MATH3230A - Numerical Analysis
Exercises on Floating point arithmetic

1 Rounding errors

1. Determine the absolute and relative errors when approximating p by p^* where
   - $p = 3$, $p^* = 3.1$;
   - $p = 0.00003$, $p^* = 0.000031$;
   - $p = 30000$, $p^* = 31000$;
2. Perform rounding on the following binary numbers and compute the relative error:
   - $(1.001101)_2$ to 3 digits after the binary point;
   - $(1.0001110101)_2$ to 2 digits after the binary point;
   - $(0.110110)_2$ to 4 digits after the binary point.
3. Show that the relative errors for rounding and chopping a binary number x to n digits are
$$\frac{\hat{x} - x}{x} \leq \begin{cases} 2^{-n} & \text{rounding,} \\ 2^{-(n+1)} & \text{chopping.} \end{cases}$$

2 Machine precision

1. For a floating-point format with 9 bits, given 1 bit is reserved for the sign of the binary number, how many bits should be assigned to the mantissa and to the exponent (unbiased) to create the largest possible binary number. Here you do not need to reserve for the case where all the entries in the exponents are 1 for special values.
2. Consider two formats: (a) 6 bits for the unbiased exponent and 5 bits for the mantissa, (b) 5 bits for the unbiased exponent and 6 bits for the mantissa. What are the total number of binary decimals that can be represented in both formats? Here we say that a number can be represented if it has the following form
$$x = (1.f_1 \cdots f_k)_2 \times 2^{(m_1 \cdots m_p)_2}$$
where $k = 5$, $p = 6$ for (a) and $k = 6$, $p = 5$ for (b).

3. For a floating-point format with 18 bits, suppose 1 bit is reserved for the sign, 12 bits for the exponent and 5 bits for the mantissa. Excluding the special cases where all the exponents are zero or are ones, compute the value which the exponent has to be offset so that there is an equal number of non-negative (counting 0) and negative exponents.

   Also compute the smallest, second smallest, largest finite and second largest finite normalized number in this format.

3 Loss of significance

1. Suggest ways to avoid loss of significance in the following computations

   - $\sqrt{x^4+4}-2$;
   - $e^x - e^1$;
   - $\log x - 1$;

2. Let x and y be positive normalized floating-point binary machine numbers of the form

$$x = r \times 2^n, \quad y = s \times 2^m \text{ for } 1 \le r, s < 2, \text{ and } n > m.$$

   Suppose

$$1 - \frac{y}{x} \le 2^{-p}$$

   show by writing $x-y$ as $z \times 2^n$ for some appropriate z that $z < 2^{-p}$. This means that at least p spurious zeros are attached to the right end of the mantissa for the difference $x - y$ and we can expect that at least p significant binary bits are lost when computing $x - y$.

3. Similar setting to the above, but now assume

$$1 - \frac{y}{x} \ge 2^{-q}.$$

   Employing a similar argument, show that at most q significant binary bits are lost when computing $x - y$.

4. For the following questions use the results of Q2:

   - find a lower bound on the input z if we want to lose at most 3 significant binary bits in the calculation $f(z) = \sqrt{z^2+1} - 1$;
   - find a lower bound on the input z if we want to lose at most 2 significant binary bits in the calculation $f(z) = \log(z+1) - \log(z)$;

4 Error analysis

1. Let ε_M denote the machine epsilon for a certain floating-point format. Suppose for $n \in \mathbb{N}$ we have values $\{\delta_i\}_{i=1}^n$ with

$$|\delta_i| \leq \varepsilon_M, \quad n\varepsilon_M < 1.$$

Using induction, show that

$$\Pi_{i=1}^n (1+\delta_i) = 1+\theta_n \text{ where } |\theta_n| \leq \frac{n\varepsilon_M}{1-n\varepsilon_M}.$$

2. Consider the matrices

$$A = \begin{pmatrix} a & b \\ 0 & c \end{pmatrix}, \quad B = \begin{pmatrix} d & e \\ 0 & f \end{pmatrix}$$

for machine numbers a, b, c, d, e, f.

- Compute the floating-point representation $fl(AB)$ of the product matrix AB.
- Express your answer in the form

$$fl(AB) = \begin{pmatrix} a & bX_1 \\ 0 & cX_2 \end{pmatrix} \begin{pmatrix} dX_3 & eX_4 \\ 0 & f \end{pmatrix} = (A+E_A)(B+E_B)$$

  with backward error matrices E_A and E_B.
- Derive an estimate for the matrix norms of E_A and E_B in terms of ε_M.

3. Consider a function operator $f : X \to Y$ between two vector spaces X and Y, and its floating-point algorithm $\tilde{f} : X \to Y$. Let ε_M denote the machine epsilon. We say that the algorithm is backward stable if

$$\tilde{f}(x) = f(\tilde{x})$$

for some $\tilde{x}$ satisfying

$$\frac{\|\tilde{x}-x\|}{\|x\|} \leq C\varepsilon_M.$$

for some constant C. Show that

- the multiplication $\tilde{f}(x_1, x_2) = fl(fl(x_1)fl(x_2))$ is backward stable;
- the addition $\tilde{f}(x_1, x_2) = fl(fl(x_1) + fl(x_2))$ is backward stable;
- the division $\tilde{f}(x) = fl(fl(x)/fl(x))$ for $x \neq 0$ is NOT backward stable;
- the operation $\tilde{f}(x) = fl(x)+1$ for x close to zero is NOT backward stable.