Locating Faulty Resistors in a Network

Matthew Hudelson
University of Washington

October 17, 1989

Abstract

In this paper I describe some methods for determining the location of a resistor in a network whose value has changed relative to a known network. All other resistors are assumed unchanged. The methods rely on the availability of the Dirichlet to Neumann maps for the networks.

1 Introduction

1.1 Preliminaries

We consider, as in Curtis and Morrow [2], networks of resistors in the plane. For simplicity, only square networks will be considered. Define Ω_n as a square network containing n^2 interior nodes, $4n$ boundary nodes and $2n(n+1)$ segments. Figure 1 shows Ω_5.

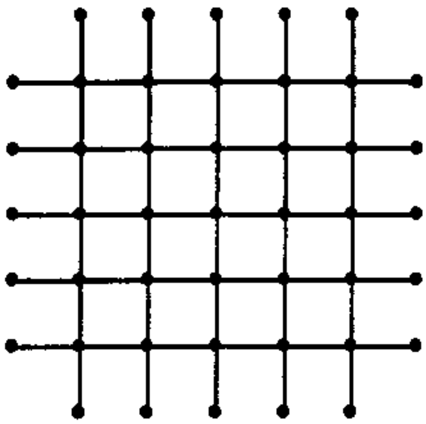


Figure 1

Let $S_n = \{\text{segments connecting nodes of } \Omega_n\}$. Then to describe a network of resistors $\Gamma_{n,\gamma}$, we pair with Ω_n a function $\gamma : S_n \mapsto R^+$ where R^+ is the set of positive real numbers. Considering a segment σ in S_n as a resistor, $\gamma(\sigma)$ is the conductance of σ (the reciprocal of its resistance).

1.2 The Dirichlet to Neumann Map Λ

Given $\Gamma_{n,\gamma}$, we desire the map Λ which will yield the currents entering the network through the boundary nodes, given the potentials at the boundary nodes. In order to construct such a map, we number the boundary nodes starting at the left-most node on the North side of the grid. We then proceed clockwise as in figure 2.

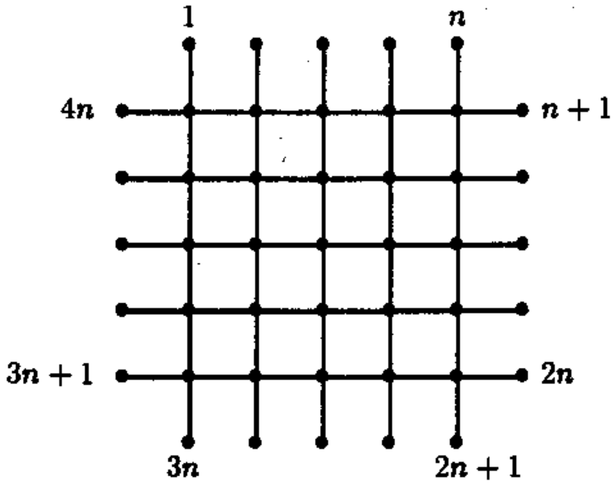


Figure 2

Armed with this numbering scheme, we then place a potential of 1 at boundary node i and 0 at all other boundary nodes. Then $\Lambda_{i,j}$ = the current entering the network at boundary node j, interpreting a negative current as one exiting the network. Then if we place voltages u_i at the corresponding boundary nodes and form a column vector $\mathbf{u}$ from the voltages then the currents are given by

$$\Lambda \mathbf{u} = \mathbf{c}. \tag{1}$$

The entry c_i of $\mathbf{c}$ is the current at the boundary node i.

2 The Alpha Methods

In this section, I describe procedures to exactly determine the location of a deviant resistor based on relationships among certain rows in the two Λ matrices.

2.1 Computing the α's

We consider the following problem: Given a network $\Gamma_{n,\gamma}$, place zero currents along the East edge and zero voltages along the South, East, and North edges, except at boundary node k along the North edge, where we place a voltage of 1. Letting α_i be the voltage of the ith node along the West edge, we have in figure 3:

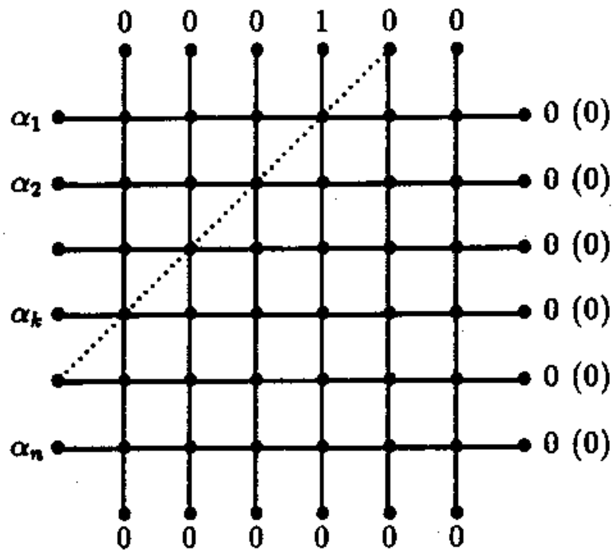


Figure 3

Currents in Parentheses

Starting from the East edge, we can use Kirkhoff's and Ohm's Laws to show that all voltages on and below the dotted diagonal line are zero, which also implies that all currents on the South edge are also zero. From this, we also obtain $\alpha_i = 0$ for $i > k$. It remains for us to determine α_i for $i \leq k$.

