Pham, Jonathan
Sit, Hilarie
CEE 3200

Final Project – Mechanical Engineering Version

Due date: 5/17/18

Section 1: Quarter Car Model

Part a)

A polynomial least squares fit was performed on the provided force data to determine the coefficients for the below spring and damping force equations. See the attached least squares fit MATLAB codes.

$$Spring\ force: \quad F_{sp} = k_1 \Delta x + k_2 \Delta x^2 + k_3 \Delta x^3$$

$$Damping\ force: \quad F_d = c_1 \Delta \dot{x} + c_2 \Delta \dot{x}^2$$

$$with \quad \Delta x = x_s - x_u \quad and \quad \Delta \dot{x} = \dot{x}_s - \dot{x}_u$$

The coefficients were determined to be the following:

$$k_1 = 0.0124 * 10^6$$
$$k_2 = -0.0737 * 10^6$$
$$k_3 = 3.1704 * 10^6$$

$$c_1 = 905.2896$$
$$c_2 = 254.2550$$

The spring and damping force data plots, including their respective least squares fits, are shown below.

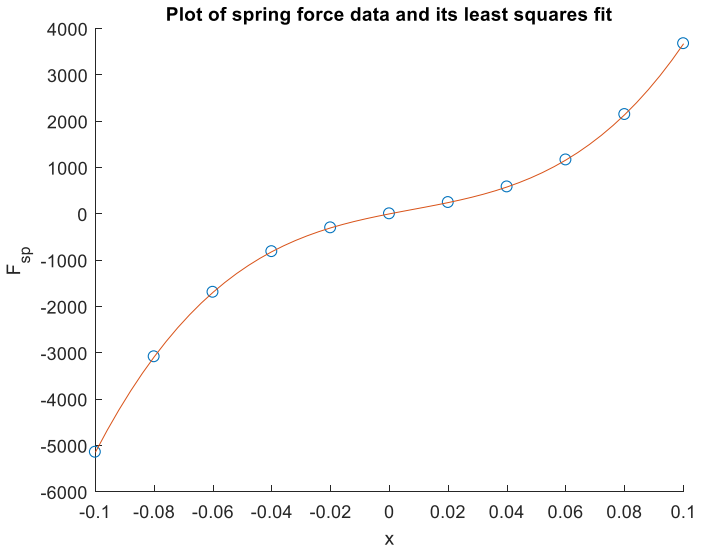


Figure 1. Plot of spring force data and its corresponding least squares fit

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

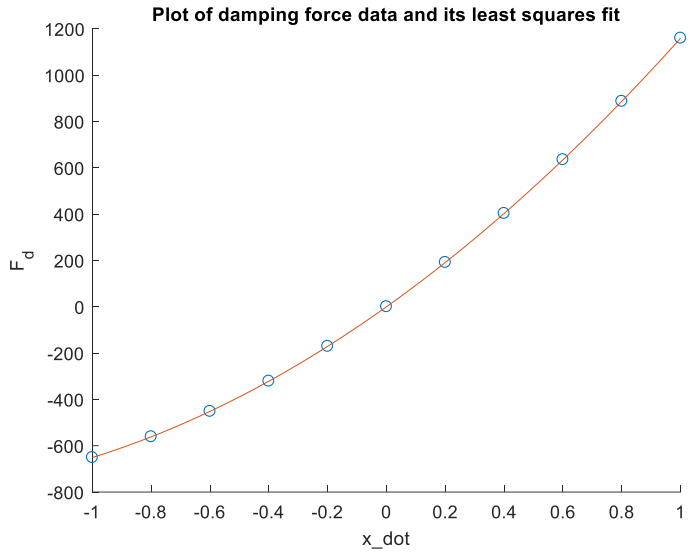


Figure 2. Plot of damping force data and its corresponding least squares fit

Part b)

The following 2-equation system of 2nd order ODEs describes the displacement, velocity, and acceleration of the quarter car model (QCM).

