Chapter1: Introduction

1.1 REPRESENTATION OF NUMBERS ON A COMPUTER (Decimal and binary representation)

Numbers can be represented in various forms. The familiar decimal system (base 10) uses ten digits 0, 1, ... , 9. A number is written by a sequence of digits that correspond to multiples of powers of 10. As shown in Fig. 1-1, the first digit to the left of the decimal point corresponds to 10^0. The digit next to it on the left corresponds to 10^1 , the next digit to the left to 10^2 , and so on. In the same way, the first digit to the right of the decimal point corresponds to 10^{-1}, the next digit to the right to 10^{-2}, and so on.

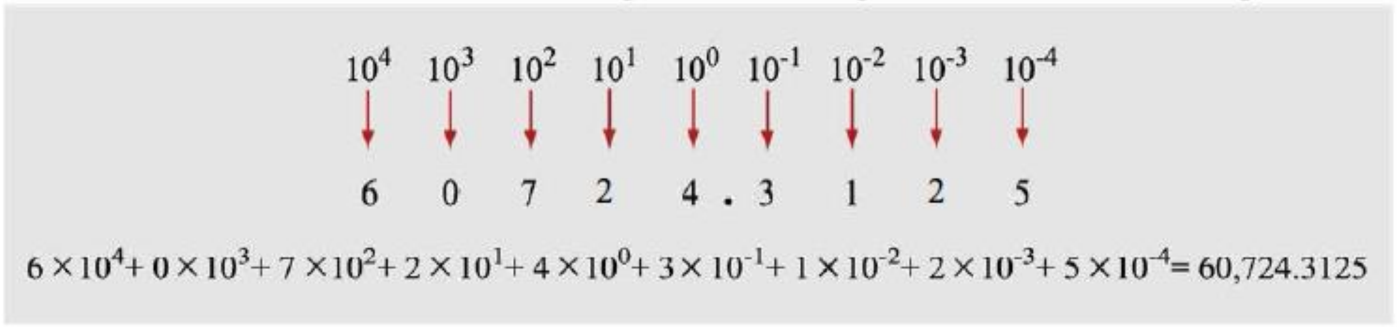
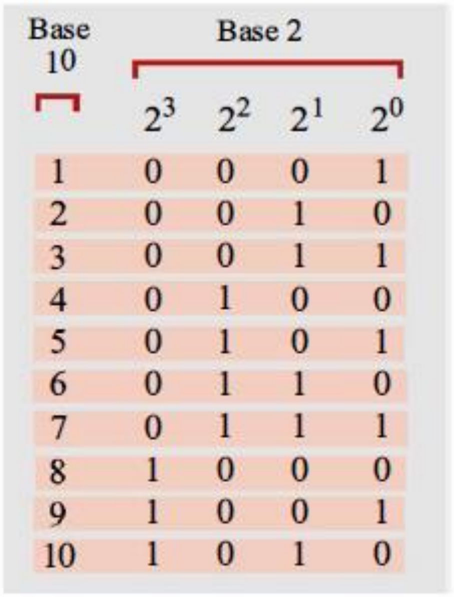

10^4 10^3 10^2 10^1 10^0 10^{-1} 10^{-2} 10^{-3} 10^{-4}
6 0 7 2 4 . 3 1 2 5
$6\times10^4+0\times10^3+7\times10^2+2\times10^1+4\times10^0+3\times10^{-1}+1\times10^{-2}+2\times10^{-3}+5\times10^{-4}=60{,}724.3125$

Fig. 1-1: Representation of the number 60,724.3125 in the decimal system (base 10).

In general, however, a number can be represented using other bases. A form that can be easily implemented in computers is the binary (base 2) system. In the binary system, a number is represented by using the two digits 0 and 1. A number is then written as a sequence of zeros and ones that correspond to multiples of powers of 2. The first digit to the left of the decimal point corresponds to 2^0. The digit next to it on the left corresponds to 2^1, the next digit to the left to 2^2, and so on. In the same way, the first digit to the right of the decimal point corresponds to r^1, the next digit to the right to r^2, and so on. The first ten digits 1, 2, 3, . . . , 10 in base 10 and their representation in base 2 are shown in Fig. 1-2. The representation of the number 19.625 in the binary system is shown in Fig. 1-3.



