

# Resistor networks in R: the ResistorArray package

Robin K. S. Hankin

---

## Abstract

This paper introduces the **ResistorArray** package of R routines, for analysis of resistor networks. An earlier version of this vignette was published as [Hankin \(2006\)](#).

*Keywords:* Resistors, Resistor Arrays.

---

Many elementary physics courses show how resistors combine in series and parallel (Figure 1); the equations are

$$R_{\text{series}} = R_1 + R_2 \quad (1)$$

$$R_{\text{parallel}} = \frac{1}{R_1^{-1} + R_2^{-1}} \quad (2)$$

However, these rules break down for many systems such as the Wheatstone bridge (Figure 2); the reader who doubts this should attempt to apply equations 1 and 2 and find the resistance between nodes 1 and 4 in the general case. This paper introduces **ResistorArray**, an R ([R Core Team 2012](#)) package to determine resistances and other electrical properties of such networks.

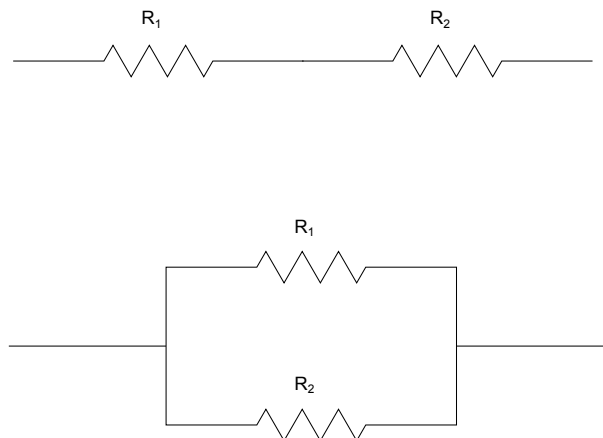


Figure 1: Two resistors in series (top) and parallel (bottom)

Although this paper uses the language of electrical engineering, the problem considered is clearly very general: many systems are composed of isolated nodes between which some quantity flows and the steady states of such systems are generally of interest. Package **ResistorArray** has been applied to such diverse problems as the diffusion of nutrients among fungal hyphae networks, the propagation of salinity between (moored) oceanographical buoys, and hydraulic systems such as networks of sewage pumps. The general problem of determining

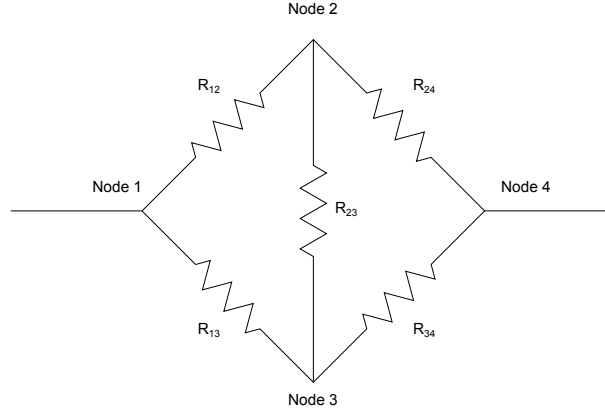


Figure 2: The Wheatstone bridge

the resistance between two nodes of a resistor network requires matrix techniques. Consider a network of  $n$  nodes, with node  $i$  connected to node  $j$  by a resistor of resistance  $R_{ij}$ . Then the network has a “conductance matrix”  $\mathcal{L}$  with

$$\mathcal{L}_{ij} = \begin{cases} -1/R_{ij} & \text{if } i \neq j \\ \sum_{k \neq j} 1/R_{kj} & \text{if } i = j \end{cases} \quad (3)$$

Thus  $\mathcal{L}$  is a symmetrical matrix, whose row sums and column sums are zero (and is therefore singular). Then the analogue of Ohm’s law (viz  $V = IR$ ) would be

