Mathematical Modeling and Engineering Problem solving

Chapra: Chapter-1

Mathematical Modeling and Engineering Problem solving

- Requires understanding of engineering systems
  - By observation and experiment
  - Theoretical analysis and generalization
- Computers are great tools, however, without fundamental understanding of engineering problems, they will be useless.

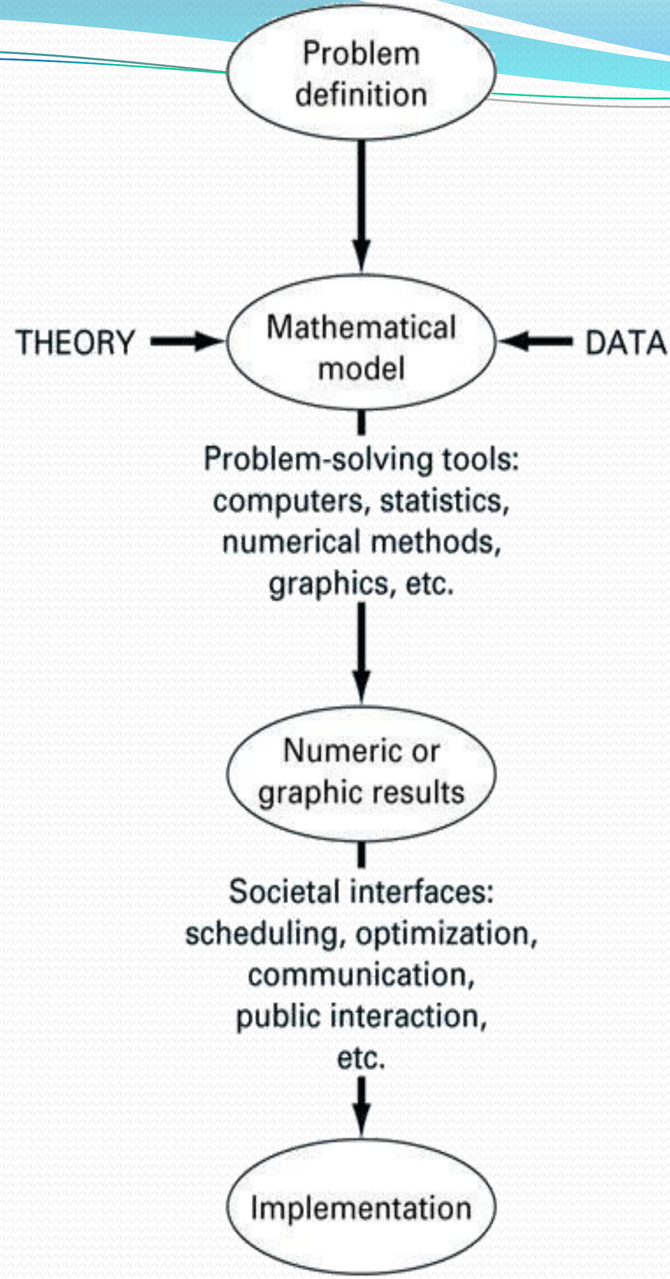


Fig. 1.1
Chapra

- A mathematical model is represented as a functional relationship of the form

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

- ***Dependent variable***: Characteristic that usually reflects the state of the system
- ***Independent variables***: Dimensions such as time and space along which the systems behavior is being determined
- ***Parameters***: reflect the system's properties or composition
- ***Forcing functions***: external influences acting upon the system

Newton's 2nd law of Motion

- States that "*the time rate change of momentum of a body is equal to the resulting force acting on it*."

- The model is formulated as

$$\boldsymbol{F = m\,a}$$

$\boldsymbol{F}$ = net force acting on the body (N)

$\boldsymbol{m}$ = mass of the object (kg)

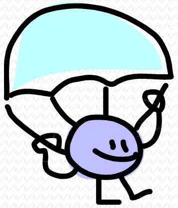
$\boldsymbol{a}$ = its acceleration (m/s^2)

- Formulation of Newton’s 2nd law has several characteristics that are typical of mathematical models of the physical world:
  - It describes a natural process or system in mathematical terms
  - It represents an idealization and simplification of reality
  - Finally, it yields reproducible results, consequently, can be used for predictive purposes.

- Some mathematical models of physical phenomena may be much more complex.