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

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

This system (equations 1 and 2) is converted into the following 4-equation system of 1st order ODEs:

$$let\ X_1 = x_s \ \ and \ \ X_2 = \dot{x}_s \ \rightarrow \ then \quad \dot{X}_1 = X_2 \quad and \quad m_s\dot{X}_2 + F_{sp} + F_d = 0$$

$$let\ X_3 = x_u \ \ and \ \ X_4 = \dot{x}_u \rightarrow \ then \quad \dot{X}_3 = X_4 \quad and \quad m_u\dot{X}_4 - F_{sp} - F_d = [k_t(q - X_3) + c_t(\dot{q} - X_4)]$$

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

Final 4-equation system of 1st order ODEs:

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = \frac{-F_{sp} - F_d}{m_s}$$

$$\dot{X}_3 = X_4$$

$$\dot{X}_4 = \frac{[k_t(q - X_3) + c_t(\dot{q} - X_4)] + F_{sp} + F_d}{m_u}$$

Part c)

See the attached 4th order Runge-Kutta and simulation MATLAB codes.

Section 2: Simulation

Part a)

See the attached 4th order Runge-Kutta and simulation MATLAB codes.

Characteristic time scale and timestep, h, used:

V = 10 km/hr: **T = 1.8720 seconds and h = T/100 = 0.0187 seconds**

V = 40 km/hr: **T = 0.4680 seconds and h = T/50 = 0.0094 seconds**

Part b)

The plots of displacements and velocities for the sprung and unsprung masses at velocities of 10 km/hr and 40 km/hr are shown below in Figure 3 and Figure 4, respectively. In Figure 5 and Figure 6, respectively, the displacement and velocity of the sprung mass are compared for V = 10 km/hr and V = 40 km/hr, **From Figures 5 and 6, observe that for the sprung mass displacement and velocity, the amplitudes are significantly greater for V = 40 km/hr than they are for V = 10 km/hr. Additionally, the periods of oscillation and times of decay are shorter for V = 40 km/hr than they are for V = 10 km/hr.**

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

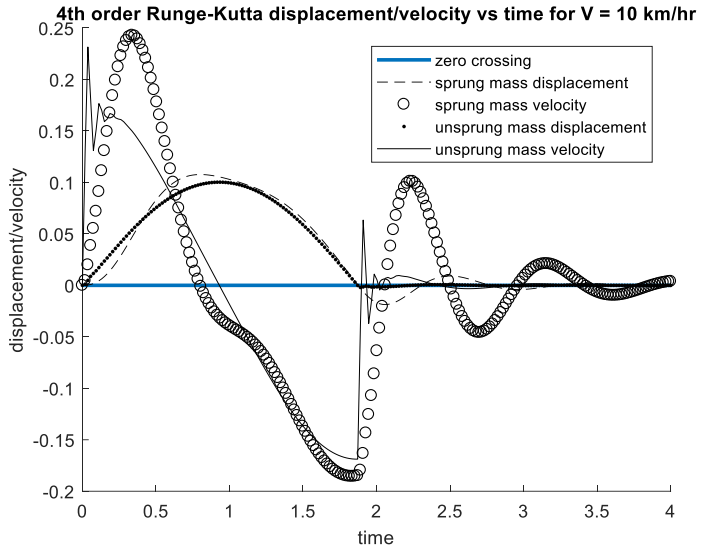


Figure 3. Displacements and velocities of sprung and unsprung masses versus time using 4th order Runge-Kutta for V = 10 km/hr

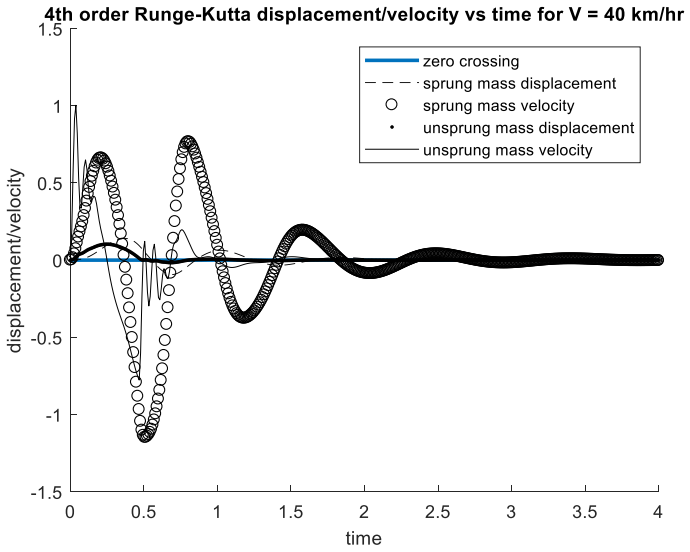


Figure 4. Displacements and velocities of sprung and unsprung masses versus time using 4th order Runge-Kutta for V = 40 km/hr

