Applied Numerical Methods for Civil Engineering

CGN 3405 - 0002

Mathematical Modeling & Engineering Problem Solving

Xinyu Chen

Assistant Professor

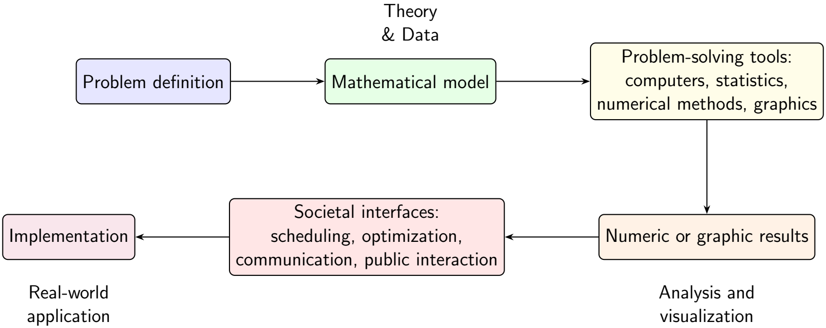
University of Central Florida

How to understand

Applied Numerical Methods for Civil Engineering?

Numerical methods are techniques by which **mathematical problems** are formulated so that they can be solved with **arithmetic operations**.

Engineering Problem Solving Process



Bungee Jumping

Engineering Task.

- A bungee jumping company needs to **predict velocity vs. time** during free-fall to design safe bungee cords.
- **Key Questions**:
    - What is the **maximum velocity** reached?
    - How long until maximum velocity?
    - What cord length is needed?

Physical Forces F_g and F_a

Two Main Forces: Physical Forces Acting on Jumper

$$\boxed{F = F_g - F_a = m \cdot g - c_d \cdot v^2}$$

- Gravity (Downward)

$$F_g = m \cdot g$$

with

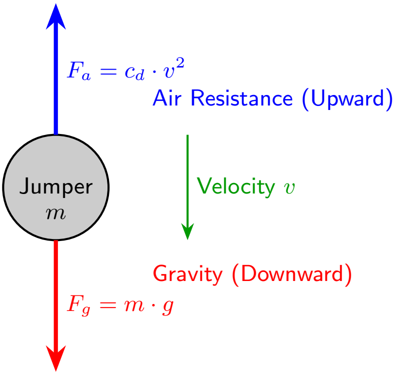
  - m = mass (kg)
  - $g = 9.81\,\text{m/s}^2$, gravitational acceleration

- Air Resistance (Upward)

$$F_a = c_d \cdot v^2$$

with

  - c_d = drag coefficient (kg/m)
  - v = velocity



Newton's Second Law

Mathematical Model - Newton's Second Law

- From $F = m \cdot a$:

$$F = \boxed{m\frac{\mathrm{d}v}{\mathrm{d}t} = m \cdot g - c_d \cdot v^2}$$

- Divide by m:

$$\underbrace{\frac{\mathrm{d}v}{\mathrm{d}t} = g - \frac{c_d}{m}v^2}_{\text{Ordinary Differential Equation!!!}}$$

  in terms of the differential rate of change in velocity.

- Initial condition:

$$v(0) = 0 \quad \text{(starts from rest)}$$

- **Problem definition**: Solve the velocity of the jumper in free fall as a function of time.
- **Why Numerical Methods?**
  - Real engineering problems often **do not have simple analytical solutions**!

Euler's Method (Numerical)

Euler's Method - The Simplest Numerical Approach

- Essential idea:

  Approximate continuous change with small discrete time steps Δt.

- Rewrite the formula of Bungee jumper velocity:

$$
\begin{aligned}
\underbrace{v_{t+\Delta t}}_{\text{new}} &= v_t + \Delta t \cdot \frac{\mathrm{d}v_t}{\mathrm{d}t} \\
&= \underbrace{v_t}_{\text{old}} + \underbrace{\Delta t}_{\text{time step size}} \cdot \underbrace{\left(g - \frac{c_d}{m} v_t^2\right)}_{\text{acceleration}}
\end{aligned}
$$

  from the ordinary differential equation:

$$
\frac{\mathrm{d}v}{\mathrm{d}t} = g - \frac{c_d}{m} v^2
$$

Euler's Method (Numerical)

Euler's Method - The Simplest Numerical Approach

- Formula of Bungee jumper velocity:

$$\underbrace{v_{t+\Delta t}}_{\text{new}} = \underbrace{v_t}_{\text{old}} + \underbrace{\Delta t}_{\text{time step size}} \cdot \underbrace{\left(g - \frac{c_d}{m} v_t^2\right)}_{\text{acceleration}}$$

- Computing **bungee jumper velocity** (step-by-step):
  - Start at $t = 0$ and $v = 0$
  - Compute **acceleration**:

$$a = g - \frac{c_d}{m} v_t^2$$

  - Update **velocity**:

$$v_{t+\Delta t} = v_t + \Delta t \cdot a$$

  - Increment time: $t = t + \Delta t$
  - Repeat!

A Real Case

Input. Mass $m = 50\,\text{kg}$, drag coefficient $c_d = 0.25$, and initial velocity $v_0 = 0$. (Given $\Delta t = 1\,\text{s}$)
Output. Bungee jumper velocity v_t.

