Significant Digits

We know that all computers operate with a fixed length of numbers. In particular, we have seen that the floating-point representation requires the mantissa to be of a specified number of digits. Some numbers cannot be represented exactly within a given number of decimal digits. For example, the number π is equal to π = 3.14159265358979323846……

Such numbers can never be represented accurately. We may write it as 3.14, 3.14159, or 3.141592653. In all cases, some digits have been omitted.

Note that transcendental and irrational numbers do not have a terminating representation. Some rational numbers also have repeating decimal patterns. For instance, the rational number 2/7 = 0.285714285714…….. Suppose we write 2/7 as 0.285714 and π as 3.14159. Then, we say the numbers contain six significant digits.

The concept of significant digits has been introduced primarily to indicate the accuracy of a numerical value. For example, if, in the number ($y = 23.40657$), only the digits 23406 are correct, then we may say that (y) has five significant digits and is correct to only *three* decimal places.

In general, when a number is said to be "good to four digits," it means that the number has four significant digits. The omission of certain digits from a number results in what is called **roundoff error**. The following statements describe the notion of significant digits.

Rules for Determining Significant Digits

1. **All non-zero digits are significant.**
   Example: 2, 345, and 7.89 each contain only significant digits.
2. **All zeros between non-zero digits are significant.**
   Example: In 203 and 50.07, all digits are significant.
3. **All zeros to the right of a decimal point and following a non-zero digit are significant.**
   Example: 2.250, 65.0, and 0.230 all have **three significant digits**.
4. **Zeros to the left of the first non-zero digit and following a decimal point are not significant.**
   Example: In 0.0023, only the digits 2 and 3 are significant, so the number has **two significant digits**.
5. **When the decimal point is not written, trailing zeros are not considered significant**.
   For example, the number **4500** may be written as 45×10^2 and contains **only two significant digits**.
   However, **4500.0** contains **four significant digits**, as the presence of the decimal point indicates greater precision.

Further Examples:

- 7.56 has **three significant digits**.
- 7.560×10^4 has **four significant digits**.

- 7.5600×10^4 has **five significant digits**.

Note: Integer numbers with trailing zeros should be written in **scientific notation** to clearly indicate the number of significant digits.

Accuracy and Precision in Relation to Significant Digits

The concepts of **accuracy** and **precision** are closely related to significant digits:

1. **Accuracy** refers to the **number of significant digits** in a value.
   For example, the number **57.396** is accurate to **five significant digits**.
2. **Precision** refers to the **number of decimal places**, i.e., the **order of magnitude** of the last digit.
   The number **57.396** has a precision of **0.001** or 10^{-3}.

Characteristics of Numerical Methods

Numerical methods exhibit certain computational characteristics during implementation. It is important to consider these characteristics while choosing a particular method for implementation. The characteristics that are critical to the success of implementation are: **accuracy**, **rate of convergence**, **numerical stability**, and **efficiency**.

1. Accuracy

Every method of numerical computing introduces errors. These may be either due to using an approximation in place of an exact mathematical procedure (called **truncation errors**) or due to the inexact representation and manipulation of numbers in the computer (called **round-off errors**). These errors affect the accuracy of the results. The results we obtain must be sufficiently accurate to serve the purpose for which the mathematical model was built. Choice of a method is, therefore, very much dependent on the particular problem.

Example:

- Approximating the derivative of *f(x)* using finite difference:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

introduces truncation errors because it does not use the exact derivative formula.

- Using floating-point arithmetic in a computer to represent numbers like π causes round-off errors.

2. Rate of Convergence

Many numerical methods are based on the idea of an iterative process. This process involves generation of a sequence of approximations with the hope that the process will converge to the required solution. Certain methods converge faster than others. Some methods may not converge at all. It is, therefore, important to test for convergence before a method is used. Rapid convergence takes less execution time on the computer. There are several techniques for accelerating the rate of convergence of certain methods.

Example:

- The **Newton-Raphson method** converges faster than the **bisection method** when solving a non-linear equation such as $f(x)=0$, provided the initial guess is close to the root.
  - Bisection method: Linear convergence
  - Newton-Raphson method: Quadratic convergence

3. Numerical Stability

Another problem introduced by some numerical computing methods is that of numerical instability. Errors introduced into a computation, from whatever source, propagate in different ways. In some cases, these errors tend to grow exponentially, with disastrous computational results. A computing process that exhibits such exponential error growth is said to be **numerically unstable**. We must choose methods that are not only fast but also stable.

Numerical instability may also arise due to **ill-conditioned problems**. There are many problems which are inherently sensitive to round-off errors and other uncertainties. Thus, we must distinguish between sensitivity of methods and sensitivity inherent in problems. When the problem is ill-conditioned, there is nothing we can do to make a method numerically stable.

Example:

- **Solving a system of linear equations** using Gaussian elimination without pivoting can be unstable if the matrix is nearly singular, as small errors can lead to large deviations in results.
- Approximating e^{-x} for large x using a truncated series is unstable due to loss of significance (round-off error accumulation).

4. Efficiency

One more consideration in choosing a numerical method for solution of a mathematical model is **efficiency**. It refers to the amount of effort required by both human and computer to implement the method. A method that requires less computing time and less programming effort, while still achieving the desired accuracy, is always preferred.

Example:

- The **Gauss-Seidel method** for solving large systems of linear equations is more efficient than direct matrix inversion for sparse matrices, as it requires less memory and computational effort.
- In numerical integration, **Simpson's Rule** is more efficient than the **trapezoidal rule** for smooth functions because it achieves higher accuracy with fewer subintervals.