
 2*beta*g*a*dt^2)*(beta - 1/2))/(a*(a^2 + 4*a*beta*dt^2*g +
 2*beta^2*dt^4*g^2)),                    (dt*(a^2 + 2*beta*g*a*dt^2))/
(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2),
       (a*beta*dt^3*g)/(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2)]
% [                                       (2*a*beta*dt^2*g*((2*g*(beta
 - 1/2)*dt^2)/a + 1))/(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2)
 - (2*dt^2*g*(a^2 + 2*beta*g*a*dt^2)*(beta - 1/2))/(a*(a^2 +
 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2)),
                        (((2*g*(beta - 1/2)*dt^2)/a + 1)*(a^2 +
 2*beta*g*a*dt^2))/(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2)
 - (2*beta*dt^4*g^2*(beta - 1/2))/(a^2 + 4*a*beta*dt^2*g +
 2*beta^2*dt^4*g^2),                                 (2*a*beta*dt^3*g)/
(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2),
      (dt*(a^2 + 2*beta*g*a*dt^2))/(a^2 + 4*a*beta*dt^2*g +
 2*beta^2*dt^4*g^2)]
% [          (2*dt*g*(gamma - 1))/a - (2*(beta*gamma*dt^3*g^2
 + a*gamma*dt*g)*((2*g*(beta - 1/2)*dt^2)/a + 1))/(a^2 +
 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2) - (2*dt^3*g^2*gamma*(beta
 - 1/2))/(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2),
 (2*dt^2*g*(beta*gamma*dt^3*g^2 + a*gamma*dt*g)*(beta - 1/2))/
(a*(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2)) - (dt*g*(gamma
 - 1))/a + (a*dt*g*gamma*((2*g*(beta - 1/2)*dt^2)/a + 1))/(a^2 +
 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2), 1 - (2*dt*(beta*gamma*dt^3*g^2
 + a*gamma*dt*g))/(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2),
                           (a*dt^2*g*gamma)/(a^2 + 4*a*beta*dt^2*g +
 2*beta^2*dt^4*g^2)]
% [ (4*dt^2*g*(beta*gamma*dt^3*g^2 + a*gamma*dt*g)*(beta
 - 1/2))/(a*(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2))
 - (2*dt*g*(gamma - 1))/a + (2*a*dt*g*gamma*((2*g*(beta -
 1/2)*dt^2)/a + 1))/(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2),
      (2*dt*g*(gamma - 1))/a - (2*(beta*gamma*dt^3*g^2 +
 a*gamma*dt*g)*((2*g*(beta - 1/2)*dt^2)/a + 1))/(a^2 + 4*a*beta*dt^2*g
 + 2*beta^2*dt^4*g^2) - (2*dt^3*g^2*gamma*(beta - 1/2))/(a^2 +
```

```
 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2),
 (2*a*dt^2*g*gamma)/(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2), 1 -
 (2*dt*(beta*gamma*dt^3*g^2 + a*gamma*dt*g))/(a^2 + 4*a*beta*dt^2*g +
 2*beta^2*dt^4*g^2)]
%
%
% B =
%

     ((a^2 + 2*beta*g*a*dt^2)*(beta*dt^2*sin(dt*w*(n + 1))*(F0/(a*m)
 - (2^(1/2)*F0)/(2*a*m)) - dt^2*sin(dt*n*w)*(F0/(a*m) - (2^(1/2)*F0)/
(2*a*m))*(beta - 1/2)))/(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2) -
 (a*beta*dt^2*g*(beta*dt^2*sin(dt*w*(n + 1))*(F0/(a*m) - (2^(1/2)*F0)/
(a*m)) - dt^2*sin(dt*n*w)*(F0/(a*m) - (2^(1/2)*F0)/(a*m))*(beta -
 1/2)))/(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2)
%

