



UNIVERSITI TEKNOLOGI MALAYSIA

SME 3023 Applied Numerical Methods

Engineering Problem Solving

Abu Hasan Abdullah

Faculty of Mechanical Engineering

Sept 2012



1 Engineering Problem

Outline

- 1 Engineering Problem
- 2 Analysis of Engineering Problem
 - Problem Statement
 - Mathematical Model
 - Solution
 - Verification
 - Examples

- 1 Engineering Problem
- 2 Analysis of Engineering Problem
 - Problem Statement
 - Mathematical Model
 - Solution
 - Verification
 - Examples
- 3 Accuracy and Precision

Outline

- 1 Engineering Problem
- 2 Analysis of Engineering Problem
 - Problem Statement
 - Mathematical Model
 - Solution
 - Verification
 - Examples
- 3 Accuracy and Precision
- 4 Error
 - Absolute & Relative Errors
 - Absence of True Value
 - Sources

Outline

- 1 Engineering Problem
- 2 Analysis of Engineering Problem
 - Problem Statement
 - Mathematical Model
 - Solution
 - Verification
 - Examples
- 3 Accuracy and Precision
- 4 Error
 - Absolute & Relative Errors
 - Absence of True Value
 - Sources
- 5 Propagation of Error
 - In Arithmetic Operations
 - Examples

Outline

- 1 Engineering Problem
- 2 Analysis of Engineering Problem
 - Problem Statement
 - Mathematical Model
 - Solution
 - Verification
 - Examples
- 3 Accuracy and Precision
- 4 Error
 - Absolute & Relative Errors
 - Absence of True Value
 - Sources
- 5 Propagation of Error
 - In Arithmetic Operations
 - Examples
- 6 Bibliography

Picture of the Problem

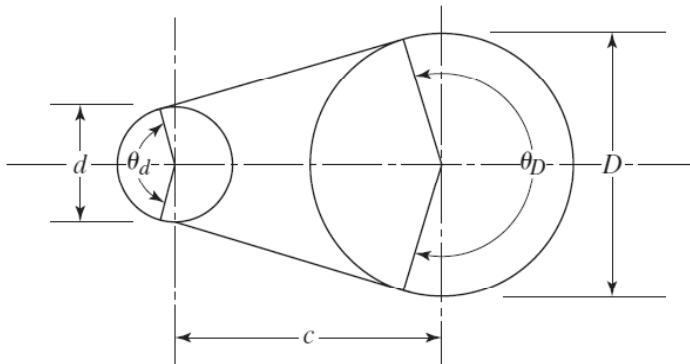


Figure 1 : Open belt drive.

Statement of the Problem

The length L of a belt in an open-belt drive, Figure 1, is given by

$$L = \sqrt{4c^2 - (D - d)^2} + \frac{1}{2}(D\theta_D + d\theta_d) \quad (1)$$

where

$$\theta_D = \pi + 2 \sin^{-1} \left(\frac{D - d}{2c} \right) \qquad \theta_d = \pi - 2 \sin^{-1} \left(\frac{D - d}{2c} \right)$$

c is the centre distance, D is the diameter of the larger pulley, d is the diameter of the smaller pulley, θ_D is the angle of contact of the belt with the larger pulley, and θ_d is the angle of contact of the belt with the smaller pulley.

If a belt having a length 11 m is used to connect the two pulleys with diameters 0.4 m and 0.2 m, determine the centre distance between the pulleys.

Analysis of Engineering Problem

- ① **Problem Statement:** Recognise and understand the problem (what is it that needed to be solved?).
- ② **Governing Equations or Mathematical Models:** Identify parameters affecting the problem, make the necessary assumptions, develop **mathematical model** or **governing equations** (based on theories from Engineering Mathematics and other Engineering Subjects).
- ③ **Solution:** Solution of the governing equations may make use of the **computer programming** (why?).
- ④ **Verification:** Verify and interpret the solution (right/wrong?).

Analysis of Engineering Problem

Problem Statement

The length of a belt in an open-belt drive, L , is given by

$$L = \sqrt{4c^2 - (D - d)^2} + \frac{1}{2}(D\theta_D + d\theta_d) \quad (2)$$

where

$$\theta_D = \pi + 2 \sin^{-1} \left(\frac{D - d}{2c} \right) \quad \theta_d = \pi - 2 \sin^{-1} \left(\frac{D - d}{2c} \right)$$

c is the centre distance, D is the diameter of the larger pulley, d is the diameter of the smaller pulley, θ_D is the angle of contact of the belt with the larger pulley, and θ_d is the angle of contact of the belt with the smaller pulley, see Figure-2.8 of Rao (2002).

