SF1550 Numerical Methods

Project
Electrical Networks

Subject:Numerical Methods
Laborant I: Villim Prpic
Laborant II: Jonathan Tadesse
Klass: CTMAT, Engineering Mathematics
Date: 2023-04-23
Professor: Olof Runborg

Contents

1 Background

In this project, we will study electrical networks. The networks consist of several nodes that are interconnected with resistors. An example is shown in the image below, where the nodes are marked with dots and the resistors with rectangles.

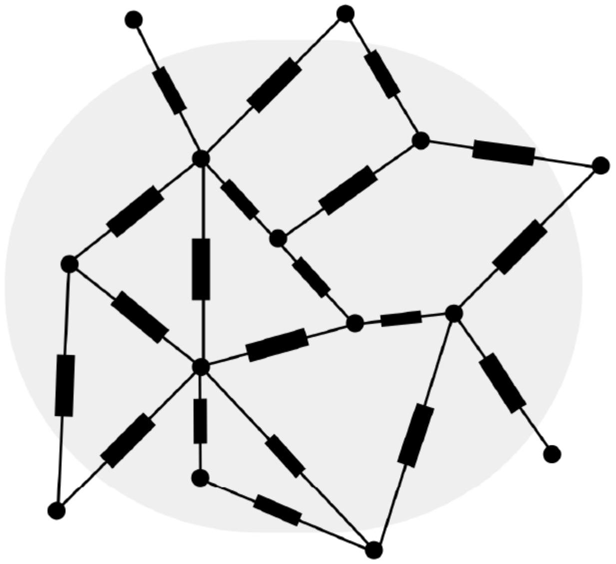

Figure 1: Schematics

We assume that there are n nodes in the network and we call them $x_1, x_2, \ldots, x_n$. Two nodes x_i and x_j can be interconnected with (max) one resistor, which we denote as r_{ij}. We let the electrical potential in node x_j be U_j and the current flowing from node x_i to x_j be I_{ij}. We also define the conductance as $k_{ij} = 1/r_{ij}$. Ohm's law states that

$$I_{ij} = \frac{U_i - U_j}{r_{ij}} = k_{ij}\left(U_i - U_j\right)$$

Note that the current in the opposite direction has the opposite sign, $I_{ji} = -I_{ij}$. Finally, we denote the total current leaving a node as I_j, which is defined as

$$I_i = \sum_{\text{all nodes connected to } i} I_{ij}$$

1.1 Kirchhoff's Laws

[1] **Kirchhoff's Current Law:**

$$\sum_{i=1}^{n} I_i = 0$$

Where n is the number of circuit elements, and I_i is the current flowing through the ith element. The law states that the algebraic sum of all currents entering and leaving a node must be zero.

Kirchhoff's Voltage Law:

$$\sum_{i=1}^{n} V_i = 0$$

Where n is the number of circuit elements, and V_i is the voltage drop across the ith element. The law states that the algebraic sum of all voltages around a closed loop in a circuit must be zero.

[1] Sears, F., Zemansky, M. W., Young, H. D., Freedman, R. A. (2015). University Physics with Modern Physics (14th ed.). Pearson.

2 The Kirchhoff matrix

Let $\boldsymbol{U} = (U_1, \ldots, U_n)^T$ and $\boldsymbol{I} = (I_1, \ldots, I_n)^T$ be vectors with the potentials and currents of all nodes, respectively. We will first show that these vectors are related as

$$\boldsymbol{I} = K\boldsymbol{U}, \quad K \in \mathbb{R}^{n \times n}$$

where the matrix K depends on the conductances k_{ij}. The matrix K is called the Kirchhoff matrix of the network [2] The Kirchhoff matrix, is a mathematical tool we will use to model our electrical network. It is a square matrix of size $n \times n$, where n is the number of nodes in the network.
First, we will determine K in terms of k_{ij} using equations:

$$I_{ij} = \frac{U_i - U_j}{r_{ij}} = k_{ij}\left(U_i - U_j\right)$$
$$I_i = \sum_{\text{all nodes connected to } i} I_{ij}$$

In order to simplify the resulting formulas, we can define k_{ij} for all node pairs with index $i \neq j$, and set $k_{ij} = 0$ when the nodes x_i and x_j are not interconnected.

First, we compute the total current I_i flowing into each node by summing up the currents I_{ij} between node i and all other nodes j.

$$I_i = \sum_{j=1}^{n} I_{ij} = \sum_{j=1}^{n} \left(-k_{ij}U_j + k_{ij}U_i\right) = -\sum_{j=1}^{n} k_{ij}U_j + \sum_{j=1}^{n} k_{ij}U_i.$$

Next we can write a matrix equation that represents the same idea for each node.

