4.1 L' équation au premier ordre :

$$\begin{cases} u = \dot{q} \\ \dot{u} = -w^2 q \end{cases}$$

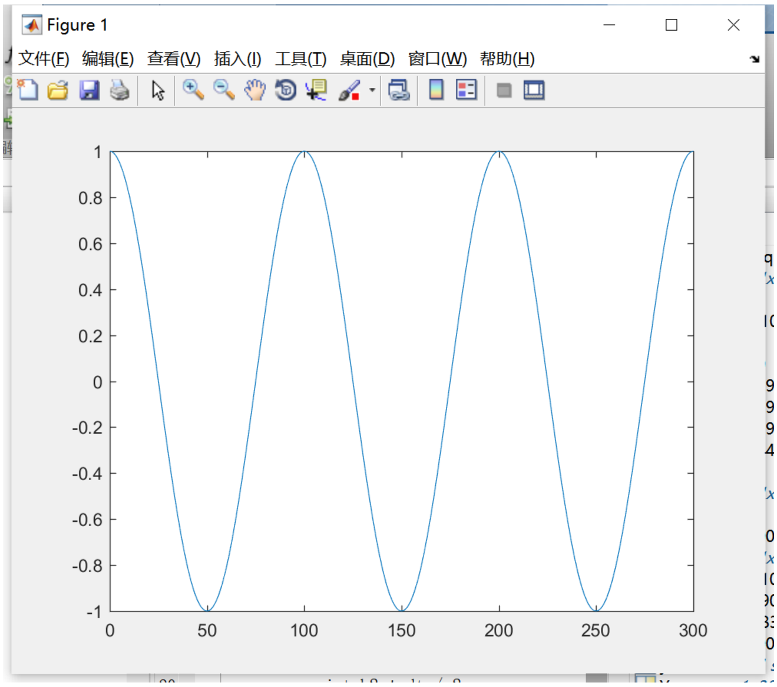
4.2 Programmer de Runge Kutta. La fonction est comme le suivant :

```
function [ dUc ] = cal_f( Uc, tc, w0c )
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
dUc = zeros(2,1);

dUc(1) = Uc(2);
dUc(2) = - (2 * pi)^2 * Uc(1);

end
```

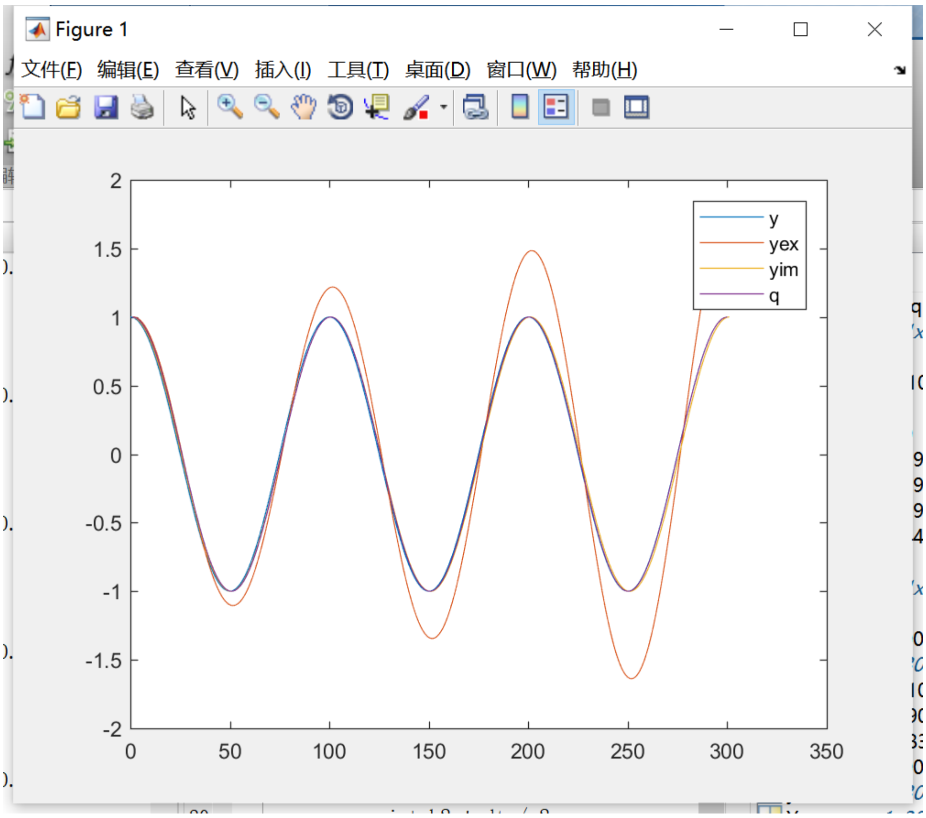
On utilise cette fonction comme le suivant :

```
q0 = 1;
dq0 = 0;
dt = 0.01;
t = (0: dt: 3)';
np = size(t, 1);
q = zeros(np, 1);
dq = zeros(np, 1);
q(1) = q0;
dq(1) = dq0;
qj = [q0; dq0];

for inc = 2: np
    tc = t(inc-1);
    xc = qj;
    k1 = cal_f(xc, tc, 0);
    xc = qj + k1 * dt / 2;
    k2 = cal_f(xc, tc + dt /2, 0);
    xc = qj + k2 * dt / 2;
    k3 = cal_f(xc, tc + dt / 2, 0);
    xc = qj + k3 * dt;
    k4 = cal_f(xc, tc + dt, 0);
    k = (k1 + 2 * k2 + 2 * k3 + k4)/6;
    qj = qj + k* dt;
    q(inc) = qj(1);
    dq(inc) = qj(2);
end
plot(t,dq)
```

4.3 La figure obtenue par RUNGE KUTTA



Je dessine la solution exacte, schéma d' EULER explicite, schéma d' EULER implicite et schéma de RUNGE KUTTA dans un même figure :



y : solution exacte ; yex : Euler explicite ; yim : Euler implicite ; q : Runge Kutta

On trouve que y, q et yim superpose bien mais il y a une grande différence entre yex et y, ça implique que c’ est une fasse solution et de plus yex diverge lentement. Et Euler implicite et Runge Kutta sont des bonnes solutions.

4.4 Les différentes valeurs de E pour les quatre schémas.

T	E explicite	E implicite	Runge Kutta	Solution exacte
0.01	64.37	19.78	19.74	19.7392

On trouve que pour la valeur de E, Runge Kutta est le plus proche de la valeur exacte. Donc c’ est la schéma solution parmi les trois.

Code de Matlab

```
q0 = 1;
dq0 = 0;
dt = 0.01;
t = (0: dt: 3)';
np = size(t, 1);
q = zeros(np, 1);
dq = zeros(np, 1);
q(1) = q0;
dq(1) = dq0;
qj = [q0; dq0];

for inc = 2: np
    tc = t(inc-1);
    xc = qj;
    k1 = cal_f(xc, tc, 0);
    xc = qj + k1 * dt / 2;
```

```
    k2 = cal_f(xc, tc + dt /2, 0);
    xc = qj + k2 * dt / 2;
    k3 = cal_f(xc, tc + dt / 2, 0);
    xc = qj + k3 * dt;
    k4 = cal_f(xc, tc + dt, 0);
    k = (k1 + 2 * k2 + 2 * k3 + k4)/6;
    qj = qj + k* dt;
    q(inc) = qj(1);
    dq(inc) = qj(2);
end
plot(t,dq)

%plot(t, q)
qq=diff(q)
ddq=diff(dq)
E=1/2*(dq*ddq+2^2*pi^2*q*qq)
```