If a belt having a length 11 m is used to connect the two pulleys with diameters 0.4 m and 0.2 m, determine the centre distance between the pulleys.

Analysis of Engineering Problem

Mathematical Model

- Defined as a **formulation** or equation that expresses the essential features of a physical system or process in mathematical terms.
- Its simplest form can be represented as a **functional relationship** thus

$$\text{Dependent variable} = f(\text{independent variables, parameters, forcing functions})$$

where

- dependent variable:** a characteristic that reflects the behaviour/state of system
- independent variables:** dimensions (time, space, mass) along which the system's behaviour that is being determined
- parameters:** reflective of system's properties or composition
- forcing functions:** external influences acting on the system
- Mathematical model ranges from a simple algebraic relationship to large complicated set of DE. Mathematical models (a.k.a. **governing equations**) are derived by applying physical laws such as
 - Equilibrium Equation
 - Newton's Law of Motion
 - Conservation Laws: Mass, Momentum, Energy
 - Equation of State

Analysis of Engineering Problem

Mathematical Model

- Specific to our open belt drive problem in Figure 1,

Mathematical Model

$$L = \sqrt{4c^2 - (D - d)^2} + \frac{1}{2}(D\theta_D + d\theta_d)$$

where

$$\theta_D = \pi + 2 \sin^{-1} \left(\frac{D - d}{2c} \right)$$

$$\theta_d = \pi - 2 \sin^{-1} \left(\frac{D - d}{2c} \right)$$

which is a well known relationship, readily derived for us.

- In the majority of engineering problems, the engineer might have to derive the mathematical model from the first principles.

Analysis of Engineering Problem

Solution

- Solution of the governing equation or mathematical model may appear as
 - Transcendental Functions
 - Linear or Nonlinear Algebraic Equations
 - Homogeneous Equations leading to an Eigenvalue Problem
 - Ordinary or Partial Differential Equations
 - Equations involving Integrals or Derivatives

which are either **closed-form** or **open-ended**.

- **Closed-form** mathematical expression, e.g.

$$I_1 = \int_a^b x e^{-x^2} dx = \left[-\frac{1}{2} e^{-x^2} \right]_a^b = -\frac{1}{2} e^{-b^2} + \frac{1}{2} e^{-a^2} = \frac{1}{2} (e^{-a^2} - e^{-b^2})$$

leads to *analytical solution*

- **Open-ended** mathematical expressions, e.g.

$$I_1 = \int_a^b f(x) dx = \int_a^b e^{-x^2} dx$$

need to be *approximated numerically*

Analysis of Engineering Problem

Solution: Computer Program

- Nowadays, approximated numerical solutions are done by developing a **computer program**.
- Because numerical methods deal extensively with approximations connected with the manipulation of numbers, **accuracy**, **precision** and **error** feature prominently in programming the solution. We shall cover these later!
- Steps in computer program development:
 - **Algorithm Design**: Listing down of the sequence of steps to define the problem at hand. Techniques available: **algorithm**, **flowchart**, **pseudocode**
 - **Program Coding**: Writing these steps in a computer language.
 - **Debugging**: Testing the program to ensure that it is error-free and reliable.
 - **Documentation**: Making the program easy to understand and use through manual or guide.

Note:

See **SKMM 1013 Programming for Engineers** for details.

Analysis of Engineering Problem

Solution: Computer Program

- **Algorithm:** A general sequence of the logical steps in solving a specific problem.
- **Flowchart:** A graphical representation of the algorithm. Better suited for visualizing complex algorithms.
- **Pseudocode:** Uses **code-like** statements in place of the graphical symbols of flowchart. Easier to develop a program with it than with a flowchart.
- Elements of good algorithm
 - Each step must be **deterministic** i.e. not ambiguous.
 - The process must end after a **finite** number of steps.
 - The algorithm must be **general** enough to deal with any contingency.