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} \sum_{j\neq 1} k_{1j} & -k_{12} & \cdots & -k_{1n} \\ -k_{21} & \sum_{j\neq 2} k_{2j} & \cdots & -k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -k_{n1} & -k_{n2} & \cdots & \sum_{j\neq n} k_{nj} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_n \end{bmatrix}$$

Here a clear candidate for a Kirchhoff matrix emerges.

$$K = \begin{bmatrix} \sum_{j\neq 1} k_{1j} & -k_{12} & \cdots & -k_{1n} \\ -k_{21} & \sum_{j\neq 2} k_{2j} & \cdots & -k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -k_{n1} & -k_{n2} & \cdots & \sum_{j\neq n} k_{nj} \end{bmatrix}$$

Using the matrix representation we can write out the formula for calculating the current I_1.

$$\begin{aligned} I_1 &= \left(\sum_{j\neq 1}^{n} k_{1j}\right) U_1 - k_{12} \cdot U_2 \ldots - k_{1n} \cdot U_n \\ &= \left(k_{12} + k_{13} \cdots k_{1n}\right) U_1 - k_{12} \cdot U_2 \ldots - k_{1n} U_n \\ &= k_{12} \cdot (U_1 - U_2) + k_{13} \cdot (U_1 - U_3) \cdots k_{1n} \left(U_1 - U_n\right) \\ &= \sum_{j=2}^{n} k_{1j} \left(U_1 - U_j\right) \Rightarrow I_i = \sum_{j=1}^{n} I_j \end{aligned}$$

q.e.d.

This is in accordance with Kirchhoff's Current Law. Total current entering a node is equal to the total current

[2] `https://en.wikipedia.org/wiki/Kirchhoff's_theorem`

leaving the node, or in other words, the algebraic sum of currents at any node in a circuit is zero. We will now check that the candidate matrix K is indeed a Kirchhoff's matrix. In general, we say that a matrix $A \in \mathbb{R}^{n\times n}$ with elements $\{a_{ij}\}$ is a Kirchhoff matrix if and only if:

- The matrix is symmetric, $a_{ij} = a_{ji}$
- The row sums of A are zero, $\sum_{j=1}^{n} a_{ij} = 0$
- Off-diagonal elements are either negative or zero, $a_{ij} \leq 0$ when $i \neq j$

First we show that the matrix is symmetric, $a_{ij} = a_{ji}$.

$$K = \begin{bmatrix} \sum_{j\neq 1} k_{1j} & -k_{12} & \cdots & -k_{1n} \\ -k_{21} & \sum_{j\neq 2} k_{2j} & \cdots & -k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -k_{n1} & -k_{n2} & \cdots & \sum_{j\neq n} k_{nj} \end{bmatrix}$$

K is symmetric because $k_{ij} = k_{ji}$ for all i and j.

Second, we show that sum of each row in K is zero, which can be seen from the definition of K_{ii} and K_{ij} for $i \neq j$.

$$\sum_{j=1}^{n} k_{j1} = \sum_{j\neq 1} (k_{j1}) - \sum_{j=2} k_{1j} = \sum_{j=2} (k_{j1} - k_{j1}) = \sum_{j=2} 0$$

q.e.d.

Third, we show that off-diagonal elements are either negative or zero: $K_{ij} \leq 0$ if $i \neq j$ since $k_{ij} \geq 0$ and $K_{ij} = -k_{ij}$.

For the matrix K, we had that the elements were given by:

$$k_{ij} = \begin{cases} \sum_{j\neq i} k_{ij} & \text{om } i = j \\ -k_{ij} & \text{om } i \neq j \end{cases} \quad \text{q.e.d.}$$

Therefore, K satisfies all three properties of a Kirchhoff matrix.

2.1 Schur complement

We now assume that there are $m < n$ external nodes (outside the gray area) and $n-m$ internal nodes. We also arrange the nodes such that the external nodes are $x_1, \ldots, x_m$ and the internal nodes are $x_{m+1}, \ldots, x_n$. We connect the external nodes to voltage sources and want to calculate the current flowing from them. In other words, we assume that $U_1, \ldots, U_m$ are known and we want to calculate $I_1, \ldots, I_m$. We divide $\boldsymbol{U}$ into external and internal nodes as $\boldsymbol{U}^T = \left[\boldsymbol{U}_{\text{external}}^T\, \boldsymbol{U}_{\text{internal}}^T\right]$ where $\boldsymbol{U}_{\text{external}} = (U_1, \ldots, U_m)^T \in \mathbb{R}^m$ and $\boldsymbol{U}_{\text{internal}} = (U_{m+1}, \ldots, U_n)^T \in \mathbb{R}^{n-m}$. Divide $\boldsymbol{I}$ in the same way. We can then write $\boldsymbol{I} = KU$ in block form as:

$$\begin{pmatrix} \boldsymbol{I}_{\text{external}} \\ \boldsymbol{I}_{\text{internal}} \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{U}_{\text{external}} \\ \boldsymbol{U}_{\text{internal}} \end{pmatrix}.$$

