Homework 8

Ch En 263 – Numerical Tools

Due date: 12 May 2020

Instructions

- For the problems in Excel, submit a workbook named "LastName_FirstName_HW8.xlsx" where each worksheet tab is named "Problem_1", "Problem_2", etc.
- For the problems in Python, submit a separate file for each problem named "LastName_FirstName_HW8_ProblemXX.py" where XX is the problem number.
- For your convenience, optional Excel and Python template files are available on the course website.
- If needed, a supplementary handwritten or typed document can be submitted via pdf on Learning Suite with the name "LastName_FirstName_HW8.pdf".
- **Please report how long it took you to complete the assignment (in hours) in the "Notes" section on Learning Suite.**

Problems

1. Visit finance.yahoo.com and select a stock by searching for a company name and selecting the "Historical Data" tab. Download the data for the past three months and import it into to a your Excel worksheet. Generate a single plot that shows the "High" and "Low" data. Format the plot for readability by:
   - Adjusting the plot size to be 4 in. high and 6 in. wide,
   - Plotting the data as lines (of different color) instead of markers,
   - Adjusting the y-axis to zoom in enough to see a difference between the two curves,
   - Naming the y-axes (e.g. Price, $),
   - Deleting the Chart Title,
   - Eliminating the grid-lines,
   - Putting a legend in the top-right corner that labels "High" and "Low",
   - Adjusting the font size to be 16, and
   - Rotating the text-direction of the dates on the x-axis to be vertical.
2. Do the following in a Python file.
   (a) Define arrays for the following equations

$$y_1 = 2x^{-3} + 2x^{-1}$$
$$y_2 = 2x^{-3}$$
$$y_3 = 2x^{-1}$$

   where $x \in [10^{-2}, 10^{2}]$ with at least 100 points in each array. *Hint: Use the numpy function* `logspace`.

(b) Write x, y_1, y_2, and y_3 to a text file called `power_law.dat` with a header that says `# columns:x, y1, y2, y3`.

(c) Read x, y_1, y_2, and y_3 back in from the text file `power_law.dat` from disk, and store them as the variables t, s_1, s_2, s_3

(d) Plot t versus s_1, s_2, and s_3 with the following formatting specifications, and save the plot as `power_law.pdf`.

- Make the plot a log-log plot.
- Make the default font size 16.
- Plot s_1 as a black solid line of width 2, s_2 as a blue dashed line of default width and s_3 as a red dotted line of default width.
- Label the x-axis with a symbol, σ.
- Label the y-axis with the word "Energy"
- Title the figure "Power-Law Crossover"
- Change the x-axis limits to span from 10^{-2} to 10^{2} and the y-axis from 10^{-5} to 10^{5}
- Add a legend where s_1 is labeled as "exact", s_2 is labeled as "large σ" and s_3 is labeled as "small σ".

Hint. You are going to have to read the online documentation to learn how to do some of this formatting. I did this on purpose, so you can get comfortable doing this because it is an important skill.

3. Do the following using both an Excel sheet and a Python file.

(a) In an Excel sheet, make a column called `n` that ranges from 0 to 100, another column where `t` that ranges from 0 to 10 with increments of $\Delta t = 0.1$ and a third column called `A` which obeys the formula

$$A_{n+1} = A_n + \Delta t \left[\frac{A_{in} - A_n}{\tau} - k A_n^2 \right]$$

where $A_{in} = 1$, $\tau = 2$, $k = 0.1$, $\Delta t = 0.1$ and $A_0 = A(t = 0) = 0$. *Hint: Set $A_0 = 0$ in the first cell. Plug in $n = 0$ into the equation above to find a formula for the next cell, A_1 that references the cell above (A_0). Then copy and paste down the column.*

(b) Export the t and A as columns to a text file called `EEsoln.csv`.

(c) Import the data into a Python file and plot A versus t. Format the plot with labels for the x- and y-axis and make the font size 16. Show the plot, but do not save it to disk.

4. Do the following in a Python file.

(a) Write a function called `doubledot(A, B)` that takes two matrices (2D numpy arrays) as arguments and returns a value using the following formula:

$$\boldsymbol{A} : \boldsymbol{B} = \sum_i \sum_j A_{ij} B_{ij}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are $N \times N$ matrices. *Hint: This is called a double-dot product. It is like a dot product for matrices. It should give you a single number.*

(b) Download the files `A.dat` and `B.dat` from the course website. Read your files into Python and execute your function on A and B. Print the resulting number.