- Complex models may not be solved exactly or require more sophisticated mathematical techniques than simple algebra for their solution

  - Example, modeling of a falling parachutist:



$$\frac{dv}{dt} = \frac{F}{m}$$

c is the proportionality constant called the *drag coefficient(kg/s)*

$$F = F_D + F_U$$

$$F_D = mg$$

$$F_U = -cv$$

$$\frac{dv}{dt} = \frac{mg - cv}{m}$$

Exact Solution

- This is a differential equation and is written in terms of the differential rate of change *dv*/*dt* of the variable that we are interested in predicting.
- If the parachutist is initially at rest (*v* = 0 at *t* = 0), using calculus

$$\frac{dv}{dt} = g - \frac{c}{m}v$$

$$v(t) = \frac{gm}{c}\left(1 - e^{-(c/m)t}\right)$$

•Independent variable

•Dependent variable

•Forcing function

•Parameters

Numerical Solution

$$\frac{dv}{dt} \cong \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i} = g - \frac{c}{m} v(t_i)$$

$$v(t_{i+1}) = v(t_i) + \left[g - \frac{c}{m} v(t_i) \right] (t_{i+1} - t_i)$$

Find and compare the values of *v(t)* at t={0, 2, 4, 6, 8 …}

- Using Exact solution
- Using Numerical solution
- Compare the results

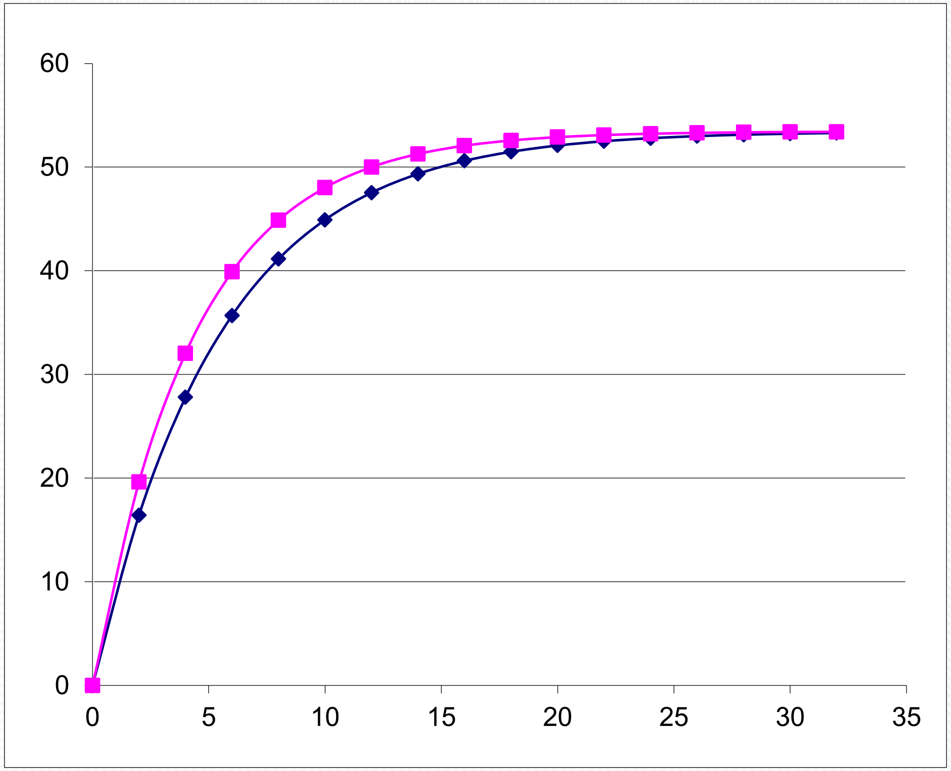
$$v(t) = \frac{gm}{c}\left(1 - e^{-(c/m)t}\right)$$

m= 68.1

c= 12.5

g= 9.81

Δt= 2



t	Actual	Estimate
0	0	0
2	16.42172	19.62
4	27.79763	32.03736
6	35.67812	39.89621
8	41.13722	44.87003
10	44.91893	48.01792
12	47.53865	50.01019
14	49.35343	51.27109
16	50.61058	52.06911
18	51.48146	52.57416
20	52.08475	52.89381
22	52.50267	53.09611
24	52.79218	53.22415
26	52.99273	53.30518
28	53.13166	53.35646
30	53.2279	53.38892
32	53.29457	53.40946

Why Numerical?

