Tutorial

A Brief Introduction to Computational Tools

Serhat Beyenir

28 October 2025

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Full text available at https://creativecommons.org/licenses/by-sa/4.0/.

Contents

List of Figures

List of Tables

Preface

A Brief Introduction to Computational Tools presents a collection of tutorials based on lecture notes from classes, designed to give learners clear and essential insights into key topics. No previous programming experience is required. Each tutorial guides you step by step through the concepts with hands-on examples.

Chapter 1

Python Tutorial

This tutorial introduces Python programming, covering basic concepts with examples to illustrate key points. We will start by using Python as a calculator, then explore variables, functions, and control flow.

1.1 Requirements

To follow this tutorial, you must have Python (version 3.10 or later) installed on your computer. Python is available for Windows, macOS, and Linux. Additionally, ensure you have a text editor or an Integrated Development Environment (IDE) to write Python code. We recommend Positron, a user-friendly IDE with a built-in terminal for running Python scripts, though other editors like VS Code or PyCharm are also suitable.

Alternatively, you can use a web-based environment like python-fiddle.com

1.2 Basic Syntax

Python uses indentation (typically four spaces) to define code blocks. A colon (`:`) introduces a block, and statements within the block must be indented consistently. Python is case-sensitive, so `Variable` and `variable` are distinct identifiers. Statements typically end with a newline, but you can use a backslash (\) to continue a statement across multiple lines.

```
total = 1 + 2 + 3 + \
        4 + 5
print(total)  # Output: 15
```

Basic syntax rules:

- Comments start with `#` and extend to the end of the line.

- Strings can be enclosed in single quotes (`'`), double quotes (`"`), or triple quotes (`'''` or `"""`) for multi-line strings.
- Python is case-sensitive, so `Variable` and `variable` are different identifiers.

1.3 The `print()` Function

The `print()` function displays output in Python.

```python
name = "Rudolf Diesel"
year = 1858
print(f"{name} was born in {year}.")
```

Output: `Rudolf Diesel was born in 1858.`

1.4 Formatting in `print()`

The following table illustrates common f-string formatting options for the `print()` function:

Format	Code	Example	Output
Round to 2 decimals	`f"{x:.2f}"`	`print(f"{3.14159:.2f}")`	`3.14`
Round to whole number	`f"{x:.0f}"`	`print(f"{3.9:.0f}")`	`4`
Thousands separator	`f"{x:,.2f}"`	`print(f"{1234567.89:,.2f}")`	`1,234,567.89`
Percentage	`f"{x:.1%}"`	`print(f"{0.756:.1%}")`	`75.6%`
Currency style	`f"${x:,.2f}"`	`print(f"${1234.5:,.2f}")`	`$1,234.50`

Note: The currency symbol (e.g., `$`) can be modified for other currencies (e.g., `€`, `£`) based on the desired locale.

1.5 Variables and Data Types

Variables store data and are assigned values using the `=` operator.

```python
x = 10
y = 3.14
name = "Rudolph"
```

Python has several built-in data types, including:

- Integers (`int`): Whole numbers, e.g., `10`, `-5`
- Floating-point numbers (`float`): Decimal numbers, e.g., `3.14`, `-0.001`

- Strings (`str`): Text, e.g., `"Hello"`, `'World'`
- Booleans (`bool`): `True` or `False`

1.5.1 Arithmetic Operations

```
a = 10
b = 3
print(a + b)  # Addition: 13
print(a - b)  # Subtraction: 7
print(a * b)  # Multiplication: 30
print(a / b)  # Division: 3.3333...
print(a // b) # Integer Division: 3
print(a ** b) # Exponentiation: 1000
```

1.5.2 String Operations

```
first_name = "Rudolph"
last_name = "Diesel"
full_name = first_name + " " + last_name  # Concatenation using +
print(full_name)  # Output: Rudolph Diesel
print(f"{first_name} {last_name}")  # Concatenation using f-string
print(full_name * 2)  # Repetition: Rudolph DieselRudolph Diesel
print(full_name.upper())  # Uppercase: RUDOLPH DIESEL
```

Note: String repetition (`*`) concatenates the string multiple times without spaces. For example, `full_name * 2` produces `Rudolph DieselRudolph Diesel`.

1.6 Python as a Calculator in Interactive Mode

Python's interactive mode allows you to enter commands and see results immediately, ideal for quick calculations. To start, open a terminal (on macOS, Linux, or Windows) and type:

```
python3  # Use 'python' on Windows if 'python3' is not recognized
```

You should see the Python prompt:

```
>>>
```

Enter expressions and press **Enter** to see results:

```
2 + 3  # Output: 5
7 - 4  # Output: 3
6 * 9  # Output: 54
8 / 2  # Output: 4.0
8 // 2 # Output: 4
2 ** 3 # Output: 8
```

1.6.1 Parentheses for Grouping

```
(2 + 3) * 4  # Output: 20
2 + (3 * 4)  # Output: 14
```

1.6.2 Variables

```
x = 10
y = 3
x / y  # Output: 3.3333333333333335
```

1.6.3 Exiting Interactive Mode

To exit, type:

```
exit()
```

Alternatively, use: - **Ctrl+D** (macOS/Linux) - **Ctrl+Z** then Enter (Windows)

1.7 Control Flow

Control flow statements direct the execution of code based on conditions.

1.7.1 Conditional Statements

Conditional statements allow you to execute different code blocks based on specific conditions. Python provides three keywords for this purpose:

- **`if`**: Evaluates a condition and executes its code block if the condition is `True`.
- **`elif`**: Short for "else if," it checks an additional condition if the preceding `if` or `elif` conditions are `False`. You can use multiple `elif` statements to test multiple conditions sequentially, and Python will execute the first `True` condition's block, skipping the rest.
- **`else`**: Executes a code block if none of the preceding `if` or `elif` conditions are `True`. It serves as a fallback and does not require a condition.

The following example uses age to categorize a person as a Minor, Adult, or Senior, demonstrating how `if`, `elif`, and `else` work together.

```
# Categorize a person based on their age
age = 19
if age < 18:
    print("Minor")
elif age <= 64:
    print("Adult")
```

```
else:
    print("Senior")
```

Output: `Adult`

1.7.2 For Loop

A `for` loop iterates over a sequence (e.g., list or string).

```
components = ["piston", "liner", "connecting rod"]
for component in components:
    print(component)
```

Output:

```
piston
liner
connecting rod
```

1.7.3 While Loop

A `while` loop executes as long as a condition is true. Ensure the condition eventually becomes false to avoid infinite loops.

```
count = 0
while count <= 5:
    print(count)
    count += 1
```

Output:

```
0
1
2
3
4
5
```

1.8 Functions

1.8.1 The `def` Keyword

Functions are reusable code blocks defined using the `def` keyword. They can include default parameters for optional arguments.

```
def add(a, b=0):
    return a + b
print(add(5))      # Output: 5
```

```
print(add(5, 3))   # Output: 8

def multiply(*args):
    result = 1
    for num in args:
        result *= num
    return result
print(multiply(2, 3, 4))  # Output: 24
```

1.8.2 The lambda Keyword

The `lambda` keyword creates anonymous functions for short, one-off operations, often used in functional programming.

```
celsius_to_fahrenheit = lambda c: (c * 9 / 5) + 32
print(celsius_to_fahrenheit(25))  # Output: 77.0
```

1.9 The math Module

The `math` module provides mathematical functions and constants.

```
import math
print(math.sqrt(16))  # Output: 4.0
print(math.pi)        # Output: 3.141592653589793
```

```
import math
angle = math.pi / 4  # 45 degrees in radians
print(math.sin(angle))  # Output: 0.7071067811865475 (approximately √2/2)
print(math.cos(angle))  # Output: 0.7071067811865476 (approximately √2/2)
print(math.tan(angle))  # Output: 1.0
```

Note: Floating-point arithmetic may result in small precision differences, as seen in the `sin` and `cos` outputs.

```
import math
print(math.log(10))       # Natural logarithm of 10: 2.302585092994046
print(math.log(100, 10))  # Logarithm of 100 with base 10: 2.0
```

