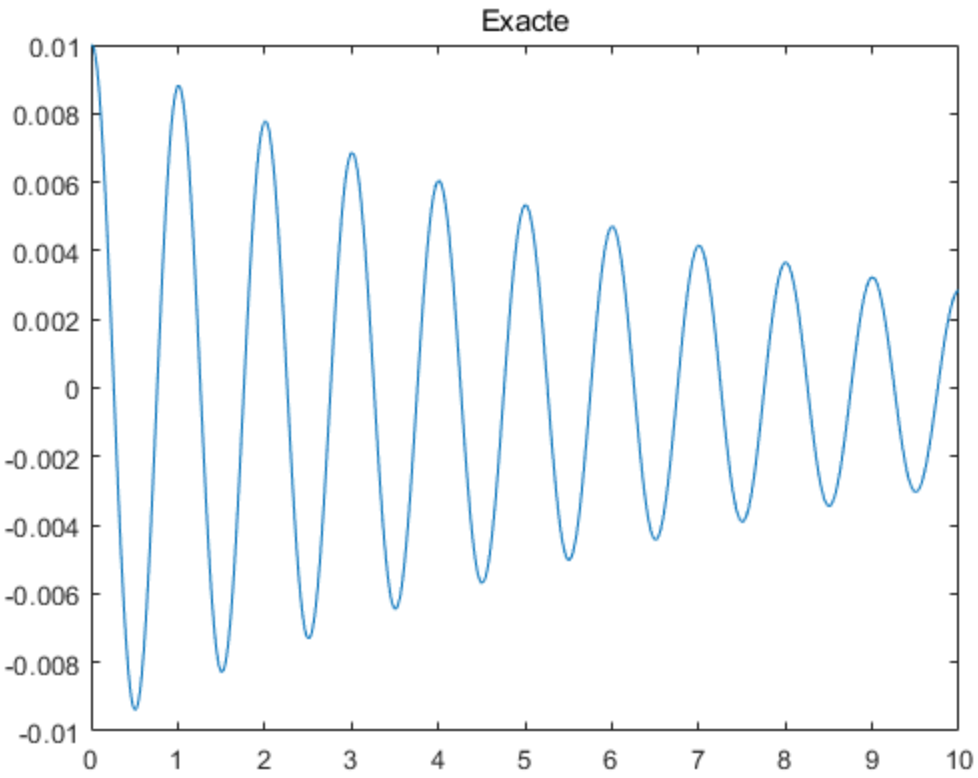
cas d'oscillateur amorti

Table of Contents

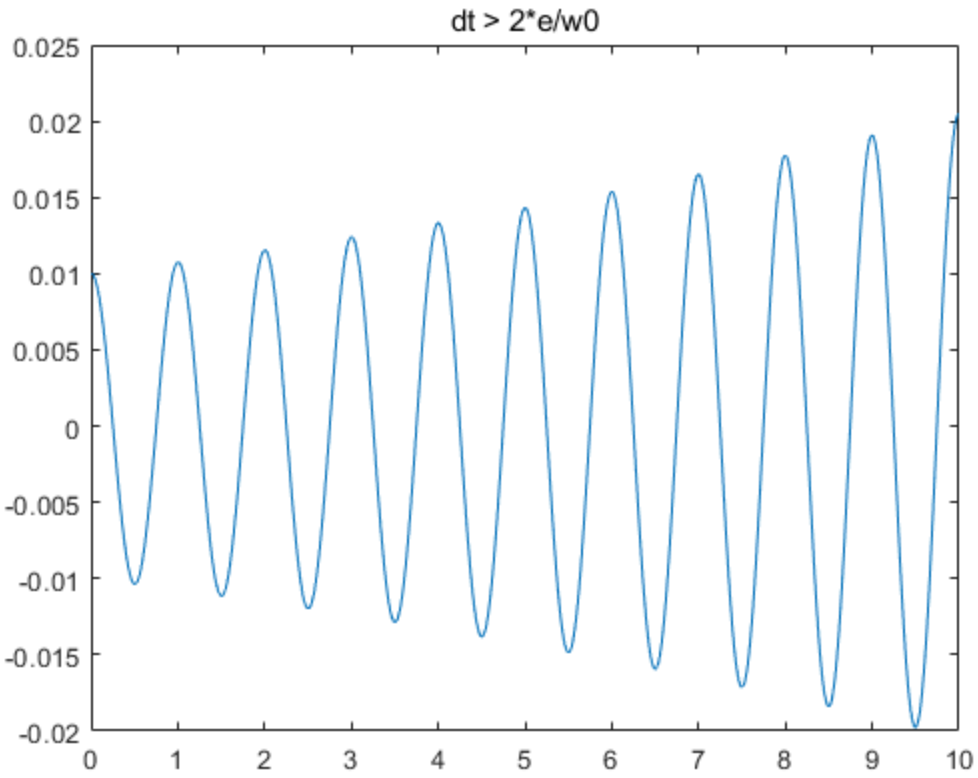
1.1

```
T0 = 1;
w0 = 2*pi;
w0c = w0*w0;
e = 0.02;
m = 1; % la valeur de m n'influence pas le calcul
b = 2*e*w0*m;
x0 = 0.01;
dx0 = 0;
F = 0;
x = [];
x(1) = x0;
n = 1;
omega = w0*(1-e^2)^0.5;
for t = 0:0.01:10*T0
    n = n + 1;
    x(n) = exp(-e*w0*t)*(x0*cos(omega*t) + (e*w0*x0 + dx0)/
omega*sin(omega*t));
end
