Programming 5 Homework - MATLAB Intro

This homework has 2 parts:

- Part 1: Weather Data Plotting
- Part 2: Calculate Trajectory

Part 1 will be started and gone over in class along with some lessons on details pertaining to element by element calculation and plotting in MATLAB necessary to complete it. Part 2 will be started in class but you'll be able to work on it independently or in small groups without as much of a scaffold.

Part 1: Weather Data Plotting

This part will be started and gone over in class.

For this part, we will pull data from the GHCN (Global Historical Climatology Network). We'll use an existing script for that part, taking advantage of the work it's already done. You'll apply some conversions to that data and then plot the results.

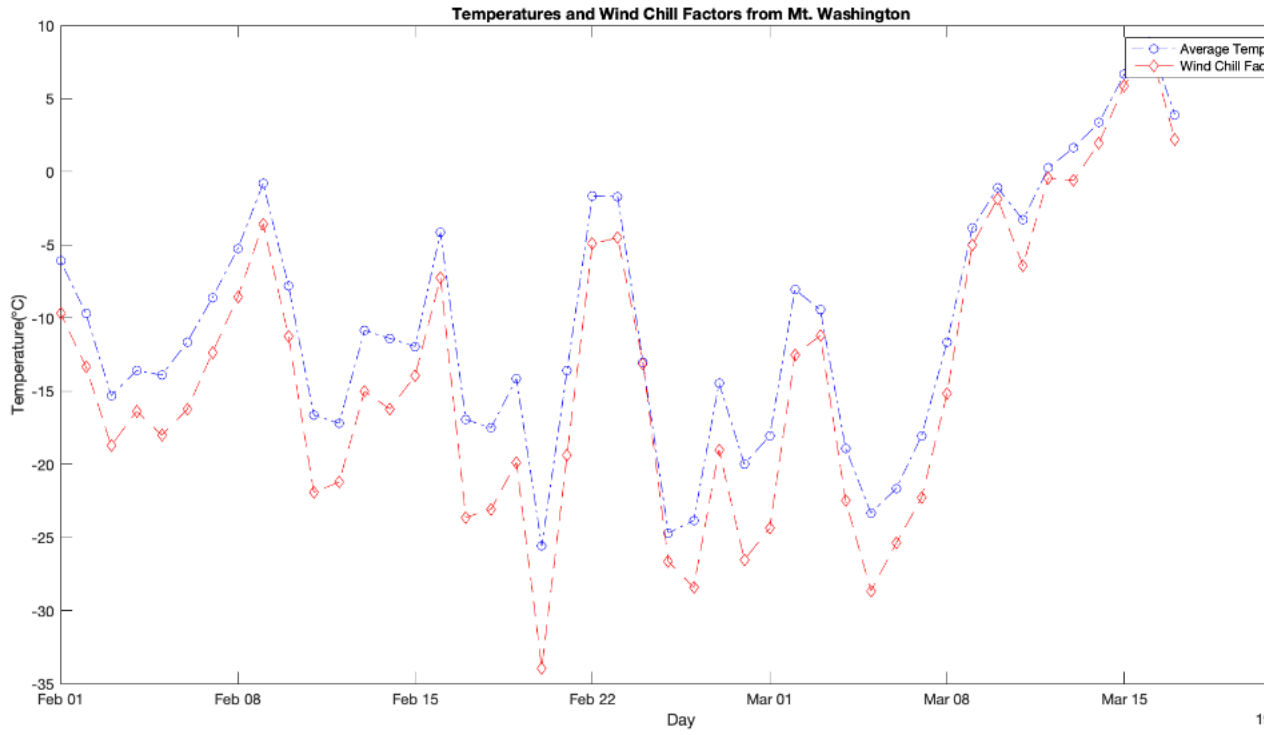
IMPORTANT WARNING:

The NOAA database does not have 100% coverage. If equipment breaks, it can take some time to address and therefore they're missing measurements for that day. If you're missing some data, if there's breaks in your graph or even missing entirely, then just run it again and try another date. If you want to see an example of broken data, try a fairly recent date like "2021-08-15". The NOAA weather station suffered some major damage in a storm that summer.

Create a program that adheres to the following pseudocode:

- `PROGRAM ML1H1:`
- `    Ask user for a start date`
- `    Ask user for a number of days`
- `    Load Temperature and Wind Velocity data for that time period`
- `    Calculate daily Avg. Temp and Wind Chill Factor`
- `    Plot Temperatures and wind chill factor`
- `END`

Here is an example of an expected output. You do not have to perfectly match the overall height and width of how it's laid out or the dates you're examining but you do need to have the same line specifications. The title and axis labels should also be similarly descriptive but don't need to match exactly.