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

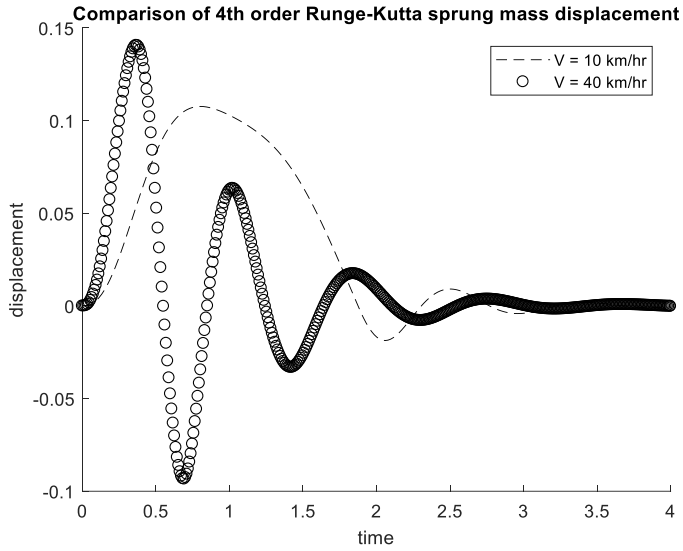


Figure 5. Comparison of 4th order Runge-Kutta displacement of sprung mass between V = 10 km/hr and V = 40 km/hr

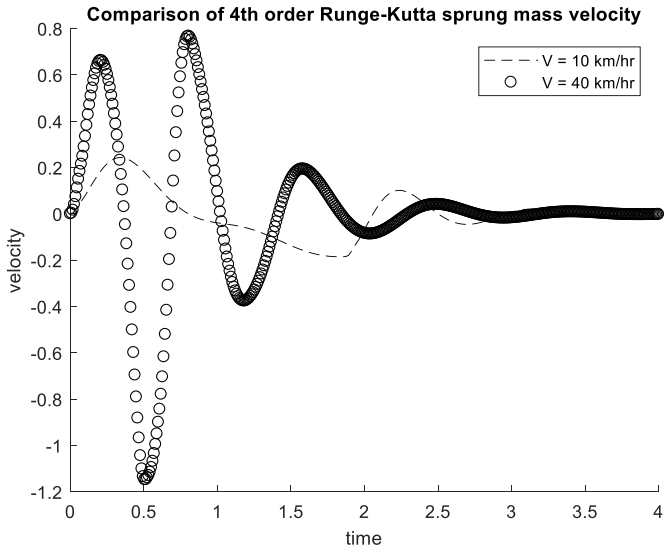


Figure 6. Comparison of 4th order Runge-Kutta velocity of sprung mass between V = 10 km/hr and V = 40 km/hr

Part c)

In the forward Euler method, the following calculations are required:

$$4 - equation\ system\ of\ 1st\ order\ ODEs\ to\ solve: \dot{X}_1, \dot{X}_2, \dot{X}_3, \dot{X}_4 \quad (from\ Section\ 1, part\ b)$$

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

$$f_j(t_i, x_i) = \dot{X}_j \quad for \quad j = 1{:}\,4$$

$$t_{i+1} = t_i + h$$

$$x_{i+1} = x_i + f_j(t_i, x_i) * h$$

See the attached forward Euler and simulation MATLAB codes.

The plot of the displacement and velocities of the sprung and unsprung masses at V = 40 km/hr computed using the forward Euler method is shown below.