Base 10	Base 2			
	2^3	2^2	2^1	2^0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0

Figure 1-2: Representation of numbers in decimal and binary forms.

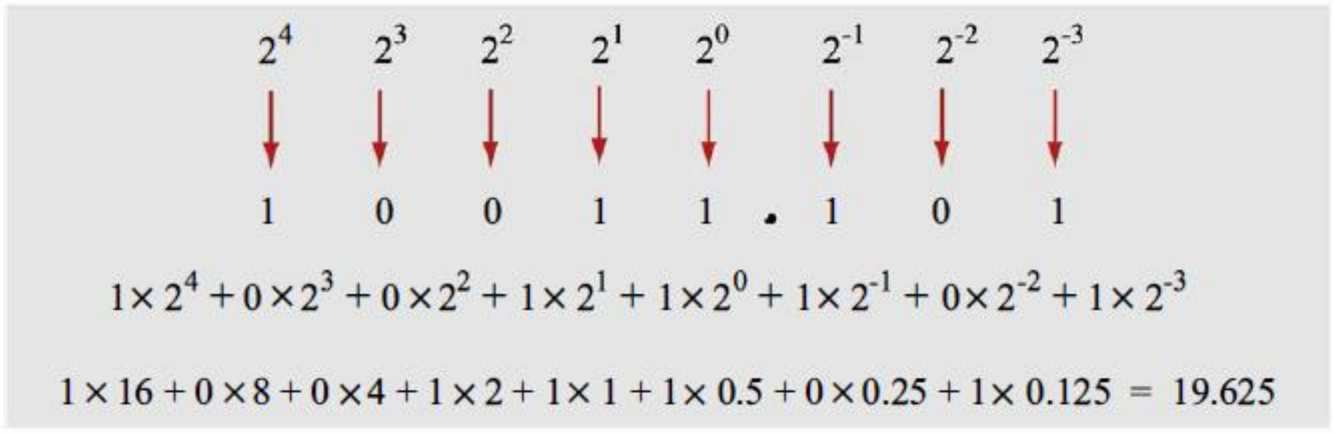

2^4 2^3 2^2 2^1 2^0 2^{-1} 2^{-2} 2^{-3}
1 0 0 1 1 . 1 0 1
$1\times 2^4+0\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0+1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}$
$1\times 16+0\times 8+0\times 4+1\times 2+1\times 1+1\times 0.5+0\times 0.25+1\times 0.125 = 19.625$

Figure 1-3: Representation of the number 19.625 in the binary system (base 2).

Another example is shown in Fig. 1-4, where the number 60,724.3125 is written in binary form.

