CALIFORNIA INSITUTE OF TECHNOLOGY

Ph20 Assignment 2

Kyriacos Xanthos

October 25, 2019

1 Introduction

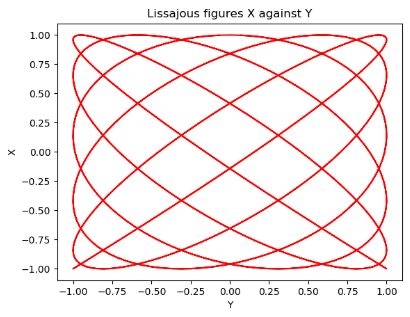
In this assignment I learned how to use version control with git, learned how to use command-line arguments and how to create a makefile. I split my code from assignment 1 to two different programs in order to create a dependency structure which was later used in the makefile.

2 Soursecode

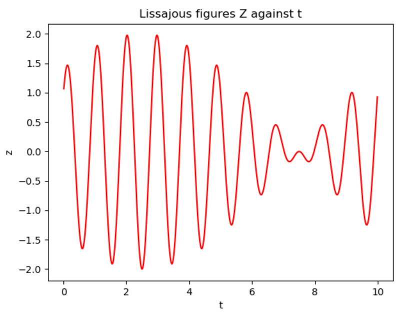
In figure 2.2 you can see the code of my first program. I imported sys in order to have command-line arguments. I assigned each input value to one command line argument, separated by a space each time. I changed all the types to floats in order to be able to use them in multiplications and math functions. Note that I export my CSV files in a folder in my computer. If one wants to run this code then a change of file directory should be made.

My second program imports the CSV files created in the first program using np.genfromtxt and uses them as arrays to plot either the X-Y plot or the Z-t plot. The dependency on which plot to produce is in the command-line arguments, where I have decided to plot the X-Y curve if the first argument is a single digit number and the Z-t curve if the argument is a double digit number. If anything else is input as an argument, the code will print "Insert correct argument".