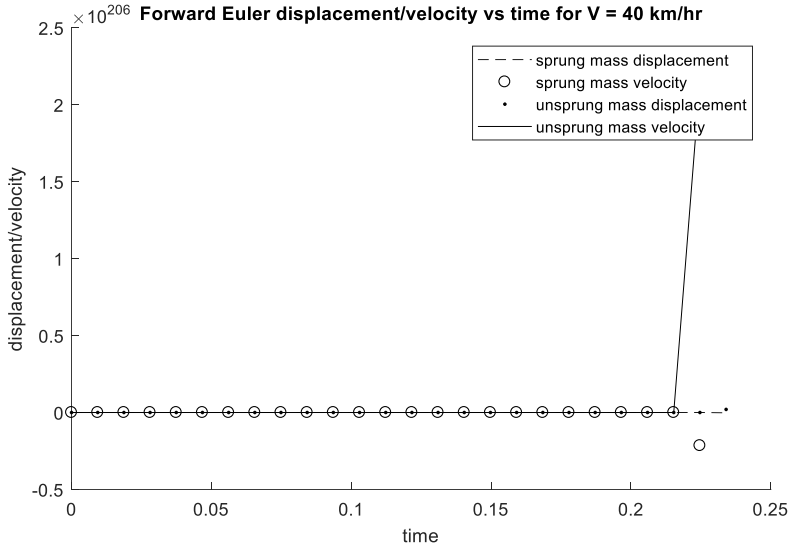


Figure 7. Plot displacements and velocities of sprung and unsprung masses versus time using forward Euler for V = 40 km/hr using a timestep, h, of T/50 = 0.0094 seconds (timestep used in Section 2, part a)

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

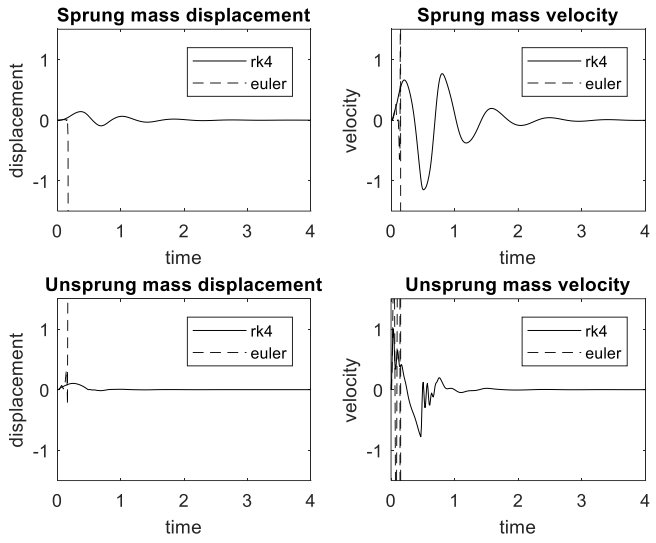


Figure 8. Comparison of displacements and velocities of sprung and unsprung masses between 4th order Runge-Kutta and forward Euler using a timestep, h, of T/50 = 0.0094 seconds (timestep used in Section 2, part a)

Using the timestep size used in Section 2, part a), at time = 2T, the absolute difference between the 4th order Runge-Kutta and the forward Euler method for each of the four unknown variables is:

Sprung mass displacement: NaN
Sprung mass velocity: NaN
Unsprung mass displacement: NaN
Unsprung mass velocity: NaN
where NaN is an extremely large number (similar to infinity)

In order for the above absolute difference to drop below 10^-3 for all four unknown variables, **the timestep, h, must decrease to approximately T/(50*37) = 0.000253 seconds.**

The total number of timesteps run for this case is:

(t_end - t0) / h = 4 ssec / 0.000253 sec = **~15810 time steps**

When utilizing the tic toc feature in MATLAB to time both RK4 and Forward Euler with ~15810 time steps **we can see that Forward Euler is computationally faster by 3 times** (elapsed time of 0.5429 vs 0.1837). This makes sense because Forward Euler only requires the calculation of one slope per step while RK4 requires the calculation of four slopes per step. This difference explains the faster computation speed of Forward Euler

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

Section 3. Data Analysis

Part a)

The sprung mass acceleration of the car at both velocities was computed via a O(h^2) centered finite difference method.

O(h^2) centered finite difference scheme:

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}$$

$$so, \quad \ddot{x}_s = \frac{x_s(t_{i+1}) - 2x_s(t_i) + x_s(t_{i-1})}{h^2}$$

The plots of the acceleration at V = 10 km/hr and V = 40 km/hr computed using the centered finite difference scheme are shown in Figure 9 and Figure 10, respectively.

