Bungee Jump Simulation (Second order Ordinary Differential Equations)

A team of engineering students is writing a MATLAB program to simulate bungee jumping for a company designing bungee jumping equipment.

The new **bungee core** will be **100 meters long** and be made of highly specialized rubber with a linear stress vs. strain ratio throughout the entire 400% elongation.

Average bungee jumpers' mass is estimated to be **80 kg** (176.37 pounds).

The bungee jumpers usually make high-altitude jumps. The purpose of the simulation is to calculate a safe jump altitude for this new bungee core so that the jumper will not hit the ground.

In addition, the company wants to ensure the maximum acceleration during the jump do **not exceed 2 g's (1 g = 9.806 m / sec^2).** The arresting force caused by such acceleration may injure some of the jumpers.

The equation to use for the analysis is **Newton's Second Law**,

$$F = ma$$

where **F** is the sum of the gravitational, aerodynamic drag, and bungee forces acting on the jumper, **m** is the mass of the jumper (which is **80 kg**), and **a** is the acceleration. **Define the distance the jumper falls as the variable x (which is a function of time, x(t)).** The jumper's **velocity and acceleration are then represented as x' and x'', respectively**. The Newton's equation to solve for acceleration:

$$x'' = F / m$$

Next, determines the forces making up F. The gravitational force will be the jumper's weight, which is:

$$\begin{aligned} W &= m\,g \\ &= (80 \text{ kg})\ (9.806 \text{ m/s}^2) = 784.48 \text{ N} \end{aligned}$$

The **aerodynamic drag, D**, will be proportional to the square of the jumper's velocity, $D = c\,(x')^2$, but the value of c, the constant of proportionality, is unknown. However, experienced skydivers know that **the terminal velocity in a free-fall is about 56 m/s (120 mph)**. At that speed, the aerodynamic drag is equal to the weight of the jumper, so c can be determined using:

$$\begin{aligned} c &= D / (x')^2 \\ &= (784.48 \text{ N}) / (56 \text{ m/s})^2 \\ &= \mathbf{0.2502 \text{ kg/m}} \end{aligned}$$

Finally, after the jumper has fallen 100 m, the slack in the bungee will be eliminated, and it will begin to exert **an arresting force, B, of 10 N for every meter that it is stretched beyond 100 m**.

The new bungee core also has a **viscous friction force, R, once it begins to stretch**, which is given by:

$$R = -\,1.5\,x'$$

Thus, there will be two regions for computing the acceleration. **The first equation will be used when the distance x is less than or equal to 100 m:**

$$x'' = F / m = (W - D) / m = (784.48 - 0.2502\,(x')^2) / 80$$

A second equation will be used when x is greater than 100 m:

$$x'' = F / m = (W - D - B - R) / m = (784.48 - 0.2502\,(x')^2 - 10\,(x - 100) - 1.5\,x') / 80$$

Integrate for the interval beginning at 0 seconds to find the acceleration, velocity, and distance as a function of time from the beginning of the jump (which is assumed to occur at t = 0) **to 400 seconds after the jump.**

What are the peak values of acceleration, velocity, and distance? You may need to experiment with the simulation step size in order to choose one that will give you the peak values. A negative acceleration or negative velocity value means the jumper is going up. Remember to check the peak values for both up and down directions.

(Hint: You may want to utilize the **min**(x) or **max**(x) MATLAB functions to find the minimum or maximum values in your vector arrays.)

The company does **not** want the maximum acceleration **to exceed 2 g's (1 g = 9.806 m / sec^2)**.
Is the estimate of the peak acceleration higher or lower?

How close does the jumper come to reaching the estimated terminal velocity of 55 m / sec?

How far does the jumper initially fall before he starts back up?

How many seconds does he initially fall before he starts back up?

How high should the company advertise to be the minimum altitude to jump with this new bungee core to ensure a safety factor of 2? (Hint: twice the longest distance the bungee jumper falls away from the jump off point.)

At the end of the 400 seconds simulation, what is the final length of the bungee core?

<u>Simulation Results:</u>

- Create one **Simulink model** to simulate the bungee jumper's distance (x) vs. t, the velocity (x') vs. t and acceleration (x'') vs. t for the first 400 seconds of the jump.

- Use the "**To File**" blocks in the Simulink model to save the simulation result, distance (x) vs. t, the velocity (x') vs. t and acceleration (x'') vs. t, to a .mat file.

- Create a **MATLAB program** that will automatically **activate the Simulink model and run the Simulink model**. Load the .mat file generated by the Simulink to the MATLAB program.

- Use the **ode45 method** to simulate the bungee jumper's distance (x) vs. t, the velocity (x') vs. t and acceleration (x'') vs. t for the first 400 seconds of the jump again.

- Generate **one 3x2 subplots** to plot and compare the **Simulink solution** and the **ode45 solution** side-by-side.

  The figure should include the following:
  - Plot the bungee jumper's distance (x) vs. t chart with the Simulink Solution on the left, and plot the x vs. t chart using the ode45 solution on the right.
  - Plot the bungee jumper's velocity (x') vs. t chart with the Simulink Solution on the left, and plot the x' vs. t chart using the ode45 solution on the right.
  - Plot the bungee jumper's acceleration (x'') vs. t chart with the Simulink Solution on the left, and plot the x'' vs. t chart using the ode45 solution on the right.

- **Answer all the simulation analyses questions** with the **fprintf** commands and print key analyses parameters and answers on screen **or** to a text file.

- **Title and label plots clearly.**

- Add lots of comments. Make this simulation a demo worthy program for interviews.