University of Utah
School of Computing

cs 6210 **Problem Set One** **Due: Sept 22, 2005**

---

1. (15 pts) Design an algorithm to compute

$$w(a,c,d) = \frac{ad}{c^2 - d^2} - \frac{a}{c+d} \quad a, c, d \text{ machine numbers}$$

such that the relative error of the computed result is less than $K2^{-t}$. (Find K as small as possible.)

2. (25 pts) Show that Horner's algorithm for computing Newton's form of a polynomial is NWB (numerically well behaved).

3. (30 pts)

(a) Write a program to compute and approximate value of the derivative of a function f using the formula

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}.$$

Test your code for $f(x) = s9n(x)$ at $x = 1$, and plot the error as a function of $n = 10^{-k}$, $k = 0, 1, \ldots, 16$ (using a log scale for h and the error).

(b) Repeat part (a) for the formula

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

4. (30 pts) Compute e^{-x} for $x = 5, 10, 20, 30, 40, \ldots$ by using:

(a) $e^{-x} = \sum_{i=0}^{\infty} (-x)^i / i!$.

(b) $e^{-x} = 1 / \left(\sum_{n=0}^{\infty} x^i / i! \right)$.

(c) standard $\exp(-x)$.

Compare and explain the results. (Remark: you may terminate the summation $\sum_{i=0}^{\infty} b_i$ whenever $fl(S_n + b_{n+1}) = S_n$, where $S_n = fl(\sum_{i=0}^{\infty} b_i)$, for $b_i \to 0$.

Extra Credit (challenge) problem.

Derive and implement a NWB algorithm for computing roots of cubic equations $x^3 + ax^2 + bx + c = 0$. (Compare problem 1.11 of Heath's book.)