You may use the following existing resources to aid in developing your program. Just so happens there's an existing "weather downloader" MATLAB script you can use - https://raw.githubusercontent.com/boconn7782/CourseCode/main/MATLAB/P5H1_support.m . This program:

- Downloads data from the NOAA weather stations based on a hard coded date and a user inputed number of days
- Loads some weather data
  - The High Temp, Low Temp, and Wind Velocity for each of the days in the time period
- Plots the raw data without any updates to the line specifications

Assignment Requirements

- Create a MATLAB script based on the provided pseudocode
- Incorporate the useful code from the provided script
- Run your script using a date of your choosing
  - Use one without data gaps
  - Use a time period of 20-50 days
- Make sure your code is well-commented (use the examples provided and in the support script as baselines)

Part 2: Calculate Trajectory

Background Info

This exercise will use MATLAB to calculate and plot the frictionless trajectories of a projectile such as you have studied in physics or will see soon. You will learn to use MATLAB to define variable names, use trig functions, solve a quadratic equation, define an array, use array multiplication and exponentiation ("`.*`" and "`.^`" in MATLAB) to calculate arrays of *x*- and *y*-points, and then plot these points to see the trajectory.

The problem is illustrated in Figure 1. A ball is projected at an angle of 60° with an initial velocity of 100 m/sec and traces out a parabolic trajectory as shown in Figure 1. Similar to the warm-up exercise, we want to

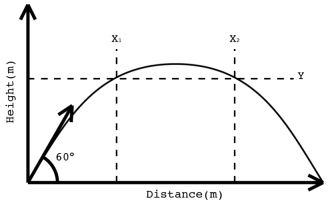


Figure 1: Height vs. Distance for Projectile Motion Problem

find the distance $x_2 - x_1$ for *several* different heights, y. The initial velocity can be broken into a horizontal and vertical component, V_{0x} and V_{0y}, and then the usual kinematic equations can be used for the *x*- and *y*-displacement.

Since there is no horizontal force in the *x*-direction, the *x*-component (distance) after an elapsed time *t* is given by:

$$x = V_{0x}t = V_0 cos(60)t$$

In the *y*-direction (height) there is a constant downward acceleration vector also known as gravity, $g = -9.8\frac{m}{s^2}$, as well as the initial upward component of velocity, $V_{0y} = V_0 sin(60)$, and the height of the ball at a time t is given by:

$$y = y_0 + V_{0y}t + .5gt^2$$

The instantaneous *y*-velocity is given by:

$$V_y = V_{0y} + gt$$

At the top of the trajectory $V_y = 0$, so the time to reach the top can be found from:

$$t_{top} = \frac{-V_{0y}}{g}$$

So, the first step is to review trajectory physics, and the next step is to become acquainted with how MATLAB functions so that it may be used to solve this and other real-world problems. The nice thing is that with MATLAB, you never have to solve for $y = f(x)$ to calculate and display trajectories. As for the trajectory physics, the important factors are calculating the maximum height and the total flight time.

Maximum Height of the Ball:

The maximum height of the ball occurs at the when $t = t_{top}$, So we can then calculate $y(t_{top})$ which reduces to $h_{max} = \frac{-{V_{0y}}^2}{2*g}$

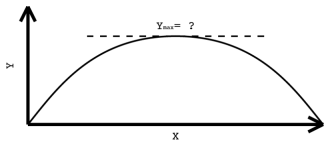


Figure 3: Calculate Max Height

Total flight time:

Since we are are only concerned with the path from $X_{initial} = 0$ to $X_{end} = 0$, the Total Flight Time is just double t_{top} .

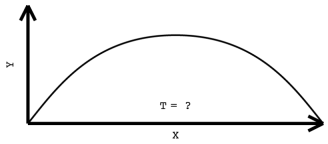


Figure 4: Calculate Flight Time

You should not have to derive any more than what's been provided above. And to repeat, YOU DO NOT NEED TO DERIVE ANYTHING MORE THAN WHAT'S BEEN PROVIDED. This is not about deriving new equations. It is about programming a MATLAB script that uses the ones provided. You may feel the need to reach out for a refresher on some of the physics though and that's fine. Prof. O'Connell and the red vests are available to help.

Writing a MATLAB Script

Write a MATLAB script that takes the initial velocity and angle as an input and outputs the maximum height of the projectile and its air time.

