12-5-2008

Determining the Shape of a Resistor Grid

Esther Chiew

Vincent Selhorst-Jones

Recommended Citation

Chiew, Esther and Selhorst-Jones, Vincent, "Determining the Shape of a Resistor Grid" (2008). *Mathematical Sciences Technical Reports (MSTR)*. 33.
https://scholar.rose-hulman.edu/math_mstr/33

Determining the Shape of a Resistor Grid

E. Chiew and V. Selhorst-Jones

Adviser: Kurt M. Bryan

Mathematical Sciences Technical Report Series
MSTR 08-04

December 5, 2008

Department of Mathematics
Rose-Hulman Institute of Technology
http://www.rose-hulman.edu/math

Fax (812)-877-8333 Phone (812)-877-8193

Determining the Shape of a Resistor Grid through Boundary Measurements*

Esther Chiew
Vincent Selhorst-Jones

Rose-Hulman, Summer 2008 REU

December 5, 2008

1 Introduction

Impedance imaging has received a lot of attention in the past two decades, as a means for non-destructively imaging the interior of a conductive object. One injects a known electrical current pattern into an object at the exterior boundary, then measures the induced potential (voltage) on some portion of the boundary. The goal is to recover information about the interior conductivity of the object, which (we hope) influences the voltages we measure. Of course one can also use multiple input currents and measured voltages. A variation on this problem is that of "boundary identification," in which some portion of the object's boundary is unknown (and inaccessible), and must be determined from this type of current in-voltage out data. This might be due to the boundary having been corroded or otherwise damaged. See [5] for a survey of some of the extensive work that has been done on the impedance imaging problem and applications.

The equation that governs the flow of current through a conductive object is an elliptic partial differential equation, and this constitutes the *forward problem*; indeed, if the object has constant conductivity then the equation which governs the electric potential in the object is merely Laplace's equation. The recovery of the internal state of the object, or some inaccessible portion of the boundary, is the *inverse problem.* In this paper our goal is to analyze a couple specific instances of this type of inverse problem, in the two-dimensional setting. First, we consider the problem of locating an internal defect consisting of missing material, that is, a *void* in the object, under the assumption that the material of which the object is comprised has constant (known) conductivity. We also

*This work was partially supported by an NSF-REU grant, DMS-0647121

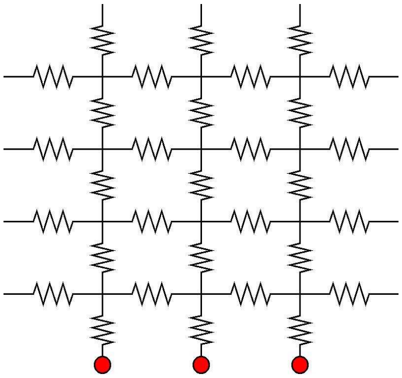

Figure 1: A grid of 1 ohm resistors.

consider the problem of determining the profile of some inaccessible portion of the boundary of the object, again under the assumption of constant conductivity.

However, rather than consider the continuous versions of these inverse problem, we will consider a discrete approximation. Specifically, we replace the domain that defines the object of interest by a (mostly) rectangular planar network of identical resistors, as illustrated in Figure 1. It can be shown that as the spacing of the grid is refined (and the resistors scaled appropriately) the electrical properties of the grid, in particular, the voltages induced by given input current patterns, converge to the relevant solutions of the continuous problem; we'll say more on this below. For our grids we'll use 1 ohm resistors, for convenience. Inverse problems related to resistor networks have been studied extensively by Curtis and Morrow (see, e.g., [2]) and REU students at the University of Washington.

Our discrete version of a void will be a network with one or more missing interior nodes and adjacent resistors, as illustrated in Figure 2. We refer to this as the *missing node* case. An example of an "unknown boundary" inverse problem is illustrated in Figure 5, in which the red resistors are considered to be missing, but grounded, that is, both ends are held at a zero volt potential. A variety of boundary condition models are possible in the inverse problem, and we discuss these below. We refer to this inverse problem as the *varying backside* case. It is intended as a model for the situation in which material may be missing or corroded away from some inaccessible portion of the object's boundary. This problem has been studied in both the electrical and thermal settings; see, for example, [1] or [3].

