Homework 8

1 Question 45

2 Question 46

3 Question 47

4 Question 48

5 Question 49

```
import numpy as np
from math import *
import matplotlib.pyplot as plt
import hw7, hw6
from hw3 import ridder

def graph(X,Y,title,xlab='x',ylab='y',clf=True):
    if clf:
        plt.clf()
    if len(Y.shape)>1:
        for i in range(Y.shape[1]):
            plt.subplot(1,Y.shape[1],i+1)
            plt.plot(X,Y[:,i], '-',linewidth=3)
            plt.title(title+" "+ylab+"("+xlab+")")
            plt.grid(True)
            plt.xlabel(xlab); plt.ylabel(ylab); ylab = ylab +'\''
            plt.xlim(min(X),max(X))
    else:
        plt.plot(X,Y, '-',linewidth=3)
        plt.title(title+" "+ylab+"("+xlab+")")
        plt.grid(True)
        plt.xlabel(xlab); plt.ylabel(ylab); ylab = ylab +'\''
        plt.xlim(min(X),max(X))


    plt.gcf().set_size_inches(18.5, 10.5)
    plt.savefig("figures/"+title+'.png',bbox_inches='tight')

#P45: Page 305 Problem 8
#Hint: See the predefined functions in "module run_kut5" provided in the boo
#Also, look at Example 8.1 to have an idea.

def P45():
    def initCond(u):
        return np.array([0.0,u])
    def r(u):
        X,Y = hw7.integrate(F,xStart,initCond(u),xStop,h,tol=tol)
        y = Y[len(Y)-1]
        r = y[0] - 1.0
        return r
    def F(x,y):
        F = np.zeros(2)#Y=[y,y',y'']
        F[0] = y[1]
        F[1] = -1*(1-0.2*x)*(y[0]**2)
```

```
        return F
    xStart = 0.0
    xStop = pi/2
    u1 = 0.77
    u2 = 0.8
    h = 0.01
    tol=1e-3
    u = ridder(r,u1,u2)
    print(u)
    X,Y = hw7.integrate(F,xStart,initCond(u),xStop,h,tol=tol)
    graph(X,Y,"P45")

#P46: Page 305 Problem 9
#Hint: See the predefined functions in "module run_kut5" provided in the boo
#Also, look at Example 8.1 to have an idea.
def P46():
    def initCond(u):
        return np.array([0.0,u])
    def r(u):
        X,Y = hw7.integrate(F,xStart,initCond(u),xStop,h,tol=tol)
        y = Y[len(Y)-1]
        r = y[0] - -1.0
        return r
    def F(x,y):
        F = np.zeros(2)#Y=[y,y',y'']
        F[0] = y[1]
        F[1] = -2*y[1]-3*y[0]**2
        return F
    xStart = 0.0
    xStop = 2
    u1 = -1.0
    u2 = -0.95
    h = 0.01
    tol=1e-3
    u = ridder(r,u1,u2)
    print(u)
    X,Y = hw7.integrate(F,xStart,initCond(u),xStop,h,tol=tol)
    graph(X,Y,"P46")

#P47: Page 305 Problem 12
#Use beta=15 as the infinite value (no need to repeat the computation for bo
#Hint: See the predefined functions in "module run_kut5" provided in the boo
#Also, look at Example 8.1 to have an idea.
def P47():
    def initCond(u):
```

```
        return np.array([1.0,u])
    def r(u):
        X,Y = hw7.integrate(F,xStart,initCond(u),xStop,h,tol=tol)
        y = Y[len(Y)-1]
        r = y[0] - 0.0
        return r
    def F(x,y):
        F = np.zeros(2)#Y=[y,y']
        F[0] = y[1]
        F[1] = (1-np.exp(-1*x))*y[0]
        return F
    xStart = 0.0
    xStop = 15.0
    u1 = -0.5
    u2 = -0.6
    h = 0.001
    tol = 1e-6
    #u = ridder(r,u1,u2,tol=tol)
    u=-0.61
