CME 108/MATH 114
Introduction to Scientific Computing
Summer 2018

Problem Set 1

Due: Friday, July 6 at 5:00pm

1. (10 points) *(Order of operations, Bradie* 1.1.6*).* Consider the computation of the following sum,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i b_j,$$

where every a_i and every b_j is a real number.

(a) How many multiplications and how many additions are required to compute the sum? Your answer should depend on n.

(b) Modify the summation to an equivalent form which reduces the number of operations needed for evaluation. How many multiplications and how many additions are required to compute the sum in this revised form?

2. (10 points) *(Superlinearity, Bradie* 1.2.14*).* A sequence p_n which converges to p is said to be of *order* α with *asymptotic error constant* λ if

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^{\alpha}} = \lambda,$$

for **positive** constants α and λ.

A sequence p_n converges to p *superlinearly* if

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|} = 0.$$

Show that if $p_n \to p$ of order α for any $\alpha > 1$, then p_n converges superlinearly to p.

3. (5 points) *(Floating point equivalents.)* Consider the FPS $\mathbf{F}(10, 5, 0, 32)$. Provide the floating point equivalent for each of the following numbers. Consider both chopping and rounding.

(a) e

(b) 1/7

4. (15 points) *(Roundoff error).* In class we showed that the relative rounding error for the chopping scheme is bounded by β^{1-p}, where β is the base of the FPS and p is its precision.

Show that the relative roundoff error associated with the rounding scheme is at most $\frac{1}{2}\beta^{1-p}$. That is, show that

$$\epsilon_{\mathrm{rel}} = \frac{|\mathbf{fl}_{\mathrm{round}}(x) - x|}{|x|} \leq \frac{1}{2}\beta^{1-p},$$

for any $x \neq 0 \in \mathbb{R}$.

5. (10 points) *(Square root via Newton's method).* Given a real number a, construct a function f_a such that $\sqrt{a}$ is a zero of f_a.

Using Newton's method, write a recursive relation to compute successively better approximations of $\sqrt{a}$.

Hint: We saw this recursion in the first day of class!

6. (10 points) *(Secant method, Bradie* 2.5.1*).* The so-called "Secant Method" is a rootfinding scheme related to Newton's method. Suppose you want to find a root of a given function f. In many applications, f' cannot be calculated directly. To circumvent the issue, we replace the derivative term in Newton's method by the approximation

$$f'(p_n) \approx \frac{f(p_n) - f(p_{n-1})}{p_n - p_{n-1}}.$$

That is, the slope of the tangent line at $x = p_n$ is replaced by the slope of the *secant* line formed between $x = p_n$ and $x = p_{n-1}$.

Note that calculating p_{n+1} requires knowledge of both p_n and p_{n-1}, so the Secant Method requires two starting values, p_0 and p_1.

The function $f(x) = \ln(1+x) - \cos(x)$ has a root in the interval $(0, 1)$. Perform the secant method to determine p_4, the fourth approximation to the location of the root, using $p_0 = 0$ and $p_1 = 1$.

7. (30 points) *(Rootfinding with MATLAB).*

(a) Implement the bisection method in MATLAB, and use it to find the root of

$$f(x) = \ln(1 + x) - \cos(x)$$

in $(0, 1)$. Start with $a_1 = 0$ and $b_1 = 1$. Use the stopping criterion

$$\frac{|b_n - a_n|}{2} \leq \epsilon,$$

with $\epsilon = 10^{-6}$.

Report the values of p_i for $i = 1, 2, \ldots, n$.

(b) Implement Newton's method in MATLAB and use it to find the root of f in $(0, 1)$, with f as above. Start with $p_0 = 1/2$ and stop at iteration n if

$$|p_n - p_{n-1}| \leq \epsilon,$$

with $\epsilon = 10^{-6}$.

Report the values of p_i for $i = 1, 2, \ldots, n$.

(c) In a sentence, compare the performance of the two rootfinding schemes used.

(d) (10 points) (BONUS!) Implement the secant method in MATLAB, and use it to approximate the root of f in $(0, 1)$, with f as above. Start with $p_0 = 0$ and $p_1 = 1$ and use the same stopping criterion as in (b). In a sentence, compare the performance of the three rootfinding methods.