PHYS 566 HW1

Matthew Epland
Department of Physics, Duke University, Durham, NC 27707, USA
(Dated: January 27, 2016)

Python code was written to compute $\sin(x)$ and $\sin(x)/x$ between $-10.0 \leq x \leq 10.0$ in steps of 0.05. The values were plotted, written to a text file, read from the text file, and plotted again.

I. OUTPUTS

A. Part 1

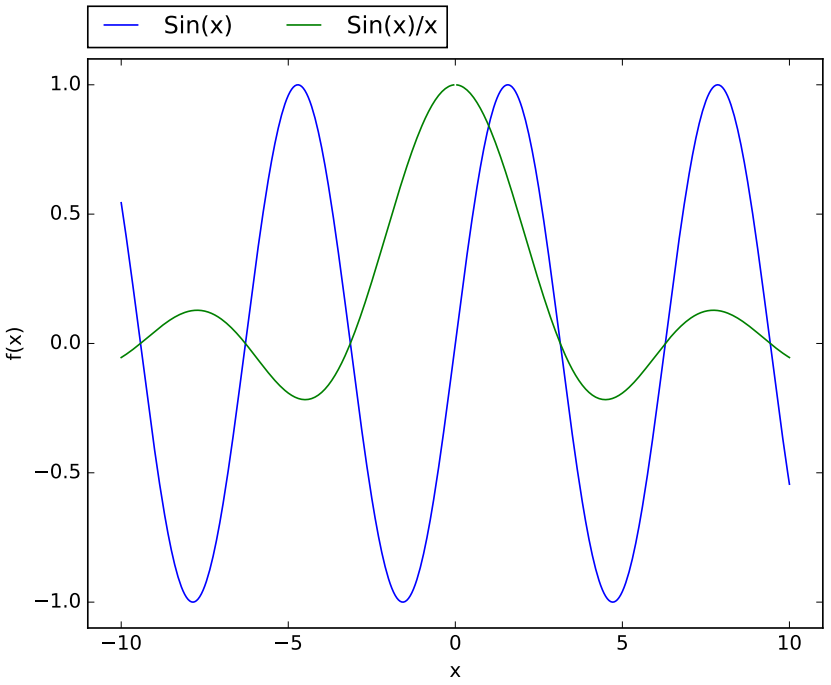

FIG. 1: Output plot of $\sin(x)$ and $\sin(x)/x$ from Part 1.

B. Part 2

Below are the first 30 lines of plain text output from part 2 in the format, x `tab` $\sin(x)$ `tab` $\sin(x)/x$. See `data.txt` for the full dataset.

−10.00	0.544021	−0.054402
−9.95	0.501405	−0.050392
−9.90	0.457536	−0.046216
−9.85	0.412523	−0.041881
−9.80	0.366479	−0.037396
−9.75	0.319519	−0.032771
−9.70	0.271761	−0.028017
−9.65	0.223323	−0.023142
−9.60	0.174327	−0.018159
−9.55	0.124895	−0.013078
−9.50	0.075151	−0.007911

−9.45	0.025219	−0.002669
−9.40	−0.024775	0.002636
−9.35	−0.074708	0.007990
−9.30	−0.124454	0.013382
−9.25	−0.173889	0.018799
−9.20	−0.222890	0.024227
−9.15	−0.271333	0.029654
−9.10	−0.319098	0.035066
−9.05	−0.366066	0.040449
−9.00	−0.412118	0.045791
−8.95	−0.457141	0.051077
−8.90	−0.501021	0.056294
−8.85	−0.543648	0.061429
−8.80	−0.584917	0.066468
−8.75	−0.624724	0.071397
−8.70	−0.662969	0.076203
−8.65	−0.699557	0.080874
−8.60	−0.734397	0.085395

