Worksheet for ODE in Python - Ma121

File odeUtils for numerical methods

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Feb 9 10:43:39 2022
@author: H. El-Otmany
@ This file contain all numerical method  for ODE
@ Input data :
    - dydx = f(x,y),
    - start value x_0 = a,
    - end value x_f = b,
    - number of subdivison n
    - Initial condition y0
@ Output : numerical solution y
@Algorithme d Euler (Runge Kutta d'ordre un)
Euler(f , y0, x_0, x_f , N):
    h <--- (x_f - x_0)/N
    Ly <--- y_0, Lx <---- x_0
Pour k de 1 a N faire
    y0 <---y0 + h.f(t0, y0)
    t0 <---t0 + h
    Ly <---Ly, y0; # stocker les solutions
    Lx <---Lx, t0 # stocker les abscisses
retourner Ly; liste des ordonnees y_k, k = 0; 1;... ; N
          Lx; liste des ordonnees x_k, k = 0; 1;... ; N
"""

import numpy as np
import matplotlib.pyplot as plt


#Euler method for iniline function
def ode_EulerExp(f, a, b, y0, N):
    h = (b-a) / N #step size if h is constant
    Lx = [a]  #Time list
    Ly = [y0] #Initial condition of velocity dy/dx
    x = a
    y = y0
    for i in range(1,N+1):
        #if h isn't constant, we use h=x[i+1]-x[i]
        y += h*f(x, y)
        x += h
        Lx.append(x)
        Ly.append(y)
    return (Lx, Ly)

#Euler method for ODE second order
def EulerSystem(f, a, b, y0, z0, N):
    h = (b-a)/N    #step size if h is constant
    x = a
    y = y0
    z = z0
    Lx = [x]
    Ly = [y]
    Lz = [z]
    for i in range(N):
        #if h isn't constant, we use h=x[i+1]-x[i]
        vf = f(x,y,z)
        y = y+h*z
        z = z+h*vf
        x = x+h
        Lx.append(x)
        Ly.append(y)
        Lz.append(z)
    return (Lx,Ly,Lz)

#Euler method for vector functions F(x,Y) with Y= (y1,y2,....,yn)
#Ok for inline function F(x,y) = a g(y) + b k(x) for example.
def ode_VectEulerExp(f, a, b, ic, N):
      h = (b - a) / N #step size if h is constant
      Lx = np.linspace(a, b, N)
      Ly = np.empty((N, np.size(ic)), dtype = float)
      Ly[0,:] = ic
      for i in range(N-1):
          #if h isn't constant, we use h=x[i+1]-x[i]
          Ly[i+1,:] = Ly[i,:] + h*f(Lx[i],Ly[i,:])
      return (Lx, Ly)
```

File testCase for testing functions

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Feb  8 21:15:05 2022
@ This file contain all test cases
@author: H. El-Otmany
@ This file contain all functions used in TP2
"""
import numpy as np
from math import exp,sin,pi


"""Section 1 - Question 3"""
#Example 1: definition of function for ODE: y'=y; y(0) =1; [a;b]=[0;1]
def f1Q3(x,y):
    return y
def exac1Q3(x):
    return exp(x)
#Example 2: definition of function for ODE: y' + y=x^2 +1; y(0) = 0; [a;b]=[0;5]
def f2Q3(x,y):
    return x*x+1-y
def exac2Q3(x):
    return x*x-2*x+3-3*exp(-x)
#Example 3: definition of function for ODE: (x+1)y'-xy+1=0; y(0) = 2; [a;b]=[0;5]
def f3Q3(x,y):
    return (x*y-1)/(x+1)
def exac3Q3(x):
    return (exp(x)+1)/(x+1)

"""Section 1 - Question 4"""
#Example 1: definition of function for ODE: y'=-xy+1; y(0) =0; [a;b]=[0;5]
def f1Q4(x,y):
    return -x*y + 1

#Example 2: definition of function for ODE: y' = x^2 + y^2; y(0) = 0; [a;b]=[0;1]
def f2Q4(x,y):
    return x*x + y*y

#Example 3: definition of function for ODE: y'=sin(x)sin(y); y(0) = pi/2; [a;b]=[0;10]
def f3Q4(x,y):
    return sin(x)*sin(y)

#Example 4: definition of function for ODE: (x+y)y'=1; y(0) = 5; [a;b]=[0;10]
```

```python
def f4Q4(x,y):
    return 1/(x+y)


"""Section 2 - Question 2"""
#Example 1: definition of function for y"+y=0; y(0) = 0; y'(0)=1; [a;b]=[0;5]
def F1Q2(x,y,z):
    return -y
#Example 1: definition of function for ODE: y"+y=0; y(0) = 0; y'(0)=1; [a;b]=[0;5]
#Y =(y,y'), compute Y' =(y',y") = (y', -y)
# This function is used for ode_VectEulerExp method
def F1Q2vect(x,y):
    [z, dz] = y
    return np.array([dz, -z])

def exac1Q2(x):
    return sin(x)

#Example 2: definition of function for ODE: y"-3y'+2y=x^2; y(0) = 0;y'(0)=0 [a;b]=[0;2]
def F2Q2(x,y,z):
    return x*x+3*z-2*y

def exac2Q2(x):
    return x*x/2+3*x/2+7/4-2*exp(x)+1/4*exp(2*x)

"""Section 2 - Question 3"""
#Example 1: definition of function for ODE: y"+sin(y)=0; y(0) = 3;y'(0)=0 [a;b]=[0;20]
def F1Q3(x,y,z):
    return -sin(y)
#Example 2: definition of function for ODE: y"+xy=0; y(0) = 1;y'(0)=0 [a;b]=[0;5]
def F2Q3(x,y,z):
    return -sin(y)
```

