Lab 6

Waves

**Purpose of the experiment:**

The purpose of this lab is to find the motion of a string in different situations and graph data in regards to the motion.

**Problem Description:**

In lab 5, we will be focusing our attention on the a string and will estimate the shape of the wave at different time intervals



*(Equation 1 – Central equation of a wave motion)*

The string runs along the horizontal (x) axis and y represents the vertical displacement as a function of time. Equation 1 will be written in finite difference form to create an algorithm to solve the wave equation. This is achieved by discretization of x = i \* (change in x) and t = j \* (change in t). Equation 1 will be rewritten as equation 2 below.



*(Equation 2 – Discretization and algorithm of equation 1)*

Equation 2 assumes the shape of the wave, the vertical displacement y at each point, is known at the steps j and j-1 and uses the information from those iterations to calculate time step j+1. R is defined below in equation 2.1.



*(Equation 2.1-Dimesionless variable r)*

With equation 2 and 2.1, and knowledge of the initial condition at time t = 0, the boundary conditions at the two end points of the string, we can determine the shape of the wave at any point in time.

**Code:**

In the beginning of my code for all the scripts, I imported useful functions numpy and and matplotlib.pyplot for mathematical and plotting uses. I then declared the initial values that would be used to solve for the equation 1 and 2.

c = 300

dx = 0.01

r = 1

l = 1

t = 10

dt = r\*dx/c

r = c\*dt/dx

A = 1

w = 3600

n = int(l/dx)

Y = np.zeros([n+1,3])

*(Figure 1 – These are my initializing statements. All Python scripts will have blocks of code in the beginning like this but not exactly with all the variables)*

In all scripts, c represents propagation speed of the wave in m/s. l is length of the string in meters. dx is the grid spacing on the string in meters. r is the dimensionless variable to relate dt and dx to c. dt is in seconds. t is time in seconds. dx represents the grid spacing on the string which is in meters. n is the number of grib points on the string n + 1 or -1 depending on the script. Y will store the values vertical displacement at each time. Y will be a (n+1) by 3 matrix where Y will hold the values at j-1, j, and j+1 since we need all values to calculate for j+1 based off equation 2. A is the amplitude (only relevant to “Lab 6 sine” python script). w is the period (only relevant to “Lab 6 sine” python script).

Y[0,0] = 0; Y[n,0] = 0; Y[0,1] = 0; Y[n,1] = 0; Y[0,2] = 0; Y[n,2]= 0

for j in range(2):

for i in range(n):

Y[i,j]=np.exp(-1000\*(i\*dx - 0.3)\*\*2)

Y[0,0] = 0; Y[n,0] = 0; Y[0,1] = 0; Y[n,1]= 0

for j in range(300):

for i in range(n-2):

temp\_2 = (2\*(1-(r\*\*2))) \* (Y[i+1,1])- Y[i+1,0]+ ((r\*\*2)\*(Y[i+2,1]+Y[i,1])); Y[i+1,2] = temp\_2

for i in range(n-2):

Y\_0 = Y[i+1,1]; Y\_1 = Y[i+1,2]

Y[i+1,0]= Y\_0; Y[i+1,1]= Y\_1

*(Figure 2 – Nested for loops for my calculations)*

I then defined the shape of the string at time zero and set the first row of Y to be equal to the initial displacement. It also has a loop j (time step) and calculate Y as a function of time. At each time step I would calculate the position along the string. The grid points along the string are between the interval 0 and n. To calculate the Y at every position within the string, it is mandatory to define another loop. It will also calculate the third row based off the first two rows of the matrix excluding the end points. Equation 2 will be used where j is a value of 1 and for i + 1. The code in figure 2 will represent equation 2.2 below.



*(Equation 2.2 – describes Figure 2)*

When the string has fixed end points, the displacement at the end of the points is always zero.

For a string with free end points, the displacement at the end points are equivalent to the displacements the points that are one spatial unit in from the ends. Since I can’t assign two array elements to be the same, since updating one will update both, we need a temporary auxiliary variable just like in the same way it is used in bottom of figure 2. At the bottom of figure 2, I also shift the rows of Y to calculate the future values. I then plotted the waves at several time steps.

