Description

Electrical properties of resistor networks such as the resistance between any two nodes and the current that passes along any wire

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>ResistorArray</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.0</td>
</tr>
<tr>
<td>Date:</td>
<td>2007-01-23</td>
</tr>
<tr>
<td>License:</td>
<td>GPL</td>
</tr>
</tbody>
</table>

The package uses matrix methods to determine resistances between nodes of a resistor network.

Author(s)

Robin K. S. Hankin
Maintainer: <r.hankin@noc.soton.ac.uk>

References

Rnews 2006.

Examples

```
# resistance between opposite corners of a skeleton cube:
resistance(cube(),1,7)  # known to be 5/6 Ohm

# resistance of a Jacob's ladder:
resistance(ladder(60),1,2)  # should be about (sqrt(5)-1)/2

# Google aptitude test:
array.resistance(1,2,15,17)  # analytical answer 4/pi-1/2
```

array.resistance          Resistance between two arbitrary points on a regular lattice of unit resistors
Description

Given two points on a regular lattice of electrical nodes joined by unit resistors (as created by `makefullmatrix()`), returns the resistance between the two points, or (optionally) the potentials of each lattice point when unit current is fed into the first node, and the second is earthed.

Usage

```r
array.resistance(x.offset, y.offset, rows.of.resistors, cols.of.resistors, give.pots = FALSE)
```

Arguments

- `x.offset` - Earthed node is at (0,0), second node is at (x.offset, y.offset)
- `y.offset` - Earthed node is at (0,0), second node is at (x.offset, y.offset)
- `rows.of.resistors` - Number of rows of resistors in the network (positive integer)
- `cols.of.resistors` - Number of columns of resistors in the network (positive integer)
- `give.pots` - Boolean, with TRUE meaning to return a matrix of potentials of the electrical nodes, and FALSE meaning to return the resistance between the origin and the current input node

Details

Note that the electrical network is effectively toroidal.

Author(s)

Robin K. S. Hankin

See Also

`makefullmatrix`

Examples

```r
jj.approximate <- array.resistance(1,2,15,17,give=FALSE)
jj.exact <- 4/pi-1/2
print(jj.exact - jj.approximate)

persp(array.resistance(4,0,14,16,give=TRUE),theta=50,r=1e9,expand=0.6)
```
circuit

Mensurates a circuit given potentials of some nodes and current flow into the others

Description

Given a conductance matrix, a vector of potentials at each node, and a vector of current inputs at each node (NA being interpreted as “unknown”), this function determines the potentials at each node, and the currents along each edge, of the whole circuit.

Usage

circuit(L, v, currents=0, use.inverse=FALSE, give.internal=FALSE)

Arguments

L Conductance matrix
v Vector of potentials; one element per node. Elements with NA are interpreted as “free” nodes, that is, nodes that are not kept at a fixed potential. The potential of these nodes is well defined by the other nodes in the problem. Note that such nodes must have current inputs (which may be zero) specified by argument currents
currents Vector of currents fed into each node. The only elements of this vector that are used are those that correspond to a node with free potential (use NA for nodes that are at a specified potential). The idea is that each node has either a specified voltage, or a specified current is fed into it; not both, and not neither. Observe that feeding zero current into a node at free potential is perfectly acceptable (and the usual case)
use.inverse Boolean, with default FALSE meaning to use solve(A, b) and TRUE meaning to use solve(A), thus incurring the penalty of evaluating a matrix inverse, which is typically to be avoided if possible.

The default option should be faster most of the time, but YMMV

give.internal Boolean, with TRUE meaning to return also a matrix showing the node-to-node currents, and default FALSE meaning to omit this

Value

Depending on the value of Boolean argument give.internal, return a list of either 2 or 4 elements:

potentials A vector of potentials. Note that the potentials of the nodes whose potential was specified by input argument v retain their original potentials; symbolically all(potentials[!is.na(v)] == v[!is.na(v)])
currents Vector of currents required to maintain the system with the potentials specified by input argument v
internal.currents
Matrix showing current flow from node to node. Element \([i,j]\) shows current flow from node \(i\) to node \(j\). This and the next two elements only supplied if argument `give.internal` is TRUE.

power
The power dissipated at each edge.

total.power
Total power dissipated over the resistor network.

