MTH 5315 NUMERICAL METHODS FOR PDE HOMEWORK 2 REPORT

Max Le ID:901223283

Thursday 8th February 2018

1 PART 1

We have the following problem:

$$u_t - u_x = 0$$
$$x \in [0, 1]$$

With prescribed initial condition and periodic boundary conditions for the scheme:

$$\frac{U_j^{k+1} - 0.5(U_{j+1}^k + U_{j-1}^k)}{\Delta t} - \frac{U_{j+1}^k - U_{j-1}^k}{2\Delta x} = 0 \tag{1}$$

1.1 For dt = 0.5dx

1.1.1 1st initial condition

$$u(x, 0) = sin(4\pi x)$$

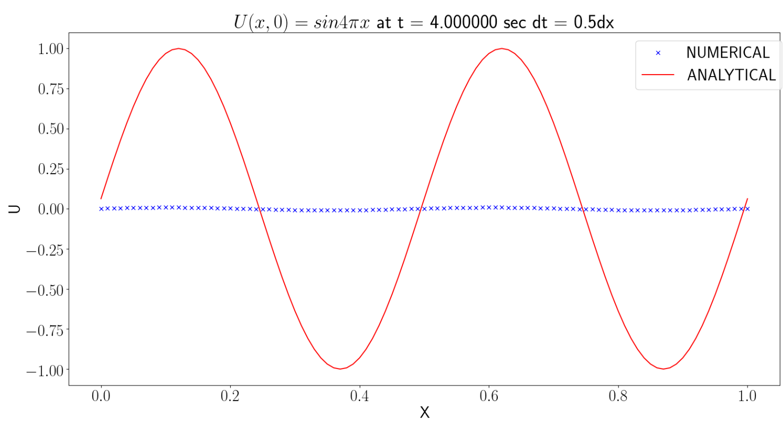

Figure 1: Numerical vs Analytical at t = 4 seconds for 1st IC, dt = 0.5dx

Observation: The numerical solution's amplitude dissipates as time goes to 4 seconds. We can see visually that the solution lost approximately 90% of its amplitude. In the code, this is computed as follow:

$$\%damping = 100 * \left(1 - \frac{max(numerical) - min(numerical)}{max(analytical) - min(analytical)}\right) \tag{2}$$

The is calculated to be around 99.13%. This makes sense because we can see that the solution loses most of its amplitudes.

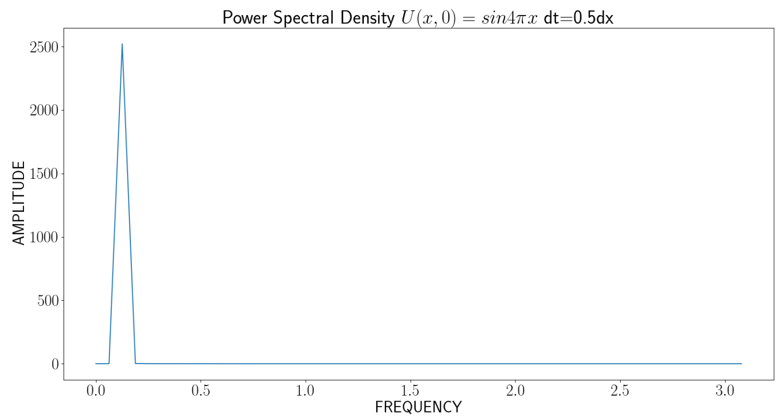

Figure 2: Power spectral density for 1st IC, dt = 0.5dx

Observation: For this power spectral density plot, we see that there is only 1 peak. This peak is the wave number, or 4π. The frequency that this occurs is, $4\pi\Delta x \approx 0.125$. In the code, this is calculated by looking at the maximum value on the power spectral density plot, or fmax in the code.

