

1. Let F be a computer number system of 64 bits. Find the following

- The largest and smallest number.
- The smallest normalized positive number.
- The smallest positive number.
- Give examples of *underflow* and *overflow*.
- The machine precision.
- Find upper bounds of the absolute and relative errors of $fl_c(x)$ approximating x using the rounding approach.

Note that the specifics may differ slightly with different computers and compilers.

iSolution: For a 64-bits computer number system, the largest exponential is

$$S_{max} = 2^0 + 2^1 + \dots + 2^9 = 2^{10} - 1 = 1023.$$

- The largest fraction = $0.111\dots 1 = 1 - 2^{-52}$. The largest positive number = $2^{1023}(1 - 2^{-52}) = 8.9885 \times 10^{307}$. The smallest number = -8.9885×10^{307} .
- The smallest normalized positive number is

$$0.100\dots 1 \times 2^{-1023} = 1.1125 \times 10^{-308} \quad (1)$$

while the smallest positive number is

$$0.000\dots 0 \times 2^{-1023} = 2^{-1023} \times 2^{-52} \sim 2.4702 \times 10^{-324}. \quad (2)$$

- Any number whose magnitude is larger than 9.0000×10^{307} will be overflow. Any number whose magnitude is between 0 and 2.4702×10^{-324} will be under-flow.
- The machine precision is $\epsilon = \frac{1}{2}2^{-52} = 1.1102 \times 10^{-16}$.
- Let $x = 0.d_1d_2\dots d_nd_{n+1}\dots \times \beta^b$, $d_1 \neq 0$, $0 \leq d_i \leq \beta - 1$. Using chopping, we would have $fl_c(x) = 0.d_1d_2\dots d_n \times \beta^b$. Thus we have

$$|fl_c(x) - x| \leq \beta^{b-n}, \quad \frac{|fl_c(x) - x|}{|x|} \leq \frac{\beta^{b-n}}{\beta^{-1+b}} = \beta^{-n+1} = 2\epsilon$$

That is, the upper error bounds are twice as much as those using round-off approach.

2. Assume we use a computer to evaluate the following expressions

$$(a) f = xyz, \quad (b) f = x + y + z,$$

where x , y , and z are real numbers. Find upper bounds of absolute and relative errors. Assume all the numbers involved are in the range of the computer number system. (**Note:** Pay attention to the upper bounds and absolute values, e.g., $\delta_5 \leq 5\epsilon$ is wrong, it should be $|\delta_5| \leq 5\epsilon$.)

Solution: For $fl(xyz)$, we should have

$$fl(xyz) = x(1 + \delta_x)y(1 + \delta_y)z(1 + \delta_z)(1 + \delta_3)(1 + \delta_4)$$

The five δ 's corresponding to 3 inputs, two multiplications, therefore the absolute error is

$$|xyz - fl(xyz)| = |xyz(\delta_x) + \delta_y\delta_z + \delta_3 + \delta_4 + O(\epsilon^2)| \leq 5|xyz|\epsilon.$$

The relative error bound is simply (assuming $xyz \neq 0$)

$$\frac{|xyz - fl(xyz)|}{|xyz|} \leq 5\epsilon.$$

For the second part, we have

$$fl(x + y + z) = ((x(1 + \delta_x) + y(1 + \delta_y))(1 + \delta_3) + z(1 + \delta_z))(1 + \delta_4).$$

The five δ 's corresponding to 3 inputs, two additions, therefore the absolute error is

$$|x + y + z - fl(x + y + z)| \leq |x\delta_x| + y\delta_y + (x + y)\delta_3 + z\delta_z + (x + y + z)\delta_4 O(\epsilon^2) \leq 5(|x| + |y| + |z|)\epsilon.$$

Note that the upper bound is not unique and may have different forms. the relative error bound can be expressed as

$$\begin{aligned} \frac{|x + y + z - fl(x + y + z)|}{|x + y + z|} &\leq \left(\frac{|x\delta_x| + y\delta_y + z\delta_z}{|x + y + z|} + \frac{|(x + y)\delta_3|}{|x + y + z|} + \delta_4 \right) \\ &\leq 5 \frac{|x| + |y| + |z|}{|x + y + z|} \epsilon. \end{aligned}$$

Note that if all x, y, z have the same sign, then the relative error is bounded by 5ϵ . Otherwise, the relative error can be anything depends on the denominator.

3. Design an algorithm (in pseudo-code form) to evaluate the following

(a) $\log(1 + x)/x$ in the interval $[-0.5, 0.5]$.

(b) $b - \sqrt{b^2 - \delta}$, where b and δ are two parameters with $b^2 - \delta \geq 0$.

You need to consider all possible scenarios.

Solution: The first function has a removable singularity at $x = 0$ since $\lim_{x \rightarrow 0} f(x) = 1$. If x is small, then $\log(1 + x) = x - x^2/2 + O(x^3)$. Therefore the pseudo-code is

```

if abs(x) > 1e-15
    y = log(1+x)/x
else
    y = x - x*x/2
end

```

For the second part, if $b > 0$, we may have cancellation if δ is small, but we know $b - \sqrt{b^2 - \delta} = \frac{\delta}{b + \sqrt{b^2 - \delta}}$. Thus we can have the pseudo-code

```

if b > 0
    y= e / ( b+ sqrt(b*b-e) )
else
    y = b - sqrt(b*b-e)
end

```

Note that we use e to represent δ .