        (2*a*beta*dt^2*g*(beta*dt^2*sin(dt*w*(n + 1))*(F0/(a*m) -
 (2^(1/2)*F0)/(2*a*m)) - dt^2*sin(dt*n*w)*(F0/(a*m) - (2^(1/2)*F0)/
(2*a*m))*(beta - 1/2)))/(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2)
 - ((a^2 + 2*beta*g*a*dt^2)*(beta*dt^2*sin(dt*w*(n + 1))*(F0/(a*m)
 - (2^(1/2)*F0)/(a*m)) - dt^2*sin(dt*n*w)*(F0/(a*m) - (2^(1/2)*F0)/
(a*m))*(beta - 1/2)))/(a^2 + 4*a*beta*dt^2*g + 2*beta^2*dt^4*g^2)
%  dt*gamma*sin(dt*w*(n + 1))*(F0/(a*m) - (2^(1/2)*F0)/(2*a*m))
 - (2*(beta*gamma*dt^3*g^2 + a*gamma*dt*g)*(beta*dt^2*sin(dt*w*(n
 + 1))*(F0/(a*m) - (2^(1/2)*F0)/(2*a*m)) - dt^2*sin(dt*n*w)*(F0/
(a*m) - (2^(1/2)*F0)/(2*a*m))*(beta - 1/2)))/(a^2 + 4*a*beta*dt^2*g
 + 2*beta^2*dt^4*g^2) - dt*sin(dt*n*w)*(F0/(a*m) - (2^(1/2)*F0)/
(2*a*m))*(gamma - 1) - (a*dt*g*gamma*(beta*dt^2*sin(dt*w*(n +
 1))*(F0/(a*m) - (2^(1/2)*F0)/(a*m)) - dt^2*sin(dt*n*w)*(F0/(a*m)
 - (2^(1/2)*F0)/(a*m))*(beta - 1/2)))/(a^2 + 4*a*beta*dt^2*g +
 2*beta^2*dt^4*g^2)
%    (2*(beta*gamma*dt^3*g^2 + a*gamma*dt*g)*(beta*dt^2*sin(dt*w*(n
 + 1))*(F0/(a*m) - (2^(1/2)*F0)/(a*m)) - dt^2*sin(dt*n*w)*(F0/
(a*m) - (2^(1/2)*F0)/(a*m))*(beta - 1/2)))/(a^2 + 4*a*beta*dt^2*g
 + 2*beta^2*dt^4*g^2) - dt*gamma*sin(dt*w*(n + 1))*(F0/(a*m) -
 (2^(1/2)*F0)/(a*m)) + dt*sin(dt*n*w)*(F0/(a*m) - (2^(1/2)*F0)/
(a*m))*(gamma - 1) + (2*a*dt*g*gamma*(beta*dt^2*sin(dt*w*(n +
 1))*(F0/(a*m) - (2^(1/2)*F0)/(2*a*m)) - dt^2*sin(dt*n*w)*(F0/(a*m)
 - (2^(1/2)*F0)/(2*a*m))*(beta - 1/2)))/(a^2 + 4*a*beta*dt^2*g +
 2*beta^2*dt^4*g^2)
```

Q1.2

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

e = [];
```

```
for dt = linspace(0, 1, 1001)
e = [e, max(abs(eig(eval(A))))];
end

dt = linspace(0, 1, 1001);
subplot(1, 1, 1);
plot(dt, e);
title('Le module de valeur propre de différant pas');

% On trouve que quand le pas est inférieure à 0.024, tous les
 modules de valeur propre est presque égale à 1, et quand le pas est
 supérieure à 0.024, les modules de valeur propre supérieure à 1.
```