We do this by examining Λ for $\Gamma_{n,\gamma}$ so as to rely only on boundary data. Let u be the column vector formed by the voltages in figure 3. We then have $(\Lambda \mathbf{u})_i = 0$ for $k < i < 4n - k + 1$, or in terms of the α's,

$$\Lambda_{i,k} + \sum_{j=1}^{k}(\alpha_j \Lambda_{i,4n+1-j}) = 0 \tag{2}$$

which allows us to form $4n - 2k$ equations in k unknowns. However, Curtis and Morrow point out that these equations, overdetermined as they are, do have a unique solution.

Thus, by finding the relationship which exists between columns k and $4n-k+1$ through $4n$, we determine α_1 through α_k. Furthermore, by rotating and reflecting the boundary conditions as needed and re-indexing equation 1, we can similarly determine α's around any other corner of the network.

2.2 The Intersecting Staircase Method

Consider the network shown in figure 4, the currents being shown in parentheses:

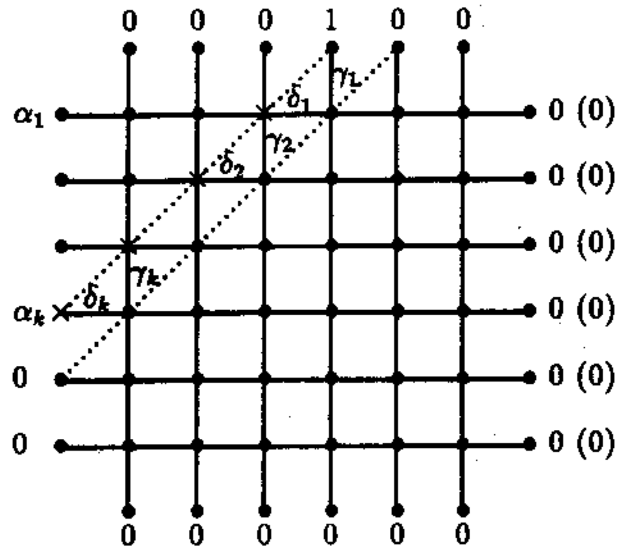


Figure 4

Currents in Parentheses

We define a *staircase* as a set of resistors between two adjacent diagonal lines, such as those in figure 4. Since the voltages along and below the lower diagonal are zero and all currents through resistors below the lower diagonal are zero, we have that the current through γ_1 must be $(1-0)\gamma_1$ by Ohm's Law. Then by Kirkhoff's Law, the same current must then pass through δ_1. Therefore, to satisfy Ohm's Law, the voltage at the northernmost node marked with an $\times$ must be $-\gamma_1/\delta_1$. Continuing to apply Ohm's and Kirkhoff's Laws in this manner along the staircase, we obtain at the mth $\times$ from the North edge a voltage $v_m = (-1)^m \prod_{i=1}^{m} \frac{\gamma_i}{\delta_i}$ Therefore, we obtain

$$\alpha_k = (-1)^k \prod_{i=1}^{k} \frac{\gamma_i}{\delta_i} \tag{3}$$

since α_k is obtained at the kth $\times$ from the North edge. Again, suitable rotations and reflections of figure 4 can be used to obtain α_k for other edges.

Notice, then, that the last α depends only on the resistors in the staircase which leads to it. Therefore, if we have exactly one faulty resistor, it will rest on two staircases, as in figure 5, and it will affect the values of α and β as shown.

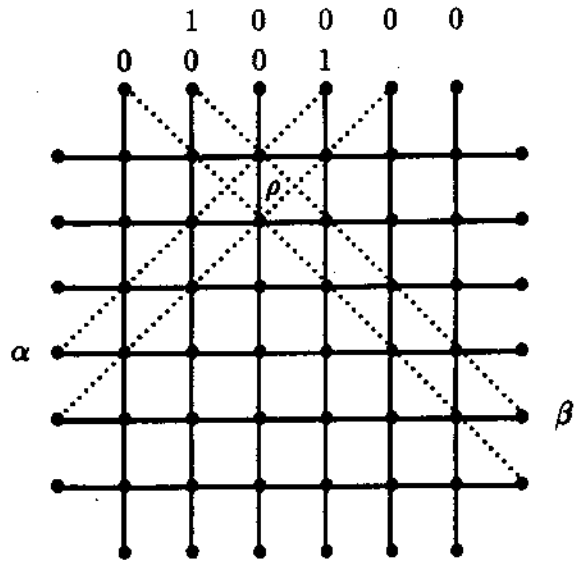


Figure 5

ρ = faulty resistor

Therefore, to locate the faulty resistor, we calculate all the final α's and β's for the faulty network and compare them with the corresponding α's and β's for a standard network. The differing α's and β's will correspond to two intersecting staircases, pinpointing the faulty resistor at the point of intersection.

2.3 The Single Staircase Method

Here I describe a detection method which, upon finding a staircase with the faulty resistor, uses the next to the last α to determine where the resistor is. We must assume that the resistors in the standard network are all identically 1, and that the resistance of the faulty resistor is known. We let ρ be this known resistance. Then we have the situation in figure 6:

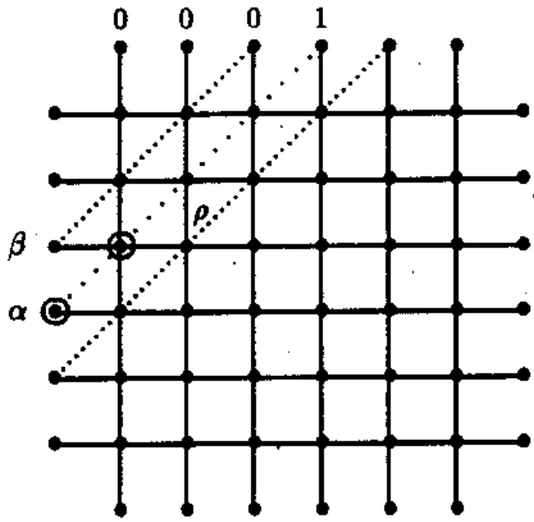


Figure 6

ρ = faulty resistor

There are two possible cases for a single faulty resistor: either $|\alpha| = \rho$ or $|\alpha| = 1/\rho$. In figure 6, we have $|\alpha| = \rho$. If $|\alpha| = 1/\rho$ then we follow a similar derivation which yields a different formula for locating the resistor.