t = linspace(0,10*T0,n);
plot(t,x);
title('Exacte')
```



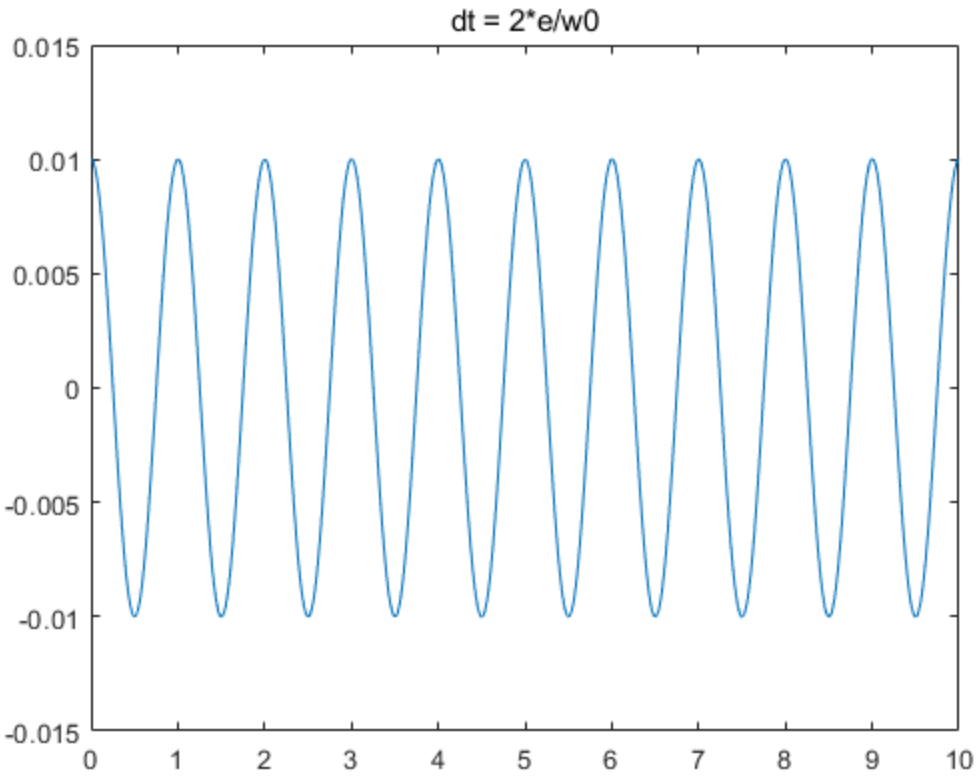
1.1.a

```
ti = 2*e/w0; % ti = 0.0064
dt = 0.01; % dt > 2*e/w0
A1 = [1,dt;-dt*w0c,1-2*dt*e*w0];
X1 = [x0;dx0];
X1b = [];
dX1b = [];
X1b(1) = x0;
dX1b(1) = dx0;
n1 = 1;
for t = 0:dt:10*T0
    n1 = n1 + 1;
    X1 = A1*X1;
    X1b(n1) = X1(1,1);
    dX1b(n1) = X1(2,1);
end
t1 = linspace(0,10*T0,n1);
plot(t1,X1b);
title('dt > 2*e/w0');
% Remarque: x diverge;
```



