Лабораторная работа № 12

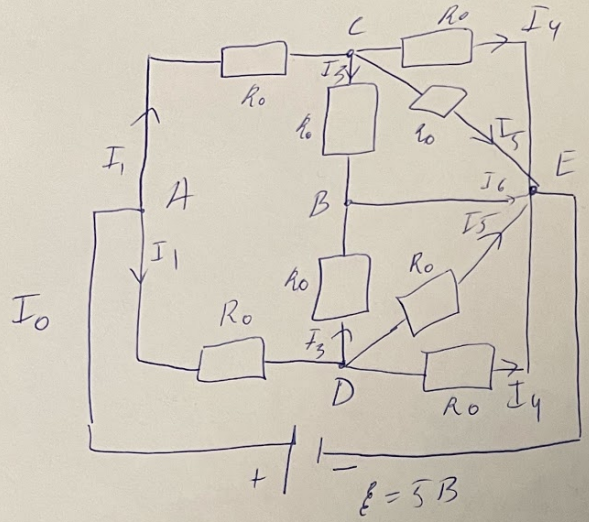
«Применение численных методов в решении расчетных задач на примере метода узловых потенциалов»

Цель работы

Реализовать задачу по вариантам.

Задача

1. Решить задачу Университета Штутгарта.
2. Решить задачу по вариантам.



$$I_0 = 2I_1$$

$$I_1 = I_3 + I_4 + I_5$$

$$2I_3 = I_6$$

$$2(I_4 + I_5) + I_6 = I_0 = 2I_1$$

$$I_1 = \frac{\varphi_A - \varphi_C}{R_0}$$

$$I_3 = \frac{\varphi_C - \varphi_B}{R_0}$$

$$I_4 = \frac{\varphi_C - \varphi_E}{R_0}$$

$$I_5 = \varphi_C - \varphi_E$$

$$I_6 = \frac{\varphi_B - \varphi_E}{R_0}$$

$$\begin{cases} \varphi_A + \varphi_B - 4\varphi_C + 2\varphi_E = 0 \\ -3\varphi_B + 2\varphi_C + \varphi_E = 0 \\ -2\varphi_A + 4\varphi_B + 6\varphi_C - 5\varphi_E = 0 \\ \varphi_A - \varphi_E = 5 \end{cases}$$