1.9.1 Converting Between Radians and Degrees

The `math` module provides `math.radians()` to convert degrees to radians and `math.degrees()` to convert radians to degrees, which is useful for trigonometric calculations.

```
import math
degrees = 180
radians = math.radians(degrees)
```

```
print(f"{degrees} degrees is {radians:.3f} radians")  # Output: 180 degrees is 3.142 radians

radians = math.pi / 2
degrees = math.degrees(radians)
print(f"{radians:.3f} radians is {degrees:.1f} degrees")  # Output: 1.571 radians is 90.0 degrees
```

1.10 Writing Python Scripts

Write Python code in a `.py` file and run it as a script. Create a file named `script.py`:

```
# script.py
import math
print("Square root of 16 is:", math.sqrt(16))
print("Value of pi is:", math.pi)
print("Sine of 90 degrees is:", math.sin(math.pi / 2))
print("Natural logarithm of 10 is:", math.log(10))
print("Logarithm of 100 with base 10 is:", math.log(100, 10))
```

To run the script, open a terminal, navigate to the directory containing `script.py` using the `cd` command (e.g., `cd /path/to/directory`), and type:

```
python3 script.py  # or python script.py on Windows
```

Output:

```
Square root of 16 is: 4.0
Value of pi is: 3.141592653589793
Sine of 90 degrees is: 1.0
Natural logarithm of 10 is: 2.302585092994046
Logarithm of 100 with base 10 is: 2.0
```

1.11 Summary

This tutorial covered Python basics, including syntax, variables, data types, operations, control flow, and functions. Python's rich ecosystem includes libraries like:

- **NumPy**: For numerical computations and array manipulations.
- **Matplotlib**: For data visualization and plotting.
- **Pandas**: For data manipulation and analysis with tabular data structures.
- **Pint**: For handling physical quantities and performing unit conversions.

You can explore these libraries to enhance your Python programming skills further. For example installing them can be done using `pip`:

```
pip install numpy matplotlib pandas pint
```

`pip` is Python's package manager for installing and managing additional libraries.

Chapter 2

SI Units

The International System of Units (SI) is the globally accepted standard for measurement. Established to provide a consistent framework for scientific and technical measurements, SI units facilitate clear communication and data comparison across various fields and countries. The system is based on seven fundamental units: the meter for length, the kilogram for mass, the second for time, the ampere for electric current, the kelvin for temperature, the mole for substance, and the candela for luminous intensity.

Table 2.1: Base SI units.

Physical Quantity	SI Base Unit	Symbol
Length	Meter	m
Mass	Kilogram	kg
Time	Second	s
Electric Current	Ampere	A
Temperature	Kelvin	K
Amount of Substance	Mole	mol
Luminous Intensity	Candela	cd

Table 2.2: Derived SI units.

Physical Quantity	Derived SI Unit	Symbol
Area	Square meter	m^2
Volume	Cubic meter	m^3
Speed	Meter per second	m/s
Acceleration	Meter per second squared	m/s^2
Force	Newton	N
Pressure	Pascal	Pa
Energy	Joule	J
Power	Watt	W
Electric Charge	Coulomb	C
Electric Potential	Volt	V
Resistance	Ohm	Ω
Capacitance	Farad	F
Frequency	Hertz	Hz
Luminous Flux	Lumen	lm
Illuminance	Lux	lx
Specific Energy	Joule per kilogram	J/kg
Specific Heat Capacity	Joule per kilogram Kelvin	$J/(kg \cdot K)$

Table 2.3: Common multiples and submultiples for SI units.

Factor	Prefix	Symbol
10^9	giga	G
10^6	mega	M
10^3	kilo	k
10^2	hecto	h
10^1	deca	da
10^{-1}	deci	d
10^{-2}	centi	c
10^{-3}	milli	m
10^{-6}	micro	µ

2.1 Unity Fraction

The **unity fraction** method, or **unit conversion using unity fractions**, is a systematic way to convert one unit of measurement into another. This method relies on multiplying by fractions that are equal to one, where the numerator and the denominator represent the same quantity in different units. Since any number multiplied by one remains the same, unity fractions allow for seamless conversion without changing the value.

The principle of unity fractions is based on:

1. **Setting up equal values**: Write a fraction where the numerator and denominator are equivalent values in different units, so the fraction equals one. For example, $\frac{1km}{1000m}$ is a unity fraction because 1 km equals 1000 m.
2. **Multiplying by unity fractions**: Multiply the initial quantity by the unity fraction(s) so that the undesired units cancel out, leaving only the desired units.

2.1.1 Classwork

Example 2.1. Suppose we want to convert 5 kilometers to meters.

1. Start with 5 kilometers:
$$5\,\mathrm{km}$$
2. Multiply by a unity fraction that cancels kilometers and introduces meters. We use $(\frac{1000\,\mathrm{m}}{1\,\mathrm{km}}), since\, 1\,\mathrm{km} = 1000\,\mathrm{m}$:
$$5\,\mathrm{km} \times \frac{1000\,\mathrm{m}}{1\,\mathrm{km}} = 5000\,\mathrm{m}$$
3. The kilometers km cancel out, leaving us with meters m:

$$5\,\mathrm{km} = 5000\,\mathrm{m}$$

This step-by-step approach illustrates how the unity fraction cancels the undesired units and achieves the correct result in meters.

Unity fractions can be extended by using multiple conversion steps. For example, converting hours to seconds would require two unity fractions: one to convert hours to minutes and another to convert minutes to seconds. This approach ensures accuracy and is widely used in science, engineering, and other fields that require precise unit conversions.

Example 2.2. Convert $15\,\mathrm{m/s}$ to $\mathrm{km/h}$.

1. Start with $15\,\mathrm{m/s}$.
2. To convert meters to kilometers, multiply by $\frac{1\,\mathrm{km}}{1000\,\mathrm{m}}$.
3. To convert seconds to hours, multiply by $\frac{3600\,\mathrm{s}}{1\,\mathrm{h}}$.

$$15\,\mathrm{m/s} \times \frac{1\,\mathrm{km}}{1000\,\mathrm{m}} \times \frac{3600\,\mathrm{s}}{1\,\mathrm{h}} = 54\,\mathrm{km/h}$$

The meters and seconds cancel out, leaving kilometers per hour: $54\,\mathrm{km/h}$.

2.1.2 Self-Test

Instructions:

1. Use unity fraction to convert between derived SI units.
2. Show each step of your work to ensure accuracy.
3. Simplify your answers and include correct units.

---

1. **Speed**
   Convert $72\,\mathrm{km/h}$ to $\mathrm{m/s}$.
2. **Force**
   Convert $980\,\mathrm{N}$ (newtons) to $\mathrm{kg} \cdot \mathrm{m/s}^2$.
3. **Energy**
   Convert $2500\,\mathrm{J}$ (joules) to kJ.
4. **Power**
   Convert $1500\,\mathrm{W}$ (watts) to kW.
5. **Pressure**
   Convert $101325\,\mathrm{Pa}$ (pascals) to kPa.
6. **Volume Flow Rate**
   Convert $3\,\mathrm{m}^3/\mathrm{min}$ to $\mathrm{L/s}$.

7. **Density**
   Convert $1000\,\mathrm{kg/m^3}$ to $\mathrm{g/cm^3}$.
8. **Acceleration**
   Convert $9.8\,\mathrm{m/s^2}$ to $\mathrm{cm/s^2}$.
9. **Torque**
   Convert $50\,\mathrm{N\cdot m}$ to $\mathrm{kN\cdot cm}$.
10. **Frequency**
    Convert 500 Hz (hertz) to kHz.
11. **Work to Energy Conversion**
    A force of 20 N moves an object 500 cm. Convert the work done to joules.
12. **Kinetic Energy Conversion**
    Calculate the kinetic energy in kilojoules of a 1500 kg car moving at 72 km/h.
13. **Power to Energy Conversion**
    A machine operates at 2 kW for 3 hours. Convert the energy used to megajoules.
14. **Pressure to Force Conversion**
    Convert a pressure of 200 kPa applied to an area of $0.5\,\mathrm{m^2}$ to force in newtons.
15. **Density to Mass Conversion**
    Convert $0.8\,\mathrm{g/cm^3}$ for an object with a volume of $250\,\mathrm{cm^3}$ to mass in grams.

