Part A of this project is in a separate document on the course website. Both parts have the same deadline.

You must work either on your own or with one partner. If you work with a partner you must first register as a group in CMS and then submit your work as a group. Adhere to the Code of Academic Integrity. For a group, “you” below refers to “your group.” You may discuss background issues and general strategies with others, but the work that you submit must be your own. In particular, you may discuss general ideas with others but you may not work out the detailed solutions with others. It is not OK for you to see or hear another student’s code and it is certainly not OK to copy code from another person or from published/Internet sources. If you feel that you cannot complete the assignment on your own, seek help from the course staff.

Objectives

Completing this project will solidify your understanding of user-defined functions and vectors (1-d arrays) and allow you to continue exploring MATLAB graphic functions. In this part of the project you will perform a sensitivity analysis, which is an important concept and tool in engineering and computational science.

Ground Rule

As before, do not use the break or continue commands.

1 Further Analysis of the Golf Ball Trajectory Model

In Project 2 we simulated the trajectory of a golf ball launched into the air and subject to air drag. Now we will perform a sensitivity analysis on the trajectory model. A sensitivity analysis is used to explore how the result of a model changes when the inputs to the model are systematically varied, i.e., we ask the question “how sensitive is the result to variations in the model parameters?”

Specifically, in this part of Project 3 you will investigate how the trajectory changes given changes in the launch angle $\phi$ and the coefficient of friction $k$. We decompose the problem into two parts: (1) implementing the trajectory simulation model as an independent function and (2) performing the sensitivity analysis by running the model repeatedly with systematically varied inputs and producing visualizations.

1.1 Simulation model as a function

Implement the following function:

```matlab
function [xvec, yvec] = golfTraj(dt,x0,y0,v0,phi,k)

% Simulate the trajectory of a golf ball from launch to landing.
% Input parameters
% dt: time step used in simulation, a positive value in seconds
% x0,y0: scalar initial x- and y-positions, in meters. y0>=0.
% v0: scalar initial velocity, a positive value in m/s
% phi: launch angle in radians, a positive value <pi/2
% k: friction coefficient, a positive value <1
% Return parameters
% xvec,yvec: vectors of the same length storing the positions of the
% golf ball: xvec(i), yvec(i) is the position of the ball
% after the ith time step. The last value in yvec should be
% set to zero.
```

The simulation ends when the golf ball returns to the ground, i.e., when its $y$ position becomes zero or negative. The last iteration of the simulation will likely produce a negative $y$ value instead of an “exact” zero. You will set that last value in yvec to zero.

Below are the relevant equations for the trajectory model; the detailed explanations can be found in the Project 2 document.
Given the initial velocity $v_0$ and launch angle $\phi$, the initial velocities in the x- and y-directions are $v_x = v_0 \cos \phi$ and $v_y = v_0 \sin \phi$. Let $\Delta t$ be a discrete time step. At each step of the simulation, compute the new velocities and positions as follows:

$$
\begin{align*}
    v_{x_{\text{new}}} &= v_x - \Delta t \cdot k \cdot v_x \sqrt{v_x^2 + v_y^2} \\
    v_{y_{\text{new}}} &= v_y - \Delta t \cdot (k \cdot v_y \sqrt{v_x^2 + v_y^2} + g) \\
    x_{\text{new}} &= x + v_x \cdot \Delta t \\
    y_{\text{new}} &= y + v_y \cdot \Delta t
\end{align*}
$$

where the acceleration due to gravity, $g$, is 9.81 m/s$^2$.

Your code from Project 2, correct or corrected, can be the basis for this function `golfTraj`! You may also choose to use the posted example solution of Project 2 as the basis. However, please be sure to use the appropriate parameter names as specified in the above function header. Do not modify the given function header above. Furthermore, do not just “dump” all your Project 2 golf ball simulation code into this function: remove any part that is unnecessary for this function, i.e., carefully reconsider every part of the simulation as you write the code.

### 1.2 Sensitivity analysis

You will evaluate the model’s sensitivity to changes in the parameters $\phi$ and $k$ independently. Write a script `golfTrajSensitivityAn` to perform the sensitivity analysis. Use these constant values: $\Delta t = .05$ second, initial position $x_0 = 0$, $y_0 = 0$ (in meters), and $v_0 = 100$ m/s.

#### 1.2.1 Sensitivity to launch angle $\phi$

Run your golf trajectory model, i.e., call your function `golfTraj`, on several values of $\phi$ in this range: $0 < \phi \leq \pi/2$. Use at least four and no more than six values. (Too many will produce a very busy graph.) In these simulation runs, fix the coefficient of friction, $k$, at 0.02. Plot in a figure window the trajectories for all the launch angles evaluated. The following code outline shows the relevant graphics commands for plotting the curves:

```matlab
% Close all figure windows
close all

% Start a new figure
figure

% First subplot in this figure window
subplot(2,1,1)

% See Note 1 below
hold all

% Initialize array variable for storing text. The curly brace notation will be discussed later in the course.
legendText = {};

% Loop over the set of launch angle values
for i = _____
    % Determine the trajectory for the current launch angle value
    % Call function plot to plot the trajectory, see Note 2 below
    legendText{i} = sprintf('phi=%.2f', ____);

end

% Place strings in legendText in the plot legend
legend(legendText)

% Needed before you do the next sensitivity analysis on k
hold off
```
Note 1 Users of MATLAB 2015b or a later release may use `hold on` or `hold all`, but users of older versions of MATLAB must use `hold all`, which cycles the plot colors automatically so that each call to `plot` produces a curve in a different color (within the built-in set of colors). `hold on` does not cycle the colors in older versions of MATLAB but does so in the 2015b and later releases.

Note 2 Recall that the returned vectors from function `golfTraj` do not include the beginning coordinates. Therefore you need to concatenate to those returned vectors the beginning coordinates \((x_0, y_0)\) in order to plot the complete trajectory.

Note 3 On the blank should be the current launch angle. Note the use of the curly brace, not parenthesis, around the index of the variable `legendText`. We will discuss this kind of array later in the course.

1.2.2 Sensitivity to coefficient of friction \(k\)

Continue writing code in `golfTrajSensitivityAn.m` to do a sensitivity analysis on \(k\). Fix the launch angle at \(\pi/4\) and vary \(k\) over the range of 0.01 to 0.2. Again, choose four to six \(k\) values to show the variability of the trajectory due to changes in \(k\). Use a similar graphics framework as shown above to produce a plot of all the trajectories at the chosen \(k\) values. This will be the second subplot in the figure, so start this second visualization with the command `subplot(2,1,2)`.

Finally let’s change the size of the figure window so that the two subplots are more spaced out. Write the following statement immediately below the `figure` command that was used to start the figure window:

```plaintext
    set(gcf, 'Units', 'normalized', 'Position', [.3 .2 .5 .6])
```

This `set` statement positions the lower left corner of the figure window at 30% from the left edge of the screen and 20% from the bottom edge of the screen, and sets the width and height of the figure window to be 50% and 60% of the screen’s width and height, respectively. This and a select set of graphics format options are presented in Appendix A: Refined Graphics in our textbook. You can also search the MATLAB documentation for even more options!

Submit your files `golfTraj.m` and `golfTrajSensitivityAn.m` on CMS.