- There exists many cases where analytical/exact solution is not possible
- We can have generic numeric methods for solving a variety of mathematical forms

Conservation Laws and Engineering

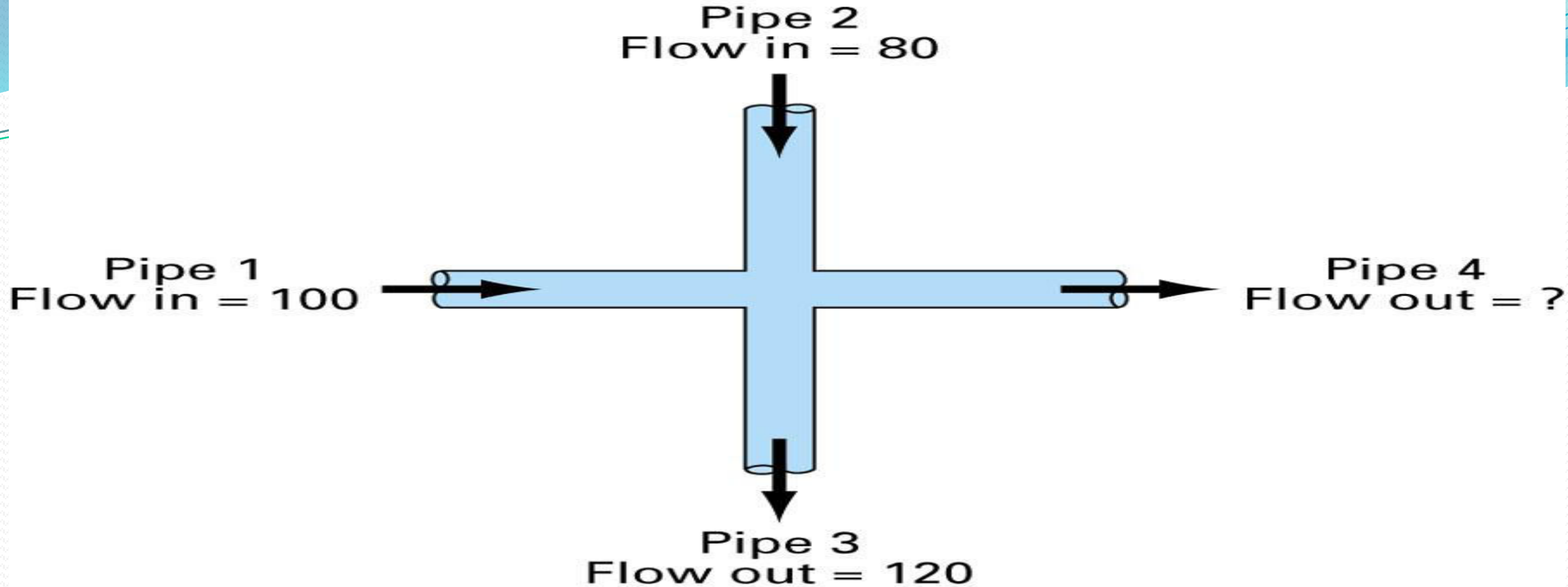
- Conservation laws are the most important and fundamental laws that are used in engineering.

$$\text{Change} = \text{increases} - \text{decreases} \qquad (1.13)$$

- Change implies changes with time (transient). If the change is nonexistent (steady-state), Eq. 1.13 becomes

$$\text{Increases} = \text{Decreases}$$

•Fig 1.6



- For steady-state incompressible fluid flow in pipes:

$$\text{Flow in} = \text{Flow out}$$

or

$$100 + 80 = 120 + \text{Flow}_4$$

Truncation Errors and Tailors Series

Chapra: Chapter-4

Truncation Error

- *Truncation errors* are those that result from using an approximation in place of an exact mathematical procedure.
- *Example:*

$$\frac{dv}{dt} \cong \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}$$

- *How much error is introduced with this approximation?*
- We can use *Tailors Series* to estimate truncation error.

