Math 128a - Homework 3 - Due Sept 21 at the beginning of class

1) Sometimes is difficult or expensive to compute $f'(x_n)$ for use in Newton's method, so we use an approximation α instead. Find the condition on α to ensure that the iteration $x_{n+1} = x_n - f(x_n)/\alpha$ will converge (at least) linearly to a zero of f if started near enough to the zero.

Answer: Let z be a zero of f and let $e_n = x_n - z$. The iteration is $x_{n+1} = x_n - \frac{f(x_n)}{\alpha}$, so $e_{n+1} = e_n - \frac{f(x_n)}{\alpha}$. Now compute using the Taylor series about z:

$$\begin{aligned} e_{n+1} &= e_n - \frac{f(x_n)}{\alpha} \\ &= e_n - \frac{1}{\alpha}[f(z) + f'(z)e_n + O(e_n^2)] \\ &= e_n - \frac{e_n}{\alpha}[f'(z) + O(e_n)] \\ &= [1 - \frac{f'(z)}{\alpha} + O(e_n)]e_n \end{aligned}$$

So in order to get linear convergence we need

$$|1 - \frac{f'(z)}{\alpha}| = \frac{|\alpha - f'(z)|}{\alpha} < 1$$

which is the same as α having the same sign as $f'(z)$ and $\frac{|f'(z)|}{2} < |\alpha|$.

2) Show that when $f(z) = f'(z) = 0 \neq f''(z)$, i.e. z is a double zero, then Newton's method only converges linearly to z (with constant .5) when started close enough to z, not quadratically. Show that if z is a double zero, then the modified Newton iteration $x_{n+1} = x_n - 2 \cdot f(x_n)/f'(x_n)$ is quadratically convergent. (Note: A similar argument shows that if x is a root of multiplicity k, then $x_{n+1} = x_n - k \cdot f(x_n)/f'(x_n)$ converges quadratically, but you do not have to show this.)

Answer: If $f(z) = f'(z) = 0 \neq f''(z)$ then we compare $e_n = x_n - z$ and $e_{n+1} = x_{n+1} - z$ as follows:

$$\begin{aligned} e_{n+1} &= e_n - \frac{f(x_n)}{f'(x_n)} \\ f(x_n) &\approx \frac{f''(z)e_n^2}{2} \\ f'(x_n) &\approx f''(z)e_n \\ |e_{n+1}| &\approx |e_n - \frac{1}{2}e_n| \\ &= \frac{1}{2}|e_n| \end{aligned}$$

which implies linear convergence of x_n to z with $c = 1/2$. (A similar argument shows that, for a root of mulitplicity $k > 1$, convergence is linear with constant $c = 1 - 1/k$, which gets slower (c gets closer to 1) for higher multiplicity k.

So if $f(z) = f'(z) = 0 \neq f''(z)$, we try the modified iteration $x_{n+1} = x_n - 2\frac{f(x_n)}{f'(x_n)}$. This

converges quadratically because:

$$\begin{aligned} e_{n+1} &= e_n - 2\frac{f(x_n)}{f'(x_n)} \\ f(x_n) &= \frac{f''(z)e_n^2}{2!} + \frac{f'''(\xi)e_n^3}{3!} \\ &\approx \frac{f''(z)e_n^2}{2!} + \frac{f'''(z)e_n^3}{3!} \\ f'(x_n) &\approx f''(z)e_n \quad \text{so :} \\ |e_{n+1}| &\approx |e_n - 2(\frac{e_n}{2} + \frac{f'''(z)e_n^2}{6f''(z)})| \\ &= \frac{f'''(z)}{3f''(z)} \cdot |e_n|^2 \end{aligned}$$

3) Suppose we want to implement the squareroot function using Newton's method $x_{n+1} = (x_n + a/x_n)/2$ to solve $f(x) = x^2 - a = 0$. The input a can be any positive normalized IEEE double precision number (zero and subnormal numbers are handled as special cases). We want to quickly find a starting value x_1 that guarantees that 3 Newton steps are enough to get nearly full precision (relative error near 2^{-52}) no matter what a is. We will compute x_1 with a little bit-fiddling and a table as follows:

Part 1. Using the floating point representation $a = 2^e(1 + f)$, where $0 \le f < 1$, rewrite a as $a = 2^{2e'}(1 + f')$, where $e' = \lfloor e/2 \rfloor$ (i.e. $e/2$ rounded down to the nearest integer) and f' is modified appropriately. What range can $1 + f'$ lie in?

Answer: Write $a = 2^e(1 + f) = 2^{2e'}(1 + f')$, where $e' = \lfloor e/2 \rfloor$. If e is even, then $e' = e/2$, $2e' = e$ and so $1 \le 1 + f' = 1 + f < 2$. If e is odd, then $e' = (e - 1)/2$, $2e' = e - 1$ and so $2 \le 1 + f' = 2(1 + f) < 4$. Altogether $1 \le 1 + f' < 4$.