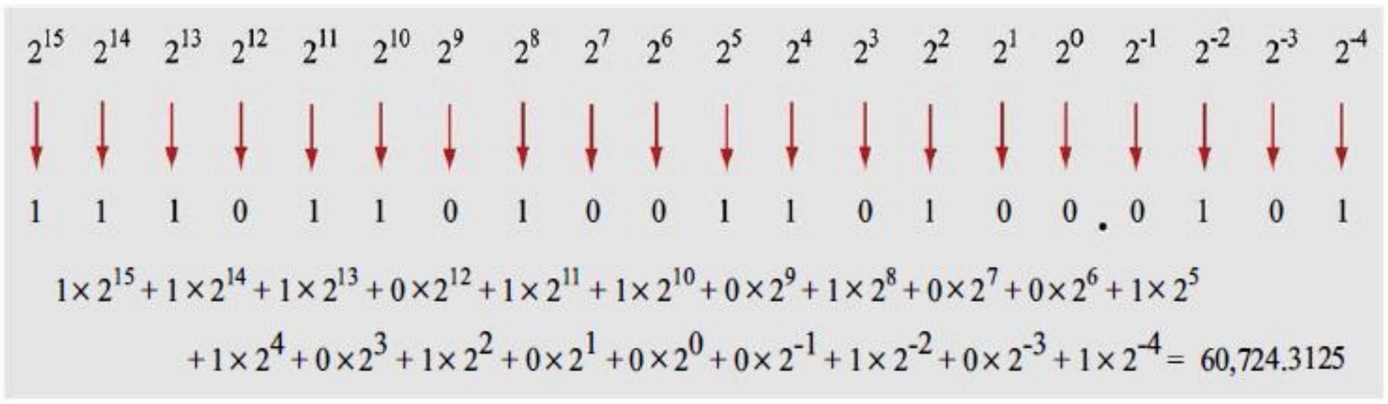

2^{15} 2^{14} 2^{13} 2^{12} 2^{11} 2^{10} 2^9 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0 2^{-1} 2^{-2} 2^{-3} 2^{-4}
1 1 1 0 1 1 0 1 0 0 1 1 0 1 0 0 . 0 1 0 1
$1\times 2^{15}+1\times 2^{14}+1\times 2^{13}+0\times 2^{12}+1\times 2^{11}+1\times 2^{10}+0\times 2^9+1\times 2^8+0\times 2^7+0\times 2^6+1\times 2^5$
$+1\times 2^4+0\times 2^3+1\times 2^2+0\times 2^1+0\times 2^0+0\times 2^{-1}+1\times 2^{-2}+0\times 2^{-3}+1\times 2^{-4}=$ 60,724.3125

Figure 1-4: Representation of the number 60,724.3125 in the binary system (base 2).

Computers store and process numbers in binary (base 2) form:

Each binary digit (one or zero) is called a bit (for binary digit). Binary arithmetic is used by computers because modem transistors can be used as extremely fast switches. Therefore, a network of these may be used to represent strings of numbers with the "1" referring to the switch being in the "on" position and "O" referring to the "off' position. Various operations are then performed on these sequences of ones and zeros.

1.1.2 Floating point representation

To accommodate large and small numbers, real numbers are written in floating point representation. Decimal floating point representation (also called scientific notation) has the form:

$$d.ddddd \times l0^P \qquad (1.\ 1)$$

One digit is written to the left of the decimal point, and the rest of the significant digits are written to the right of the decimal point. The number d.dddddd is called the mantissa. Two examples are:

6519.23 written as 6.51923×10^3

0.00000391 written as 3.91×10^{-6}

The power of 10, p, represents the number's order of magnitude, provided the preceding number is smaller than 5. Otherwise, the number is said to be of the order of p + 1. Thus, the number 3.91×10^{-6} is of the order of 10^{-6}, $O(10^{-6})$, and the number 6.51923×10^3 is of the order of 10^4(written as $O(10^4)$). Binary floating point representation has the form:

$$1.\ bbbbbb \times 2^{bbb} \quad \text{(b is a decimal digit)} \qquad (1.\ 2)$$

In this form, the mantissa is . bbbbbb , and the power of 2 is called the exponent. Both the mantissa and the exponent are written in a binary form. The form in Eq. (1. 2) is obtained by normalizing the number (when it is written in the decimal form) with respect to the largest power of 2 that is smaller than the number itself. For example, to write the number 50 in binary floating point representation, the number is divided (and multiplied) by $2^5 = 32$ (which is the largest power of 2 that is smaller than 50):

$$50=\frac{50}{2^5}\times 2^5=1.5625\times 2^5 \quad \text{Binary floating point form: } 1.1001\times 2^{101}$$

Two more examples are:

$$1344=\frac{1344}{2^{10}}\times 2^{10}=1.3125\times 2^{10} \quad \text{Binary floating point form: } 1.0101\times 2^{1010}$$

$$0.3125=\frac{0.3125}{2^{-2}}\times 2^{-2}=1.25\times 2^{-2} \quad \text{Binary floating point form: } 1.01\times 2^{-10}$$

