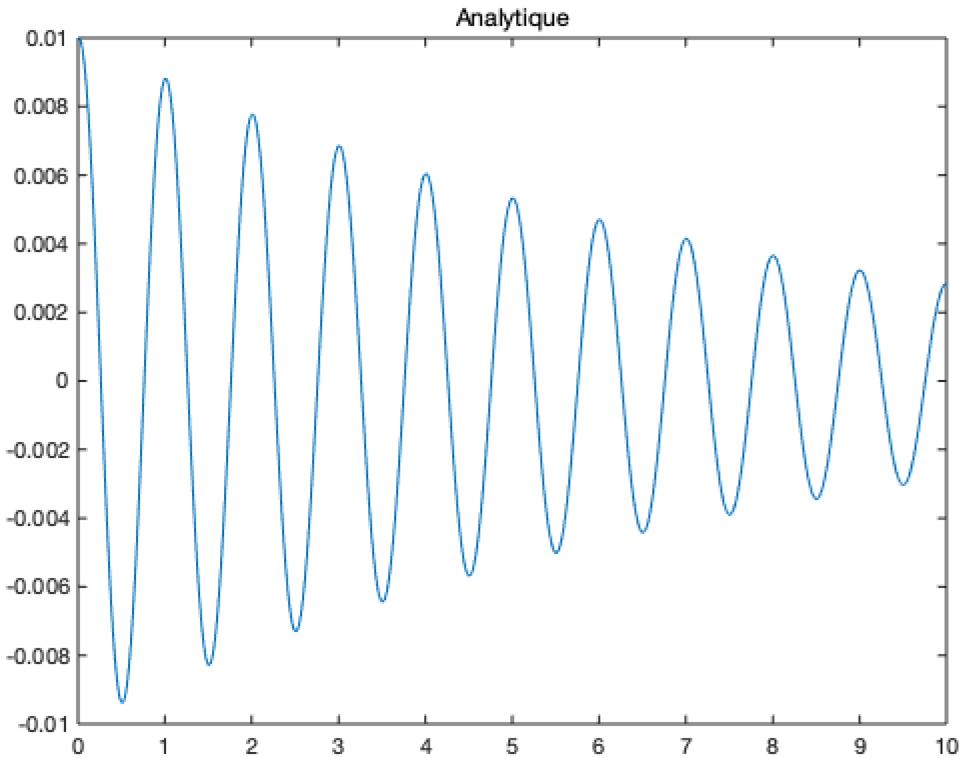
Etude d'un oscillateur linéaire marti à un degré de liberté

Table of Contents

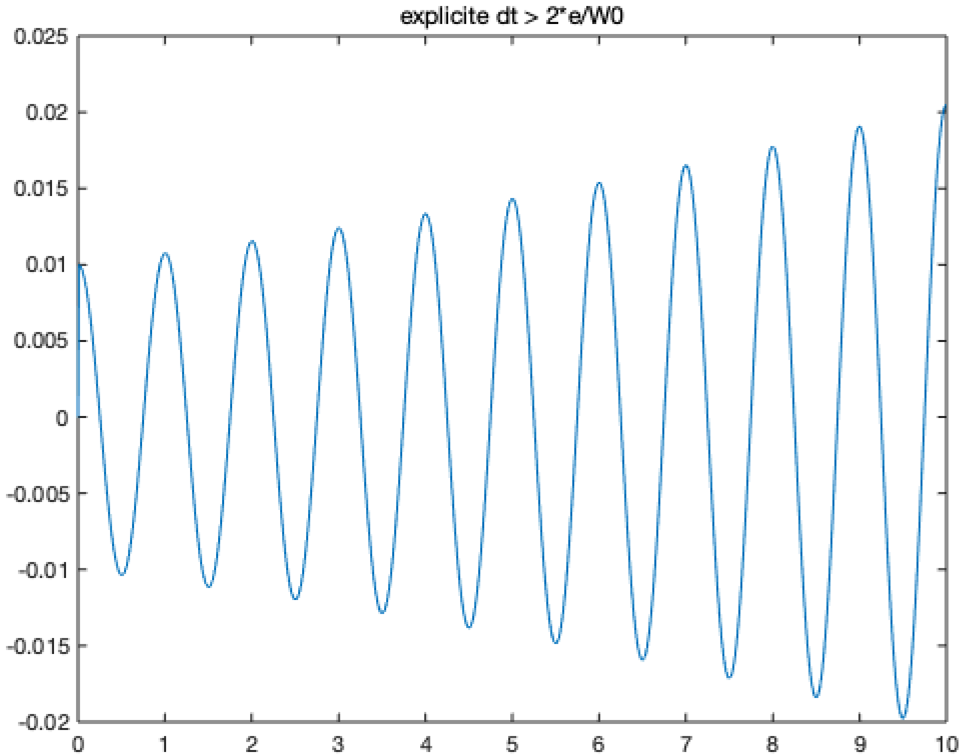
1.1

```
T0 = 1;
W0 = 2*pi/T0;
e = 0.02;
m = 1;
b = 2*e*W0*m;
X0 = 0.01;
dX0 = 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');
```



