CVEN 302-501
Computer Applications in Engineering and Construction
Homework #1 Course administration and introduction to Matlab

Date distributed : 9.2.2005
Date due : 9.9.2005 at 5:00 pm

Return your solution either in class or in my mail box (WERC Rm. 235H) by the date shown above. Please show all your work and follow the rules outlined in the course syllabus.

1 Computer Account

If you do not already have a CivilStudent account, drop by the Civil Engineering Computer Lab in CE 215 to sign up for one. It usually takes about one day for your new account to become active.

2 Set up NEO Account

Periodically, I may need to send emails to the class or to individual students. Since I only have access to your NEO email address, you need to set up email forwarding to another account if you are not in the practice of checking your NEO email. Please do one of the following:

- If you regularly check your NEO account: do nothing.
- If you do not check your NEO account regularly: log on to NEO, select the link in the upper right corner for [Directory], select the Edit link in the upper right corner underneath the statement logged in as..., click the radio button for is forwarded to and type in the email address you regularly check.

If you need help, please consult the Computer Help Desk at 845-8300.
3 Order of Operations

Write Matlab commands based on the standard order of operations using + - * / ^ () and the functions pi, exp, sqrt, log, and sin to evaluate the following expressions. Trust the order of operations rules and use a minimum number of parentheses. Use the Matlab interactive session to evaluate your expressions and use a hand-held calculator to check your results. Use the following parameter values when checking your results.

\begin{align}
    a &= 15 \\
    b &= \pi \\
    c &= \exp(1) \\
    d &= \sqrt{2} \\
    t &= 0.0125
\end{align}

(1)

Example: \( y = \frac{12-a}{b} \sin^2(c) \)

Answer: \( y = (12 - a)/b * \sin(c)^2 \) evaluates in Matlab to \( y = -0.1611 \).
Hand calculation agrees with Matlab.

1. \( x = 3a^2 + 2b\sqrt{c} - d^{1/a} \)
2. \( s = \frac{b-c}{a-d} \)
3. \( r = \frac{1}{1-\ln(\frac{5}{2})+\frac{1}{2}} \)
4. \( w = 2^5 \sin(at^2) \)

4 Problems from Chapra (2005)

Do problems 3.2 and 3.10. Turn in your Matlab .m-files and a printout of the output of your programs.

5 Bungee Jumper

5.1 Part 1

Write a Matlab program to solve for the velocity \( v \) of a bungee jumper as a function of distance \( x \) below the jump point and for the location and value of the maximum acceleration. Plot the velocity as a function of distance and identify the maximum acceleration point on the graph.

The model equations for the jump for \( x \) above the ground level \( H \) are:

\[ \frac{dx}{dt} = v \]  

(2)
and
\[
\frac{dv}{dt} = \begin{cases} 
g - \frac{c_d}{m}v^2 & \text{if } x \leq L, \\
g - \frac{c_d}{m}v^2 - (x - L)k & \text{otherwise}
\end{cases}
\]  

(3)

where \(g\) is the acceleration of gravity, \(c_d\) is a second-order drag coefficient, \(m\) is the mass of the jumper, \(L\) is the relaxed length of the bungee cord, and \(k\) is the spring constant of the bungee cord.

Your program should calculate using Euler’s method with a constant step size in time. Chapra (2005) presents an example program for the free-fall region on page 54.

Your program should calculate the position and velocity as arrays, one value at each calculation time \(t\). You can test your program with the following input values:

\[
\begin{align*}
\text{dt} & = 0.1; \quad \% \text{ time step (s)} \\
\text{ti} & = 0; \quad \% \text{ initial time (s)} \\
\text{tf} & = 25; \quad \% \text{ final time (s)} \\
\text{vi} & = 0; \quad \% \text{ initial velocity (s)} \\
\text{xi} & = 0; \quad \% \text{ initial position (m)} \\
\text{m} & = 68.1; \quad \% \text{ mass (kg)} \\
\text{cd} & = 0.25; \quad \% \text{ second order drag coefficient (kg/m)} \\
\text{L} & = 200; \quad \% \text{ length of bungee cord (m)} \\
\text{k} & = 0.3; \quad \% \text{ spring constant of bungee cord (1/s}^2) \\
\text{H} & = 350; \quad \% \text{ height of jump point above the ground (m)}
\end{align*}
\]

These values should give you a plot like in Figure 1.

5.2 Part 2

Use your program to design the proper values of \(L\) and \(k\) such that the following criteria are met:

- The maximum acceleration should be \(1.5g\).
- The lowest point of descent should be 50 m above the ground.

Turn in:

1. Your notes for designing your program.
2. Print-outs of your program and all related .m-files.
3. A plot of the jump trajectory for the design condition in Part 2.
4. The design values for the solution to Part 2.
Figure 1: Solution to Problem 5 for the test input data.

References