1.2 ERRORS IN NUMERICAL SOLUTIONS

Numerical solutions can be very accurate but in general are not exact. Two kinds of errors are introduced when numerical methods are used for solving a problem. One kind, which was mentioned in the previous section, occurs because of the way that digital computers store numbers and execute numerical operations. These errors are labeled round-off errors. The second kind of errors is introduced by the numerical method that is used for the solution. These errors are labeled truncation errors. Numerical methods use approximations for solving problems. The errors introduced by the approximations are the truncation errors. Together, the two errors constitute the total error of the numerical solution, which is the difference (can be defined in various ways) between the true (exact) solution (which is usually unknown) and the approximate numerical solution. Round-off, truncation, and total errors are discussed in the following three subsections.

1.2.1 Round-Off Errors

Numbers are represented on a computer by a finite number of bits . Consequently, real numbers that have a mantissa longer than the number of bits that are available for representing them have to be shortened. This requirement applies to irrational numbers that have to be represented in a finite form in any system, to finite numbers that are too long, and to finite numbers in decimal form that cannot be represented exactly in binary form. A number can be shortened either by chopping off, or discarding, the extra digits or by rounding. In chopping, the digits in the mantissa beyond the length that can be stored are simply left out. In rounding, the last digit that is stored is rounded.

As a simple illustration, consider the number 2/3. (For simplicity, decimal format is used in the illustration. In the computer, chopping and rounding are done in the binary format.) In decimal form with four significant digits, 2/3 can be written as 0.6666 or as 0.6667. In the former instance, the actual number has been chopped off, whereas in the latter instance, the actual number has been rounded. Either way, such chopping and rounding of real numbers lead to errors in numerical computations, especially when many operations are performed. This type of numerical error (regardless of whether it is due to chopping or rounding) is known as round-off error. Example 1-1 shows the difference between chopping and rounding.

Example 1-1: Round-off errors

Consider the two nearly equal numbers p = 9890.9 and q = 9887. 1 . Use decimal floating point representation (scientific notation) with three significant digits in the mantissa to calculate the difference between the two numbers, (p - q) . Do the calculation first by using chopping and then by using rounding.

SOLUTION

In decimal floating point representation, the two numbers are:

$$p = 9.8909 \times 10^3 \text{ and } q = 9.8871 \times 10^3$$

If only three significant digits are allowed in the mantissa, the numbers have to be shortened. If chopping is used, the numbers become:

$$p = 9.890 \times 10^3 \text{ and } q = 9.887 \times 10^3$$

Using these values in the subtraction gives:

$$p - q = 9.890 \times 10^3 - 9.887 \times 10^3 = 0.003 \times 10^3 = 3$$

If rounding is used, the numbers become:

$$p = 9.891 \times 10^3 \text{ and } q = 9.887 \times 10^3 \text{ (q is the same as before)}$$

Using these values in the subtraction gives:

$$p - q = 9.891 \times 10^3 - 9.887 \times 10^3 = 0.004 \times 10^3 = 4$$

The true (exact) difference between the numbers is 3.8. These results show that, in the present problem, rounding gives a value closer to the true answer.

The magnitude of round-off errors depends on the magnitude of the numbers that are involved since, as explained in the previous section, the interval between the numbers that can be represented on a computer depends on their magnitude. Round-off errors are likely to occur when the numbers that are involved in the calculations differ significantly in their magnitude and when two numbers that are nearly identical are subtracted from each other.

