

Marco Karier

Advance Software Engineering

Pre-task-1

April 11th, 2017

Numerical Methods:

A mathematical tool designed to solve numerical problems. Some math problems like polynomial problems can get very large and can be solved in a way to keep accuracy by performing the equation in multiple steps. However, if the problem gets too large then a computer can't keep track of the accuracy so then a computer needs to be able to keep accuracy but still perform the action. Basically, as the computer solves math problems it will adjust the numbers it stores to keep them as accurate as possible while still solving the math problems. There has been many analysis going into the most efficient way rounding numbers without major loss of accuracy.

Resource: https://en.wikipedia.org/wiki/Numerical_analysis

(Computer) number representation:

The internal representation of numeric values in computers. Normally the numbers are stored in groups of bits which makes basic math easier for the computer. In some cases, to save space, numbers have been divided into groups of bytes to represent numbers. The output representations can be different based on the user format. Usually the numbers are stored as either real or integer numbers which can behave differently but can be able to be converted between the two internally. In low level assembly languages, much of the mathematical equations must be performed by the programmer. Higher level languages have ways to perform those actions but they have their own rounding errors.

Resource: https://en.wikipedia.org/wiki/Computer_number_format

Fixed-point representation:

A fixed-point number representation is a real data type for a number that has a fixed number of digits after the decimal point (Also referred to as a radix point). It is used for more accurate calculations for things like bank transactions or scientific data collection. The problem is that we can't just round the decimal point at a random spot its best to do it at the furthest point

Resource: https://en.wikipedia.org/wiki/Floating-point_arithmetic

Floating-point hazards:

A fixed-point number allows for better accuracy but the arithmetic can be very space consuming. The danger is rounding the numbers at the wrong byte and causing a loss of accuracy. Another danger is overflowing. If for example, a computer is dividing numbers like 22 divided by 7, which are not ending numbers and will cause the memory to over flow. Also, the different ways the number data type stores number effect how the numbers accuracy and the way they do math. When programing and floating point accuracy is an import factor then picking the correct data type has to be considered. Some datatypes can more bytes for more accurate float storage but might have problems when rounding.

Resource: https://en.wikipedia.org/wiki/Floating-point_arithmetic#Accuracy_problems