From the earlier discussion, the voltages along the lowest diagonal are all zero. From Kirkhoff's and Ohm's Laws, the magnitudes of the voltages going Southwest down the central diagonal are 1, until reaching the circled ones whose magnitudes are ρ. Now, using the values along the central diagonal, we can compute the values along the upper diagonal. We can do this iteratively using the following diagram:

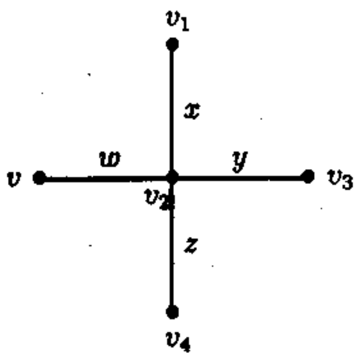


Figure 7

Here the v's are voltages and w, x, y, and z are conductances. Then from Kirkhoff's law, we obtain an expression for v:

$$wv = (w + x + y + z)v_2 - xv_1 - yv_3 - zv_4 \tag{4}$$

We then use Figure 7 and equation 4 repeatedly, placing v_2 on each node of the center diagonal and v_1 on the lowest node on the top diagonal for which a voltage is known. Then equation 4 becomes

$$wv = (w + x + y + z)v_2 - xv_1 \tag{5}$$

since v_3 and v_4 both lie on the lower diagonal and are both zero. We can simplify further, noting that ρ lies between the middle and lower diagonals. This implies that $w = x = 1$ and therefore equation 5 becomes

$$v = (2 + y + z)v_2 - v_1. \tag{6}$$

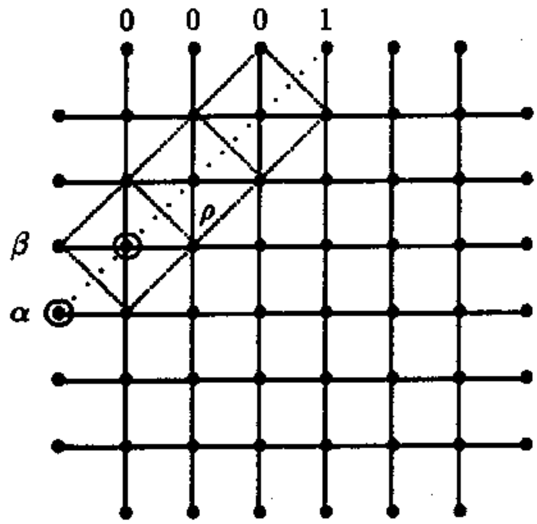


Figure 8

ρ = faulty resistor

The computation of β is then done in three stages, considering crosses in succession southwest along the middle diagonal. First, before reaching a node touching ρ, all conductances in the cross are 1. Also, $v_2 = \pm 1$ and equation 6 becomes

$$v = 4v_2 - v_1. \tag{7}$$

This will generate a sequence of voltages along the upper diagonal of $0, -4, 8, -12, \ldots$ until we reach the cross containing ρ. For this cross, equation 6 becomes

$$v = (3 + \rho)v_2 - v_1 \tag{8}$$

and v becomes $(-1)^k(\rho+4k-1)$ if the first cross containing ρ is the kth cross along the middle diagonal. In subsequent crosses, $v_2 = \pm\rho$ and equation 6 becomes

$$v = 4v_2 - v_1. \tag{9}$$

We then obtain for the $(k+j)$th cross,

$$v = (-1)^{k+j}[(4j+1)\rho + 4k - 1]. \tag{10}$$

Assume that the mth staircase from the Northwest corner produces the faulty α. In figure 8, $m = 4$. Then β is computed from the $m-1$st cross. So, again in figure 8, β would be computed from the third cross. Therefore, from this information and from 10, we obtain two equations in two unknowns, k and j:

$$k + j = m - 1 \tag{11}$$

$$\beta = (-1)^{m-1}[(4j+1)\rho + 4k - 1]. \tag{12}$$

Solving these equations for k gives us the which cross the resistor is in. Doing so yields:

$$k = \frac{|\beta| - 4m\rho + 3\rho + 1}{4 - 4\rho}. \tag{13}$$

$|\beta|$ appears in equation 13 since β and $(-1)^{m-1}$ are the same sign.

Equation 13 was derived for the case when $|\alpha| = \rho$. We can follow similar reasoning to derive a corresponding equation when $|\alpha| = 1/\rho$. Following the derivation yields the following two equations, corresponding to equations 11 and 12:

$$k + j = m - 1 \tag{14}$$

$$\beta = (-1)^{m-1}[(4k-1)\frac{1}{\rho} + 4j + 1]. \tag{15}$$

Solving these for k yields:

$$k = \frac{|\beta| + \frac{1}{\rho} - 4m + 3}{\frac{4}{\rho} - 4}. \tag{16}$$

We may therefore use equations 13 or 16, whichever is appropriate, to determine which cross to find the faulty resistor.

3 The Delta Method

Here I describe a tentative method for locating a small number of faulty resistors in a network, given information about single deviant resistors.