2.1.3 Answer Key

1. $72\,\mathrm{km/h} = 20\,\mathrm{m/s}$
2. $980\,\mathrm{N} = 980\,\mathrm{kg\cdot m/s^2}$
3. $2500\,\mathrm{J} = 2.5\,\mathrm{kJ}$
4. $1500\,\mathrm{W} = 1.5\,\mathrm{kW}$
5. $101325\,\mathrm{Pa} = 101.325\,\mathrm{kPa}$
6. $3\,\mathrm{m^3/min} = 50\,\mathrm{L/s}$
7. $1000\,\mathrm{kg/m^3} = 1\,\mathrm{g/cm^3}$
8. $9.8\,\mathrm{m/s^2} = 980\,\mathrm{cm/s^2}$
9. $50\,\mathrm{N\cdot m} = 5\,\mathrm{kN\cdot cm}$
10. $500\,\mathrm{Hz} = 0.5\,\mathrm{kHz}$
11. $20\,\mathrm{N} \times 5\,\mathrm{m} = 100\,\mathrm{J}$
12. Kinetic energy $= 1500\,\mathrm{kg} \times (20\,\mathrm{m/s})^2\,/2 = 300\,\mathrm{kJ}$
13. $2\,\mathrm{kW} \times 3\,\mathrm{hours} = 21.6\,\mathrm{MJ}$
14. $200\,\mathrm{kPa} \times 0.5\,\mathrm{m^2} = 100{,}000\,\mathrm{N}$
15. $0.8\,\mathrm{g/cm^3} \times 250\,\mathrm{cm^3} = 200\,\mathrm{g}$

2.2 Condenser Vacuum

Condenser vacuum gauge reads 715 mmHg when barometer stands at 757 mmHg. State the absolute pressure in kN/m² and bar.

2.2.1 Given Data

$$P_{atm} = 757 \text{ mmHg}, \quad P_{vac} = 715 \text{ mmHg}$$

2.2.2 Absolute Pressure in mmHg

$$P_{abs} = P_{atm} - P_{vac} = 757 - 715 = 42 \text{ mmHg}$$

2.2.3 Convert mmHg → kN/m²

$$P = \rho g h = 13{,}600 \cdot 9.81 \cdot 0.001 = 133.416 \text{ Pa} = 0.133416 \text{ kN/m}^2$$

$$P_{abs} = 42 \cdot 0.133416 = 5.6034 \text{ kN/m}^2$$

2.2.4 Convert kN/m² → bar

$$P_{abs} = \frac{5.6034}{100} = 0.056 \text{ bar}$$

2.2.5 Final Answers

$$\boxed{P_{abs} = 42 \text{ mmHg} = 5.6034 \text{ kN/m}^2 = 0.056 \text{ bar}}$$

2.2.6 Code

```
P_atm_mmHg = 757
P_vac_mmHg = 715
MMHG_TO_KN_M2 = 0.133416
KNM2_TO_BAR = 1 / 100
P_abs_mmHg = P_atm_mmHg - P_vac_mmHg
P_abs_kNm2 = P_abs_mmHg * MMHG_TO_KN_M2
P_abs_bar = P_abs_kNm2 * KNM2_TO_BAR
print(f"Absolute Pressure = {P_abs_mmHg:.2f} mmHg")
print(f"Absolute Pressure = {P_abs_kNm2:.3f} kN/m²")
print(f"Absolute Pressure = {P_abs_bar:.4f} bar")
```

2.3 Oil Flow in Tubes

Oil flows full bore at a velocity of $V = 2$ m/s through 16 tubes of diameter $d = 30$ mm. Density of oil: $\rho = 0.85$ g/mL. Find **volume flow rate** (L/s) and **mass flow rate** (kg/min).

2.3.1 Cross-sectional area of one tube

$$A = \pi \frac{d^2}{4} = \pi \frac{0.03^2}{4} \approx 7.0686 \times 10^{-4}\ \mathrm{m}^2$$

2.3.2 Total area and volume flow rate

$$A_{\text{total}} = 16 \cdot 7.0686 \times 10^{-4} \approx 0.01131\ \mathrm{m}^2$$

$$\dot{v} = A_{\text{total}} \cdot V \approx 0.02262\ \mathrm{m}^3/\mathrm{s} \approx 22.62\ \mathrm{L/s}$$

2.3.3 Mass flow rate

$$\dot{m} = \rho \cdot \dot{v} = 850 \cdot 0.02262 \approx 19.227\ \mathrm{kg/s} \approx 1153.6\ \mathrm{kg/min}$$

2.3.4 Final Answers

$$\text{Volume flow rate: } \dot{v} \approx 22.6\ \mathrm{L/s}$$

$$\text{Mass flow rate: } \dot{m} \approx 1154\ \mathrm{kg/min}$$

2.3.5 Code

```
import math
v = 2.0
N = 16
d = 0.03
rho = 0.85 * 1000
A = math.pi * d**2 / 4
A_total = N * A
v_dot_m3_s = A_total * v
v_dot_L_s = Q_m3_s * 1000
m_dot_kg_s = rho * Q_m3_s
m_dot_kg_min = m_dot_kg_s * 60
print(f"Volume flow rate: {v_dot_L_s:.2f} L/s")
print(f"Mass flow rate: {m_dot_kg_min:.2f} kg/min")
```

2.4 Gauge Pressure

An oil of specific gravity (relative density) SG = 0.8 is contained in a vessel to a depth of $h = 2$ m. Find the **gauge pressure** at this depth in kPa.

2.4.1 Gauge Pressure

$$P_g = \rho g h$$

where

ρ = density of fluid (kg/m^3)

g= acceleration due to gravity (9.81 m/s^2)

h = depth (m)

2.4.2 Compute the density of oil using specific gravity

Specific gravity is defined as

$$\mathrm{SG} = \frac{\rho_{\text{oil}}}{\rho_{\text{water}}}$$

where $\rho_{\text{water}} = 1000\ \ \text{kg/m}^3$. Thus,

$$\rho_{\text{oil}} = \mathrm{SG} \times \rho_{\text{water}} = 0.8 \times 1000 = 800\ \text{kg/m}^3$$

2.4.3 Compute the gauge pressure

$$P_g = \rho g h = 800 \times 9.81 \times 2$$

$$P_g = 15696\ \text{Pa} \approx 15.7\ \text{kPa}$$

2.4.4 Answer

The **gauge pressure** at a depth of 2 m in the oil is:

$$\boxed{15.7\ \text{kPa}}$$

2.4.5 Code

```
# Gauge Pressure Calculation for Oil

# Given data
specific_gravity = 0.8  # SG of oil
depth_m = 2.0           # depth in meters
g = 9.81                # acceleration due to gravity in m/s²
rho_water = 1000        # density of water in kg/m³

# Compute density of oil
rho_oil = specific_gravity * rho_water

# Compute gauge pressure (Pa)
P_g_Pa = rho_oil * g * depth_m

# Convert to kPa
P_g_kPa = P_g_Pa / 1000

# Print results
print(f"Density of oil: {rho_oil:.1f} kg/m³")
print(f"Gauge pressure at {depth_m} m depth: {P_g_Pa:.1f} Pa ({P_g_kPa:.2f} kPa)")
```

2.5 Absolute Pressure from Manometer Reading

A water manometer shows a pressure in a vessel of 400 mm **below atmospheric pressure**. The atmospheric pressure is measured as 763 mmHg. Determine the **absolute pressure** in the vessel in kPa.