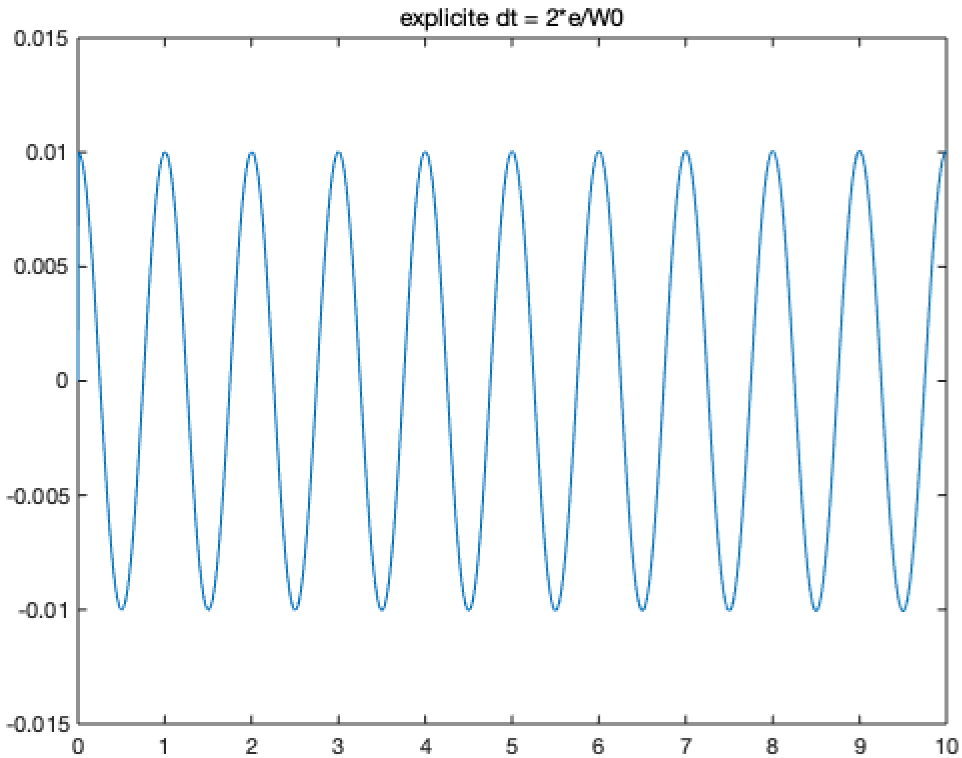
1.1.a

```
%2*e/W0=0.0064
clf;
dta=0.01;
A = [1,dta;-dta*W0^2,1-2*dta*e*W0];
X = [X0;dX0];
n=1;
x1a=[];
dx1a=[];
for t = 0:dta:10*T0
    n = n + 1;
    X = A*X;
    x1a(n) = X(1,1);
    dx1a(n) = X(2,1);
end
ta =linspace(0,10*T0,n);
plot(ta,x1a);
title('explicite dt > 2*e/W0');
%on peut voir que x diverge
```



