Assignment 1, Solution

Problem 1 (2 pts.) The IEEE-754 floating-point standard specifies that single-precision floating-point numbers be represented in a base-2 format with a 24-bit significand:

$$\hat{x} = \pm a_0.a_1a_2\cdots a_{23} \times 2^e \tag{1}$$

We say that the nonzero quantity $\hat{x}$ is normalized if $a_0 = 1$ and $-126 \leq e \leq 127$. We define the representation of zero as $0.00\cdots0 \times 2^0$

a) What is the normalized IEEE single-precision representation of the number 8.125? Express your answer in the form of Eq. (1). In other words, specify the values of $a_1, a_2, \cdots a_{23}$ and e.
Answer:

$$8.125 = 2^3 + 2^{-3} = 1000.001_2 = 1.000001\underbrace{00\cdots0}_{17\text{ zeros}} \times 2^3$$

b) If we change the significand of 8.125 by 1 ulp, by how much does the value of the floating-point representation change? Express your answer as a power of 2.
Answer: If we change the last bit by 1 ulp, then the value of the significand changes by 2^{-23}, and the value of the the floating-point number itself changes by $8 = 2^3$ times this amount, or 2^{-20}

Problem 2 (3 pts.)

Let $x = 1/7$

a) Find the (infinite, eventually repeating) binary representation of x.
Answer:

$$x = \frac{1}{8} + \frac{1}{64} + \frac{1}{512} + \cdots = 0.001\overline{001}_2$$

because

$$\frac{1}{8} + \frac{1}{64} + \frac{1}{512} + \cdots = \frac{\frac{1}{8}}{1 - \frac{1}{8}} = \frac{1}{7} = x$$

b) Find the IEEE-754 single-precision representation $\hat{x}$ of x when rounding to nearest. That is, find the closest approximation to x by an expression of the form (1).
Answer:. We can rewrite the answer in (a) as $1.001\overline{001} \times 2^{-3}$, which must be rounded to the nearest binary number after the 23rd place to the right of the binary point.

$$\frac{1}{7} = 1.\underbrace{001\cdots001}_{7\text{triples}}\underbrace{00}_{a_{22}a_{23}} \mid 1\overline{001} \times 2^{-3}$$

Since the nearest binary number after the 23rd place is 1, the value that is represented in the computer is rounded up:

$$\hat{x} = 1.\underbrace{001\cdots001}_{7\text{triples}}\underbrace{01}_{a_{22}a_{23}} \times 2^{-3}$$

c) Use geometric series to find an exact expression for the error $\hat{x} - x$ due to the rounding in **b)**.
Answer:.

$$\hat{x} = \left(1 + \frac{1}{8} + \frac{1}{8^2} + \cdots + \frac{1}{8^7} + 2^{-23}\right) \times 2^{-3}$$

$$\hat{x} = \frac{1}{7}\left(1 - \frac{1}{8^8}\right) + 2^{-26}$$

So

$$\hat{x} - x = \frac{1}{7}\left(1 - \frac{1}{8^8}\right) + 2^{-26} - \frac{1}{7} = 2^{-26} - \frac{1}{7}2^{-24} = \left(1 - \frac{4}{7}\right)2^{-26} = \frac{3}{7}2^{-26}$$

Problem 3 (1 pts.)
What fraction is represented by the repeating decimal sequence $0.135\overline{135}$? Express your answer as p/q in lowest terms.
Answer:

From the geometric series formula,

$$0.135\overline{135} = 135 \times \left(10^{-3} + 10^{-6} + \cdots\right) = \frac{135}{1000} \times \frac{1}{1 - 10^{-3}} = \frac{135}{1000} \times \frac{1000}{999} = \frac{135}{999} = \frac{5}{37}$$

Problem 4 (2 pts.)
True or false: If x has a terminating base-2 expansion, then x has a terminating base-10 expansion. State a proof using a result given in lecture or find a counterexample.
Answer:

True. The theorem quoted in class is that the ratio p/q (in lowest terms) has a terminating base-b expansion if and only if q divides some power of b. In this case, $q = 2^k$ for some k so q divides 10^k .

