Numerical Analysis

Math 370 Spring 2009 MWF 11:30am - 12:25pm Fowler 110

 http://faculty.oxy.edu/ron/math/370/09/

Class 1

SUMMARY Machine Representation of Numbers
CURRENT READING Mathews, Chapter 1

Warm-Up Computers generally do **not** represent numbers using the decimal (base 10) system. Instead, they commonly represent numbers using binary. We shall warm up by trying to recall how to convert numbers from one base to another. In groups of 2 or 3 do the following exercises

$$1000010_{\mathbf{2}} = \qquad 127_{\mathbf{10}} =$$

$$.101100110000001_{\mathbf{2}} = \qquad 66_{\mathbf{10}} =$$

DEFINITION

A **machine number** is the name we give to the representation of an actual number which a computer stores in memory. In general, computers use a **normalized, floating-point binary representation of a number**. Instead of storing the quantity x a computer stores a binary approximation to it, which we shall write as $fl(x)$.

We call the difference between x and $fl(x)$ the **round-off error**.

For example, in certain IBM computers,

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

The number q is called the **mantissa**. It is a 24-bit finite binary fraction.
The integer c is called the **exponent** or, sometimes, the **characteristic**.
The integer s is called the **sign bit**. (0 is positive, 1 is negative)

Here is how a typical single-precision floating-point number $fl(x)$ is represented in a 32-bit computer:

0	1000010	1011 0011 0000 0100 0000 0000

Let's compute what decimal number this represents:

Write down the machine number which is NEXT SMALLEST

Write down the machine number which is NEXT LARGEST

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?

Write down the machine number you think is THE LARGEST positive number this computer can represent in memory

$Z =$ 

$$Z =$$

Write down the machine number you think is THE SMALLEST positive number this computer can represent in memory

$A =$ 

$$A =$$

Overflow and Underflow

If the computer has to represent a number greater than Z an error called **OVERFLOW** occurs and all computations cease.

If the computer has to represent a number smaller than A an error called **UNDERFLOW** occurs and in most cases the number is actually replaced by a zero.

Decimal Machine Numbers and Floating Point Numbers

We can represent the machine numbers from above as having the form

$$\pm 0.d_1d_2d_3 \cdots d_k \times 10^n, \qquad 1 \leq d_1 \leq 9, 0 \leq d_i \leq 9$$

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

$$y = 0.d_1d_2d_3 \cdots d_kd_{k+1} \cdots \times 10^n$$

GROUPWORK

Write down the following numbers in the same notation using that y is written in.

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

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

Questions

Will you be able to represent all these numbers perfectly accurately 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

Rounding

Exercise

Write down the decimal machine number (floating point) representation for 3546.16527

(a) using 6-digit chopping

(b) using 6-digit 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 $p \neq 0$

Example

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