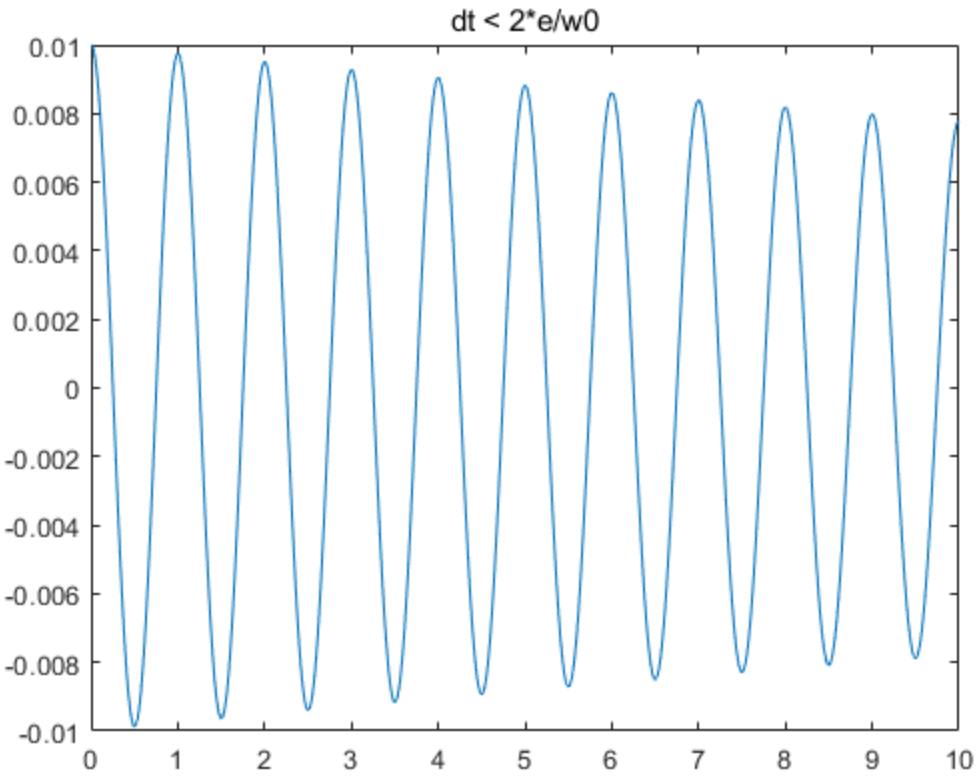
1.1b

```
dt = ti;% dt = 2*e/w0
A2 = [1,dt;-dt*w0c,1-2*dt*e*w0];
X2 = [x0;dx0];
X2b = [];
dX2b = [];
X2b(1) = x0;
dX2b(1) = dx0;
n2 = 1;
for t = 0:dt:10*T0
    n2 = n2 + 1;
    X2 = A2*X2;
    X2b(n2) = X2(1,1);
    dX2b(n2) = X2(2,1);
end
t2 = linspace(0,10*T0,n2);
plot(t2,X2b);
title('dt = 2*e/w0');
% Remarque:x est sinusoidale, ni converge ni diverge.
```



