Capacitor Charging and Discharging Simulation

Team: Team 12

Project Report

---

1. Introduction

Capacitors are devices that store electrical energy in an electric field, widely used in circuits for various functions. One important application is their role in RC circuits, which are circuits composed of resistors (R) and capacitors (C). This project focuses on simulating the charging and discharging processes of a capacitor in an RC circuit and visualizing these behaviors using mathematical models and graphical plots.

This report details the development of a simulation tool, its mathematical foundation, and the use of Git and GitHub for collaboration, version control, and documentation.

---

2. Basic Principles of RC Circuits

2.1 Components of an RC Circuit

An RC circuit consists of a resistor and a capacitor connected in series or parallel. When a voltage is applied across the circuit, the capacitor starts charging through the resistor. Over time, the voltage across the capacitor increases and the current decreases. When the voltage source is removed, the capacitor discharges, releasing the stored energy.

2.2 Time Constant

The time constant, denoted as τ\tauτ, represents how quickly the capacitor charges or discharges. It is given by the formula:

$$\tau = R \times C$$

Where:

- R is the resistance in ohms (Ω\Omega)
- C is the capacitance in farads (F)

The time constant defines how fast the capacitor charges to approximately 63% of the applied voltage or discharges to 37% of its initial charge in one time constant.

---

3. Mathematical Models

3.1 Charging of a Capacitor

The voltage across the capacitor during the charging process is given by:

Vcharging(t) =V0 * (1 - np.exp(-t / τ))

Where:

- V0 is the initial voltage across the capacitor
- t is the time
- e is Euler's constant
- τ is the time constant

3.2 Discharging of a Capacitor

- The voltage across the capacitor during the discharging process is expressed as:

  Vdischarging(t) = V0 * np.exp(-t / τ)

- This equation shows that the voltage exponentially decays over time as the capacitor discharges.

We have implemented the mathematical model using python

Code Explanation:

The code simulates the charging and discharging behavior of a capacitor in a series RC circuit. It calculates the voltage across the capacitor at different time points using the appropriate mathematical equations.

Key Variables and Constants:

- R: Resistance of the resistor (100 ohms).
- C: Capacitance of the capacitor (0.001 farads).

