Determining the Shape of a Resistor Network

by Peter Engrav
September 1, 1989
University of Washington

Section 1.1 -- Introduction

We give a method for finding a map from Dirichlet data (voltages on the boundary nodes) to Neumann data (currents through the boundary nodes) for general resistor networks. We then present, without proof, a method for determining the shape of some networks.

Section 1.2 -- Definitions

We define a **resistor network** A to be a set of r resistors and a set of n nodes. Each of the nodes lies at a point in space where one or more resistors terminate, and each of the resistors connects exactly two nodes. We let

$$N = \{N_1, N_2, \ldots, N_n\}$$

be the set of nodes,

$$R = \{\{R_{1,1}, R_{1,2}\}, \{R_{2,1}, R_{2,2}\}, \ldots, \{R_{r,1}, R_{r,2}\}\}$$

be the set of resistors, and

$$G = \{G_1, G_2, \ldots, G_r\}$$

be the set of the conductances of the resistors. (Conductance is the multiplicative inverse of resistance.) A node N_i is a **neighbor** of another node N_j if the set $\{i,j\}$ is in R; that is, if $R_{k,1} = i$ and $R_{k,2} = j$ for some k in $[1,r]$. All nodes in N must have at least one neighbor; those with exactly one neighbor are called **boundary nodes** and those with more than one are called **interior nodes**. Thus, N is the union of two disjoint subsets, the set

$$B = \{B_1, B_2, \ldots, B_b\}$$

of b boundary nodes and the set

$$I = \{I_1, I_2, \ldots, I_{n-b}\}$$

of interior nodes. An example of a resistor network is shown in Figure 1.1.

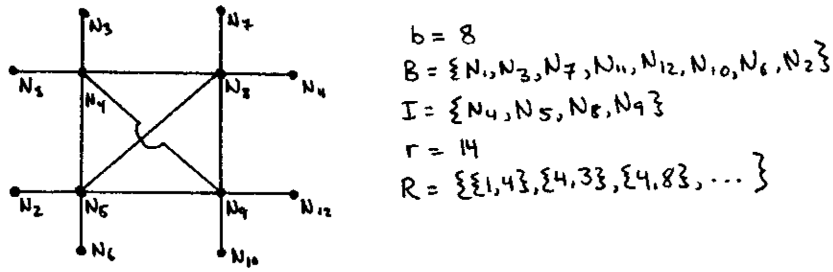


Figure 1.1

Section 2 -- Finding a Map from Voltage to Current Data

Were we to actually construct a physical network we could attach electronic equipment to measure the voltage and current at each boundary node. We would like to find a function relating these two quantities. Currents are related only to differences in voltage, however, so there exist multiple combinations of voltages which give rise to the same currents. Thus we can only hope to discover a function from the boundary voltages to the boundary currents and not the other way around. Without proof, we assume that the currents are linear in the voltages, so this function may be expressed as a b by b matrix L, where each entry $L_{i,j}$ is the current that flows into boundary node B_i per volt at boundary node B_j.

In order that this matrix be useful we must have numbered the boundary nodes in a logical manner. For the most part we will be considering **planer networks**; that is, networks all the nodes and resistors of which lie in a plane. For such networks it is sensible to number the boundary nodes clockwise around the network.

Section 2.1 -- Finding the Matrix L

Let us assume for the moment that we can determine the boundary currents for a specific set of boundary voltages. (A method will be given in Section 2.2.) Then we can find L a column at a time. To find the ith column, we set the voltage at boundary node B_i to 1 and the voltage at all other boundary nodes to 0. Solving for the resultant boundary currents, $I(B_j)$ for j in [1,b], we let $L_{i,j} = I(B_j)$. Repeated application of this procedure generates the entire L matrix.

Section 2.2 -- Finding Boundary Currents from Boundary Voltages

We now substantiate the assumption made in the previous section. To find the currents flowing through each boundary node we need to know the voltages at each end of the boundary resistors. We are given the voltage at each boundary node, $U(B_i)$ for i in [1,b], but we must calculate the voltage at the single neighbor of each boundary node. To find these potentials we will need to solve for the voltage at each interior node, $U(I_i)$ for i in [1,n-b]. For each interor node we can set up an equation using Kirchhoff's Law, which states that the total current flowing into each node must be zero. The current flowing into I_i from a neighbor N_j is a known constant (the conductance of the resistor connecting I_i to N_j) times the difference in potential of the two nodes, $U(N_j)-U(I_i)$. If N_j is a boundary node then this difference is just a known constant minus one of the variables for which we are looking. On the other hand, if N_j is an interior node then the voltage drop is just the difference of two of these variables. Summing the four incoming currents and setting this function equal to zero gives for each interior node I_i a linear equation in from one to five variables, depending on how many of I_i's neighbors are boundary nodes. Taking these equations simultaneously we generate a system of n-b linear equations in n-b unknowns. Solving gives $U(I_i)$ for all i in [1,n-b].

