These are the lecture notes for CSC349A Numerical Analysis taught by Rich Little in the Spring of 2018. They roughly correspond to the material covered in each lecture in the classroom but the actual classroom presentation might deviate significantly from them depending on the flow of the course delivery. They are provide as a reference to the instructor as well as supporting material for students who miss the lectures. They are simply notes to support the lecture so the text is not detailed and they are not thoroughly checked. Use at your own risk.

1 Overview

The material covered in this lecture is partially based on handout 3. This material consists of the floating point number representation and round off errors.

2 Number systems

2.1 Round-off errors

Round-off errors originate from two factors:

- finite representations of possibly infinitely long numbers
- finite range of values from a possibly infinite range

All dependent on the *word size* - maximum size of the string of bits used.

2.2 Number systems

- **Decimal:**
  - base-10
  - digits: 0,1,2,3,4,5,6,7,8,9
  - powers of 10 positional system
  - Ex: 86409
- **Binary:**

  - base-2
  - digits: 0,1
  - powers of 2 positional system
  - Ex: 101011

2.3 Computer representation

- Positive integers
  - What we can represent depends on the word size
  - Ex: 8-bits, then $43_{10} = 00101011_2$
  - What is the range? $2^8 = 256$ values from 0 to $2^8 - 1$
  - Ex: 16-bits, then $43_{10} = 0000000000101011_2$
  - What is the range? $2^{16} = 65,536$ values from 0 to $2^{16} - 1$
- Negative integers
  - Signed magnitude method - use leftmost bit for the sign
  - usually 0 for '+' and 1 for '-'
  - Ex: 8-bits, then $-43_{10} = 10101011_2$
  - What is the range? only 7 bits so 128 values with a lead 0 and 128 with a lead 1
  - But, two of them - 10000000 and 00000000 - represent the same number, 0
  - We usually let 10000000 = -128 giving the range -128 to 127

2.4 Real numbers

- How do we interpet real numbers in decimal?
- Ex: Consider 82.3801
- What about in binary?
- Ex: Consider 101.1101
- Going from decimal to binary with real numbers?
- Now, how do we represent them in a computer?

3 Floating-point number system

A floating-point number system is a finite approximation to the (infinite) real/complex number system.

In normalized floating-point number systems real numbers are represented in a form:

$$\pm 0.d_1 d_2 d_3 \dots d_k \times b^e \tag{1}$$

Examples $+0.1234 \times 10^2$, -0.1111×2^4, three fingered alien floating-point number $+0.2222 \times 3^3$, not normalized 0.01111×10^2 should be instead 0.1111×10^3.

The first part is called the *mantissa*, b is the base and e is the *exponent.* The normalization refers to the property that the first digit of the *mantissa* should be non-zero i.e $1 \leq d_1 \leq b-1$. The remaining digits can be zero and are also constrained by the base i.e $0 \leq d_i \leq b-1$. Because $d_1 \neq 0$, k is the number of significant digits in the mantissa. It is called the *precision* of the the floating point system. For example in a floating-point binary system each digit will be either 0 or 1 and in a deciaml floating-point system each digit will be between 0 and $9 = 10 - 1$. Common computer examples $b = 2, k = 24$ single precision, $b = 2, k = 53$ double precision.

Consider the interval between any 2 consecutive powers of the base b let's say b^{t-1} and b^t. Then the first floating point number in that interval will be $0.100 \dots 0 \times b^t = b^{t-1}$. The next higher number in that representation will be $0.100 \dots 1 \times b^t$ and so on until the highest number that uses b^{t-1} as the base which will be $0.(b-1)(b-1) \dots (b-1) \times b^t$. For example in decimal that would be $0.999 \dots 9 \times 10^t$.

All floating point numbers with exponent equal to t are in the interval $[b^{t-1}, b^t)$. In this interval there are exactly $(b-1)b^{k-1}$ distinct floating point nubmers and they are equally spaced.

The distance between any 2 consecutive numbers is:

$$\frac{b^t - b^{t-1}}{(b-1)b^{k-1}} = \frac{(b-1)b^{t-1}}{(b-1)b^{k-1}} = b^{t-k} \tag{2}$$

The spacing between numbers gets larger as t gets larger. In this course we will frequently be using $b = 10$ and precision $k = 4$ as it makes calculations by hand easier and more easily understood by humans.