3.1 Constructing the Set $\mathcal{D}$ of Elementary Delta Vectors

We shall only consider networks whose resistors each assume one of two values, 1 and ρ. We begin by considering a basic network $\Gamma_{n,\gamma}$ where $\gamma(\sigma) = 1$ for all resistors σ in the network. Let Λ_0 be the Dirichlet to Neumann Map for $\Gamma_{n,\gamma}$. Now, if we index the $2n(n+1)$ resistors in the network, we can associate with each resistor σ_k a γ_k such that

$$\gamma_k(\sigma_i) = \begin{cases} \rho & \text{if } i = k \\ 1 & \text{if } i \neq k \end{cases}$$

Thus, we can now obtain new Dirichlet to Neumann Maps Λ_k, one corresponding to each network Γ_{n,γ_k} generated by the γ_k's.

We now construct the set $\mathcal{D}$. First, we define, $\Delta_k = \Lambda_0 - \Lambda_k$. We then form a *delta vector* δ_k by setting $(\delta_k)_i = |(\Delta_k)_{i,i}|$, that is, we take the diagonal entries of Δ_k and place their absolute values into the vector δ_k. The signs of the diagonal entries of Δ_k are the same; they are all positive if $\rho < 1$ and negative if $\rho > 1$. Since they are all the same sign, the discussion to follow becomes more convenient if we only consider their magnitude.

We then set $\mathcal{D} = \{\delta_k : 1 \leq k \leq 2n(n+1)\}$. We then call $\mathcal{D}$ the set of *elementary* delta vectors, as these vectors are generated from changing only one resistor in the basic network.

3.2 Locating Faulty Resistors Using the Approximate Additivity of Delta Vectors

In this section I shall proceed informally. I called $\mathcal{D}$ the set of elementary delta vectors because this set can be used to approximate more complicated situations. Assume we change the conductances of a small number $(2 \leq k \leq 6)$ of resistors from 1 to ρ, and let $\delta_a, \delta_b, \ldots, \delta_m$ be the corresponding elementary delta vectors for these resistors individually. Also, let δ be the delta vector produced by changing all the conductances simultaneously. Then the following seem to hold:

$$\delta \approx \delta_a + \delta_b + \cdots + \delta_m \tag{17}$$

$$\delta > \delta_i \text{ for all } i \in \{a, b, \ldots, m\} \tag{18}$$

By inequality 18, I mean that all components of δ are greater than the corresponding components of δ_i.

Assume that we are given a δ corresponding to a change of a small number of resistors. We can now use (17) and (18) to generate a subset of $\mathcal{D}$ corresponding to the resistors which were changed. We use both in order to narrow down our search for this subset more quickly. The first step is to eliminate all $\delta_j \in \mathcal{D}$ which have any components larger than those of δ. We can do this thanks to (18), since all components of a delta vector are positive. Now, if any δ_j equals δ we're done since this means only one resistor was changed, namely resistor σ_j. If not, we form the set $\mathcal{D}_1$ from the δ_j's which survived. From the set $\mathcal{D}_1$, we construct the sets $\mathcal{D}_m = \{(\delta_{i_1}, \ldots, \delta_{i_m}) : i_1 < \ldots < i_m\}$.

For each element $(\delta_{i_1}, \ldots, \delta_{i_m}) \in \mathcal{D}_m$, we compute $\zeta = \sum_{j=1}^{m} \delta_{i_j}$. Recalling that the δ's and ζ are vectors, we form a new vector $\mathbf{d}$ by dividing, component by component, our target δ by ζ. Thus, $\mathbf{d}_j = \delta_j / \zeta_j$. This is done so each component contributes equally in the determination of how well ζ approximates δ.

In order to proceed, we set $\mathbf{1} = (1, 1, \ldots, 1)$, where $\mathbf{1}$ has as many components as $\mathbf{d}$. Then if we compute the angle between $\mathbf{d}$ and $\mathbf{1}$, we have a measure of how well ζ approximates δ. Therefore, we want

$$\cos(\theta) = \frac{\mathbf{d} \cdot \mathbf{1}}{\|\mathbf{1}\| \, \|\mathbf{d}\|} = \frac{\sum_{j=1}^{m} d_j}{\sqrt{m} \, \|\mathbf{d}\|} \tag{19}$$

to be as close to 1 as possible. Whichever element of $\mathcal{D}_m$ sets $\cos(\theta)$ closest to 1 should correspond to a set of faulty resistors approximating the desired set. The best approximation among the $\mathcal{D}_m$'s often yields a set of faulty resistors very similar to the target set, especially for small sets. Unfortunately, the method seems to decay fairly rapidly, especially for sets containing more than eight faulty resistors.

4 Experimental Results

Here, I describe some computer simulations of these various detection methods. Most of the needed linear algebra routines were programmed using the `LINPACKD` routines `DGBFA`, `DGBSL`, `DGEFA`, `DGESL`, `DQRDC`, and `DQRSL`. These routines are described in detail in Golub and Van Loan [4].