Q1.3

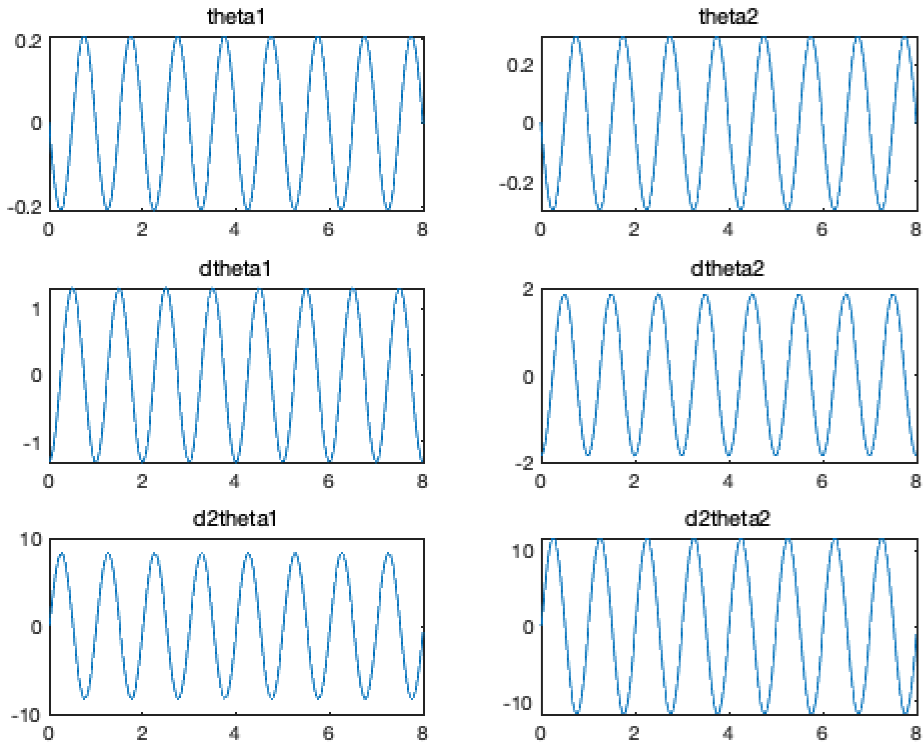
```
theta1_0 = 0;
theta2_0 = 0;
dtheta1_0 = - 1.31519275;
dtheta2_0 = - 1.85996342;

q0 = [theta1_0; theta2_0];
dq0 = [dtheta1_0; dtheta2_0];
d2q0 = eval(M4) * q0;
```

Q1.4

```
% U = [q; dq]
% Un1 = A * Un + B, d2q = M4 * q + M5 * sin(w * t).
% Ce sont les relations.
```

Q1.5

```
T0 = 8;
dt = 0.02;
U = [q0; dq0];
q = [q0];
dq = [dq0];
d2q = [d2q0];
for n = 0 : (T0 / dt - 1)
U = eval(A) * U + eval(B);
q = [q, U(1:2)];
dq = [dq, U(3:4)];
d2q = [d2q, eval(M4 * U(1:2) + M5 * sin(w * n * dt))];
end

t = (0 : (T0 / dt)) * dt;
subplot(3, 2, 1);
plot(t, q(1, :));
title('theta1');
subplot(3, 2, 3);
plot(t, dq(1, :));
title('dtheta1');
subplot(3, 2, 5);
plot(t, d2q(1, :));
title('d2theta1');
subplot(3, 2, 2);
plot(t, q(2, :));
title('theta2');
subplot(3, 2, 4);
plot(t, dq(2, :));
title('dtheta2');
subplot(3, 2, 6);
plot(t, d2q(2, :));
title('d2theta2');
```



