CS371/AMATH242 Winter 2019: Assignment 1

Instructor: Maryam Ghasemi

Due Friday, February 1, 11:59 pm via Crowdmark

---

1. [5 marks] Consider a fictitious floating point number system $F[3, 3, 1]$ corresponding to base 3, three digits for the normalized mantissa, and one digit for the exponent, that uses rounding when storing real numbers (rounding 5's up if needed). Nonzero numbers have the form:$\pm 0.d_1d_2d_3 \times 3^e$, $-1 \leq e \leq 1$. For each of the following questions, provide a brief explanation.

   (a) What is the largest positive normalized value that can be stored in F?

   (b) What is the smallest positive normalized value that can be stored in F?

   (c) What is the value of $\epsilon_{machine}$, the smallest value x such that $fl(1 + x) > fl(1)$ in F?

   (d) How many different nonzero numbers (positive and negative) can be stored in F?

   (e) Give *concrete* examples of real values (using base 10) that will cause overflow and underflow when converted to the floating point number system F.

2. [3 marks] Let $x = 2$. Consider evaluating

$$z = \sqrt{x^2 + 1}$$

   using the floating point number system F as defined in Question 1. (Note that x is a floating point number in F.)

   (a) Use a calculator to compute z and then show that the real number z is smaller than the largest floating point number in F. Thus, both x and z are within range of representable floating point numbers.

   (b) If one uses the floating point number system F to compute z using the formula above, what would happen to the floating point representation for x^2? Derive another formula for z such that such an error would not occur. Briefly explain how your formula works.

   (c) Use the floating point number system F to compute z using the formula derived in (b). Show all your calculations. Note that you should perform rounding after each operation.

3. [3 marks] Suppose two points (x_0, y_0) and (x_1, y_1) are on a straight line, where y_0 and y_1 are different. Two formulas are available to find the x-intercept of the line:

$$x = \frac{x_0y_1 - x_1y_0}{y_1 - y_0} \qquad and \qquad x = x_0 - \frac{(x_1 - x_0)y_0}{y_1 - y_0}$$

   (a) Show that both formulas are algebraically equivalent.

   (b) Using the points $(x_0, y_0) = (1.31, 3.24)$ and $(x_1, y_1) = (1.93, 4.76)$ and three-digit rounding arithmetic, compute the x-intercept both ways. Show your workings, and show where there is a loss of information due to rounding in your calculations.

   (c) Which method is better and why? Will it better for all values of (x_0, y_0) and (x_1, y_1)? Explain your answer.

4. [11 marks] Consider the following problem:

   The sum of two numbers is 20. Add each number to its square root, and multiply together. The product of the two sums is approximately 155.55.

   (a) Define a function f(x) such that finding a root of f(x) solves the above problem. You may assume x is a number between 1 and 9, inclusive.

   (b) Write Matlab code to find a root of your function f(x)=0 using the Newton's method. Continue the iterations until $|f(x)| \leq 10^{-6}$. Include a table of the iterate values, when starting from $x_0 = 1$ and $x_0 = 5$.

   (c) Without implementating the bisection algorithm:

      (i) Prove that [1,9] is an appropriate initial bracket for a root of f.

      (ii) Determine the approximate number of iterations required by the bisection algorithm to find the interval $[a_k, b_k]$ of length$\leq 10^{-6}$.

5. [8 marks] Moler's Numerical Computing with Matlab website

   (http://www.mathworks.com/moler/ncmfilelist.html) includes many useful Matlab functions, including `powersin.m` (which can be downloaded):

```
function s = powersin(x)
%POWERSIN Power series for sin(x).
% y = POWERSIN(x) tries to compute sin(x) from its power series.
%
% Copyright 2014 Cleve Moler
% Copyright 2014 The MathWorks, Inc.

s = 0;
t = x;
n = 1;
while s+t ~= s;
          s = s + t;
          t = -x.^2/((n+1)*(n+2)).*t;
          n = n + 2;
end
```

   This function computes an approximation to using its power series expansion (which is infinite): $\sin(x) = \sum_{n=1}^{\infty}(-1)^{n+1}\frac{x^{2n-1}}{(2n-1)!}$

   (a) The while loop terminates when `s+t` does not differ from `s`. Is this the same as continuing while `t` is not 0?

   (b) Modify `powersin` so that it returns three values corresponding to the calculation of the sum `s`; `function [s, maxterm, numterms] = powersin(x)` where `maxterm` is the largest term (value of t) calculated, and `numterms` is the number of terms of the power series used to calculate sum `s` (note that the first term, `x`, is calculated before the loop begins). Include a listing of the modified `powersin` function with your submitted solution.

   (c) Call the new version of `powersin` for $x = \pi/2, 11\pi/2, 21\pi/2$, and $31\pi/2$. For each value of `x`, show the final values of `s`, `maxterm`, and `numterms`, and calculate the absolute value of the relative error in the computed sum for each `x`.

(d) Based on your results, what can you conclude about the use of the `powersin` algorithm for various values of `x`? Does the distance of `x` from 0 affect the results?