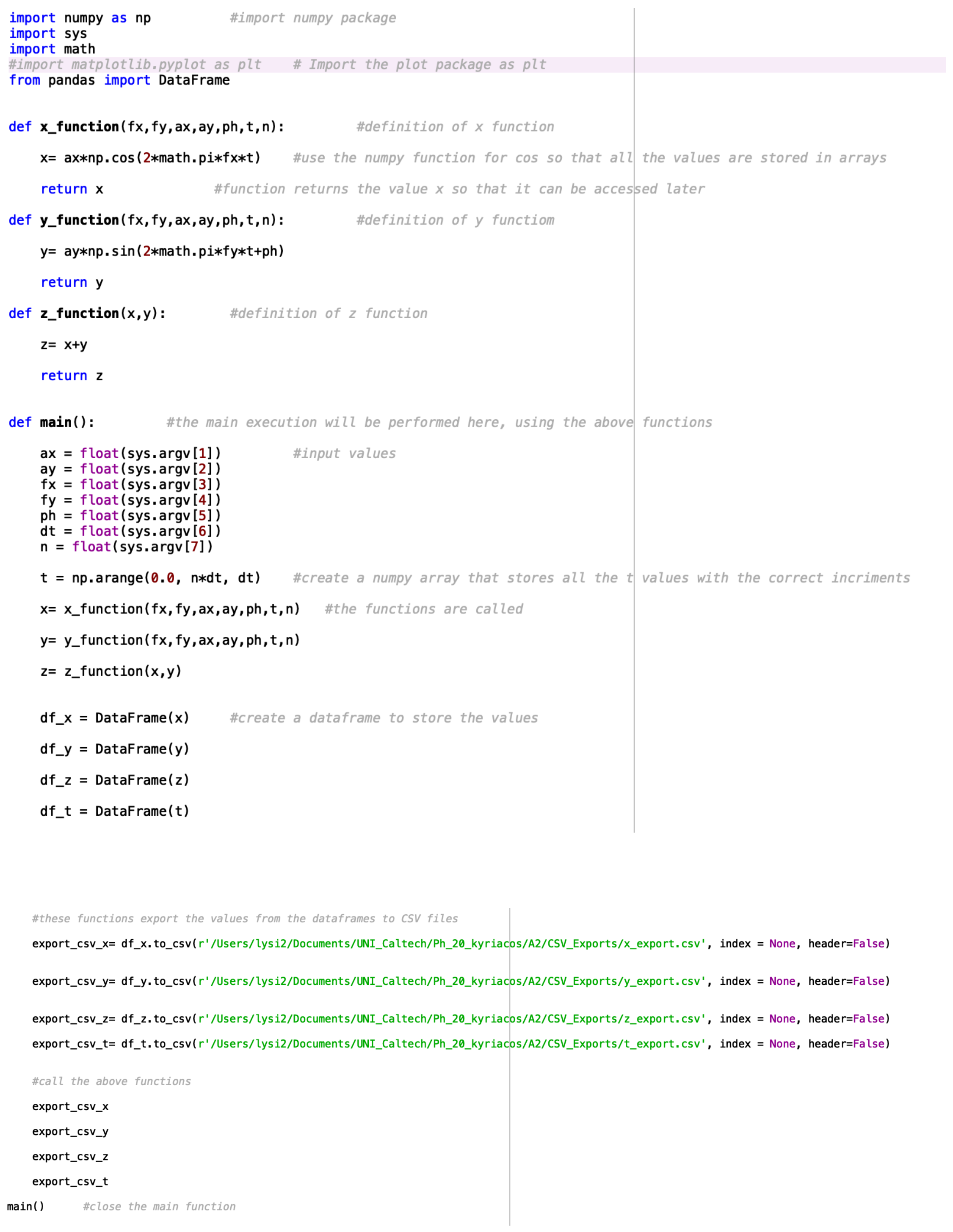
The command-line input I use is: python3 Ph20_A2.py 1 1 1 1.1 0 0.01 1000 . This runs my code in the terminal and uses the first 7 numbers as input to my code to produce lists. The first number corresponds to ax, the second to ay etc. In figure (a) and (b) below the output graphs of the above input are shown.



