PH502: Scientific Programming Concepts

Irish Centre for High End Computing (ICHEC)

September 23, 2020

Overview

- In this lecture we will continue on from last week discussing floating point and logical arithmetic.
- This week's practical will demonstrate the issue with floating point arithmetic.

Floating Point Arithmetic

- Not only are there errors in representing numbers but additional errors are introduced by arithmetic operations.
- The IEEE standard specifies that these operations must return the correctly rounded result.
- Floating point addition is not associative and multiplication is not associative nor distributive. This is due to rounding. Assume each floating point number has 7 significant figures and let $a = 1234.567, b = 45.67834, c = 0.0004$.

$$\begin{aligned} a + b &= 1280.245 \\ (a+b)+c &= \color{red}{1280.245} \\ b + c &= 45.67874 \\ a + (b+c) &= \color{red}{1280.246} \end{aligned}$$

- Even though the above error is small, when performing many operations these small errors can accumulate.

Rounding

- In the course of a calculation if the result is not a number that can be represented, it must be rounded to one that can.
- There are different rounding techniques that can be employed:
  1. "Round to nearest", number is rounded to the nearest value, if the number is half-way between then rounded up to an even number and down to an odd one.
  2. "Round to plus infinity", number is rounded to the smallest value greater than the original.
  3. "Round to minus infinity", number is rounded to the largest value that is smaller than the original.
  4. "Round to zero", number is rounded to the closest to zero.

Examples of Rounding

- Here the rounding modes are illustrated by rounding floating point numbers to integers.

Mode Number	Nearest	$+\infty$	$-\infty$	Zero
1.5	2	2	1	1
-1.5	-2	-1	-2	-1
2.5	2	3	2	2
-2.5	-2	-2	-3	-2

Avoiding Over/Underflows

- An arithmetic operation can overflow (max value exceeded), result $r - value = \pm\infty$. Underflow is when $r - value$ is smaller than minimum, result set to ± 0.
- If ab and cd underflow because b and d are very small, then

$$\frac{(ab)}{(cd)} = NaN \qquad (1)$$

$$\left(\frac{a}{c}\right) \times \left(\frac{b}{d}\right) \neq NaN \qquad (2)$$

- If underflowing sets the result to zero then in the top equation there is a division by zero. Dividing by zero may result in a NaN or set to $\pm\infty$.
- It is not always possible to avoid such errors. Restructuring expressions can help as well as using double precision variables.

Logical Arithmetic

- Logical expressions can be implied when the result is either TRUE or FALSE, *e.g.* $a > b$, $a \leq b$.
- Important generic expressions are: a equals b, a not equal to b.
- Logical variables or expressions can be combined. Below is the truth table for logical operators, *Not*, *And* and *Or*. "T" is true and "F" is false.

Operator	a	b	Result
Not	T		F
	F		T
And	T	T	T
	T	F	F
	F	F	F
Or	T	T	T
	T	F	T
	F	F	F