Tailors Series

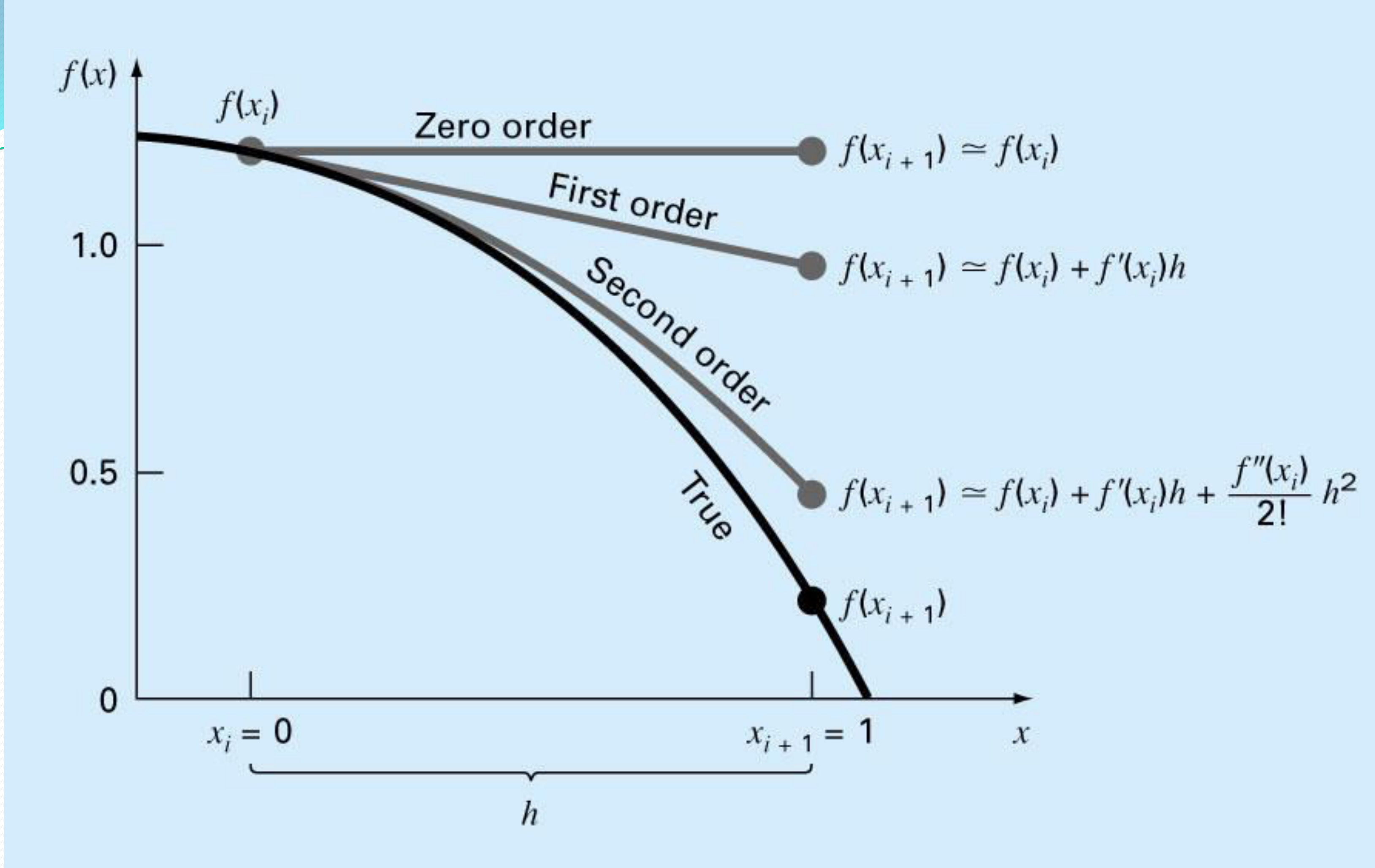
- *n^{th} order* approximation

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{f''}{2!}(x_{i+1} - x_i)^2 + \dots$$
$$+ \frac{f^{(n)}}{n!}(x_{i+1} - x_i)^n + R_n$$

- $(x_{i+1} - x_i) = h$ *step size* (define first)

$$R_n = \frac{f^{(n+1)}(\varepsilon)}{(n+1)!} h^{(n+1)} \qquad x_i \le \varepsilon \le x_{i+1}$$

Reminder term, R_n, accounts for all terms from (n+1) to infinity.



$$f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$$

The approximation of *f(x)* at *x=1* by zero-order, fist-order and second-order Tailor series expansion

Insight

- Each additional term contributes to the approximation
- n^{th}-order Tailor Series gives exact value of n^{th}-order polynomial
- Inclusion of a few terms gives an approximation that is good enough for practical purpose.
- The Remainder:
  - ε is not exactly known.
  - Need to determine $f^{n+1}(x_{i+1})$, which require the determination of the $(n+1)$th derivative of $f(x)$. If we know $f(x)$ then we do not need to use Tailors series!
  - Yet, $R_n=O(h^{n+1})$ gives insight into error. E.g., if error is $O(h)$ then halving step size will halve the error. If error is $O(h^2)$ then halving the step size will quarter the error, and so on.