Part 2. Since $a^{1/2} = 2^{e'}(1 + f')^{1/2}$, it suffices to apply Newton to $1 + f'$. We will pick a starting value x_1 by looking it up in a table. Write $1 + f' = bb.bbbbb..._2$, i.e. as a binary number with a fractional part, where each b is a bit. Suppose we take the leading k bits of this number (which approximate $1 + f'$, the better for larger k), and look up its square root $x(1)$ in a table (k bits means we need a table of 2^k values, which we index into just by interpreting the leading k bits as a binary integer). What is the smallest possible k that guarantees $x(1)$ is accurate enough to guarantee that we only need 3 Newton iterations? 2 Newton iterations? Do not worry about roundoff in your analysis. (In actual IEEE arithmetic, which requires the square root to be correctly rounded, extra care in the algorithm is needed to ensure this, but this algorithm is nearly accurate enough, and has been frequently used in the past.)

Answer: Let $y = 1 + f'$. The problem is: Given y with $1 \le y < 4$, let $\widehat{y}$ be the closest k-bit binary number to y, so that $|y - \widehat{y}| \le 2^{-k+1}$ (because we have two bits in the integer part) with $1 \le \widehat{y} \le 4$. Now we wa nt to find x such that $y = x^2$ and we are given (by the table) $\widehat{x}$ such that $\widehat{y} = \widehat{x}^2$. We start Newton's method for $f(x) = x^2 - y$ with $x_0 = \widehat{x}$, and we want to know how many iterations it will take, in terms of k, to get to maximum IEEE precision, which, for $1 \le x \le 2$ is an error of less than or equal to 2^{-53}. We begin by bounding the

initial error $e_0 = x_0 - x$ from the table, given that x_0 is the exact square root of $\widehat{y} \approx y$:

$$\begin{aligned} 2^{-k+1} &\geq |\widehat{y} - y| \\ &= |\widehat{x}^2 - x^2| \\ &= |x_0^2 - x^2| \\ &= |x_0 - x| \cdot |x_0 + x| \\ &\geq 2|x_0 - x| \quad \text{because } 1 \leq x, x_0 \quad \text{, so} \\ |e_0| &= |x - x_0| \\ &\leq 2^{-k} \end{aligned}$$

Next, we want $|e_n| \leq 2^{-53}$. We know that $|e_{n+1}| \leq C|e_n|^2$ where $C \approx \frac{f''(x)}{2f'(x)}$. But $f'(x) = 2x$ and $f''(x) = 2$ so $C \approx \frac{1}{2x} \leq \frac{1}{2}$ (since $1 \leq x$). Thus $|e_{n+1}| \leq \frac{1}{2}|e_n|^2$, so $|e_n| \leq \frac{1}{2^n}|e_0|^{2^n} \leq 2^{-n-k2^n} \leq 2^{-53}$. The inequality that says $|e_n| \leq 2^{-53}$ is thus $n + k2^n \geq 53$; actually we want to find k that make $n = 2$ Newton iterations and $n = 3$ Newton iterations work. Thus we need $k \geq \frac{53-n}{2^n}$. For $n = 3$ we have $k = 7$ (a table of $2^k = 2^7 = 128$ words) and for $n = 2$ we have $k = 13$ (a table of $2^k = 2^{13} = 8096$ words). This shows that we may trade off time and space, with a (much) bigger table (larger 2^k) yielding a faster algorithm (smaller n). This tradeoff of time and space is common in designing numerical algorithms.

4) Prove that, if $F : [a, b] \to \mathbf{R}$, if F' is continuous and if $|F'(x)| < 1$ on $[a, b]$ then F is a contraction on $[a, b]$. Does F necessarily have a fixed point?

Answer: *Proof:* Since $|F'(x)| < 1$ on $[a, b]$ and $|F'|$ is a continuous function on the closed interval $[a, b]$, $|F'|$ has a maximum value on $[a, b]$, say λ, and $\lambda < 1$. Now pick two arbitrary points $x, y \in [a, b]$. By the Mean-Value Theorem there exists a point c between x and y such that

$$\begin{aligned} \frac{F(x) - F(y)}{x - y} &= F'(c) \quad \text{so :} \\ \frac{|F(x) - F(y)|}{|x - y|} &= |F'(c)| \\ &\leq \lambda \quad (\text{since } c \in [a, b]), \text{ so :} \\ |F(x) - F(y)| &\leq \lambda |x - y| \end{aligned}$$

Therefore F is a contraction. □

F does not necessarily have a fixed point: for example, $F(x) = x/2 - 1$ has no fixed points as a map $F : [1, 2] \to \mathbf{R}$.

5) Prove that if F is a continuous map of $[a, b]$ into $[a, b]$ then F must have a fixed point. Then determine if this assertion is true for functions from $\mathbf{R}$ to $\mathbf{R}$.

Answer: *Proof:* We want to find some $x \in [a, b]$ such that $F(x) = x$; alternatively, letting $G(x) = F(x) - x$, we want to find a value of x such that $G(x) = 0$. Note that $G(a) = F(a) - a \geq 0$ because $a \leq F(a) \leq b$ and that $G(b) = F(b) - b \leq 0$ because $a \leq F(b) \leq b$. Thus, since G is continuous, there must exist x between a and b such that $G(x) = 0$. □

This is not true for functions from $\mathbf{R}$ to $\mathbf{R}$: for example, $F(x) = x - 1$ has no fixed points.