Q1.6

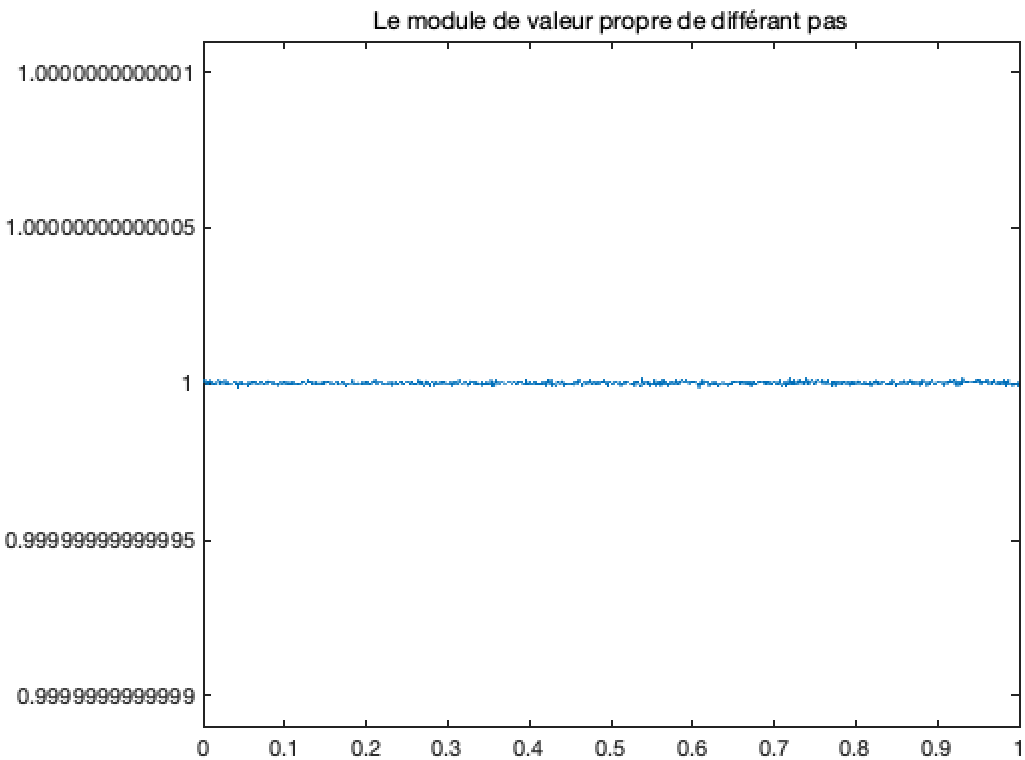
```
q(:, 1 : 3); % ce sont les valeurs de q à 0s , dt , 2dt.
q(:, 0.5 / dt + 1); %c'est le valeur de q à 0.5s.
% Ce sont
% 0   -0.0263   -0.0522   -0.299e-3
% 0   -0.0372   -0.0738   -0.423e-3
dq(:, 1 : 3); % ce sont les valeurs de dq à 0s , dt , 2dt.
dq(:, 0.5 / dt + 1); %c'est le valeur de dq à 0.5s.
% Ce sont
% -1.32   -1.30   -1.27   1.31
% -1.86   -1.85   -1.80   1.86
d2q(:, 1 : 3); % ce sont les valeurs de d2q à 0s , dt , 2dt.
d2q(:, 0.5 / dt + 1); %c'est le valeur de d2q à 0.5s.
% Ce sont
% 0    0.302    1.33   0.737
% 0    0.428    1.89   1.04
```

Q2.1

```
% En utilisant le résultat de Q1.1, c'est la même matrice.
% Puisqu'elle est trop grande, je ne remonte pas.
```

Q2.2

```
beta = 0.25;
e = [];
for dt = linspace(0, 1, 1001)
e = [e, max(abs(eig(eval(A))))];
end

dt = linspace(0, 1, 1001);
subplot(1, 1, 1);
plot(dt, e);
title('Le module de valeur propre de différant pas');

% Le module de valeur propre est toujours presque égale à 1.
```



Q2.3

```
% U = [q; dq]
% Un1 = A * Un + B, d2q = M4 * q + M5 * sin(w * t).
% Ce sont les mêmes relations que Q1.4, mais beta change.
```

