Numerical Analysis

Math 370 Fall 1998 MWF 11:30am - 12:25pm
©1998 Ron Buckmire Fowler 127

Class 2: **Wednesday September 2**

SUMMARY Floating Point Numbers and Round-Off Error

CURRENT READING Burden & Faires Section 1.2

Round-Off Error
Recall that a typical single-precision floating-point number $fl(x)$ is represented in a computer by a 32-bit "word":

s	c (7-bits)	q (24-bits)
0	1000010	101100110000010000000000

In this case,

$$fl(x) = (-1)^s \times q \times 16^{c-64}$$

where the signum, characteristic and mantissa are below.

$$\begin{aligned} s &= 0 \\ c &= 1000010_2 \\ q &= 0.101100110000010000000000_2 \end{aligned}$$

We know that $1000010_2 = 66_{10}$ and that $0.101100110000010000000000_2 = 0.6992797852_{10}$
Use the formula for $fl(x)$ to write down the decimal number x this represented by this bit of computer data:

Now write down the data representation for the machine number which is NEXT SMALLEST to $fl(x)$

$fl(x)_{prev} =$

Now write down the data representation for the machine number which is NEXT LARGEST to $fl(x)$

$fl(x)_{next} =$

Now, if we had lots of time, and a computer which kept a lot of significant digits, we could compute that $fl(x)_{prev} = 179.0156097412109375$ and $fl(x)_{next} = 179.0156402587890625$

Questions
What does this tell you about how this computer will represent **any** number between 179.0156097412109375 and 179.0156402587890625?

What can you conclude about the different between the "real number line" and the "machine number line"? In what ways are they different?

Floating Point Numbers

We can represent the machine numbers stored using the previous data representation as having the form

$$\pm 0.d_1 d_2 d_3 \cdots d_k \times 10^n, \qquad 1 \leq d_1 \leq 9, 0 \leq d_i \leq 9$$

In our specific case $k = 6$ and $-78 \leq n \leq 76$

Any positive real number y can be normalized to be written in the form

$$y = 0.d_1 d_2 d_3 \cdots d_k d_{k+1} \cdots \times 10^n$$

<u>GROUPWORK</u>

Write down the following numbers in scientific notation using the form y is written in.

$$0.000747 = \qquad\qquad 314.159265 =$$

$$970000000 = \qquad\qquad -42.0 =$$

Will you be able to represent all these numbers perfectly accuately if you only get to keep 6 significant figures (i.e. $k = 6$)?

How do computer manufacturers solve the problem of representing real numbers using a finite number of digits? Clearly an approximation to the number has to be made. The two choices are:

Chopping

In this case all the digits after d_k are **ignored** ("chopped off")

Rounding

In this case if the value of $d_{k+1} \geq 5$ then d_k is replaced by $d_k + 1$

<u>Exercise</u>

Write down the decimal machine number representation for 3546.16527

(a) using chopping

(b) using rounding

Absolute Error and Relative Error

If $\tilde{p}$ is an approximation to p, the **absolute error** is $|\tilde{p} - p|$, and the **relative error** is $\dfrac{|\tilde{p} - p|}{|p|}$, provided $p \neq 0$

<u>Example</u>

Let's compute the relative and absolute errors involved in chopping and rounding 3546.16527 using a 6-digit decimal machine number representation.