Lab #5 -- Root Finding Methods/Applications and Working with Polynomials in Matlab

Overview

We have briefly discussed bracketing methods, open methods, and Matlab's hybrid ***fzero*** routine in class -- now, in this lab, you will get a chance to apply these methods for a couple of different application problems. The first problem asks you to compare the three methods (using the ***bisection***, ***secant***, and ***fzero*** routines) to illustrate their basic use and relative efficiency on a simple example involving the relationship between the volume and radius of a specific tank geometry. The 2nd problem, however, simply focusses on finding the solution to a practical engineering application. In situations like this, I recommend that you use Matlab's ***fzero*** routine, since this function employs an industry-standard robust solution algorithm -- whereas the other two routines discussed in class were used primarily to illustrate the basic concepts associated with bracketing and open methods for solving nonlinear equations. Finally, the 3rd demonstration problem simply illustrates several of the built-in Matlab functions for working with polynomials -- which is a very common task required for many engineering analyses.

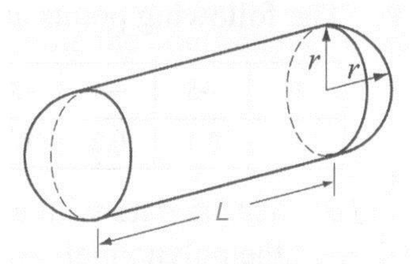
Upon completion of this lab you should be more comfortable with solving nonlinear equations and working with polynomials in Matlab. Several of the topics addressed here are nearly identical to the subjects addressed as part of your next HW assignment -- so make sure you get a good handle on these topics before leaving the lab session and before starting your next HW…

Problem 1 -- Comparison of Different Root Finding Methods (Volume of a Tank)

A gas tank is shaped as a cylinder with hemispherical caps at the ends as shown in the sketch. The radius of the cylinder is r and the length is L. The volume of the tank is given by

$$V = \pi r^2 L + \frac{4}{3}\pi r^3$$

With r and L given, this expression gives a straightforward ***explicit*** relationship for V as a function of r. However, if r is the unknown, for a given V, then the above equation represents an ***implicit*** equation for the radius, r, as a function of V.



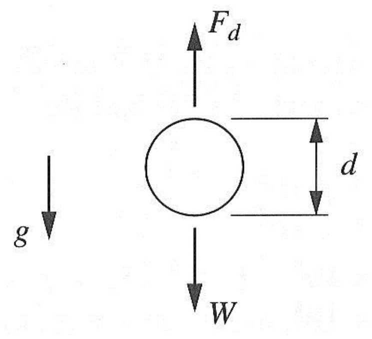
To illustrate your understanding of the difference between explicit and implicit equations, answer the following questions for the given system:

a. If r = 3 ft and L = 7 ft, what is the volume of the tank in gallons?

b. For the same length, what tank radius is needed to give a volume of 2500 gallons? As a comparison of the various root-finding methods discussed in class, you should use the ***bisection***, ***secant***, and ***fzero*** routines to answer this question. Do the solutions to the classical textbook methods (bisection and secant techniques) agree with Matlab's more sophisticated ***fzero*** routine?

c. Now, compare the relative efficiency of the various methods by tabulating the number of function evaluations needed to obtain the desired solution to Part b. Are your results as expected based on your knowledge of the different techniques?

Problem #2: Terminal Velocity of a Falling Sphere



The sketch to the right depicts a solid sphere of diameter d falling through the air. After its release, the sphere accelerates until it reaches its terminal velocity. The terminal velocity is obtained when the drag force due to air friction, F_d, balances the weight of the sphere, W.

The weight of the sphere is simply its mass, m, times the gravitational acceleration ($g = 9.8\ m/s^2$),

$$W = mg \qquad (1)$$

and the drag force can be computed as

$$F_d = c_d \frac{1}{2} \rho v^2 A \qquad (2)$$

where c_d is a dimensionless empirical drag coefficient, ρ is the density of the air, v is the velocity of the sphere, and $A = (\pi d^2)/4$ is the frontal area of the falling sphere.

A common correlation for the aerodynamic drag of smooth spheres is

$$c_d = \frac{24}{Re} + \frac{6}{1+\sqrt{Re}} + 0.4 \qquad \text{for} \quad Re \le 2 \times 10^5 \qquad (3)$$

where Re is the Reynolds number,

$$Re = \frac{\rho v d}{\mu} \qquad (4)$$

and μ is the dynamic viscosity of the air.

Now, since the properties of air (ρ and μ in this application) vary with temperature, the terminal velocity will also be temperature dependent. To investigate this dependence, we will assume that air is an ideal gas, giving

$$\rho(T) = \frac{P}{RT} \qquad (5)$$

for the desired density (in kg/m^3) vs. temperature relationship, and we will use a cubic polynomial fit for the variation of dynamic viscosity (kg/m-s) with temperature, or

$$\mu(T) = b_1 T^3 + b_2 T^2 + b_3 T + b_4 \qquad (6)$$

where the coefficients for the polynomial fit are b_1 = 2.156954157e-14, b_2 = -5.332634033e-11, b_3 = 7.477905983e-8, and b_4 = 2.527878788e-7, R is the gas constant for air (R = 287.0 J/kg-K), and P and T are the absolute pressure (in N/m^2) and temperature (in K), respectively.

a. Assuming standard atmospheric pressure ($P = 101300\ N/m^2$), m = 0.5 kg, and d = 15 cm, compute and plot the terminal velocity of the sphere over the temperature range from -60 C to 60 C.

b. Once you have v vs. T, you can re-compute several intermediate variables to help you rationalize that your solution makes physical sense. In particular, you should also generate a series of curves (possibly in subplots) for ρ vs. T, μ vs. T, Re vs. T, and c_d vs. T. Are all these plots reasonable? Do they help justify the behavior observed for v vs. T in Part a?

Note: In many engineering analyses the available correlations often don't cover the full range of interest. In these cases, when the needed data are incomplete, we often use the available correlations anyway, but assign a larger uncertainty to the overall analysis because of the increased uncertainty in the empirical correlations. In the current problem, you should use eqn. (3) for the drag coefficient for all Re values, and address how the computed results might vary based on the actual computed Re values. Within this context, you might check out a Fluid Mechanics text for a discussion of flow over smooth spheres…

Problem #3: Polynomial Operations in Matlab

a. Using Matlab's ***polyval*** function, evaluate and plot the following polynomial:

$$f(x) = x^6 - 8.25x^5 + 33.625x^4 - 109.75x^3 + 189.75x^2 - 307x + 285 \quad \text{for } 0 \le x \le 5$$

How many real roots are there? What does this tell you about the number of complex roots?

b. Use Matlab's ***fzero*** routine to find all the **real roots** of this polynomial. Explain the procedure used.

c. Using Matlab's ***conv*** function, multiply all the linear factors together to find a new polynomial. Now, using ***deconv***, divide the original polynomial by the new polynomial. What is the polynomial that remains after performing this operation? It should be a factor of f(x), which suggests that the remainder should be zero -- is this true?

d. Using Matlab's built-in ***roots*** command, find the roots of the polynomial determined in Part c. Are these consistent with your answer to Part a (i.e. how many complex roots do you get)?

e. Finally, as a check on the various manipulations done here, simply use ***roots*** to find all the roots of the original polynomial given above. Is everything self-consistent?

Note: The purpose of the above manipulations is to get you familiar with some of the Matlab functions that work with polynomials. Do all the computational components for this problem as part of a single Matlab file. Make sure you follow the instructions given above carefully and, in the end, everything should make sense and indeed be self-consistent…