$$\Downarrow$$

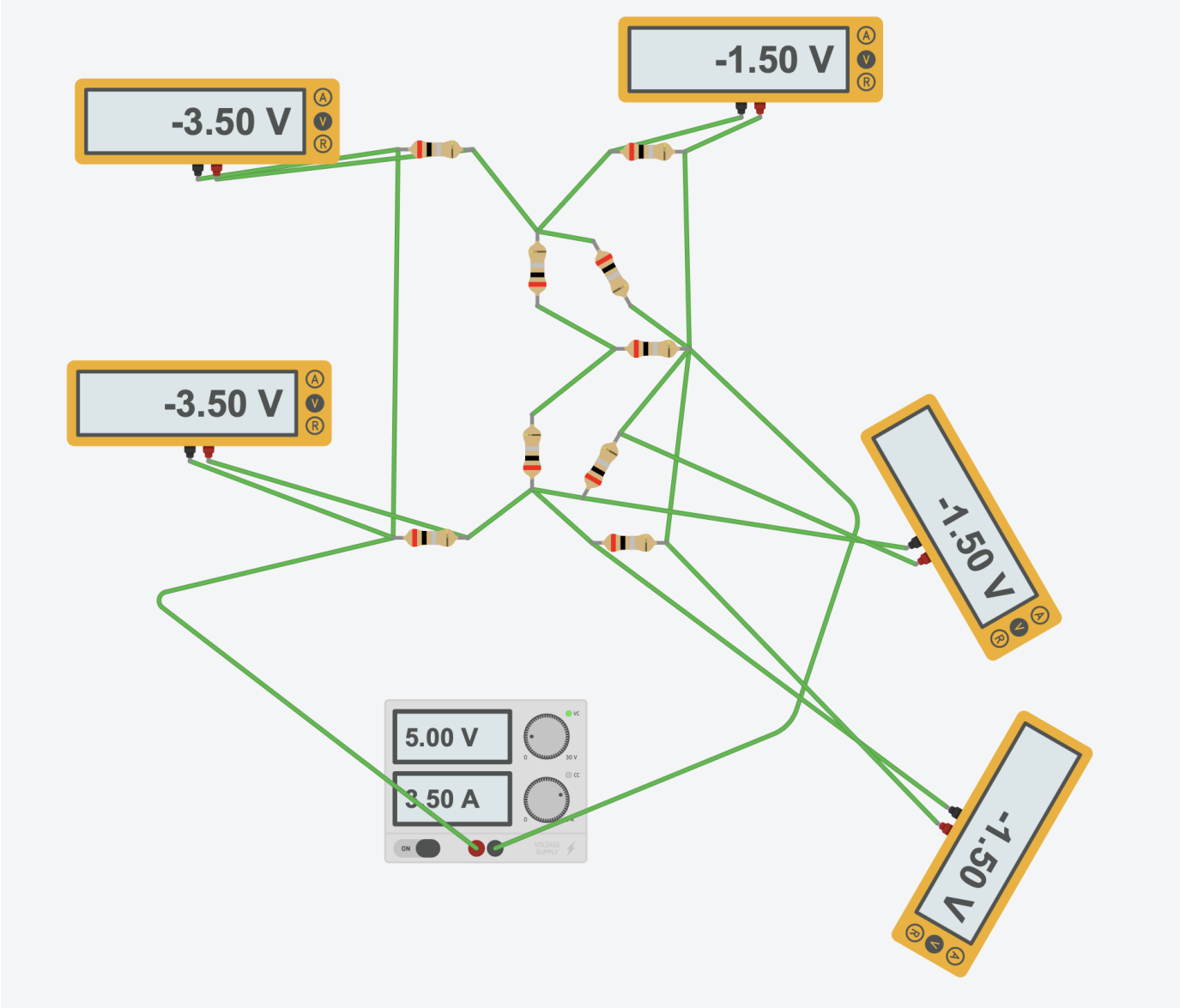
$$\begin{cases} \varphi_A + \varphi_B - 4\varphi_C = 0 \\ -2\varphi_A + \varphi_B + 6\varphi_C = 0 \\ \varphi_A = 5 \end{cases}$$

```
import numpy as np


def prepare_to_gauss(m, v):
```

```
    a = m
    for i in range(0, len(v)):
        a[i].append(v[i])
    return a


def gauss_method(a):
    n = len(a)
    x = [0 for i in range(n)]

    for i in range(n):
        if a[i][i] == 0.0:
            print('Divide by zero detected!')

        for j in range(i + 1, n):
            rat = a[j][i] / a[i][i]

            for k in range(n + 1):
                a[j][k] = a[j][k] - rat * a[i][k]

    x[n - 1] = a[n - 1][n] / a[n - 1][n - 1]
    for i in range(n - 2, -1, -1):
        x[i] = a[i][n]

        for j in range(i + 1, n):
            x[i] = x[i] - a[i][j] * x[j]

        x[i] = x[i] / a[i][i]

    return x


def solve_phi(r0):
    #  PHIa + PHIb - 4PHIc = 0
    # -2PHIa + PHIb + 6PHIc = 0
    #  PHIa                 = 5

    a = [
        [1, 1, -4],
        [-2, 1, 6],
        [1, 0, 0]
    ]
    b = [0, 0, 5]
    x = gauss_method(prepare_to_gauss(a, b))
    phi = {"a": x[0], 'b': x[1], 'c': x[2]}
    print(phi)
    i6 = phi['b'] / r0
    i5 = phi['c'] / r0
    i4 = i5
    i3 = (phi['c'] - phi['b']) / r0
    i1 = (phi['a'] - phi['c']) / r0
    i0 = i1 * 2
    return [i0, i1, i3, i4, i5, i6]
```

```
def solve_shtudgard(r1, r2, r3, r4, r5, r6, r7):
    a = [
        [(1 / r3 + 1 / r6), -(1 / r3), 0, 0],
        [-(1 / r3), (1 / r5 + 1 / r3 + 1 / r2), -(1 / r4), -(1 / r2)],
        [0, -(1 / r4), (1 / r7 + 1 / r4 + 1 / r1), -(1 / r1)],
        [0, -(1 / r2), -(1 / r1), (1 / r2 + 1 / r1)]
    ]
    b = np.array([-0.01, 0, 0, 0.01])
    x = gauss_method(prepare_to_gauss(a, b))
    v = {"1": x[0], '2': x[1], '3': x[2], '4': x[3]}
    print(v)
    i1 = (v['4'] - v['3']) / r1
    i2 = (v['4'] - v['2']) / r2
    i3 = (v['1'] - v['2']) / r3
    i4 = (v['2'] - v['3']) / r4
    i5 = (v['2'] - 0) / r5
    i6 = (v['1'] - 0) / r6
    i7 = (v['3'] - 0) / r7
    return i1, i2, i3, i4, i5, i6, i7


if __name__ == '__main__':
    r0 = 2
    pr = solve_phi(r0)
    for i in range(len(pr)):
        if i > 1:
            print(f"i{i+1}: {pr[i]}")
        else:
            print(f"i{i}: {pr[i]}")

    print()

    st = solve_shtudgard(r1=100000, r2=0.5, r3=0.5, r4 = 100000, r5=100000, r6=1000
    for i in range(len(st)):
        print(f"i{i}: {st[i]}")
```

```
{'a': 5.0, 'b': 1.0, 'c': 1.5}
i0: 3.5
i1: 1.75
i3: 0.25
i4: 0.75
i5: 0.75
i6: 0.5

{'1': -0.0029999900000641532, '2': 0.0019999949999858465, '3': 0.0029999900000
i0: 3.999985000045282e-08
i1: 0.009999960000149999
i2: -0.0099999700001
i3: -9.999950000297175e-09
i4: 1.9999949999858464e-08
i5: -2.9999900000641533e-08
i6: 2.999990000015564e-08
```


-1.50 V
-3.50 V
-3.50 V
-1.50 V
5.00 V
3.50 A
-1.50 V