File mainProg to run for solving ODE with different methods

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Jan 19 18:43:39 2022
@author: Hammou El-Otmany
@ file used for testing
"""
import sys
from math import *
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pylab import *

#Import functions defined on utils.py and testCase.py
from ode_utils import *
from testCase import *

#Section 1 - Question 3
#ODE: y'=y; y(0) = 0;[a;b]=[0;2]
# fi is given by f(x,y) = y = F1Q3, see testCase.py
#define variables (a,b,N) and initial conditions ic=y0
a = 0
b = 2
ic = 1
# Question 1 - Exemple 1
def example1_Q3():
    for N in [10, 100, 1000]:
        plt.figure(N+1)
        ###You can also use T,Y = ode_EulerExp(f1, a, b, ic, N)
        X,Y = ode_EulerExp(f1Q3, a, b, ic, N)
        plt.plot(X,Y,label = 'Explicit Euler with N='+ str(N))
        if exac1Q3:
            Y  = [exac1Q3(x) for x in X]
            plt.plot(X,Y, label = 'Theoretical solution')
        plt.xlabel('x')
        plt.ylabel('y and dy/dx')
        plt.title('Exact & numerical solution, ODE: y'= y on [0; 2]')
        plt.legend()
        plt.grid()
example1_Q3()

#Section 2 - Question 2
#Data for EulerSystem method, ODE: y"+y=0; y(0) = 0; y'(0)=1; [a;b]=[0;5]
##f is given by f(x,y,z) = F1Q2, see testCase.py
#define variables (a,b,N) and initial conditions (y(0),y'(0))
a = 0
b = 5
y0 = 0
z0 = 1
def EulerSystem_Q2():
    for N in [10,100,1000]:
        plt.figure(N+2)
        X,Y,Z = EulerSystem(F1Q2, a, b, y0, z0, N)
        plt.plot(X,Y,label='Numerical solution '+str(N))
        if exac1Q2:
            Y = [exac1Q2(x) for x in X]
            plt.plot(X,Y, label = 'Theoretical solution')
        plt.xlabel('x')
        plt.ylabel('y and dy/dx')
        plt.title('Exact & numerical solution, ODE: y"+y=0 on [0; 5]')
        plt.legend()
        plt.grid()
EulerSystem_Q2()

#Data for ode_vectEulerExp method, ODE: y"+y=0; y(0) = 0; y'(0)=1; [a;b]=[0;5]
#Hf is given by F(x, (y1,y2)) = F1Q2vect, see testCase.py
#define variables (a,b,N) and initial conditions ic=(y(0),y'(0))
a = 0
b = 5
ic = np.array([0,1])
def example2_Q2():
    for N in [10, 100, 1000]:
        plt.figure(N+3)
        ###Explicit Euler
        X,Y = ode_VectEulerExp(F1Q2vect, a, b, ic, N)
        y, dy = Y[:,0], Y[:,1]
        plt.plot(X, y,label = 'Explicit Euler: solution y(x) with N='+ str(N))
        if exac1Q2:
            Y = [exac1Q2(x) for x in X]
            plt.plot(X,Y, label = 'Theoretical solution')
        plt.xlabel('x')
        plt.ylabel('y and dy/dx')
        plt.title('Exact & numerical solution, ODE: y"+y=0 on [0; 5]')
        plt.legend()
        plt.grid()
example2_Q2()
```