Note
The SI unit of potential is the “Volt”; the SI unit of current is the “Ampere.”

Author(s)
Robin K. S. Hankin

See Also
resistance

Examples

# reproduce first example on `?cube`:
\[
v <- c(0, rep(NA, 5), 1, NA)
circuit(cube(), v)
circuit(cube(), v + 1000)
\]

# problem: The nodes of a skeleton cube are at potentials 1, 2, 3, ..., volts. What current is needed to maintain this? Ans:
circuit(cube(), 1:8)

# sanity check: maintain one node at 101 volts:
circuit(cube(), c(rep(NA, 7), 101))

# now, nodes 1-4 have potential 1, 2, 3, 4 volts. Nodes 5-8 each have one Amp shoved in them. What is the potential of nodes 5-8, and what current is needed to maintain nodes 1-4 at their potential? # Answer:
v <- c(1:4, rep(NA, 4))
currents <- c(rep(NA, 4), rep(1, 4))
circuit(cube(), v, currents)

# Now back to the resistance of a skeleton cube across its sqrt(3) diagonal. To do this, we hold node 1 at 0 Volts, node 7 at 1 Volt, # and leave the rest floating (see argument `v` below); we # seek the current at nodes 1 and 7 # and insist that the current flux into the other nodes is zero # (see argument `currents` below):
circuit(l = cube(), v = c(0, NA, NA, NA, NA, NA, 1, NA), currents = c(NA, 0, 0, 0, 0, NA, 0, 0))
# Thus the current is 1.2 ohms and the resistance (from V=IR)
# is just 1/1.2 = 5/6 ohms, as required.

## Specifications

### Description

Various conductance matrices for simple resistor configurations including a skeleton cube

### Usage

- `cube(x=1)`
- `octahedron(x=1)`
- `tetrahedron(x=1)`
- `dodecahedron(x=1)`
- `icosahedron(x=1)`

### Arguments

- **x**  
  Resistance of each edge. See details section

### Details

Function `cube()` returns an eight-by-eight conductance matrix for a skeleton cube of 12 resistors. Each row/column corresponds to one of the 8 vertices that are the electrical nodes of the compound resistor.

In one orientation, node 1 has position 000, node 2 position 001, node 3 position 101, node 4 position 100, node 5 position 010, node 6 position 011, node 7 position 111, and node 8 position 110.

In `cube()`, `x` is a vector of twelve elements (a scalar argument is interpreted as the resistance of each resistor) representing the twelve resistances of a skeleton cube. In the orientation described below, the elements of `x` correspond to `R_{12}`, `R_{14}`, `R_{15}`, `R_{23}`, `R_{26}`, `R_{34}`, `R_{37}`, `R_{48}`, `R_{56}`, `R_{58}`, `R_{67}`, `R_{78}` (here `R_{ij}` is the resistance between node `i` and `j`). This series is obtainable by reading the rows given by `platonic("cube")`. The pattern is general: edges are ordered first by the row number `i`, then column number `j`.

In `octahedron()`, `x` is a vector of twelve elements (again scalar argument is interpreted as the resistance of each resistor) representing the twelve resistances of a skeleton octahedron. If node 1 is “top” and node 6 is “bottom”, the elements of `x` correspond to `R_{12}`, `R_{13}`, `R_{14}`, `R_{15}`, `R_{23}`, `R_{25}`, `R_{26}`, `R_{34}`, `R_{36}`, `R_{45}`, `R_{46}`, `R_{56}`. This may be read off from the rows of `platonic("octahedron")`.

To do a Wheatstone bridge, use `tetrahedron()` with one of the resistances `Inf`. As a worked example, let us determine the resistance of a Wheatstone bridge with four resistances one ohm and one of two ohms; the two-ohm resistor is one of the ones touching the earthed node.