1.1c

```
dt = 0.8*ti; % dt < 2*e/w0
A3 = [1,dt;-dt*w0c,1-2*dt*e*w0];
X3 = [x0;dx0];
X3b = [];
dX3b = [];
X3b(1) = x0;
dX3b(1) = dx0;
n3 = 1;
for t = 0:dt:10*T0
    n3 = n3 + 1;
    X3 = A3*X3;
    X3b(n3) = X3(1,1);
    dX3b(n2) = X3(2,1);
end
t3 = linspace(0,10*T0,n3);
plot(t3,X3b);
title('dt < 2*e/w0');
% Remarque: x converge.
```



1.1d

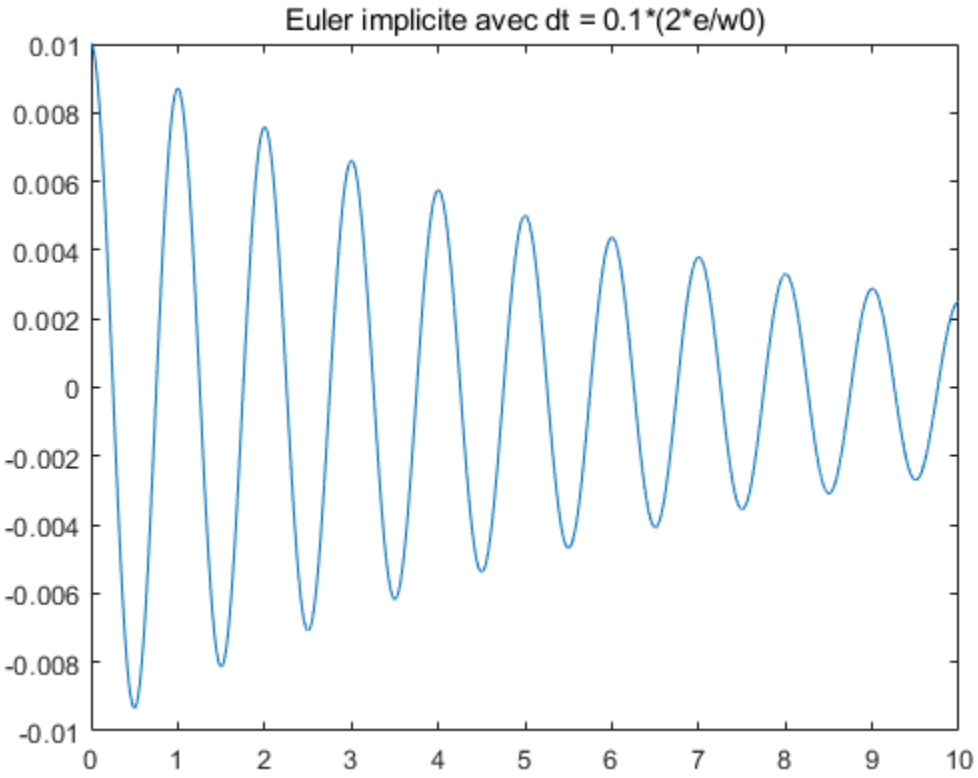
```
%le rapport de dt/(2*e/W0) est un critere de la solution, et le
 rapport
%doit etre plus petit que 1 pour que la solution soit precis (converge
 parce que l'amortissement)
```

