Machine Numbers

In computers the storage of and calculations with (real) numbers are implemented using the **decimal representations** of these numbers. Thus, there is an immediate problem, because (as we have seen) the representations of these numbers generally require **infinite decimal strings** and a computer can store at most only a **finite** amount of information. For this reason, when using computers to deal with real numbers we have to make approximations by discarding the infinite tail in the decimal representation after some specific point.

Definition : By a (non-zero) **machine number** we mean a number represented as a decimal of the form:

$$x = \pm\,(0\,.\,c_1\,c_2\,c_3\,\ldots\,c_n) \times 10^m,$$

where $c_1,\ c_2,\ c_3,\ \ldots\ ,\ c_n \in \{0,\ 1,\ 2,\ \ldots\ ,\ 9\}$ with $c_1 \neq 0$. Both n and m are integers. The integer n is called the word length and m is called the exponent.

Examples :

1. $x = 0.1234567 \times 10^4$ has word length $= 7$ and exponent $= 4$.

   If we were to work "by hand" with this number we would not write it in this way, rather we would write simply $x = 1234.567$.

2. $y = 0.9999 \times 10^{-2}$ has word length $= 4$ and exponent $= -2$.

   In this case, "by hand" we would write $y = 0.009999$.

Given any real number, for example $x = 0.00777777\ \ldots$ or $y = -568.544444\ \ldots$, we first convert this number to a decimal of the form:

$$\pm\,(0\,.\,c_1\,c_2\,c_3\,\ldots\,c_n\,\ldots) \times 10^{\text{some integer}}, \qquad \text{where } c_1 \neq 0.$$

So here we would write

$$x = (0.777777\ldots) \times 10^{-2} \quad \text{and} \quad y = -(0.568544444\ldots) \times 10^3.$$

Once we have done this, and once we have decided on the word length, we must eliminate the infinite tail of the decimal. This we do in one of two ways, namely, by chopping or by rounding. This we now explain.

Chopping : The number $x = \pm\,(0\,.\,c_1\,c_2\,c_3\,\ldots\,c_n\,c_{n+1}\,\ldots)\times 10^m$, where $c_1 \neq 0$, is converted to a machine number of word length n, using chopping, simply by deleting all the digits after c_n. That is x is replaced by

$$\tilde{x} = \pm\,(0\,.\,c_1\,c_2\,c_3\,\ldots\,c_n\,)\times 10^m.$$

Example : The real numbers $x = 0.00777777\ldots$ and $y = -568.544444\ldots$, are written as machine numbers of word length 5, using chopping as follows:

- x is replaced by $\tilde{x} = (0.77777\,)\times 10^{-2}$, and
- y is replaced by $\tilde{y} = -(0.56854\,)\times 10^3$

.

Rounding : The number $x = \pm\,(0\,.\,c_1\,c_2\,c_3\,\ldots\,c_n\,c_{n+1}\,\ldots)\times 10^m$, where $c_1 \neq 0$, is converted to a machine number of word length n, using rounding, as follows:

- **IF** $c_{n+1} < 5$, replace x by

  $$\tilde{x} = \pm\,(0\,.\,c_1\,c_2\,c_3\,\ldots\,c_n\,)\times 10^m$$

  just as we did in the case of chopping.

- **Else** (That is, if $c_{n+1} \geq 5$) replace x by

  $$\tilde{x} = \pm\,(0\,.\,c_1\,c_2\,c_3\,\ldots\,c_{n-1}\,c_n\;+10^{-n}\,)\times 10^m.$$

  That is, delete all digits after c_n and then increase the digit c_n by 1.

Example : The real numbers $x = 0.00777777\ldots$ and $y = -568.544444\ldots$, are written as machine numbers of word length 5, using rounding as follows:

- x is replaced by $\tilde{x} = (0.77778\,)\times 10^{-2}$, because the 6^{th} (non-zero) digit $= 7 > 5$,
- y is replaced by $\tilde{y} = -(0.56854\,)\times 10^3$, because the 6^{th} (non-zero) digit $= 4 < 5$.

Remark [1]: When using rounding, one other small readjustment will have to made in cases like $x = 99.9998877\ldots$ for example. To write x as a as machine number of word length 5, using rounding we proceed as follows:

$$[\,99.9998877\ldots\,] \longrightarrow [\,(0.999998877\ldots)\times 10^2\,] \overset{\text{rounding}}{\longrightarrow} [\,(1.0000)\times 10^2\,] \longrightarrow [\,(0.1)\times 10^3\,].$$

The small readjustment alluded to above is given at the last arrow.

Remark [2]: The **maximum error** that arises when we delete the tail of a decimal is calculated as follows: If

$$x = (0 \,.\, c_1\, c_2\, c_3\, \ldots\, c_n\, c_{n+1}\, \ldots) \quad \text{is replaced by} \quad \tilde{x} = (0 \,.\, c_1\, c_2\, c_3\, \ldots\, c_n\,),$$

then the error is given by

$$\begin{aligned} \text{Error} &= 0 \,.\, \underbrace{0\,0\,0\,\ldots\,0}_{n \text{ terms}}\, c_{n+1}\, \ldots \\ &\leq 0 \,.\, \underbrace{0\,0\,0\,\ldots\,0}_{n \text{ terms}}\, 9\,9\,9\,9\,\ldots \\ &= 0 \,.\, \underbrace{0\,0\,\ldots\,0\,1}_{n \text{ terms}} \\ &= \left(\frac{1}{10}\right)^n. \end{aligned}$$

That is, the error introduced, when all terms of a decimal which are more than n digits beyond the decimal point are deleted, is at most $(1/10)^n$.

