

# Spring-Mass System - Euler Integration

Author: Rajesh Bhaskaran, Cornell University

[Problem Specification](#)

[1. Euler Integration](#)

[2. Array Pre-Allocation](#)

[3. Plotting](#)

[4. Function Creation](#)

[5. Structure Creation](#)

[Exercises](#)

[Comments](#)

## Step 1: Euler Integration

We start by specifying constants such as the spring mass  $m$  and spring constant  $k$  as shown in the following video.

We next specify the initial conditions and run the code that we have so far as shown in the video below. While running the code, we use the debugger to step through statements to understand what they do.

We have seen students write up huge chunks of code at one go that produces some obscure error. Students find that no amount of cursing and swearing at the computer fixes the error. To avoid this trap, we will build our code incrementally, testing at each stage. We first implement the Euler's integration method for *one* time-step as shown below and then will extend it to multiple time-steps.

We move on to extend our code, or script in MATLAB lingo, to perform the Euler integration over multiple time-steps by looping over the appropriate statements.

Our script saves only the final location of the mass. We are also interested in saving the intermediate locations so that we can make a plot of position vs. time. We do this by creating arrays for the position and time.

[Go to Step 2: Array Pre-Allocation](#)

[Go to all MATLAB Learning Modules](#)