1.1.2 2nd initial condition

$$u(x,0) = sin^{10}(2\pi x)$$

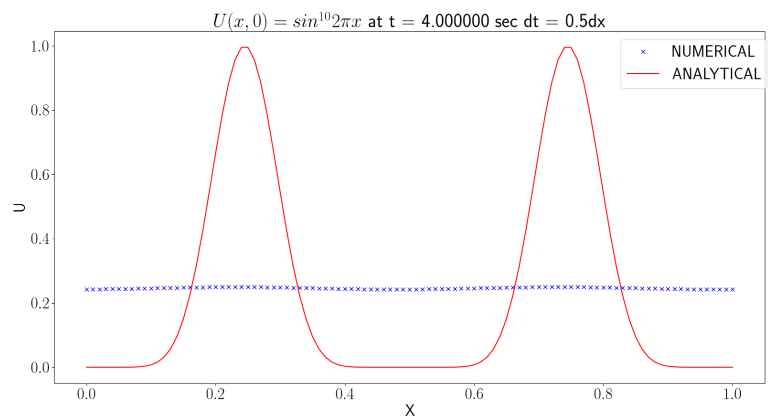

Figure 3: Numerical vs Analytical at t = 4 seconds for 2nd IC, dt = 0.5dx

Observation: This numerical solution still dissipates as time goes to 4 seconds. Although, this time the solution does not go to 0, but it fluctuates between

0.2 and 0.4. Using the similar formula to calculate the percentage damping as above, we see that the computer gives 99.28% damping

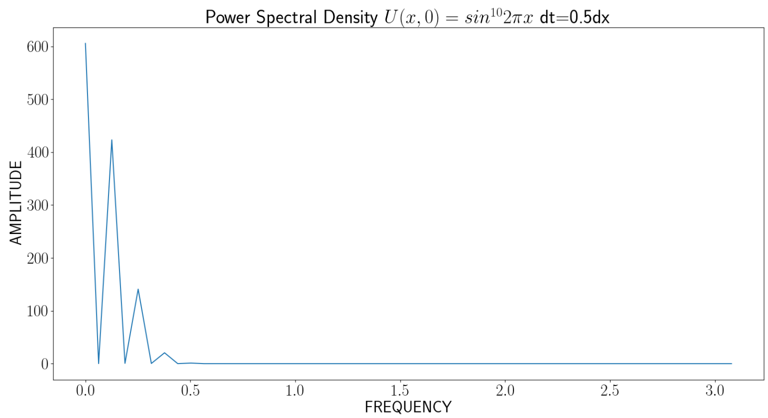

Figure 4: Power spectral density for 2nd IC, dt = 0.5dx

Observation: The power spectral density plot has multiple peaks. This could be due to numerical errors. The max peak should occur at 2π or frequency of $2\pi\Delta x \approx 0.06$. The computer gives 0 as the answer for this max peak. The reason for this could be due to numerical inaccuracy,that there is not enough points to resolve the maximum peak. Upon closer inspection, we can see that there are jagged lines in the numerical solution, which is difficult for the program to determine which one is the maximum peak.

1.2 For dt = dx

1.2.1 1st initial condition

$$u(x, 0) = sin(4\pi x)$$

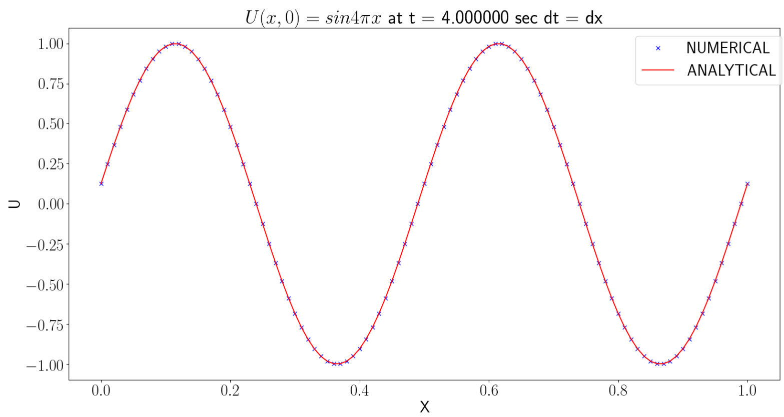

Figure 5: Numerical vs Analytical at t = 4 seconds for 1st IC, dt = dx

Observation: As dt = dx, or as we reach the maximum CFL condition, the numerical solution is exactly the analytical one and does not dissipate. Moreover, when the value of damping is computed, the code returns 0.0 as the answer. This makes sense because there are no damping.