4.1 Alpha Method Experiments

Most of my efforts revolved around trying to improve the calculation of α's in order to find faulty resistors in large networks (up to Ω_{24}). In order to accomplish this, I resorted to differing the arrangement of the α's as shown in figure 9:

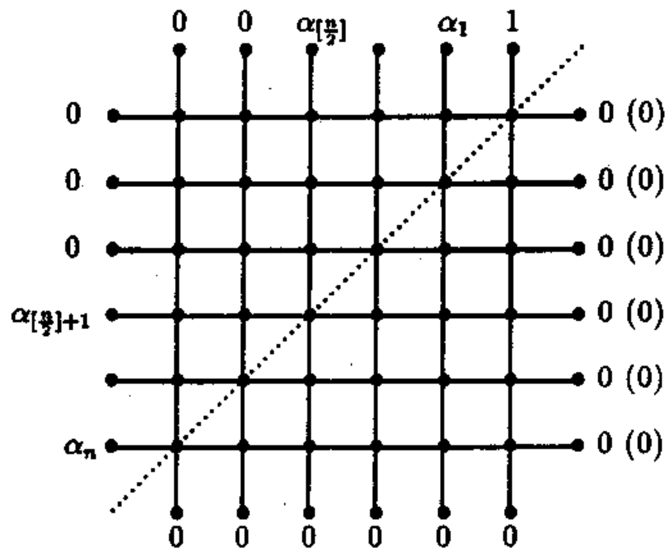


Figure 9

$[x]$ =greatest integer $\leq x$

In other words, I placed half of the α's next to the "1" and the other half leading to the opposite corner. This kept the relative magnitude of the largest α's as small as possible.

The following chart summarizes experiments conducted using standard networks containing resistors all of whose conductances are 1. In each case, the resistor closest to and to the north of center had its conductance increased to 100. Also, the computer would declare that a change had occurred if a staircase which should have produced $\alpha_n = 1$ was in error by more than 2%. In no event, for any $\Omega_n, n \leq 24$, did the program fail to find the exact location of the resistor with conductance of 100.

Network Size	Maximum Errors in α_n			Fault found Between
	1.0×10^{2}	1	1.0×10^{-2}	
10	6.7×10^{-9}	5.6×10^{-12}	4.4×10^{-13}	(5,5) and (6,5)
13	1.4×10^{-6}	2.2×10^{-10}	3.5×10^{-11}	(7,6) and (7,7)
16	4.3×10^{-5}	1.8×10^{-8}	2.9×10^{-9}	(8,8) and (9,8)
18	4.0×10^{-4}	3.7×10^{-7}	3.5×10^{-8}	(9,9) and (10,9)
20	8.8×10^{-3}	2.6×10^{-6}	8.7×10^{-7}	(10,10) and (11,10)
21	2.1×10^{-2}	2.0×10^{-5}	3.0×10^{-6}	(11,10) and (11,11)
22	9.2×10^{-2}	3.2×10^{-5}	4.1×10^{-6}	(11,11) and (12,11)
23	4.8×10^{-1}	1.3×10^{-4}	8.6×10^{-6}	(12,11) and (12,12)
24	1.82	6.0×10^{-4}	1.1×10^{-4}	(12,12) and (13,12)

The last column gives the coordinates of the two nodes surrounding the faulty resistor. These coordinates are oriented so that (1,1) represents the upper left corner interior node and (n,n) represents the lower right corner interior node.

In order to test the single staircase method, I had to arrange the α's in their original order, all along one side. This, unfortunately, lessened the accuracy of the measurements. Therefore, the simulation yielded meaningless results for Ω_n, $n > 19$. The following table encapsulates the results:

Network Size	m	Cross Location Determined by Equation		Fault found Between
		13	16	
10	10	4.5000000005301	4.9999999994699	(5,5) and (6,5)
11	11	4.9999999999512	5.5000000000488	(6,5) and (6,6)
12	12	5.5000000249843	5.9999999750157	(6,6) and (7,6)
13	13	5.9999999997753	6.5000000002247	(7,6) and (7,7)
14	14	6.4999985068789	7.0000014931211	(7,7) and (8,7)
15	15	7.0000001109226	7.4999998890774	(8,7) and (8,8)
16	16	7.5001878666356	7.9998121333644	(8,8) and (9,8)
17	17	8.0000295362242	8.4999704637758	(9,8) and (9,9)
18	18	8.5311332885963	8.9688667114037	(9,9) and (10,9)
19	19	9.0031100105497	9.4968899894503	(10,9) and (10,10)

In each of these runs, I took the same faulty locations as in the previous table. Also, I took the worst of the two cases, considering staircases running from the north side to either the east or the west side. The second column, m, is the number of the staircase on which the faulty resistor was found. Staircase 1 would be the shortest staircase containing only two corner resistors. It is possible that better results could be obtained by considering staircases leading from the east or the west to the north. Note that the incorrect equation (from the third or fourth columns) yields an answer close to $k+\frac{1}{2}$ for some integer k. Also, note that adding the results from the third and fourth columns leads to a sum which is apparently *exactly* $k + \frac{1}{2}$ for some integer k. I do not know why these results occur.

4.2 Delta Method Experiments

I offer some pictoral output showing test runs of the delta method. In each case, when a resistor having conductance 100 was detected, the printout shows "#####". One that was missed is depicted by "+++++", and one that was mistakenly identified (a "phantom") is depicted by "?????".