- Vb: Supply voltage for charging (5 volts).
- V0: Initial voltage for discharging (set to Vb by default).
- t_final: Total simulation time (0.3 seconds).
- dt: Time step (0.01 seconds).
- time_points: Array of time points for the simulation.
- Vc_charge: Array to store the capacitor voltage during charging.
- Vc_discharge: Array to store the capacitor voltage during discharging

```python
import numpy as np
# Constants
R = 100  # Resistance in ohms
C = 0.001  # Capacitance in farads
Vb = 5  # Supply voltage in volts (for charging)
V0 = Vb  # Initial voltage for discharging (can be adjusted)
t_final = 1.5  # Total time for the simulation (in seconds)
dt = 0.01  # Time step
time_points = np.arange(0, t_final, dt)  # Create time array

# Prepare arrays for voltage across capacitor
Vc_charge = np.zeros_like(time_points)  # Voltage array for charging
Vc_discharge = np.zeros_like(time_points)  # Voltage array for discharging

# Mathematical model for charging the capacitor
for i, t in enumerate(time_points):
    Vc_charge[i] = Vb * (1 - np.exp(-t / (R * C)))  # Charging equation

# Mathematical model for discharging the capacitor
for i, t in enumerate(time_points):
    Vc_discharge[i] = V0 * np.exp(-t / (R * C))  # Discharging equation

# Output the results
print("Charging Phase:")
for t, voltage in zip(time_points, Vc_charge):
    print(f"Time: {t:.2f} s, Voltage across Capacitor (Charging): {voltage:.2f} V")

print("\nDischarging Phase:")
for t, voltage in zip(time_points, Vc_discharge):
    print(f"Time: {t:.2f} s, Voltage across Capacitor (Discharging): {voltage:.2f} V")
```

```
Charging Phase:
Time: 0.00 s, Voltage across Capacitor (Charging): 0.00 V
Time: 0.01 s, Voltage across Capacitor (Charging): 0.48 V
Time: 0.02 s, Voltage across Capacitor (Charging): 0.91 V
Time: 0.03 s, Voltage across Capacitor (Charging): 1.30 V
Time: 0.04 s, Voltage across Capacitor (Charging): 1.65 V
Time: 0.05 s, Voltage across Capacitor (Charging): 1.97 V
Time: 0.06 s, Voltage across Capacitor (Charging): 2.26 V
Time: 0.07 s, Voltage across Capacitor (Charging): 2.52 V
Time: 0.08 s, Voltage across Capacitor (Charging): 2.75 V
Time: 0.09 s, Voltage across Capacitor (Charging): 2.97 V
Time: 0.10 s, Voltage across Capacitor (Charging): 3.16 V
Time: 0.11 s, Voltage across Capacitor (Charging): 3.34 V
Time: 0.12 s, Voltage across Capacitor (Charging): 3.49 V
Time: 0.13 s, Voltage across Capacitor (Charging): 3.64 V
Time: 0.14 s, Voltage across Capacitor (Charging): 3.77 V
Time: 0.15 s, Voltage across Capacitor (Charging): 3.88 V
Time: 0.16 s, Voltage across Capacitor (Charging): 3.99 V
Time: 0.17 s, Voltage across Capacitor (Charging): 4.09 V
Time: 0.18 s, Voltage across Capacitor (Charging): 4.17 V
Time: 0.19 s, Voltage across Capacitor (Charging): 4.25 V
Time: 0.20 s, Voltage across Capacitor (Charging): 4.32 V
Time: 0.21 s, Voltage across Capacitor (Charging): 4.39 V
Time: 0.22 s, Voltage across Capacitor (Charging): 4.45 V
Time: 0.23 s, Voltage across Capacitor (Charging): 4.50 V
Time: 0.24 s, Voltage across Capacitor (Charging): 4.55 V
Time: 0.25 s, Voltage across Capacitor (Charging): 4.59 V
Time: 0.26 s, Voltage across Capacitor (Charging): 4.63 V
Time: 0.27 s, Voltage across Capacitor (Charging): 4.66 V
Time: 0.28 s, Voltage across Capacitor (Charging): 4.70 V
Time: 0.29 s, Voltage across Capacitor (Charging): 4.72 V
```

```
Discharging Phase:
Time: 0.00 s, Voltage across Capacitor (Discharging): 5.00 V
Time: 0.01 s, Voltage across Capacitor (Discharging): 4.52 V
Time: 0.02 s, Voltage across Capacitor (Discharging): 4.09 V
Time: 0.03 s, Voltage across Capacitor (Discharging): 3.70 V
Time: 0.04 s, Voltage across Capacitor (Discharging): 3.35 V
Time: 0.05 s, Voltage across Capacitor (Discharging): 3.03 V
Time: 0.06 s, Voltage across Capacitor (Discharging): 2.74 V
Time: 0.07 s, Voltage across Capacitor (Discharging): 2.48 V
Time: 0.08 s, Voltage across Capacitor (Discharging): 2.25 V
Time: 0.09 s, Voltage across Capacitor (Discharging): 2.03 V
Time: 0.10 s, Voltage across Capacitor (Discharging): 1.84 V
Time: 0.11 s, Voltage across Capacitor (Discharging): 1.66 V
Time: 0.12 s, Voltage across Capacitor (Discharging): 1.51 V
Time: 0.13 s, Voltage across Capacitor (Discharging): 1.36 V
Time: 0.14 s, Voltage across Capacitor (Discharging): 1.23 V
Time: 0.15 s, Voltage across Capacitor (Discharging): 1.12 V
Time: 0.16 s, Voltage across Capacitor (Discharging): 1.01 V
Time: 0.17 s, Voltage across Capacitor (Discharging): 0.91 V
Time: 0.18 s, Voltage across Capacitor (Discharging): 0.83 V
Time: 0.19 s, Voltage across Capacitor (Discharging): 0.75 V
Time: 0.20 s, Voltage across Capacitor (Discharging): 0.68 V
Time: 0.21 s, Voltage across Capacitor (Discharging): 0.61 V
Time: 0.22 s, Voltage across Capacitor (Discharging): 0.55 V
Time: 0.23 s, Voltage across Capacitor (Discharging): 0.50 V
Time: 0.24 s, Voltage across Capacitor (Discharging): 0.45 V
Time: 0.25 s, Voltage across Capacitor (Discharging): 0.41 V
Time: 0.26 s, Voltage across Capacitor (Discharging): 0.37 V
Time: 0.27 s, Voltage across Capacitor (Discharging): 0.34 V
Time: 0.28 s, Voltage across Capacitor (Discharging): 0.30 V
Time: 0.29 s, Voltage across Capacitor (Discharging): 0.28 V
```

