1. Mathematical modeling and engineering problem solving

- mathematical modeling
- simple mathematical model
- conservation laws

Knowledge and understanding vs tools

- effective use of computational tools depends on insight into engineering systems
- even advanced tools are ineffective without a deep understanding of the system
- computers enhance problem-solving but rely on knowledge of system behavior

Gaining understanding

- empirical: observations and experiments yield data and qualitative insights
- theoretical: repeated patterns lead to fundamental laws (*e.g.*, conservation laws)

Empirical and theoretical problem solving

Empirical approach

- observe, measure, and experiment
- identify patterns and trends in data

Theoretical approach

- formulate principles and laws
- derive predictions and explanations

effective engineering integrates both approaches

Data-theory relationship

- new data improves or updates models
- theories guide how experiments are designed
- theories unify observations into key principles

Mathematical models and problem solving

a *mathematical model* represents a physical system using equations

Types of models

- empirical: derived from observed data (*e.g.*, curve fitting)
- theoretical: based on physical laws (*e.g.*, Newton's laws)

Problem solving process

- *problem definition*: specify the system, goals, and constraints
- *data and theory*: integrate observations with fundamental laws
- *mathematical model*: represent the system with equations
- *problem-solving tools*: apply numerical methods, computation, and statistics
- *implementation*: produce quantitative or graphical results
- *societal interfaces*: interpret, optimize, communicate, and apply outcomes

Outline

- mathematical modeling
- **simple mathematical model**
- conservation laws

Mathematical model

in general, a mathematical model can be expressed as a functional relationship:

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

- f: a multi-variable function representing the model
- dependent variable: system response or state (*e.g.*, velocity)
- independent variables: dimensions such as time or space (*e.g.*, t)
- parameters: fixed system properties (*e.g.*, mass)
- forcing functions: external inputs or influences (*e.g.*, applied force)

Example: Newton's second law

Physical law: net force acting on an object equals its mass times acceleration:

$$F = ma$$

where F = net force (N), m = mass (kg), a = acceleration (m/s^2)

Rewritten in model form

$$a = \frac{F}{m}$$

- dependent variable: a (acceleration)
- forcing function: F (net force)
- parameter: m (mass)
- no independent variable (time-independent for this simple case)

Range of mathematical models

Simple vs. complex models

- simple models: algebraic equations (*e.g.*, $F = ma$)
- complex models: sets of differential equations
  - example: modeling fluid flow or heat transfer

Solution approaches

- analytical: exact solutions (possible for simple models)
- numerical: approximate solutions for complex models

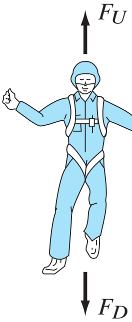
Role of numerical methods

- enable solutions for complex models where analytical methods fail
- provide systematic approximations with controllable error
- connect mathematical models to practical engineering outcomes
- example: solving differential equations for dynamic systems

numerical methods bridge theory and application

Example: falling parachutist

Problem: model the velocity of a parachutist under gravity and air resistance



- apply Newton's second law to a dynamic system
- account for forces: gravity (downward) and air resistance (upward)
- derive a differential equation for velocity
- solve analytically or numerically

Example: falling parachutist

applying Newton's law $ma = F$ (conservation of momentum) with

$$a = \frac{dv}{dt}, \quad F = F_U + F_D, \quad F_D = mg, \quad F_U = -cv$$

gives the differential equation

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

- m = mass kg, $g = 9.81$ m/s^2 (gravitational acceleration)
- c = drag coefficient (kg/s), v = velocity
- can be solved exactly (next page)
- numerical approach: approximate using finite differences (covered shortly)

Example: falling parachutist

Solution: for initial condition $v(0) = 0$

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

- maps to general model form: $v(t) = f(t, m, c, g)$
- parameters: m, c
- forcing function: g
- independent variable: t
- dependent variable: $v(t)$
- predicts terminal velocity as $t \longrightarrow \infty$: $v \longrightarrow \frac{gm}{c}$

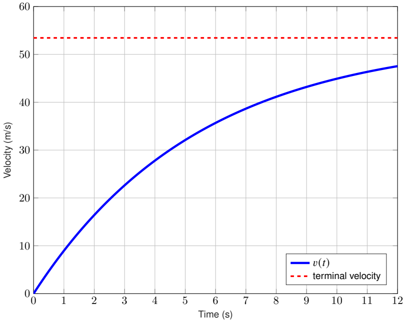
Example: falling parachutist

set parameters: $m = 68.1$, kg, $c = 12.5$ kg/s, $g = 9.81$ m/s^2

$$v(t) = \frac{9.81 \cdot 68.1}{12.5}(1 - e^{-(12.5/68.1)t}) = 53.44(1 - e^{-0.18355t}) \text{ m/s}$$

```
m = 68.1; c = 12.5; g = 9.81;
t = 0:0.1:12;
v = (g*m/c)*(1 - exp(-(c/m)*t));
plot(t, v); xlabel('Time (s)'); ylabel('Velocity (m/s)');
```