2.5.1 Relationship between absolute and gauge pressure

$$P_{\text{abs}} = P_{\text{atm}} + P_{\text{gauge}}$$

Since the manometer shows a pressure **below atmospheric**, the gauge pressure is negative:

$$P_{\text{gauge}} = -\rho_{\text{water}} g h$$

2.5.2 Convert atmospheric pressure to Pa using

$$P = \rho g h = 13{,}600 \cdot 9.81 \cdot 0.001 = 133.416 \text{ Pa}$$

So

$$P_{\text{atm}} = 763 \times 133.416 \approx 101{,}801 \text{ Pa} \approx 101.8 \text{ kPa}$$

2.5.3 Compute gauge pressure

Water column height:

$$h = 400 \text{ mm} = 0.4 \text{ m}$$

Density of water: $rho_{\text{water}} = 1000 \text{ kg/m}^3$, $g = 9.81 \text{ m/s}^2$

$$P_{\text{gauge}} = -\rho g h = -1000 \times 9.81 \times 0.4$$

$$P_{\text{gauge}} = -3924 \text{ Pa} \approx -3.92 \text{ kPa}$$

2.5.4 Compute absolute pressure

$$P_{\text{abs}} = P_{\text{atm}} + P_{\text{gauge}} \approx 101.8 - 3.92 \approx 97.9 \text{ kPa}$$

2.5.5 Answer

The **absolute pressure** in the vessel is:

$$\boxed{97.9 \text{ kPa}}$$

2.5.6 Code

```
# Absolute Pressure Calculation from Water Manometer

# Given data
h_mm = 400               # manometer reading in mm (below atmospheric)
atm_mmHg = 763           # atmospheric pressure in mmHg
rho_water = 1000         # density of water in kg/m³
g = 9.81                 # gravity in m/s²
mmHg_to_Pa = 133.416     # conversion factor

# Convert manometer height to meters
h_m = h_mm / 1000

# Convert atmospheric pressure to Pa
P_atm_Pa = atm_mmHg * mmHg_to_Pa

# Gauge pressure (negative because below atmospheric)
P_gauge_Pa = - rho_water * g * h_m

# Absolute pressure
```

```
P_abs_Pa = P_atm_Pa + P_gauge_Pa

# Convert to kPa
P_abs_kPa = P_abs_Pa / 1000

# Print results
print(f"Atmospheric pressure: {P_atm_Pa:.1f} Pa ({P_atm_Pa/1000:.1f} kPa)")
print(f"Gauge pressure: {P_gauge_Pa:.1f} Pa ({P_gauge_Pa/1000:.2f} kPa)")
print(f"Absolute pressure in the vessel: {P_abs_Pa:.1f} Pa ({P_abs_kPa:.2f} kPa)")
```

Chapter 3

Heat and Work

3.1 Heat Required to Heat Steel

A steel block of mass $m = 5$ kg and specific heat capacity $c = 480$ J/kg $\cdot$ K is heated from $T_1 = 15°\text{C}$ to $T_2 = 100°\text{C}$. Determine the **heat required**.

3.1.1 The formula for heat

$$Q = mc\Delta T \tag{3.1}$$

where

$\Delta T = T_2 - T_1$ is the temperature change.

3.1.2 Compute the temperature change

$$\Delta T = T_2 - T_1 = 100 - 15 = 85 \text{ K}$$

3.1.3 Compute the heat required

$$Q = mc\Delta T = 5 \times 480 \times 85$$

$$Q = 204,000 \text{ J}$$

3.1.4 Answer

The **heat required** to raise the temperature of the steel is:

$$\boxed{204 \text{ kJ}}$$

3.1.5 Code

```
# Heat Required to Heat Steel

# Given data
mass = 5            # kg
specific_heat = 480 # J/kg·K
T_initial = 15      # °C
T_final = 100       # °C

# Temperature change
delta_T = T_final - T_initial

# Heat required (in J)
Q_J = mass * specific_heat * delta_T

# Convert to kJ
Q_kJ = Q_J / 1000

# Print results
print(f"Temperature change: {delta_T} K")
print(f"Heat required: {Q_J:.0f} J ({Q_kJ:.0f} kJ)")
```

```
# Interactive Heat Calculation

# Get user input
mass = float(input("Enter the mass of the object (kg): "))
specific_heat = float(input("Enter the specific heat capacity (J/kg·K): "))
T_initial = float(input("Enter the initial temperature (°C): "))
T_final = float(input("Enter the final temperature (°C): "))

# Calculate temperature change
delta_T = T_final - T_initial

# Calculate heat required
Q_J = mass * specific_heat * delta_T
Q_kJ = Q_J / 1000

# Display results
print("\nCalculation Results:")
print(f"Temperature change: {delta_T:.2f} K")
print(f"Heat required: {Q_J:.2f} J ({Q_kJ:.2f} kJ)")
```

3.2 Finding Specific Heat Capacity

A liquid of mass $m = 4$ kg is heated from $T_1 = 15°\text{C}$ to $T_2 = 100°\text{C}$. The heat supplied is $Q = 714$ kJ. Determine the **specific heat capacity** c of the liquid.

3.2.1 Recall the formula for heat

$$Q = mc\Delta T$$

where $\Delta T = T_2 - T_1$.

3.2.2 Convert heat to joules

$$Q = 714 \text{ kJ} = 714 \times 1000 = 714,000 \text{ J}$$

3.2.3 Compute temperature change

$$\Delta T = T_2 - T_1 = 100 - 15 = 85 \text{ K}$$

3.2.4 Solve for specific heat capacity

$$c = \frac{Q}{m\Delta T} = \frac{714,000}{4 \times 85}$$

$$c = \frac{714,000}{340} \approx 2100 \text{ J/kg} \cdot \text{K}$$

3.2.5 Answer

The **specific heat capacity** of the liquid is:

$$\boxed{c \approx 2100 \text{ J/kg} \cdot \text{K}}$$

3.2.6 Code

```
# Specific Heat Capacity Calculation

# Given data
mass = 4          # kg
T_initial = 15    # °C
T_final = 100     # °C
Q_kJ = 714        # heat supplied in kJ

# Convert heat to joules
Q_J = Q_kJ * 1000
```

```
# Temperature change
delta_T = T_final - T_initial

# Calculate specific heat capacity
c = Q_J / (mass * delta_T)

# Print results
print(f"Temperature change: {delta_T} K")
print(f"Specific heat capacity: {c:.0f} J/kg·K")
```

```
# Interactive Specific Heat Capacity Calculator

# Get user input
mass = float(input("Enter the mass of the liquid (kg): "))
T_initial = float(input("Enter the initial temperature (°C): "))
T_final = float(input("Enter the final temperature (°C): "))
Q_kJ = float(input("Enter the heat supplied (kJ): "))

# Convert heat to joules
Q_J = Q_kJ * 1000

# Temperature change
delta_T = T_final - T_initial

# Calculate specific heat capacity
c = Q_J / (mass * delta_T)

# Display results
print("\nCalculation Results:")
print(f"Temperature change: {delta_T:.2f} K")
print(f"Specific heat capacity: {c:.2f} J/kg·K")
```

3.3 Work Done by Fluid in Cylinder

A fluid in a cylinder is at pressure $P = 700$ kPa. It is **expanded at constant pressure** from a volume of $V_1 = 0.28$ m^3 to $V_2 = 1.68$ m^3. Determine the **work done**.

3.3.1 The formula for work done at constant pressure

$$W = P\Delta V \tag{3.2}$$

where

$$\Delta V = V_2 - V_1$$

3.3.2 Compute the change in volume

$$\Delta V = 1.68 - 0.28 = 1.40 \text{ m}^3$$

3.3.3 Convert pressure to Pa

$$P = 700 \text{ kPa} = 700 \times 10^3 \text{ Pa} = 700,000 \text{ Pa}$$

3.3.4 Compute the work done

$$W = P\Delta V = 700,000 \times 1.40$$

$$W = 980,000 \text{ J} \approx 980 \text{ kJ}$$

3.3.5 Answer

The **work done** by the fluid during expansion is:

$$\boxed{980 \text{ kJ}}$$

3.3.6 Code

```
# Work Done by Fluid at Constant Pressure

# Given data
P_kPa = 700           # pressure in kPa
V1 = 0.28             # initial volume in m³
V2 = 1.68             # final volume in m³

# Convert pressure to Pa
P_Pa = P_kPa * 1000

# Compute change in volume
delta_V = V2 - V1

# Compute work done (J)
W_J = P_Pa * delta_V

# Convert to kJ
W_kJ = W_J / 1000

# Print results
print(f"Pressure: {P_Pa} Pa")
print(f"Change in volume: {delta_V:.2f} m³")
print(f"Work done: {W_J:.0f} J ({W_kJ:.0f} kJ)")
```

```
# Interactive Work Done Calculator

# Get user input
P_kPa = float(input("Enter the constant pressure (kPa): "))
V1 = float(input("Enter the initial volume (m³): "))
V2 = float(input("Enter the final volume (m³): "))

# Convert pressure to Pa
P_Pa = P_kPa * 1000

# Compute change in volume
delta_V = V2 - V1

# Compute work done (J)
W_J = P_Pa * delta_V

# Convert to kJ
W_kJ = W_J / 1000

# Display results
print("\nCalculation Results:")
print(f"Pressure: {P_Pa:.0f} Pa ({P_kPa:.0f} kPa)")
print(f"Change in volume: {delta_V:.2f} m³")
print(f"Work done: {W_J:.0f} J ({W_kJ:.2f} kJ)")
```

3.4 Enthalpy Calculation

120 kg of steam at $p = 1000$ kPa and $T = 200°$C has internal energy $u = 2623$ kJ/kg and specific volume ($v = 0.2061\ \ \mathrm{m}^3/\mathrm{kg}$ Calculate the specific enthalpy (h) and the total enthalpy for the 120-kg mass.

