EGR 141: Project 1

Summary: The goal of this lab is to help put together all of our basic MATLAB concepts before we move on to more formal programming. You should not use any MATLAB commands or concepts that are discussed in future chapters and sections (still no loops or if statements).

- Each of the following problems should have a script and, possibly, a function associated with them.
- For each problem, the script file should be called something appropriate, such as *Proj1_1_yourName.m*
- Include any functions that you needed to create in order to complete the problem. Name them whatever is indicated in the problem.
- Make sure to include any data you created (such as the .csv for problem 3).
- If more than one problem is in a script type *pause;* in between. Clearly indicate where the code for each problem begins by using a comment block. Start each new problem with a *clear* .
- If my example output "lines up nicely" then your output should as well.
- All output statements should output variables, not pre-computed constants. For example, if I ask you to output $r/2$ when $r = 3$, then you should set r to be three then output as *fprintf(' r/2 = %f ',r/2);* and not *fprintf('r/2 =1.5')* or *fprintf(r/2 = %f',3/2)*.
- Note that example output for each problem is not necessarily correct output (I intentionally change numbers so my answers will not always match your answers).

1. (3pts) From Calculus, recall that we cannot always find an explicit anti-derivative of a function $f(x)$ (such as $f(x) = \cos\frac{1}{x}$). Due to this fact, we are unable to compute the definite integral

$$\int_a^b f(x)\,dx$$

using the Fundamental Theorem of Calculus, but we can estimate this integral using a sum. Create three functions, *myLeftSum*, *myTrap*, and *mySimp* that each take in an anonymous function, $f(x)$, and the number of points in a grid, N, then ouput a single number, corresponding to

$$\int_0^1 f(x)\,dx.$$

We approximate this integral in a different way for each function. We

- Create N equally spaced points between 0 and 1. Store them in a vector x.
- Estimate the integral by computing the following sums
  - *myLeftSum:*

$$\int_0^1 f(x)\,dx \approx \frac{1}{N-1}\left(f(x_1) + f(x_2) + f(x_3) + f(x_4) + f(x_5) + \ldots + f(x_{N-2}) + f(x_{N-1})\right)$$

    This corresponds to using the *left Riemann sum* for approximating an integral. We sum areas of rectangles, using the function value at the left endpoint as the height
  - *myTrap:*

$$\int_0^1 f(x)\,dx \approx \frac{1}{2(N-1)}\left(f(x_1) + 2f(x_2) + 2f(x_3) + 2f(x_4) + 2f(x_5) + \ldots + 2f(x_{N-2}) + 2f(x_{N-1}) + f(x_N)\right)$$

    This corresponds to using the *trapezoidal rule* for approximating an integral. Instead of summing areas of rectangles, one uses a trapezoid instead.

– *mySimp*:

$$\int_0^1 f(x)\,dx \approx \frac{1}{3(N-1)}\left(f\left(x_1\right)+4f\left(x_2\right)+2f\left(x_3\right)+4f\left(x_4\right)+2f\left(x_5\right)+\ldots+2f\left(x_{N-2}\right)+4f\left(x_{N-1}\right)+f\left(x_N\right)\right)$$

This corresponds to using *Simpson's rule* for approximating an integral. Instead of summing areas of trapezoids, one fits a quadratic polynomial to the values of the integrand at the start, middle, and end of each rectangle.

Test your code with $f(x) = e^{-x^2}$ with $N = 11$, 21 and 41. Note that

$$\int_0^1 e^{-x^2}\,dx \approx 0.746824132.$$

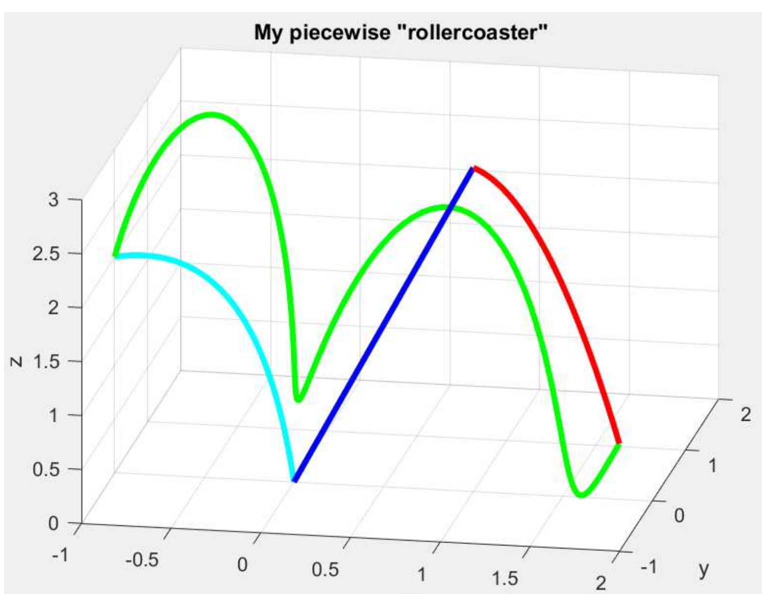
Nicely output all results to the screen using 8 decimal places. As usual, you are not allowed to use concepts beyond what we have covered (no loops).

```
Approximation to integral of cos(1/(x+1))
 For N=41
  L Sum: 0.757633582152
   Trap: 0.761849585352
   Simp: 0.761887150173
```