We can then easily calculate the current flowing into each boundary node B_i using the equation,

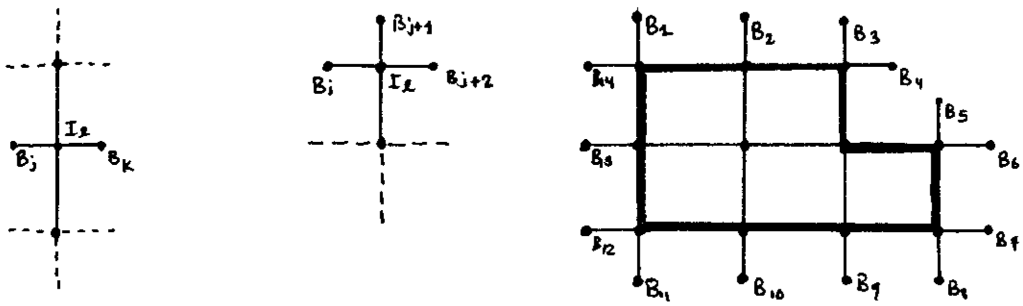
$$I(i) = G_j[U(B_i)-U(I_k)],$$

where j is the number of the resistor that connects B_i to its single neighbor and I_k is that neighbor.

Section 3 -- The Inverse Problem, Determining A from L

The problem is significantly more difficult in reverse. Curtis and Morrow [1] give an algorithm which solves for G from L if the shape of the network (the sets N and C) is given and of a specific form, called **rectangular**. It is likely that a program to find G, N, and C from L for any resistor network cannot be written. Perhaps, however, we may succeed for some networks more general than rectangular.

Let a **regular resistor network** R be a planer resistor network A with the following additional properties. First, every interior node I_i has exactly four neighbors, one in each of the four compass directions from I_i. The distance between any two neighboring interior nodes is 1. To prevent boundary nodes from overlapping, the distance between each boundary node and its single neighbor is $^1/_3$. Finally, each interior node can have among its four neighbors at most two boundary nodes, and if it has two then they must be sequential. Figures 3.1.a and 3.1.b show resistor networks that are not regular; the first has two neighboring boundary nodes, but they are not sequential. The second has too many neighboring boundary nodes. Figure 3.1.c is a regular resistor network.



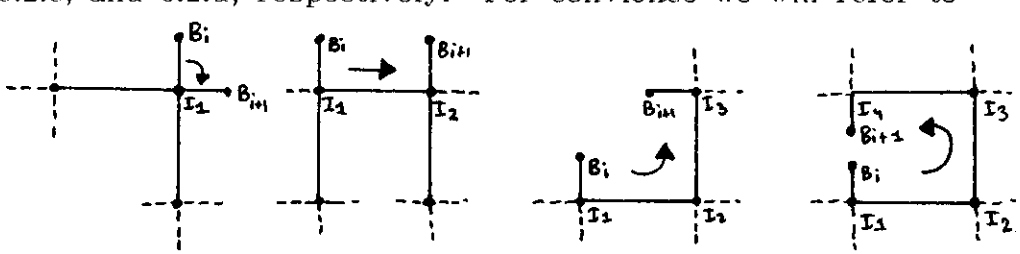
Figures 3.1.a, 3.1.b, 3.1.c

To further simplify matters, we shall ignore G and seek only to recover the shape of R, the sets N and C.

Section 3.1 -- The Boundary Curve of R

To find the shape of a regular resistor network R it is necessary only to find its boundary curve. This was in fact the motivation behind the definition of a regular network. The **boundary curve dR** of a regular network R is the curve generated by connecting, in order, the neighbors of the boundary nodes of R. In Figure 3.1.c the boundary curve is emphasized. Note that the restrictions which prevent the networks shown Figures 3.1.a and 3.1.b from being regular keep the boundary curve from doubling back on itself. For a regular network, dR is always simple closed curve.