---

4. Implementation

4.1 Simulation Code

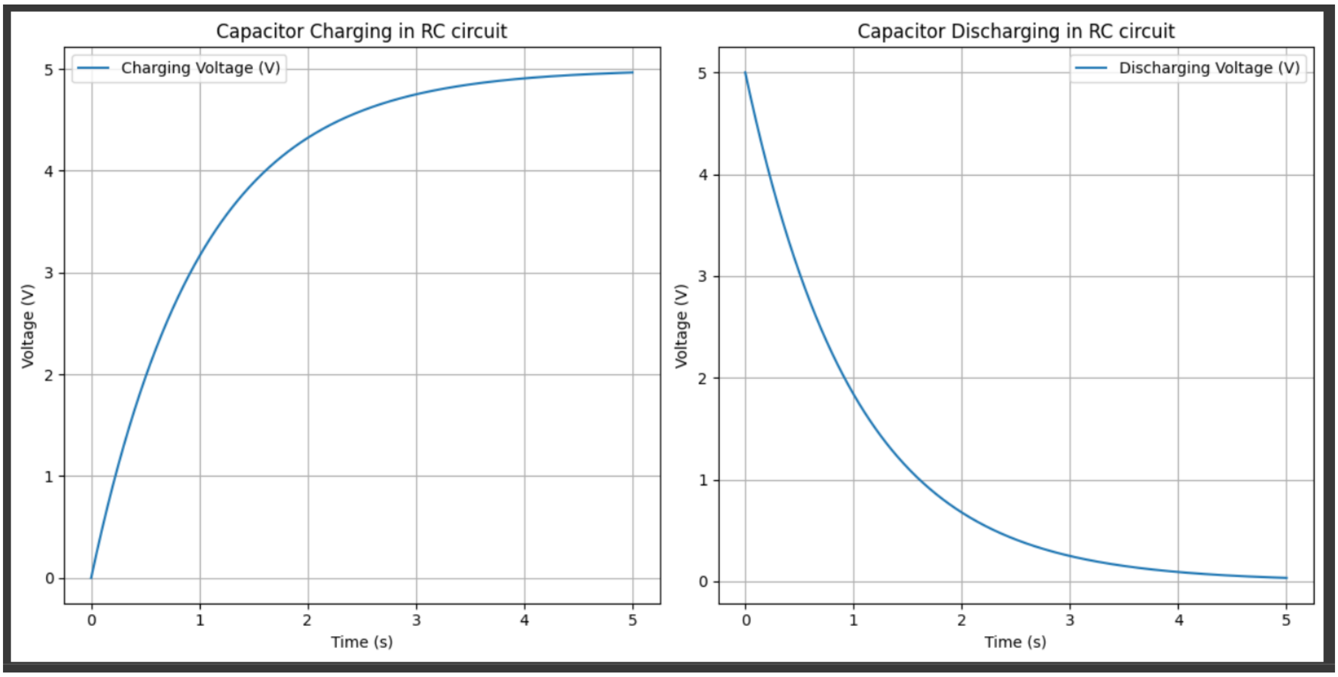
A Python script was developed to simulate both the charging and discharging phases of the capacitor. We used the following tools and libraries:

- **Numpy**: For numerical operations
- **Matplotlib**: For plotting graphs
- **SymPy**: For symbolic calculations (optional)

Below is a simplified code used to generate the charging and discharging curves:

```python
import numpy as np
import matplotlib.pyplot as plt
import sympy as sp

# Constants
R = 1.0  # Resistance (Ohms)
C = 1.0  # capacitor in Farad(F)
V0 = 5.0  # Initial voltage (volts)
tau = R * C  # Time constant

# Time values
t = np.linspace(0, 5 * tau, 500)

# Charge functions for charging and discharging
Capacitor_charging = V0 * (1 - np.exp(-t / tau))
Capacitor_discharging = V0 * np.exp(-t / tau)

# Create plots
plt.figure(figsize=(12, 6))
```

```python
# Plot charging
plt.subplot(121)
plt.plot(t, Capacitor_charging, label='Charging Voltage (V)')
plt.xlabel('Time (s)')
plt.ylabel('Voltage (V)')
plt.title('Capacitor Charging in RC circuit')
plt.grid()
plt.legend()

# Plot discharging
plt.subplot(122)
plt.plot(t, Capacitor_discharging, label='Discharging Voltage (V)')
plt.xlabel('Time (s)')
plt.ylabel('Voltage (V)')
plt.title('Capacitor Discharging in RC circuit')
plt.grid()
plt.legend()

# Show the plots
plt.tight_layout()
plt.show()
```



The charging curve of a capacitor in an RC circuit shows an exponential rise in voltage over time, starting from zero and approaching the applied voltage V0. The rate of this rise is governed by the time constant $\tau=RC$, which determines how quickly the capacitor charges.

Conversely, the discharging curve represents the voltage across the capacitor decaying exponentially as it releases stored energy. Starting from V0, the voltage decreases over time, approaching zero, with the time constant τ determining the speed of this process. Both curves illustrate the capacitor's behavior as it stores and releases energy in the circuit.

---

5. Utilization of Git and GitHub

5.1 Repository Setup

A GitHub repository was created to facilitate collaboration among the team members. The repository allowed team members to track the progress, manage versions of the code, and document the project efficiently.

5.2 Branching and Collaboration

To keep the project organized, we followed GitHub's best practices:

- **Branching**: Each team worked on separate branches for their.

- **Pull Requests (PRs)**: Code reviews were conducted before merging changes into the main branch. PRs allowed us to review code, identify errors, and improve collaboration.
- **Issues and Projects**: Tasks were divided into issues to assign team members specific responsibilities. GitHub Projects was used to track progress and ensure timely completion.

Team 1 : Implement the mathematical models for charging/discharging is done by Manjushri.

Team 2 : Create graphs to visualize the results using Matplotlib is done by Vedha.

Team 3: Document the tool and provide a guide for usage is done by Nithyashree S and Sushma N Y.

---

6. Conclusion

This project demonstrates the simulation of capacitor charging and discharging processes in RC circuits using Python and graphical representations. Additionally, Git and GitHub were instrumental in managing code, collaborating effectively, and maintaining the project's progress.