I then created a new python file called “Lab 6 Free” where it would graph the motion of the wave in a string for which one end is fixed and another end is oscillating. The calculations and program flow was like Figure 2 the calculations differed slightly as shown in Figure 3 below

Y[0,0] = 0; Y[n,0] = 0; Y[0,1] = 0; Y[n,1] = 0; Y[0,2] = 0; Y[n,2]= 0

for j in range(2):

for i in range(n):

Y[i,j]=np.exp(-1000\*(i\*dx - 0.3)\*\*2)

Y[0,0] = 0; Y[n,0] = 0; Y[0,1] = 0; Y[n,1]= 0

for j in range(300):

for i in range(n-2):

temp\_2 = (2\*(1-(r\*\*2))) \* (Y[i+1,1])- Y[i+1,0]+ ((r\*\*2)\*(Y[i+2,1]+Y[i,1])); Y[i+1,2] = temp\_2

for i in range(n-2):

Y\_0 = Y[i+1,1]; Y\_1 = Y[i+1,2]

Y[i+1,0]= Y\_0; Y[i+1,1]= Y\_1

*(Figure 3 – Like figure 2 except the calculation in for loops and boundary conditions are slightly different since it is fixed)*

In figure 4 below, it describes the driving force like by calculating in methods similar to figure 2 and 3.

for j in range(3000):

for i in range(n):

Y[0,1]= A\*np.sin((j)\*w\*dt)

Y[0,2]= A\*np.sin((j+1)\*w\*dt)

temp\_element2 = 2\*(1-r\*\*2)\*Y[i,1]-Y[i,0]+(r\*\*2)\*(Y[i+1,1]+Y[i-1,1])

Y[i,2] = temp\_element2

for i in range(n+1):

Y0 = Y[i,1]; Y1 = Y[i,2]

Y[i,0]= Y0; Y[i,1]= Y1

(Figure 4 – Describes the driving force)

**Equations Solved & Algorithms Used:**

In this lab, I solved for equations 1 which is a 2nd order differential equation. The algorithm used for solving and plotting the equations was solved using the Gaussian pluck. We used the equation because we needed to figure out initial string profile before we did our calculations. We then used equation 2 with Gaussian pluck method to calculate future values to plot that would best describe equation 1.

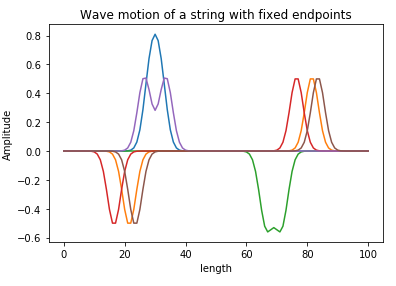
The Gaussian pluck is described below in equation 3



*(Equation 3 – Initial string profile or the Gaussian pluck)*

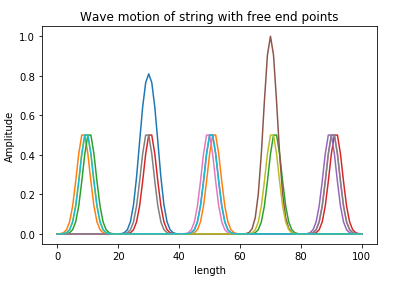
Where the displacement is centered at x\_0, and k determines the width of the Gaussian envelope.

**Results & Analysis**



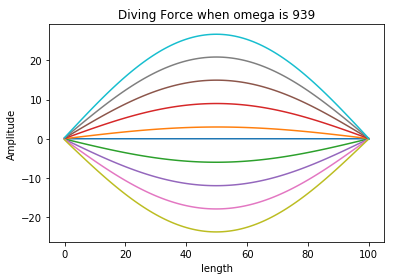
*(Figure 5 – Graph of the Wave motion of a string with fixed endpoints)*

When the end of a string is fixed, displacement of the string at the end must be zero. A transverse wave traveling along the string towards the fixed end will be reflected in the opposite direction. When the string is fixed at both ends, two waves traveling in the opposite direction simply bounce back and forth.



*(Figure 6 – Graph of the wave motion of a string with free end points)*

In figure 6, it describes the motion of a wave of a string with fixed end points. Unlike in figure 5, the motion of the wave will not be reflected but it will “bounce” back in the opposite direction. Two waves traveling in the opposite direction will eventually overlap each other and combine amplitudes.



*(Figure 7 – Driving Force)*

In figure 7, it describes the driving force and fundamental harmonic of the wave motion of the string. My value to achieve the first harmonic was 939 which is where the length of the string was 1 meter.