MTH 343 Numerical Analysis Lecture 2: Review of Computer Arithmetic

Sheikh Abdul Raheem Ali

February 1, 2019

Remarks

1. Numerical Analysis requires such tedious & repetitive operations that only a computer can perform quickly & without and mistakes.
2. Computers are dumb and must be given complete instructions of every step. Programs can be written in any language you like.
3. Writing code is not very important because extensive commercial software packages are available.
   (a) IMSL: International Mathematics & Statistics Library
   (b) NAG: Numerical Algorithm Group
   (c) LAPACK: Linear Algebra package

   Alternatives: Computer Algebra Systems

   (a) Mathematica
   (b) Maple
   (c) MATLAB

Floating-Point Arithmetic

In computers, numbers are stored as floating point quantities in the general form:

$$\pm \cdot (d_1 d_2 d_3 \dots d_p) \cdot \beta^e,$$

where p = precision, the number of significant bits (digits), e = an integer exponent ranging from E_{min} to E_{max}, β = the number base, normally 2, 10, 16, d_i : ranges from 0 to $\beta - 1$, and $d_1 d_2 d_3 \dots d_p$ is called the fractional part (mantissa).

Sometimes numbers are normalized: $0.023- > 0.23 \cdot 10^{-1}$

Let us examine the case $\beta = 10$ (Decimal)

$$\begin{aligned} 3216 &= 3 \cdot 10^3 + 2 \cdot 10^2 + 1 \cdot 10^1 + 6 \cdot 10^0 \\ &= 10^4(3 \cdot 10^{-1} + 2 \cdot 10^{-2} + 1 \cdot 10^{-3} + 6 \cdot 10^{-4}) \\ &= (.3216) \cdot 10^4 \end{aligned}$$

Now let us examine the case $\beta = 2$ (Binary)

$$\begin{aligned} 65 &= 2^6 + 2^0 \\ &= 2^7(2^{-1}) + 2^{-7} \\ &= (.1000001)_2 \cdot 2^7 \end{aligned}$$

$$\begin{aligned} 23 &= 2^4 + 2^3 + 2^2 \\ &= 2^5(2^{-1} + 2^{-2} + 2^{-3}) \\ &= (.111)_2 \cdot 2^5 \end{aligned}$$

$$\begin{aligned} 85 &= 2^6 + 2^4 + 2^2 + 2^0 \\ &= 2^7(2^{-1} + 2^{-3} + 2^{-6} + 2^{-7}) \\ &= (.1010011)_2 \cdot 2^7 \end{aligned}$$

$$\begin{aligned} 5.75 &= 2^2 + 2^0 + 2^{-1} + 2^{-2} \\ &= 2^3(2^{-1} + 2^{-3} + 2^{-4} + 2^5) \\ &= (.10111)_2 \cdot 2^3 \end{aligned}$$

$$0.6 = (.1001100110011001\ldots)_2$$

This last example shows us a conversion error: the decimal is recurring, but since the computer only has a finite number of bits, the value is truncated at some point.

Definition 1 (Round off error:) *The error that is produced when a computer is used to perform real-number calculations is called round-off error.*

There are two ways of truncating the mantissa:

1. Chopping
2. Rounding

Ex. $13.76573 = .1376|573 \cdot 10^2$
4 digits chopping: $.1376 \cdot 10^2$ 4 digits rounding: $.1377 \cdot 10^2$
Numbers are **rounded** when stored in the floating point format.