Modeling, Simulation and Analysis
CS 250
Spring 2018
Lab 1: Runge-Kutta 4

The purpose of this lab is to get practice implementing the Runge-Kutta 4 simulation method. For some of this, it is not immediately obvious how to do the implementation—that's why these are lab exercises rather than HW exercises, so I can help with every detail, and you can talk with each other about your code! (You will implement RK4 on your next HW assignment, too, with the standard conditions of Academic Integrity for coding HWs.)

1. Read about *anonymous functions* in Matlab. The **Additional Notes / Readings** page on our course website contains a link to a page of documentation about anonymous functions in Matlab, but you are also encouraged to do a web-search on the topic, to see what information is available.

2. Implement a Runge-Kutta 4 simulation of the *undamped weighted spring* system in our textbook (Chapter 3.2, pages 79–83). In particular, the purpose is to implement the full RK4 method yourself, including the four approximations needed for this second-order dynamical system, rather than using a pre-defined RK4 library.

   You should also implement an Euler's Method simulation of the system first, both to gain understanding of the system and to be able to compare Euler's Method and RK4 on this system. The real purpose of this exercise, though, is the RK4 simulation, not any Euler-based simulation.

   Recall from our lecture notes that RK4 uses functions $f(\cdots)$ to compute derivatives as part of the calculations. For your implementation, use anonymous functions to implement those derivatives—e.g., `dVdt = @(···) ...;` for the time-derivative of velocity $\frac{dv}{dt}$—and then use them in computing the four RK4 approximations ∂_i. (You will also be asked to do this on HW3.)

   Note that the intention of this exercise is not to implement just any undamped weighted spring system—the goal is to implement the specific one presented in the textbook (and in our course Lecture Notes). For your convenience, Table 1 on the next page contains the parameters and equations needed for the system.

   Please be sure to simulate the system (using RK4) with timesteps $dt = 0.2$ and $dt = 0.02$, just to see what the difference is. (You should notice a major difference!) Please also run your Euler simulation with those timesteps—that could be a good illustration of the difference between the relative error of Euler and RK4!

3. The **Additional Notes / Readings** page on our course website also contains a link to a document (on a Davidson College website) containing a brief summary of how to apply RK4 to second-order differential equations. Although I suspect it is too concise to make doing this lab a straightforward task, it may serve as a good reference for later simulations involving RK4.

Description	**Component of System**
gravitational constant	$-9.81\frac{m}{s^2}$ (be careful of the sign!)
spring constant	$k = 10\frac{N}{m}$ (where 1 $Newton = 1N = \frac{kg \cdot m}{s^2}$)
mass of weight W	$m = 0.2kg$
unweighted length of spring	`unweighted_length` $= 1m$
initial displacement of spring	`init_displacement` $= 0.3m$
initial length of spring	`unweighted_length` + `weight_displacement` + `init_displacement`
displacement (from unweighted length)	$s =$ ((current length) - `unweighted_length`)
equation for gravitational force	$F = mg$ (be careful of the sign of g!)
equation for restoring force	$F = -k \cdot s$
weight displacement	`weight_displacement` (= $\frac{-mg}{k}$ ... do you see why?)
acceleration	$a = \frac{\text{sum of forces}}{m}$
differential equation of acceleration (of spring's current length)	$a = \frac{dv}{dt}$ (where v is velocity, described below)
differential equation of velocity (of spring's current length)	$v = \frac{dp}{dt}$ (where variable p is for spring length)

Table 1: Table of values and equations for the undamped weighted spring simulation.