Problem 5 (3 pts.)

The heart of many numerical methods, such as Newton's method, involves the solution of a linear system of equations. This problem illustrates some of the inherent mathematical problems that can exist in the solution of linear systems.

a) Let n be a positive integer and consider the linear system

$$\begin{aligned} x + y &= a \\ x + (1 + 2^{-n})y &= b \end{aligned}$$

Express x and y in terms of a, b, and 2^n.
Answer:

Solving for x and y yields $x = a - 2^n(b - a)$. and $y = 2^n(b - a)$

b) Suppose that we are using IEEE single-precision arithmetic and that the value of b changes by 1 ulp. By how much do the values of x and y change?
Answer:.

Suppose we replace b by $\hat{b} = b(1 + 2^{-p})$, where p is the precision ($p = 23$ in IEEE single precision). Then the solution to the perturbed system is

$$\begin{aligned} \hat{x} &= a - 2^n(\hat{b} - a) \\ \hat{y} &= 2^n(\hat{b} - a) \end{aligned}$$

Therefore, the computed values of x and y change by

$$\begin{aligned}\Delta x &= \hat{x} - x = -2^n(\hat{b} - b) = -2^n \mathrm{ulp}(b) \\ \Delta y &= \hat{y} - y = 2^n(\hat{b} - b) = 2^n \mathrm{ulp}(b)\end{aligned}$$

This exercise illustrates the problem of ill conditioning in linear systems. Even if n is relatively modest, say $n = 10$, then if b is subject to roundoff error, or if the value of b is known only to a few bits of accuracy due to measurement error, then the values of x and y may have no significant digits (i.e., the computed values of x and y may be garbage). Linear systems that are nearly singular are always ill conditioned.

Problem 6 (4 pts.)

The machine epsilon for IEEE single-precision numbers is $2^{-23} \approx 1.2 \times 10^{-7}$. In this respect, single-precision IEEE floating-point is roughly equivalent to 7 decimal digits. Nevertheless, 7 decimal digits do not suffice to represent an IEEE single-precision number uniquely—in fact, you need 9 decimal digits. This exercise outlines a proof of this claim.

Consider real numbers x such that $10 \le x < 16$, i.e., the interval $[10, 16)$.

a) The numbers in $[10, 16)$ that are exactly representable in 7 decimal digits are 10.00000, 10.00001, $10.00002, \cdots$, 15.99999. How many numbers are in this set?
Answer: There are 6 choices for the left of the decimal point and 10^5 choices for the right of the decimal point. Therefore, there are $6 \times 10^5 = 600,000$ decimal numbers in this set.

b) The numbers in $[10, 16)$ that are exactly representable in IEEE single precision run from

$$10_{10} = 1.01\underbrace{00\cdots00}_{21\text{ bits}} \times 2^3 \quad \text{to} \quad (16 - 2^{-20})_{10} = 1.11\underbrace{11\cdots11}_{21\text{ bits}} \times 2^3$$

How many numbers are in this set?
Answer:
There are 6 choices for the first 4 bits and 2^{20} choices for the remaining bits. Therefore, there are $6 \times 2^{20} = 6,291,456$ binary floating-point numbers in this set.

c) Apply the pigeonhole principle to the example above to show that at least two different IEEE single-precision numbers have the same 7-digit decimal representation.
Answer:
Since there are more binary floating-point numbers than 7-digit decimal numbers in this interval, the pigeonhole principle implies that at least two binary floating-point numbers must map to the same 7-digit decimal approximation. Thus 7 decimal digits do not suffice to reproduce the original single-precision floating-point number exactly.

d) Explain why 8 decimal digits also are not enough to represent IEEE single-precision numbers uniquely.
Answer:
Sample answer: The numbers in $[10, 16)$ that are exactly representable in 8 decimal digits are 10.000000, 10.000001, $10.000002, \cdots$, 15.999999. There are 6 choices for the left of the decimal point and 10^6 choices for the right of the decimal point. Therefore, there are $6 \times 10^6 = 6,000,000$ decimal numbers in this set, which is smaller the total floating-point numbers: $6,291,456$. The pigeonhole

principle implies that at least two floating-point numbers in this interval map to the same 8-digit decimal representation.