Let us precisely defined what we mean by "rectangular network" in this paper. Our networks consist of a set of nodes at lattice points (j, k) in the plane, with $0 \leq j \leq n-1$ and $0 \leq k \leq m-1$ for some positive integers m and n (thus the network has m rows and n columns); we omit the "corner nodes"

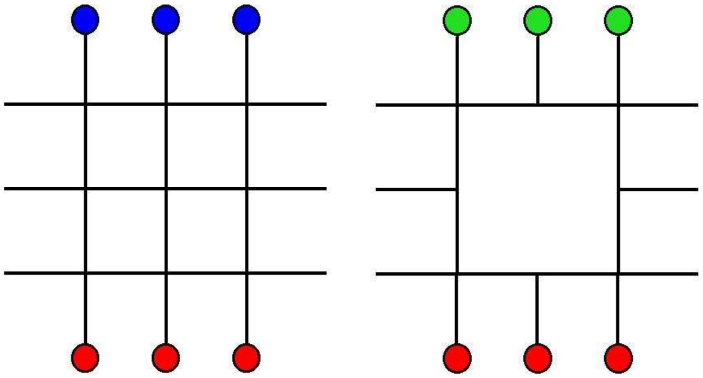

Figure 2: Both grids have same frontside data and insulated sides.

$(0,0),(0,m-1),(n-1,0),(n-1,m-1)$. The nodes (j,k) in which $j=0$ or $k=0$ will be called "boundary nodes". For case of a full rectangular network each "interior node" (j,k) (nodes with $1 \leq j \leq n-2, 1 \leq k \leq m-2$), will be connected to nodes $(j-1,k),(j+1,k),(j,k-1),(j,k+1)$ by a 1 ohm resistor. In the missing node or varying backside case some nodes and the attached resistors may be omitted (to simulate missing material, i.e., corrosion). Alternatively, certain nodes may be forced to have zero electric potential; a physical motivation is given below. In any case, in our networks all interior nodes will be connected to at least TWO other nodes. All of our networks will be connected (as graphs), that is, there will be a conductive path from any given node to any other node.

In [2] the authors present results for a closely related inverse problem. Specifically, consider a fully rectangular grid of resistors in which all resistors are present (no missing interior nodes or resistors), but suppose the resistance themselves are unknown (not necessarily 1 Ohm). The authors show that knowledge of all possible input current/boundary voltages pairs (the "current to voltage" or "Dirichlet to Neumann" map) uniquely determines all interior resistances, at least provided the resistances are strictly positive and finite.

In our analysis of each inverse problem, a computation of fundamental importance is that of *continuation of Cauchy data*: given the voltage at each exterior node along, say, the bottom of a rectangular network (e.g., that of Figure 1) and the input current of the corresponding resistor we can use Ohm's Law to obtain the voltage on the nodes one row above. If in addition we have boundary data for the sides of the grid, we can also use Kirchoff's current law to continue the voltage data up the grid, row by row, finding out voltages as we go. Depending on the outcome of this computation we will make decisions about missing nodes or the shape of the backside (in this case, top) of the network. Details of the computation are given below.

In the Section 2 we consider the forward problem, that is, the equations that govern the flow of current through a resistor grid, relevant boundary conditions, and the continuation of Cauchy data. In Section 3 we provide an algorithm for the varying backside inverse problem, and in Section 4 we have an algorithm for the missing node case; both sections have computational examples. In Section 5 we actually consider the extension of these techniques to the case in which periodic heating is used for imaging, at a fixed frequency.

2 The Forward Problem and Continuation of Cauchy Data

In this section, we discuss the fundamental equations that govern the forward problem, and provide a careful examination of the process of continuing Cauchy data. In our rectangular networks all nodes will be junctions of at most 4 resistors, though, for example, exterior nodes will correspond to only one resistor. For now we will also focus on fully rectangular networks, those with no missing interior or boundary nodes.

2.1 Fundamental Equations and Properties

Before we proceed to an analysis of the inverse problems of interest we need a a careful formulation of the forward problems. It is also helpful to establish some general properties for the behavior of electrical networks.

The resistor network is governed by two fundamental equations,

$$\text{Ohm's Law: } V = IR \tag{1}$$

$$\text{Kirchoff's Current Law: } \sum_{\text{resistors}} i = 0 \tag{2}$$

where in equation (1) V denotes the potential drop across a resistor of resistance R and I is the current through the resistor (in the direction of the voltage drop, i.e., current flows from higher potential to lower potential). In equation (2) i denotes the input electrical current into a fixed node through a given resistor, and the sum is taken over all input connections to the node. These two equations are sufficient to determine the voltage at any node in the network, if the voltage at each boundary node is specified, as shown below. Looking at any interior node, as in Figure 3, and recalling that in our network each resistor has 1Ω resistance, we have the following equations

$$\begin{aligned} V_1 - V &= i_1 \\ V_2 - V &= i_2 \\ V_3 - V &= i_3 \\ V_4 - V &= i_4 \\ \sum_{k=1}^{4} i_k &= 0 \end{aligned}$$