    print(u)
    X,Y = hw7.integrate(F,xStart,initCond(u),xStop,h,tol=tol)
    graph(X,Y,"P47")


#P48: Page 317 Problem 7
#Use m=10 as the number of mesh spaces. You do not need to plot.
#Calculate y for x = 0., 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0.
#Hint: See the predefined functions in "module LUdecomp3" provided in the bo
#Also, look at Example 8.6 to have an idea (don't print values of x and y u
#for loop like the example, just return arrays x and y).
def LUdecomp3(c,d,e):
  n = len(d)
  for k in range(1,n):
    lam = c[k-1]/d[k-1]
    d[k] -= lam*e[k-1]
    c[k-1] = lam
  return c,d,e
def LUsolve3(c,d,e,b):
  n = len(d)
  for k in range(1,n):
    b[k] -= c[k-1]*b[k-1]
  b[n-1] = b[n-1]/d[n-1]
  for k in range(n-2, -1, -1):
    b[k] = (b[k]-e[k]*b[k+1])/d[k]
  return b
```

```
def P48():
    def equations(x,h,m):
        d = np.ones(m+1)*(-2+h**2)
        c = np.ones(m)*(1-h)
        e = np.ones(m)*(1+h)
        b = np.zeros(m+1)
        d[0] = 1.0
        e[0] = 0.0
        b[m] = 1.0
        d[m] = 1.0
        c[m-1] = 0.0
        return c,d,e,b

    xStart = 0.0
    xStop = 1.0
    m = 10
    h = (xStop - xStart)/m
    x = np.arange(xStart,xStop+h,h)
    c,d,e,b = equations(x,h,m)
    c,d,e = LUdecomp3(c,d,e)
    y = LUsolve3(c,d,e,b)
    plt.clf()
    plt.plot(x,x*np.exp(1-x), '-',linewidth=7)
    graph(x,y,"P48",clf=False)

#P49: Page 319 Problem 15
#You do not need to plot. Use m=1000 as the number of mesh spaces.
#Use your program to list y for x = 0., 0.001, 0.002, 0.998, 0.999, 1.0.
#Hint: See the predefined functions in "module LUdecomp5" provided in the bo

def LUdecomp5(d,e,f):
    n = len(d)
    for k in range(n-2):
        lam = e[k]/d[k]
        d[k+1] = d[k+1] - lam*e[k]
        e[k+1] = e[k+1] - lam*f[k]
        e[k] = lam
        lam = f[k]/d[k]
        d[k+2] = d[k+2] - lam*f[k]
        f[k] = lam
    lam = e[n-2]/d[n-2]
    d[n-1] = d[n-1] - lam*e[n-2]
    e[n-2] = lam
    return d,e,f
```

```
def LUsolve5(d,e,f,b):
    n = len(d)
    b[1] = b[1] - e[0]*b[0]
    for k in range(2,n):
        b[k] = b[k] - e[k-1]*b[k-1] - f[k-2]*b[k-2]
    b[n-1] = b[n-1]/d[n-1]
    b[n-2] = b[n-2]/d[n-2] - e[n-2]*b[n-1]
    for k in range(n-3,-1,-1):
        b[k] = b[k]/d[k] - e[k]*b[k+1] - f[k]*b[k+2]
    return b

def P49():
    def equations(x,h,m):
        d = np.ones(m+1)*(6+gamma)
        e = np.ones(m)*(-4)
        f = np.ones(m-1)
        b = np.ones(m+1)*(h**4)
        #TODO Make sure this is right/finish
        d[0] = 1.0
        b[0] = 0.0
        e[0] = 0.0
        f[0] = 0.0

        c[1] = 7
        d[1] = -4


        b[m] = 0.0
        d[m] = 1.0
        c[m-1] = 0.0
        f[m-1] = 0.0


        return c,d,e,b

    xStart = 0.0
    xStop = 1.0
    m = 1000
    h = (xStop - xStart)/m
    x = np.arange(xStart,xStop+h,h)
    c,d,e,b = equations(x,h,m)
    c,d,e = LUdecomp3(c,d,e)
    y = LUsolve3(c,d,e,b)
    graph(x,y,"P49")
```

```
if __name__ == "__main__":
    #P45()
    #P46()
    #P47()
    P48()
    P49()
```