For the internal nodes $x_{m+1}, \ldots, x_n$, Kirchhoff's law states that the net current flowing out of a node without a current source is zero, i.e., $\boldsymbol{I}_{\text{internal}} = 0$.
We will show that:

$$\boldsymbol{I}_{\text{external}} = SU_{\text{external}}, \quad S \in \mathbb{R}^{m\times m}$$

where the matrix S depends on K. We will now provide a formula for S in terms of the blocks K_{ij} that holds when K_{22} is invertible. This matrix is called the Schur complement [3] of K for the given partition, or the response matrix of the network.

[3] https://en.wikipedia.org/wiki/Schur_complement

To show that $I_{external} = SU_{external}$, we use the block partitioning of $I = KU$ given in the problem statement:

$$\begin{pmatrix} \boldsymbol{I}_{\text{external}} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{U}_{\text{external}} \\ \boldsymbol{U}_{\text{internal}} \end{pmatrix}$$

Since $I_{internal} = 0$, we have $K_{21}U_{external} + K_{22}U_{internal} = 0$. Solving for $U_{internal}$, we get:

$$\boldsymbol{U}_{\text{internal}} = -K_{22}^{-1} K_{21} \boldsymbol{U}_{\text{external}}$$

By substituting this expression for $U_{internal}$ in the block partitioning of $I = KU$, we obtain:

$$\begin{pmatrix} \boldsymbol{I}_{\text{external}} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{U}_{\text{external}} \\ -K_{22}^{-1} K_{21} \boldsymbol{U}_{\text{external}} \end{pmatrix}$$

By multiplying the matrix multiplication on the right-hand side and equating the components in the resulting vectors, we obtain:

$$\begin{aligned} \boldsymbol{I}_{\text{external}} &= K_{11} \boldsymbol{U}_{\text{external}} + K_{12} \left(-K_{22}^{-1} K_{21} \boldsymbol{U}_{\text{external}} \right) \\ &= \left(K_{11} - K_{12} K_{22}^{-1} K_{21} \right) \boldsymbol{U}_{\text{external}} \end{aligned}$$

Therefore, we have $I_{external} = SU_{external}$, where:

$$S = K_{11} - K_{12} K_{22}^{-1} K_{21}$$

S is the Schur complement of K for the given block partition. This formula holds when K_{22} is invertible.

$$\boldsymbol{I}_{\text{external}} = S \boldsymbol{U}_{\text{external}}, \quad S \in \mathbb{R}^{m \times m}$$

It can be shown that S is also a Kirchhoff matrix. This means that S corresponds to another, smaller network that is equivalent to the original network.

We will show that:

- S is symmetric, i.e., $a_{ij} = a_{ji}$.
- The row sums of S are zero, i.e., $\sum_{j=1}^{n} a_{ij} = 0$, when K_{22} is invertible.

First, S is symmetric.
Since K is symmetric and the partition into block matrices is consistent, S is also symmetric.

$$S^\top = S$$

$$\begin{gathered} S^\top = (K_{11} - K_{12} K_{22}^{-1} K_{21})^\top = K_{11}^\top - (K_{12} K_{22}^{-1} K_{21})^\top = \\ = K_{11}^\top - K_{21}^\top (K_{22}^{-1})^\top K_{12}^\top = \end{gathered}$$

Since $K_{12}^T = K_{21}$. And K_{11} and K_{22} symmetrical.

$$= K_{11} - K_{12} K_{22}^{-1} K_{21} = S$$

Second, the row sums of S are zero, i.e., $\sum_{j=1}^{n} a_{ij} = 0$, when K_{22} is invertible.

$$S = K_{11} - K_{12} K_{22}^{-1} K_{21}$$

We know that:

$$U_{internal} = K_{22}^{-1} K_{21} U_{external}$$

By Kirchhoff's law we know that:

$$U_{internal} = -U_{external} \Rightarrow K_{22}^{-1} K_{21} = -1$$

$$S = K_{11} - (K_{12} \cdot (-1))S = K_{11} + K_{12}$$

We already know that the row sum of $K_{11} + K_{12} = 0$. Thus we have proven that the row sum of S is also equal to 0.