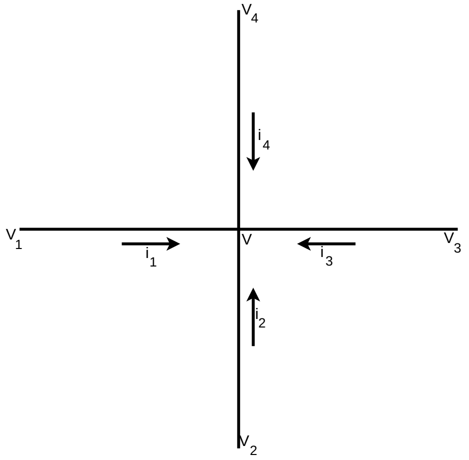

Figure 3: Inner Node for Electricity

with the V_k, i_k as shown in Figure 3. Summing the first four equations above, making use of the last, and solving for V gives us the relation $V = \frac{V_1+V_2+V_3+V_4}{4}$. In fact, it's easy to see more generally that if a node with voltage V is connected to r other nodes with voltages $V_1, \ldots, V_r$ then

$$V = \frac{V_1 + \cdots + V_r}{r} \tag{3}$$

Equation (3) is also known as the *mean-value property*, for it expresses the fact that the voltage at any node is simply the average of the nodes to which it is connected. Equation (3) is used by the algorithm that we use to propagate voltages through a resistor network.

An immediate consequence of equation (3) is

Lemma 1. *If an interior node with voltage V is connected to r other nodes with voltages $V_1, \ldots, V_4$ and $V \geq V_k$ for $1 \leq k \leq r$ then $V = V_1 = V_2 = \cdots = V_r$.*

A similar conclusion holds if we replace $V \geq V_k$ with $V \leq V_k$.

From Lemma 1 follows the Maximum Principle:

Theorem 2. *If the maximum value for the voltage on a connected network is attained at some interior node then the voltage on the entire network is constant.*

Proof. First, suppose that an interior node with voltage V is connected to $r > 0$ other nodes with voltages $V_1, \ldots, V_r$, with $V_k \leq V$ for $1 \leq k \leq r$. From Lemma 1 if any $V_k < V$ then we obtain a contradiction, so we must conclude that $V_k = V$ for all k. Thus if a node has voltage V greater than or equal to all other nodes to which it is connected, all of these nodes are in fact at voltage V.

Now consider two distinct nodes in the network, say with voltages V and $\tilde{V}$, where the node at voltage V is an interior node at which the maximum voltage for the network is attained. Since the network is connected we can find a path between these two nodes, say with intermediate voltages $V_1, V_2, \ldots, V_k$ at the nodes along the path (all of which must be interior). From the paragraph above we must have $V = V_1 = V_2 \cdots = V_k = \tilde{V}$. Thus all nodes in the network must have voltage V.

□

An easy variation of Theorem 2 is that if the minimum value for the voltage on a connected network is attained at some interior node then the voltage on the entire network is constant. In any case, the maximum and minimum values for a network with non-constant voltages must be attained on the boundary of the network.

2.2 Forward Problem

In the forward problem, we begin with a known resistor network, such as one that has a known back surface, or one in which we know which interior nodes are missing. We also have one boundary condition for each exterior boundary node, either the voltage at the node or the input current of the adjacent resistor. Under these conditions one can show (see [2]) that the voltage at each node is uniquely determined, except in the case in which only input boundary currents are specified; in this case the voltages are determined only up to an arbitrary additive constant.

To illustrate, let us consider a full rectangular network such as that of Figure 1 in the case that all boundary node voltages are specified. The claim is that equation (3), applied at each interior node, uniquely determines all interior nodal voltages. If we also specify the boundary voltages. To show that the voltages are uniquely determined, suppose there are two sets of nodal voltages that meet the specified boundary data and equation (3), say voltages V_{jk} at node (j, k) and another set of voltages W_{jk}. Then it's easy to see that the voltages $\tilde{V}_{jk} = V_{jk} - W_{jk}$ also satisfy the equations (1) and (2). Moreover, $\tilde{V}_{jk} = 0$ at any boundary node. Suppose some interior node attains a positive voltage. Then the maximum voltage on the whole network must be attained at some interior node (p, q), so $\tilde{V}_{pq} > 0$. From the Maximum Principle, Theorem 2 we obtain $\tilde{V}_{jk} = \tilde{V}_{pq} > 0$ for all nodes, contradicting the zero boundary voltage. So we must have $\tilde{V}_{jk} \leq 0$ at all interior nodes. A similar argument with the minimum principle shows $\tilde{V}_{jk} \geq 0$ at all nodes, and so $\tilde{V}_{jk} = 0$ at all nodes.