Analysis of Engineering Problem

Solution: Computer Program–Flowchart

SYMBOL	NAME	Name	Function
	Terminal	Terminal	Represents the beginning or end of a program.
	Flowlines	Flowlines	Represents the flow of logic. The humps on the horizontal arrow indicate that it passes over and does not connect with the vertical flowlines.
	Process	Process	Represents calculations or data manipulations.
	Input/output	Input/Output	Represents inputs or outputs of data and information.
	Decision	Decision	Represents a comparison, question, or decision that determines alternative paths to be followed.
	Junction	Junction	Represents the confluence of flowlines.
	Off-page connector	Off-page Connector	Represents a break that is continued on another page.
	Count-controlled loop	Count-controlled loop	Used for loops which repeat a pre-specified number iterations.

Figure 2 : Some of the symbols used in flowcharting.

Analysis of Engineering Problem

Solution: Computer Program–Algorithm & Pseudocode

Problem Statement:

Find roots of equation $ax^2 + bx + c = 0$ using the quadratic formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Before the actual program is written, we need to outline an **algorithm** and/or **pseudocode** for solving this problem:

Algorithm

- 1 Start
- 2 Read coefficients a , b and c
- 3 Implement quadratic formula. Avoid division by zero, allow for complex roots.
- 4 Display solution i.e. values of x
- 5 Stop

Pseudocode

```
DO
  READ a, b, c
  root1 = (-b + SQRT(b^2 - 4ac))/(2a)
  root2 = (-b - SQRT(b^2 - 4ac))/(2a)
  PRINT root1, root2
  PRINT 'Try again? Answer yes or no'
  READ response
  IF response = 'no' EXIT
ENDDO
```

Analysis of Engineering Problem

Solution: Computer Program–Coding

- A program is a sequence of instructions to the computer for it to solve a particular problem. A set of programs is called code.
- Programs are written in some programming language, e.g. C/C++, Fortran, Matlab, Basic, Pascal, Java.
- Programs are stored in files which are a sequence of bytes which is given a name and stored on a disk.
- A program is a file containing a sequence of statements, each of which tells the computer to do a specific action.
- Once a program is run or executed the commands are followed and actions occur in a sequential manner.
- If the program is designed to interact with the outside world, then it must have input and output.
- A program is said to have a bug if it contains a mistake or it does not function in the way it is intended to.
- Bugs can happen both in the logic of the program, and in the commands.

Analysis of Engineering Problem

Verification

The final step of any engineering analysis should be the verification of results.

- Various sources of error can contribute to wrong results. Common sources of error include:
 - misunderstanding a given problem,
 - making incorrect assumptions to simplify the problem,
 - applying a physical law that does not truly fit the given problem, and
 - incorporating inappropriate physical properties
- Before you present your solution or the results to your instructor or, later in your career, to your manager, you need to learn to think about the calculated results. You need to ask yourself the following question:
 - Do the results make sense?

A good engineer must always find ways to check results.

- Ask yourself this additional question:
 - What if I change one of the given parameters. How would that change the result?
- Then consider if the outcome seems reasonable.

Analysis of Engineering Problem

Verification

- If you formulate the problem such that the final result is left in parametric (symbolic) form, then you can experiment by substituting different values for various parameters and look at the final result.
- In some engineering work, actual physical experiments must be carried out to verify one's findings.
- Starting today, get into the habit of asking yourself if your solution to a problem makes sense.
- Asking your instructor if you have come up with the right answer or checking the back of your textbook to match answers are not good approaches in the long run. You need to develop the means to check your results by asking yourself the appropriate questions.
- Remember, once you start working for hire, there are no answer books. You will not want to run to your boss to ask if you did the problem right!



Analysis of Engineering Problem

Example Problem 1

Problem Statement:

Assuming that the thrust T of a screw propeller is dependent upon diameter D , speed of advance v , fluid density ρ , rotational speed of propeller N and coefficient of viscosity μ , derive an expression that relates all the parameters involved and solve for T .

Mathematical Model:

Through *dimensional analysis*

$$T = \rho v^2 D^2 f\left(\frac{\mu}{\rho v D}, \frac{ND}{v}\right)$$

Analysis of Engineering Problem

Example Problem 2

Problem Statement:

Given temperature in degrees Fahrenheit, the temperature in degrees Kelvin is to be computed and shown.

