1 We have the following two 64-bit representations of machine numbers.

a. 1 10000000101 0101000100000000000000000000000000000000000000000000

b. 0 01111111100 1100101000000000000000000000000000000000000000000000

What are the decimal values of their characteristics c and mantissas f? What decimal numbers do they represent? Write your answers as fractions.

Solution a. $s = 1$, $c = 2^0 + 2^2 + 2^{10} = 1029$, $f = 2^{-2} + 2^{-4} + 2^{-8} = \dfrac{81}{256}$.

This represents the number

$$(-1)^1 \cdot 2^{1029-1023} \cdot \left(1 + \frac{81}{256}\right) = -2^6\left(\frac{337}{2^8}\right) = -\frac{337}{4}.$$

b. $s = 0$, $c = 2^2 + 2^3 + \cdots + 2^9 = \dfrac{4(1-2^8)}{1-2} = 1020$, $f = 2^{-1} + 2^{-2} + 2^{-5} + 2^{-7} = \dfrac{101}{128}$.

So it represents the number

$$(-1)^0 \cdot 2^{1020-1023} \cdot \left(1 + \frac{101}{128}\right) = 2^{-3}\left(\frac{229}{2^7}\right) = \frac{229}{1024}.$$

---

2 Let $y = 12.837954$.

a. Find out its floating-point form using 5-digit chopping.

b. Find out its floating-point form using 6-digit rounding. (Hint: sometimes rounding may affect more than one digit.)

Solution a. $y = 0.12837954 \cdot 10^2$, so $\mathrm{fl}(y) = 0.12837 \cdot 10^2$.

b. $y = 0.12837954 \cdot 10^2$, so $\mathrm{fl}(y) = 0.128380 \cdot 10^2$.

---

3 Suppose your computer uses 64-bit-long real format. If you use it to calculate $2 + 10^{-20} - 2$, what would be the answer? Explain why it is the case. (Hint: although it is not necessary, you may use your computer to find the answer. For example, type in Matlab `2+10e-20-2` and press Enter.)

Solution The precision of 64-bit floating point numbers is roughly 10^{-16}, i.e., 10^{-20} is too small, so its floating point representation is 0. Hence, the result of the calculation would be

$$\mathrm{fl}\big(\mathrm{fl}(2) + \mathrm{fl}(10^{-20}) - \mathrm{fl}(2)\big) = \mathrm{fl}(2 + 0 - 2) = \mathrm{fl}(0) = 0.$$

4 Let

$$g(x) = \frac{e^{2x} + e^{-x} - 2}{x}.$$

a. Find $\lim_{x\to 0} g(x)$.

b. Use three-digit rounding arithmetic to evaluate $g(0.1)$. (Hint: here in the finite-digit arithmetic, the exponential function e^y is understood as $\mathrm{fl}(e^{\mathrm{fl}(y)})$.)

c. Replace each exponential function by its third Maclaurin polynomial, i.e., Taylor expansion at 0, up to the quadratic term, and repeat the above evaluation.

d. The actual value is $g(0.1) = 1.26240176$. Find relative errors for the values obtained in the above two parts. Which approach has a larger error? Explain the possible reason(s).

Solution a. By L'Hôpital's,

$$\lim_{x\to 0} \frac{e^{2x} + e^{-x} - 2}{x} = \lim_{x\to 0} \frac{2e^{2x} - e^{-x}}{1} = 1.$$

b. We first convert everything to their floating point representation:

$$\mathrm{fl}(e^{-0.1}) = \mathrm{fl}(0.904837418) = 0.905$$
$$\mathrm{fl}(e^{2\cdot 0.1}) = \mathrm{fl}(1.2214027582) = \mathrm{fl}(0.12214027582 \cdot 10^1) = 0.122 \cdot 10^1 = 1.22$$

Then

$$\mathrm{fl}(g(0.1)) = \mathrm{fl}\left(\frac{\mathrm{fl}(\mathrm{fl}(e^{2\cdot 0.1}) + \mathrm{fl}(e^{-0.1})) - 2}{0.1}\right) = \mathrm{fl}\left(\frac{\mathrm{fl}(0.2125 \cdot 10^1) - 2}{0.1}\right) = \mathrm{fl}\big(0.13 \cdot 10^1\big) = 1.3.$$

c. The third Maclaurin polynomial of our functions are

$$M_3[e^{2x}](x) = 1 + 2x + 2x^2$$
$$M_3[e^{-x}](x) = 1 - x + \frac{x^2}{2}$$

Then we can approximate $g(x)$ with

$$g(x) \approx \frac{x + \frac{5}{2}x^2}{x} = 1 + \frac{5}{2}x,$$

so

$$\mathrm{fl}(g(0.1)) \approx \mathrm{fl}(\mathrm{fl}(1) + \mathrm{fl}(0.25)) = \mathrm{fl}(1 + 0.25) = 1.25.$$

d. $\dfrac{|1.26240176 - 1.3|}{1.26240176} \approx 0.030, \quad \dfrac{|1.26240176 - 1.25|}{1.26240176} \approx 0.001$

The first approach has the larger relative error. This is likely because for $x > 0$, the Maclaurin series is an underestimate, since the Lagrange remainder of the numerator is given by, for $\xi_1, \xi_2 \in [0, 0.1]$,

$$\big(8e^{2\xi_1} - e^{-\xi_2}\big) \cdot \frac{(-0.1)^3}{6} \le \big(8e^{0.2} - e^0\big) \cdot \frac{(-0.1)^3}{6} < 0.$$

Since using three-digit rounding to calculate $g(1)$ gives an overestimate and our Maclaurin polynomials give us a slightly smaller number than that, using Maclaurin polynomials gives us a better estimate.

One calculation where we got a big error in the first method was adding together the exponentials: 2.125 became 2.13, which is a fairly large error. We didn't perform any calculation like this with the polynomials since their coefficients did not require more than 2-digits precision. This is likely the main reason why the second method worked better.