For a more mathematical approach we can:

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$K_{11} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = -K_{12} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$K_{21} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = -K_{22} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$\Leftrightarrow -K_{22}^{-1} K_{21} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = - \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

We can show that K_{22} is positive semi-definite if the network is connected, i.e., if there exists a path in the network between all pairs of nodes. Every row in K_{22} has at least one positive element. This implies that K_{22} is diagonally dominant. A symmetric diagonally dominant matrix with positive diagonal entries is positive definite. Therefore, since K_{22} is symmetric and positive definite, it is positive semi-definite.

Lemma 1: A matrix is called symmetric if and only if its off-diagonal elements satisfy the following conditions:

$$(1) \begin{cases} \forall x_i \in \mathbb{C} \\ x_{ij} = x_{ji} \end{cases}$$

Lemma 2: A matrix is called positive-definite if:

$$\begin{cases} \forall x_{ij} > 0, i = j \\ \forall x_{ij} < 0, \text{ otherwise} \end{cases}$$

Since $k_{22} \in K$ where $k_{22} \in R^{(n-m)x(n-m)}$, we can write that the sum of elements in each row is greater than or equal to 0. By combining Lemma (1), (2), and (3), we have, by definition, that a matrix that satisfies these conditions and is positive semi-definite.

We know that for Kirchhoff matrix is symmetric, that off-diagonal elements are either zero or negative, that the diagonal is strictly positive and that the row sum has to be zero. By assuming that the network is connected we know that each row in a matrix needs to contain at least one element e.i that we can't have a row with all zeroes. We can show that K_{22} is invertible if the network is connected. This can be done by proving that the matrix K_{22} is positive-definite.

An $n \times n$ symmetric real matrix M is said to be positive-definite if $\mathbf{x}^\top M \mathbf{x} > 0$ for all non-zero $\mathbf{x}$ in $\mathbb{R}^n$.

$$M \text{ positive-definite} \iff \quad \mathbf{x}^\top M \mathbf{x} > 0 \text{ for all } \mathrm{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$$

In our Kirchhoff matrix:

$$K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}$$

$$U^T K U = \frac{1}{2} \sum_{j \neq i} k_{ij} \left(U_i - U_j\right)^2 \geqslant 0$$

We observe two sub-matrices K_{21} and K_{22} and want to show that K_{22} is positive-definite. Note the the sum of the row of K has to be zero. Imagine a case where we have:

$$= \begin{bmatrix} \alpha & \beta & 0 & 0 \\ 0 & 0 & \gamma & \delta \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} \epsilon & \kappa & \rho \\ 0 & \zeta & \mu \\ \xi & \theta & \eta \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} = x + y \neq 0$$

It can never be equal to 0 because at least one inner node has to be connected to at least one outer node.

3 Application to an electrical network

The Schur complement of the Kirchhoff matrix can be used to simplify the analysis of networks with multiple inputs and outputs. This matrix represents the reduced network, which can be used to determine the output signals of the network given the input signals. We now consider a specific network given in the Figure 2 below. It consists of six nodes and five resistances. The nodes x_1, x_2, x_3, x_4 are external nodes.

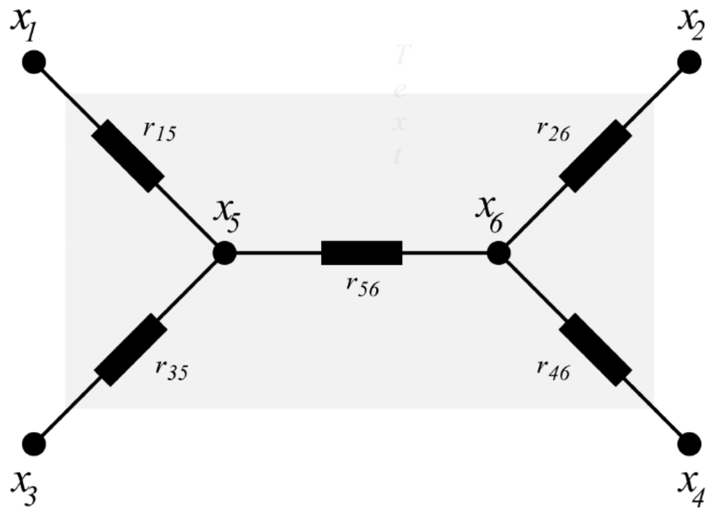

Figure 2: Network

We can collect all the conductance's into a vector k, defined as $k = (k_{15}, k_{26}, k_{56}, k_{35}, k_{46})^T$ and then write MATLAB functions that, given k, calculate the corresponding Kirchhoff matrix $K = K(k)$ and response matrix $S = S(\boldsymbol{k})$. The MATLAB code for the corresponding Kirchhoff matrix and the Response matrix can be found in the appendix[4]. To give a more concrete example we can assume that given the conductances:

$$k = \begin{pmatrix} k_{15} \\ k_{26} \\ k_{56} \\ k_{35} \\ k_{46} \end{pmatrix} = \begin{pmatrix} 9.0 \cdot 10^{-3} \\ 2.5 \cdot 10^{-3} \\ 1.0 \cdot 10^{-3} \\ 0.5 \cdot 10^{-3} \\ 4.5 \cdot 10^{-3} \end{pmatrix} \quad \Omega^{-1}.$$

[4]For the code please refer to section 6.0.1 and 6.0.2 in the appendix.