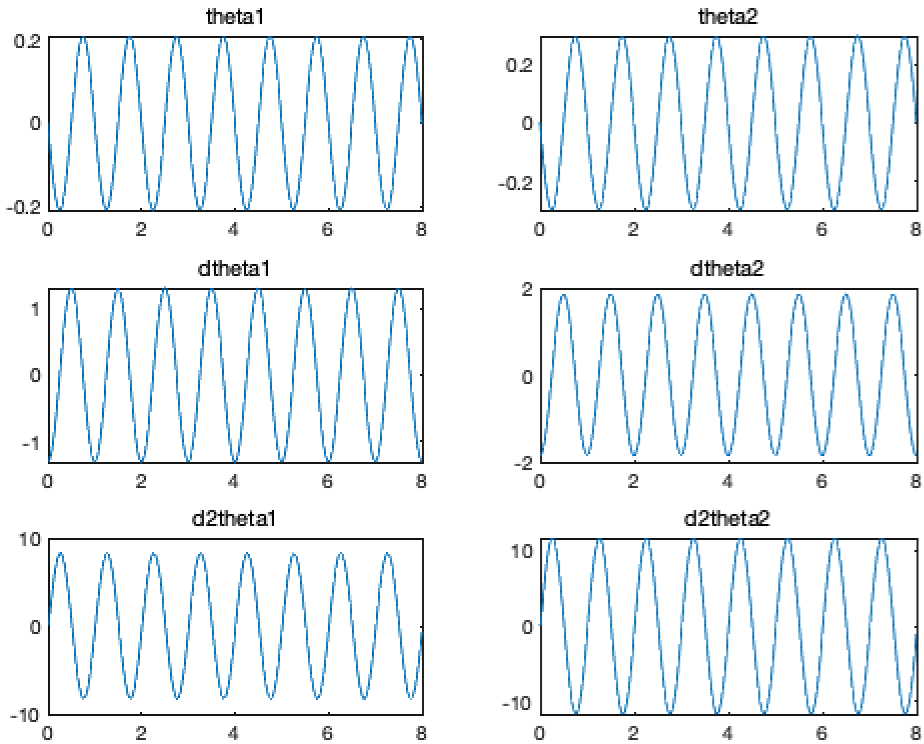
Q2.4

```
% Un1 = A * Un + B
```

```
% Je ne comprend pas ce que Q2.3 et Q2.4 veulent.
% A mon avis, c'est la même chose.
```

Q2.5

```
T0 = 8;
dt = 0.02;
U = [q0; dq0];
q = [q0];
dq = [dq0];
d2q = [d2q0];
for n = 0 : (T0 / dt - 1)
U = eval(A) * U + eval(B);
q = [q, U(1:2)];
dq = [dq, U(3:4)];
d2q = [d2q, eval(M4 * U(1:2) + M5 * sin(w * n * dt))];
end

t = (0 : (T0 / dt)) * dt;
subplot(3, 2, 1);
plot(t, q(1, :));
title('theta1');
subplot(3, 2, 3);
plot(t, dq(1, :));
title('dtheta1');
subplot(3, 2, 5);
plot(t, d2q(1, :));
title('d2theta1');
subplot(3, 2, 2);
plot(t, q(2, :));
title('theta2');
subplot(3, 2, 4);
plot(t, dq(2, :));
title('dtheta2');
subplot(3, 2, 6);
plot(t, d2q(2, :));
title('d2theta2');
```



Q2.6

```
q(:, 1 : 3); % ce sont les valeurs de q à 0s , dt , 2dt.
q(:, 0.5 / dt + 1); %c'est le valeur de q à 0.5s.
% Ce sont
% 0   -0.0262   -0.0520   -0.0009
% 0   -0.0371   -0.0735   -0.0013
dq(:, 1 : 3); % ce sont les valeurs de dq à 0s , dt , 2dt.
dq(:, 0.5 / dt + 1); %c'est le valeur de dq à 0.5s.
% Ce sont
% -1.32   -1.30   -1.27   1.31
% -1.86   -1.85   -1.80   1.86
```