Problem 7 (5 pts.)

The floating-point representation in Eq. (1) can be stored in 32 bits, as follows: 1 bit for the sign, 23 bits for $a_1, a_2, \cdots a_{23}$, and 8 bits for the exponent. There is no need to store a_0 since it is always 1 if $x \neq 0$.

However, Eq. (1) is not the only possible way to encode a floating-point number in 32 bits. Another approach, which we'll call the $(\pm, s, l)$ format, uses a sign bit, a significand s, and an integer "level" l, as follows. Let s be expressed as a binary number such that $1 \leq s < 2$, and require that $-7 \leq l \leq 7$. We can express real numbers greater than or equal to 1 in magnitude in the following way. If $l = 0$, then the real number represented by $(\pm, s, 0)$ is $\pm s$. If $l = 1$, then $(\pm, s, 1)$ represents $\pm 2^s$; if $l = 2$, then $(\pm, s, 2)$ represents $\pm 2^{2^s}$; $l = 3$, $\pm 2^{2^{2^s}}$; and so on. Numbers between 0 and 1 in magnitude are represented by negative values of l: $(\pm, s, -1)$ represents $\pm 2^{-s}$; $(\pm, s, -2)$ represents $\pm 2^{-2^s}$, etc. The $(\pm, s, l)$ format is just as compact as the IEEE format: l requires 4 bits, the sign takes 1 bit, and s can occupy the remaining bits of a 32-bit word.

a) What real number is represented by $(+, 1.5, 0)?(-, 1.5, 1)?(+, 1.5, 2)?(-, 1.5, -1)?$

Answer:

$(+, 1.5, 0) = 1.5$; $(-, 1.5, 1) = -2^{1.5} = -2\sqrt{2}$; $(+, 1.5, 2) = 2^{2^{1.5}} = 2^{2\sqrt{2}}$; $(-, 1.5, -1) = -2^{-1.5}$

b) What is the set of representable positive values for $l = 0, 1, 2, 3, 4$?

Answer:

$l = 0 : [1, 2)$

$l = 1 : [2^1, 2^2) = [2, 4)$

$l = 2 : [2^{2^1}, 2^{2^2}) = [4, 16)$

$l = 3 : [2^{2^{2^1}}, 2^{2^{2^2}}) = [16, 2^{16})$

$l = 4 : [2^{2^{2^{2^1}}}, 2^{2^{2^{2^2}}}) = [2^{16}, 2^{2^{16}}) = [2^{16}, 2^{65,536})$

c) The number $G = 10^{100}$ is called a *googol*; 10^G is a *googolplex*. Express the largest representable value of $(+, s, 4)$ as an approximate power of G.

Answer:

$G^n = 2^{65,536} \implies n \log G = 65,536 \log 2 \implies n = (65,536 \log 2)/100 \approx 197$

So the largest representable value is approximately G^{197}.

d) Is the largest representable value of $(+, s, 5)$ greater or less than a *googolplex*? Explain.

Answer:

The largest representable value of $(+, s, 5)$ is $2^{2^{65,536}} \approx 2^{G^{197}}$

Let $2^{G^{197}} = 10^a$. So, $a = \log\left(2^{G^{197}}\right) = G^{197} \log 2 \gg G$

Therefore, $2^{G^{197}}$ is much larger than 10^G.

e) Invent your own terminology as necessary to describe the largest representable values of $(+, s, 6)$ and $(+, s, 7)$.

Answer:

Sample answer: Define H (for humongous) as $H = 2^{2^{65,536}}$. Then the largest representable value for $(+, s, 6)$ is 2^H and for (+, s, 7) is 2^{2^H}.

Although overflow and underflow are unlikely to occur in the $(\pm, s, l)$ format, there are no simple algorithms for addition and multiplication, which is why we use formats like Eq. (1).