Evaluating the Error between a model prediction and actual behaviour

Prof. Anil Kokaram : Assignment 1

1 Jan 2023

This assignment requires you to use the solution for the parachutist ODE you examined in class, to analyse the Red Bull Scientific data. The template code for this assignment is `redbull_lab_template`. Submit your Matlab code to the blackboard site. YOU MUST USE THE NAMING CONVENTION `GroupNumber_LastName_FirstName.m` for the name of the file when you submit. Failure to do so may result in no marks for this submission. **This assignment counts for 5% of your final mark.**

The Red Bull Stratos Jump Team have recorded their scientific data online **here**. The key data recording the velocity of Mr. B. wrt time is in the CSV[1] file RedBullJumpData.csv which you can download from blackboard. the lines of code below show you how to read from a CSV file in Matlab.

[1] A CSV file is a Comma Separated Value file

```
jumpdata = csvread('RedBullJumpData.csv');
t_redbull = jumpdata(:,1);
v_redbull = jumpdata(:,2);
terminal_velocity = jumpdata(:,3);
N_timestamps = length(t_redbull);
```

`t_redbull` is a vector containing the timestamps at which the velocities in `v_redbull` were measured. The number of instants at which the time and velocity pairs were measured is `N_timestamps`.

Note that the third column in the file is the measured *terminal velocity profile* of the falling object. It was reported by the team but is not clearly defined. In the last question you use this data to model the descent more accurately.

As you will see, the fall is initially unhindered by drag. Then as the atmosphere causes drag, the acceleration changes and eventually Mr. B. reaches terminal velocity. Our idealised solution does not take into account the change of drag with time and so it is likely to be quite a poor model of what happens to Mr. B. as he falls.

Your single submitted Matlab script should address every one of the questions below. The answer to each part should be stored in the variable name as specified. UNLESS OTHERWISE STATED, ALL LINES IN PLOTS SHOULD USE A

LINEWIDTH OF 2.0. PLEASE USE THE TEMPLATE CODE PROVIDED.

1. Plot the measured values of velocity versus time from the JumpData.CSV file. Put your line into a plot handle called `h_part1` using `h_part1 = plot(....)` as usual. Use the color red for the line. Mark each point on your line with a red $\times$ mark. Render this plot as Figure 1. Fix your axes to a time duration of 180 secs and a maximum velocity of $400m/s^2$. Label the axes appropriately, show the grid and also adjust the fontsize of the axis labels to be 24.

2. Superimpose on that plot another line (in dashed blue), which shows the velocity versus time for an object in freefall without the effect of drag. Put that line into a plot handle called `h_part2`. Note that this implies that acceleration is constant and so $v(t) = at$, where acceleration is in m/s^2 and $a = g$. Assign your freefall velocity to a variable `v_freefall` at the same time instants as `t_redbull`.

The percentage error in velocity here is measured as the percentage of the idealised freefall velocity and not a percentage of the measured velocity.

3. Estimate the time at which Mr. B. starts to enter the atmosphere. Do this either by using the plot in Figure 1 (i.e. by graphical/manual estimation) or otherwise. You may assume that when the actual measured velocity deviates from idealised freefall by $>= 5\%$ then he has entered the atmosphere. Print out to the command line a report of this value i.e. *Mr. B. enters the earth's atmosphere at X secs after he jumps.* Assign that value to the variable `hit_instant`.

Note that you should use the same timesteps as in the JumpData file. That means instead of a constant times interval between each point in your Euler solution you will have to change the time interval in line with the size in the recorded file.

4. Assume that the velocity of Felix as he falls is modelled by the first order differential equation that we used in the previous laboratory. Starting from the initial conditions at $t = 56$ secs use your numerical solution code from the previous laboratory to generate an estimate of the velocity of Mr B after that time i.e. up to 180 secs. Assume initial conditions are as measured in the file. Use $c/m = 3/60$. Superimpose the resulting numerical solution of velocity versus time on your plot in Figure 1 using a dashed green line **with (+) markers for each point**. Assign that line to the plot handle `h_part4`. Assign your velocity vector to the variable `v_numerical_1`.

5. Calculate the percentage **absolute** error between your numerical solution and the actual measured value of velocity of Mr. B at 64 and 170 secs. Your code must also print this out on the command line using the text as follows. *The Percentage error at 64 and 170 secs is X and Y respectively.* Assign your percentage error measurements to a 2 element vector `per_error`.

The percentage error here is measured as the percentage of the actual measured velocity.

6. As you can see the model we have used in lectures is unable to capture what actually happens. There are two problems with our idealised model. Firstly, the drag force is proportional to $v(t)^2$ and not $v(t)$. Secondly, the drag force increases with decreasing altitude. The actual differential equation is therefore as follows.

The measurement labelled *terminal velocity* in the data file is therefore actually a measurement that expresses the instantaneous drag force in terms of the terminal velocity that the object would have attained had it continued to fall under the influence of constant drag.

$$\frac{dv(t)}{dt} = g - \frac{c(t)}{m}v^2(t) \tag{1}$$

Solve this new differential equation using Euler's solution, starting with initial conditions at $t = 56$ secs as previously and ending at $t = 100$ secs. It turns out that the measured data `terminal_velocity` allows you work out $c(t)/m$ as follows.

You can do this using a simple modification to your existing Euler solution. Note that you'll need to update the drag constant at every timestamp as its no longer constant. Remember that the measured terminal velocity is that column 3 which you already have from the very first step in this assignment.

$$\frac{c(t)}{m} = \frac{g}{\bar{v}(t)^2} \tag{2}$$

where $\bar{v}(t)$ is the measured terminal velocity at timestamp t.

Superimpose this new velocity solution on your existing plot using a dashed line coloured black with $(+)$ markers for each point. Assign that velocity vector to the variable `h_part6`. Report your measured absolute percentage error at $t = 100$s using *The error at t = 100 secs using my estimated drag information is X.*. Assign that error to the variable `est_error`.

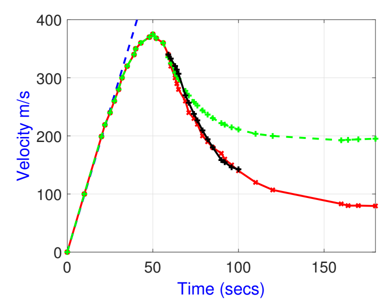

Figure 1: Your final plot should look something like this.