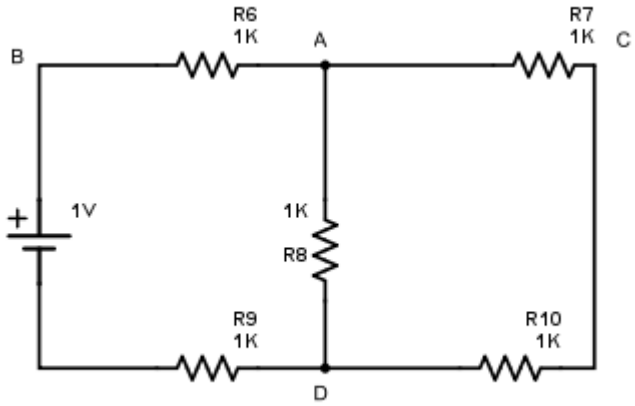
Finding Voltages in an Electrical Circuit

10 points

In this problem, we will set up a linear system to model the current in a simple electrical circuit like this one:



To make this prediction, we will be given the layout of the network and voltages at some points (e.g. at both ends of a current source). To do so, we need a few facts about how electricity behaves.

Ohm's Law

Ohm's law (https://en.wikipedia.org/wiki/Ohm's_law) states that the current I through a conductor between two points is directly proportional to the difference in voltage ΔV between the two points. The constant of proportionality is the resistance of the conductor R, so

$$I = \frac{\Delta V}{R}.$$

Kirchhoff's Current Law

At any node (junction) in an electrical circuit, the sum of currents flowing into that node is equal to the sum of currents flowing out of that node. For instance, at junction A in the above sample circuit, if the current between B and A is I_{BA}, between A and C is I_{AC} and between A and D is I_{AD}, Kirchoff's law gives

$$I_{\mathrm{BA}} = I_{\mathrm{AC}} + I_{\mathrm{AD}}.$$

The current is directed, so $I_{\mathrm{XY}} = -I_{\mathrm{YX}}$ for any neighboring circuit points X and Y. So, we can rewrite the above equation in the form,

$$0 = I_{\mathrm{AB}} + I_{\mathrm{AC}} + I_{\mathrm{AD}}.$$

Equations for Voltages

Our goal will be to determine a voltage at each junction, given voltages at the ground and current-source nodes. We dnote the voltages for junctions A, B, C, and D in the above example circuit as V_A, V_B, V_C, and V_D. Combining Kirchoff's current law with Ohm'slaw, we obtain

$$\frac{V_A - V_B}{R6} + \frac{V_A - V_C}{R7} + \frac{V_A - V_D}{R8} = 0.$$

In the sample circuit, in addition to similar equations for junctions C and D, we also know that $V_B = 1$, since it is the current at the source.

Setting up a Laplacian System

We can represent the circuit as a graph, with a vertex for each junction and edges for each connection. With an appropriate choice of edge-weights, the Laplacian of our weighted graph will provide the linear system we need to solve to determine the voltages. A Laplacian L of a graph with edges $E \in V \times V$ and edge-weights $w \in E \to \mathbb{R}$ is given by

$$L_{ij} = \begin{cases} -w(i,j) & \text{if } (i,j) \in E \\ \sum_{(i,k)\in E} w(i,k) & \text{if } i = j \\ 0 & \text{if } (i,j) \notin E \text{ and } i \neq j \end{cases}$$

The edge-weights in the graph should have magnitude proportional to the inverse of the resistance at each connection.
Modify L appropriately for each ground and source node (note that the current does not obey Kirchoff's law at these nodes). Overall, the modified matrix $\bar{L}$ should be defined so that solving

$$\bar{L}\mathbf{v} = \mathbf{u},$$

where $\mathbf{u}$ are the initial ground/source voltages, yields voltages $\mathbf{v}$ at all nodes/junctions.

Processing the Data

The resistances of the circuit will be given to you in a Python list of dictionaries called `circuit`, e.g.

```
circuit = [
    {1: 1300, 3: 3800, 4: 4900},
    {2: 1500, 3: 900, 8: 2100},
    {3: 1000, 7: 3500}
    ]
```

In this circuit, there is a resistance of 1300 Ohms between node 0 and node 1. There is also a resistance of 3800 Ohms between node 0 and node 3, etc.
More precisely, `circuit[i][j] = k` implies that there is a resistance of `k` Ohms from node `i` to node `j`. Note that you need to read this network description in 'both' directions in a way: If there is a resistance of `k` Ohms between node `i` and node `j`, there is also one between node `j` and node `i`. (The network is 'undirected'.) The second connection will not explicitly be given in `circuit`.
In addition to the resistances, you are given a dictionary `fixed_voltages`. This dictionary gives voltages that are known from the start--such as at the positive and negative terminals of a battery. For example if

```
fixed_voltage[i] = k
```

then the voltage at node i is k Volts. Use this dictionary to construct the vector of right-hand-sides ($\mathbf{u}$ above) and store it in `rhs`. To realize the equation for the fixed voltages, you will need to outright replace that node's equation with one representing `voltages[i] = fixed_voltages[i]` (and clear out any existing, network-related entries).

You should first form the Laplacian matrix L, storing it in `network_mat`. Then modify L to account for the fixed voltages (compute $\bar{L}$), and store the result in `mat`. Finally, solve for the voltages ($\mathbf{v}$ above, using `numpy.linalg.solve`) and store the result in `voltages`. In that array, `voltages[i] = v` means that the voltage at the node `i` is `v`.

INPUT:

- `nnodes` : Number of nodes in circuit.
- `circuit` : List containing `nnodes` dictionaries, that contain directional resistances as decribed above.
- `fixed_voltages` : The voltages at source and ground nodes.
- `plot_circuit(circuit, fixed_voltages, voltages)` : A function that allows you to view the circuit that you have plotted. To use the function call `plot_circuit(circuit,fixed_voltages,voltages)`. The third argument is optional.

OUTPUT:

- `rhs` : Right-hand sides ($\mathbf{w}$) for the system of linear equations.
- `network_mat` : Laplacian matrix induced from the resistances (`circuit`).
- `mat` : Modified Laplacian matrix encoding the coefficients of the linear equations.
- `voltages` : The voltage of each node in the circuit.

Problem set-up code (click to view)

Starter code (click to view)

Answer*

```
import numpy as np
import numpy.linalg as la

# Feel free to run this. It will show you the circuit we are considering.
# (This changes every time.)
plot_circuit(circuit, fixed_voltages)

# Once you've found the voltages, you may use this version:
# plot_circuit(circuit, fixed_voltages, voltages)
```

Press F9 to toggle full-screen mode. Set editor mode in user profile (/profile/).

Save answer | Submit answer for feedback

(You may still change your answer after you submit it.)