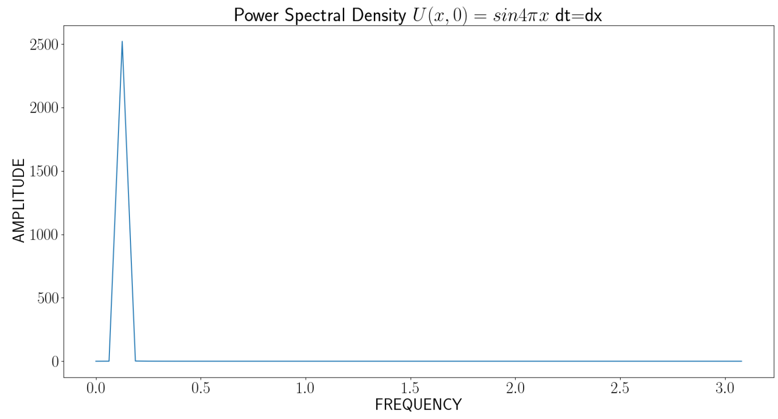

Figure 6: Power spectral density for 1st IC, dt = dx

Observation: We use the same initial condition, with only changes in the CFL condition. Therefore, the peak on the power spectral density should stay be the same, which it is. This is because the nature of the equation is the same, but the rate of convergence changes.

1.2.2 2nd initial condition

$$u(x, 0) = sin^{10}(2\pi x)$$

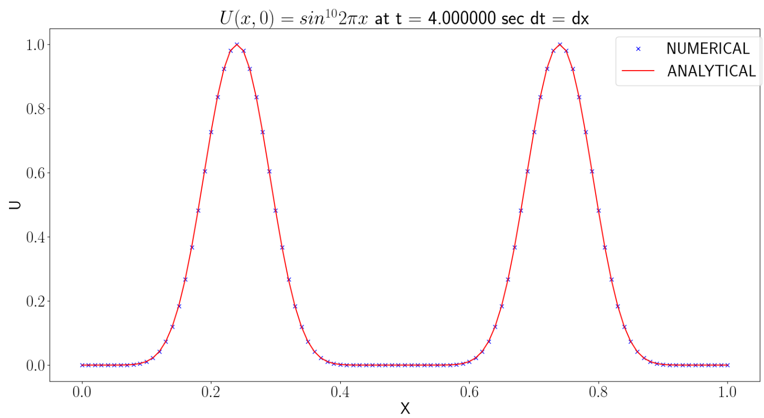

Figure 7: Numerical vs Analytical at t = 4 seconds for 2nd IC,dt = dx

Observation: Similar to the previous part, max CFL condition guarantees convergence and we can see that the numerical solution matches the analytical one perfectly. As expected, when the value for damping is computed, the computer returns 0 as the answer (no damping, exact solution)

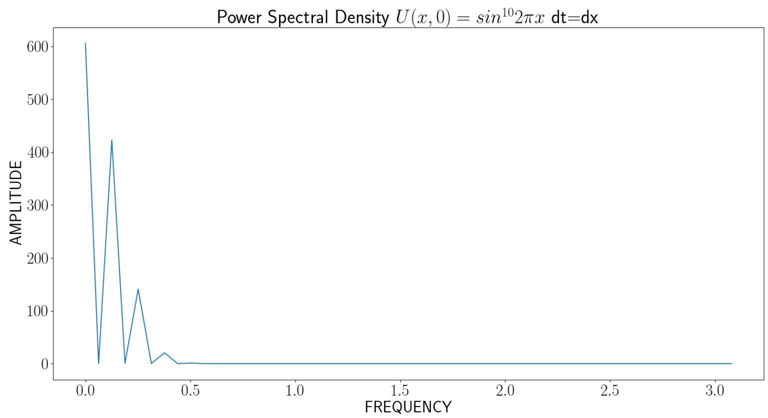

Figure 8: Power spectral density for 2nd IC, dt = dx

Observation: Again, we only change the CFL conditions, so the peak of the power spectral density should be similar to when dt = 0.5dx.

