# Project 3: Putting It All Together

**Due date: Last Day of Class**

## 1 Overview:

The goal of this project is to give you an opportunity to combine various numerical techniques you've learned this term to solve a practical problem in parameter estimation. The problem is this: from a small set of observations of the position of a spring, write software to estimate the spring parameters, and use your software to predict the position of the spring at a given time \( t \) in the future.

The suite of methods you bring to bear on this problem are directly relevant to a huge range of practical, real-world problems. If you understand how these methods fit together for this project, you will be in an excellent position to tackle a wide range of problems in science and engineering.

## 2 Details

### 2.1 Teams:

**Team 1:** Kyle, Janessa, Troy, Bryan  
**Team 2:** Lizzi, Alden, Luke, Cameron

### 2.2 The Goal:

The goal of this project is to write software that will fit a sinusoidal curve to a time sequence of position samples. The samples will represent the height of a mass attached vertically to a spring, and will be taken by an acoustic measuring device with a reasonable degree of accuracy. The period of the oscillation depends on the restoring force of the spring, the friction coefficient, and the mass of the object, while the rate at which the oscillations decay depends on the friction coefficient and the mass. Using a known mass, we will first collect data and use it solve for the friction coefficient and the restoring force constant of some particular spring. We will then then use these results as the starting point for our next task, which is to use this spring to estimate the mass of a different object. We do this by hanging this new object on the spring, taking more data, and using this new data (as well as existing estimates for the restoration and damping coefficients) to solve for the mass.

Your code needs to be able to:

- Use the data to generate an initial guess for the damping and friction coefficients
- Refine this initial guess via a Gauss-Newton type algorithm
- Predict the vertical position of the mass into the future by solving the force equation numerically.

Note that the second and third items above are related: you need the third to do the second.

### 2.3 Background Theory (Second Order Constant Coefficient ODEs):

Recall that Newton’s Law of Force states the force on an object is equal to its mass times its acceleration, i.e.

\[ F = ma. \]

In the case of an object oscillating vertically on a spring, the relevant forces are

1. gravity
2. the restorative force of the spring
3. the force of friction or air resistance.

If $y$ represents the vertical position of the mass when the spring is neither compressed nor extended, then Newton’s
Law gives the following ODE:

$$y'' + \frac{\rho}{m} y' + \frac{k}{m} y = -g.$$ 

Recall that the solution to this equation depends on the value of

$$\lambda_1 = \frac{-\rho + \sqrt{\rho^2 - 4km}}{2m}, \quad \lambda_2 = \frac{-\rho - \sqrt{\rho^2 - 4km}}{2m}.$$ 

**Case 1 (Overdamped):** $\rho^2 > 4km$: then $\lambda_1$ and $\lambda_2$ are both real, and the solution is

$$y(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t} - \frac{g}{b}.$$ 

**Case 2 (Critically damped):** $\rho^2 = 4km$: then $\lambda_1 = \lambda_2 \equiv \lambda$, and the solution is

$$y(t) = c_1 e^{\lambda t} + c_2 te^{\lambda t} - \frac{g}{b}.$$ 

**Case 3 (Underdamped):** $\rho^2 < 4km$: then $\lambda_1$ and $\lambda_2$ are complex conjugates, given by

$$\lambda_{1,2} = -\frac{\rho}{2m} \pm \frac{\sqrt{4km - \rho^2}}{2m},$$ 

and the solution is

$$y(t) = e^{\alpha t} (c_1 \cos \omega t + c_2 \sin \omega t) - \frac{g}{b},$$ 

where

$$\alpha = -\frac{\rho}{2m}, \quad \omega = \frac{\sqrt{4km - \rho^2}}{2m}.$$ 

All of the mass/spring combinations that we look at in this project will fall into **Case 3**, i.e. the mass will oscillate at least several times.

### 2.4 What you will be given and what you need to produce

As a class we are going to take measurements of the height of a known mass attached to one particular spring. You will receive these measures as a Matlab file (a .mat file). The measurements will be in an $n \times 2$ matrix, with rows representing time/height pairs. You need to use this data to generate preliminary guesses for $\rho$ and $k$, and then to refine these guesses via Gauss-Newton.

Will will then take measurements of an unknown mass on the same spring. You’ll get the data in the same form as before. This time, you will need to use the data to provide the best estimate of the mass. You’ll want to use your previous estimates for $k$ and $\rho$ to make this estimate.

I want you to turn in

1. Your code: it should be stored in its own folder, and be easy for me to figure out how to run it. (A README file is highly recommended.) Try to make your code as modular as possible, i.e. divide things into functions that can be independently tested and verified.

2. A project report: it should explain how your method works, and how successful it is solving test problems. (You will need to generate these test problems yourself, of course.) Comment on notable difficulties and notable successes.