Of course equation (3) applied at each interior node yields a set of $(m - 2)(n-2)$ linear equations $\mathbf{AV} = \mathbf{b}$ in the same number of unknowns, in which $\mathbf{b}$ is determined from the boundary data. We've shown above that if $\mathbf{b} = \mathbf{0}$ then the system has only $\mathbf{V} = \mathbf{0}$ as a solution, so we conclude that $\mathbf{A}$ is an invertible matrix, and the system is uniquely solvable for any $\mathbf{b}$ (any boundary data).

In summary, by using the mean-value property we generate exactly enough equations in order to solve for all unknown voltages in our resistor network. The solution to this linear set of equations gives us all the voltages of the nodes, as well as the current flow within all resistors (via Ohm's Law).

2.3 Continuation of Cauchy Data

As mentioned above, the continuation of Cauchy data is of central importance to our analysis and algorithms. For ease, we will start by considering only the case of a perfectly rectangular grid. However, understanding this case will be very useful in creating the algorithms necessary to deal with cases in later sections.

First, notice that given the voltage V_1 of a node and the current I from that node to another adjacent node with unknown voltage V_2 through a resistor of known resistance R, we can use Ohm's Law to find the resultant voltage in the second node, for $V_1 - V_2 = IR$, so we can solve for $V_2 = V_1 - IR$. To expand on this idea, suppose we are given Cauchy data on the entire front side (bottom) of a fully rectangular network like that of Figure 1. More specifically, let $V_{j,0}$ denote the voltages along the bottom of the network (the red nodes in Figure 1), where $1 \leq j \leq n-2$; the second subscript "0" indicates the 0th row of the network. We also have the input currents i_j for $1 \leq j \leq n-2$. We can use the to extrapolate and determine the voltages in row 1 of the network, as

$$V_{j,1} = V_{j,0} - i_j \tag{4}$$

for $1 \leq j \leq n-2$, since all resistors are 1 ohm. Note however, that this procedure doesn't determine the side voltages $V_{0,1}$ or $V_{n-1,1}$. Computing these requires knowledge of the boundary conditions on the sides of the network. We will assume in this paper that the sides of the network are insulated (no input current), which forces $V_{0,1} = V_{1,1}$ and $V_{n-1,1} = V_{n-2,1}$. Thus we can determine all voltages in the first row from knowledge of the Cauchy data on row 0. Of course we now the analogous Cauchy data for row 1 which we can use to extrapolate to the voltages for row 2, and so on. The procedure is illustrated graphically in Figure 4, with all side nodes $V_{0,j}$ and $V_{n-1,j}$ omitted for simplicity.

2.4 Propagation by Matrix

The extrapolation procedure above can be cast in a slightly more useful form if make use of the mean value property. First let use define vectors $\vec{a}_i$ in $\mathbb{R}^n$ as

$$(\vec{a}_i)_j = V_{j,i}$$

where $V_{i,j}$ is the voltage at node (i,j) and $(\vec{a}_i)_j$ denotes the jth component of the vector $\vec{a}_i$. Here $0 \leq i \leq m-1$. Thus $\vec{a}_i$ simply encodes the voltage data for row i in the network, omitting the side nodes (which have the same voltages as their adjacent nodes, since we assume zero input current.) The extrapolation procedure above allows use to compute $\vec{a}_i$ from values $\vec{a}_j$ with $j < i$. Note, however, a single row of voltages is not enough to be able to propagate upwards. We can compute $\vec{a}_1$ from $\vec{a}_0$ and the input currents, but we cannot compute $\vec{a}_2$ from $\vec{a}_1$ alone, since computation of the relevant currents from row 1 to row 2 requires knowledge of the row 0 voltages. In general, computation of $\vec{a}_i$ requires knowledge of $\vec{a}_{i-1}$ and $\vec{a}_{i-2}$.