2 PART 2

We have the following problem:

$$u_t + au_x = 0$$
$$R = \frac{\Delta t}{\Delta x}$$

For the following scheme:

$$u_j^{k+1} + \frac{R}{4}(U_{j+1}^{k+1} - U_{j-1}^{k+1}) = U_j^k - \frac{R}{4}(U_{j+1}^k - U_{j-1}^k) \tag{3}$$

2.1 Analyze dissipative and dispersive qualities of the scheme

We start by substituting the following Fourier mode into our numerical scheme:

$$u_j^k = \hat{u}e^{i(\alpha+ib)k\Delta t}e^{i\beta j\Delta x} \tag{4}$$

After dividing by common $\hat{u}$ expression. We get:

$$e^{i(\alpha+ib)\Delta t} = 1 - \frac{R}{4}(e^{i\beta\Delta x} - e^{-i\beta\Delta x}) - \frac{R}{4}(e^{i(a+ib)\Delta t}e^{i\beta\Delta x} - e^{i(a+ib)\Delta t}e^{-i\beta\Delta x})$$

Letting $\beta\Delta x = \theta$. Collecting terms, isolating the exponential terms on one side and simplifying, we get:
In other words:

$$e^{i(a+ib)\Delta t} + e^{i(a+ib)\Delta t}\frac{R}{4}\left(e^{i\theta} - e^{-i\theta}\right) = 1 - \frac{R}{4}\left(e^{i\theta} - e^{-i\theta}\right)$$

$$e^{-b\Delta t}e^{i\alpha\Delta t} = \frac{1 - \frac{R}{4}(e^{i\theta} - e^{-i\theta})}{1 + \frac{R}{4}(e^{i\theta} - e^{-i\theta})}$$

For dissipation, we want the norm of the RHS for $e^{-b\Delta t}$

$$\begin{aligned}
\left|e^{-b\Delta t}\right|^2 &= \left|\frac{1 - \frac{R}{4}(e^{i\theta} - e^{-i\theta})}{1 + \frac{R}{4}(e^{i\theta} - e^{-i\theta})}\right|^2 \\
&= \left|\frac{1 - \frac{R}{2}(isin\theta)}{1 + \frac{R}{2}(isin\theta)}\right|^2 \\
&= \left|\frac{1 - \frac{R}{2}(isin\theta)}{1 + \frac{R}{2}(isin\theta)} * \frac{1 - \frac{R}{2}(isin\theta)}{1 - \frac{R}{2}(isin\theta)}\right|^2
\end{aligned}$$

Simplifying, splitting into real and imaginary parts

$$\begin{aligned}
\left|e^{-b\Delta t}\right|^2 &= \left(\frac{1-\dfrac{R^2}{4}\sin^2\theta}{1+\dfrac{R^2}{4}\sin^2\theta}\right)^2 + \left(\frac{iR\sin\theta}{1+\dfrac{R^2}{4}\sin^2\theta}\right)^2 \\
&= \frac{1}{1+\dfrac{R^2}{4}\sin\theta} * \left[1+\frac{R^2}{2}\sin^2\theta+\frac{R^4}{16}\sin^4\theta\right] \\
&= \frac{\dfrac{1}{16}\left[16+8R^2\sin^2\theta+R^4 sin^4\theta\right]}{\left(\dfrac{1}{4}(4+R^2\sin^2\theta)\right)^2} \\
&= \frac{\left(R^2\sin^2\theta+4\right)^2}{\left(R^2\sin^2\theta+4\right)^2} \\
&= 1
\end{aligned}$$