Given for steam:

- mass $m = 120\ \ \mathrm{kg}$
- pressure $P = 1000\ \ \mathrm{kPa}$
- temperature $T = 200°\mathrm{C}$ (not needed for calculation)
- internal energy $u = 2623\ \ \mathrm{kJ/kg}$
- specific volume $v = 0.2061\ \ \mathrm{m}^3/\mathrm{kg}$

The specific enthalpy is

$$h = u + Pv \tag{3.3}$$

Note: with (P) in kPa and (v) in m(^3)/kg, the product (Pv) has units kJ/kg (since (1 kPa m^3 = 1 kJ)).

Compute (Pv):

$$Pv = 1000 \times 0.2061 = 206.1 \text{ kJ/kg}$$

So the specific enthalpy is:

$$h = 2623 + 206.1 = 2829.1 \text{ kJ/kg}$$

Total enthalpy for the 120 kg of steam:

$$H = m\,h = 120 \times 2829.1 = 339{,}492 \text{ kJ}$$

3.4.1 Answers

- Specific enthalpy: $\boxed{h = 2829.1 \text{ kJ/kg}}$
- Total enthalpy for 120 kg: $\boxed{H = 339{,}492 \text{ kJ}}$

3.4.2 Code

```python
# Given data
m = 120          # kg
P = 1000         # kPa
u = 2623         # kJ/kg
v = 0.2061       # m³/kg

# Enthalpy calculation
h = u + P * v          # kJ/kg
H_total = m * h        # kJ

print(f"Specific enthalpy h = {h:.1f} kJ/kg")
print(f"Total enthalpy H = {H_total:,.0f} kJ")
```

Chapter 4

Thermal Expansion

Thermal expansion is the tendency of materials to change their shape, area, and volume in response to a change in temperature. When most substances are heated, their particles move more vigorously and tend to occupy more space, leading to an increase in dimensions. Conversely, when substances are cooled, they generally contract. This phenomenon occurs in solids, liquids, and gases, although the degree and nature of expansion vary depending on the material's state and properties.

4.1 Linear Expansion

This occurs along a specific dimension or direction, primarily in long, narrow objects (like rods or beams). When the temperature of a solid object increases, its length expands by an amount proportional to its original length and the temperature change. The equation for linear expansion is:

$$\Delta L = \alpha L_0 \Delta T \tag{4.1}$$

where:

- ΔL is the change in length,
- α is the coefficient of linear expansion (unique to each material),
- L_0 is the original length, and
- ΔT is the temperature change.

4.2 Superficial Expansion

Applicable to two-dimensional surfaces, such as sheets or plates. Here, both length and width expand, leading to an increase in surface area. The formula for area expansion is:

$$\Delta A = 2\alpha A_0 \Delta T \tag{4.2}$$

where:

- ΔA is the change in area,
- A_0 is the initial area, and
- ΔT is the temperature change.

4.3 Volumetric Expansion

Relevant for three-dimensional objects (like solids, liquids, and gases). The volume of an object expands with temperature, especially in fluids where this effect is more pronounced. The formula is:

$$\Delta V = \beta V_0 \Delta T \tag{4.3}$$

where:

- ΔV is the change in volume,
- β is the coefficient of volumetric expansion, which is approximately three times the linear expansion coefficient for isotropic solids,
- V_0 is the initial volume, and
- ΔT is the temperature change.

Chapter 5

Heat Transfer

5.1 Conduction

This method transfers heat through direct contact, as faster-vibrating molecules pass energy to slower ones. In solids, heat moves from molecule to molecule, and between objects in contact, it flows from the hotter to the cooler one until temperatures equalize.

An example of conduction is an iron bar having one end in contact with a flame. The other end will soon become hot due to the conduction of heat from molecule to molecule through the iron.

$$Q = \frac{kAt\Delta T}{s} \tag{5.1}$$

5.2 Convection

Convection transfers heat through the movement of fluids (liquids or gases). When a fluid is heated, it expands, becomes less dense, and rises, while cooler, denser fluid moves in to replace it—creating a convection current that spreads heat.

Natural convection occurs without mechanical aid, while forced convection uses devices like pumps or fans.

If fluid movement is created by a pump or a fan, heat is being transferred by forced convection. Examples are a pump circulating hot water through a building heating system, a fan forcing air through an automobile radiator, or a forced draft fan pushing hot gases through a boiler.

5.3 Radiation

Radiation transfers heat through electromagnetic waves that travel in straight lines and can pass through a vacuum. When these waves strike a surface, they may be absorbed (increasing temperature), reflected, or transmitted. Dark, rough surfaces absorb more radiation, while shiny, smooth ones reflect it.

Examples include heat from the sun reaching Earth and radiant heat in a boiler furnace.

In a steam boiler, radiation occurs in the furnace. Any heating surfaces that are directly exposed to the furnace will receive heat directly by radiation from the flame. These include the waterwalls and some generating tubes of a watertube boiler, radiant superheater tubes (located at the outlet of the furnace), and the furnace walls of a firetube boiler.

Chapter 6

Perfect Gases

The terms “perfect gas” and “ideal gas” are often used interchangeably in physics and engineering, but “ideal gas” is the more standard term in modern scientific literature.

An ideal gas is a theoretical model that remains in the gaseous state under all conditions, as it does not account for phase changes such as condensation. Real gases, such as nitrogen, oxygen, and dry air, are often approximated as ideal gases under typical conditions (e.g., low pressures and high temperatures relative to their critical points), but they may condense under extreme conditions. Saturated steam, being at the liquid-vapor phase boundary, does not behave as an ideal gas due to significant intermolecular forces, whereas superheated steam approximates ideal gas behavior when sufficiently far from its condensation point.

In an ideal gas, pressure, temperature, and volume are interrelated, enabling the calculation of changes in one property based on the others. These relationships are governed by Boyle’s Law, Charles’s Law, and the ideal gas law, which collectively describe the behavior of an ideal gas under varying conditions.

6.1 Boyle’s Law

Robert Boyle (1627–1691), a physicist, investigated the behavior of an ideal gas at constant temperature. By controlling the addition or removal of heat during changes in the volume and pressure of a confined gas, he maintained a constant temperature. He discovered that, under these conditions, the absolute pressure of a gas is inversely proportional to its volume. That is, as the volume increases, the pressure decreases, and conversely, as the volume decreases, the pressure increases. This can be stated as:\index{Boyle’s Law}.

$$P \propto \frac{1}{V} \tag{6.1}$$

6.2 Charles's Law

Charles's Law, named after Jacques Charles (1746–1823), a French physicist, states that for an ideal gas maintained at constant pressure, the volume of the gas is directly proportional to its absolute temperature (measured in Kelvin). This means that as the temperature of a gas increases, its volume increases proportionally, provided the pressure remains constant. Conversely, as the temperature decreases, the volume decreases. The law assumes the gas behaves as an ideal gas. This can be stated as:\index{Charles's Law}.

$$V \propto T \tag{6.2}$$

6.3 Boyle's Law — Pressure–Volume Relationship

Assuming compression according to the law $pV = \text{constant}$:

1. Calculate the final volume when $1\,\text{m}^3$ of gas at a pressure of $120\,\text{kN/m}^2$ is compressed to a pressure of $960\ \text{kN/m}^2$.
2. Calculate the initial volume of a gas at a pressure of $1.05\,\text{bar}$ that will occupy a volume of $5.6\,\text{m}^3$ when it is compressed to a pressure of $42\,\text{bar}$.