For example, consider the quadratic equation:

$$x^2 - 100.000lx + 0.01 = 0 \qquad (1.3)$$

for which the exact solutions are $x_1 = 100$ and $x_2 = 0.0001$. The solutions can be calculated with the quadratic formula:

$$x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a} \text{ and } x_2=\frac{-b-\sqrt{b^2-4ac}}{2a} \qquad (1.4)$$

Using MATLAB (Command Window) to calculate x_1 and x_2 gives:

```
>> format long
>> a = 1; b = -100.0001; c = 0.01;
>> root = sqrt(b^2 - 4*a*c)
root =
        99.999899999999997
>> x1 = (-b + root)/(2*a)
xl =
      100
>> x2 = (-b - root)/(2*a)
x2 =
     1.000000000033197e-004
```

The value that is calculated by MATLAB for x_2 is not exact due to round-off errors. The round-off error occurs in the numerator in the expression for x_2 • Since b is negative, the numerator involves subtraction of two numbers that are nearly equal.

Another example of round-off errors is shown in Example 1-2.

Example 1-2: Round-off errors

Consider the function:

$$f(x) = x(\sqrt{x} - \sqrt{x-1}) \qquad (1.5)$$

(a) Use MATLAB to calculate the value of f(x) for the following three values of x:

$$x = 10,\ x = 1000\ ,\ \text{and } x = 100000\ .$$

(b) Use the decimal format with six significant digits to calculate f(x) for the values of x in part (a). Compare the results with the values in part (a).

SOLUTION

(a)

```
>> format long g
>> x = [10 1000 100000] ;
>> Fx = x.*(sqrt(x) - sqrt(x-1))
Fx =
    1.6227766016838 15.8153431255776 158.114278298171
```

(b) Using decimal format with six significant digits in Eq. (1.5) gives the following values for f(x):

$$f(10) = 10\left(\sqrt{10} - \sqrt{10-1}\right) = 10(3.16228\ - 3)\ =\ 1.62280$$

This value agrees with the value from part (a), when the latter is rounded to six significant digits.

$f(1000) = 1000\left(\sqrt{1000} - \sqrt{1000-1}\right)$ = 1000(31.6228-31.6070) = 15.8

When rounded to six significant digits, the value in part (a) is 15.8153.

$f(100000) = 100000\left(\sqrt{100000} - \sqrt{100000-1}\right)$= 100000(316.228-316.226) = 200

When rounded to six significant digits, the value in part (a) is 158.114.

The results show that the rounding error due to the use of six significant digits increases as x increases and the relative difference between $\sqrt{x}$ and $\sqrt{x-1}$decreases.

1.2.2 Truncation Errors

Truncation errors occur when the numerical methods used for solving a mathematical problem use an approximate mathematical procedure. A simple example is the numerical evaluation of sin(x), which can be done by using Taylor's series expansion :

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \qquad (1.6)$$

The value of sin $\left(\frac{\pi}{6}\right)$ can be determined exactly with Eq. (1.6) if an infinite number of terms are used. The value can be approximated by using only a finite number of terms. The difference between the true (exact) value and an approximate value is the truncation error, denoted by E^{TR} . For example, if only the first term is used:

sin $\left(\frac{\pi}{6}\right) = \frac{\pi}{6}$ = 0.5235988 , E^{TR} = 0.5 - 0.5235988 = -0.0235988

If two terms of the Taylor's series are used:

sin $\left(\frac{\pi}{6}\right) = \frac{\pi}{6} - \frac{\frac{\pi^3}{6}}{3!}$ = 0.4996742 , E^{TR} = 0.5 - 0.4996742 = 0.0003258

Another example of truncation error that is probably familiar to the reader is the approximate calculation of derivatives. The value of the derivative of a function f(x) at a point x_1 can be approximated by the expression:

$$\frac{df(x)}{dx}|x = x_1 = \frac{f(x_2)-f(x_1)}{x_2-x_1} \qquad (1.7)$$

where x_2 is a point near x_1 • The difference between the value of the true derivative and the value that is calculated with Eq. (1.7) is called a truncation error. The truncation error is dependent on the specific numerical method or algorithm used to solve a problem. Details on truncation errors are discussed as various numerical methods are presented. The truncation error is independent of round-off error; it exists even when the mathematical operations themselves are exact.