Mathematical Model:

From Physics, these two temperature scales are related through

$$T_k = \left(\frac{T_F - 32}{1.8} \right) + 273.15$$

and the parameters involved in this problem are T_K and T_F

Analysis of Engineering Problem

Example Problem 2

Algorithm

- 1 Start
- 2 Get the temperature in Fahrenheit, T_F
- 3 Compute the temperature in Kelvin using the formula:

$$T_k = \left(\frac{T_F - 32}{1.8} \right) + 273.15$$

- 4 Show the temperature in Kelvin, T_k
- 5 Stop

Pseudocode

```
Start
Read TF
TK = (TF-32)/1.8 + 273.15
Print TK
Stop
```

Analysis of Engineering Problem

Example Problem 3

Problem Statement:

Determine the mass of the bungee jumper with a drag coefficient of 0.25 kg/m to have a velocity of 36 m/s after 4 s of free fall.

Note: The acceleration due to gravity is 9.81 m/s.

Mathematical Model:

From Physics and Mechanics the fall velocity as function of time is

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh \left(\sqrt{\frac{gc_d}{m}} t \right)$$

or, on re-arranging in terms of a function of mass, $f(m)$,

$$f(m) = \sqrt{\frac{gm}{c_d}} \tanh \left(\sqrt{\frac{gc_d}{m}} t \right) - v(t) = 0$$

Analysis of Engineering Problem

Example Problem 3

Solution:

The governing equation indicates a root-finding problem.

Algorithm

- 1 Choose lower m_L and upper m_U guesses for the root such that the function changes sign over the interval. This can be checked by ensuring that $f(m_L) \times f(m_U) < 0$.

- 2 An estimate of the root m_R is determined by

$$m_R = \frac{m_L + m_U}{2}$$

- 3 Make the following evaluations to determine in which subinterval the root lies:
 - If $f(m_L) \times f(m_U) < 0$, root lies in lower subinterval; repeat step 2.
 - If $f(m_L) \times f(m_U) > 0$, root lies in upper subinterval; repeat step 2.
 - If $f(m_L) \times f(m_U) = 0$, root equals m_R ; terminate computation.

Analysis of Engineering Problem

Example Problem 3

Program Source Code

```
C*****
C          BISECTION METHOD          *
C*****
C    TO FIND A SOLUTION TO F(X)=0 GIVEN THE CONTINUOUS FUNCTION
C    F ON THE INTERVAL <A,B>, WHERE F(A) AND F(B) HAVE
C    OPPOSITE SIGNS:
C
C    INPUT:  ENDPOINTS A,B; TOLERANCE TOL;
C            MAXIMUM INTERACTIONS NO.
C    OUTPUT: APPROXIMATE SOLUTION P OR A
C            MESSAGE THAT THE ALGORITHM FAILS.
C
C    CHARACTER NAME*14,NAME1*14,AA*1
C    INTEGER INP,OUP,FLAG
C    LOGICAL OK
C    REAL A,B,FA,FB,X,TOL
C    INTEGER NO
C*****
C          DEFINE YOUR FUNCTION HERE          *
C*****
C    DEFINE F
C!    F(X)=X**3+4*X**2-10
C    OPEN(UNIT=5,FILE='CON',ACCESS='SEQUENTIAL')
C    OPEN(UNIT=6,FILE='CON',ACCESS='SEQUENTIAL')
C    WRITE(6,*) 'This is the Bisection Method.'
C    WRITE(6,*) 'Has the function F been created in the program? '
C    WRITE(6,*) 'Enter Y or N '
C    WRITE(6,*) ' '
C    READ(5,*) AA
```

Analysis of Engineering Problem

Example Problem 3

Program Source Code (continued)

```
IF(( AA .EQ. 'Y' ) .OR. ( AA .EQ. 'y' )) THEN
  OK = .FALSE.
10  IF (OK) GOTO 11
      WRITE(6,*) 'Input endpoints A < B   separated by blank '
      WRITE(6,*) ' '
      READ(5,*) A, B
      IF (A.GT.B) THEN
          X = A
          A = B
          B = X
      ENDIF
      IF (A.EQ.B) THEN
          WRITE(6,*) 'A cannot equal B '
          WRITE(6,*) ' '
      ELSE
          FA = F( A )
          FB = F( B )
          IF ( FA * FB .GT. 0.0 ) THEN
              WRITE(6,*) 'F(A) and F(B) have same sign '
              WRITE(6,*) ' '
          ELSE
              OK = .TRUE.
          ENDIF
      ENDIF
      GOTO 10
11  OK = .FALSE.
12  IF (OK) GOTO 13
      WRITE(6,*) 'Input tolerance '
      WRITE(6,*) ' '
```