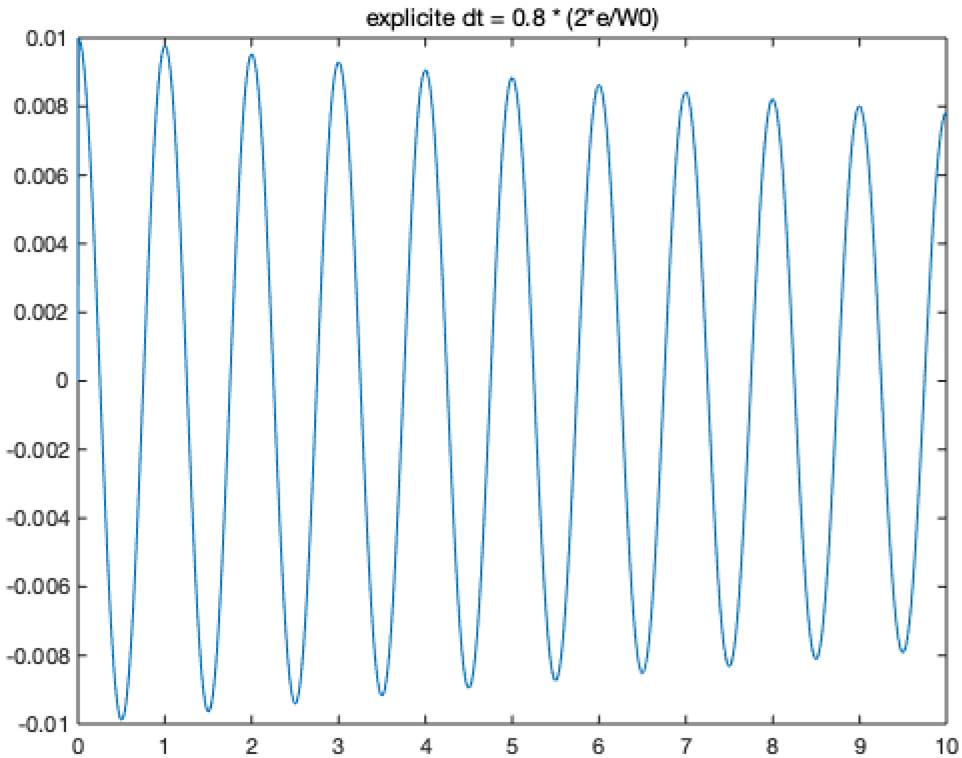
1.1.b

```
clf;
dtb=0.0064;
A = [1,dtb;-dtb*W0^2,1-2*dtb*e*W0];
X = [X0;dX0];
n=1;
x1b=[];
dx1b=[];
for t = 0:dtb:10*T0
    n = n + 1;
    X = A*X;
    x1b(n) = X(1,1);
    dx1b(n) = X(2,1);
end
tb =linspace(0,10*T0,n);
plot(tb,x1b);
title('explicite dt = 2*e/W0');
%on peut voir que x est sinusoidale,
%ni converge ni diverge.
```



1.1.c

```
clf;
dtc=0.0064*0.8;
A = [1,dtc;-dtc*W0^2,1-2*dtc*e*W0];
X = [X0;dX0];
n=1;
x1c=[];
dx1c=[];
for t = 0:dtc:10*T0
    n = n + 1;
    X = A*X;
    x1c(n) = X(1,1);
    dx1c(n) = X(2,1);
end
tc =linspace(0,10*T0,n);
plot(tc,x1c);
title('explicite dt = 0.8 * (2*e/W0)');
%on peut voir que x converge
```