The shortest path connecting any pair of sequential boundary nodes can pass through 1, 2, 3, or 4 interior nodes. Each of these gives rise to specific construction, shown in Figures 3.2.a, 3.2.b, 3.2.c, and 3.2.d, respectively. For convience we will refer to



Figures 3.2.a, 3.2.b, 3.2.c, 3.2.d

these situations as, respectively, a **right turn**, a **straight**, a **left**

turn, and a **U-turn**. Curtis and Morrow's rectangular networks [1] are regular networks with only right turns and straights permitted. We will not be so bold as to include U-turns, but will allow left turns. All we need to determine, then, is whether the boundary curve of R turns right, turns left, or continues straight between each pair of sequential boundary nodes.

Section 3.2 -- Overview of Method for Determining Shape of R

First we find all the right turns; this, it will be shown, is relatively straightforward. Since the boundary curve is simple and closed, if there are N_R right turns there must be $N_L = N_R - 4$ left turns. We will construct a list of N_P possible sites for left turns and then simply test all the ways to place N_L left turns in N_P sites. For each combination, we will check to make sure the resultant boundary curve is indeed closed and simple; if it is, we call the network with this boundary curve a **solution network**. For a given network R, we hope to make N_P small enough that there is only one solution network.

Section 3.3 -- Finding Right Turns

It is comparably easy to find right turns; whenever there is a right turn between sequential boundary nodes B_i and B_{i+1}, the ith and i+1st columns of the matrix L will be multiples of each other except perhaps in rows i and i+1.

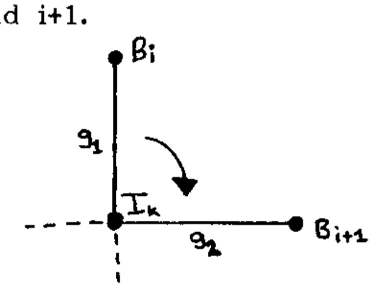


Figure 3.3

Let R be a regular grid network with a right turn between boundary nodes B_i and B_{i+1}. The relavent portion of R is shown in Figure 3.3. Consider $L_{i,j}$ and $L_{i+1,j}$ where j is the number of any row except i or i+1. $L_{i,j}$ is the current that flows into the network through node Bi if the voltage at node B_j, $U(B_j)$, is set to 1 and the voltage at all other boundary nodes is set to 0. Specifically, $U(B_i) = U(B_{i+1}) = 0$. No matter what the unpictured portion of R looks like, the voltage at I_k will stabilize to some value, $U(I_k)$. The current flowing into the network through nodes B_i and B_{i+1} is given by Ohm's Law;

$I(B_i) = (U(B_i)-U(I_k))g_1$ and

$I(B_{i+1}) = (U(B_{i+1})-U(I_k))g_2$, but

$U(B_i) = U(B_{i+1}) = 0$, so

$I(_i) = -U(I_k)g_1$ and $I(B_{i+1}) = -U(I_k)g_2$. Thus

$I(B_i) = (g_2/g_1)I(B_{i+1})$; that is,

$L_{i,j} = (g_2/g_1)L_{i+1,j}$.

Thus, all we need to do to find right turns is find sequential columns i and i+1 of the L matrix which are multiples of each other except in rows i and i+1. We can then easily determine N_R, the number of right turns. N_L, the number of left turns, must be four less than N_R.

Section 3.4 -- Testing Possible Networks

As a first guess we will assume that the left turns might belong anywhere we haven't already put a right turn. Since there are b boundary nodes in all, there must be $N_P = b - N_R$ possible sites for the left turns. The N_L left turns and $b - (N_L + N_R)$ straights must be distributed among these sites. There are $(N_P!)/(N_P-NL_)!$ ways to place N_L left turns in N_P sites.

Each candidate boundary curve dS must be tested to make sure that it represents the boundary of some regular network. We verify that dS never overlaps itself except to begin and end at the same point. If dS is simple and closed, the network S is called a **solution network.** Applying this algorithm to actual networks quickly shows that we need to eliminate some of the possible sites for left turns; we would hope to find a single solution network S, but in fact we obtain rather many for each L matrix.

We therefore present a criterion by which to eliminate some of the possible sites and decrease N_P. That this method works is left entirely unproven; in fact, in one instance it will be shown not to limit the set of solution networks to a single answer. It does, however, work often enough to warrent contemplation.

