MATH 340 - Lab Instructor: Valeria Barra
LAB 2 Assignment
DUE Tuesday 02-02-2016

Root-finding: Bisection Method

As outlined in your textbook, the bisection method serves to find a zero of a real-valued function $f(x)$. The first thing to keep in mind is that we want to solve an equation of the form $f(x) = 0$ for x. Therefore if the function is not given in that form, you should recast it to have everything on one side. For instance, if you are asked to find the root of $cos(x) = x$, you should write it as $f(x) = cos(x) - x$ in order to solve the equation $f(x) = 0$ for x.

This being said, your book uses the notation $[a, b]$ for the interval in which we are finding the root. Call this root r, such that $f(r) = 0$. The main assumption under which the method works is that the function has a change of sign at the endpoints of the interval, therefore a zero in between. This is translated in the condition `f(a)*f(b)<0`.

A *pseudo-code* to implement this method is briefly described by the following steps:

Given the initial interval $[a, b]$ s. t. $f(a)f(b) < 0$
`while (b-a)/2 > tol`

1. Define $c = (a + b)/2$. % this is our approximation to the root r
2. `if f(c) == 0`, stop, end % in this case we are lucky, we've already found r
3. `if f(a)f(c)<0`,
4. `b=c`
5. `else`
6. `a=c`, end

`end` % end of while loop

Again, update the approximate solution $c = (a + b)/2$.

Problem

Write your own code for the bisection method. In addition to what the pseudo-code provided above does, it is worthwhile to count how many steps it takes to converge to the solution (hint: to keep track, increase a counter variable n in the loop, with the instruction `n = n+1`). At each step, print out the interval endpoints, the midpoint, and function values at the endpoints and the approximate root (the midpoint). Also, print out a message to say whether you have found the solution according to the condition in the step no. 2, or your best approximation after the loop has ended.

Now use your code to solve the problems below. For each one write a sentence or two conclusion as appropriate. For example,"My code found the root to be ... after ... steps to within an error of ... in x and the function value at this value of x is ... This seems to be as expected as the error theory implies it should take about ... steps". Of course, if something different occurs, youd want to highlight that aspect.

1. Find an interval of length 1 with integer endpoints on which a solution to $x^3 = 9$ lies. Then use your bisection code starting with this interval to find the solution to within 10^{-4}.

2. Find an interval of length 1 with integer endpoints on which a solution to $6 + \cos^2 x = x$ lies. Then use your bisection code starting with this interval to find the solution to within 10^{-4}.

3. Suppose we want to know what interest rate we would need to achieve (assuming constant interest rates) such that the initial deposit of $\$50,000$ and annual contribution of $\$10,000$ will grow to $\$1,000,000$ in 20 years. Thus we need to solve $1000000 = \left(50000 + \frac{10000}{r}\right) e^{20r} - \frac{10000}{r}$ for the interest rate, r. Find a reasonable interval on which the solution ought to lie and use the bisection code to find r to within 10^{-6}.

4. Use your bisection code to approximate a root of $f(x) = \frac{|x-2|}{x^3-2x^2+x-2}$ on the interval $[1, 4]$ with error no more than 10^{-4}.

5. Use your bisection code to approximate a root of $f(x) = \frac{|x-2|}{x^3-2x^2+x-2}$ on the interval $[4, 6]$ with error no more than 10^{-4}.

6. Find appropriate intervals and use your bisection code to find ALL the root(s) of $|x|e^x = 0.25$

7. Use your bisection code to find the root of $27x^3 - 27x^2 + 9x = 1$ on $[0, 1]$ to within 10^{-6}.

8. Use your bisection code to find the root of $64x^3 - 48x^2 + 12x = 1$ on $[0, 1]$ to within 10^{-6}.

MATLAB hints:

For this assignment, since you have to apply your bisection scheme for multiple problems, you may want to write your bisection method in a separate MATLAB `function`, and call it in turn with the different data given. The signature of you `function` may look like:
function [r] = YourBisectionMethod(f, a, b, tol)

To pass a function $f(x)$ to your bisection method, in MATLAB you want to use a function handle, that is a symbolic expression of a dummy variable. Example:
`f = @(t) (t^2);` % this defines a symbolic function of the dummy variable t, and it tells MATLAB to square its argument

When you want to evaluate a function defined via a function handle at a specific point a, you simply use the math-like syntax `f(a)`.