University of Victoria
Computer Science

CSC349A Numerical Analysis

Lecture 4

Rich Little

University of Victoria

2018

Table of Contents I

Floating-point arithmetic

University of Victoria Computer Science

Floating-point arithmetic is a simulation of real arithmetic.

- We will use the notation fl to denote the floating-point representation of a real number x as $fl(x)$.
- Also, the floating-point representation of arithmetic operations such as:

$$fl(a+b), fl(a-b), fl(a \times b), fl(a/b)$$

  where a and b are floating-point numbers.
- The implementation of these floating-point operations (in either software or hardware) depends on several factors, and includes for example choices such as whether to use rounding or chopping and the number of significant digits used for floating-point addition and subtraction.

Idealized floating-point arithmetic

University of Victoria Computer Science

Definition

For simplicity, we will consider only **"idealized" floating-point arithmetic** which is defined as follows. Let $\bullet$ denote any of the basic arithmetic operations $+ - \times /$ and let x and y denote floating point numbers. $fl(x \bullet y)$ is obtained by performing **exact arithmetic** on x and y, and then **rounding or chopping** this result to k significant digits.

Note 1

University of Victoria Computer Science

Although no actual digital computers or calculators implement floating-point arithmetic that way (it's too expensive as it would require a very long accumulator for doing addition and subtraction), idealized floating-point arithmetic:

- Behaves very much like any actual implementation
- Is very simple to do in hand computations
- Has accuracy almost identical to that of any implementation

Note 2

If *fl* is applied to an arithmetic expression containing more than one arithmetic operation, then each of the arithmetic operations must be replaced by its corresponding floating-point operation.

Note 3

With idealized floating-point arithmetic, the maximum relative error in $fl(x \bullet y)$ is the same as the maximum relative error in converting a real number z to floating-point form. Thus, for a **single** floating-point operation $+ - \times /$, the **relative error is very small**: it is $< b^{1-k}$ with chopping, or $\frac{1}{2} b^{1-k}$ (with rounding). However, the relative erorr in a floating-point computation **might be large** if more than one floating-point operation is performed.

Table of Contents I

Subtractive cancellation

University of Victoria Computer Science

- Loss of significant digits due to the subtraction of *nearly* equal floating-point numbers.
- In the case of $+$, $\times$ and $/$, there is little significant loss of precision.
- However, this is not the case with subtraction, where close to all the significant digits may be lost.
- Handout 4 presents four examples of this and some of the ways of avoiding it.

Subtractive cancellation - examples

University of Victoria Computer Science

Using $b = 10$, $k = 4$ idealized floating-point aritmetic evaluate each of the following expressions for the given x.

- Example 1: $fl(\sqrt{x^2+1} - x)$ at $x = 65.43$ using rounding
- Example 2: $fl(x - \sin x)$ at $x = 0.01234$ using chopping
- Example 3: $fl(1 - \sin x)$ at $x = 1.56$ radians using rounding
- Example 4: $fl(\frac{1}{2x-1} - \frac{x+2}{x-2})$ vs. $fl(\frac{-2x(x+1)}{(2x-1)(x-2)})$ for values of x near -1 and 2 using rounding