Computer Numbers and their Precision, I Number Storage

Learning goal: To understand how the ways computers store numbers lead to limited precision and how that introduces errors into calculations.

Learning objective

Computational and mathematical objectives:

- To understand that exact or whole numbers can be stored as integers.
- To understand that the division of two integers is always rounded down to a smaller integer.
- To understand that numbers can also be stored in scientific or engineering notation (floating point numbers).
- To understand the advantages and disadvantages of floating point numbers.
- To understand that computer errors occur when an integer becomes too large, or too negative.
- To understand that computer errors known as *overflow* occur when the exponent of a floating point number becomes too large.
- To understand that computer errors known as *underflow* occur when the exponent of a floating point number becomes too negative.
- To understand that truncation or roundoff occurs when the mantissa of a floating point number becomes too long.
- To understand some of the consequences of the roundoff of floating point numbers.

Science model/computation objectives:

- To understand computer storage of floating point numbers, and be able to empirically determine *machine precision*
- To understand that just as laboratory experiments always have limits on the precision of their measurements, so too do computer simulations have limits on the precision of their numbers.
- To understand the difference between “precision” and “accuracy”.
- Students will practice the following scientific skills:
    - Doing numerical experiments on the computer.

Activities

In this lesson, students will:

- Perform calculations on the computer as experiments to determine the computer’s limits in regard to the storage of numbers.
- Perform numerical calculations to see how the rules of mathematics are implemented.
- Sum the series for the sine function to see the effect of error accumulation.