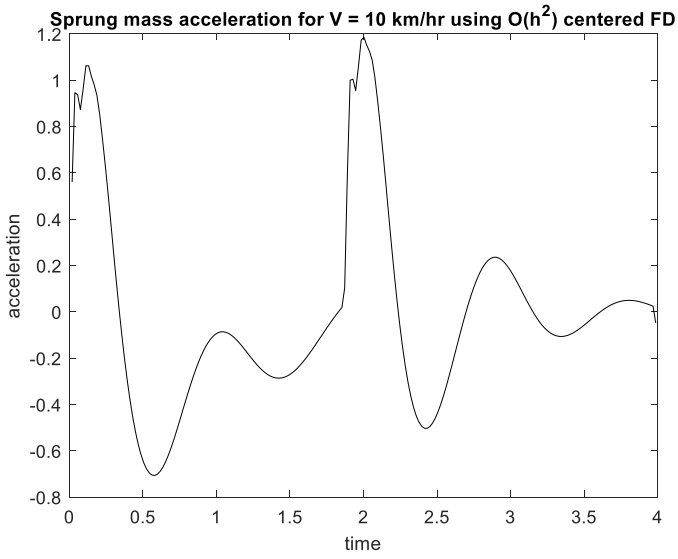


Figure 9. Plot of sprung mass acceleration versus time for V = 10 km/hr using the O(h^2) centered finite difference scheme

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

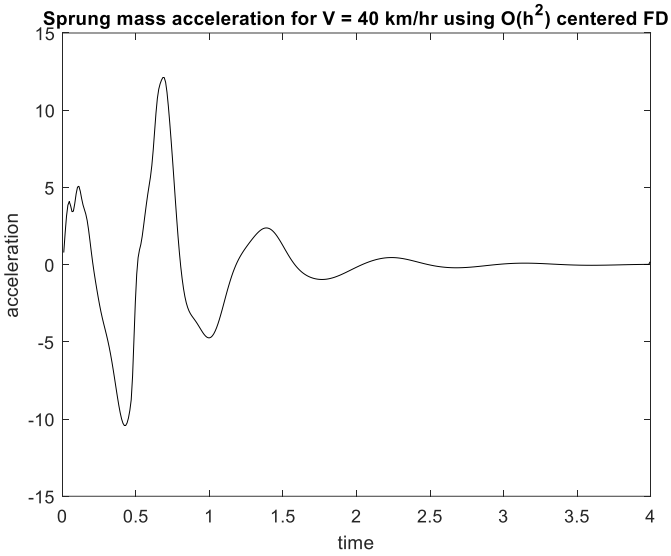


Figure 10. Plot of sprung mass acceleration versus time for V = 40 km/hr using the O(h^2) centered finite difference scheme

The maximum sprung mass acceleration at both car velocities computed with the centered finite difference method is shown in the table below. This table also includes the passenger comfort level associated with the acceleration.

V = 10 km/hr	V = 40 km/hr
Max acceleration = 1.1852 m/s^2	Max acceleration = 12.1177 m/s^2
Comfort level: Very uncomfortable	Comfort level: Extremely uncomfortable

Table 1. Maximum acceleration and comfort level for each velocity level using the O(h^2) centered finite difference scheme

Part b)

The sprung mass acceleration of the car at both velocities was computed via a O(h) forward finite difference method.

O(h) forward finite difference scheme:

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2}$$

$$so, \quad \ddot{x}_s = \frac{x_s(t_{i+2}) - 2x_s(t_{i+1}) + x_s(t_i)}{h^2}$$

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

The plots of the acceleration at V = 10 km/hr and V = 40 km/hr computed using the forward finite difference scheme are shown in Figure 11 and Figure 12, respectively.

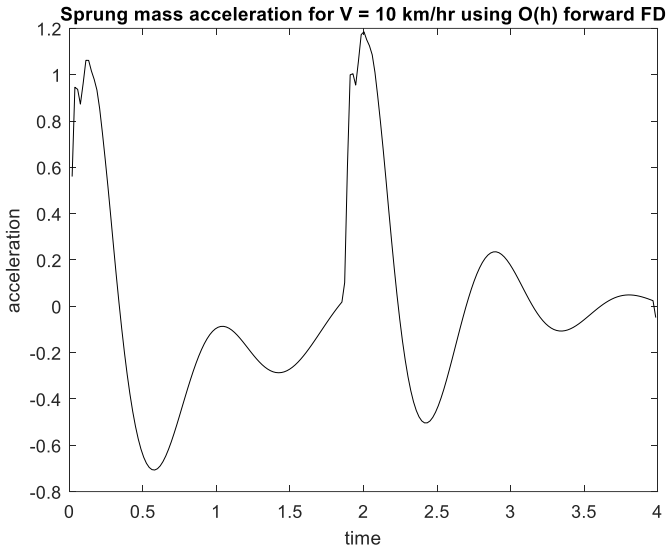


Figure 11. Plot of sprung mass acceleration versus time for V = 10 km/hr using the O(h) forward finite difference scheme