To do this, first draw a tetrahedron with four nodes. Then say we want the resistance between node 1 and node 3; thus edge 1-3 is the infinite one. `platonic("tetrahedron")` gives us the order of the nodes,
edges: 12, 13, 14, 23, 24, 34. Thus the conductance matrix is given by \( jj \leftarrow \text{tetrahedron}(c(2, \text{Inf}, 1, 1, 1, 1)) \) and the resistance is given by \( \text{resistance}(jj, 1, 3) \) [compare the analytical answer of \( \frac{117}{99} \) ohms].

**Author(s)**

Robin K. S. Hankin

**References**


**Examples**

```r
resistance(cube(), 1, 7)  # known to be 5/6 ohm
resistance(cube(), 1, 2)  # known to be 7/12 ohm
resistance(octahedron(), 1, 6)  # known to be 1/2 ohm
resistance(octahedron(), 1, 5)  # known to be 5/12 ohm
resistance(dodecahedron(), 1, 5)
```

---

**currents**  
*Calculates currents in an arbitrary resistor array*

**Description**

Calculates currents in an arbitrary resistor array

**Usage**

```r
currents(L, earth.node, input.node)
currents.matrix(L, earth.node, input.node)
```

**Arguments**

- **L**: Lagrangian conductance matrix  
- **earth.node**: Number of node that is earthed (that is, at a potential of zero)  
- **input.node**: Number of node that has current put into it (a notional one Amp)

**Details**

The methods used by the two functions are different; see documentation for \( \text{resistance}() \) for further details on input args 2 and 3.
Value

Function `currents()` returns a three column matrix, each row of which corresponds to an edge. The first two columns show the node numbers specifying the edge, and the third shows the current flowing along it.

Function `currents.matrix()` uses a different method to return a matrix of the same size as the conductance matrix. Each element of the returned matrix shows the current flowing along the specified edge.

Note

This function is essentially a simplified version of `circuit()`.

Author(s)

Robin K. S. Hankin

Examples

```r
currents(cube(),1,7)
currents.matrix(cube(),1,7)
```

#check above solution: print out the currents flowing into each node:
zapsmall(apply(currents.matrix(cube(),1,7),1,sum))

<table>
<thead>
<tr>
<th>hypercube</th>
<th>Conductance matrix of a Boolean hypercube</th>
</tr>
</thead>
</table>

Description

Returns the conductance matrix of an n-dimensional hypercube

Usage

`hypercube(n)`

Arguments

- `n` Integer giving the dimension of the hypercube

Details

The row and columnnames give the coordinates of each node (which are in binary order)

Value

Returns a conductance matrix
Note

In the case of a 3D cube, the nodes are in a different order from that returned by cube() (which uses Maple’s scheme).

Author(s)

Robin K. S. Hankin

References

http://f2.org/maths/resnet/

See Also

cube

Examples

hypercube(4)

resistance(hypercube(5),1,32)  # cf exact answer of 8/15
resistance(hypercube(5),1,2)   # cf exact answer of n <= 5; (2^n-1)/(n*2^(n-1))=31/80

Description

A potentially infinite resistor network. Consider node 1 to be Earth. Nodes 2, . . . , n are each connected to node 1 by a resistor. For 1 < i < n, node i is connected to node i + 1.

Usage

ladder(n, x = 1, y = 1, z = NULL)

Arguments

n  Number of nodes
x  Resistance of resistors connected to node 1 (earth). Standard recycling rules are used
y  Resistance of the other resistors (ie those not connected to earth). Standard recycling rules are used
z  Resistance of all resistors in the network. If non-NULL, x and y are discarded

Value

Returns a standard conductance matrix
Author(s)
Robin K. S. Hankin

See Also

cube, series

Examples

# Resistance of an infinite Jacob's ladder with unit resistors is known
# to be (sqrt(5)-1)/2:

phi <- (sqrt(5)-1)/2
resistance(ladder(20),1,2) - phi
resistance(ladder(60),1,2) - phi