4. Which of the following two formulas in computing π is better?

$$\pi = 4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} + \dots \right)$$

$$\pi = 6 \left(0.5 + \frac{0.5^3}{2 \cdot 3} + \frac{3(0.5)^5}{2 \cdot 4 \cdot 5} + \frac{3 \cdot 5(0.5)^7}{2 \cdot 4 \cdot 6 \cdot 7} + \dots \right).$$

You can write a short Matlab code to compare. Consider both accuracy and speed.

Solution: The second formula is better for two reasons: 1), it converges much faster; 2), we only have multiplication/divisions, and absolute additions so there is no risk of cancellations as in the first formula which the signs are alternating. It appears that the first formula require less computation for each time. But we need to consider the total operations needed to reach the same order accuracy. In this sense, the second formula has also fewer computations.

5. We can use the following three formulas to approximate the first derivative of a function $f(x)$ at x_0 .

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}$$

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0 - h)}{2h}$$

$$f'(x_0) \approx \frac{f(x_0) - f(x_0 - h)}{h}$$

When we use computer to find an approximation of a derivative (used in optimization and many areas), we need to balance the errors from the algorithm (truncation error) and round-off errors (from computer).

- (a) Which formula is the most accurate in theory? *Hint:* Find the absolute error using the Taylor expansion at $x = x_0$.
- (b) Write a program to compute the derivative with
- $f(x) = x^2$, $x_0 = 1.8$.
 - $f(x) = e^x \sin x$, $x_0 = 0.55$.

Plot the errors versus h using log-log plot with labels and legends if necessary. In the plot, h should range from 0.1 to the order of machine constant (10^{-16}) with h being cut by half each time (i.e., $h = 0.1$, $h = 0.1/2$, $h = 0.1/2^2$, $h = 0.1/2^3$, \dots , until $h \leq 10^{-16}$.)

Tabulate the absolute and relative errors corresponding to $h = 0.1$, $0.1/2$, $0.1/4$, $0.1/8$, and $0.1/16$ (that is, difference choices of h compared with that used in the plots). The ratio (should

be around 2 or 4) is defined as the quotient of two consecutive errors. **Analyze and explain** your plots and tables. What is the best h for each case with and without round-off errors?

$1/h$	error (a)	ratio	error (b)	ratio	error (c)	ratio
10		–		–		–
20						
40						
80						
160						

The ratio is defined as, for example

$$ratio = \frac{|\text{error for } n = 10|}{|\text{error for } n = 20|}.$$

Solution: Using Taylor expansion at x_0 , we can easily get

$$\left| f'(x_0) - \frac{f(x_0+h) - f(x_0)}{h} \right| = \frac{h}{2} |f''(x_0)| + O(h^2)$$

$$\left| f'(x_0) - \frac{f(x_0+h) - f(x_0-h)}{2h} \right| = \frac{h^2}{6} |f'''(x_0)| + O(h^4)$$

$$\left| f'(x_0) - \frac{f(x_0) - f(x_0-h)}{h} \right| = \frac{h}{2} |f''(x_0)| + O(h^2)$$

Therefore the second formula is more accurate without presence of round-off errors. Note that for $f(x) = x^2$, the second formula gives the exact derivative!

The errors decrease by factor of two if we halve h for the first and the third formulae, and factor of four for the second formula which you should be able to see from your table except the exact one.

When h is much bigger than the machine precision, the formula error is dominant. As h gets small enough, the round-off will eventually catch up. The best h is where the magnitude of the formula and round-off error are about the same. For the first formula, that is when

$$\frac{h}{2} |f''(x_0)| \approx \frac{|f(x_0)|}{h} \epsilon,$$

which gives roughly $h = \sqrt{\epsilon} \sim 10^{-8}$. For the second formula, it should be roughly

$$\frac{h^2}{6} |f'''(x_0)| \approx \frac{|f(x_0)|}{h} \epsilon,$$

which gives roughly $h = \sqrt[3]{\epsilon} \sim 10^{-5}$. Note that, the round-off can be estimated by, for example

$$\begin{aligned} \text{ifl} \left(\frac{f(x_0+h) - f(x_0)}{h} \right) &= \frac{f(x_0+h)(1+\delta_1) - f(x_0)(1+\delta_2)}{h} (1+\delta_3) \\ &= \frac{f(x_0+h) - f(x_0)}{h} + \frac{f(x_0+h)}{h} \delta_1 + \frac{f(x_0)}{h} \delta_2 + \frac{f(x_0+h) - f(x_0)}{h} \delta_3, \end{aligned}$$

where δ_1 and δ_2 are the relative errors when we evaluate $f(x_0+h)$ and $f(x_0)$, δ_3 is the relative error for the division, in general we have $\delta_i \leq C\epsilon$ for some error constants C .

Note: Please submit any **computer code(s)** through <http://courses.ncsu.edu/ma580/> or <http://courses.ncsu.edu/csc580/> depending on the course that you registered in to save paper. But you need to attach your **plots, tables, arranged outputs, analysis** along with your homework.