Homework 1: CS321, Fall 2014

Answer Sheet

1. Check the binary-octal table, we have

\[(45653.127664)_8 = (100 101 110 101 011.001 010 111 110 110 100)_2\]

For conversion to decimal numbers, we need to treat the integer and fractional parts separately. For the integer part:

\[(45653)_8 = 3 \times 8^0 + 5 \times 8^1 + 6 \times 8^2 + 5 \times 8^3 + 4 \times 8^4\]
\[= 3 + 8(5 + 8(6 + 8(5 + 4(8)))) = (19371)_{10}\]

For the fractional part, we compute

\[(.127664)_8 = 1^{-1} + 2 \times 8^{-2} + 7 \times 8^{-3} + 6 \times 8^{-4} + 6 \times 8^{-5} + 4 \times 8^{-6}\]
\[= (1 \times 8^0 + 2^4 + 7 \times 8^3 + 6 \times 8^2 + 6 \times 8^1 + 4) 8^{-6}\]
\[= ((((((1)8 + 2)8 + 7)8 + 6)8 + 6)8 + 7) 8^{-6}\]
\[= \frac{44980}{262144} = (.17158508...)_{10}\]

Checking the binary-hexadecimal table, we have

\[(C553E000)_{16} = (1100 0101 0101 0011 1110 0000 0000 0000)_{2}\]

Note that this problem is NOT to convert this hexadecimal number to a decimal number. This hexadecimal number is an IEEE 32 bit representation of a binary number.

The first bit is “1”, so this number is negative. The next 8 bits, (10001010)₂, represent the exponent, which is (using the binary-octal table)

\[(010 001 010)_2 = (212)_8 = 2 \times 8^0 + 1 \times 8^1 + 2 \times 8^2 = (138)_{10}\]

Because of the 127 shift convention, the actual exponent is 138 − 127 = 11. It follows that the decimal number is

\[-(1.101 001 111 100)_2 \times 2^{11} = -(110 100 111 110)_2 = -(6476)_8 = -(3390)_{10}\]

2. When \(x\) and \(y\) are machine numbers, they can be stored exactly. However, their operations may result in a number that cannot be stored exactly. We can see that

\[\text{fl}(xy) = xy(1 + \delta_1), \quad \text{with} \quad |\delta_1| \leq 2^{-24}\]
Because $xy$ is computed first, it follows that
\[
\text{fl}((xy)z) = \text{fl}(\text{fl}(xy)z) = \text{fl}(xy(1 + \delta_1)z) = (xy(1 + \delta_1)z)(1 + \delta_2) \\
= xyz(1 + \delta_1)(1 + \delta_2) = xyz(1 + \delta_1 + \delta_2 + \delta_1\delta_2) \\
\approx xyz(1 + \delta)
\]
Here $|\delta_1| \leq 2^{-24}$, $|\delta_2| \leq 2^{-24}$, and $|\delta| = |\delta_1 + \delta_2| \leq |\delta_1| + |\delta_2| \leq 2^{-23}$. We ignored them higher order term $\delta_1\delta_2 \leq 2^{-48}$.

3. This problem just needs you to give an example. First assume your computer works on two (or more) decimal digit arithmetic. Let $a = 0.51$, $b = 0.52$, $c = 0.54$, we have
\[
a + (b + c) = 0.51 + (0.52 + 0.54) = 0.51 + 1.1 = 1.6
\]
Note that $0.52 + 0.54 = 1.06 \approx 1.1$, $0.51 + 1.1 = 1.61 \approx 1.6$, since the computer can only store two decimal digits and will do correct rounding after each step of the computations.
\[
(a + b) + c = (0.51 + 0.52) + 0.54 = 1.0 + 0.54 = 1.5
\]

4. The most important point here is to realize that if a number is divided by 2, it is equivalent to moving the binary point to the right by one position. Thus, the best way to compute the machine $\epsilon$ is to set up a loop to check if the identity $1 + \epsilon = 1$ holds, with initially setting $\epsilon = 0.1$. If the identity holds, you exit the loop, otherwise you divide $\epsilon$ by 2, i.e., by setting $\epsilon = \epsilon/2$. Note that in binary computation $0.1/2 = 0.01$, moving the binary point to the right by one position, When the loop exits, you get your machine $\epsilon$.

Note that, in order to prevent the code from running indefinitely due to coding error, you may want to set a maximum number of iterations for the loop, and check the value of $\epsilon$ to see if it is small upon exit.

5. The computation with double precision will have $f(0.1) = 0.10333 \times 10^{-1}$.

Straightforward evaluation with five decimal digits is
\[
f(0.1) = e^{0.1} - \cos(0.1) - \sin(0.1) = 0.11052 \times 10^1 - 0.99500 - 0.99833 \times 10^{-1} = 0.10367 \times 10^{-1}
\]
The difference is $0.34 \times 10^{-4}$. Note that at each step, only five decimal digits are kept.

However, using Taylor series, we have
\[
e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \cdots
\]
\[
\cos x = 1 - \frac{x^2}{2} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots
\]
\[
\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots
\]

so

\[
f(x) = x^2 + \frac{x^3}{3} + \frac{x^6}{2600} + \cdots
\]

Taking the first two terms, we have \( f(0.1) = 0.01 + 0.33333 \times 10^{-3} = 0.10333 \times 10^{-1} \).

Therefore, Taylor series computation yields more accurate result.

6. When \( x \) is large, \( \sqrt{x + 2} \approx \sqrt{x} \), so the direct evaluation of \( f(x) \) will lose significant digits. We can use rationalization to avoid the subtraction of two almost equal numbers.

\[
f(x) = \frac{(\sqrt{x + 2} - \sqrt{x})(\sqrt{x + 2} + \sqrt{x})}{\sqrt{x + 2} + \sqrt{x}} = \frac{2}{\sqrt{x + 2} + \sqrt{x}} \approx \frac{1}{\sqrt{x}}
\]

This rationalization process transforms the subtraction two almost equal numbers into the summation of two almost equal numbers. (The last step of approximation is not necessary).