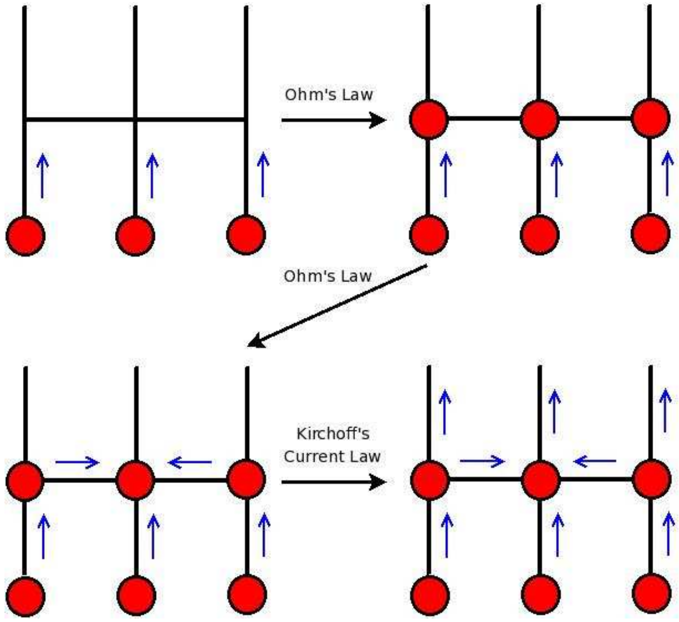

Figure 4: Red means a known voltage and blue means a known current.

To elaborate, let us suppose that the voltages in rows 0 and 1 have been determined. Then the mean value property gives, for any node in row 1 with $1 \leq j \leq n-2$, that

$$V_{j,1} = \frac{1}{4}(V_{j,0} + V_{j,2} + V_{j-1,1} + V_{j+1,1}).$$

We can solve for $V_{j,2}$ as

$$V_{j,2} = 4V_{j,1} - V_{j,0} - V_{j-1,1} - V_{j+1,1}. \tag{5}$$

Note that in the special case that $j = 0$ equation (5) becomes $V_{1,2} = 3V_{1,1} - V_{1,0} - V_{2,1}$ since $V_{0,1} = V_{1,1}$ from the zero input current condition. Similarly we find $V_{n-2,2} = 3V_{n-2,1} - V_{n-2,0} - V_{n-3,1}$.

More generally, the same argument shows that

$$V_{j,i} = 4V_{j,i-1} - V_{j,i-2} - V_{j-1,i-1} - V_{j+1,i-1} \tag{6}$$

with the special cases $V_{1,i} = 3V_{1,i-1} - V_{1,i-2} - V_{2,i-1}$ and $V_{n-2,i} = 3V_{n-2,i-1} - V_{n-2,i-2} - V_{n-3,i-1}$. Each row can be computed from knowledge of the two preceding rows. Note that the computation is linear. As a consequence, it has a convenient matrix formulation.

To propagate upwards using matrix notation we will create a new vector $\vec{b_i}$ that is simply the combination of two layers of voltage:

$$\vec{b_i} = \begin{bmatrix} \vec{a_i} \\ \vec{a_{i-1}} \end{bmatrix}, \quad \vec{b_i} \in \mathbb{R}^{2n}.$$

In these terms, knowledge of the initial frontside data is equivalent to knowing $\vec{b_1}$. With this notation we create a $2n \times 2n$ matrix $\mathbf{M}$ that propagates voltage data up one layer at time. It's easy to verify that that equation (6) can be reformulated as $\vec{b_i} = \mathbf{M}\vec{b}_{i-1}$ where

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & -\mathbf{I}_n \\ \mathbf{I}_n & \mathbf{0}_n \end{bmatrix}, \quad \text{where} \tag{7}$$

$$\mathbf{A} = \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & -1 & 3 \end{bmatrix} \tag{8}$$

and $\mathbf{I}_n$ denotes the $n \times n$ identity matrix, $\mathbf{0}_n$ the $n \times n$ zero matrix. With this construction, the top layer of $\mathbf{M}$ builds the next row of voltages by using the two previous rows in combination with the mean-value property (this is why there are 4's in the middle of $\mathbf{A}$ but a 3 on each end). Simultaneously, the bottom layer "moves" the present top row of voltages down so it can be used with the next iteration of $\mathbf{M}$. Thus, when we consider propagation from this point of view, we can obtain the following lemma:

Lemma 3. *In a fully rectangular network with insulated sides, the Cauchy data on the the bottom uniquely determines the voltage of all nodes in the network.*

Remark 1. *It should be clear that the Cauchy data can in fact be specified on any side of the network. Note also that the boundary conditions on the top of the network play no role in the propagation of the Cauchy data from the bottom.*

2.5 Instability

Of course, few measurements are totally without error, and since we might hope to apply this idea to real world situations, it would be a good idea to look at what effect error has on how data propagates. In fact, propagating the frontside/bottom data inwards is a process highly susceptible to error. Minor differences in initial measurements can cause extremely different results, and even tiny changes due to roundoff error can have serious effects when compounded over multiple iterations. Let's look at a quick example where we introduce a small quantity of random error to the data. Let us consider two vectors $\vec{b_1}$ and $\vec{b_1}^*$,

$$\vec{b_1} = \begin{bmatrix} .875 & .875 & .875 & .875 & .875 & 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

while $\vec{b_1}^*$ has a random error of between 0 and 0.01% in each component, giving

$$\vec{b_1}^* = \begin{bmatrix} .874999125 & .875021125 & .874951875 & .874893 & .87509525 \\ & .999992 & 1.000021 & .999944 & .999905 & 1.000086 \end{bmatrix}.$$