Published with MATLAB® R2018a

Oscillateur non linéaire
à un degré de liberté

Table of Contents

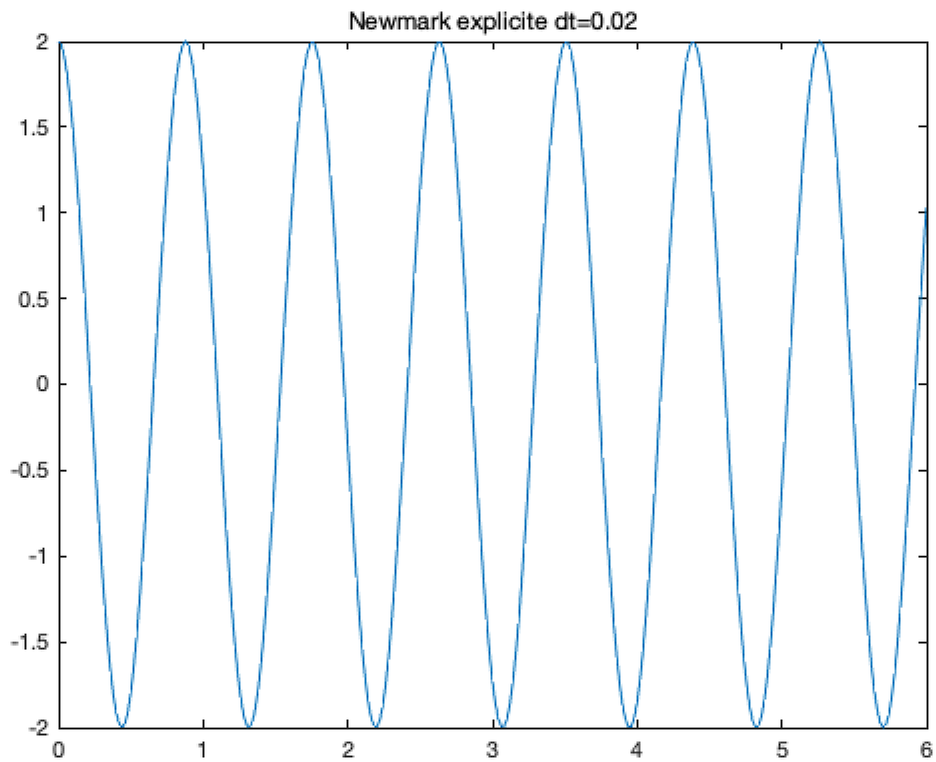
Q1.1

```
q0 = 2;
dq0 = 0;
w0 = 2*pi;
alpha = 0.1;
ddq0 = - w0^2*q0*(1+alpha*q0^2);
T0=6;
gama1=0.5;beta1=0;
%on sait les relations
%q1(inc+1) = q1(inc) + dt1 * dq1(inc)+ dt1*dt1*0.5*ddq1(inc)
%ddq1(inc+1)=- w0^2*q1(inc+1)*(1+alpha*q1(inc+1)^2)
%dq1(inc+1) = dq1(inc) +0.5*dt1 * (ddq1(inc) + ddq1(inc+1))
```

Q1.2

```
dt1 =0.02;
t1 =(0:dt1:T0)';
np1=size(t1,1);
q1=zeros(np1,1);
dq1=zeros(np1,1);
ddq1=zeros(np1,1);
energ1=zeros(np1,1);

q1(1)=q0;
dq1(1)=dq0;
ddq1(1)=ddq0;

for inc =1:(np1-1)
    q1(inc+1) = q1(inc) + dt1 * dq1(inc)+ dt1*dt1*0.5*ddq1(inc);
    ddq1(inc+1)=- w0^2*q1(inc+1)*(1+alpha*q1(inc+1)^2);
    dq1(inc+1) = dq1(inc) +0.5*dt1 * (ddq1(inc) + ddq1(inc+1));
```

```
end
plot(t1,q1)
title('Newmark explicite dt=0.02')
```



Q1.3

```
q1(1);%t=0
q1(2);%t=dt
q1(3);%t=2*dt
q1(301);%t=T0
% les valeurs numériques de q à 0s, dt, 2dt et T0 sont:
% 2   1.9779   1.9123   1.0329
```

Q2.1

```
gama2=0.5;beta2=0.25;
%on cherche à minimiser la valeur absolue de: ddq+w0^2*q*(1+alpha*q^2)
%on voudrais cette valeur egale 0
```