Effect of Step Size

Estimate

$$v(t_{i+1}) = v(t_i) + \left[g - \frac{c}{m} v(t_i) \right] (t_{i+1} - t_i)$$

Actual

$$v(t) = \frac{gm}{c}\left(1 - e^{-(c/m)t}\right)$$

m=	68.1
c=	12.5
g=	9.81
Δt=	2

t	Actual	Estimate	Error%
0	0	0	
2	16.42172	19.62	19.47591
4	27.79763	32.03736	15.25213
6	35.67812	39.89621	11.82263
8	41.13722	44.87003	9.074043
10	44.91893	48.01792	6.899074
12	47.53865	50.01019	5.199019
14	49.35343	51.27109	3.885579
16	50.61058	52.06911	2.881848

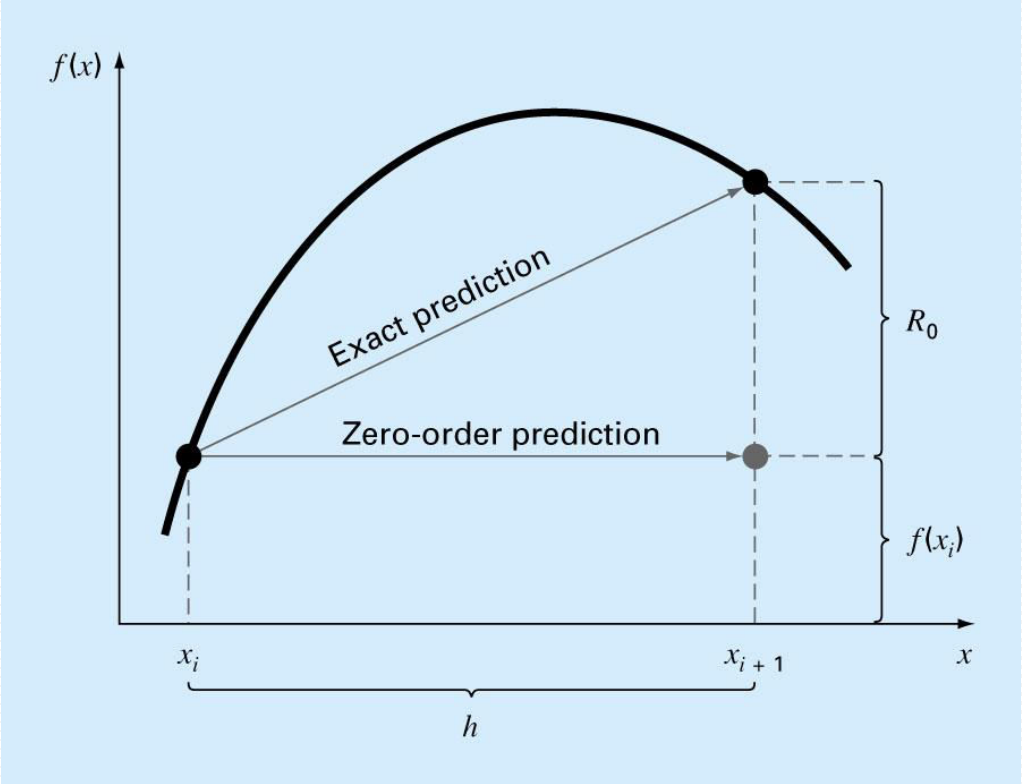
t	Actual	Estimate	Error%
0	0	0	
1	8.962318	9.81	9.458288
2	16.42172	17.81934	8.510793
3	22.63024	24.35854	7.637128
4	27.79763	29.69744	6.834436
5	32.09849	34.05637	6.099607
6	35.67812	37.6152	5.429315
7	38.65748	40.52079	4.820066
8	41.13722	42.89306	4.268249
9	43.20112	44.82988	3.770176
10	44.91893	46.4112	3.322138
11	46.34867	47.70225	2.920443
12	47.53865	48.75633	2.561458
13	48.52908	49.61693	2.241647
14	49.35343	50.31957	1.957596
15	50.03953	50.89323	1.706043
16	50.61058	51.36159	1.483897

Insight : R_n

$$f(x_{i+1}) \cong f(x_i)$$

$$R_0 = f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!} + \cdots\cdots$$