1.2

Euler implicite

```
dt = 0.1*2*e/w0; % dt est plus petit que 2*e/w0 pour que la solution
 soit précis
A4 = [1+2*dt*e*w0,dt;-dt*w0c,1]/(1 + 2*dt*e*w0 + dt^2*w0c); % on
 calcule la matrice d'amplification du cas amorti
X4 = [x0;dx0];
X4b = [];
dX4b = [];
X4b(1) = x0;
dX4b(1) = dx0;
n4 = 1;
for t = 0:dt:10*T0
    n4 = n4 + 1;
    X4 = A4*X4;
    X4b(n4) = X4(1,1);
```

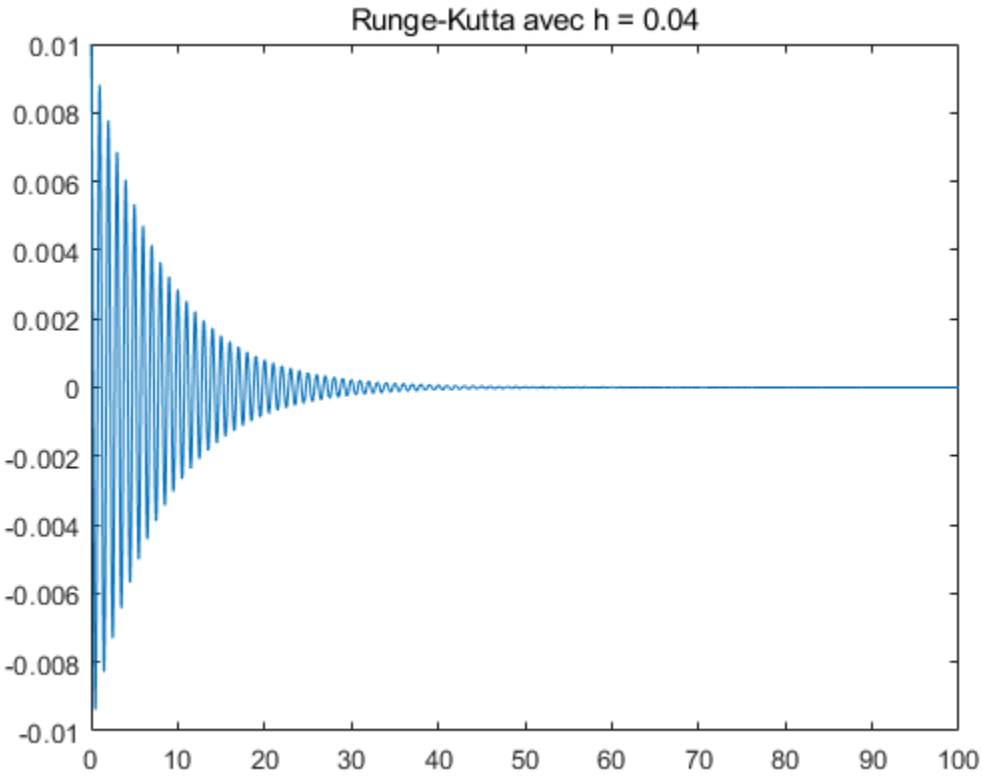
```
    dX4b(n4) = X4(2,1);
end
t4 = linspace(0,10*T0,n4);
plot(t4,X4b);
title('Euler implicite avec dt = 0.1*(2*e/w0)');
% comme on l'a fait dans le TD1,la solution d'Euler implicite est
 toujours
% convergente. Mais la le rapport de dt/(2*e/W0) influence la vitesse
 de
% converge
```



1.3a h = 0.04

```
h = 0.04;
dt = h*2*2^0.5/w0;
A5 = [0,1;-w0c,-2*e*w0];% on calcule la matrice d'amplification de
 Runge-Kutta du cas amorti
X5 = [x0;dx0];
X5b = [];
dX5b = [];
X5b(1) = x0;
dX5b(1) = dx0;
n5 = 1;
for t = 0:dt:100*T0
    n5 = n5 + 1;
    k1 = A5*X5;
```