Now compare the results from iterating by M seven times on each vector:

$$M^7\vec{b_1} = \vec{b_8} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & .125 & .125 & .125 & .125 & .125 \end{bmatrix},$$

$$M^7\vec{b_1}^* = \vec{b_8}^* \approx \begin{bmatrix} -.1905 & -.3880 & 2.5116 & -3.9357 & 2.0026 \\ & .0554 & .1260 & .5705 & -.6820 & .5550 \end{bmatrix}.$$

Clearly, the propagation of boundary data is highly unstable. Even with only a minor error present in the initial data, our final results can be very far from accurate. Since it is unlikely that we will be presented with only perfect data in the future, we must find some way to regularize propagation so as to stabilize it, even if at the expense of some accuracy. Since the behavior of $\mathbf{M}^k$ as k increases is largely determine by the eigenvalues of $\mathbf{M}$, an obvious choice would be to consider the eigenvalues of M and identify which ones are particularly large, and thus contributing most to the propagation of error. This will allow us to devise a strategy to combat the instability.

2.6 Regularization through Eigenvalues

2.6.1 Eigenvalues for A

We begin by considering the eigenvalues of the matrix **A**, above, since it has such a deep relation with **M**. From inspection of a few small test cases on can hypothesize that the eigenvalues of **A** are

$$\lambda_k = 4 + 2\cos\left[\frac{k\pi}{n}\right], \quad 1 \leq k \leq n, \tag{9}$$

with corresponding eigenvectors

$$\vec{v_k} = \begin{bmatrix} \vdots \\ \cos\left[(n-k)(j-\frac{1}{2})\frac{\pi}{n}\right] \\ \vdots \end{bmatrix} \}j\text{th component}\,. \tag{10}$$

(In fact, it turns out that **A** is symmetric positive definite, hence its eigenvectors span $\mathbb{R}^n$ and are orthogonal. This is just a version of the *Spectral Theorem*; see any book on linear algebra, e.g., [4].) In this case however, we can in fact prove directly that the above-mentioned quantities are the eigenvalues and eigenvectors for **A**.

Proof. Note that

$$\mathbf{A}\vec{v_k} = \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} \vdots \\ \cos\left[(n-k)(j-\frac{1}{2})\frac{\pi}{n}\right] \\ \vdots \end{bmatrix}$$

$$= \begin{bmatrix} 3\cos\left[(n-k)(\frac{1}{2})\frac{\pi}{n}\right] - \cos\left[(n-k)(\frac{3}{2})\frac{\pi}{n}\right] \\ \vdots \\ -\cos\left[(n-k)(j-\frac{3}{2})\frac{\pi}{n}\right] + 4\cos\left[(n-k)(j-\frac{1}{2})\frac{\pi}{n}\right] - \cos\left[(n-k)(j+\frac{1}{2})\frac{\pi}{n}\right] \\ \vdots \\ -\cos\left[(n-k)(n-\frac{3}{2})\frac{\pi}{n}\right] + 3\cos\left[(n-k)(n-\frac{1}{2})\frac{\pi}{n}\right] \end{bmatrix}.$$

Thus we have three distinct cases that we need to show are equivalent to multiplying by the eigenvalue. We denote these cases "Top", "Middle", and "Bottom". But before beginning to work with the three cases, there are two useful trigonometric identities we should note:

$$\cos(n\alpha) = 2\cos\left[(n-1)\alpha\right]\cos(\alpha) - \cos\left[(n-2)\alpha\right],$$

$$-2\cos\left[(n-k)\frac{\pi}{n}\right] = -2\cos\left(\pi + \frac{-k\pi}{n}\right) = 2\cos\left(\frac{k\pi}{n}\right).$$

