**POWER SYSTEMS LOAD FLOW ANALYSIS**

##### **INFORMATION:**

Solve Power flow Equations(Newton-Raphson/Gauss-Seida)

 Inputs: Bus data, line data

 Output: Bus Voltage, Power Losses, Efficiency

 Libraries: numpy, scipy, matplotlib

 Application: Transmission line &amp; grid analysis

**Source code:**

import numpy as np

import matplotlib.pyplot as plt

# --- Input Data ---

bus\_data = np.array([

    [1, 0, 1.06, 0, 0, 0, 0, 0],        # Slack Bus

    [2, 2, 1.00, 0, 0, 0, 0.2, 0.1],    # PQ Bus

    [3, 2, 1.00, 0, 0, 0, 0.45, 0.15]   # PQ Bus

])

line\_data = np.array([

    [1, 2, 0.02, 0.06, 0.03],

    [1, 3, 0.08, 0.24, 0.025],

    [2, 3, 0.06, 0.18, 0.02]

])

tolerance = 1e-6

max\_iterations = 10

# --- Y-Bus Formation ---

def form\_ybus(line\_data, num\_buses):

    Y = np.zeros((num\_buses, num\_buses), dtype=complex)

    for line in line\_data:

        i, j = int(line[0]) - 1, int(line[1]) - 1

        R, X, B = line[2], line[3], line[4]

        Z = complex(R, X)

        Y\_line = 1 / Z

        Y[i, i] += Y\_line + 1j \* B / 2

        Y[j, j] += Y\_line + 1j \* B / 2

        Y[i, j] -= Y\_line

        Y[j, i] -= Y\_line

    return Y

# --- Gauss-Seidel Load Flow ---

def gauss\_seidel\_load\_flow(bus\_data, line\_data, max\_iter, tol):

    nb = len(bus\_data)

    Ybus = form\_ybus(line\_data, nb)

    V = np.array([bus[2] \* np.exp(1j \* np.radians(bus[3])) for bus in bus\_data])

    P = (bus\_data[:, 4] - bus\_data[:, 6])

    Q = (bus\_data[:, 5] - bus\_data[:, 7])

    for iteration in range(max\_iter):

        V\_prev = V.copy()

        for i in range(nb):

            if bus\_data[i, 1] == 0:

                continue

            sumYV = sum(Ybus[i, j] \* V[j] for j in range(nb) if j != i)

            S = complex(P[i], Q[i])

            V[i] = (1 / Ybus[i, i]) \* ((S.conjugate() / V[i].conjugate()) - sumYV)

        if np.max(np.abs(V - V\_prev)) < tol:

            break

    # Calculate losses

    losses = 0

    for line in line\_data:

        i, j = int(line[0]) - 1, int(line[1]) - 1

        Z = complex(line[2], line[3])

        I = (V[i] - V[j]) / Z

        losses += abs(I)\*\*2 \* Z.real

    # Efficiency

    total\_load = np.sum(bus\_data[:, 6])

    efficiency = (total\_load / (total\_load + losses)) \* 100

    # Output

    print("\n--- Bus Voltages ---")

    for i, v in enumerate(V):

        print(f"Bus {i+1}: |V| = {abs(v):.4f} pu, Angle = {np.degrees(np.angle(v)):.4f}°")

    print(f"\nTotal Power Losses: {losses:.4f} MW")

    print(f"System Efficiency: {efficiency:.2f} %")

    return V, losses, efficiency

# --- Run and Get Results ---

voltages, losses, efficiency = gauss\_seidel\_load\_flow(bus\_data, line\_data, max\_iterations, tolerance)

# --- Plotting Bus Voltages ---

bus\_ids = np.arange(1, len(voltages) + 1)

voltage\_magnitudes = np.abs(voltages)

voltage\_angles = np.degrees(np.angle(voltages))

plt.figure(figsize=(10, 5))

# Plot voltage magnitude

plt.subplot(1, 2, 1)

plt.bar(bus\_ids, voltage\_magnitudes, color='skyblue')

plt.xlabel('Bus Number')

plt.ylabel('Voltage Magnitude (pu)')

plt.title('Bus Voltage Magnitudes')

plt.grid(True)

# Plot voltage angle

plt.subplot(1, 2, 2)

plt.bar(bus\_ids, voltage\_angles, color='salmon')

plt.xlabel('Bus Number')

plt.ylabel('Voltage Angle (degrees)')

plt.title('Bus Voltage Angles')

plt.grid(True)

plt.tight\_layout()

plt.show()

Output:

--- Bus Voltages ---

Bus 1: |V| = 1.0600 pu, Angle = 0.0000°

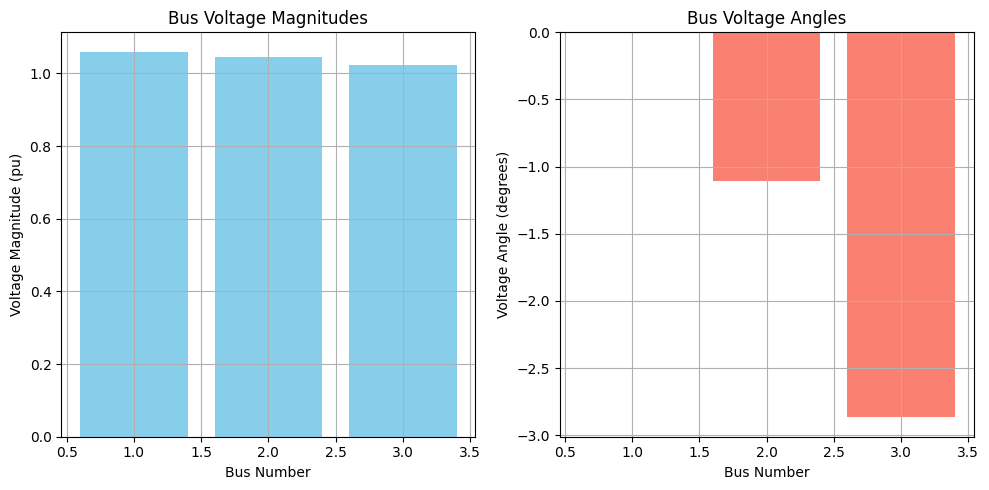
Bus 2: |V| = 1.0443 pu, Angle = -1.1108°

Bus 3: |V| = 1.0224 pu, Angle = -2.8681°

Total Power Losses: 0.0109 MW

System Efficiency: 98.34 %

Graph:



Conclusion:

Load flow analysis helps us understand how electricity moves through a power system. It tells us how much voltage, current, and power is at different parts of the system. This analysis is important to make sure the system runs smoothly, safely, and efficiently. By doing load flow analysis, engineers can plan for future demand, prevent overloads, and improve the overall performance of the power network.