We connect x_1 to 9 volts and ground the other external nodes. That is, $U_1 = 9$ and $U_2 = U_3 = U_4 = 0$. Using the theory and the MATLAB code provided in the appendix we can compute the currents in the external nodes.

$$S = \begin{pmatrix} 0.0012 & -0.0003 & -0.0004 & -0.0005 \\ -0.0003 & 0.0017 & -0.0000 & -0.0014 \\ -0.0004 & -0.0000 & 0.0005 & -0.0000 \\ -0.0005 & -0.0014 & -0.0000 & 0.0019 \end{pmatrix} \Omega^{-1}$$

$$\begin{pmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{pmatrix} = \begin{pmatrix} 0.0012 & -0.0003 & -0.0004 & -0.0005 \\ -0.0003 & 0.0017 & -0.0000 & -0.0014 \\ -0.0004 & -0.0000 & 0.0005 & -0.0000 \\ -0.0005 & -0.0014 & -0.0000 & 0.0019 \end{pmatrix} \begin{pmatrix} 9 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 10.8 \cdot 10^{-3} \\ -2.7 \cdot 10^{-3} \\ -3.6 \cdot 10^{-3} \\ -4.5 \cdot 10^{-3} \end{pmatrix} A$$

Observe that the sum of the currents is zero. The sum of the currents into a node equals the sum of the currents out of the node. In this case, the current entering node x_1 is equal to the sum of the currents leaving x_1 and by repeating this logic the sum of currents leaving the whole network. This result is a consequence of the fact that the network is connected and the Kirchhoff's laws hold, as we have shown.

4 The inverse problem

We now consider the inner (gray) part of the network as a black box. We assume that only the external nodes are accessible and that all resistors inside the box are unknown. Through measurements on the external nodes, we want to determine these resistors. This is called an inverse problem, where the model that determines $\boldsymbol{I}$ given k and U is known and easy to compute, but the inverse, finding k from I and U, is more difficult. Similar problems arise in, for example, tomography, where a wave is sent through a material, and from the wave's observed behavior, one wants to determine the material; there, the equation for the wave given the material is well known, but the inverse relationship is very complicated.

$$\boldsymbol{I} = K(\boldsymbol{k})\boldsymbol{U}$$

where $\boldsymbol{I} \in \mathbb{R}^n$, $\boldsymbol{U} \in \mathbb{R}^m$, and $\boldsymbol{k} \in \mathbb{R}^d$ are the current, voltage, and resistance vectors, respectively, and $K(\boldsymbol{k})$ is the Kirchhoff matrix depending on the resistances. We want to solve the inverse problem

$$\boldsymbol{k} = F(\boldsymbol{I}, \boldsymbol{U})$$

where F is some function that maps the current and voltage to the resistance.

We connect 1 volt to an external node, ground the rest, and measure the currents from the external nodes using an ammeter. The procedure is repeated for all four external nodes. The measured currents are given in the table below. The values are given in milliamperes.

Node	1	2	3	4
I_1	1.10	−0.19	−0.37	−0.55
I_2	−0.18	1.11	−0.15	−0.77
I_3	−0.38	−0.16	0.95	−0.40
I_4	−0.57	−0.80	−0.42	1.75

From these measurements, you are required to determine the conductances in the network. This can be done by solving an overdetermined nonlinear system of equations using the Gauss-Newton method [5].

Let k be the unknown conductances. The above measurement series gives a matrix S_0 that would satisfy $S(\boldsymbol{k}) = S_0$ if the measurements were completely accurate. Note that due to measurement errors, S_0 is not perfectly symmetric, and its row and column sums are not exactly zero. We can formulate the problem as an overdetermined nonlinear system of equations $\boldsymbol{F}(\boldsymbol{k}) \approx 0$, where the elements of $\boldsymbol{F}$ consist of the elements of S and S_0, where each element in S is a function of $\boldsymbol{k}$. Thus we can write MATLAB functions that calculate $\boldsymbol{F}$ and its Jacobian matrix J.

$$\frac{d}{ds}A(s)B(s) = \frac{dA(s)}{ds}B(s) + A(s)\frac{dB(s)}{ds}, \quad \text{and especially to} \quad \frac{d}{ds}A(s)A(s)^{-1} = \frac{d}{ds}I = 0$$