Analysis of Engineering Problem

Example Problem 3

Program Source Code (continued)

```
        READ(5,*) TOL
        IF (TOL.LE.0.0) THEN
            WRITE(6,*) 'Tolerance must be positive '
            WRITE(6,*) ' '
        ELSE
            OK = .TRUE.
        ENDIF
        GOTO 12
13      OK = .FALSE.
14      IF (OK) GOTO 15
        WRITE(6,*) 'Input maximum number of iterations '
        WRITE(6,*) '- no decimal point '
        WRITE(6,*) ' '
        READ(5,*) NO
        IF ( NO .LE. 0 ) THEN
            WRITE(6,*) 'Must be positive integer '
            WRITE(6,*) ' '
        ELSE
            OK = .TRUE.
        ENDIF
        GOTO 14
15      CONTINUE
    ELSE
        WRITE(6,*) 'The program will end so that the function F '
        WRITE(6,*) 'can be created '
        OK = .FALSE.
    ENDIF
    IF (.NOT.OK) GOTO 40
```

Analysis of Engineering Problem

Example Problem 3

Program Source Code (continued)

```
WRITE(6,*) 'Select output destinations: '
WRITE(6,*) '1. Screen '
WRITE(6,*) '2. Text file '
WRITE(6,*) 'Enter 1 or 2 '
WRITE(6,*) ' '
READ(5,*) FLAG
IF ( FLAG .EQ. 2 ) THEN
    WRITE(6,*) 'Input the file name in the form - '
    WRITE(6,*) 'drive:name.ext'
    WRITE(6,*) 'with the name contained within quotes'
    WRITE(6,*) 'as example:  'A:OUTPUT.DTA'' '
    WRITE(6,*) ' '
    READ(5,*) NAME1
    OUP = 3
    OPEN(UNIT=OUP,FILE=NAME1,STATUS='NEW')
ELSE
    OUP = 6
ENDIF
WRITE(6,*) 'Select amount of output '
WRITE(6,*) '1. Answer only '
WRITE(6,*) '2. All intermediate approximations '
WRITE(6,*) 'Enter 1 or 2 '
WRITE(6,*) ' '
READ(5,*) FLAG
WRITE(OUP,*) 'BISECTION METHOD'
IF (FLAG.EQ.2) THEN
    WRITE(OUP,004)
    004 FORMAT(3X,'I',15X,'P',12X,'F(P)')
ENDIF
```

Analysis of Engineering Problem

Example Problem 3

Program Source Code (continued)

```
C      STEP 1
      I=1
C      STEP 2
016   IF (I.GT.N0) GOTO 020
C      STEP 3
C      COMPUTE P(I)
      P=A+(B-A)/2
      FP=F(P)
      IF (FLAG.EQ.2) THEN
        WRITE(OUT,005) I,P,FP
005   FORMAT(1X,I3,2X,E15.8,2X,E15.8)
      ENDIF
C      STEP 4
      IF( ABS(FP).LE.1.0E-20 .OR. (B-A)/2 .LT. TOL) THEN
C      PROCEDURE COMPLETED SUCCESSFULLY
        WRITE(OUT,002) P, I, TOL
        GOTO 040
      ENDIF
C      STEP 5
      I=I+1
C      STEP 6
C      COMPUTE A(I) AND B(I)
      IF( F(A)*FP .GT. 0) THEN
        A=P
      ELSE
        B=P
      ENDIF
      GOTO 016
020   CONTINUE
```

Analysis of Engineering Problem

Example Problem 3

Program Source Code (continued)

```
C      STEP 7
C      PROCEDURE COMPLETED UNSUCCESSFULLY
      WRITE(OUT,003) NO
      IF(OUT.NE.6) WRITE(6,003) NO
040    CLOSE(UNIT=5)
      CLOSE(UNIT=OUT)
      IF (OUT.NE.6) CLOSE(UNIT=6)
      STOP
002    FORMAT(1X,'THE APPROXIMATE SOLUTION IS',1X
*,E15.8,1X,'AFTER',1X,I2,1X,'ITERATIONS, WITH TOLERANCE'
*,1X,E15.8)
003    FORMAT(1X,'THE METHOD FAILS AFTER',1X,I2,1X,'
*ITERATIONS')
      END
```

Accuracy and Precision

Because numerical methods deal extensively with **approximations** connected with the manipulation of numbers, accuracy, precision and error feature prominently in programming the solution. We shall now look at them in more details.