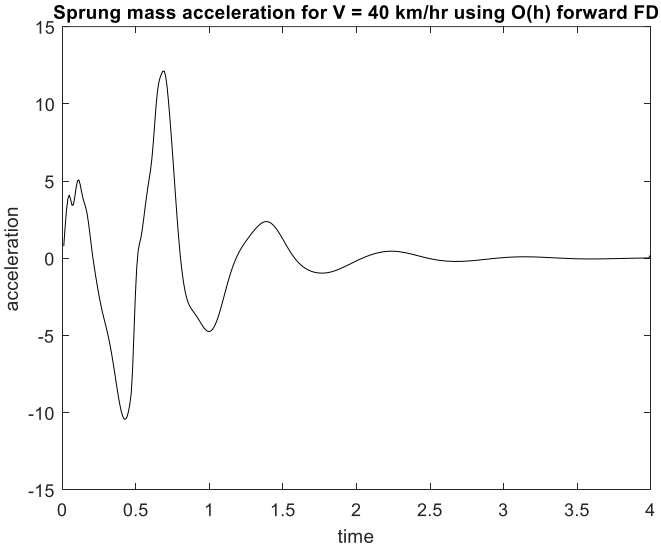


Figure 12. Plot of sprung mass acceleration versus time for V = 40 km/hr using the O(h) forward finite difference scheme

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

The maximum sprung mass acceleration at both car velocities computed with the forward finite difference method is shown in the table below. This table also includes the passenger comfort level associated with the acceleration.

V = 10 km/hr	V = 40 km/hr
Max acceleration = 1.1852 m/s^2	Max acceleration = 12.1177 m/s^2
Comfort level: Very uncomfortable	Comfort level: Extremely uncomfortable

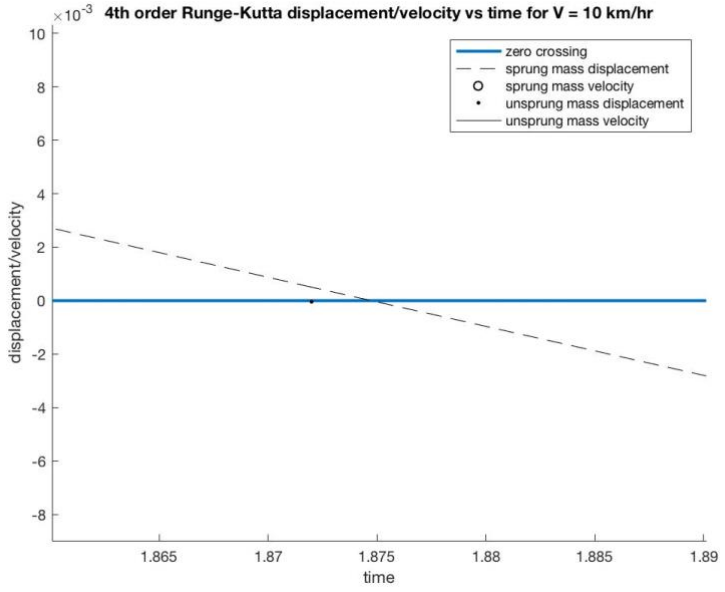
Table 2. Maximum acceleration and comfort level for each velocity level using the O(h) forward finite difference scheme

From Table 2, the maximum acceleration and comfort level is listed for each velocity. **Notice that the acceleration values and the comfort levels using the O(h) forward finite difference scheme are the exact same as the values and levels using the O(h^2) centered finite difference scheme.**

