

Lab #3a -- Programming with Matlab

Overview

This lab will focus on the use of **function subprograms** and **conditionals** in Matlab. These coding features are common to all programming languages and are two of the key topics that were illustrated as part of our class lessons on Programming in Matlab (in Lesson #3). This lab will allow you to put the in-class discussions into practice with two separate student-oriented programming tasks. There will be no instructor-led problems this time but, instead, lots of individual help will be available, as needed -- so don't hesitate to get your questions resolved during the lab...

By the end of this lab you should have sufficient background to tackle the programming tasks requested as part of HW3a, and be well on your way to becoming competent with a variety of programming features with Matlab...

Get started writing some Matlab scripts and functions of your own

In the time remaining in this lab (after the initial quiz), you should do your best to solve the following two programming problems. Both tasks highlight the use of function subprograms in Matlab, and the second one also introduces the use of conditional tests in Matlab. Be sure to follow the special instructions that are given, since the goal here is to illustrate several different options available when using functions and conditional tests. Also be sure to complete these tasks outside of class if you run out of time during the lab period.

Problem 1 -- Air Properties

The temperature dependence of the density, specific heat at constant pressure, dynamic viscosity, and thermal conductivity of air can be approximated with the following relationships:

$$\rho = \frac{P}{RT} \quad \text{air density (kg/m}^3\text{)}$$

$$c_p = a_1 T^3 + a_2 T^2 + a_3 T + a_4 \quad \text{specific heat (J/kg-K)}$$

$$\mu = b_1 T^3 + b_2 T^2 + b_3 T + b_4 \quad \text{viscosity (kg/m-s)}$$

$$k = c_1 T^3 + c_2 T^2 + c_3 T + c_4 \quad \text{thermal conductivity (W/m-K)}$$

where the coefficients for the polynomial fits are given in the table below, R is the gas constant for air ($R = 287.0$ J/kg-K), and P and T are the absolute pressure and temperature, respectively.

i	a_i	b_i	c_i
1	-2.455322455e-7	2.156954157e-14	-2.486402486e-12
2	6.701631702e-4	-5.332634033e-11	-2.871794872e-8
3	-2.992579643e-1	7.477905983e-8	9.629059829e-5
4	1.042503030e+3	2.527878788e-7	2.060606061e-5

- Write a Matlab function file to compute the above four air properties versus temperature and pressure (note that c_p , μ , and k are not functions of pressure). The function should take a

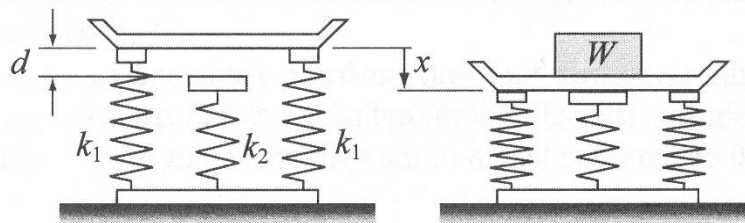
vector of temperatures (K) and a **single pressure (N/m²)** as input, and pass back four vectors containing the values of ρ , c_p , μ , and k based on the above equations. For consistency, the first line of your function file should be

function [rho,cp,mu,k] = air_props(T,P)

- b. Once your function file is working properly, write a main program that uses it to generate ρ , c_p , μ , and k for a range of temperatures from 0 C to 200 C in increments of 20 C for $P = 1$ atm. As output, the main program should create a formatted table of results using a series of Matlab's *fprintf* commands, and also generate a 2x2 set of subplots to display the temperature variation of these specific properties for air at atmospheric pressure. Be sure to properly label both the plot and the table.

Problem 2 -- Weight on a Scale

A scale is made of a tray attached to springs as shown in the sketch. The springs are linear, with the restoring force being proportional to the spring displacement. When an object is placed on the tray, the tray moves down and the weight of the object can be determined from the displacement of the tray. For a small weight, W , only the two outer springs support the object. However, if the object is heavy enough, the tray makes contact with the third spring in the middle of the scale.



Using the following values for a particular scale,

$$k_1 = 800 \text{ N/m}, \quad k_2 = 1700 \text{ N/m}, \quad d = 2 \text{ cm}$$

perform the following:

- Write a user-defined function that computes the weight W of the object for a given displacement, x , of the tray where the function only allows scalar values for the displacement x . Call this function **weight_1s.m.**, where the **s** denotes **scalar-only** capability.
- Write a similar user-defined function to compute the weight W of the object where, this time, the function allows for x to be a vector of displacement values. Call this function **weight_1v.m.**, where the **v** denotes **vector** capability.
- Now write a script file that calls both of your functions from Parts a and b and plots the weight of the object as a function of displacement over the range $0 \leq x \leq 4$ cm. Plot both curves for W vs. x on the same plot using different symbols and/or line styles so that they can be easily distinguished. Do the curves overlap as they should? Is it clear how the main program differs depending upon whether or not the function file allows vector capability? Explain...