Errors associated with **calculations** and **measurements** can be characterized with regard to their accuracy and precision.

- **Accuracy** refers to how closely a computed or measured value agrees with true value. The opposite, *inaccuracy* (also called *bias*), is defined as systematic deviation from truth.
- **Precision** refers to how closely individual computed or measured value agrees with each other. The opposite, *imprecision* (also called *uncertainty*), refers to the magnitude of the scatter.

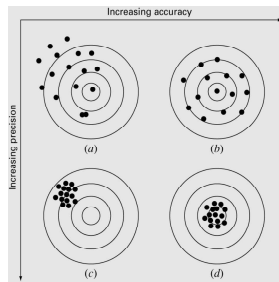


Figure 3 : Concepts of accuracy and precision. (a) Inaccurate and imprecise; (b) accurate and imprecise; (c) inaccurate and precise; (d) accurate and precise.



Error

- Numerical methods should be sufficiently accurate or unbiased to meet the requirements of a particular engineering problem. They should be precise enough for adequate engineering design.
- Error** is the collective term to represent both **inaccuracy** and **imprecision** of predictions by numerical methods. If $\tilde{x}$ is an **approximation** of **true value**, x , then ...

- true or **absolute error** is defined as

$$E_x = x - \tilde{x} \quad (3)$$

- and **relative error** is defined as

$$R_x = \frac{x - \tilde{x}}{x}, x \neq 0 \quad (4)$$

- $\tilde{x}$ is an approximation of x to d **significant digits** if d is the largest integer for which

$$\left| \frac{x - \tilde{x}}{x} \right| < \frac{1}{2} 10^{-d} \quad (5)$$

Error

Example Problem 1

Problem Statement:

Suppose that you are asked to measure the lengths of a bridge and a rivet, and came up with 9,999 cm and 9 cm, respectively. If the true values are 10,000 cm and 10 cm, respectively, compute the *absolute* error and the *relative* error (in %) for each case.

Solution:

Absolute error for measuring

$$\text{bridge: } E_x = x - \tilde{x} = 10000 - 9999 = 1 \text{ cm}$$

$$\text{rivet: } E_x = x - \tilde{x} = 10 - 9 = 1 \text{ cm}$$

Percent relative error for measuring

$$\text{bridge: } R_x = \frac{x - \tilde{x}}{x} \times 100 = \frac{1}{10000} \times 100 = 0.01\%$$

$$\text{rivet: } R_x = \frac{x - \tilde{x}}{x} \times 100 = \frac{1}{10} \times 100 = 10\%$$

Error

In Absence of True Value

- How do we determine error estimates in the absence of knowledge regarding the true value?
- Example: Many numerical methods use an *iterative approach* to compute answers. In such approach, a **present approximation** is made on the basis of a **previous approximation** i.e. process is performed repeatedly, or iteratively, to successfully compute better and better approximations.
- In this case, error is estimated as the difference between previous and current approximations, thus

$$\epsilon = \frac{\text{current approximation} - \text{previous approximation}}{\text{current approximation}}$$

- ① Errors in mathematical modeling:
 - simplifying approximation,
 - assumption made in representing physical system by mathematical equations
- ② Blunders:
 - undetected programming errors,
 - silly mistakes
- ③ Errors in input:
 - due to unavoidable reasons e.g. errors in data transfer,
 - uncertainties associated with measurements
- ④ Machine errors:
 - rounding,
 - chopping,
 - overflow,
 - underflow
- ⑤ Truncation errors associated with mathematical process:
 - approximate evaluation of an infinite series,
 - integral involving infinity