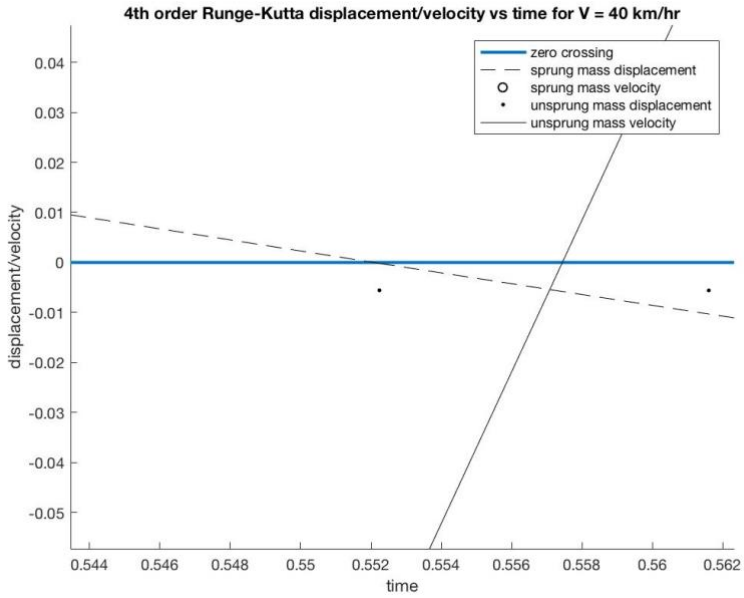
Part c)

For each of the two car velocities, the approximated location of the first zero-crossing time for the sprung mass displacement was determined visually. Construction of the fifth-degree interpolating Lagrange polynomial utilized six data points around this approximation. The bisection method was then used on the Lagrange IP find the more precise location of zero-crossing.

From RK4 graphs, **for V = 10km/hr, the first time of zero-crossing is approximately at t = 1.8748s. For V = 40km/hr, the approximate time is t = 0.5520s.**



Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18



To build the Lagrange interpolating polynomial, we first construct the six Lagrange basis functions for each of the six data points. They are construct such that it goes to zero everywhere except the corresponding point. These basis functions are multiplied to the function value at the corresponding point to form the Lagrange interpolating polynomial.

$$L_{5,0} = \frac{(t - t_1)(t - t_2)(t - t_3)(t - t_4)(t - t_5)}{(t_0 - t_1)(t_0 - t_2)(t_0 - t_3)(t_0 - t_4)(t_0 - t_5)}$$

$$L_{5,1} = \frac{(t - t_0)(t - t_2)(t - t_3)(t - t_4)(t - t_5)}{(t_1 - t_0)(t_1 - t_2)(t_1 - t_3)(t_1 - t_4)(t_1 - t_5)}$$

$$L_{5,2} = \frac{(t - t_0)(t - t_1)(t - t_3)(t - t_4)(t - t_5)}{(t_2 - t_0)(t_2 - t_1)(t_2 - t_3)(t_2 - t_4)(t_2 - t_5)}$$

$$L_{5,3} = \frac{(t - t_0)(t - t_1)(t - t_2)(t - t_4)(t - t_5)}{(t_3 - t_0)(t_3 - t_1)(t_3 - t_2)(t_3 - t_4)(t_3 - t_5)}$$

$$L_{5,4} = \frac{(t - t_0)(t - t_1)(t - t_2)(t - t_3)(t - t_5)}{(t_4 - t_0)(t_4 - t_1)(t_4 - t_2)(t_4 - t_3)(t_4 - t_5)}$$

$$L_{5,5} = \frac{(t - t_0)(t - t_1)(t - t_2)(t - t_3)(t - t_4)}{(t_5 - t_0)(t_5 - t_1)(t_5 - t_2)(t_5 - t_3)(t_5 - t_4)}$$

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

$$P_5(t) = L_{5,0}\,x_0 + L_{5,1}\,x_1 + L_{5,2}\,x_2 + L_{5,3}\,x_3 + L_{5,4}\,x_4 + L_{5,5}\,x_5$$

Where for V=10 km/hr, $t_0 = 1.8346$, $t_1 = 1.8533$, $t_2 = 1.8720$, $t_3 = 1.8907$, $t_4 = 1.9094$, $t_5 = 1.9282$ & $x_0 = 0.0074$, $x_1 = 0.0040$, $x_2 = 0.0005$, $x_3 = -0.0029$, $x_4 = -0.0061$, $x_5 = -0.0090$