6.3.1 Given:

The compression (or expansion) follows the law:

$$p_1 V_1 = p_2 V_2$$

6.3.2 Final Volume of Gas

Given: $p_1 = 120\ \ \text{kN/m}^2, \quad V_1 = 1\ \ \text{m}^3, \quad p_2 = 960\ \ \text{kN/m}^2$

From $p_1 V_1 = p_2 V_2$:

$$V_2 = \frac{p_1 V_1}{p_2}$$

Substitute:

$$V_2 = \frac{120 \times 1}{960} = 0.125\ \text{m}^3$$

Final Volume

$\boxed{V_2 = 0.125\ \text{m}^3}$

6.3.3 Initial Volume of Gas

Given: $p_1 = 1.05 \ \text{bar}, \quad V_2 = 5.6 \ \text{m}^3, \quad p_2 = 42 \ \text{bar}$

From $p_1 V_1 = p_2 V_2$:

$$V_1 = \frac{p_2 V_2}{p_1}$$

Substitute:

$$V_1 = \frac{42 \times 5.6}{1.05}$$

$$V_1 = 224 \text{ m}^3$$

Initial Volume

$\boxed{V_1 = 224 \text{ m}^3}$

6.3.4 Code

```
# Boyle's Law Calculator (p1 * V1 = p2 * V2)

print("Boyle’s Law Calculator: p V  = p V \n")

# Get known values (enter 0 for the unknown)
p1 = float(input("Enter initial pressure p  (in kN/m² or bar): "))
V1 = float(input("Enter initial volume V  (m³): "))
p2 = float(input("Enter final pressure p  (in kN/m² or bar): "))
V2 = float(input("Enter final volume V  (m³): "))

print()

# Determine which variable is missing (the one entered as 0)
if V2 == 0:
    V2 = (p1 * V1) / p2
    print(f"Final volume V  = {V2:.4f} m³")

elif V1 == 0:
    V1 = (p2 * V2) / p1
    print(f"Initial volume V  = {V1:.4f} m³")

elif p2 == 0:
    p2 = (p1 * V1) / V2
    print(f"Final pressure p  = {p2:.4f}")
```

```
elif p1 == 0:
    p1 = (p2 * V2) / V1
    print(f"Initial pressure p  = {p1:.4f}")

else:
    print("All variables entered - nothing to calculate!")

print("\nUnits:")
print("• If you used bar, results are in bar.")
print("• If you used kN/m², results are in kN/m².")
```

6.4 Compression — Stroke Volume and Final Temperature

A gas is compressed in a cylinder from $p_1 = 1$ bar and $_1 = 35°\text{C}$ at the beginning of the stroke to $p_2 = 37$ bar at the end of the stroke. The clearance volume is $V_c = 850\ \text{cm}^3$. The compression index is $n = 1.32$. Find the stroke (swept) volume V_s and the temperature at the end of compression T_2.

6.4.1 Solution

For a polytropic compression with $pV^n =$ constant (here $n = 1.32$ and with clearance V_c:

Start of compression (BDC) volume:

$$V_1 = V_c + V_s$$

End of compression (TDC) volume:

$$V_2 = V_c$$

From $(p_1 V_1^n = p_2 V_2^n)$ we get

$$\left(\frac{V_2}{V_1}\right)^n = \frac{p_1}{p_2}$$

so

$$\frac{V_c}{V_c + V_s} = \left(\frac{p_1}{p_2}\right)^{1/n}.$$

Rearrange to solve for V_s:

$$V_s = V_c\left[\left(\frac{p_2}{p_1}\right)^{1/n} - 1\right].$$

Substitute numbers (convert V_c to m³: $850\ \text{cm}^3 = 850 \times 10^{-6}\text{m}^3 = 0.00085\ \text{m}^3$:

$$V_s = 0.00085\left[\left(\tfrac{37}{1}\right)^{1/1.32} - 1\right]$$

Numerical evaluation:

$$V_s \approx 0.012255\ \text{m}^3 = 12.26\ \text{L}.$$

6.4.2 Final temperature

For a polytropic process the temperature ratio is

$$\frac{T_2}{T_1} = \left(\frac{p_2}{p_1}\right)^{\frac{n-1}{n}}.$$

Convert T_1 to kelvin: $T_1 = 35^\circ\text{C} = 308.15\ \text{K}$.

$$T_2 = 308.15\left(\frac{37}{1}\right)^{\frac{1.32-1}{1.32}}$$

Numerical evaluation:

$$T_2 \approx 739.49\ \text{K} = 466.34^\circ\text{C}.$$

6.4.3 Answers

- Stroke (swept) volume: $\boxed{V_s \approx 0.01226\ \text{m}^3\ (12.26\ \text{L})}$
- Temperature at end of compression: $\boxed{T_2 \approx 739.5\ \text{K}\ =\ 466.3^\circ\text{C}}$.

6.4.4 Code

```
# Given data
p1 = 1       # bar
p2 = 37      # bar
Vc_cm3 = 850 # clearance volume in cm³
n = 1.32     # compression index
T1_C = 35    # °C

# Unit conversions
Vc = Vc_cm3 * 1e-6      # m³
T1 = T1_C + 273.15       # K

# Stroke volume calculation
```

```
Vs = Vc * ((p2 / p1)**(1/n) - 1)

# Final temperature calculation
T2 = T1 * (p2 / p1)**((n - 1) / n)
T2_C = T2 - 273.15

print(f"Stroke volume Vs = {Vs:.5f} m³ ({Vs*1000:.2f} L)")
print(f"Final temperature T2 = {T2:.2f} K ({T2_C:.2f} °C)")
```

Chapter 7

IC Engines

Internal combustion (IC) engines derive their name from fuel burning inside the engine cylinder. The combustion heats the air in the cylinder, increasing its pressure and moving the piston, whose reciprocating motion is converted to rotary motion by the crankshaft.

IC engines are classified by their ignition method:

- Compression Ignition: Air is highly compressed, raising its temperature so injected fuel ignites spontaneously.
- Spark Ignition: A fuel–air mixture is compressed and then ignited by a spark plug.

Chapter 8

Ideal Cycles

8.1 Ideal Constant-Volume (Otto) Cycle

The Ideal Constant-Volume (Otto) Cycle, commonly referred to as the Otto cycle, is a thermodynamic cycle that models the operation of a spark-ignition internal combustion engine, such as those used in gasoline-powered automobiles. It is an idealized model that assumes the working fluid is an ideal gas and describes the sequence of processes that convert thermal energy into mechanical work.

8.1.1 P-V Diagram Representation

The Otto cycle on a P-V diagram forms a closed loop:

- 1-2: Steep upward curve (pressure increases, volume decreases, adiabatic compression).
- 2-3: Vertical line upward (pressure and temperature increase, volume constant, heat addition).
- 3-4: Downward curve (pressure decreases, volume increases, adiabatic expansion).
- 4-1: Vertical line downward (pressure and temperature decrease, volume constant, heat rejection).

8.2 Otto Cycle Example

In an ideal constant-volume (Otto) cycle the temperature at the beginning of compression is 50°C. The volumetric compression ratio is $r = 5 : 1$. The heat supplied during the cycle is $q_{in} = 930$ kJ/kg of working fluid. Take $\gamma = 1.4$ and $c_v = 0.717$ kJ/kg $\cdot$ K.

Calculate:

A. The maximum temperature attained in the cycle.
B. The work done during the cycle per kg of working fluid.
C. The ideal thermal efficiency of the cycle.

8.2.1 Theory and formulas

For an ideal Otto cycle (constant-volume heat addition) the relevant steps and formulas are:

1. Isentropic compression $1 \to 2$:

$$T_2 = T_1\, r^{\gamma-1}$$

2. Constant-volume heat addition $2 \to 3$:

$$q_{in} = c_v(T_3 - T_2) \quad \Rightarrow \quad T_3 = T_2 + \frac{q_{in}}{c_v}$$

3. Isentropic expansion $3 \to 4$:

$$T_4 = T_3\, r^{1-\gamma}$$

4. Heat rejected:

$$q_{out} = c_v(T_4 - T_1)$$

5. Net work per unit mass:

$$w = q_{in} - q_{out}$$

6. Thermal efficiency:

$$\eta = \frac{w}{q_{in}} = 1 - \frac{q_{out}}{q_{in}}$$

(For the ideal Otto cycle the formula reduces to the well-known expression $\eta = 1 - r^{1-\gamma}$, which provides a check.)

