<u>FINAL SYNTHESIS PROJECT</u>: MECHANICAL ENGINEERING
Due Tuesday, May 17th at noon
(A special box will be set up in Hollister 220 for drop-off)

Reading Assignment
All your previous Problem Sets and any associated reading material.

Learning Objectives

a) Solve 2nd order ODE's by converting them to a system of 1st order ODE's.
b) Explore and analyze the solution to the above system using techniques learned throughout the semester in this class.

NOTE: You are expected to turn hard-copies of your code-listings.
It is also imperative that on a separate sheet you ***outline which part of the work each team member has performed***.

PROJECT STATEMENT

1. Quarter Car Model

The quarter car model (QCM), shown in Figure 1, is a model for the response of the right (or left) front suspension of a car to road forcing. In this project, you will consider the simpler case of a passive QCM, where no active controller is used to enhance the damping efficiency of the suspension.

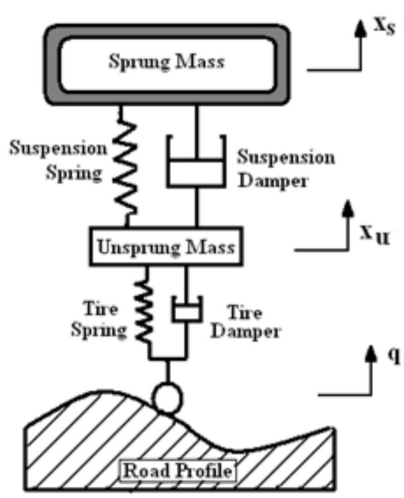

Fig. 1: Two degree of freedom passive quarter car model.

The governing equations of the QCM are

$$m_s \ddot{x}_s + F_{sp} + F_d = 0 \qquad (1)$$

$$m_u \ddot{x}_u - F_{sp} - F_d = [k_t(q - x_u) + c_t(\dot{q} - \dot{x}_u)] \qquad (2)$$

where x_u, x_s and q are the displacements of the sprung (car chassis), unsprung mass and road (from a given reference level), respectively. The values of mass, stiffness and damping coefficients are given in the table below.

Sprung mass (m_s)	236.12 Kg
Unsprung mass (m_u)	23.61 Kg
Tire Stiffness (k_t)	181818.88 N/m
Tire Damping Coefficient (c_t)	13.854 N sec/m

Table 1: Suspension parameters for test Quarter Car Model.

In (1) and (2), F_{sp} and F_d are the spring and damper forces which are nonlinear functions of the aggregate displacement, $\Delta x = x_s - x_u$, and aggregate velocity, $\Delta \dot{x} = \dot{x}_s - \dot{x}_u$

$$\mathrm{F_{sp}} = k_1 \Delta x + k_2 \Delta x^2 + k_3 \Delta x^3, \quad (3)$$

$$F_d = c_1 \Delta \dot{x} + c_2 \Delta \dot{x}^2 \quad . \quad (4)$$

You will have to compute the coefficients k_1, k_2, k_3 and c_1, c_2 through a least squares curve-fit outlined below.

The road profile (induced displacement and velocity) is given by

$$q = A\sin(\omega t) \qquad \text{for } 0 \le t \le T \qquad (5)$$

This represents a model circular bump, modeled by half a sine wave with amplitude A=0.1m and length L=5.2m. T is the time needed for the car to go over the bump and its value in seconds is given by

$$T = L / (V \cdot 1000 / 3600) \quad (6)$$

where V is the car velocity in *km/h*. The period of the above sine wave is

$$2T = 2\pi / \omega \quad . \quad (7)$$

a) The attached files *springforce.mat* and *dampingforce.mat* contain datasets with select values of F_{sp} and F_d (in N) as a function of Δx and $\Delta \dot{x}$, respectively. Perform a polynomial least squares fit on each dataset to compute the values of coefficients k_1, k_2, k_3 and c_1, c_2. On two separate plots, show the F_{sp} and F_d data and the corresponding least-squares fits.

b) Now convert equations (1) and (2) into a system of four 1st order ODE's. Write out clearly this system of ODEs.

c) You will now *numerically solve the above system of ODEs* using a 4th order Runge Kutta method:

To this end, you will adapt the attached code *rk4sys*. This is the same code that is found in figure 22.8 of the Chapra textbook. Examples of how to build the M-file that holds the system of ODEs and how to call *rk4sys* may be found in section 22.5.3 and case study 22.6 of the textbook. The comments in this code are also extremely helpful.

NOTE: As an alternative, you **can use built-in Matlab commands or *Simulink* as alternative ways to build this solve**r. Critical in this direction is that you use a 4-th order accurate Runge-Kutta solver that uses a *fixed timestep*. Your choice for this is *ode4*, which we cannot help you with when implementing it to solve a system of ODEs.

Do not use *ode45* ! Even if you get your output in uniformly spaced times, this will involve a lower-accuracy interpolation that will degrade the accuracy of the 4th order Runge-Kutta method.