We can solve the inverse problem by following these steps:

1. We define the function F that calculates the residual vector of the nonlinear equation system $\boldsymbol{F}(\boldsymbol{k}) \approx 0$, where $\boldsymbol{k}$ is the vector of unknown conductances, and $\boldsymbol{F}$ is given by $\boldsymbol{F}(\boldsymbol{k}) = S(\boldsymbol{k}) - S_0$.
2. We define the function J that calculates the Jacobian matrix of $\boldsymbol{F}$.
3. Then we choose a suitable initial guess, for example $k = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T$.
4. By implementing the Gauss-Newton algorithm with a stopping criterion based on the norm of the residual vector and the maximum number of iterations.
5. We can compute the resulting conductances and resistances, the sum of squared residuals, and the number of iterations.

Results[6]

The solution for the conductances is: $k = \begin{bmatrix}1.6021 & 1.3821 & 4.3456 & 1.225 & 4.0351\end{bmatrix}^T$.
The corresponding resistances are: $r = \begin{bmatrix}0.62418 & 0.72355 & 0.23012 & 0.8163 & 0.24782\end{bmatrix}^T$.
The sum of squared residuals is 0.0001257.
The algorithm converged after 5 iterations.

[5] `https://en.wikipedia.org/wiki/Gauss-Newton_algorithm`
[6] For the code please refer to section 6.0.3 in the appendix.

Assume now that we have an ammeter with a margin of error of 2%. All values in S_0 therefore have an uncertainty of $\pm 2\%$. Using experimental error analysis we can determine the uncertainty that this leads to in the calculated resistances.

The idea behind calculating the uncertainty is to see how much the conductances change when each element of the matrix is perturbed by 2%. We can do this by calculating the perturbed conductances for each perturbed matrix and then subtracting the normal conductances from the perturbed conductances. Finally, the absolute differences are summed up to obtain a difference. The element with the highest value in this vector corresponds to the most uncertain resistance.

Results[7]

The perturbation leads to the following uncertainties in the calculated resistances:

r_{15}: 0.0217 Ω
r_{26}: 0.0250 Ω
r_{56}: 0.0268 Ω
r_{35}: 0.0267 Ω
r_{46}: 0.0177 Ω

The results suggests that the resistance corresponding to the element r_{56} exhibits the highest degree of uncertainty. This result can be logically reasoned, as r_{56} is located in the center of the network and is connected to multiple nodes.

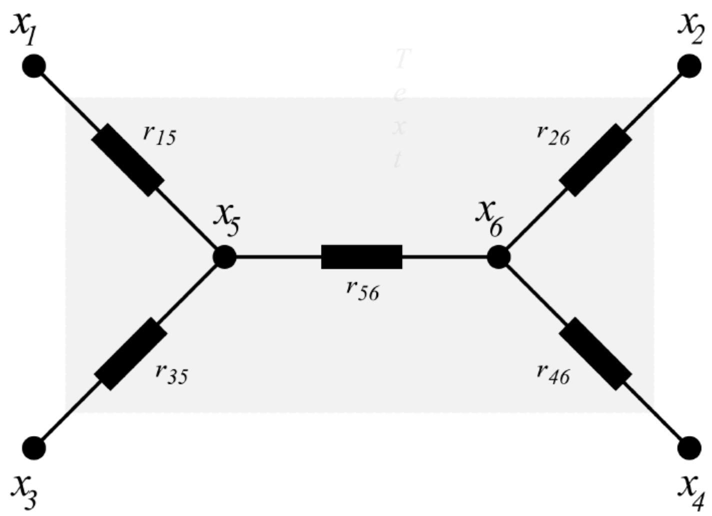

Figure 3: Network

[7] For the code please refer to section 6.0.4 in the appendix.

5 Optimization

Finlay, assume we want to design a circuit with the specification that $S(\boldsymbol{k}) = B_0 + \eta B_1$ where:

$$B_0 = \begin{pmatrix} 1.0 & -0.3 & -0.3 & -0.4 \\ -0.3 & 1.0 & 0.0 & -0.7 \\ -0.3 & 0.0 & 0.9 & -0.6 \\ -0.4 & -0.7 & -0.6 & 1.7 \end{pmatrix} \cdot 10^{-3}, \quad B_1 = \begin{pmatrix} 0.3 & 0.3 & -0.2 & -0.4 \\ 0.3 & 0.3 & -0.4 & -0.2 \\ -0.2 & -0.4 & 0.1 & 0.5 \\ -0.4 & -0.2 & 0.5 & 0.1 \end{pmatrix} \cdot 10^{-3}$$