C. Part 3

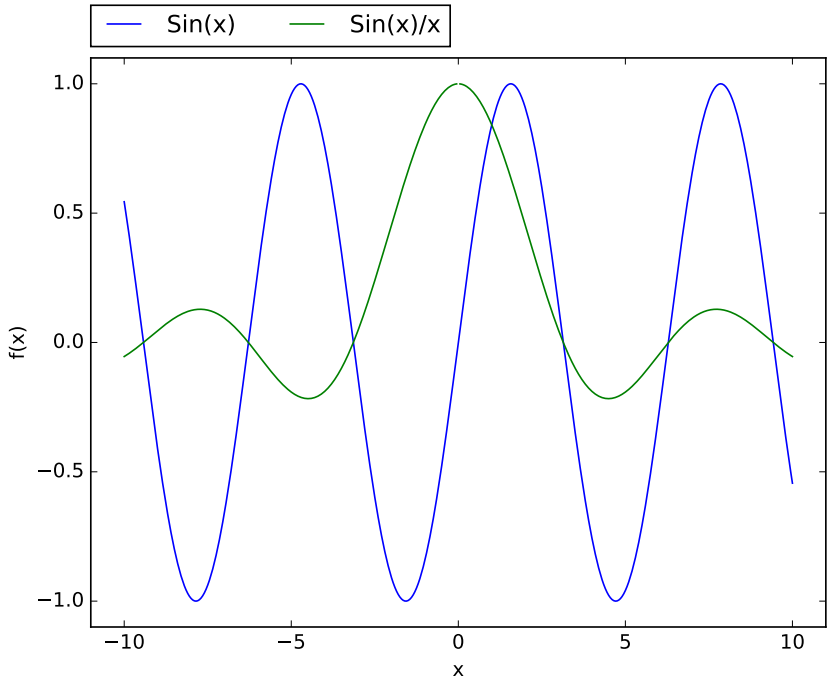

FIG. 2: Output plot of $\sin(x)$ and $\sin(x)/x$ from Part 3. Note that this plot is identical to the one from Part 1 by design.

II. CODE

The source code to make these plots can be found online at `http://github.com/mepland/PHYS_566_Computational_HW/tree/master/hw1/code`, as well as in plain text below.

A. Part 1

```
#!/usr/bin/env /home/mbe9/Documents/Spring_2016/PHYS_566_Computational_HW/anaconda/bin/python
# #! to the correct python install, not portable!!!

import os
import numpy as np
import matplotlib.pyplot as plt

# Produce the desired x ndarray, from -10.0 to 10.0 in steps of 0.05
x = np.linspace(-10.0, 10.0, 401)

# Declare the y ndarray
y1 = np.zeros(x.size)
y2 = np.zeros(x.size)

# Fill the y ndarrays
# Note that y2 at x = 0 is undefined not zero, so set it to None
for i in range(x.size):
  y1[i] = np.sin(x[i])
  if x[i] != 0.0:
    y2[i] = np.sin(x[i])/x[i]
  elif x[i] == 0.0:
    y2[i] = None

# Now plot and format
y1_plt, = plt.plot(x, y1, 'b', label='Sin(x)')
y2_plt, = plt.plot(x, y2, 'g', label='Sin(x)/x')

plt.axis([-11.0, 11.0, -1.1, 1.1])

plt.xlabel('x')
plt.ylabel('f(x)')

plt.legend(bbox_to_anchor=(0.0, 1.02, 1.0, 0.102), loc=3, ncol=2, borderaxespad=0.0)


# Set output path, hard coded...
output_path = './output'

# Create the output dir, if it already exists don't crash, otherwise raise an exception
# Adapted from A-B-B's response to http://stackoverflow.com/questions/273192/in-python-check-if-a-directory-exists-and-create-it-if-necessary
# Note in python 3.4+ 'os.makedirs(output_path, exist_ok=True)' would handle all of this...
try:
    os.makedirs(output_path)
except OSError:
    if not os.path.isdir(output_path):
        raise Exception('Problem creating output dir %s !!!\nA file with the same name probably already exists, please fix the conflict and run again.' % output_path)

# Print the plots
plt.savefig(output_path+'/part1.pdf')
plt.savefig(output_path+'/part1.eps')
plt.savefig(output_path+'/part1.png')
plt.savefig(output_path+'/part1.jpeg')

print 'Done!'
```

part1_production_and_plotting.py

B. Part 2

```
#!/usr/bin/env /home/mbe9/Documents/Spring_2016/PHYS_566_Computational_HW/anaconda/bin/python
# #! to the correct python install, not portable!!!

import os
import numpy as np
import matplotlib.pyplot as plt

# Produce the desired x ndarray, from -10.0 to 10.0 in steps of 0.05
x = np.linspace(-10.0, 10.0, 401)

# Declare the y ndarrays
y1 = np.zeros(x.size)
y2 = np.zeros(x.size)

# Fill the y ndarrays
# Note that y2 at x = 0 is undefined not zero, so set it to None
for i in range(x.size):
  y1[i] = np.sin(x[i])
  if x[i] != 0.0:
    y2[i] = np.sin(x[i])/x[i]
  elif x[i] == 0.0:
    y2[i] = None

# Set output path, hard coded...
output_path = './output'

# Create the output dir, if it already exists don't crash, otherwise raise an exception
# Adapted from A-B-B's response to http://stackoverflow.com/questions/273192/in-python-check-if-a-
    directory-exists-and-create-it-if-necessary
# Note in python 3.4+ 'os.makedirs(output_path, exist_ok=True)' would handle all of this...
try:
    os.makedirs(output_path)
except OSError:
    if not os.path.isdir(output_path):
        raise Exception('Problem creating output dir %s !!!\nA file with the same name probably already
    exists, please fix the conflict and run again.' % output_path)

# write output to file
output_file = open(output_path+'/data.txt','w')
for i in range(x.size):
        output_file.write('%(x).2f \t %(y1).6f \t %(y2).6f \n' % {'x': x[i], 'y1': y1[i], 'y2': y2[i]})

# close output file
output_file.close()

print 'Done!'
```