(a) Y against X

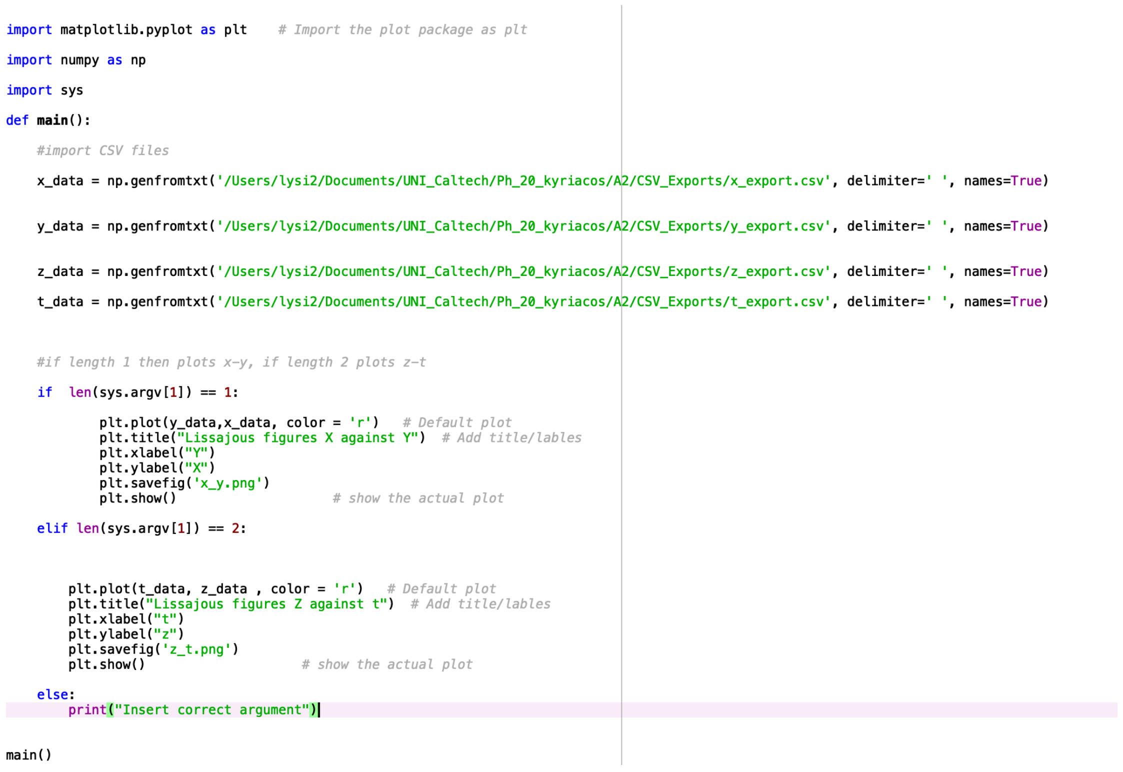


(b) Z against t



```python
import numpy as np          #import numpy package
import sys
import math
#import matplotlib.pyplot as plt    # Import the plot package as plt
from pandas import DataFrame


def x_function(fx,fy,ax,ay,ph,t,n):         #definition of x function

    x= ax*np.cos(2*math.pi*fx*t)    #use the numpy function for cos so that all the values are stored in arrays

    return x            #function returns the value x so that it can be accessed later

def y_function(fx,fy,ax,ay,ph,t,n):         #definition of y functiom

    y= ay*np.sin(2*math.pi*fy*t+ph)

    return y

def z_function(x,y):        #definition of z function

    z= x+y

    return z


def main():         #the main execution will be performed here, using the above functions

    ax = float(sys.argv[1])         #input values
    ay = float(sys.argv[2])
    fx = float(sys.argv[3])
    fy = float(sys.argv[4])
    ph = float(sys.argv[5])
    dt = float(sys.argv[6])
    n = float(sys.argv[7])

    t = np.arange(0.0, n*dt, dt)    #create a numpy array that stores all the t values with the correct incriments

    x= x_function(fx,fy,ax,ay,ph,t,n)   #the functions are called

    y= y_function(fx,fy,ax,ay,ph,t,n)

    z= z_function(x,y)


    df_x = DataFrame(x)     #create a dataframe to store the values

    df_y = DataFrame(y)

    df_z = DataFrame(z)

    df_t = DataFrame(t)



    #these functions export the values from the dataframes to CSV files

    export_csv_x= df_x.to_csv(r'/Users/lysi2/Documents/UNI_Caltech/Ph_20_kyriacos/A2/CSV_Exports/x_export.csv', index = None, header=False)

    export_csv_y= df_y.to_csv(r'/Users/lysi2/Documents/UNI_Caltech/Ph_20_kyriacos/A2/CSV_Exports/y_export.csv', index = None, header=False)

    export_csv_z= df_z.to_csv(r'/Users/lysi2/Documents/UNI_Caltech/Ph_20_kyriacos/A2/CSV_Exports/z_export.csv', index = None, header=False)
    export_csv_t= df_t.to_csv(r'/Users/lysi2/Documents/UNI_Caltech/Ph_20_kyriacos/A2/CSV_Exports/t_export.csv', index = None, header=False)

    #call the above functions

    export_csv_x

    export_csv_y

    export_csv_z

    export_csv_t

main()      #close the main function
```

