MAT 362 Spring 2007 2. Programming assignment: Nonlinear equations Name:

Instructor: Nándor Sieben
e-mail: nandor.sieben@nau.edu
http://jan.ucc.nau.edu/∼ns46/362/362.html
Office: AMB 175
Hours: MW 10:10-12:10, F 10:10-11:10

---

Write programs that implement the following algorithms:
a. Bisection method;
b. Fixed point iteration;
c. Newton's method;
d. Secant method.
You may use the bisection code on the course web site. Define separate functions for $f$ (and $f'$ for Newton's method), to make changing the function easier. At every iteration output the number of iterations, the current estimate for the root and the value of the function at this estimate. Use consistent indentation and write plenty of comments in your code. Use meaningful variable names.
Use your programs to find the root of

$$f(x) = e^{\cos(2x - \ln(x))} - 1$$

in the interval $[2, 3]$ and another root in the interval $[4, 5]$ with absolute tolerance $tol = 10^{-6}$. Compare the effectiveness of these methods. Use gnuplot to create a graph (postscript file `plot.ps`) of $f$ on the interval $[1, 6]$.

**Turn in:**

- This problem sheet.
- A summary sheet explaining what you did, how you approached the problem, what was accomplished, what was not accomplished, etc.
- A table showing how many iterations each method required for each root.
- Your analysis of the effectiveness of the methods. Include your choices for the initial approximations.

**Website:**

- Create a directory called `2nonlinear` on your web site and make all your input, output and source files available in this directory. Write the url for the website on this problem sheet.

**Hints:**

- Consult the links on the course website if you don't know how something works. Pretty much everything you might need is available on the web somewhere.
- Create a comment line in your codes that has your name and the name of the file. This makes it easy to work with printouts.
- Never hardwire constants into your code unless it's absolutely unavoidable. Use const variables instead.
- Use `indent -br foo.cc` to format your code.
- Use the `C++` functions `exp` and `log`. You need to `#include` the `math.h` header file for these functions to work.
- Use `cout << setprecision(precision) << variable` to adjust the number of digits printed. You will need to include the iomanip header file for this to work. You need to `#include` the `iomanip` header file for this to work.
- Read the gnuplot documentation on the web to figure out how to adjust the domain of $f$ on the plot.