Errors in Machine Arithmetic : Errors arise when computers carry out calculations because the computer can store only that number of digits given by the word length. Let's consider the following examples:

Example [1]: If $x = 8/9$ and $y = 1/9$, calculate $x + y$ using (1) ordinary arithmetic, (2) machine arithmetic of word length five with chopping and (3) machine arithmetic of word length five with rounding.

Solution:

1. Ordinary arithmetic: $x + y = \left[\frac{8}{9} + \frac{1}{9}\right] = \left[\frac{9}{9}\right] = 1.$

2. Machine arithmetic of word length five with chopping: To begin with, note that in ordinary arithmetic, we have the following decimal expansions:

$$x = \frac{8}{9} = 0.8888888\ldots \quad \text{and} \quad y = \frac{1}{9} = 0.1111111\ldots$$

Thus, when working in word length five with chopping, we replace x by $\tilde{x}$ and replace y by $\tilde{y}$, where

$$\tilde{x} = 0.88888 \quad \text{and} \quad \tilde{y} = 0.11111$$

so that

$$\tilde{x} + \tilde{y} = \begin{cases} 0.88888 \\ 0.11111 \\ \hline 0.99999 \end{cases}$$

That is, $\tilde{x} + \tilde{y} = 0.99999$ so that there is an error of 0.00001 when compared to the correct answer of 1. An error of this order of magnitude seems to be reasonable enough since by chopping we have already introduced errors of this order.

3. Machine arithmetic of word length five with rounding: In this case, when working in word length five with rounding, we replace x by $\tilde{x}$ and replace y by $\tilde{y}$, where

$$\tilde{x} = 0.88889 \quad \text{and} \quad \tilde{y} = 0.11111$$

so that

$$\tilde{x} + \tilde{y} = \begin{cases} 0.88889 \\ 0.11111 \\ \hline 1.00000 \end{cases}$$

Thus, in this case no error arises though this is not usually the case when rounding.

The Growth of Errors : One of the main problems with error is that it tends to grow as computations become complex. The reason is that at each step of a computation the maximum number of digits (after the first non-zero digit) that can be stored is equal to the word length. Thus, in general, at each step of the computation some additional accuracy is being lost.

Example [2] : Calculate $(1.154)^2$ using: (1) ordinary arithmetic and (2) machine arithmetic of word length four with chopping.

Solution:

1. Ordinary arithmetic: $(1.154)^2 = 1.331716$.

2. Machine arithmetic of word length four with chopping: The most important thing to note here is that at each step of the computation and for each number at most 4 digits after the first non-zero digit may be retained. To begin with, note that: $1.154 = 0.1154 \times 10$

$$\begin{array}{ll} 0.1154 \times 10 & \\ 0.1154 \times 10 & \\ \hline 0.4616 \times 10^{-2} & (\text{ i.e. } 0.1154 \times 0.0004 \times 10^2 \text{ }) \\ 0.5770 \times 10^{-1} & (\text{ i.e. } 0.1154 \times 0.005 \times 10^2 \text{ }) \\ 0.1154 \times 10^{0} & (\text{ i.e. } 0.1154 \times 0.01 \times 10^2 \text{ }) \\ 0.1154 \times 10^{1} & (\text{ i.e. } 0.1154 \times 0.1 \times 10^2 \text{ }) \end{array}$$

Now (starting with the last) we add these latter four numbers together, using machine arithmetic of word length four with chopping, to get:

$$\begin{aligned} (0.1154 \times 10)^2 &= (0.1154 + 0.0115 + 0.0057 + 0.0004) \times 10 \\ &= (0.133) \times 10 \end{aligned}$$

Example [3] : Calculate $\dfrac{(1.154)^2 - 1.33}{2}$ using:

(1) ordinary arithmetic and (2) machine arithmetic of word length four with chopping.

Solution:

1. Ordinary arithmetic:

$$\begin{aligned}\frac{(1.154)^2 - 1.33}{2} &= \frac{1.331716 - 1.33}{2} \\ &= \frac{0.001716}{2} \\ &= 0.000858 \\ &= (0.858) \times 10^{-3}\end{aligned}$$

2. Machine arithmetic of word length four with chopping: Again at each step of the computation and for each number at most 4 digits after the first non-zero digit may be retained.

$$\begin{aligned}\frac{(1.154)^2 - 1.33}{2} &= \frac{[\,(0.1154) \times 10\,]^2 - (0.133) \times 10}{2} \\ &= \frac{[\,(0.133)] \times 10 - (0.133) \times 10}{2} \quad \text{by Example [2].} \\ &= \frac{(\,0.133 - 0.133) \times 10}{2} \\ &= 0.\end{aligned}$$