Section 3.5 -- The Function F(i)

Curtis and Morrow's [1] algorithm for finding the values of the resistors in a rectangular network makes great use of the linear relations that exist between columns of the L matrix; we have already used the simplest of these to find right turns. The L matrix is riddled with columns that are linear combinations of sets of other

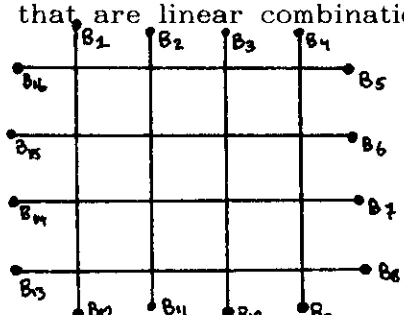


Figure 3.4

columns. For example, in the L matrix for the square network shown in Figure 3.4, the fourth column is a linear combination (multiple) of the fifth except in the fourth and fifth rows. The third column is a linear combination of the fifth and sixth except in the third through the sixth row. The column corresponding to boundary node **B_2** is a linear combination of the fifth through the seventh column, except in the second through the seventh rows. Finally, the first column is a linear combination of the fifth through the eighth columns in all rows of L except the first through the eighth. Let F(i) be the number of the first column in a set of columns of which the ith column is a linear combination. Using this notation, then, in the preceding example F(1) = F(2) = F(3) = F(4) = 5. It should come as no surprise that, still with reference to Figure 3.4, F(5) = F(6) = F(7) = F(8) = 9. These relations seem to revolve around the right turns between boundary nodes B_4 and B_5, and B_8 and B_9.

Similiar behavior may be seen in regular grid networks containing left turns. In the L matrix for the network shown in Figure 3.5,

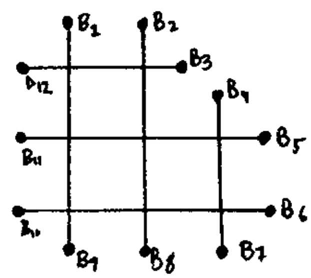


Figure 3.5

F(1) = F(2) = 3, F(3) = 7, F(4) = 5, F(5) = F(6) = 7, F(7) = F(8) = F(9) = 10, and F(10) = F(11) = F(12) = 1. Apparently F(i) is always the number of a boundary node just after a right turn. Interestingly, $B_{F(i)}$ is not the boundary node after the first right turn after B_i, but rather the boundary node after the first **net right turn**; that is, there is exactly one more right turn than left turn between boundary nodes B_i and $B_{F(i)}$.

Section 3.6 -- Finding Boundary Curves from F(i)

Assuming we can find F(i) for all boundary nodes B_i we can determine the boundary curve dR quite easily. To determine whether the boundary curve between B_i and B_{i+1} is a right turn, a straight, or a left turn, simply compare F(i) with F(i+1). If F(i) < F(i+1) then the curve in question is a right turn. If F(i) = F(i+1) then it's straight. Finally, if F(i) > F(i+1) then the boundary curve must turn left between node B_i and B_{i+1}. However, finding F(i) for all i is challenging if not impossible.

Section 3.7 -- Finding F(i)

We seek the number of the first column in a set of columns of which a single linear combination gives column #i. First we find the smallest positive integer j such that column #i is at least one linear combination of column #i+1 through column #i+j in all rows other than row #i through row #i+j. It is quite likely that columns #i+1 through #i+j are not linearly independent. We then look for the lowest **k** such that column #i+k is not a linear combination of columns #i+k+1 through #i+j; column #k is the first column in our set which we cannot remove without reducing the size of the vector space spanned by the set. We assume that F(i) = **k**.

Section 3.8 -- Problems with Section 3.7

The procedure outlined above works perfectly for **squarish networks**, networks such as those in Figures 3.4 and 3.5 with the same number of boundary nodes pointing in each direction. Even the simplest of rectangles, however, causes errors. When applied to the

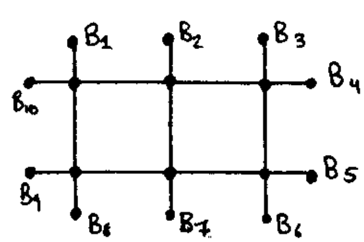


Figure 3.6

network shown in Figure 3.6, for example, the first part of the procedure would discover, correctly, that column #1 is at least one linear combination of columns #2 through #7. The procedure continues by correctly eliminating columns #2 and #3. Obviously F(1) = 4, but the procedure will remove column #4 from consideration as well. Stated somewhat imprecisely, the problem is that since the network is rectangular, node B_7 is closer to node B_4 that it ought to be. Column #4 is a linear combination of columns #6 and #7 and so it is removed. The procedure in section 3.7 eventually decides on F(1) = 6 as its (incorrect) answer.