This implies that $e^{-b\Delta t}$, or the damping factor, is a constant. Or we can also say that this scheme has no dissipations, because the new amplitude is 1* old amplitude, or no change in amplitude.
For dispersive, we need:

$$\begin{aligned}
tan(\alpha\Delta t) &= \frac{Imaginary}{Real} \\
&= \frac{-R\sin\theta}{1-\dfrac{R^2}{4}sin^2\theta} \\
&= \frac{-4R\sin\theta}{4-sin^2\theta}
\end{aligned}$$

Solving for $\alpha\Delta t$, we get:

$$\alpha\Delta t = \arctan\left(\frac{-4Rsin\theta}{4-R^2\sin^2\theta}\right)$$

Recall the following Taylor series expansions, at $x=0$:

$$sin(x) = x - \frac{x^3}{6} + \frac{x^5}{120} + ...$$

$$sin^2(x) = x^2 - \frac{x^4}{3} + \frac{2x^6}{45} - ....$$

$$\arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} + ...$$

$$\frac{1}{1-ax^2} = 1 + ax^2 + a^2x^4$$

Substituting into the expression for $tan(\alpha\Delta t)$, neglecting higher order terms (bigger than $O(\Delta x^3)$)

$$
\begin{aligned}
tan(\alpha\Delta t) &= \frac{-4Rsin\theta}{4 - R^2 sin^2\theta} \\
&= -\frac{4R(\theta - \dfrac{\theta^3}{6})}{4 - R^2(\theta^2)} \\
&= \frac{4R\theta}{4 - R^2\theta^2} - \frac{4R\theta^3}{6(4 - R^2\theta^2)} \\
&= -R\theta\left(\frac{1}{1 - \dfrac{R^2}{4}\theta^2}\right) - \frac{R\theta^3}{6}\left(\frac{1}{1 - \dfrac{R^2}{4}\theta^2}\right) \\
&= -R\theta\left(1 + \frac{R^2}{4}\theta^2\right) - \frac{R\theta^3}{6}\left(1 + \frac{R^2}{4}\theta^2\right) \\
&= -R\theta + \frac{R^3\theta^3}{4} - \frac{R\theta^3}{6} + \text{Drop because higher order term} \\
&= -R\theta + \frac{3R^3\theta^3}{12} - \frac{2R\theta^2}{12} \\
&= -R\theta + \frac{\theta^3(3R^3 - 2R)}{12} \\
&= -R\theta\left(1 + \frac{\theta^2(3R^2 - 2)}{R}\right)
\end{aligned}
$$

Now we substitute the series for $arctan(x)$:

$$
\begin{aligned}
\alpha\Delta t &= -R\theta\left(1 + \frac{\theta^2(3R^2 - 2)}{R}\right) - \frac{R^3\theta^3}{3}\left(1 + \frac{\theta^2(3R^2 - 2)}{R}\right)^3 \\
&= R\theta\left(1 + \frac{\theta^2(3R^2 - 2)}{R}\right) - \frac{R^2\theta^2}{3} + \text{Drop higher order terms} \\
&= -R\theta\left(1 + \frac{\theta^2(3R^2 - 2)}{12} - \frac{R^2\theta^2}{3}\right) \\
&= R\theta\left(1 + \theta^2\left(\frac{-4R^2 + 3R^2 - 2}{12}\right)\right) \\
&= -R\theta\left(1 - \theta^2\left(\frac{R^2 + 2}{12}\right)\right)
\end{aligned}
$$

Recall $R = \dfrac{a\Delta t}{\Delta x}$ and $\theta = \beta\Delta x$. Substituting and solving for α:

$$\alpha = -a\beta\left(1 - \theta^2\left(\frac{R^2 + 2}{12}\right)\right)$$

Recall numerical wave speed, $c = \dfrac{-\alpha}{\beta}$, we get:

$$c = -a(1 - \theta^2(R^2 + 2)\left(\frac{1}{12}\right) \tag{5}$$

2.2 Plot the errors in the speed of propagation for Fourier Modes $0 \le \beta\Delta x \le \pi$

Given R = 0.5 and a = 1.0. The following plot of the error in speed propagation is generated:

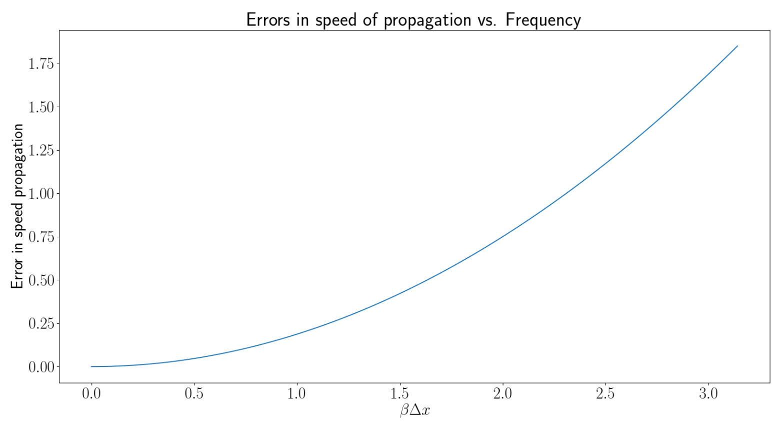

Figure 9: Error in speed of propagations for R = 0.5

Observation: It can be seen that the error in speed of propagation increases with the frequency. Mathematically, the expression for the numerical wave speed involves a term: $R^2 + 2$,unlike examples in class, where R can be treated as less than or bigger than zero. We also have 1 - this quantity;therefore, the numerical wave speed is always slower than the analytical wave speed. We can say that this scheme is absolutely dispersive.

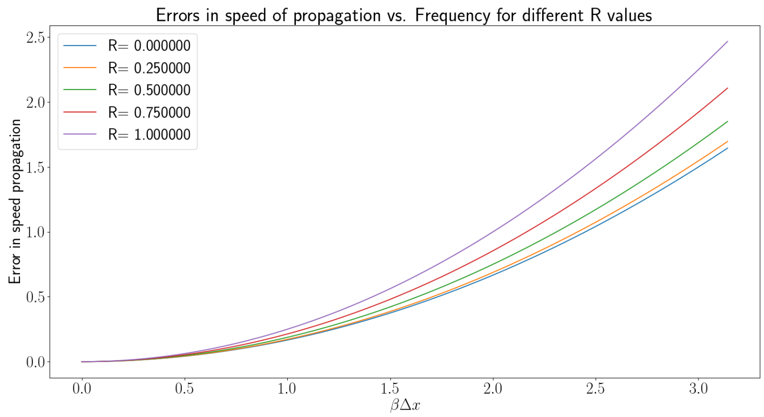

Figure 10: Error in speed of propagations for different values of R

In fact, when multiple values of R are tested, the plot shows the similar behavior: increasing error with increasing frequency.