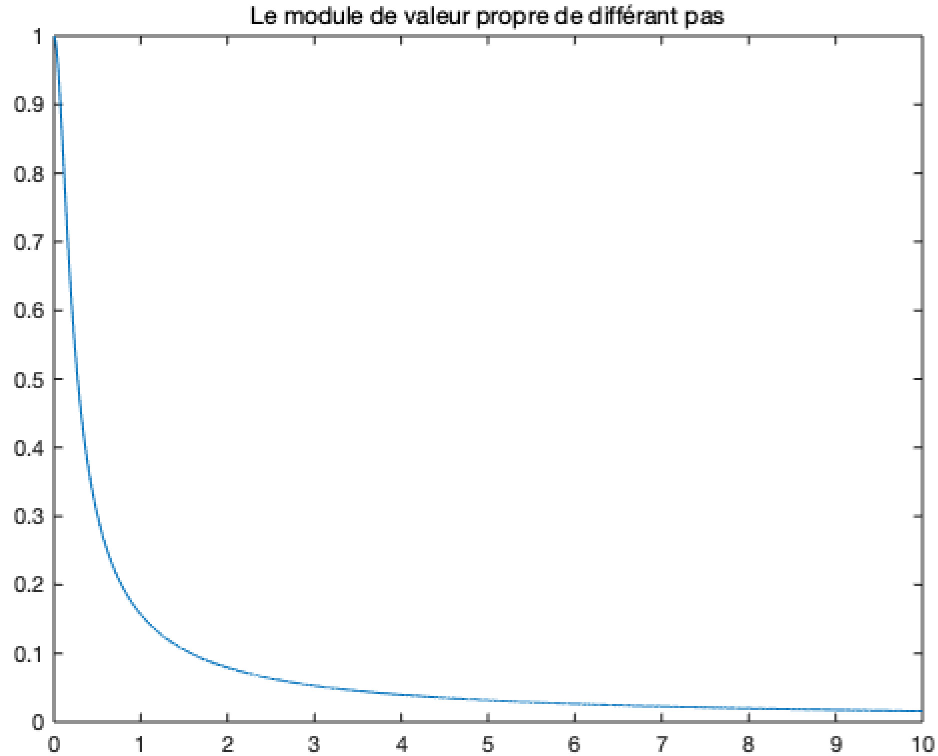
1.1.d

```
%la valeur de ddx+2*e*W0*dx+W0*x est un critere qui permet de etudier
 la
%precision de la solution, si cette valeur est pres de 0, la solution
 est
%precis
%la valeur de dt est un critere qui permet de etudier la precision de
 la solution
%et le rapport de dt/(2*e/W0) doit etre plus petit que 1 pour la
 solution est precis
```

1.2

```
clf;
dt=0:0.001:10*T0;
for j=1:length(dt)
    A_im{1,j} = [1+2*dt(j)*e*W0,dt(j);-dt(j)*W0^2,1]/(1 + 2*dt(j)*e*W0
 + dt(j)^2*W0^2);
    VPA(1,j)=max(abs(eig(A_im{1,j})));
end
plot(dt,VPA)
title('Le module de valeur propre de différant pas');
%les valeurs absolus des valeurs propres sont toujours moins de 1
%schéma inconditionnellement stable
```

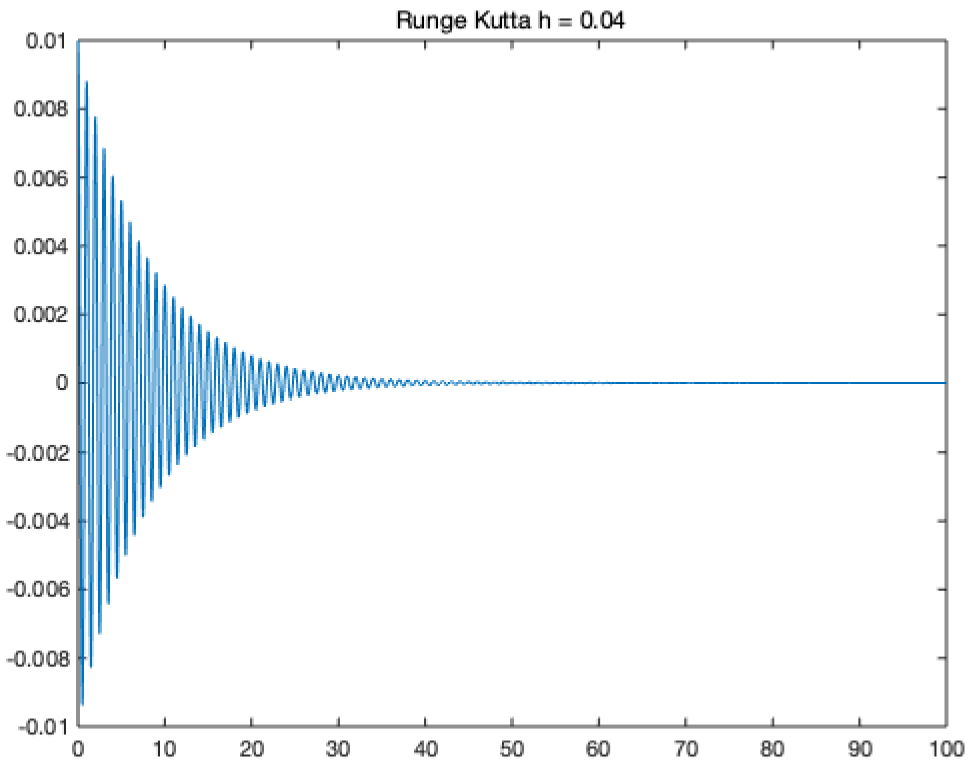
```
%donc pour implicite,il n'y a pas de temps critique
```