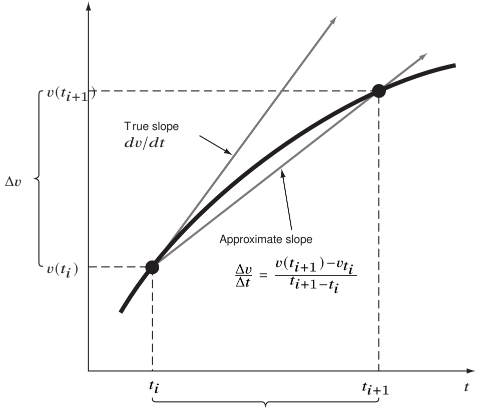
t (s)	v (m/s)
0	0.00
2	16.42
4	27.80
6	35.68
8	41.14
10	44.92
12	47.54
∞	53.44



Example: finite difference approximation

Objective: solve numerically by approximating the derivative in the parachutist model

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



Example: finite difference approximation

substitute into differential equation on page 1.10:

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

rearrange for next velocity

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

- formula: new value = old value + slope × step size
- slope: right-hand side of differential equation
- known as *Euler's method*

Example: finite difference approximation

compute parachutist velocity using Euler's method and $\Delta t = t_{i+1} - t_i = 2$ s

- parameters: $g = 9.81$ m/s^2, $m = 68.1$ kg, $c = 12.5$ kg/s
- initial condition: $v(0) = 0$ m/s

First step ($t = 0$ to $t = 2$):

$$v(2) = 0 + \left[9.81 - \frac{12.5}{68.1}(0)\right] \cdot 2 = 19.62 \text{ m/s}$$

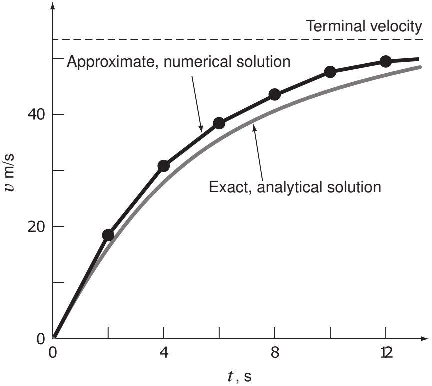
Second step ($t = 2$ to $t = 4$):

$$v(4) = 19.62 + \left[9.81 - \frac{12.5}{68.1}(19.62)\right] \cdot 2 = 32.04 \text{ m/s}$$

Example: finite difference approximation

Results ($\Delta t = 2$ s)

t (s)	v (m/s)
0	0.00
2	19.62
4	32.04
6	39.90
8	44.87
10	48.02
12	50.01
∞	53.44



Accuracy vs computational effort

- Euler's method approximates the true solution
- finite step size (Δt) causes discrepancy (error)
- smaller Δt (*e.g.*, 1 s) reduces error but increase computation
  - smaller steps improve accuracy
  - double computations per halving of step size

Outline

- mathematical modeling
- simple mathematical model
- **conservation laws**

Conservation laws

conservation laws govern engineering systems

General form

$$\text{change} = \text{increases} - \text{decreases}$$

- simple yet powerful for modeling complex systems
- applied to predict changes or balance states
- examples: mass, momentum, energy conservation
- called *time-variable* or *transient* computation

No change: system in balance

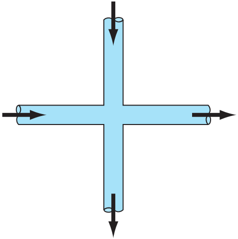
$$\text{change} = 0 \implies \text{increases} = \text{decreases}$$

- called steady-state computation
- many applications in engineering

Example: fluid flow

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

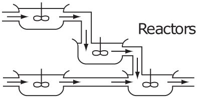
Example: pipe junction



- flow into junction equals flow out
- $100 + 80 = 120 +$ pipe 4 flow out $\Rightarrow$ pipe 4 flow out $= 60$

Engineering applications

Chemical engineering



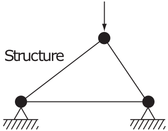
Conservation of mass

Mass balance:

Over a unit of time period

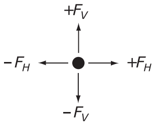
Δmass = inputs − outputs

Civil engineering



Conservation of momentum

Force balance:



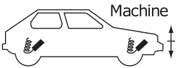
At each node

Σ horizontal forces (F_H) = 0

Σ vertical forces (F_V) = 0

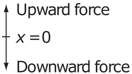
Engineering applications

Mechanical engineering



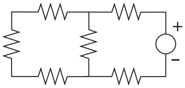
Conservation of momentum

Force balance:



$$m\frac{d^2x}{dt^2} = \text{downward force} - \text{upward force}$$

Electrical engineering



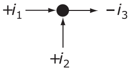
Circuit

Conservation of charge

Current balance:

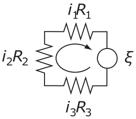
For each node

Σ current (i) = 0



Conservation of energy

Voltage balance:



Around each loop

Σ emf's − Σ voltage drops for resistors = 0

$\Sigma\,\xi - \Sigma\, iR = 0$

References and further readings

- S. C. Chapra and R. P. Canale. *Numerical Methods for Engineers* (8th edition). McGraw Hill, 2021. (Ch.1)
- S. C. Chapra. *Applied Numerical Methods with MATLAB for Engineers and Scientists* (5th edition). McGraw Hill, 2023. (Ch.1)