Class 1: **Monday August 31**

SUMMARY Machine Representation of Numbers

CURRENT READING Burden & Faires Section 1.2

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

$$100010_{\mathbf{2}} = \qquad\qquad 127_{\mathbf{10}} =$$

$$.10110011000001_{\mathbf{2}} = \qquad\qquad 66_{\mathbf{10}} =$$

Definitions A **machine number** is the name we give to the representation of an actual number which a computer stores in memory. 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	101110011000001000000000

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

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

$Z =$

$$Z =$$

Write down the machine number which 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

We can represent the machine numbers from above 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

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

So, how is this number represented by the computer, since it can only use a finite number of digits?

Chopping

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 $p \neq 0$