1.3.a h=0.04

```
clf;
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);
```

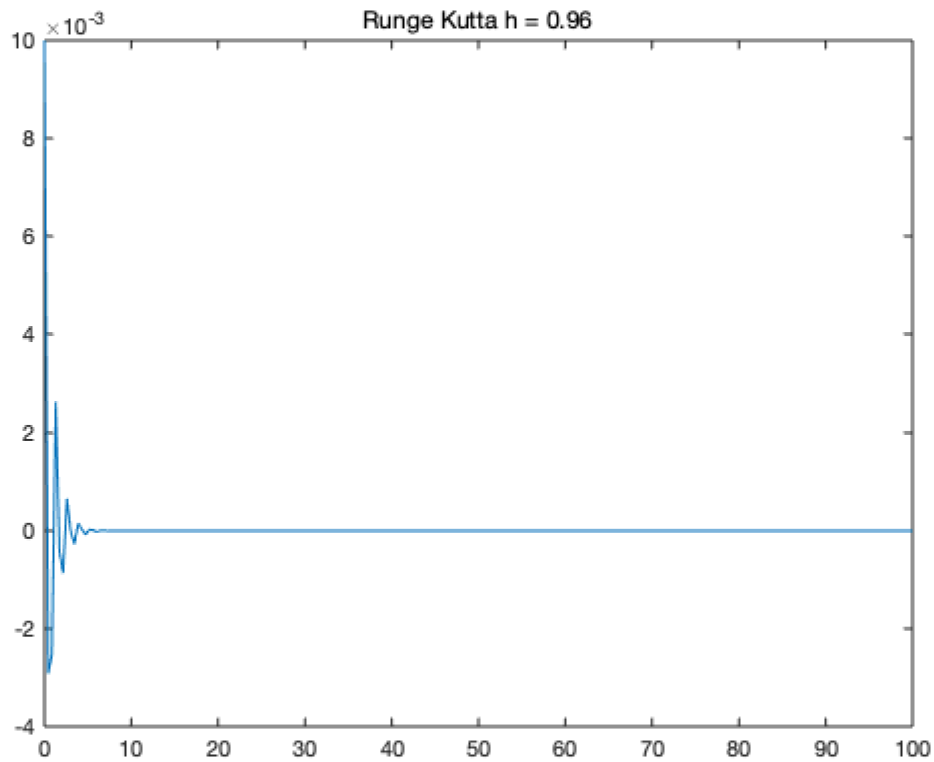
```
title('Runge Kutta h = 0.04')
```



1.3.a h=0.96

```
clf;
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);
```