5 References

1. Howard Anton and Chris Rorres. *Elementary Linear Algebra with Applications.* John Wiley & Sons, New York. ©1987

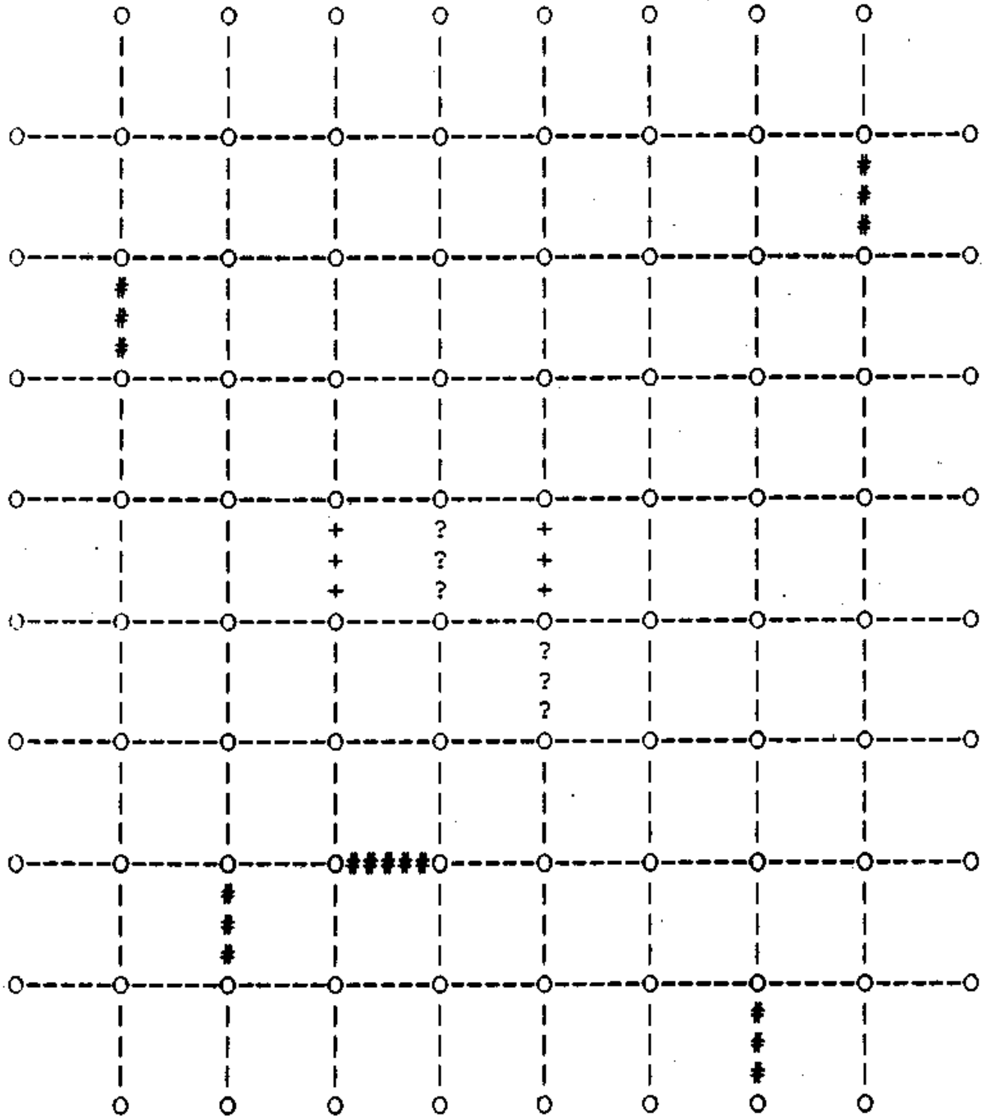
2. Edward B. Curtis and James A. Morrow. *Determining the Resistors in a Network*

3. Thad Edens. *Calculating the Resistors in a Network*

4. Gene H. Golub and Charles F. Van Loan. *Matrix Computations, Second Edition.* Johns Hopkins University Press, Baltimore and London. ©1989

```
      O     O     O     O     O     O     O
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O
      |     #     |     |     |     |     |
      |     #     |     |     |     |     |
      |     #     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O#####O-----O
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      O     O     O     O     O     O     O

  ##### = Detected
  ????? = Phantom
  +++++ = Missed
```



```
##### = Detected
????? = Phantom
+++++ = Missed
```

```
      o     o     o     o     o     o     o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     #     |     |     |     |     |
      |     #     |     |     |     |     |
      |     #     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o#####o-----o
      |     #     |     |     |     |     |
      |     #     |     |     |     |     |
      |     #     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      o     o     o     o     o     o     o
  ##### = Detected
  ????? = Phantom
  +++++ = Missed
```

```
       o      o      o      o      o      o      o
       |      |      |      |      |      |      |
       |      |      |      |      |      |      |
       |      |      |      |      |      |      |
o------o------o------o------o------o------o------o------o
       |      |      |      |      |      |      |
       |      |      |      |      |      |      |
       |      |      |      |      |      |      |
o######o------o------o------o------o------o------o------o
       |      #      |      |      |      |      |
       |      #      |      |      |      |      |
       |      #      |      |      |      |      |
o------o------o------o------o------o------o------o------o
       |      |      |      |      |      |      |
       |      |      |      |      |      |      |
       |      |      |      |      |      |      |
o------o------o------o------o------o------o------o------o
       |      |      |      |      |      |      |
       |      |      |      |      |      |      |
       |      |      |      |      |      |      |
o------o------o------o------o------o------o######o------o
       |      #      |      |      |      |      |
       |      #      |      |      |      |      |
       |      #      |      |      |      |      |
o------o------o------o------o------o------o------o------o
       |      |      |      |      |      |      |
       |      |      |      |      |      |      |
       |      |      |      |      |      |      |
o------o------o------o------o------o------o------o------o
       |      |      |      |      |      |      |
       |      |      |      |      |      |      |
       |      |      |      |      |      |      |
       o      o      o      o      o      o      o
  ##### = Detected
  ????? = Phantom
  +++++ = Missed
```

