MTH 351 – Lab 3

1. Consider the function $f(x) = x^5 - x^4 + x - 1$. Use the *Bisection method* to approximate the root to within an error tolerance of 10^{-6} with the following initial intervals: $[0, 3], [0.5, 2.0], [0.9, 1.2]$. In doing so, make a table of values which includes the approximation, error (what is the true root anyway?), and number of iterations (use a maximum of 100 iterations). Then answer the following questions:

   (a) Why does the second interval need exactly one fewer iteration?

   (b) Is there any advantage to having the root in the center of the interval, or is it better to instead be nearer to an endpoint?

2. Consider the function $f(x) = x^5 - x^4 + x - 1$. Use *Newton's method* to approximate the root to within an error tolerance of 10^{-6} with the following initial iterates: $-100, 0, .9, 0.99, 1.1, 1.4, 1000000$. In doing so, make a table of values which includes the approximation, error, and number of iterations. Then answer the following questions:

   (a) How does Newton compare to Bisection for efficiency when the initial iterate is close to the true root?

   (b) Why are the errors less than 10^{-12} if we only asked for 10^{-6}?

   (c) Without running Bisection, how many iterations would it take if the interval were $[-1000000, 1000000]$ and $\epsilon = 1.3 \times 10^{-12}$? Compare to Newton's result from $x_0 = 1000000$.

3. Consider the function $f(x) = x^5 - x^4 + x - 1$. Use the *Secant method* to approximate the root to within an error tolerance of 10^{-6} with the following initial iterates: $[x_0, x_1] = [0, 3], [0.5, 2.0], [0.9, 1.2]$. In doing so, make a table of values which includes the approximation, error, and number of iterations. Then answer the following questions:

   (a) How does this method compare to Newton and Bisection for efficiency when the initial iterate is close to the true root?

   (b) Does the distance between the initial iterates affect the number of iterations required as directly as does the size of the initial interval in Bisection?

4. Explain the output of `roots(poly(1:21))`. Hint: Recall Problem 11 from Section 2.3 of the book.

5. On your own (do not turn in), repeat 1-3 above for $f(x) = x^5 - 3x^4 + 4x^3 - 4x^2 + 3x - 1$, but answer the following questions instead:

   - *Bisection method*
     (a) Why are the number of iterations the same as for the first function?
     (b) Why is the actual error not less than the theorectically *guaranteed* error bound?!
     (c) See if writing the function a different way helps, e.g., reverse order, nested, factored (although, if we knew the factorization in advance it wouldn't be much of a root finding problem!).
   - *Newton's method*
     (a) How does Newton compare to Bisection for efficiency of solving this problem?
     (b) Why might Newton be having a problem solving for the root of this particular function and what could you do to fix the problem?
   - *Secant method*
     (a) How does this method compare to Newton and Bisection for efficiency when the initial iterate is close to the true root?
     (b) Assuming you are only using Secant method to find the roots of a function because you do not have a way to compute derivatives (and therefore cannot use Newton's method), would there be any way to fix the type of problem you are observing here?

6. On your own (do not turn in), solve for the roots of the above two functions with Matlab's `fzero` (see `help fzero` for a description and `type fzero` to see the source). The algorithm essentially does something similar to what we discussed in class in that instead of bisecting an interval, it divides it in an intelligent way based on $f(a)$ and $f(b)$ (actually it uses a "combination of bisection, secant, and inverse quadratic interpolation methods"). To which method above are the results the most similar? Does it do as well as Bisection, or Secant? *Hint: to set tol and max_its use:* `options=optimset('MaxIter',max_its,'TolX',tol);` *to call the function use:*

```
[rootiF,fval,flag,output]=fzero(fcn,interval,options);
itiF=output.iterations;
```