POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Numbers, errors and computers.

Calcoli di Processo dell' Ingegneria Chimica

Timoteo Dinelli, **Marco Mehl**

4th of October 2024.

Department of Chemistry, Materials and Chemical Enginering, G. Natta. Politecnico di Milano.
email: timoteo.dinelli@polimi.it
email: marco.mehl@polimi.it

Numbers representation.

*"In computing, floating-point arithmetic (**FP**) is arithmetic that represents subsets of real numbers using an integer with a fixed precision, called the significand, scaled by an integer exponent of a fixed base. Numbers of this form are called floating-point numbers. For example, 12.345 is a floating-point number in base ten with five digits of precision."* Wikipedia.

$$12.345 = \underbrace{12345}_{\text{Mantissa}} \times \underbrace{10^{-3}}_{\text{Exponent}}$$

Additional RECOMMENDED read:

- Floating point representations.
- LLM quantization.

Single or Double precision?

In any computer, the difference between single and double precision reflects on the number of bits used to represent a real number.

- Single precision floating point number 32 bits (4 bytes). Safe use with $\sim 7/8$ decimal digits.
- Double precision floating point number 64 bits (8 bytes). Safe use with $\sim 15/16$ decimal digits.

Elementary operations

Initialize a single precision variable in MATLAB (e.g. x = $1e^{+25}$) using the function:
x = single(1.e+25).

Working in single precision predict and calculate from x = 1.e+25 and y = 1.e+18 the following values of z:

	Single Precision	Double Precision
$z = x * y$	inf	1.0000e+43
$z = x/y$	10000000	10000000
$z = y/x$	1.0000e-07	1.0000e-07
$z = x^2$	inf	1.0000e+50
$z = y^2$	1.0000e+36	1.0000e+36
$z = 1./(x * y)$	0	1.0000e-43
$z = 1./x/y$	9.9492e-44	1.0000e-43
$z = y + 1e10$	1.0000e+18	1.0000e+18
$z = x * y/(x * y + 1)$	NaN	1

Exercises

Compute the MACHEPS

Epsilon, or machine epsilon, is an important number in computing. Machine epsilon gives the distance between a number and the next largest floating point number on your computer. This is important to calculate, as the size of the floating point number may lead to round-off errors for certain calculations. Calculating machine epsilon can be done a number of ways, and many programming languages have built-in functions that can determine this value. However, it also can be determined algorithmically with a fairly simple routine. Write a function to determine the macheps of a generic number. Plot the results for numbers from ranging from 0 to 10.

- The strategy here is to iterate as long as the difference between n and $n + epsilon/2 > 0$ by halving at each iteration the value of the machine epsilon until convergence.

Sum of the inverse of numbers

Write a script which calculates:

$$\sum_{n=1}^{1000000} \frac{1}{n}$$

in single and double precision. Then compare with the results obtained inverting the order of the sum, so by computing:

$$\sum_{n=1000000}^{1} \frac{1}{n}$$

The babilonian method

Write a function that implements the Babilonian method to compute the square root of a number with a precision of four decimal figures.

1. **Make an Initial guess**. Guess any positive number x_0.
2. **Improve the first guess**. Apply the formula $x_1 = \frac{x_0 + \frac{S}{x_0}}{2}$. The number x_1 is a better approximation to $\sqrt{S}$.
3. **Iterate until convergence**. Apply the formula $x_{n+1} = \frac{x_n + \frac{S}{x_n}}{2}$ until the convergence is reached.

Convergence is reached when the digits of x_{n+1} and x_n agree to as many decimal places as you desire.

Vancouver: a nickel at a time

Analogously to what happened on the Vancouver stock market (reference), starting from a stock value of 1000, check what happens when a random variation of $\pm 1\%$ in its value is iterated for 10000 times. Try to round or approximate (truncate) to the lower or upper second decimal figure. Compare the results with the number obtained using the computer precision.

Please verify on the help page in Matlab how the functions *rand, floor, ceil, fix, round* work.

$>> a = 1.2345$
$>> ceil(a*100)/100 \longrightarrow ans = 1.2400$
$>> floor(a*100)/100 \longrightarrow ans = 1.2300$
$>> round(a*100)/100 \longrightarrow ans = 1.2300$

$>> a = -1.2345$
$>> ceil(a*100)/100 \longrightarrow ans = -1.2300$
$>> floor(a*100)/100 \longrightarrow ans = -1.2400$
$>> round(a*100)/100 \longrightarrow ans = -1.2300$

Thank you for the attention!