$$R_0 \cong f'(x_i)h$$

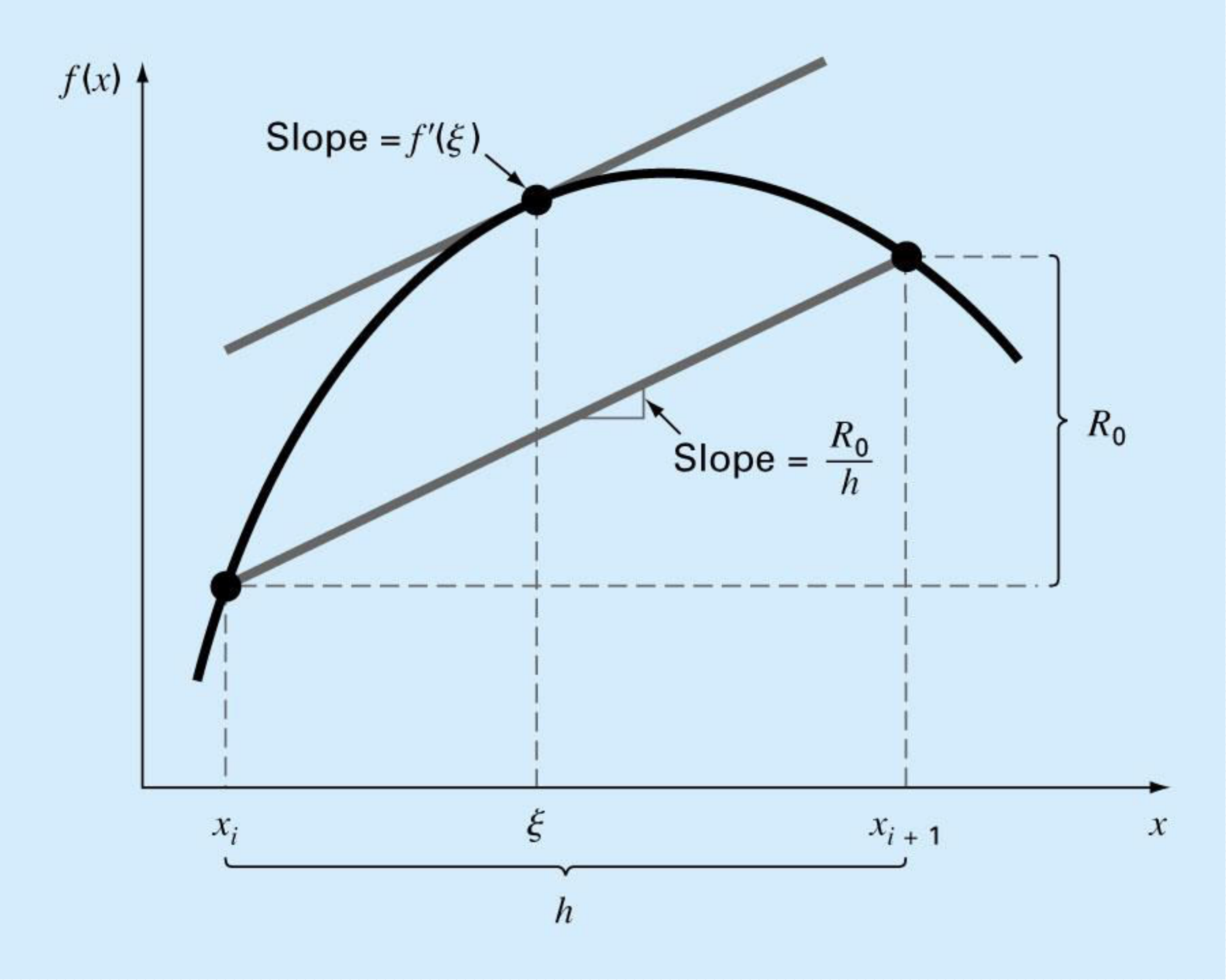


Insight : R_n

$$R_0 \cong f'(x_i)h$$

$$R_0 = f'(\varepsilon)h$$

$$R_1 = \frac{f''(\varepsilon)}{2!}h^2$$



How to get derivatives?

- We will be given value of unknown $f(x)$ for some value of x
- We can estimate $f^n(a)$, i.e. the n^{th} order derivate of $f(x)$ at $x=a$ numerically without knowing $f(x)$