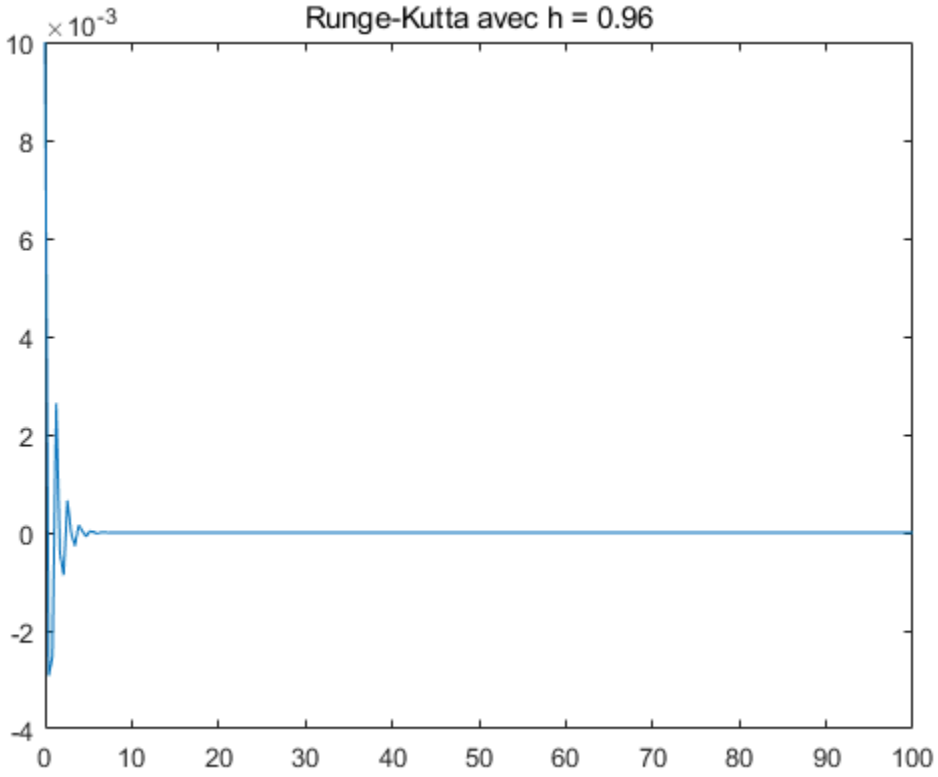
```
    k2 = A5*(X5 + k1*dt/2);
    k3 = A5*(X5 + k2*dt/2);
    k4 = A5*(X5 + k3*dt);
    K = (k1 + 2*k2 + 2*k3 + k4)/6;
    X5 = X5 + K *dt;
    X5b(n5) = X5(1,1);
    dX5b(n5) = X5(2,1);
end
t5 = linspace(0,100*T0,n5);
plot(t5,X5b);
title('Runge-Kutta avec h = 0.04')
```



1.3a h = 0.96

```
h = 0.96;
dt = h*2*2^0.5/w0;
A5 = [0,1;-w0c,-2*e*w0];% on calcule la matrice d'amplification de
 Runge-Kutta du cas amorti
X5 = [x0;dx0];
X5b = [];
dX5b = [];
X5b(1) = x0;
dX5b(1) = dx0;
n5 = 1;
for t = 0:dt:100*T0
    n5 = n5 + 1;
```