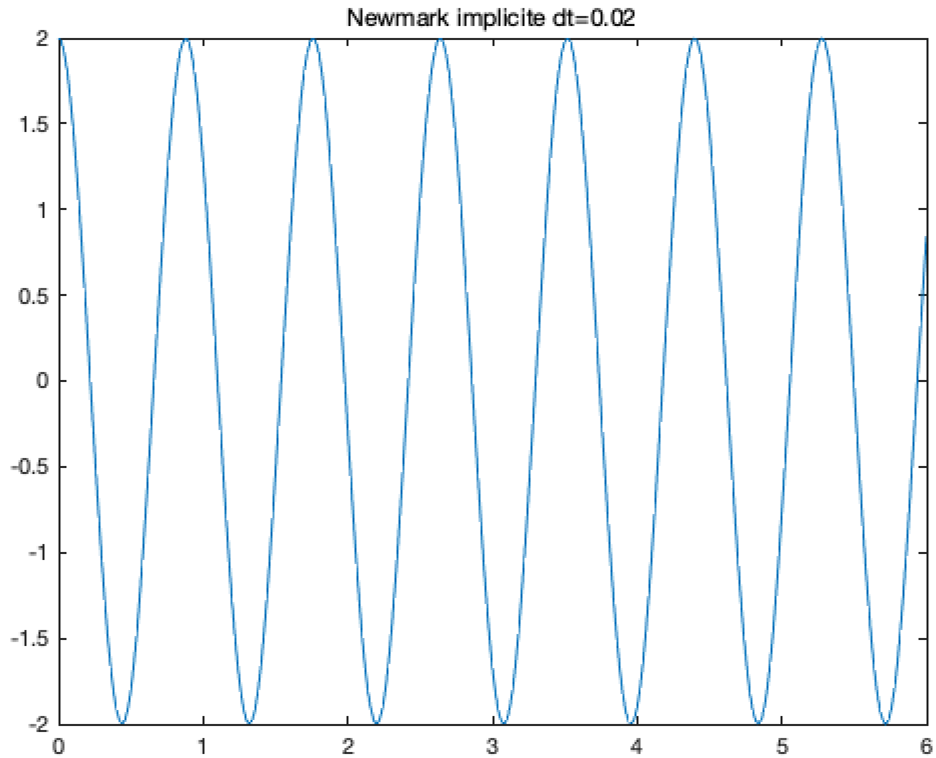
Q2.2

```
% on a cddq=-f(dd1,dq,q)/(Df/Dddq+Df/Dq*betadt^2).
% et f=ddq+w0^2*q*(1+alpha*q^2), Df/Dddq=1,
% Df/Dq=w0^2+3*w0^2*alpha*q^2
```

```
% Alors, cddq=-(ddq+w0^2*q*(1+alpha*q^2))/
(1+(w0^2+3*w0^2*alpha*q^2)*betadt^2)
```

Q2.3

```
q2=zeros(np1,1);
dq2=zeros(np1,1);
ddq2=zeros(np1,1);
energ2=zeros(np1,1);
q2(1)=q0;
dq2(1)=dq0;
ddq2(1)=ddq0;
e=0.01; %supposons le erreur est 0.01 pour verifier abs(ddq
+w0*w0*q*(1+alpha*q*q))<e
for inc =1:(np1-1)
    q2(inc+1) = q2(inc) + dt1 * dq2(inc)+ dt1*dt1*(0.5-
beta2)*ddq2(inc);
    dq2(inc+1) = dq2(inc) +dt1 *(1-gama2)*ddq2(inc);
    ddq2(inc+1)=0;
    while  abs(ddq2(inc+1)+w0*w0*q2(inc+1)*(1+alpha*q2(inc+1)*q2(inc
+1)))> e
        cddq2 = (-(ddq2(inc+1)+w0*w0*q2(inc+1)*(1+alpha*q2(inc
+1)*q2(inc+1))))/(1+beta2*dt1*dt1*(w0*w0+3*w0*w0*alpha*q2(inc
+1)*q2(inc+1)));
        cdq2=gama2*dt1* cddq2;
        cq2=beta2*dt1*dt1* cddq2;
        q2(inc+1)=q2(inc+1)+cq2;
        dq2(inc+1)=dq2(inc+1)+cdq2;
        ddq2(inc+1)=ddq2(inc+1)+cddq2;
    end
end
plot(t1,q2)
title('Newmark implicite dt=0.02')
```