Where for V=40 km/hr, $t_0 = 0.5515$, $t_1 = 0.5517$, $t_2 = 0.5520$, $t_3 = 0.5522$, $t_4 = 0.5525$, $t_5 = 0.5527$ & $x_0 = 0.006426$, $x_1 = 0.003651$, $x_2 = 0.000878$, $x_3 = -0.001894$, $x_4 = -0.004665$, $x_5 = -0.007433$

The bisection method was chosen, because this direct method guarantees convergence of the root solution. Additionally, it is the simplest in that we have already chosen an interval for the Lagrange IP and can visually tell where the point approximately is. Secant method does not make sense, because we just constructed a polynomial, and Newton-Raphson, while fast at converging, is unnecessary for this simple problem. With a tolerance of 10^{-3}, for V = 10km/hr, it took **4 iterations** to get to the solution t = **1.8895s** and for V = 40km/hr, it took **5 iterations** to get to the solution t = **0.5519s**.

Part d)

For each of the two car velocities, the energy loss due to damping by the shock absorber over the time interval, T, the car takes to go over the bump was computed via two numerical integration schemes: composite trapezoidal rule and 1/3 Simpson's rule.

Composite trapezoidal rule:

$$numerical\ integration\ value = I = \frac{h}{2}\left[\, f(x_0) + 2\sum_{i=1}^{N-1} \mathrm{f(x_i)} + f(x_N) \,\right]$$

1/3 Simpson's rule:

$$numerical\ integration\ value = I = \frac{b-a}{3N}\left[\, f(x_0) + 4\left(\sum_{i=1,3,5\ldots}^{N-1} \mathrm{f(x_i)}\right) + 2\left(\sum_{i=2,4,6\ldots}^{N-2} \mathrm{f(x_i)}\right) + f(x_N) \,\right]$$

And for both numerical integration schemes:

$$Energy\ loss\ due\ to\ damping: \qquad E_d = \int_0^{t=T} F_d \Delta\dot{x}\, dt$$

$$with \quad F_d = c_1\Delta\dot{x} + c_2\Delta\dot{x}^2 \qquad and \qquad \Delta\dot{x} = \dot{x}_s - \dot{x}_u$$

$$so, \quad f(x_i) = F_d(t_i) * \Delta\dot{x}(t_i) \qquad and \qquad I = E_d \qquad and \qquad a = 0 \qquad and \qquad b = T$$

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

The results from both numerical integration methods are shown in the table below. The relative difference in numerical integration value between the two methods is also included in the table.

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

	V = 10 km/hr	V = 40 km/hr
Composite trapezoidal rule	Ed = 4.0212	Ed = 92.4305
1/3 Simpson's rule	Ed = 4.0841	Ed = 92.3663
Relative difference	-1.5642%	0.0695%

Table 3. Energy loss due to damping at velocities of V = 10 km/hr and 40 km/hr using the composite trapezoidal rule and 1/3 Simpson's rule

The relative difference in the energy loss estimate between both numerical integration schemes was calculated using the following equation:

$$relative\ difference = \frac{E_{d,composite\ trapezoidal} - E_{d,\frac{1}{3}Simson's\ rule}}{E_{d,composite\ trapezoidal}} * 100\%$$

Observe that the use of the 1/3 Simpson's rule does not produce radically different results. Instead, both methods produce very similar approximations of the energy loss, as indicated by the relatively small relative difference values.

Note that both numerical integration schemes used computational, "experimental" data computed from the simulation phase of this project. Analytical equations for the damping force and the aggregate velocity of the car as functions of time were not available/used. The Gauss Legendre quadrature method requires the use of analytical, closed form expressions to compute the integration points. Due to the unavailability of such equations for the damping force and the aggregate velocity**, the Gauss Legendre quadrature method cannot be used to compute the integral for the energy loss due to damping.**

Pham, Jonathan
Sit, Hilarie
Due date: 5/17/18

Tasks performed by each team member:

Jonathan Pham

- Convert QCM equations into system of 1st order ODEs
- Runge Kutta simulation
- Finite difference
- Numerical integration
- Write-up

Hilarie Sit

- Least squares fit
- Runge Kutta simulation
- Euler simulation
- Zero crossing
- Write-up