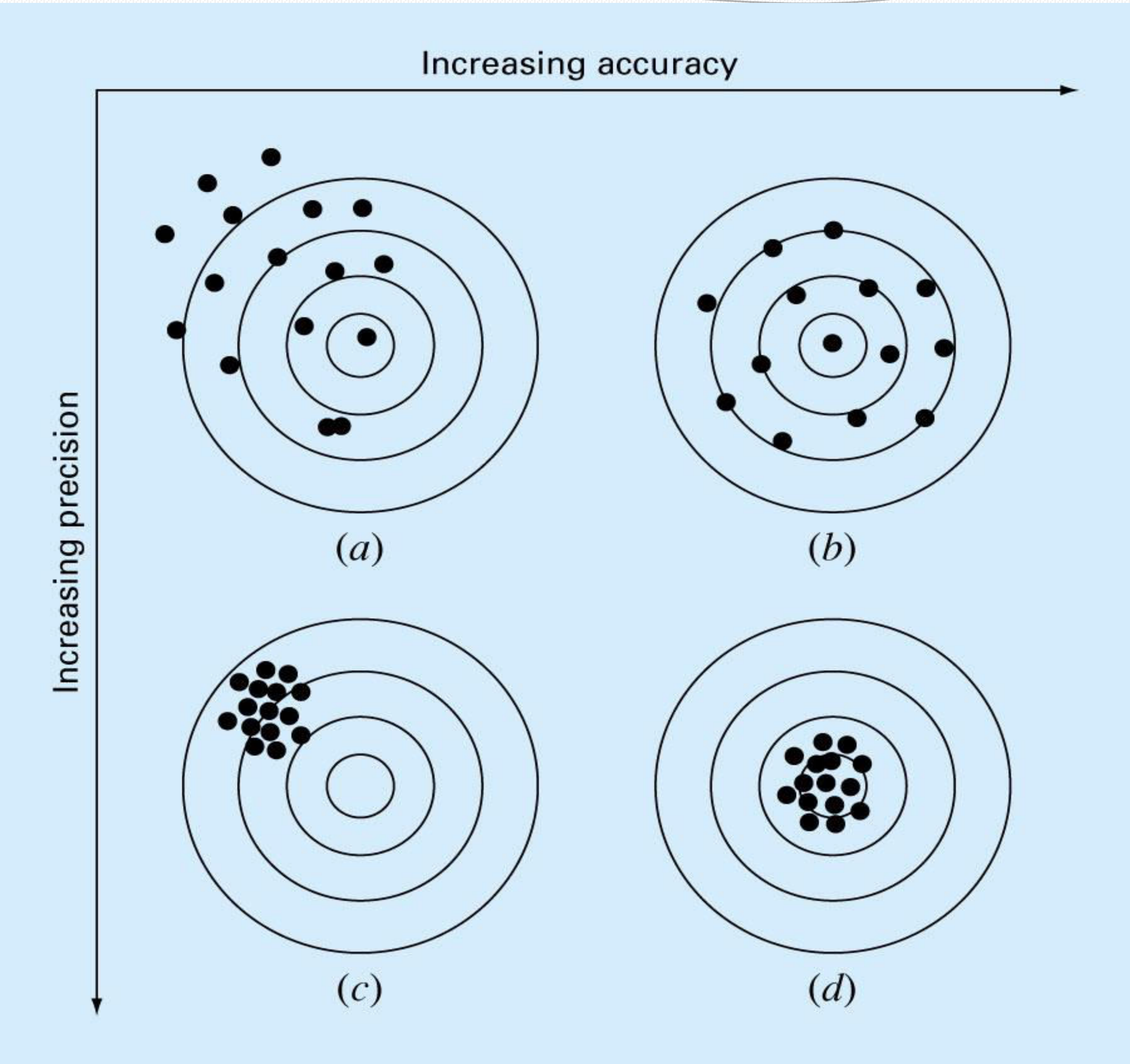
Approximations & Round-off Errors

Chapra: Chapter-3

- For many engineering problems, we cannot obtain analytical solutions
- Numerical methods yield approximate solution that are close to the exact analytical solution. We cannot exactly compute the errors associated with numerical methods.
  - Only rarely given data are exact, since they originate from measurements. Therefore there is probably error in the input information.
  - Algorithm itself usually introduces errors as well, e.g., unavoidable round-offs, etc …
  - The output information will then contain error from both of these sources.
- How confident we are in our approximate result?
- The question is "*how much error is present in our calculation and is it tolerable?*"

- Accuracy. How close is a computed or measured value to the true value
- Precision (or *reproducibility*). How close is a computed or measured value to previously computed or measured values.
- Inaccuracy (or *bias*). A systematic deviation from the actual value.
- Imprecision (or *uncertainty*). Magnitude of scatter.

•Fig. 3.2



Significant Figures

- Number of significant figures indicates precision. Significant digits of a number are those that can be *used* with *confidence,* e.g., the number of certain digits plus one estimated digit.

53,8<u>00</u> How many significant figures?

5.38×10^4	3
5.380×10^4	4
5.3800×10^4	5

Zeros are sometimes used to locate the decimal point not significant figures.

0.00001753	4
0.0001753	4
0.001753	4

Error Definitions

True Value = Approximation + Error

E_t = True value – Approximation (+/-)

•True error

$$\text{True fractional relative error} = \frac{\text{true error}}{\text{true value}}$$

$$\text{True percent relative error, } \varepsilon_t = \frac{\text{true error}}{\text{true value}} \times 100\%$$

- For numerical methods, the true value will be known only when we deal with functions that can be solved analytically (simple systems). In real world applications, we usually not know the answer a priori. Then

$$\varepsilon_a = \frac{\text{Approximate error}}{\text{Approximation}} \times 100\%$$

- *Iterative approach,* example Newton's method

$$\underset{(+/-)}{\varepsilon_a} = \frac{\text{Current approximation - Previous approximation}}{\text{Current approximation}} \times 100\%$$

- Use absolute value.
- Computations are repeated until stopping criterion is satisfied.

$$\left|\varepsilon_a\right| \langle \varepsilon_s$$

•Pre-specified % tolerance based on the knowledge of your solution

- If the following criterion is met

$$\varepsilon_s = (0.5 \times 10^{(2-n)})\%$$

you can be sure that the result is correct to at least n significant figures.