part2_production.py

C. Part 3

```
#!/usr/bin/env /home/mbe9/Documents/Spring_2016/PHYS_566_Computational_HW/anaconda/bin/python
# #! to the correct python install, not portable!!!

import os
import numpy as np
import matplotlib.pyplot as plt

# Set input path, hard coded...
input_path = './output'

# open input file and see how many data points/lines there are
input_file = open(input_path+'/data.txt','r')
num_data = sum(1 for line in input_file)
#print 'Input file %s has %i lines' % (input_file.name, num_data)

# Declare the y ndarray
x  = np.zeros(num_data)
y1 = np.zeros(num_data)
y2 = np.zeros(num_data)

# Return to the top of the file and read the data in
input_file.seek(0)

for i in range(num_data):
  line = input_file.readline()
  x[i]  = line.split('\t')[0]
  y1[i] = line.split('\t')[1]
  y2[i] = line.split('\t')[2]
  #print '%(x).2f \t %(y1).6f \t %(y2).6f' % {'x': x[i], 'y1': y1[i], 'y2': y2[i]}

# Close the input file
input_file.close()


##########################################################
# Recycle the plotting code from part 1

# Now plot and format
y1_plt, = plt.plot(x, y1, 'b', label='Sin(x)')
y2_plt, = plt.plot(x, y2, 'g', label='Sin(x)/x')

plt.axis([-11.0, 11.0, -1.1, 1.1])

plt.xlabel('x')
plt.ylabel('f(x)')

plt.legend(bbox_to_anchor=(0.0, 1.02, 1.0, 0.102), loc=3, ncol=2, borderaxespad=0.0)


# Set output path, hard coded...
output_path = './output'

# Create the output dir, if it already exists don't crash, otherwise raise an exception
# Adapted from A-B-B's response to http://stackoverflow.com/questions/273192/in-python-check-if-a-
    directory-exists-and-create-it-if-necessary
# Note in python 3.4+ 'os.makedirs(output_path, exist_ok=True)' would handle all of this...
try:
    os.makedirs(output_path)
except OSError:
    if not os.path.isdir(output_path):
        raise Exception('Problem creating output dir %s !!!\nA file with the same name probably already
    exists, please fix the conflict and run again.' % output_path)

# Print the plots
plt.savefig(output_path+'/part3.pdf')
plt.savefig(output_path+'/part3.eps')
plt.savefig(output_path+'/part3.png')
plt.savefig(output_path+'/part3.jpeg')

print 'Done!'
```

part3_plotting.py

PHY566 Homework #1

Due Date: Jan. 28th, 5:00pm via Sakai

Programming & Plotting Basics

Install Python on your computer (in case it does not have it installed) and write the following small programs:

a) write a Python program that calculates $\sin(x)$ and $\sin(x)/x$ from -10.0 to 10.0 in steps of 0.05 and plots both functions into one figure with proper axis annotations [4 points]

b) write a 2nd Python program (based on your first) that creates an ASCII file with a table. The first column of the table should contain floating point numbers from -10.0 to 10.0 in steps of 0.05. The second column should contain the value of the function $\sin(x)$, with x being the number in the 1st column. The 3rd column should contain the value of the function $\sin(x)/x$, again with x being the number in the first column. [3 points]

c) write a 3rd Python program that reads in the ASCII file you have generated and produces the same figure as the first one. Note that this program should solely rely on the table that you read in for populating the figure and should **not** attempt to calculate the functions $\sin(x)$ and $\sin(x)/x$. Save this figure as jpeg, eps and as pdf file. [3 points]

Your homework submission (via Sakai) should consist of:

- the source code of your three Python programs
- the data table you have generated with your 2nd program
- the pdf, eps and jpeg figures showing the content of your data table.

Please do **not** use iPython Notebooks, but write your Python code as a regular text file