EXTRA CREDIT: One way towards building a "sanity test" for your numerical solver (both Runge-Kutta and Euler) is to build an analytical solution to test against. You can only do this via Laplace Transforms for the *linearized problem*, where you would retain only the linear terms for the spring and damping forces in Equations (3) and (4). The linearized problem is only valid for small displacements and velocities but can serve as a very useful comparison to compare your numerical model to.

If you have not worked with Laplace Transforms worry not and consult the class notes from lecture 24 on alternative ways to check your results.

2. Simulation

a) You will run your model for 2 different car velocities values of 10 and 40 *km/h*. Your simulation will end at time t=4s. Select your simulation timestep, h, such that you have *at least 50 timesteps per the characteristic time* scale of your problem, i.e. the time it takes the car to go over the bump.

Perform a *timestep independence test*: Run first with $h=T/50$. Then repeat a simulation with $h=T/100$. Compare your results for x_u and x_s for each of the two timesteps. If these are *visually indistinguishable* (no need for any quantitative tests), your RK4-based simulations $h=T/50$ will be those you will analyze. Otherwise, continue reducing your timestep by ½ until you obtain *timestep independence*.

The *initial values* for all 4 of your unknown variables should be set to zero.

b) For your 2 different values of car velocity V, plot $x_u, x_s, \dot{x}_u$ and $\dot{x}_s$ as a function of time. Use one plot with four MATLAB *subplots,* one for each variable. How do the amplitude and structure (period of oscillation, time of decay to very small values) of x_s and $\dot{x}_s$ change with V?

c) Write a simple forward Euler code that solves the system of ODEs you constructed in part 1(b). Outline explicitly the calculations needed to advance your vector of unknown variables from time t_i to t_{i+1}.

Now, focus on the case of V=40 km/h. Run your Forward Euler code with the time step you used in part 2(a) . Using four MATLAB *subplots*, one for each variable, plot $x_u, x_s, \dot{x}_u$ and $\dot{x}_s$ as a function of time contrasting the Forward Euler method to the RK4 technique. At the time t=2T, what is the absolute difference between the two methods for the four unknown variables?

Through trial and error, determine how much smaller you have to make h, for the above absolute difference to drop below 10^{-3} for all four unknown variables. How many total time steps did you have to run your Forward Euler code for in this case? Using the same timing tools you employed in Problem Set 04 determine whether, for this value of time step h, your Forward Euler code is computationally less costly than your RK4 code. Can you explain your answer?

HINT: Think how many RHS evaluations each method involves over the course of a full simulation.

3. Data Analysis

In this part of the project, you will focus on your RK4 results.

a) For your 2 different values of car velocity V, plot the sprung mass acceleration, $\ddot{x}_s$, as a function of time, t. Compute $\ddot{x}_s$ using a $O(h^2)$ centered finite difference scheme.

Now construct a table of the value of maximum recorded value of sprung mass acceleration as a function of car velocity V across all 2 runs. Taking into account the table below which shows the four passenger comfort ranges as a function of acceleration, indicate in which comfort range each of the 2 values of V resides within.

Max. Acceleration (m^2/sec)	Passenger Comfort Level
0 to 0.8 m^2/sec	Comfortable
0.8 to 1.6 m^2/sec	Uncomfortable
1.6 to 2.5 m^2/sec	Very uncomfortable
2.5 m^2/sec and over	Extremely Uncomfortable

Table 2: Comfort level ranges for test Quarter Car Model.

b) Repeat the exercise above using a $O(h)$-accurate finite difference scheme. Is your assessment of passenger comfort range for each value of car velocity impacted by the accuracy of your numerical derivative computation?

c) For each of the 2 car velocities, what is the first zero-crossing time for the sprung mass displacement, x_s, timeseries ?

To this end, sample 6 data points to the left and right of the visually identified zero crossing point, construct the 5th degree Interpolating Polynomial, $P_5(t)$, in Lagrange form for these points. Outline your steps in building this polynomial.

$P_5(t)$ will be the function whose zero crossing you will have to compute using the nonlinear root finding technique of your choice. Compute this crossing time with an approximate absolute error tolerance of 10^{-3}. Justify why you chose the particular technique.

d) For each of the 2 car velocities, compute numerically how much energy is lost due to damping by the shock absorber over the time interval T the car takes to go over the bump. This energy lost due to damping is given by the integral:

$$E_d = \int_0^{t=T} F_d \Delta\dot{x} dt \quad , \ (7)$$

Where F_d is given by equation (4) and $\Delta\dot{x} = \dot{x}_s - \dot{x}_u$.

Perform your numerical integration using both composite trapezoidal and 1/3 Simpson's rules ensuring that you have an odd number of points. What is the relative difference between the two different estimates of E_d. Does the use of the composite 1/3 Simpson's rule produce radically different results?

If you needed more accuracy could you use Gauss Legendre quadrature to compute the integral in equation (7)?