Example 3.2

- In mathematics, function can often be represented by infinite series. For example, the exponential function can be computed using the *Maclaurin series exapansion* as:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots\dots\dots\dots\dots\dots + \frac{x^n}{n!} + \dots\dots\dots$$

Thus, as more terms are added in sequence, the approximation becomes a better and better estimate.

Starting with the simplest version, $e^x=1$, add terms one at a time to estimate $e^{0.5}$. After each new term is added, compute the true and approximate percent relative errors. Note that the true value of $e^{0.5}$ = 1.648721. Add terms until the absolute value of the approximate error estimate ε_a falls below a pre-specified error criterion ε_s conforming to three significant digits.

Round-off Errors

- Numbers such as π, e, or $\sqrt{7}$ cannot be expressed by a fixed number of significant figures.
- Computers use a base-2 representation, they cannot precisely represent certain exact base-10 numbers.
- Fractional quantities are typically represented in computer using "floating point" form, e.g.,

•Integer part

$$m.\boxed{b^e}$$

•exponent

•mantissa

•Base of the number system used

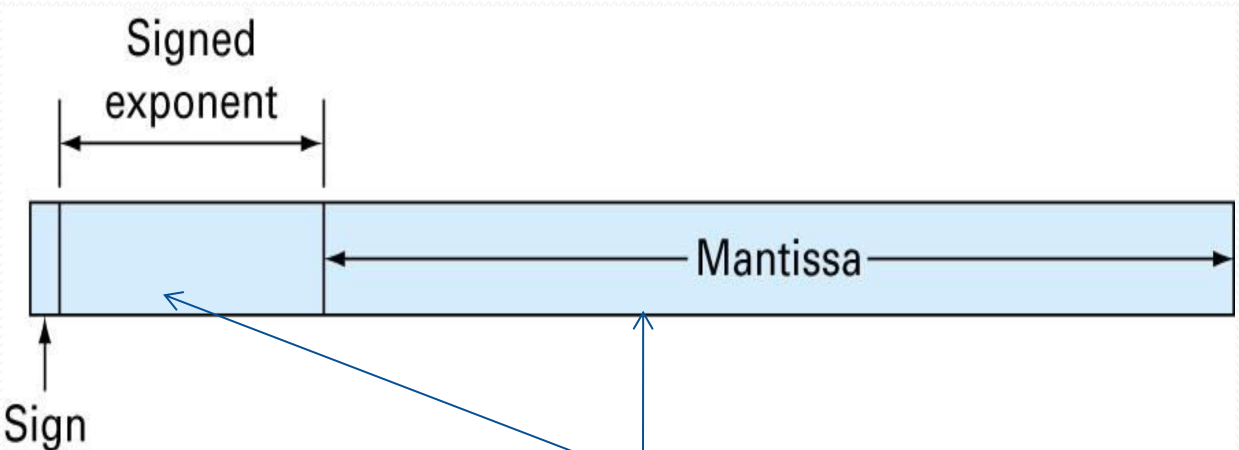
•Integer part

•exponent

$$m \times \boxed{b^e}$$

•mantissa

•Base of the number system used



$$156.78 \Leftrightarrow 0.15678 \times 10^3$$

$$\frac{1}{34} = 0.029411765$$

Suppose only 4 decimal places to be stored

$$0.0294 \times 10^0 \qquad \frac{1}{b} \le |m| < 1$$

- Normalized to remove the leading zeroes. Multiply the mantissa by 10 and lower the exponent by 1

$0.294\underline{1} \times 10^{-1}$

•Additional significant figure is retained

$$\frac{1}{b} \leq |m| < 1$$

Therefore

for a base-10 system $0.1 \leq m < 1$

for a base-2 system $0.5 \leq m < 1$

- Floating point representation allows both fractions and very large numbers to be expressed on the computer. However,
  - Floating point numbers take up more room.
  - Take longer to process than integer numbers.
  - Round-off errors are introduced because mantissa holds only a finite number of significant figures.

Chopping

Example:

π=3.14159265358 to be stored on a base-10 system carrying 7 significant digits.

π=3.141592 chopping error ε_t=0.00000065

If rounded

π=3.141593 ε_t=0.00000035

- Some machines use chopping, because rounding adds to the computational overhead. Since number of significant figures is large enough, resulting chopping error is negligible.