```
"""
Author   : Abraham Flores
File     : p231.py
Language : Python 3.6
Created  : 3/30/2018
Edited   : 3/30/2018

San Digeo State University
MTH 693b : Computational Partial Differential Equations

Strikwerda 2.3.1 : Instability

One Way Wave Equation
    U_t + U_x = 0
    x = [-1,3]
    t = [0,1]

             |  1-|x|  for |x|<= 1
    u_0(x) = |
             |    0       else

    FTFS
    h = 1/10
    lambda = 0.8

    Boundaries:
        u(t,-1) = 0
        v{n+1,M} = v{n+1,M-1}

    show that the instability grows by ~|g(pi)| per time step.

"""
import os,glob
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

#Generates intial value function
def intial_foo(x):
    if abs(x) <= 1:
        return 1-abs(x)
    return 0

def best_fit(X, Y):

    xbar = sum(X)/len(X)
    ybar = sum(Y)/len(Y)
    n = len(X) # or len(Y)

    numer = sum([xi*yi for xi,yi in zip(X, Y)]) - n * xbar * ybar
    denum = sum([xi**2 for xi in X]) - n * xbar**2

    b = numer / denum
    a = ybar - b * xbar

    return a, b


def plot(x,U,bounds,time,title,fileLoc):
    sns.set(font_scale = 2)
    sns.set_style("darkgrid", {"axes.facecolor": ".9"})
    fig,ax = plt.subplots()
    fig.set_size_inches(14.4,9)
    plt.plot(x,U,linewidth=3.0,label="t = "+ str(round(time,3)),color="r")
    plt.axis(bounds)
    plt.xlabel('x (Spatial)')
    plt.ylabel('U(t,x)')
    plt.title(title)

    plt.legend()
    plt.savefig(fileLoc+".png")
    plt.close()
```

```
def amp_plot(ratios,h,lamb):
    g_pi = 1+2*lamb
    x = np.arange(0,len(ratios),1)
    sns.set(font_scale = 2)
    sns.set_style("darkgrid", {"axes.facecolor": ".9"})
    fig,ax = plt.subplots()
    fig.set_size_inches(14.4,9)
    plt.scatter(x,ratios,marker='.',color="r")
    plt.xlim(0, len(ratios)+1)
    plt.ylim(0,round(max(ratios)))
    plt.xlabel('Time Step')
    plt.ylabel(r'$Log_{10}[L_{2}(v^{n})/L_{2}(u_{0})]$')
    plt.title(r"Strikwerda: 2.3.1 : FTFS : $g(\pi)$ = "+str(g_pi))

    a, b = best_fit(x, ratios)
    yfit = [a + b * xi for xi in x]
    plt.plot(x, yfit,linewidth=3.0,color="k",label=r"$10^{SLOPE}$: "+str(round(pow(1
0,b),5)))
    plt.legend()

    plt.savefig("instablity_h_"+str(h)+"_lamb_"+str(lamb)+".png")
    plt.close()

def makeGif(gifName):
    os.chdir('Figures')
    #Create txt file for gif command
    fileList = glob.glob('*.png') #star grabs everything,
    fileList.sort()
    #writes txt file
    file = open('FileList.txt', 'w')
    for item in fileList:
        file.write("%s\n" % item)
    file.close()

    os.system('convert -delay 10 @FileList.txt ' + gifName + '.gif')
    os.system('del FileList.txt')
    os.system('del *.png')
    os.chdir('..')

def FTFS_gif(h,lamb):
    #generate array of intial values at t = 0
    x = np.arange(-1,3+h,h)
    temp = []
    for dx in x:
        temp.append(intial_foo(dx))
    next_ = np.array(temp)

    title = "Strikwerda: 2.3.1 Instability"
    bounds = [-1,3,0,1]

    steps = int(1.0/(lamb*h)) + 2
    for t in range(steps):
        time = t*lamb*h
        #plot

        str_time = '0'*(4-len(str(t)))+str(t)
        outFile = "Figures\LF" + str_time
        plot(x,next_,bounds,time,title,outFile)

        #implement Scheme
        next_ =  (1+lamb)*next_ - lamb*np.roll(next_,-1)

        #Boundary Conditions
        next_[-1] = next_[-2]
        next_[0]  = 0

    #makeGif
    makeGif("FTFS_h_"+str(h)+"_lamb_"+str(lamb))
    return 0

def FTFS(h,lamb):
    #generate array of intial values at t = 0
    x = np.arange(-1,3+h,h)
```

```python
    temp = []
    for dx in x:
        temp.append(intial_foo(dx))
    next_ = np.array(temp)

    #L2 Norm of Intial data
    L2_naught = np.sqrt(sum(next_*next_))
    L2_ratio = [1]

    steps = int(1.0/(lamb*h)) + 2
    for t in range(steps):
        #implement Scheme
        next_ =  (1+lamb)*next_ - lamb*np.roll(next_,-1)

        #Boundary Conditions
        next_[-1] = next_[-2]
        next_[0]  = 0

        #L2 norm at the current time level
        L2 = np.sqrt(sum(next_*next_))
        L2_ratio.append(L2/L2_naught)

    return L2_ratio


if __name__ == '__main__':

    FTFS_gif(1/10,0.8)

    H = [1/10,1/20,1/40]
    L = [1/10,1/2,0.8,2.5]

    for h in H:
        for l in L:
            L2 = FTFS(h,l)
            amp_plot(np.log10(L2),h,l)


"""
Report:

    We see from the instability figures that the intial time steps due not
    adhere to the normal amplifcation factor. Most likely due to the
    boundary conditions. We can negate this effect by increasing the number
    of time steps. As seen in smaller values of h. The inverse log of the
    slopes in the plots should be equal to the value of |g(pi)|. We
    see in plots with many time steps the average is extremly close to the
    exact value.

    The graph of the unstable solution simply blows up as it should with
    exponential growth in time.

"""
```