3 PYTHON CODE

```
import numpy as np
import matplotlib.pyplot as plt
import numpy.fft as fft
import matplotlib

matplotlib.rcParams.update({'font.size':24})
plt.rc('text',usetex = True)


#DEFINE FUNCTIONS

#initial Soln
def initialSoln(x):
    return np.sin(4.*np.pi*x)
    # return np.sin(2.*np.pi*x)**10

def analyticSoln(x,t):
    return initialSoln(x + t*np.ones_like(x))

#DEFINE BASIC PROPERTIES

a = 1.
dx = 0.01
dt =0.5*dx
imax = 101
tmax = 4#seconds
tmin = 0
nmax = int((tmax-tmin)/(dt))


#EMPTY ARRAYS
x_vector = np.linspace(0.,1.,101)
u0 = np.zeros(imax)
uf = np.zeros(imax)
u_analytic = np.zeros(imax)

#CALCULATE INITIAL CONDITIONS

uf = initialSoln(x_vector)


##PLOT OF POWER SPECTRAL DENSITY

yhat = fft.fft(uf)
amplitude = np.real(yhat*np.conjugate(yhat))
freq = x_vector*(2*np.pi) #beta delta x term = f
plt.figure(1)
# plt.plot(freq[:50],amplitude[:50])

## FOR DT = 0.5DX
# plt.title('Power Spectral Density ' r'$U(x,0) = sin4\pi x $' '\
                                  tdt=0.5dx' )
# plt.title('Power Spectral Density ' r'$U(x,0) = sin^{10} 2\pi x $
                                  ' '\tdt=0.5dx')
```

```python
## FOR DT = DX
# plt.title('Power Spectral Density ' r'$U(x,0) = sin4\pi x $' '\
                                    tdt=dx' )
# plt.title('Power Spectral Density ' r'$U(x,0) = sin^{10} 2\pi x $
                                    ' '\tdt=dx')


# plt.xlabel('FREQUENCY')
# plt.ylabel('AMPLITUDE')


#MAX PEAK
fmax = freq[np.argmax(amplitude)]

print(fmax)


#START MAIN LOOP


plt.figure(1)
#time loop
for t in range(nmax+1):


    #main finite difference loop
    for i in range(1,imax-1):
        u0[i] = (dt/(2.*dx))*(uf[i+1]-uf[i-1])+ 0.5*(uf[i+1]+uf[i-1
                                            ])

    #populated analytical soln

    u_analytic =  analyticSoln(x_vector,(t+1)*dt)

    #embedding periodic BC
    u0[0] = (dt/(2.*dx))*(uf[1]-uf[imax-2])+ 0.5*(uf[1]+uf[imax-2])
    u0[imax-1] = u0[0]

    #updating uf
    uf = u0.copy()

    time = t*dt

    #quantitatively damping

    h_numerical = np.max(uf)-np.min(uf)
    h_real = np.max(u_analytic)-np.min(u_analytic)
    damping = h_numerical/h_real

    # plt.clf()
    # plt.xlabel('X')
    # plt.ylabel('U')
    # plt.plot(x_vector,uf,'xb',label='NUMERICAL')
    # plt.plot(x_vector,u_analytic,'r-',label='ANALYTICAL')
    # plt.legend(loc=9,bbox_to_anchor=(0.90,1.0))
    # plt.pause(0.001)
    # plt.draw()
```

```
    # plt.show()


plt.xlabel('X')
plt.ylabel('U')
plt.plot(x_vector,uf,'xb',label='NUMERICAL')
plt.plot(x_vector,u_analytic,'r-',label='ANALYTICAL')
plt.legend(loc=9,bbox_to_anchor=(0.90,1.0))
plt.show()



# plt.figure(2)

##FOR DT = 0.5DX
# plt.title(r'$U(x,0) = sin4\pi x $' '\tat t = %f sec \t dt = 0.5dx
                                    '   %(time))
# plt.title(r'$U(x,0) = sin^{10} 2\pi x $' '\tat t = %f sec \t dt =
                                    0.5dx'   %(time))


##FOR DT = DX
# plt.title(r'$U(x,0) = sin4\pi x $' '\tat t = %f sec \t dt = dx '
                                    %(time))
# plt.title(r'$U(x,0) = sin^{10} 2\pi x $' '\tat t = %f sec \t dt =
                                    dx'   %(time))

# plt.xlabel('X')
# plt.ylabel('U')
# plt.plot(x_vector,uf,'xb',label='NUMERICAL')
# plt.plot(x_vector,u_analytic,'r-',label='ANALYTICAL')
# plt.legend(loc=9,bbox_to_anchor=(0.90,1.0))
# plt.show()


# print(1-damping)


## PART 2.


# theta = np.linspace(0,np.pi,101)
# a = 1.
# R = 0.5

# c= a*(1-(theta**2)*(R**2 + 2.)/(12))

# error = a - c

# plt.plot(theta,error,label='R = 0.5')

# plt.title('Errors in speed of propagation vs. Frequency')
# plt.xlabel(r'$ \beta \Delta x $')
# plt.ylabel('Error in speed propagation')
```