The parameter η is free and can be chosen in the interval $\eta \in [0, 1]$. The best choice of $\boldsymbol{k}$ now depends on η and we denote the least squares solution to $S(\boldsymbol{k}) \approx B_0 + \eta B_1$ as $\boldsymbol{k} = \boldsymbol{k}(\eta)$. We also let $\boldsymbol{r}(\eta) = (r_1(\eta), \ldots, r_5(\eta))^T$ be the corresponding resistances. The cost of a resistor with resistance r ohms is assumed to follow the price curve $P(r)$ given by the formula:

$$P(r) = 1 + 10 \left(\frac{r}{1000}\right)^2$$

The total price of the resistors for a given η is therefore:

$$P_{\text{tot}}\,(\eta) = \sum_{j=1}^{5} P\left(r_j(\eta)\right).$$

Our goal is to find the optimal value of $\eta \in [0, 1]$ that minimizes the total cost $P_{\text{tot}}(\eta)$. We will use the Matlab solution from the previous task to calculate r, given η, and then use the golden section search method[1] to find the optimal value η.

Results[8]

- Optimal η: 0.719645
- Total cost: 20.932183
- Error square sum for optimal η: 1.136868×10^{-13}
- Number of iterations in golden section search: 29
- Tolerance for golden section search: 1.000000×10^{-13}

[8]For the code please refer to section 6.0.5 in the appendix.

6 Appendix-MATLAB code

6.0.1 KirchhoffMatrix

```
function K = KirchhoffMatrix(k)
% k: vektor med konduktanser
% K: Kirchoffmatrisen f or det givna n atverket
%k=(k_{15}, k_{26}, k_{56}, k_{35}, k_{46})^T
n = 6; % antal noder
K = zeros(n); % initialisera K-matrisen
% fyll i Kirchoffmatrisen
K(1,1) = k(1);
K(2,2) = k(2);
K(3,3) = k(4);
K(4,4) = k(5);
K(5,5) = k(1)+k(4)+k(3);
K(6,6) = k(3)+k(2)+k(5);
K(1,5)= -k(1);
K(5,1)= -k(1);
K(2,6) = -k(2);
K(6,2) = -k(2);
K(5,6) = -k(3);
K(6,5) = -k(3);
K(3,5) = -k(4);
K(5,3) = -k(4);
K(4,6) = -k(5);
K(6,4) = -k(5);
end
```

6.0.2 ResponsMatrix

```
function S = ResponsMatrix(k)
% k: vektor med konduktanser
% S: responsmatrisen f or det givna n atverket
n = 6; % antal noder
m = 4; % antal yttre noder
% ber akna delmatriserna
K = KirchhoffMatrix(k);
K11 = K(1:m,1:m);
K12 = K(1:m,m+1:n);
K21 = K(m+1:n,1:m);
K22 = K(m+1:n,m+1:n);
% ber akna responsmatrisen
S = K11 - K12*(K22\K21);
end
```

6.0.3 The Inverse problem - Part I

```matlab
clc
close all

% Parametrar
tol = 1e-6;
max_iter = 100;
h = 1e-6;

% Startgissning
k_guess = ones(5,1);

S0 = [ 1.10 -0.19 -0.37 -0.55;
      -0.18  1.11 -0.15 -0.77;
      -0.38 -0.16  0.95 -0.40;
      -0.57 -0.80 -0.42  1.75];

% Gauss-Newton iteration
iter = 0;
while iter < max_iter
  Fk = F_vector(k_guess, S0);
  Jk = Jacobian_numerical(k_guess, S0, h);
  delta_k = -Jk \ Fk;
  k_guess = k_guess + delta_k;

  if norm(delta_k) < tol
    break;
  end

  iter = iter + 1;
end

k_solution = k_guess;
felkvadrat_sum = sum(F_vector(k_solution, S0).^2) / 16;

% Calculate the resistances
r_solution = 1 ./ k_solution;
%------------------Störningsberäkning--------------------
% Teori? [y-y']/yA*A^-1 *[x-x']/x
% S0 har felgräns 2%, räkna ut först k_solution sedan stör med felgräns,
% [x_exp-x]  approximation till felgräns

disp(['Solution (conductances) is: ' num2str(k_solution')])
disp(['Resistancerna är: ' num2str(r_solution')])
disp(['Felkvadratsumma: ' num2str(felkvadrat_sum)])
disp(['Antal iterationer: ' num2str(iter)])
```