8.2.2 Code

```
# Otto-cycle calculation (per kg)

# Given data
r = 5.0             # compression ratio
gamma = 1.4
cv = 0.717          # kJ/kg·K
T1_C = 50.0         # °C
q_in = 930.0        # kJ/kg
```

```
# Conversions
T1 = T1_C + 273  # K

# Step 1: T2 (after isentropic compression)
T2 = T1 * r**(gamma - 1)

# Step 2: T3 (after constant-volume heat addition) -> Tmax
T3 = T2 + q_in / cv

# Step 3: T4 (after isentropic expansion)
T4 = T3 * r**(1 - gamma)

# Heat rejected
q_out = cv * (T4 - T1)

# Net work per kg
w_net = q_in - q_out

# Efficiency
eta = w_net / q_in

# Print results
print(f"T1 = {T1:.2f} K ({T1 - 273:.2f} °C)")
print(f"T2 = {T2:.2f} K ({T2 - 273:.2f} °C)")
print(f"T3 = {T3:.2f} K ({T3 - 273:.2f} °C)  <-- Tmax")
print(f"T4 = {T4:.2f} K ({T4 - 273:.2f} °C)")
print()
print(f"q_in = {q_in:.2f} kJ/kg")
print(f"q_out = {q_out:.2f} kJ/kg")
print(f"w_net = {w_net:.2f} kJ/kg")
print(f"efficiency = {eta*100:.2f} %")
```

8.2.3 Using Actual Volume Ratios Instead of Direct r

In the ideal Otto cycle we have written:

$$T_2 = T_1\, r^{\gamma - 1},$$

with $r = V_1/V_2$. Here we make the volume-dependence explicit using V_1 and V_2.

8.2.4 Given

- $T_1 = 50^\circ\text{C} = 323$ K

- Volumetric compression ratio $r = \dfrac{V_1}{V_2} = 5$ (we'll set $V_1 = 5,\ V_2 = 1$ so $V_1/V_2 = 5$
- Heat supplied $q_{in} = 930$ kJ/kg
- $\gamma = 1.4,\ c_v = 0.717$ kJ/kg $\cdot$ K

8.2.5 Volume-explicit relations

Isentropic compression (1→2) (using volumes):

$$\frac{T_2}{T_1} = \left(\frac{V_1}{V_2}\right)^{\gamma-1} \quad \Rightarrow \quad T_2 = T_1 \left(\frac{V_1}{V_2}\right)^{\gamma-1}.$$

Constant-volume heat addition (2→3):

$$T_3 = T_2 + \frac{q_{in}}{c_v}.$$

Isentropic expansion (3→4) (volume ratio inverted):

$$\frac{T_4}{T_3} = \left(\frac{V_2}{V_1}\right)^{\gamma-1} \quad \Rightarrow \quad T_4 = T_3 \left(\frac{V_2}{V_1}\right)^{\gamma-1}.$$

Heat rejected:

$$q_{out} = c_v(T_4 - T_1),$$

Net work and efficiency:

$$w = q_{in} - q_{out}, \qquad \eta = \frac{w}{q_{in}}.$$

8.2.6 Code

```
# Otto-cycle calculation using explicit volumes (per kg)

# Given data
V1 = 5.0        # arbitrary volume units (e.g. litres) at start of compression (BDC)
V2 = 1.0        # arbitrary volume units at end of compression (TDC)
gamma = 1.4
cv = 0.717      # kJ/kg·K
T1_C = 50.0     # °C
q_in = 930.0    # kJ/kg

# Conversions
T1 = T1_C + 273  # K
```

```
# Isentropic compression using volume ratio V1/V2
T2 = T1 * (V1 / V2)**(gamma - 1)

# Constant-volume heat addition
T3 = T2 + q_in / cv

# Isentropic expansion using inverted volume ratio V2/V1
T4 = T3 * (V2 / V1)**(gamma - 1)

# Heat rejected, net work and efficiency
q_out = cv * (T4 - T1)
w_net = q_in - q_out
eta = w_net / q_in

# Print results
print("--- Inputs ---")
print(f"V1 = {V1:.3f} (units), V2 = {V2:.3f} (units)  --> V1/V2 = {V1/V2:.3f}")
print(f"T1 = {T1:.2f} K ({T1-273:.2f} °C)")
print()
print("--- Results ---")
print(f"T2 = {T2:.2f} K ({T2-273:.2f} °C)")
print(f"T3 = {T3:.2f} K ({T3-273:.2f} °C)  <-- Tmax")
print(f"T4 = {T4:.2f} K ({T4-273:.2f} °C)")
print()
print(f"q_in = {q_in:.2f} kJ/kg")
print(f"q_out = {q_out:.2f} kJ/kg")
print(f"w_net = {w_net:.2f} kJ/kg")
print(f"efficiency = {eta*100:.2f} %")
```

8.3 Ideal Constant-Pressure (Diesel) Cycle

The Diesel cycle is a thermodynamic model of a compression-ignition engine, like those in marine diesel engines. Unlike the Otto cycle (constant-volume heat addition), the Diesel cycle assumes heat is added at constant pressure, reflecting the actual combustion in diesel engines.

The Diesel cycle consists of four thermodynamic processes—two isentropic (adiabatic and reversible), one constant-pressure, and constant-volume that describe the idealized behavior of a gas undergoing compression, combustion, expansion, and exhaust in a compression-ignition engine.

8.3.1 P-V Diagram Representation

The Diesel cycle on a P-V diagram forms a closed loop:

- 1-2: Steep upward curve (pressure increases, volume decreases, adiabatic compression).
- 2-3: Horizontal line (pressure constant, volume increases, constant-pressure heat addition).
- 3-4: Downward curve (pressure decreases, volume increases, adiabatic expansion).
- 4-1: Vertical line downward (pressure and temperature decrease, volume constant, heat rejection).

8.4 Diesel Cycle Example

Given

- $T_1 = 50^{\circ}\mathrm{C} = 323$ K
- Compression ratio $r = \dfrac{V_1}{V_2} = 5$
- Heat supplied $q_{in} = 930$ kJ/kg (added at constant pressure)
- $\gamma = 1.4, \quad c_v = 0.717$ kJ/kg · K

Notes: For constant-pressure heat addition (2→3),

$$q_{in} = c_p\,(T_3 - T_2), \qquad c_p = c_v\gamma.$$

The cut-off (volume) ratio is $\rho = \dfrac{V_3}{V_2} = \dfrac{T_3}{T_2}$ because $p_2 = p_3$

