

General course learning outcomes:

- demonstrate programming techniques in the construction of computer programs, including techniques: collect create store and manipulate data in larger structures such as arrays, matrices, and lists; and use control structures, such as conditionals and loops.
- apply programming techniques to solve problems in engineering, including plotting data.
- complete a team programming assignment that ties together concepts learned in the class.

Remember to document your code with comments and print labels and units when applicable. When these programs are completed, submit all files to Mimir. Remember the appropriate header information.

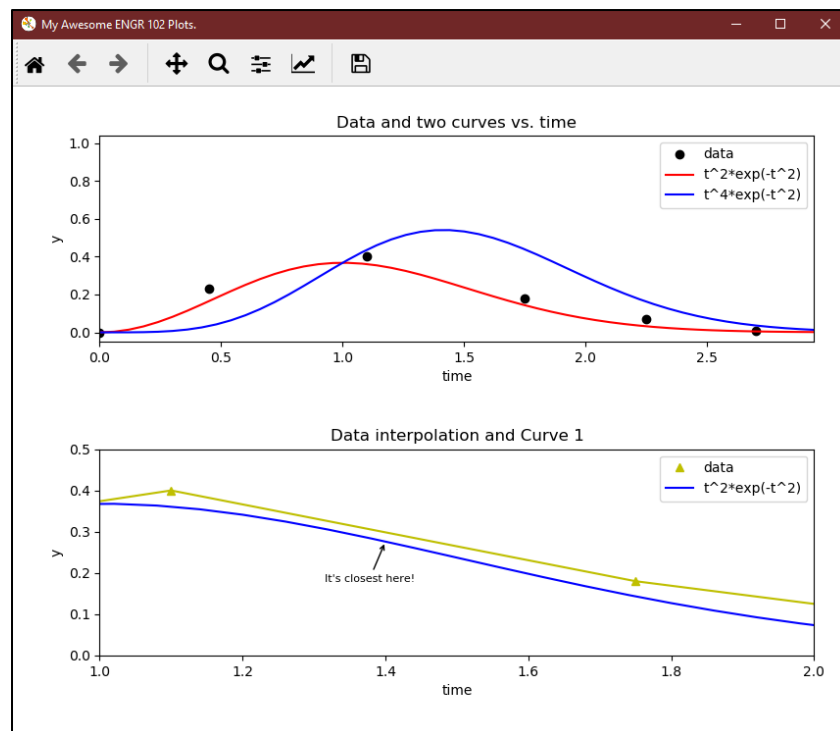
**Deliverables (what each team will be submitting) – 1 .py file:**

- Lab8\_Act.1.py

## Activity 1: Deep Plots by Jack Handy - to do in lab (team)

☒ Use lists, loops, and matplotlib functions to create professional plots in Python.

Write a Python program to generate the following plot as exactly as possible.:



In the top plot, you will be plotting 3 things: Data values, Function 1, and Function 2.

Data values:  $t = [0, 0.45, 1.1, 1.75, 2.25, 2.7]$  and  $y = [0, 0.23, 0.4, 0.18, 0.07, 0.01]$

Function 1:  $y(t) = t^2 \exp(-t^2)$

Function 2:  $y(t) = t^4 \exp(-t^2)$

In the bottom plot, you will be plotting Function 1 and data values.

**A few requirements and hints:**

- A. Use a loop to solve for each function at least 50 times between 0 and 3 and store the values in a list. *Do not use numpy for this assignment.*
- B. Don't actually interpolate the data by formula for the green line in the second plot, just plot a line between data points.
- C. You can make the legend prettier if you'd like, but it's not required (i.e., " $t^2 \exp(-t^2)$ " instead of " $t^2 * \exp(-t^2)$ ").
- D. Note the figure has a title, there are plot titles, axis labels and ranges, legends, an annotation at a specific location (get it close to the same spot), and specific colors.
- E. If your figure has axes and a title that are overlapping each other, try the `pyplot.tightlayout()` command directly before showing your plot.