```
'''
Created on 14 Oct 2014

@author: bob
'''
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
import function.function as func
import system.waveSystem as wave
import integrators.rungeKutta as rK
from math import ceil
from scipy.io.matlab.mio5_utils import scipy

def f(y,t):
    return vgl.f(t, y)

wsqr = 0.1
sigma = 1.
Ksqr = 1.
g = 1.


# create the ODE
vgl = wave.WaveSystem(func.P(Ksqr,sigma,g,wsqr),func.Q(Ksqr,sigma,g,wsqr))

# initial condition
y0 = [0.,1.]
t0 = 0
tend = 1.
h = 0.01
NbSteps = ceil((tend-t0)/h)
t_scipy  = np.linspace(t0, tend, NbSteps+1)

# solve the ODE using the integrated solver
soln_scipy = odeint(f, y0, t_scipy)
solution_scipy = soln_scipy[:, 0]


# solve the ODE using the self written runge kutta integrator
fe = rK.RungeKutta(vgl)
t_runge,soln_runge = fe.integrate(y0,t0,tend,h)
solution_runge = [soln_runge[i][0] for i in range(len(soln_runge))]

# plot results
plt.subplot(211)
plt.plot(t_runge,solution_runge)
plt.plot(t_scipy,solution_scipy)
```

```
49
50 # plot the error of the scipy method and the self implemented method
51 error = [scipy.absolute(solution_runge[i]-solution_scipy[i]) for i in rang
52 plt.subplot(212)
53 plt.plot(t_runge,error)
54
55
56 plt.show()
```