2. (3 pts) Create a three dimensional space curve graph that represents a roller coaster. You should have a continuous curve made of various piecewise functions. It is okay if your transition point between curves is not differentiable, but it must be continuous. Your curve must

- Have at least four parts, each of which needs to span a recognizeable portion of the roller coaster (it cannot be a tiny section).
- Must use at least three each of the following: A trigonometric function, an exponential function, a logarithmic function, a rational function, a non-linear polynomial.
- Your curve must be a closed curve (it must start and end at the same point). The start must correspond to $t = 0$. The end can be at whatever t you choose, but it must be very clear in your comments/coding that it is a closed curve.

Plot your curve, showing each piece with a different color. You will also be graded on how "fun" your roller coaster appears. A piecewise square is going to get very few style points. Be creative.



3. (4 pts) Navigate to https://ourworldindata.org/coronavirus-source-data. Find the current Covid dataset and download.

Our work belongs to everyone

Download the complete *Our World in Data* COVID-19 dataset
.xslx .csv .json (daily updated)

Pick a country (not the US), then edit your data so you have:

- Two columns. First is days since first reported case. Second is total cases. Make sure you keep track of the date your data originally starts.
- Trim the start of your data so that the first data point corresponds to when the number of cases is 10 or above, but retain the number of days since first reported.
- For example, this is the first bit for the US. The 13 corresponds to 13 days after the dataset starts. The 11 corresponds to 11 cases on that 13th day.

	A	B
1	13	11
2	14	11
3	15	11
4	16	12
5	17	12
6	18	12
7	19	12
8	20	12
9	21	13

- Export your data as a tab-delimited text file

- This file can now easily be read into MATLAB using the *load* command.

We're going to fit the data ($x =$ days since first reported case, $y =$number of total cases) with a line and a quadratic, using ideas learned in Lab 2. Create two functions:

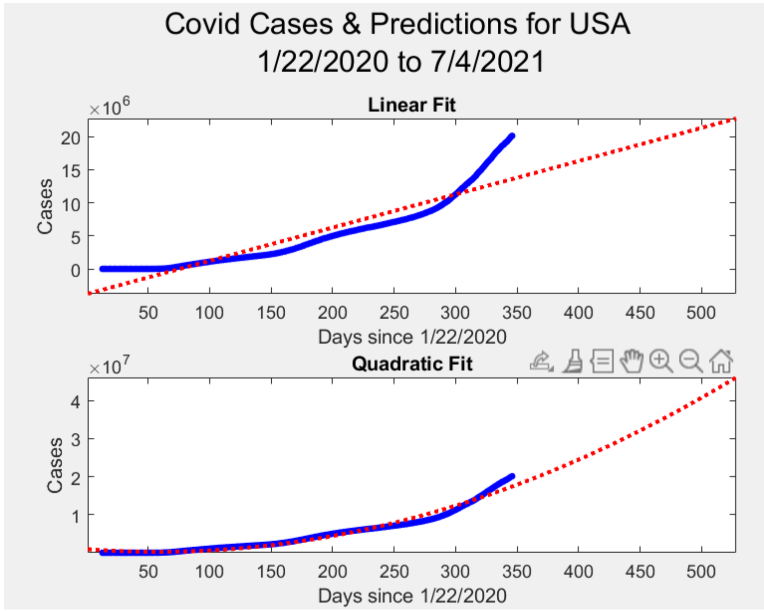
fitLine: Takes in two pieces of data, x and y. Returns two variables, m and b corresponding to the regression line $y = mx + b$.

fitQuad: Takes in two pieces of data, x and y. Returns three variables, a, b and c corresponding to the quadratic regression $y = ax^2 + bx + c$.

Do not use any built-in MATLAB functions to find the equations. Use the linear algebraic method learned in Lab 2 and do not compute an inverse of A. Output the estimate for the total number of cases (rounded to nearest integer) on 7/4/2021 for each model to the screen. Finally, using your functions, produce two plots in a 2×1 structure.

- The first plot should contain your data as single points, along with the best fit line. The line should start at 0 days since the reported cases (which may not be in your data set, as we stripped out data < 10) and go until 7/4/2021.
- The second plot should contain your data as single points, along with the best fit quadratic. The quadratic should start at 0 days since the reported cases and go until 7/4/2021.

```
Project 1 - Covid Est
 Linear prediction    : 22745203 cases on 7/4/2021
 Quadratic prediction: 46098015 cases on 7/4/2021
```


Covid Cases & Predictions for USA
1/22/2020 to 7/4/2021
Linear Fit
×10^6
Cases
20
15
10
5
0
50
100
150
200
250
300
350
400
450
500
Days since 1/22/2020
Quadratic Fit
×10^7
Cases
4
3
2
1
50
100
150
200
250
300
350
400
450
500
Days since 1/22/2020