Wu(ladder(20))[1,2]=phi

# z is the resistance of all the resistors:

ladder(n=8,z=1/(1:13))

# See how node 1 is the "earth", with resistors of conductance 1,2,...,7
# connecting to nodes 2-8. Then nodes 5 & 6, say, are connected by a
# resistor of conductance 11.

makefullmatrix      Conductance matrix for a lattice of unit resistors

Description

Conductance matrix for a lattice of unit resistors

Usage

makefullmatrix(R, C)
makefullmatrix_strict(R, C,toroidal)

Arguments

R   Number of rows of nodes
C   Number of columns of nodes
toroidal  Boolean, with TRUE meaning to return a toroidally connected lattice, and FALSE meaning to return a lattice with edges
Details

The array produced by `makefullmatrix_strict(rLcLtrueI` is toroidally connected.

Function `makefullmatrix()` is not entirely straightforward. The array produced is sort of toroidally connected. I regard this function as the canonical one because it is more elegant (see example image). Consider, for concreteness, the case with four rows and seven columns of nodes giving 28 nodes altogether. Number these columnwise so the top row is 1,5,9,13,17,21,25. Then number $n$ corresponds to the row $n$ and column $n$ of the returned matrix.

Now, ‘interior’ nodes are as expected: node 6, for example, is connected to 2,5,10,7. And the wrapping is as expected in the horizontal: 1-25, 2-26, 3-27, and 4-28, are all connected.

However, the vertical wrapping is not as might be expected. One might expect node 9, say, to be connected to 5,10 13,12; but in fact node 9 is connected to nodes 5,8,10,13. So there is a Hamiltonian path comprising entirely of vertical connections (function `makefullmatrix_strict(rLcLtrueI` returns the “expected” adjacency graph).

For the arrays returned by functions documented here, one can determine pairwise resistances using function `array.resistance()`.

Value

Returns matrix of size $RC \times RC$. Note that this matrix is singular.

Author(s)

Robin K. S. Hankin

See Also

`array.resistance`

Examples

```r
makefullmatrix(3,3)
image(makefullmatrix(4,7))  # A beautiful natural structure
image(makefullmatrix_strict(4,7,TRUE))  # A dog's breakfast
```

platonic

Adjacency of platonic solids

Description

Gives the adjacency indices of the five Platonic solids.

Usage

`platonic(a)`
resistance

Arguments

a String containing name of one of the five Platonic solids, viz “tetrahedron”, “cube”, “octahedron”, “dodecahedron”, “icosahedron”

Details

Returns a two column matrix a, the rows of which show the two vertices of an edge. Only edges with a[i,1]<i[1,2] are included.

For the dodecahedron and icosahedron, the nodes are numbered as per Maple’s scheme.

Author(s)

Robin K. S. Hankin

See Also

cube

Examples

platonic("octahedron")

<table>
<thead>
<tr>
<th>resistance</th>
<th>Resistance for arbitrarily connected networks of resistors</th>
</tr>
</thead>
</table>

Description

Given a resistance matrix, return the resistance between two specified nodes.

Usage

resistance(A, earth.node, input.node, current.input.vector=NULL, give.pots = FALSE)

Arguments

A Resistance matrix
earth.node Number of node that is earthed
input.node Number of node at which current is put in: a nominal 1 Amp
current.input.vector Vector of currents that are fed into each node. If supplied, overrides the value of input.node, and effectively sets give.pots to TRUE because if various currents are fed into the network at various points, the concept of “resistance” becomes meaningless.

Setting this argument to c(0,..,0,1,0,..0) (where the “1” is element jj) is equivalent to not setting current.input.vector and setting input.node to jj.
Resistance:

give.pots Boolean, with TRUE meaning to return the potential of each node (out.node being at zero potential); and default FALSE meaning to return just the resistance between in.node and out.node.

Details

The function’s connection to resistor physics is quite opaque. It is effectively a matrix version of Kirchoff’s law, that the (algebraic) sum of currents into a node is zero.