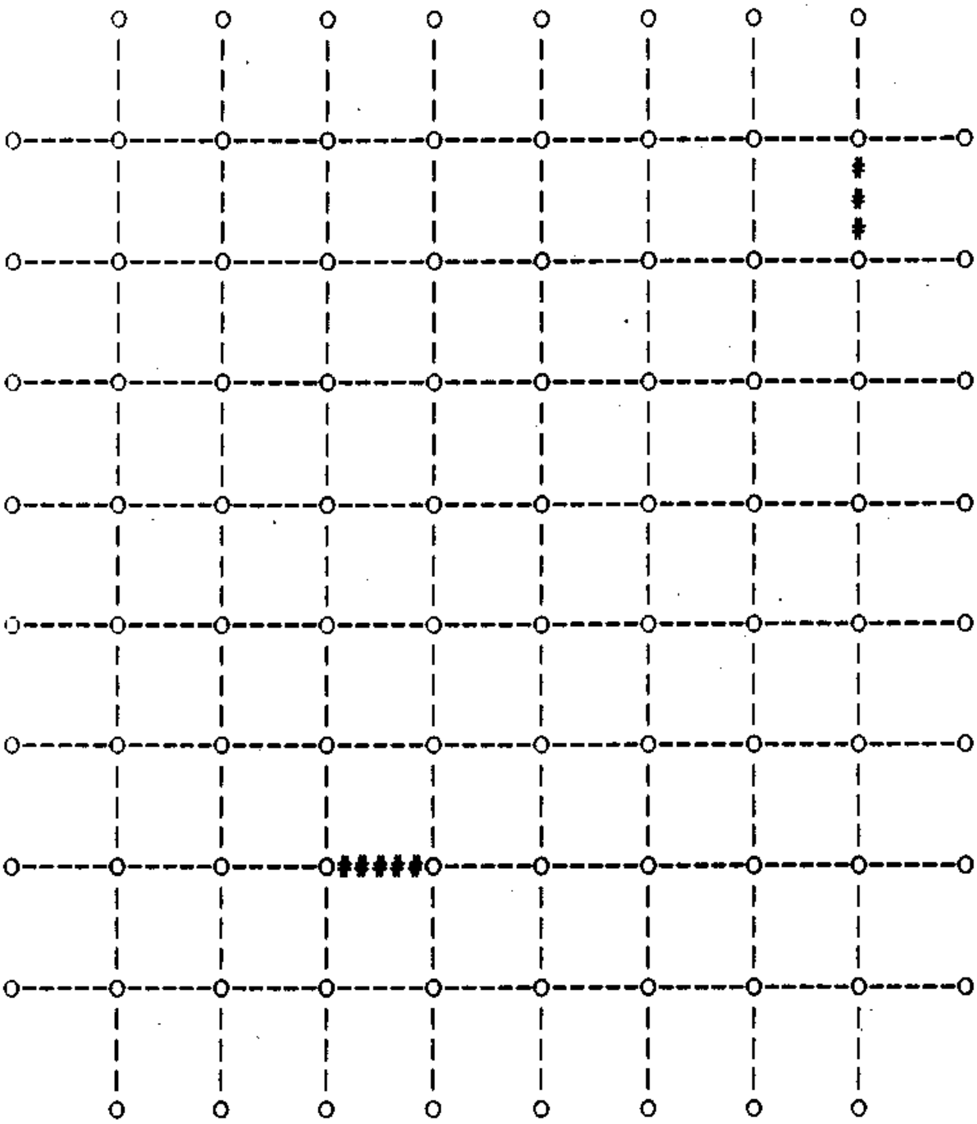
```
      o     o     o     o     o     o     o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
o-----o-----o#####o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
o#####o-----o-----o-----o-----o-----o-----o-----o
      |     +     |     |     |     |     |
      |     +     |     |     |     |     |
      |     +     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     ?     |     |     |     |     |
      |     ?     |     |     |     |     |
      |     ?     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o#####o-----o
      |     #     |     |     |     |     |
      |     #     |     |     |     |     |
      |     #     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      o     o     o     o     o     o     o
```

```
##### = Detected
????? = Phantom
+++++ = Missed
```

```
      o     o     o     o     o     o     o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
o-----o-----o#####o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
o+++++o?????o-----o-----o-----o-----o-----o-----o
      |     #     +     |     |     |     |
      |     #     +     |     |     |     |
      |     #     +     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     ?     |     |     |     |     |
      |     ?     |     |     |     |     |
      |     ?     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o#####o-----o
      |     #     |     |     |     |     |
      |     #     |     |     |     |     |
      |     #     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      |     |     |     |     |     |     |
      o     o     o     o     o     o     o
```

= **Detected**
????? = **Phantom**
+++++ = **Missed**

```
     O     O     O     O     O     O     O
     |     |     |     |     |     |     |
     |     |     |     |     |     |     |
     |     |     |     |     |     |     |
O-----O-----O+++++O-----O-----O-----O-----O-----O
     |     |     |     |     |     |     |
     |     |     |     |     |     |     |
     |     |     |     |     |     |     |
O+++++O?????O-----O-----O-----O-----O-----O-----O
     |     #     +     |     |     |     |
     |     #     +     |     |     |     |
     |     #     +     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O
     |     |     |     |     |     |     |
     |     |     |     |     |     |     |
     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O?????O-----O-----O
     |     |     |     |     ?     |     +
     |     |     |     |     ?     |     +
     |     |     |     |     ?     |     +
O-----O-----O-----O-----O?????O-----O+++++O-----O
     |     +     |     |     |     |     |
     |     +     |     |     |     |     |
     |     +     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O
     |     |     |     |     |     |     |
     |     |     |     |     |     |     |
     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O
     |     |     |     |     |     |     |
     |     |     |     |     |     |     |
     |     |     |     |     |     |     |
     O     O     O     O     O     O     O
  ##### = Detected
  ????? = Phantom
  +++++ = Missed
```



