CS206 Assignment #5 - Newton's Method in one dimension (due Week 7, Tue)

You can use any language, but the "software engieneering" aspects must be respected. Submit a PDF to GradeScope, and your code to openlab using the submit command.

1. Create a function called `MyZero(double x0, function f, function fp, function fpp, int stopMeth)`: it takes argument x_0 (the starting guess, provided by you), the name of the function f whose zero we are trying to find, f', a function that returns the first derivative of f, f'', an optional function that returns the second derivative of f (if you don't include it, then set tthe function pointer to NULL or provide some other method of indicating it's not provided); and an integer specifying which stopping criterion to use (you can define what value you accept and what they mean, but at the very least you must support the value stopMeth=0, which means "default" and can be your best-guess stopping criterion). So for example I have written mine in C, and here are some lines of my C code:

```
typedef double (*pMathFunc)(double x); // pointer to a math function
double myf(double x) { return sin(x);} // my function f
double mydf(double x) { return cos(x);} // its derivative f'
[some code deleted]
// Here is the function prototype for my zero finder.
// It returns its best guess at the value of x where f(x)=0.
double MyZero(double x0, pMathFunc f, pMathFunc fp, int stopMeth);
[more code deleted]
  // we are now inside my main() C program
  // and here is how you call my Zero function with a starting guess of x0=1
  double xStar = MyZero(1, myf, mydf, 0);
  printf("Approx zero at x*=%g; f(x*)=%g\n", xStar, myf(xStar));
```

   Once you have written your `MyZero` function, it should be very easy to call it with various different functions. We do *not* want to have to modify the code of Zero, or anything it calls, just because we change the function who's zero we want to find. This is an important *software engineering* issue that applies to Scientific software just as with any other piece of good software: you should write good code that is generally applicable, that as much as possible does not need to be re-written just because you want to solve a different problem than what you had previously written it for.

   For this assignment you may assume that the starting guess is always $x_0 = 1$.

2. To emphasize the importance of writing general-purpose scientific software, once you have written the above MyZero function, test it with a bunch of different functions $f(x)$, such as $\sin(x)$, $\cos(x)$, $\tan(x)$, $\log(x)$, x^2, x^3. (Why don't we want to try it for $\exp(x)$?). Remember that if you have written `MyZero` properly, you should not have to change any of the code inside of it when the function f and its derivative f' are provided, it just calls those functions directly to produce its iteration.

3. Try several of the stopping criteria we mentioned in class. Describe how each one behaves, and which you chose to use in your code. How many iterations does it take to converge for each of the above functions, for each stopping criterion?

4. Now try the function $f(x) = \cos(x) + 1$. What do you observe in terms of the number of iterations it takes to converge, compared to $f(x) = \cos(x)$?

5. **Root Multplicity** Now you are going to take into account the order of convergence by estimating the root multiplicity as I described in lecture. Compute m, the estimated multiplicity—you should compute it as a floating point number, not an integer, since it's only an estimate. Note that to do this, you'll need to provide an actual implemented function that computes f'', the second derivative $f''(x)$; as with the f and f', you can use an analytic expression (eg if $f(x) = \sin(x)$ then your second derivative function would return $-\sin(x)$.). Then, include the value of m in the Newton iteration as I described in class, and describe the effect on the convergence of Newton's method when $f(x) = 1 + \cos(x)$. Also compare the convergence speed and the estimated value of m for $f(x) = x^2$ and $f(x) = x^3$.