MATH 4640 Numerical Analysis - HW 1 Solutions

Austin Barton

January 15, 2025

Problem 1:

Solution.   i.   a) $2^4 * 1 + 2^3 * 1 + 2^2 * 1 + 2^1 * 0 + 2^0 * 1 + 2^{-1} * 1 + 2^{-2} * 0 + 2^{-3} * 1 + 2^{-4} * 1 + 2^{-5} * 1 = 16 + 8 + 4 + 1 + 0.5 + 0.125 + 0.0625 + 0.03125 = 29.71875$

        b) $16^2 * 2 + 16^1 * 11 + 16^0 * 3 + 16^{-1} * 15 + 16^{-2} * 15 = 512 + 176 + 3 + 0.9375 + 0.05859375 = 691.99609375$

        c) $\sum_{i=1}^{n} 2^{i-1} * 1$

        d) $\sum_{i=1}^{n} 2^{-i} * 1$ ■

   ii.   a) This is $2^5 + 2^4 + 2^2 + 2^1$ so this is 11110 in binary.

        b) This is $2^7 + 2^6 + 2^5 + 2^4 + 2^3$

        c)

Problem 2:

Solution.   a) Absolute error is

$$|10451.0023 - 10451.001| = 0.0013$$

     Relative error is

$$\frac{0.0013}{10451.0023} = .124389983 \times 10^{-6}$$

   b) Absolute error is

$$|0.451011 \times 10^4 - 0.451010 \times 10^4| = 0.01$$

     Relative error is

$$\frac{0.01}{0.451011 \times 10^4} = 0.22171 \times 10^{-5}$$

   c)

Problem 3:

Solution. NOTE TO READER: I am adjusting some basic notation to match the Python code I have written for these algorithms. It's not necessary but it makes it easier to reference code from the mathematical algorithm.

a) Algorithm

Let B be the base 2 number represented in β notation. That is, $B = (a_0 a_1 \ldots a_{l-1}.b_1 b_2 \ldots b_r)_\beta$. Note that our indices are different than the textbook, but it's the same notation nonetheless.

For bits to the left of the radix point do the following steps: (Note that following our notation, l is the number of bits left of the radix point)

Let $i \in \{0, \ldots, l-1\}$. Calculate the sum $L = \sum_{i=0}^{l-1} 2^{l-(i+1)}$.

For bits to the right of the radix point do the following steps: (Note that following our notation, r is the number of bits right of the radix point)

Let $j \in \{1, \ldots, r\}$. Calculate the sum $R = \sum_{j=1}^{r} 2^{-(i+1)}$.

The sum, $L + R$ and that is the decimal equivalent. That is, $B = L + R$ where B is the base 2 encoded number and $L + R$ is the decimal equivalent.

Note that all operations and numbers in the algorithm is assumed to be represented in base 10, but the algorithm remains the same nonetheless, so long as we understand our representation of the number of two (10 in base 2 and 2 in base 10).

Thus, our algorithm takes a binary encoded number (base 2) and converts to the decimal equivalent, $L + R$.

b) Algorithm

    Problem 4:

Solution.

    Problem 5:

Solution.

    Problem 6:

Solution.

    Problem 7:

Solution.

    Problem 8:

Solution.