= Detected
????? = Phantom
+++++ = Missed

```
      O     O     O     O     O     O     O     O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     #
      |     |     |     |     |     |     |     #
      |     |     |     |     |     |     |     #
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     #     |     |     |
      |     |     |     |     #     |     |     |
      |     |     |     |     #     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O#####O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      O     O     O     O     O     O     O     O
  ##### = Detected
  ????? = Phantom
  +++++ = Missed
```

```
      O     O     O     O     O     O     O     O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     #
      |     |     |     |     |     |     |     #
      |     |     |     |     |     |     |     #
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     +     |     |     |
      |     |     |     |     +     |     |     |
      |     |     |     |     +     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     ?     |     |     |
      |     |     |     |     ?     |     |     |
      |     |     |     |     ?     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O?????O+++++O-----O-----O-----O-----O-----O
      |     #     |     |     |     |     |     |
      |     #     |     |     |     |     |     |
      |     #     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      O     O     O     O     O     O     O     O
```

= Detected
????? = Phantom
+++++ = Missed

```
      o     o     o     o     o     o     o     o
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |     #
      |     |     |     |     |     |     |     #
      |     |     |     |     |     |     |     #
o-----o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     #     |     |     |
      |     |     |     |     #     |     |     |
      |     |     |     |     #     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
o-----o-----o-----o#####o-----o-----o-----o-----o-----o
      |     #     |     |     |     |     |     |
      |     #     |     |     |     |     |     |
      |     #     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     #     |
      |     |     |     |     |     |     #     |
      |     |     |     |     |     |     #     |
      o     o     o     o     o     o     o     o
 ##### = Detected
 ????? = Phantom
 +++++ = Missed
```

```
      O     O     O     O     O     O     O     O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     #
      |     |     |     |     |     |     |     #
      |     |     |     |     |     |     |     #
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      #     |     |     |     |     |     |     |
      #     |     |     |     |     |     |     |
      #     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     #     |     |     |
      |     |     |     |     #     |     |     |
      |     |     |     |     #     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O#####O-----O-----O-----O-----O-----O
      |     #     |     |     |     |     |     |
      |     #     |     |     |     |     |     |
      |     #     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     #     |
      |     |     |     |     |     |     #     |
      |     |     |     |     |     |     #     |
      O     O     O     O     O     O     O     O
    ##### = Detected
    ????? = Phantom
    +++++ = Missed
```

```
      O     O     O     O     O     O     O     O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     #
      |     |     |     |     |     |     |     #
      |     |     |     |     |     |     |     #
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      #     |     |     |     |     |     |     |
      #     |     |     |     |     |     |     |
      #     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     +     ?     +     |     |     |
      |     |     +     ?     +     |     |     |
      |     |     +     ?     +     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     ?     |     |     |
      |     |     |     |     ?     |     |     |
      |     |     |     |     ?     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O#####O-----O-----O-----O-----O-----O
      |     #     |     |     |     |     |     |
      |     #     |     |     |     |     |     |
      |     #     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     #     |
      |     |     |     |     |     |     #     |
      |     |     |     |     |     |     #     |
      O     O     O     O     O     O     O     O
```

**= Detected**
????? **= Phantom**
+++++ **= Missed**

```
      O     O     O     O     O     O     O     O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     #
      |     |     |     |     |     |     |     #
      |     |     |     |     |     |     |     #
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      #     |     |     |     |     |     |     |
      #     |     |     |     |     |     |     |
      #     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     +     ?     +     |     |     |
      |     |     +     ?     +     |     |     |
      |     |     +     ?     +     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     ?     |     |     |
      |     |     |     |     ?     |     |     |
      |     |     |     |     ?     |     |     |
O-----O-----O#####O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
O-----O-----O-----O#####O-----O-----O-----O-----O-----O
      |     #     |     |     |     |     |     |
      |     #     |     |     |     |     |     |
      |     #     |     |     |     |     |     |
O-----O-----O-----O-----O-----O-----O-----O-----O-----O
      |     |     |     |     |     |     #     |
      |     |     |     |     |     |     #     |
      |     |     |     |     |     |     #     |
      O     O     O     O     O     O     O     O
```

= **Detected**
????? = **Phantom**
+++++ = **Missed**

```
      o     o     o     o     o     o     o     o
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |     +
      |     |     |     |     |     |     |     +
      |     |     |     |     |     |     |     +
o-----o-----o-----o-----o-----o-----o-----o-----o-----o
      +     |     |     |     |     |     |     |
      +     |     |     |     |     |     |     |
      +     |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
      |     |     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     +     |     +     |     |     |
      |     |     +     |     +     |     |     |
      |     |     +     |     +     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     ?     |     |     |     +     |
      |     |     ?     |     |     |     +     |
      |     |     ?     |     |     |     +     |
o-----o-----o+++++o?????o-----o-----o-----o-----o-----o
      |     |     |     |     |     ?     |     |
      |     |     |     |     |     ?     |     |
      |     |     |     |     |     ?     |     |
o-----o-----o-----o+++++o-----o-----o-----o-----o-----o
      |     +     |     |     |     |     |     |
      |     +     |     |     |     |     |     |
      |     +     |     |     |     |     |     |
o-----o-----o-----o-----o-----o-----o-----o-----o-----o
      |     |     |     |     |     |     +     |
      |     |     |     |     |     |     +     |
      |     |     |     |     |     |     +     |
      o     o     o     o     o     o     o     o
   ##### = Detected
   ????? = Phantom
   +++++ = Missed
```