Follow the pseudo code below. This will not be provided in as much detail in the future, this is just because this is your first large independently developed program. For future assignments, you'll have to provide pseudocode or a flowchart to showcase your logic. For this first assignment though, the logic is laid out below with some (indented) more detailed instructions.

```
PROGRAM Trajectory:
Establish the User Interface (Typically referred to as the UI);
   Start with an explanation of to expect for the user
INPUT Ask the user for the initial velocity;
INPUT Ask the user for the initial angle;
   Include descriptive requests for inputs so the user
   knows what information is required.
Solve for the maximum height the ball reaches;
Solve for the length of time the ball is in the air;
Print the answers for the user;
```

```
   Include a description with that return so
   the user understands what the data is
Plot height versus time;
  You'll need your own time array. You don't need to derive anything or
have it autosolve in some way. Just calculate element by element then
plot.
END
```

Here is just one example of a user interface that follows the above pseudocode:

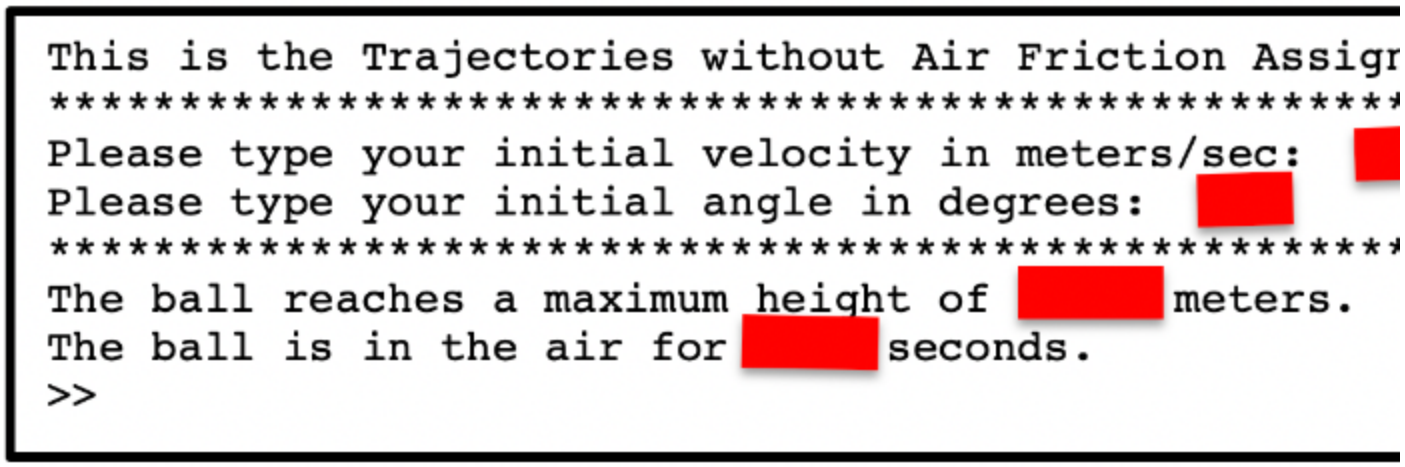


Figure 5: Example User Interface

For Extra Credit

Some assignments, but not all, will have an extra credit aspect to them. This is typically an added level of logical complexity and/or extra/more complex calculations or data handling to the assignment.

- Include a user input for an initial height (`y!=0`) and meet all requirements of Part 2 but calculate a flight path that starts from that initial non-zero height and ends at the ground (`y==0`)
- Output the max height(y_{max}) and total distance(x_{max}) travelled by the ball before hitting the ground

See the rubric for the weights of these requirements. Achieving extra credit typically meets the other requirements of the assignment so you do not need to submit extra files but you do need to indicate in the title block of the MATLAB script that you are attempting the extra credit so the TA is aware of that when grading your assignment.

Submissions

For Part 1:

- **m-file** - your MATLAB script
- **pdf** -your figure output

For Part 2

- **.m-file** - Your MATLAB Script
- **pdf** - your command window output **AND** your figure output
    - If you can't combine pdfs, one can have "_1" added to the name

Rubric

Part 1: Weather Data Plotter - 40% Total

- **-5% - Missing or Incomplete Title Block**
- **-5% - Missing or Insufficient Comments**

Programming

- 5% - Runs without errors
- 2.5% - Appropriate Commands
- 2.5% - Clean and clear UI

Functionality

- 5% - Loads data
- 10% - Calculates Wind Chill Factor

- 5% - Plots Data

Figures

- 2.5% - Axes labelled
- 2.5% - Titles
- 5% - Distinct Markers

Part 2: Calculate trajectory - 60% Total

- **-5% - Missing or Incomplete Title Block**
- **-5% - Missing or Insufficient Comments**

Programming

- 10% - Runs without errors
- 10% - Appropriate Commands
- 10% - Clean and clear UI

Returns Correct Values

- 10% - Returns Correct Max Height
- 10% - Returns Correct Time in Air

Figures

- 5% - Axes labelled
- 5% - Titles
- 5% - Distinct Markers

For Extra Credit - 5% Possible

- 3% - Include a user input for an initial height () and solve for from that height to ground ()
- 2% - Output the total distance () travelled by the ball

The following survey is for my reference to help improve future assignments. The results are not checked until after the semester is completed and participation is not required.