$$\mathcal{L}\mathbf{v} = \mathbf{i} \quad (4)$$

where  $\mathbf{v} = (v_1, \dots, v_n)$  is a vector of potentials and  $\mathbf{i} = (i_1, \dots, i_n)$  is a vector of currents; here  $i_p$  is the current flow in to node  $p$ . Equation 4 is thus a restatement of the fact that charge does not accumulate at any node. Each node of the circuit may either be fed a known current<sup>1</sup> and we have to calculate its potential; or it is maintained at a known potential and we have to calculate the current flux into that node to maintain that potential. There are thus  $n$  unknowns altogether. Thus some elements of  $\mathbf{v}$  and  $\mathbf{i}$  are known and some are unknown. Without loss of generality, we may partition these vectors into known and unknown parts:  $\mathbf{v}' = (\mathbf{v}^{k'}, \mathbf{v}^{u'})$  and  $\mathbf{i}' = (\mathbf{i}^{u'}, \mathbf{i}^{k'})$ . Thus the known elements of  $\mathbf{v}$  are  $\mathbf{v}^k = (v_1, \dots, v_p)'$ : these would correspond to nodes that are maintained at a specified potential; the other elements  $\mathbf{v}^u = (v_{p+1}, \dots, v_n)'$  correspond to nodes that are at an unknown potential that we have to calculate. The current vector  $\mathbf{i}$  may similarly be decomposed, but in a conjugate fashion; thus elements  $\mathbf{i}^u = (i_1, \dots, i_p)'$  correspond to nodes that require a certain, unknown, current to be fed into them to maintain potentials  $\mathbf{v}^k$ ; the other elements  $\mathbf{i}^k = (i_{p+1}, \dots, i_n)'$  would correspond to nodes that have a known current flowing into them and whose potential we seek. Equation 4 may thus be expressed in terms of a suitably partitioned matrix equation:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}' & \mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{v}^k \\ \mathbf{v}^u \end{pmatrix} = \begin{pmatrix} \mathbf{i}^u \\ \mathbf{i}^k \end{pmatrix} \quad (5)$$

<sup>1</sup>This would include a zero current: in the case of the Wheatstone bridge of Figure 2, nodes 2 and 3 have zero current flux so  $i_2$  and  $i_3$  are known and set to zero.

where, in R idiom,  $\mathbf{A}=\mathbf{L}[1:p,1:p]$ ,  $\mathbf{B}=\mathbf{L}[1:p,(p+1):n]$ , and  $\mathbf{C}=\mathbf{L}[(p+1):n,(p+1):n]$ .  
Straightforward matrix algebra gives

$$\mathbf{v}^u = \mathbf{C}^{-1} (\mathbf{i}^k - \mathbf{B}'\mathbf{v}^k) \quad (6)$$

$$\mathbf{i}^u = (\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}')\mathbf{v}^k + \mathbf{B}\mathbf{C}^{-1}\mathbf{i}^k \quad (7)$$

Equations 6 and 7 are implemented by `circuit()`.

## 1. Package ResistorArray in use

Consider the Wheatstone Bridge, illustrated in Figure 2. Here the resistance between node 1 and node 4 is calculated; all resistances are  $1\ \Omega$  except  $R_{34}$ , which is  $2\ \Omega$ . This resistor array may be viewed as a skeleton tetrahedron, with edge 1 missing. We may thus use function `tetrahedron()` to generate the conductance matrix; this is

```
> L <- tetrahedron(c(1, 1, Inf, 1, 1, 2))
```

```
      [,1] [,2] [,3] [,4]
[1,]    2  -1 -1.0  0.0
[2,]   -1    3 -1.0 -1.0
[3,]   -1  -1  2.5 -0.5
[4,]    0  -1 -0.5  1.5
```

Observe that  $\mathbf{L}[1,4]=\mathbf{L}[4,1]=0$ , as required. The resistance may be determined by function `circuit()`; there are several equivalent methods. Here node 1 is earthed—by setting it to zero volts—and the others are given a floating potential; viz  $\mathbf{v}=\mathbf{c}(0,\mathbf{NA},\mathbf{NA},\mathbf{NA})$ . Then one amp is fed in to node 4, zero amps to nodes 2 and 3, and node 1 an unknown current, to be determined; viz  $\mathbf{currents}=\mathbf{c}(\mathbf{NA},0,0,1)$ . The node 1-to-node 4 resistance will then be given by the potential at node 4:

```
> circuit(L,v=c(0, NA, NA, NA),currents=c(NA, 0, 0, 1))
```

```
$potentials
```

```
[1] 0.0000000 0.5454545 0.4545455 1.1818182
```

```
$currents
```

```
[1] -1 0 0 1
```

Thus the resistance is about  $1.181818\ \Omega$ , comparing well with the theoretical value of  $117/99\ \Omega$ . Note that the system correctly returns the net current required to maintain node 1 at  $0\ \text{V}$  as  $-1\ \text{A}$  (which is clear from conservation of charge). The potential difference between nodes 2 and 3 indicates a nonzero current flow along  $R_{23}$ ; currents through each resistor are returned by function `circuit()` if argument `give.internal` is set to `TRUE`. Package **ResistorArray** solves a problem that is frequently encountered in electrical engineering: determining the electrical properties of resistor networks. The techniques presented here are applicable to

many systems composed of isolated nodes between which some quantity flows; the steady states of such systems generally have an electrical analogue. This paper explicitly presents a vectorized solution method that uses standard numerical techniques, and discusses the corresponding R idiom.

## References

- Hankin RKS (2006). “Resistor networks in R: Introducing the **ResistorArray** package.” *R News*, **6**(2), 52–54.
- R Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/>.

### **Affiliation:**

Robin K. S. Hankin  
Auckland University of Technology  
2-14 Wakefield Street  
Auckland  
New Zealand  
E-mail: [hankin.rob@gmail.com](mailto:hankin.rob@gmail.com)