We now begin with the Top section:

$$\begin{aligned}
&3\cos\left[(n-k)(\frac{1}{2})\frac{\pi}{n}\right]-\cos\left[(n-k)(\frac{3}{2})\frac{\pi}{n}\right]\\
&=3\cos\left[(n-k)(\frac{1}{2})\frac{\pi}{n}\right]-\Bigg(\cos\left[(n-k)(\frac{1}{2})\frac{\pi}{n}\right]\cos\left[(n-k)\frac{\pi}{n}\right]\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad-\cos\left[(-\frac{1}{2})(n-k)\frac{\pi}{n}\right]\Bigg)\\
&=\ \cos\left[(n-k)(\frac{1}{2})\frac{\pi}{n}\right]\left(4-2\cos\left[(n-k)\frac{\pi}{n}\right]\right)\\
&=\ \left(4+2\cos\left[\frac{k\pi}{n}\right]\right)\cos\left[(n-k)(\frac{1}{2})\frac{\pi}{n}\right]. \qquad\qquad\qquad \checkmark
\end{aligned}$$

We apply a similar method for the Middle:

$$\begin{aligned}
&-\cos\left[(n-k)(j-\frac{3}{2})\frac{\pi}{n}\right]+4\cos\left[(n-k)(j-\frac{1}{2})\frac{\pi}{n}\right]\\
&\qquad\qquad\qquad\qquad\qquad-\cos\left[(n-k)(j+\frac{1}{2})\frac{\pi}{n}\right]\\
=&\ -\cos\left[(n-k)(j-\frac{3}{2})\frac{\pi}{n}\right]+4\cos\left[(n-k)(j-\frac{1}{2})\frac{\pi}{n}\right]\\
&\quad-\left(2\cos\left[(j-\frac{1}{2})(n-k)\frac{pi}{n}\right]\cos\left[(n-k)\frac{\pi}{n}\right]-\cos\left[(j-\frac{3}{2})(n-k)\frac{pi}{n}\right]\right)\\
=&\ \left(4-2\cos\left[(n-k)\frac{\pi}{n}\right]\right)\cos\left[(n-k)(j-\frac{1}{2})\frac{\pi}{n}\right]\\
=&\ \left(4+2\cos\left[\frac{k\pi}{n}\right]\right)\cos\left[(n-k)(j-\frac{1}{2})\frac{\pi}{n}\right]. \qquad\qquad\qquad \checkmark
\end{aligned}$$

Before completing this, we will need one more special identity specifically for the Bottom case:

$$\begin{aligned}
\cos\left[(n-k)\pi\right]\cos\left[\frac{-(n-k)\pi}{2n}\right] &= \cos\left[(n-k)\pi\right]\cos\left[\frac{(n-k)\pi}{2n}\right]\\
\cos\left[(n-k)\pi-\frac{(n-k)\pi}{2n}\right] &= \cos\left[(n-k)\pi+\frac{(n-k)\pi}{2n}\right]\\
\cos\left[(1-\frac{1}{2n})(n-k)\pi\right] &= \cos\left[(1+\frac{1}{2n})(n-k)\pi\right]\\
\cos\left[(n-\frac{1}{2})(n-k)\frac{\pi}{n}\right] &= \cos\left[(n+\frac{1}{2})(n-k)\frac{\pi}{n}\right]. \qquad (11)
\end{aligned}$$

Now with this identity, we can finally show the last case:

$$
\begin{aligned}
& -\cos\left[(n-k)(n-\frac{3}{2})\frac{\pi}{n}\right] + 3\cos\left[(n-k)(n-\frac{1}{2})\frac{\pi}{n}\right] \\
= \; & -\cos\left[(n-k)(n-\frac{3}{2})\frac{\pi}{n}\right] + 4\cos\left[(n-k)(n-\frac{1}{2})\frac{\pi}{n}\right] \\
& \qquad\qquad\qquad -\cos\left[(n-k)(n+\frac{1}{2})\frac{\pi}{n}\right],
\end{aligned}
$$

and by using the exact same method used in the Middle case, we obtain:

$$= \left(4 + 2\cos\left[\frac{k\pi}{n}\right]\right)\cos\left[(n-k)(n-\frac{1}{2})\frac{\pi}{n}\right]. \qquad \checkmark$$

This completes the proof that the eigenvalues and eigenvectors for **A** are as stated in equations (9) and (10).

Note that the eigenvector $\vec{v}_n = [1 \;\; 1 \;\; \cdots \;\; 1]^T$, and also $2 \leq \lambda_k \leq 6$ for all k.