6.0.4 The Inverse problem - Part II

```
%-----beräkna ny S0 med störda felgränserna------

S0=[ 1.10 -0.19 -0.37 -0.55;
      -0.18  1.11 -0.15 -0.77;
      -0.38 -0.16  0.95 -0.40;
      -0.57 -0.80 -0.42  1.75];

k_normal=perturbation(S0);

k_difference=zeros(5,1);
for i = 1:size(S0,1)
    for j = 1:size(S0,2)
        %element = S0(i,j)+0.02;
        element = S0(i,j)*1.02;
        S_stord=S0;
        S_stord(i,j)=element;
        k_stord=perturbation(S_stord);
        k_difference=k_difference+abs(k_normal-k_stord);
        % do something with the element here
    end
end
index=max(k_difference);


a=sum(k_difference);

disp(['Felgränsen är ungefär: ' num2str(a*100,2),'%']);

%----------- 1/rij-->kij , Iij

k_difference
```

6.0.5 Optimization code

```
% Definiera B0 och B1
B0 = [1.0, -0.3, -0.3, -0.4;-0.3, 1.0, 0.0, -0.7;-0.3, 0.0, 0.9, -0.6;-0.4, -0.7, -0.6, 1.7] * 1e-3;
B1 = [0.3, 0.3, -0.2, -0.4;0.3, 0.3, -0.4, -0.2;-0.2, -0.4, 0.1, 0.5;-0.4, -0.2, 0.5, 0.1] * 1e-3;

% Definiera funktionen som beräknar resistanserna givet eta
f = @(eta) perturbation(B0 + eta * B1);

% Definiera kostnadsfunktionen
P = @(r) 1 + 10 * (r / 1000).^2;

% Definiera totala kostnadsfunktionen givet eta
Ptot = @(eta) sum(P(f(eta  )));

% Definiera toleransen och max antal iterationer för gyllene snittet-sökning
tol = 1e-6;
max_iter = 100;

% Gyllene snittet-sökning
tau = (sqrt(5) - 1) / 2; % Gyllene snittet

a = 0;
b = 1;
x = a + (1-tau)*(b-a);
y = a + tau*(b-a);
```

```
fa=sum(P(f(a)));
fb=sum(P(f(b)));
fx=sum(P(f(x)));
fy=sum(P(f(y)));

iter = 0;
while abs(b - a) > tol && iter < max_iter
    %x = a + (1-tau)*(b-a);
    %y = b - tau*(b-a);
    if (fx > fy)
        a = x; fa=fx;
        x=y; fx=fy;
        y = a + (b-a)*tau; fy=sum(P(f(y)));

    else
       b=y; fb=fy;
       y=x; fy=fx;
       x=a+(b-a)*(1-tau); fx=sum(P(f(x)));
    end
    iter = iter + 1;

end

% Hitta optimala eta och resistanser
eta_opt = (a + b) / 2;
r_opt = f(eta_opt);
Ptot_opt = Ptot(eta_opt);

% Beräkna felkvadratsumman för eta_opt
S_opt = B0 + eta_opt * B1;
F_opt = F_vector(r_opt, S_opt);
felkvadratsum_opt = sum(F_opt);

% Skriv ut resultatet
fprintf('Optimala eta: %f\n', eta_opt);
fprintf('Resistanser: [%f, %f, %f, %f, %f]\n', r_opt);
fprintf('Total kostnad: %f\n', Ptot_opt);
fprintf('Felkvadratsumma för optimala eta: %e\n', felkvadratsum_opt);
fprintf('Antal iterationer i gyllene snittet-sökning: %d\n', iter);
fprintf('Tolerans för gyllene snittet-sökning: %e\n', tol);
```

6.0.6 F vector

```
function F = F_vector(k, S0)

  S = ResponsMatrix(k);
  F = S - S0;
  F = F(:);  % omforma F till en vektor
end
%creates the vector valued function that is necessary for the gauss-newton
% in main file
```

6.0.7 Jacobian numerical

```
function J = Jacobian_numerical(k, S0, h)
  n = length(k);
  J = zeros(16, n);
  F0 = F_vector(k, S0);
  for i = 1:n
    k_plus_h = k;
    k_plus_h(i) = k_plus_h(i) + h;
```

```
    F_plus_h = F_vector(k_plus_h, S0);
    J(:, i) = (F_plus_h - F0) / h;
  end
end
```

6.0.8 Perturbation

```
function [solution]=perturbation(S0)

tol = 1e-6;
max_iter = 100;
h = 1e-6;

% Startgissning
k_guess = ones(5,1);

% Gauss-Newton iteration
iter = 0;
while iter < max_iter
  Fk = F_vector(k_guess, S0);
  Jk = Jacobian_numerical(k_guess, S0, h);
  delta_k = -Jk \ Fk;
  k_guess = k_guess + delta_k;

  if norm(delta_k) < tol
    break;
  end

  iter = iter + 1;
end

solution = 1./k_guess;
end
```