Section 3.9 -- An Expanded Search for F(i)

It is possible to repair the procedure in Section 3.7 so that it gives a list of possible values for F(i) which will always include the correct answer. This is an improvment; before it would give incorrect results, whereas now it admits to its own shortcomings. The new algorithm is as follows: first, run the old procedure, obtaining some value for F(i). Then remove the last column from the set, column #i+j in the notation of Section 3.7 and column #7 in the example in Figure 3.6. Begin again at column #i+1, eliminating all columns which are linear combinations of those remaining in the set. Now, returning to the example of Figure 3.6, column #7 is not being considered and so column #4 will not be eliminated. The correct value F(1) = 4 will be obtained. Continue by removing the highest-numbered column in the set and then repeating the second part of Section 3.7, obtaining new values for F(i) until the set of columns is exhausted.

Section 3.10 -- Eliminating Sites for Left Turns

Finally we are equipped to remove more of the sites for left turns. We apply the method of Section 3.6 to all those pairs of sequential boundary nodes for which Section 3.9 gives only one value of F(i). For some reason, here left unexplored, left turns are always undetermined; whenever a single value of F(i) is found, the boundary curve between B_i and B_{i+1} is straight. Thus we may remove from consideration as sites for left turns the segments of the boundary curve between any boundary nodes B_i and B_{i+1} if F(i) is determined uniquely by the method of Section 3.9. Still more sites may be eliminated if we modify Sections 3.7 and 3.9 to run counterclockwise instead of clockwise and determine the values of F'(i), which for each i equals the number of the first boundary node counterclockwise from B_i such that there is one more right turn than left turn between boundary nodes $B_{F'(i)}$ and B_i.

Section 3.11 -- Results

No proof has been given that this method for determining the shape of a regular resistor network works. It has been tested, however, on each of the networks in Figure 3.7 and several others. In each case only one solution network, the correct one, is produced.

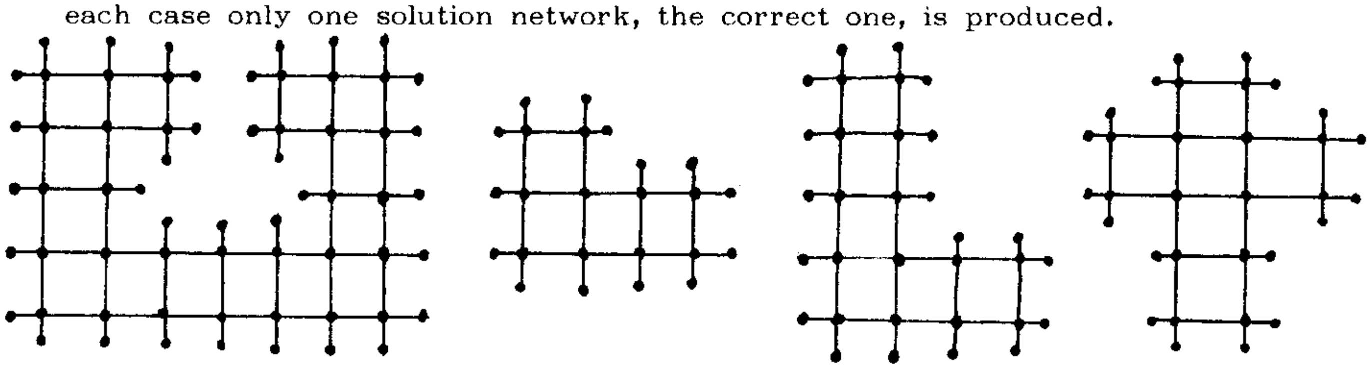
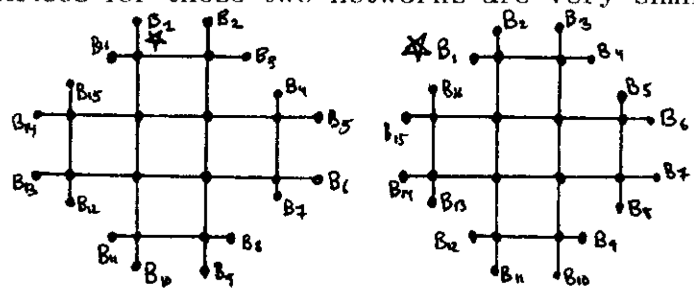


Figure 3.7

However, one network, shown in Figure 3.8.a, gives two solution networks, the correct one and one nearly identical to it, shown in Figure 3.8.b. It is not clear whether this difficulty is resolvable; the L matrices for these two networks are very similiar.



Figures 3.8.a, 3.8.b

Section 3.12 -- Questions

Can the two networks shown in Figure 3.8.a and 3.8.b be distinguished? Why do only straight segments of boundary curve give a unique value for F(i) or F'(i)? Does this method actually solve all networks or does there exist some regular network that generates multiple solution networks that are not as similiar as those in Figure 3.8? These questions remain unanswered.