- At time $t = 1$:

$$a = g - \frac{c_d}{m} v_0^2 = 9.81 - 0.005 \times 0^2 = 9.81$$
$$v_1 = v_0 + \Delta t \cdot a = 0 + 9.81 = \mathbf{9.81}$$

- At time $t = 2$:

$$a = g - \frac{c_d}{m} v_1^2 = 9.81 - 0.005 \times 9.81^2 = 9.33$$
$$v_2 = v_1 + \Delta t \cdot a = 9.81 + 1 \times 9.33 = \mathbf{19.14}$$

- At time $t = 3$

$$a = g - \frac{c_d}{m} v_2^2 = 9.81 - 0.005 \times 19.14^2 = 7.98$$
$$v_3 = v_2 + \Delta t \cdot a = 19.14 + 1 \times 7.98 = \mathbf{27.12}$$

- ...

The Basic Syntax of a `for` Loop in Python

Description.

- A `for` loop in Python is a control flow statement used to iterate over items of any sequence (such as a list, tuple, string, set, or dictionary) in the order that they appear.
- It is primarily used when you need to execute a block of code a specific, predetermined number of times or for each item in a collection.

The Basic Syntax of a `for` Loop in Python

Fibonacci Sequence.

- Definition: Given $f(1) = f(2) = 1$, the Fibonacci sequence takes the form of

$$f(n) = f(n-1) + f(n-2),\ n > 2$$

- Write down $f(3)$, $f(4)$, $f(5)$, $f(6)$, $f(7)$, $\cdots$ by yourself?

The Basic Syntax of a `for` Loop in Python

Fibonacci Sequence.

- Definition: Given $f(1) = f(2) = 1$, the Fibonacci sequence takes the form of

$$f(n) = f(n-1) + f(n-2),\ n > 2$$

- Write down $f(3)$, $f(4)$, $f(5)$, $f(6)$, $f(7)$, $\cdots$ by yourself?

$$f(3) = f(2) + f(1) = 2 \qquad f(4) = f(3) + f(2) = 3$$
$$f(5) = f(4) + f(3) = 5 \qquad f(6) = f(5) + f(4) = 8$$
$$f(7) = f(6) + f(5) = 13$$

- Python programming

```
import numpy as np

def fib(n):           # Input n>2
    f = np.zeros(n)
    f[0] = 1
    f[1] = 1
    for i in range(2, n):
        f[i] = f[i - 1] + f[i - 2]
    return f[n - 1]
```

Python Programming for Euler's Method

- **Python programming example**. Computing **bungee jumper velocity**:
  - Start at $t = 0$ and $v = 0$
  - Compute **acceleration**:

$$a = g - \frac{c_d}{m} v_t^2$$

  - Update **velocity**:

$$v_{t+\Delta t} = v_t + \Delta t \cdot a$$

  - Increment time: $t = t + \Delta t$
  - Repeat!

```python
import numpy as np

def euler(m, g, cd, v0, delta_t, time_steps):
    v = np.zeros(time_steps)                # Velocity
    v[0] = v0                               # Initial velocity
    for i in range(time_steps - 1):         # Repeat
        a = g - cd / m * (v[i] ** 2)        # Acceleration
        v[i + 1] = v[i] + delta_t * a       # Velocity
    return v
```

A Real Case

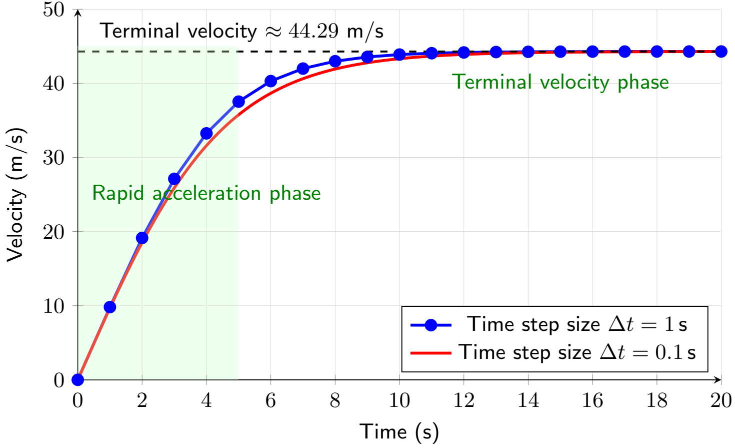
- Mass: $m = 50\,\text{kg}$
- Gravitational acceleration: $g = 9.81\,\text{m/s}^2$
- Drag coefficient: $c_d = 0.25$

```python
import numpy as np

# Parameters
m = 50                  # Mass (kg)
g = 9.81                # Gravity (m/s^2)
cd = 0.25               # Drag coefficient
v0 = 0                  # Initial velocity

# Time setup
delta_t = 1             # Time step size
t_end = 20              # Total time
time_steps = int(t_end / delta_t) + 1

# Euler's method
t = np.linspace(0, t_end, time_steps)
v = euler(m, g, cd, v0, delta_t, time_steps)
```

Velocity vs. Time

Bungee jumper **velocity vs. time** (w/ air resistance)

- Comparison between $\Delta t = 1\,\text{s}$ and $\Delta t = 0.1\,\text{s}$



Bungee Jumping

Quizzes Now!

- **Today's participation**: Please check out

  "`Class Participation Quiz 3`"

  Time slot: **3:00PM – 3:30PM**

  on Canvas.

- Online engagement (graded quizzes)

  "`Quiz 3`" (14 questions)

  Deadline: **11:59PM, January 21, 2026**

  on Canvas.