1.3 Absolute and Relative Errors

If X_E is the exact or true value of a quantity and X_A is its approximate value, then $|X_E - X_A|$ is called the ***absolute error*** E_a. Therefore absolute error:

$$E_a = |X_E - X_A| \qquad (1.8)$$

and ***relative error*** is defined by:

$$E_r = \left|\frac{X_E - X_A}{X_E}\right| \qquad (1.9)$$

provided $X_E \neq 0$ or X_E is not too close to zero. The ***percentage relative error*** is:

$$E_p = 100E_r = 100\left|\frac{X_E - X_A}{X_E}\right| \qquad (1.10)$$

Significant digits: The concept of a significant figure, or digit, has been developed to formally define the reliability of a numerical value. The *significant digits* of a number are those that can be used with confidence.

If X_E is the exact or true value and X_A is an approximation to X_E, then X_A is said to approximate X_E to t significant digits if t is the largest non-negative integer for which:

$$\left|\frac{X_E - X_A}{X_E}\right| < 5 \times 10^{-t} \qquad (1.11)$$

Example 1-3:

If $X_E = e$ (base of the natural algorithm = 2.7182818) is approximated by $X_A = 2.71828$, what is the significant number of digits to which X_A approximates X_E?

Solution:

$$\left|\frac{X_E - X_A}{X_E}\right| = \frac{e - 2.71828}{e} \; which\ is < 5 \times 10^{-6}$$

Hence X_A approximates X_E to 6 significant digits.

Example 1-4:

Let the exact or true value = 20/3 and the approximate value = 6.666.

Solution:

The absolute error is 0.000666... = 2/3000.

The relative error is (2/3000)/ (20/3) = 1/10000.

The number of significant digits is 4.

Example 1-5:

Given the number $\pi$is approximated using n = 4 decimal digits.

(a) Determine the relative error due to chopping and express it as a per cent.

(b) Determine the relative error due to rounding and express it as a per cent.

Solution:

(a) The relative error due to chopping is given by

$$E_r(\text{chopping}) = \frac{3.1415 - \pi}{\pi} = 2.949 \times 10^{-5} \text{ or } 0.002949\%$$

(b) The relative error due to rounding is given by

$$E_r(\text{rounding}) = \frac{3.1416 - \pi}{\pi} = 2.338 \times 10^{-6} \text{ or } 0.0002338\%.$$

Example 1-6:

If the number $\pi = 4\tan^{-1}(1)$ is approximated using 4 decimal digits, find the percentage relative error due to, (a) chopping (b) rounding.

Solution:

(a) Percentage relative error due to chopping

$$= \left(\frac{3.1415 - \pi}{\pi}\right)100 = \left(-2.949 \times 10^{-5}\right)100 \text{ or } -0.002949\%.$$

(b) Percentage relative error due to rounding

$$= \left(\frac{3.1416 - \pi}{\pi}\right)100 = \left(2.338 \times 10^{-6}\right)100 = 0.00023389\%$$

1.3 PROBLEMS

Problems to be solved by hand

Solve the following problems by hand. When needed, use a calculator or write a MATLAB script file to carry out the calculations.

1. Convert the binary number 1010100 to decimal format.

2. Consider the function $f(x) = \frac{1-\cos x}{\sin x}$.

a) Use the decimal format with six significant digits (apply rounding at each step) to calculate (using a calculator) f(x) for x = 0.007.

b) Use MATLAB (format long) to calculate the value of f(x). Consider this to be the true value, and calculate the true relative error, due to rounding, in the value of f(x) that was obtained in part (a).

3. Consider the function $f(x) = \frac{\sqrt{4+x}-2}{x}$.

a) Use the decimal format with six significant digits (apply rounding at each step) to calculate (using a calculator) f(x) for x = 0.001.

b) Use MATLAB (format long) to calculate the value of f(x). Consider this to be the true value, and calculate the true relative error, due to rounding, in the value of f(x) that was obtained in part (a).