Error

Sources: Due to Floating-Point Representation

- Number is expressed as fractional part, called a **mantissa** or **significand** and an integer part, called an **exponent** or **characteristic**

$$m \cdot b^e$$

where m is mantissa, b is the base of the number system being used and e the exponent. If the number has leading zeros digits, the mantissa is usually **normalized**. If $1/34 = 0.029411765 \dots$ were to be stored in a floating-point base-10 system that allows only four decimal places to be stored, then $1/34$ would be stored as

$$1/34 = 0.0294 \times 10^0 \rightarrow 0.2941 \times 10^{-1}$$

Error

Sources: Due to Truncation Error

- The discrepancy introduced by the use of an **approximate expression** in place of an **exact mathematical expression**.
- Example: Taylor's series expansion of $\ln(1+x)$

$$\begin{aligned}y(x) = \ln(1+x) &= \sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i} x^i \\&= x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \frac{1}{5}x^5 - \frac{1}{6}x^6 + \frac{1}{7}x^7 + \dots; \quad |x| \leq 1\end{aligned}$$

If $y(x)$ is approximated by the first four terms of this Taylor's series, the resulting discrepancy between the exact function $y(x)$ and the approximate function $\tilde{y}(x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4$, is called the truncation error.

Error

Sources: Due to Round-off

- Computer can only store a finite number of digits, so actual numbers may undergo **chopping** or **rounding**.
- Let a decimal number $x = 0.b_1b_2 \dots b_ib_{i+1}b_{i+2}$ where $0 \leq b_i \leq 9$ for $i \geq 1$. If the maximum number of decimal digits used in the floating-point computation is i :
 - **chopped floating-point** representation of x is $x_{\text{chop}} = 0.b_1b_2 \dots b_i$ where i th digit of x_{chop} is identical to the i th digit of x .
 - **rounded floating-point** representation of x is $x_{\text{round}} = 0.b_1b_2 \dots b_{i-1}d_i$ where $d_i (1 \leq d_i \leq 9)$ is obtained by rounding the number $d_id_{i+1}d_{i+2} \dots$ to the nearest integer.
- Numerical solution of engineering problem uses suitable algorithm and **local computational errors** involved in various steps of this algorithm will accumulate to a computational error in output
- Local computational error arise due to errors involved during arithmetic operations such as
 - subtraction of numbers of **near-equal magnitude**,
 - irrational numbers (such as $\sqrt{3}$ and π) being replaced by **machine numbers** with finite number of digits

Error

Example Problem 2

Problem Statement:

The value of e is given by $e = 2.718281828459045 \dots$. Show the seven-digit representations of e by chopping and rounding are

$$e_{\text{chop}} = 0.2718281 \times 10^1 \quad e_{\text{round}} = 0.2718282 \times 10^1$$

Solution:

Work through the example.

Propagation of Error

- Error in the output of a procedure due to the error in the input data
- Output of a procedure f is a function of input parameters $(x_1, x_2, \dots, x_n)$

$$f = f(x_1, x_2, \dots, x_n) \equiv f(\vec{X})$$

- Value of f is found by Taylor's series expansion about the approximate values $\vec{X} = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}^T$ as

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) + \frac{\partial f}{\partial x_1}(\vec{X})(x_1 - \bar{x}_1) \\ &\quad + \frac{\partial f}{\partial x_2}(\vec{X})(x_2 - \bar{x}_2) + \dots + \frac{\partial f}{\partial x_n}(\vec{X})(x_n - \bar{x}_n) \\ &\quad + \text{higher order derivative terms} \end{aligned}$$

Propagation of Error

- Neglecting higher order derivative terms, the error in the output can be expressed as

$$\begin{aligned}\Delta f &= f - \bar{f} \\ &\equiv f(x_1, x_2, \dots, x_n) - f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)\end{aligned}$$

and denoting errors in input parameters as

$$\Delta x_i = x_i - \bar{x}_i, \quad i = 1, 2, \dots, n$$

we can estimate **propagation error** as

$$\Delta f \approx \sum_{i=1}^n \frac{\partial f}{\partial x_i}(\vec{X})(x_i - \bar{x}_i)$$

Taylor's series expansion of $f(x)$ in Figure 4 at a known point x_i for a given step size h yields

$$\begin{aligned}f(x_{i+1}) &= f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 \\ &\quad + \frac{f'''(x_i)}{3!}h^3 + \dots\end{aligned}$$

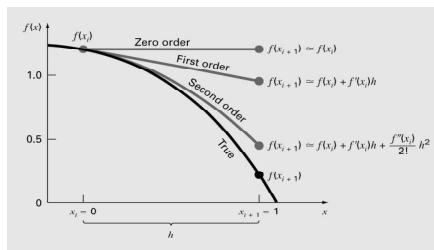


Figure 4 : Relative margin of error for neglecting higher order terms of Taylor's series.