Q2.4

```
q2(1);%t=0
q2(2);%t=dt1
q2(3);%t=2*dt1
q2(301);%t=T0
% les valeurs numériques de q à 0s, dt, 2dt et T0 sont:
% 2   1.9781   1.9131   0.8478
```

Q3.1

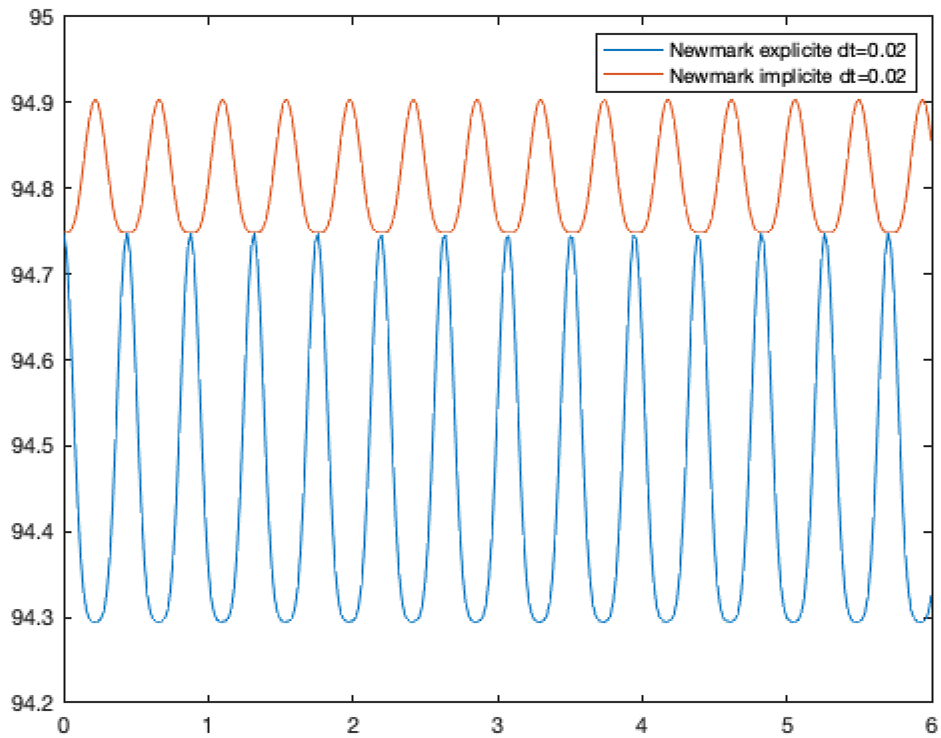
```
%il y a deux partie : l'energie cinetique et l'energie potentiel
%pour l'energie cinetique, c'est 0.5*dq^2
%pour l'energie potentiel,on fait un integrale,
%c'est 0.5*w0*w0*q*q+0.25*alpha*w0*w0*q^4
```

Q3.2

```
for inc =1:np1
    energ1(inc)= 0.5*dq1(inc)^2 +
 0.5*w0*w0*q1(inc)*q1(inc)+0.25*alpha*w0*w0*q1(inc)^4;
    energ2(inc)= 0.5*dq2(inc)^2 +
 0.5*w0*w0*q2(inc)*q2(inc)+0.25*alpha*w0*w0*q2(inc)^4;
end
```

Q3.3

```
plot(t1,energ1,t1,energ2);
legend('Newmark explicite dt=0.02','Newmark implicite dt=0.02');
%l'energie implicite est toujours plus grande de l'energie explicite
%mais, quelque fois, ils ont la meme l'energie
```



Published with MATLAB® R2018a