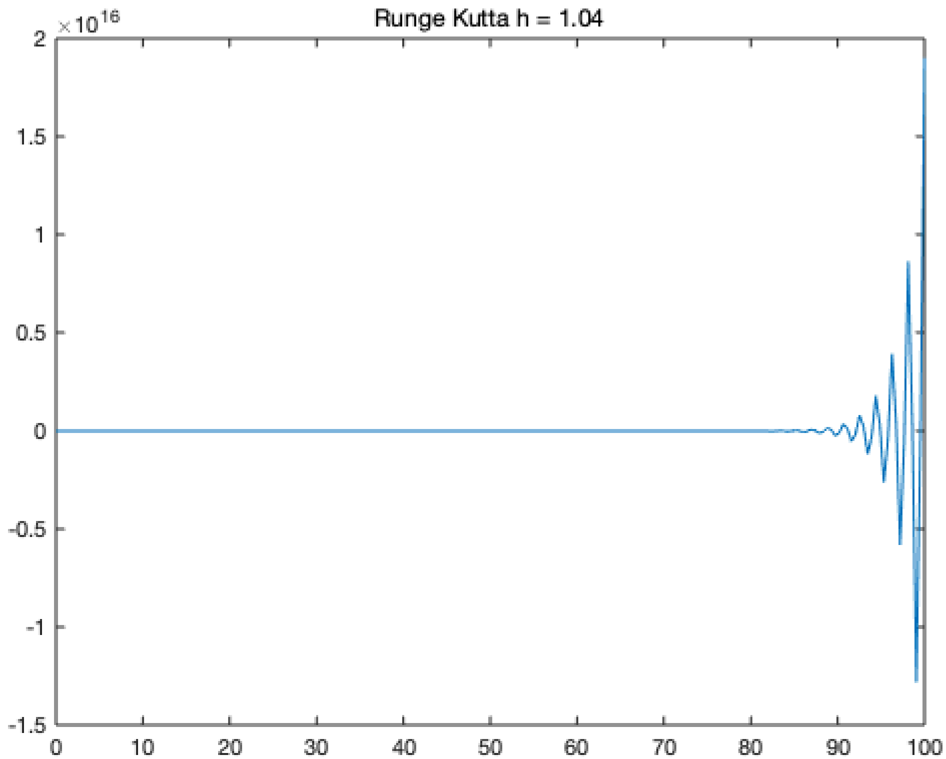
```
title('Runge Kutta h = 0.96')
```



1.3.a h=1.04

```
clf;
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')
```



1.3.a

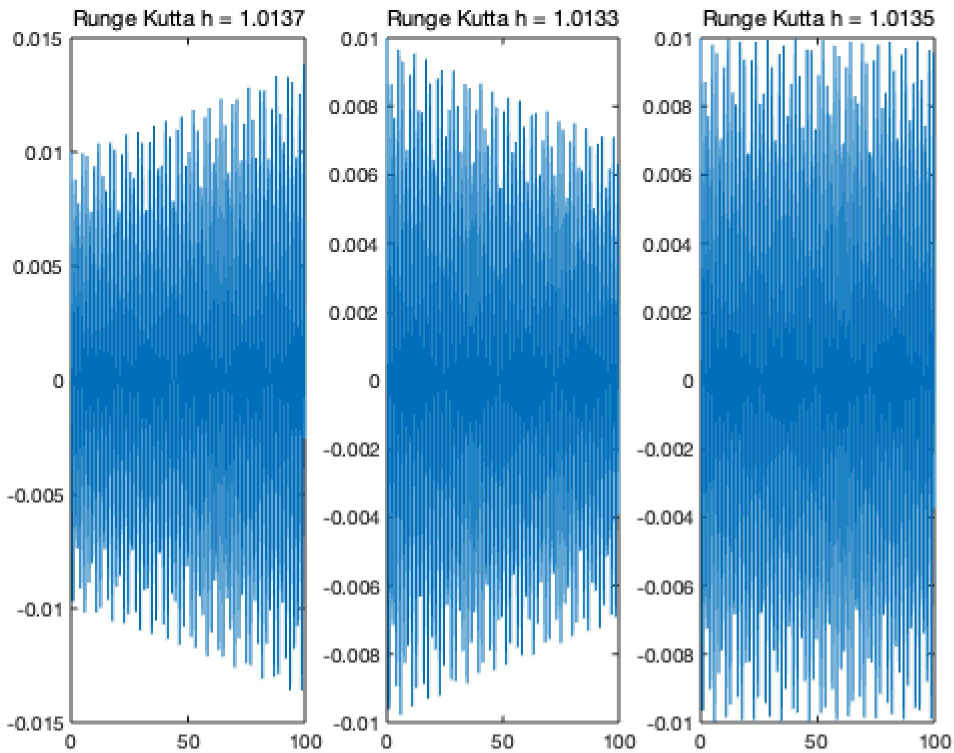
```
%la stabilite de x depend de h
% h depasse un valeur critique, dt depasse le pas de temps critique,x
 diverge.
```

1.3.b

```
%hmax = 1.0137 diverge un peu
clf;
h = 1.0137;
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);
subplot(1,3,1)
plot(t,x_rg);
title('Runge Kutta h = 1.0137')
%hmin = 1.0133 converge un peu
h = 1.0133;
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);
subplot(1,3,2)
plot(t,x_rg);
title('Runge Kutta h = 1.0133')
%hc = 1.0135;
h = 1.0135;
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);
subplot(1,3,3)
plot(t,x_rg);
title('Runge Kutta h = 1.0135')
%une valeur approximative du pas de temps critique
tc = hc * 2*2^0.5/W0; % tc = 0.4562
```



Published with MATLAB® R2019b