Note

This function is essentially a newbie wrapper for circuit(), which solves a much more general problem. The function documented here, however, is clearer and (possibly) faster; it also gives an explicit resistance if give.pots is not set.

Use function currents() (or currents.matrix()) to calculate the currents flowing in the resistor array.

Author(s)

Robin K. S. Hankin

References


See Also

array.resistance

Examples

resistance(cube(), earth.node=1, input.node=7) # known to be 5/6 ohm
resistance(cube(), 1, 7, give=TRUE)
series

Conductance matrix for resistors in series

Description

Conductance matrix for resistors of arbitrary resistance in series

Usage

series(x)

Arguments

x  The resistances of the resistors.

Details

Note: if length(x)=n, the function returns a conductance matrix of size n+1 by n+1, because n resistors in series have n+1 nodes to consider.

Author(s)

Robin K. S. Hankin

See Also

cube

Examples

## Resistance of four resistors in series:

resistance(series(rep(1,5)),1,5)  ##sic!  FOUR resistors have FIVE nodes

## What current do we need to push into a circuit of five equal
## resistors in order to maintain the potentials at 1v, 2v, ..., 6v?

circuit(series(rep(1,5)),v=1:6)  ##(obvious, isn't it?)

## Now, what is the resistance matrix of four nodes connected in series
## with resistances 1,2,3 ohms?

Wu(series(1:3))  ##Yup, obvious again.
**SquaredSquare**

**A Squared square**

**Description**

A resistor network corresponding to a squared square

**Usage**

data(SquaredSquare)

**Format**

Returns a conductance matrix

**Details**

The nodes are ordered so that the potentials are in increasing order.

**Source**

Bollobas 1998

**Examples**

data(SquaredSquare)
resistance(SquaredSquare,1,13) # should be 1

circuit(L=SquaredSquare, currents=c(NA,rep(0,11),1), v=c(0,rep(NA,12)))$potentials
# should be in increasing order

---

**Wu**

**Wu's resistance matrix**

**Description**

Returns a matrix $M$ with $M[i,j]$ is the resistance between nodes $i$ and $j$.

**Usage**

Wu(L)

**Arguments**

L Laplacian conductance matrix
Details

Evaluates Wu's resistance matrix, as per his theorem on page 6656.

Value

Returns a matrix of the same size as $\mathbf{L}$, but whose elements are the effective resistance between the nodes.

Note

In the function, the sum is not from 2 to $n$ as in Wu, but from 1 to $n-1$, because `eigen()` orders the eigenvalues from largest to smallest, not smallest to largest.

Author(s)

Robin K. S. Hankin

References


See Also

resistance

Examples

```
Wu(cube())
```
```
Wu(cube())[1,2] - resistance(cube(),1,2)
```
```
Wu(series(1:7))  # observe how resistance between, say, nodes 2 and 5 is 9 (=2+3+4)
```
Index

*Topic array
  array resistance, 2
  circuit, 4
  cube, 6
  currents, 7
  ladder, 9
  makefullmatrix, 10
  platonic, 11
  resistance, 12
  series, 14
  Wu, 15
*Topic datasets
  SquaredSquare, 15
*Topic math
  hypercube, 8
*Topic package
  ResistorArray-package, 2

array resistance, 2, 11, 13

circuit, 4
cube, 6, 9, 10, 12, 14
currents, 7
dodecahedron (cube), 6
hypercube, 8
icosahedron (cube), 6
ladder, 9
makefullmatrix, 3, 10
makefullmatrix_strict (makefullmatrix), 10
octahedron (cube), 6
platonic, 11
resistance, 5, 12, 16

ResistorArray (ResistorArray-package), 2
ResistorArray-package, 2
series, 10, 14
SquaredSquare, 15
Squaredsquare (SquaredSquare), 15
squaredsquare (SquaredSquare), 15
tetrahedron (cube), 6
Wheatstone (cube), 6
wheatstone (cube), 6
Wu, 15