Figure 2.2: Program 1

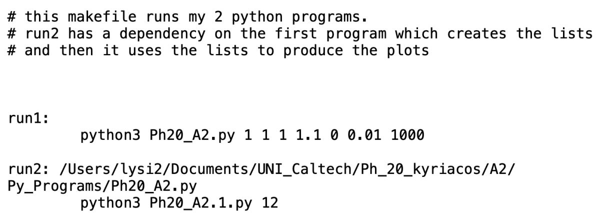


```python
import matplotlib.pyplot as plt    # Import the plot package as plt

import numpy as np

import sys

def main():

    #import CSV files

    x_data = np.genfromtxt('/Users/lysi2/Documents/UNI_Caltech/Ph_20_kyriacos/A2/CSV_Exports/x_export.csv', delimiter=' ', names=True)


    y_data = np.genfromtxt('/Users/lysi2/Documents/UNI_Caltech/Ph_20_kyriacos/A2/CSV_Exports/y_export.csv', delimiter=' ', names=True)


    z_data = np.genfromtxt('/Users/lysi2/Documents/UNI_Caltech/Ph_20_kyriacos/A2/CSV_Exports/z_export.csv', delimiter=' ', names=True)

    t_data = np.genfromtxt('/Users/lysi2/Documents/UNI_Caltech/Ph_20_kyriacos/A2/CSV_Exports/t_export.csv', delimiter=' ', names=True)


    #if length 1 then plots x-y, if length 2 plots z-t

    if  len(sys.argv[1]) == 1:

            plt.plot(y_data,x_data, color = 'r')   # Default plot
            plt.title("Lissajous figures X against Y")  # Add title/lables
            plt.xlabel("Y")
            plt.ylabel("X")
            plt.savefig('x_y.png')
            plt.show()                    # show the actual plot

    elif len(sys.argv[1]) == 2:


        plt.plot(t_data, z_data , color = 'r')   # Default plot
        plt.title("Lissajous figures Z against t")  # Add title/lables
        plt.xlabel("t")
        plt.ylabel("z")
        plt.savefig('z_t.png')
        plt.show()                    # show the actual plot

    else:
        print("Insert correct argument")


main()
```

Figure 2.3: Program 2

3 THE MAKEFILE

Below you can see a simple makefile I used to produce the whole assignment in a single make command. It first goes to run1 and runs my first program which produces the CSV files. Then it includes run2 which uses the dependency of run1 to use the CSV exports from the program and run the code. Here I used command line argument of "12" which is a double digit number so the plot produced is the Z against t. I did not use a clean function in my makefile as Python doesn't produce any object files or unnecessary .dat files.

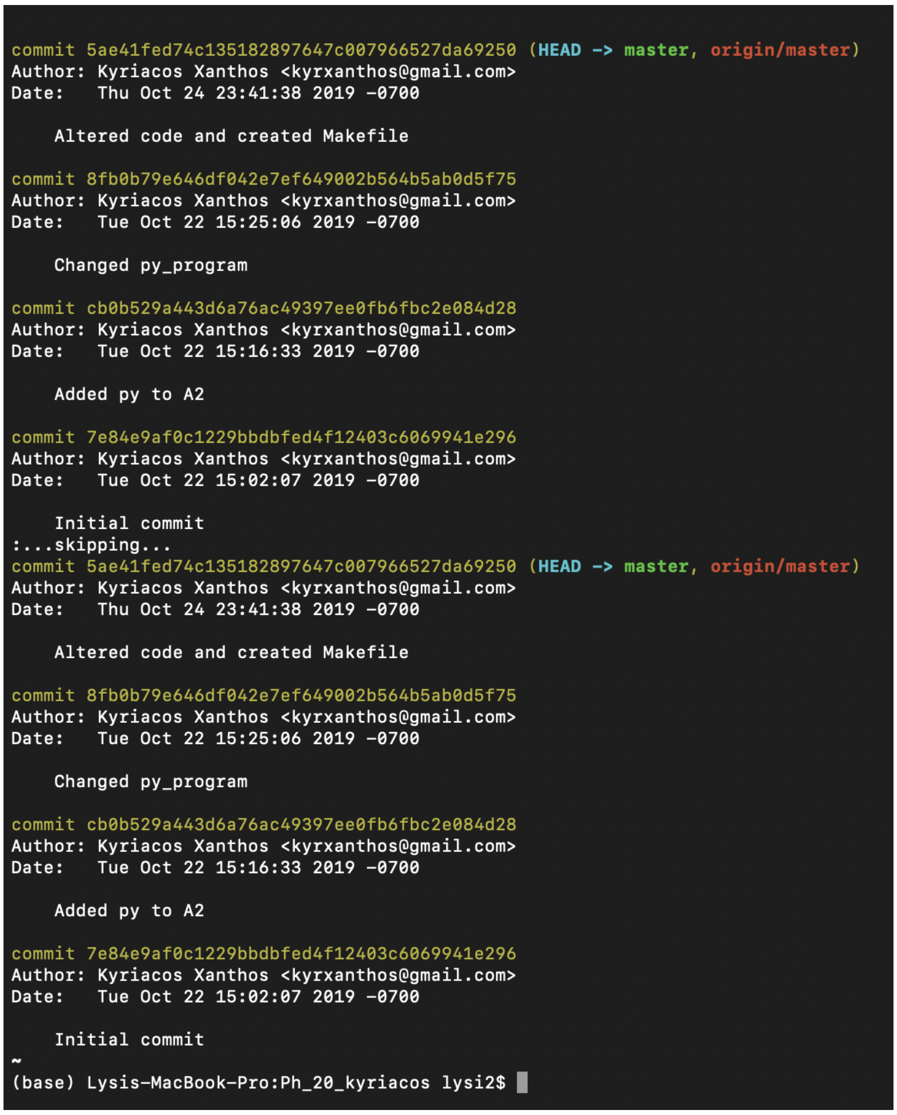


```makefile
# this makefile runs my 2 python programs.
# run2 has a dependency on the first program which creates the lists
# and then it uses the lists to produce the plots



run1:
	python3 Ph20_A2.py 1 1 1 1.1 0 0.01 1000

run2: /Users/lysi2/Documents/UNI_Caltech/Ph_20_kyriacos/A2/
Py_Programs/Ph20_A2.py
	python3 Ph20_A2.1.py 12
```