8.4.1 Code

```
# Diesel-cycle calculation (per kg) - runnable in Quarto

import numpy as np
import matplotlib.pyplot as plt

# Given
r = 5.0                      # V1 / V2
gamma = 1.4
cv = 0.717                   # kJ/kg·K
cp = cv * gamma              # kJ/kg·K
T1 = 323.0                   # K (simplified 273 + 50)
q_in = 930.0                 # kJ/kg

# State 2: isentropic compression 1->2
T2 = T1 * r**(gamma - 1)
```

```
# State 3: constant-pressure heat addition 2->3
# q_in = cp * (T3 - T2)  ->  T3 = T2 + q_in / cp
T3 = T2 + q_in / cp

# Cut-off ratio (V3/V2) since p2 = p3
rho = T3 / T2

# State 4: isentropic expansion 3->4 (V4 = V1)
# T4 = T3 * (V3/V4)^(gamma-1) = T3 * (rho / r)^(gamma-1)
T4 = T3 * (rho / r)**(gamma - 1)

# Heat rejected, net work and efficiency
q_out = cv * (T4 - T1)
w_net = q_in - q_out
eta = w_net / q_in

# Pressures (normalized) using ideal-gas relation P   T/V
P1 = 1.0
P2 = P1 * (T2/T1) * (V1 := r) / (V2 := 1.0)   # V1=r, V2=1 chosen for explicitness
P3 = P2
V3 = rho * V2
P4 = (T4 / T1) * (V1 / (V1)) * P1  # alternatively compute via ideal gas: P4 = P3 * (T4/T3) * (V3

# More consistent P4 via ideal gas:
P4 = P3 * (T4 / T3) * (V3 / V1)

# Curves for PV diagram
V_compression = np.linspace(V1, V2, 200)  # 1 -> 2 (compression)
P_compression = P1 * (V1 / V_compression)**gamma

V_expansion = np.linspace(V3, V1, 200)    # 3 -> 4 (expansion)
P_expansion = P3 * (V3 / V_expansion)**gamma

plt.figure(figsize=(8,5))
plt.plot(V_compression, P_compression, label="Isentropic compression (1→2)")
plt.plot([V2, V3], [P2, P3], 'r-', linewidth=2, label="Constant-pressure heat addition (2→3)")
plt.plot(V_expansion, P_expansion, label="Isentropic expansion (3→4)")
plt.plot([V1, V1], [P4, P1], 'r-', linewidth=2, label="Constant-volume heat rejection (4→1)")
# mark state points
plt.plot([V1, V2, V3, V1], [P1, P2, P3, P4], 'ko')
for (V,P,label) in [(V1,P1,"1"), (V2,P2,"2"), (V3,P3,"3"), (V1,P4,"4")]:
    plt.text(V, P*1.02, label, ha='center')

plt.xlabel("Volume (arbitrary units)")
plt.ylabel("Pressure (arbitrary units)")
```

```
plt.title("Ideal Diesel Cycle - P-V Diagram (normalized)")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# Print numeric results
print("---- Diesel Cycle Results (per kg) ----")
print(f"T1 = {T1:.2f} K")
print(f"T2 = {T2:.2f} K")
print(f"T3 = {T3:.2f} K  (Tmax after constant-pressure heat addition)")
print(f"T4 = {T4:.2f} K")
print()
print(f"Cut-off ratio (rho = V3/V2) = {rho:.4f}")
print(f"q_in = {q_in:.2f} kJ/kg")
print(f"q_out = {q_out:.2f} kJ/kg")
print(f"Net work w = {w_net:.2f} kJ/kg")
print(f"Ideal thermal efficiency = {eta*100:.2f} %")
```

8.5 Dual Combustion Cycle

The Dual Combustion cycle, or mixed cycle, models engines that combine constant-volume (Otto cycle) and constant-pressure (Diesel cycle) combustion. Fuel burns in two phases: a fast initial phase at constant volume, then a slower phase at constant pressure, providing a realistic approximation of modern diesel engine combustion.

8.5.1 Efficiency of the Dual Combustion Cycle

The thermal efficiency (η) of the ideal Dual Combustion Cycle is defined as the ratio of net work output to total heat input:

$\eta = \frac{W_{\text{net}}}{Q_{\text{in}}} = \frac{Q_{\text{in},1}+Q_{\text{in},2}-Q_{\text{out}}}{Q_{\text{in},1}+Q_{\text{in},2}}$

Substituting the heat terms:

$Q_{\text{in},1} = mC_v(T_3 - T_2)$

$Q_{\text{in},2} = mC_p(T_4 - T_3)$

$Q_{\text{out}} = mC_v(T_5 - T_1)$

$\eta = 1 - \frac{Q_{\text{out}}}{Q_{\text{in},1}+Q_{\text{in},2}} = 1 - \frac{C_v(T_5-T_1)}{C_v(T_3-T_2)+C_p(T_4-T_3)}$

$$\eta = 1 - \frac{(T_5 - T_1)}{(T_3 - T_2) + \gamma(T_4 - T_3)} \tag{8.1}$$

8.5.2 P-V Diagram Representation

The Dual Combustion Cycle on a P-V diagram forms a closed loop:

- 1-2: Steep upward curve (pressure increases, volume decreases, adiabatic compression).
- 2-3: Vertical line upward (pressure and temperature increase, volume constant, heat addition).
- 3-4: Horizontal line (pressure constant, volume increases, heat addition).
- 4-5: Downward curve (pressure decreases, volume increases, adiabatic expansion).
- 5-1: Vertical line downward (pressure and temperature decrease, volume constant, heat rejection).

Appendix A

SI System Common Mistakes

Using the SI system correctly is crucial for clear communication in science and engineering. Below are common mistakes in using the SI system, examples of incorrect usage, and how to correct them.

Table A.1: SI system rules and common mistakes

Concept	Mistake	Correct Usage	Notes
Use of SI Unit Symbols	`m./s`	`m/s`	Use the correct format without additional punctuation.
Spacing Between Value & Unit	`10kg`	`10 kg`	Always leave a space between the number and the unit symbol.
Incorrect Unit Symbols	`sec`, `hrs`, `°K`	`s`, `h`, `K`	Use the proper SI symbols; symbols are case-sensitive.
Abbreviations for Units	`5 kilograms (kgs)`	`5 kilograms (kg)`	Avoid informal abbreviations like "kgs"; adhere to standard symbols.

Concept	Mistake	Correct Usage	Notes
Multiple Units in Expressions	`5 m/s/s`, `5 kg/meter`^{2}	`5 m/s`^{2}, `5 kg/m`^{2}	Use compact, standardized formats for derived units.
Incorrect Use of Prefixes	`0.0001 km`	`100 mm`	Choose prefixes to keep numbers in the range (0.1 x < 1000).
Misplaced Unit Symbols	`5/s`, `kg10`	`5 s` ^{1}, `10 kg`	Symbols must follow numerical values, not precede them.
Degrees Celsius vs. Kelvin	`300°K`	`300 K`	Kelvin is written without "degree"
Singular vs. Plural Units	`5 kgs`, `1 meters`	`5 kg`, `1 meter`	Symbols do not pluralize; full unit names follow grammar rules.
Capitalization of Symbols	`Kg`, `S`, `Km`, `MA`	`kg`, `s`, `km`, `mA`	Symbols are case-sensitive; use uppercase only where specified (e.g., `N`, `Pa`).
Capitalization of Unit Names	`Newton`, `Pascal`, `Watt`	`newton`, `pascal`, `watt`	Unit names are lowercase, even if derived from a person's name, unless starting a sentence.
Prefix Capitalization	`MilliMeter`, `MegaWatt`	`millimeter`, `megawatt`	Prefixes are lowercase for (10^{-1}) to (10^{-9}), uppercase for (10^6) and larger (except `k` for kilo).
Formatting in Reports	`5`, `Temperature: 300`	`5 kg`, `Temperature: 300 K`	Always specify units explicitly.

Appendix B

Greek Letters

The following tables present the names of Greek letters and selected symbols commonly used in engineering courses, ensuring precise reference and avoiding reliance on informal descriptors such as “squiggle.”

Table B.1: Greek letters.

Lower Case	Upper Case	Name
α	A	alpha
β	B	beta
γ	Γ	gamma
δ	Δ	delta
ϵ	E	epsilon
ζ	Z	zeta
η	E	eta
θ	Θ	theta
ι	I	iota
κ	K	kappa
λ	Λ	lambda
μ	M	mu
ν	N	nu
ξ	Ξ	xi
o	O	omicron
π	Π	pi
ρ	P	rho
σ	Σ	sigma
τ	T	tau
υ	Υ	upsilon
ϕ	Φ	phi
χ	X	chi

Lower Case	Upper Case	Name
ψ	Ψ	psi
ω	Ω	omega

Table B.2: Commonly used symbols in engineering courses.

Symbol	Name	Use	Course
Δ	Delta	Change	Thermodynamics
Δ	Delta	Displacement	Naval Architecture
∇	Nabla	Volume	Naval Architecture
Σ	Sigma	Sum	Thermodynamics, Naval Architecture, Applied Mechanics
σ	Sigma	Stress	Thermodynamics, Applied Mechanics
ϵ	Epsilon	Modulus of elasticity	Thermodynamics, Applied Mechanics
η	Eta	Efficiency	Thermodynamics
ω	Omega	Angular velocity	Thermodynamics, Applied Mechanics
ρ	Rho	Density	Thermodynamics, Naval Architecture
τ	Tau	Torque	Thermodynamics, Applied Mechanics

Index