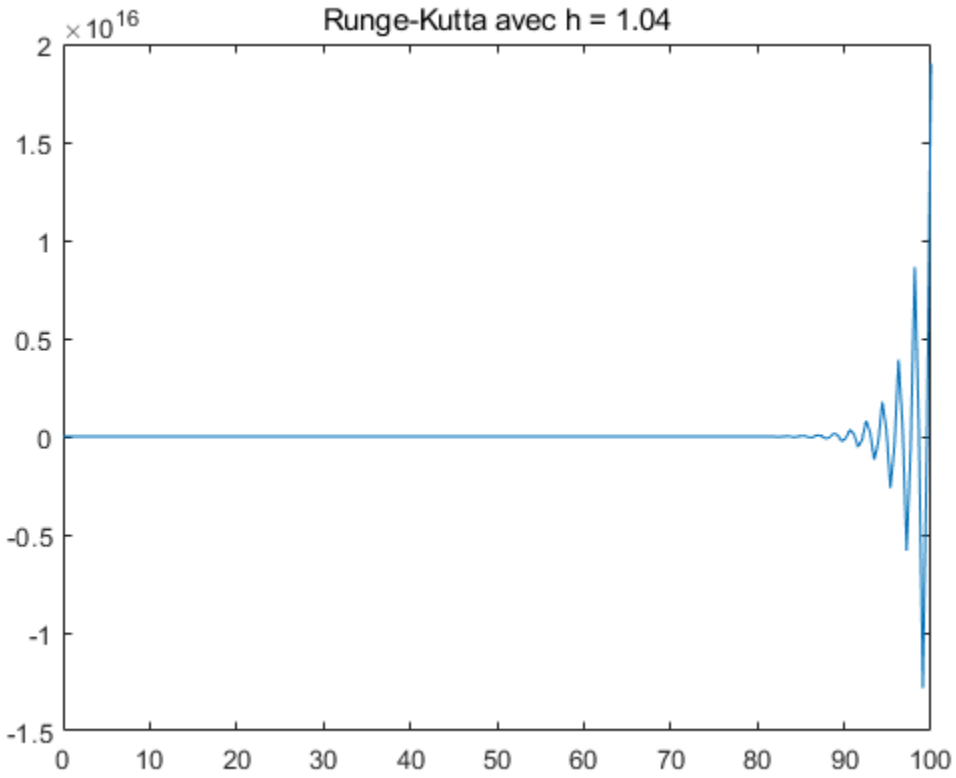
```
    k1 = A5*X5;
    k2 = A5*(X5 + k1*dt/2);
    k3 = A5*(X5 + k2*dt/2);
    k4 = A5*(X5 + k3*dt);
    K = (k1 + 2*k2 + 2*k3 + k4)/6;
    X5 = X5 + K *dt;
    X5b(n5) = X5(1,1);
    dX5b(n5) = X5(2,1);
end
t5 = linspace(0,100*T0,n5);
plot(t5,X5b);
title('Runge-Kutta avec h = 0.96')
```



1.3a h =1.04

```
h = 1.04;
dt = h*2*2^0.5/w0;
A5 = [0,1;-w0c,-2*e*w0];% on calcule la matrice d'amplification de
 Runge-Kutta du cas amorti
X5 = [x0;dx0];
X5b = [];
dX5b = [];
X5b(1) = x0;
dX5b(1) = dx0;
n5 = 1;
for t = 0:dt:100*T0
```

```
    n5 = n5 + 1;
    k1 = A5*X5;
    k2 = A5*(X5 + k1*dt/2);
    k3 = A5*(X5 + k2*dt/2);
    k4 = A5*(X5 + k3*dt);
    K = (k1 + 2*k2 + 2*k3 + k4)/6;
    X5 = X5 + K *dt;
    X5b(n5) = X5(1,1);
    dX5b(n5) = X5(2,1);
end
t5 = linspace(0,100*T0,n5);
plot(t5,X5b);
title('Runge-Kutta avec h = 1.04')
%la stabilite de x depend de h(c'est en fait depend de dt)
% il exist un point critique de h entre 0.96 et 1.04, si h est moins
 que la valeur
% critique, h est plus grand, x est moins stable
```



1.3b

on teste les valeurs entre 0.96 et 1.04

```
hc = 1.0135;
%hmax = 1.0138
%hmin = 1.0132
tc = hc * 2*2^0.5/w0; % tc = 0.4562
```

Published with MATLAB® R2019b