Propagation of Error

- If $f(x_1, x_2, \dots, x_n) \neq 0$ and $x_i \neq 0$, the **relative propagation error**, ε_f , is

$$\varepsilon_f = \frac{\Delta f}{f} = \sum_{i=1}^n \left\{ \frac{x_i}{f(\vec{X})} \frac{\partial f}{\partial x_i}(\vec{X}) \right\} \varepsilon_{x_i} \quad (6)$$

where ε_{x_i} is relative error in x_i

$$\varepsilon_{x_i} = \frac{x_i - \bar{x}_i}{x_i} \quad i = 1, 2, \dots, n$$

- The quantity

$$c_i = \left\{ \frac{x_i}{f(\vec{X})} \frac{\partial f}{\partial x_i}(\vec{X}) \right\}$$

in Eq. (6) is called the **amplification** or **condition number** of relative input error ε_{x_i} .

Propagation of Error

In Arithmetic Operations

- When two numbers are used in an arithmetic operations, the numbers cannot be stored **exactly** by the **floating-point representation**.
- Let x and y be the **exact** number and $\tilde{x}$ and $\tilde{y}$ their approximate values. Then

$$x = \tilde{x} + \varepsilon_x \quad y = \tilde{y} + \varepsilon_y$$

ε_x and ε_y denote errors in x and y , respectively.

- When arithmetic operation, say **multiplication**, is carried out on the numbers, **associated error**, E , results

$$E = xy - \tilde{x}\tilde{y} = xy - (x - \varepsilon_x)(y - \varepsilon_y) = x\varepsilon_y + y\varepsilon_x - \varepsilon_x\varepsilon_y$$

and **relative error**, R , is

$$R = \frac{E}{xy} = \frac{\varepsilon_x}{x} + \frac{\varepsilon_y}{y} - \frac{\varepsilon_x}{x} \frac{\varepsilon_y}{y} = R_x + R_y - R_x R_y \approx R_x + R_y$$

where $|R_x| \ll 1$ and $|R_y| \ll 1$

Propagation of Error

Example Problem 3

Problem Statement:

The deflection y of the top of a sailboat mast is

$$y = \frac{FL^4}{8EI}$$

where F is a uniform side loading (lb/ft), L is height (ft), E is the modulus of elasticity (lb/ft²), and I is the moment of inertia (ft⁴). Estimate the error in y given the following data:

$F = 50 \text{ lb/ft}$	$L = 30 \text{ ft}$	$E = 1.5 \times 10^8 \text{ lb/ft}^2$	$I = 0.06 \text{ ft}^4$
$\Delta F = 2 \text{ lb/ft}$	$\Delta L = 0.1 \text{ ft}$	$\Delta E = 0.01 \times 10^8 \text{ lb/ft}^2$	$\Delta I = 0.0006 \text{ ft}^4$

Solution:

Work through the example.

Bibliography

- ① STEVEN C. CHAPRA, RAYMOND P. CANALE (2006): *Applied Numerical Methods for Engineers, 5ed*, ISBN 007-124429-8, McGraw-Hill
- ② SINGIRESU S. RAO (2002): *Applied Numerical Methods for Engineers and Scientists*, ISBN 0-13-089480-X, Prentice Hall
- ③ DAVID KINCAID AND WARD CHENEY (1991): *Numerical Analysis: Mathematics of Scientific Computing*, ISBN 0-534-13014-3, Brooks/Cole Publishing Co.
- ④ STEVEN C. CHAPRA (2005): *Applied Numerical Methods with MATLAB for Engineers and Scientists*, ISBN 007-124484-0, McGraw-Hill
- ⑤ WILLIAM J. PALM III (2005): *Introduction to Matlab 7 for Engineers*, ISBN 007-123262-1, McGraw-Hill
- ⑥ JOHN D. ANDERSON, JR. (1995): *Computational Fluid Dynamics–The Basics with Applications*, ISBN 007-113210-4, McGraw-Hill

