AMSC/CMCS 466 – Introduction to Numerical Analysis I

Spring Term 2006

Instructor: Georg Dolzmann

Homework set #1

Problem 1: a) Write a MATLAB program called `bisection(f,a,b,tol)` that implements the bisection method. The input parameters are the function f, which should be written in a separate mfile `f.m`, the pair of points a and b which bracket the desired zero of f, and an error tolerance `tol`. The program should stop when the error `abs(b-a)` is less than or equal to `tol`. The output parameters are an estimate `(a+b)/2` of the zero, an error estimate, and the number of iterations.

b) Use this program to compute the smallest positive root of

$$e^{-x} = \sin(x),$$

with `tol`=10^{-15}. Plot both functions with the command `fplot` to figure out a starting bracket $[a, b]$.

c) Compare the actual number of iterations with the theoretical bound.

Problem 2: [Atkinson 2.4] Implement the algorithm *Newton* given in Section 2.2. Use it to solve the following equations:

a) $e^x - 3x^2 = 0$;
b) $x^3 = x^2 + x + 1$;
c) $e^x = 1/(0.1 + x^2)$;
d) $x = 1 + .3\cos(x)$.

You can define a subroutine

```
function [root, ier] = newton1d(f,df,x0,eps,itmax)
```

where `f` and `df` are functions implementing f and f'.

Problem 3: [Atkinson 2.12] Consider Newton's method for finding the positive square root of $a > 0$. Derive the following results, assuming $x_0 > 0$, $x_0 \neq \sqrt{a}$.

a) $x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right)$;
b) Show that for $n \geq 0$,

$$x_{n+1}^2 - a = \left(\frac{x_n^2 - a}{2x_n}\right)^2,$$

and thus $x_n \geq \sqrt{a}$ for all $n > 0$;

c) The iterates $\{x_n\}$ are a strictly decreasing sequence for $n \geq 1$.

Hint: Consider the sign of $x_{n+1} - x_n$.

d) $e_{n+1} = -e_n^2/(2x_n)$, with $e_n = \sqrt{a} - x_n$, and

$$\mathrm{Rel}(x_{n+1}) = -\frac{\sqrt{a}}{2x_n}\,\mathrm{Rel}^2(x_n), \quad n \geq 0.$$

Problem 4: [Atkinson 2.13] Newton's method is the commonly used method for calculating square roots on a computer. To use Newton's method to calculate $\sqrt{a}$, an initial guess x_0 must be chosen, and it would be most convenient to use a fixed number of iterates rather than having to test for convergence. For definiteness, suppose that the computer arithmetic is binary and that the mantissa contains 48 binary bits. Write

$$a = \widehat{a} \cdot 2^e, \quad \frac{1}{2} \leq \widehat{a} < 1.$$

This can be easily modified to the form

$$a = b \cdot 2^f, \quad \frac{1}{4} \leq b < 1$$

with f an even integer. Then

$$\sqrt{a} = \sqrt{b} \cdot 2^{f/2}, \quad \frac{1}{2} \leq \sqrt{b} < 1,$$

and the number $\sqrt{a}$ will be in standard floating point form, once $\sqrt{b}$ is known. This reduces the problem to that of calculating $\sqrt{b}$ for $\frac{1}{4} \leq b < 1$. Use the linear interpolating formula

$$x_0 = \ell(b) = \frac{1}{3}(2b+1), \quad \frac{1}{4} \leq b \leq 1$$

as an initial guess for the Newton iteration for calculating $\sqrt{b}$. Bound the error $\sqrt{b} - x_0$. Estimate how many iterates are necessary in order that

$$0 \leq x_n - \sqrt{b} \leq 2^{-48}$$

which is the limit of machine precision for b on a particular computer. (Note that the effect of rounding errors is being ignored.) How might the choice of x_0 be improved?