Figure 3.1: The Makefile

4 Git Log

In the figure below the git log is presented from the terminal where it logs each change that I have made to my files in the tracked folder. It includes altering the python program, making the makefile and the initial commit.



```
commit 5ae41fed74c135182897647c007966527da69250 (HEAD -> master, origin/master)
Author: Kyriacos Xanthos <kyrxanthos@gmail.com>
Date:   Thu Oct 24 23:41:38 2019 -0700

    Altered code and created Makefile

commit 8fb0b79e646df042e7ef649002b564b5ab0d5f75
Author: Kyriacos Xanthos <kyrxanthos@gmail.com>
Date:   Tue Oct 22 15:25:06 2019 -0700

    Changed py_program

commit cb0b529a443d6a76ac49397ee0fb6fbc2e084d28
Author: Kyriacos Xanthos <kyrxanthos@gmail.com>
Date:   Tue Oct 22 15:16:33 2019 -0700

    Added py to A2

commit 7e84e9af0c1229bbdbfed4f12403c6069941e296
Author: Kyriacos Xanthos <kyrxanthos@gmail.com>
Date:   Tue Oct 22 15:02:07 2019 -0700

    Initial commit
:...skipping...
commit 5ae41fed74c135182897647c007966527da69250 (HEAD -> master, origin/master)
Author: Kyriacos Xanthos <kyrxanthos@gmail.com>
Date:   Thu Oct 24 23:41:38 2019 -0700

    Altered code and created Makefile

commit 8fb0b79e646df042e7ef649002b564b5ab0d5f75
Author: Kyriacos Xanthos <kyrxanthos@gmail.com>
Date:   Tue Oct 22 15:25:06 2019 -0700

    Changed py_program

commit cb0b529a443d6a76ac49397ee0fb6fbc2e084d28
Author: Kyriacos Xanthos <kyrxanthos@gmail.com>
Date:   Tue Oct 22 15:16:33 2019 -0700

    Added py to A2

commit 7e84e9af0c1229bbdbfed4f12403c6069941e296
Author: Kyriacos Xanthos <kyrxanthos@gmail.com>
Date:   Tue Oct 22 15:02:07 2019 -0700

    Initial commit
~
(base) Lysis-MacBook-Pro:Ph_20_kyriacos lysi2$
```

Figure 4.1: The Git Log