2.6.2 Eigenvalues for M

Now that we have the eigenvalues and eigenvectors of **A**, we will look at **M**. Consider an arbitrary eigenvalue μ of **M** with eigenvector $\vec{w}$, i.e. $\mathbf{M}\vec{w} = \mu\vec{w}$. Instead of looking at $\vec{w}$ as a $2n$-vector though, consider it as the concatenation of two n-vectors, $\vec{w_1}$ and $\vec{w_2}$. Thus, we have

$$\mathbf{M}\vec{w} = \mathbf{M}\begin{bmatrix}\vec{w_1}\\ \vec{w_2}\end{bmatrix} = \begin{bmatrix}\mathbf{A} & -\mathbf{I}\\ \mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\vec{w_1}\\ \vec{w_2}\end{bmatrix} = \begin{bmatrix}\mathbf{A}\vec{w_1} - \vec{w_2}\\ \vec{w_1}\end{bmatrix} = \begin{bmatrix}\mu\vec{w_1}\\ \mu\vec{w_2}\end{bmatrix} = \mu\vec{w}.$$

This gives that $\mathbf{A}\vec{w_1} - \vec{w_2} = \mu\vec{w_1}$ and $\vec{w_1} = \mu\vec{w_2}$. A little rearrangement yields $(\mu^2 I - \mu A + I)\vec{w_2} = \vec{0}$, which is equivalent to

$$\mathbf{A}\vec{w_2} = \left(\mu + \frac{1}{\mu}\right)\vec{w_2}.$$

This implies $\vec{w_2}$ must be an eigenvector of **A** and $\mu + \frac{1}{\mu} = \lambda_k$ for some k. Thus the eigenvalues for **M** come in reciprocal pairs and are the solutions to the equation

$$\mu^2 - \lambda_k\mu + 1 = 0, \quad 1 \leq k \leq n.$$

We label these pairs of solutions as α_k and β_k, with

$$\alpha_k = \frac{\lambda_k - \sqrt{\lambda_k^2 - 4}}{2}, \quad \beta_k = \frac{\lambda_k + \sqrt{\lambda_k^2 - 4}}{2}, \tag{12}$$

and $\alpha_n = \beta_n = 1$. Notice that $\alpha_k < 1 < \beta_k$, while $\alpha_n = \beta_n = 1$. It's easy to check that since $2 \leq \lambda_k \leq 6$ for all k we also have $\beta_k \leq 6$. The eigenvectors for **M** are then the vectors

$$\vec{r_k^1} = \begin{bmatrix}\alpha_k\vec{v_k}\\ \vec{v_k}\end{bmatrix} \quad \text{and} \quad \vec{r_k^2} = \begin{bmatrix}\beta_k\vec{v_k}\\ \vec{v_k}\end{bmatrix}. \tag{13}$$

where $1 \leq k \leq n$. However, we find that $\vec{r_n^1} = \vec{r_n^2} = [11 \cdots 1]^T$. Indeed, it's easy to check that the eigenspace for $\mathbf{M}$ associated to the eigenvalue 1 is really one-dimensional. Thus $\mathbf{M}$ possesses only $2n-1$ independent eigenvectors and hence is not diagonalizable.

These eigenvectors have some interesting orthogonality properties that will be of use. Specifically, we find that

$$\begin{aligned} \vec{r}_j^1 \cdot \vec{r}_k^1 &= (\alpha_j \alpha_k + 1)\vec{v}_j \cdot \vec{v}_k = 0, \text{ if } j \neq k \\ \vec{r}_j^1 \cdot \vec{r}_k^2 &= (\alpha_j \beta_k + 1)\vec{v}_j \cdot \vec{v}_k = 0, \text{ if } j \neq k \\ \vec{r}_j^2 \cdot \vec{r}_k^2 &= (\beta_j \beta_k + 1)\vec{v}_j \cdot \vec{v}_k = 0, \text{ if } j \neq k. \end{aligned} \tag{14}$$

That is, each eigenvector for $\vec{r}_j^1$ for $\mathbf{M}$ is orthogonal to all other eigenvectors except its "partner" $\vec{r}_j^2$ (and itself, of course); similarly, each eigenvector for $\vec{r}_j^2$ for $\mathbf{M}$ is orthogonal to all other eigenvectors except its "partner" $\vec{r}_j^1$. Let us define a vector

$$\mathbf{q} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{15}$$

consisting of n ones and n zeros. The vector $\mathbf{q}$ is not an eigenvector for $\mathbf{M}$, but we will use $\mathbf{q}$ as a kind of stand-in for the missing eigenvector. It's easy to verify that

$$\mathbf{M}\vec{r}_n^1 = \mathbf{M}\vec{r}_n^2 = \mathbf{q} \tag{16}$$

while

$$\mathbf{M}^p \mathbf{q} = \mathbf{q}_p := \begin{bmatrix} (p+1) \\ \vdots \\ (p+1) \\ p \\ \vdots \\ p \end{bmatrix} \tag{17}$$

for any $p \geq 1$, and that $\mathbf{q}$ is orthogonal to all of the $\vec{r}_j^1$ and $\vec{r}_j^2$ except in the case $j = n$.

2.6.3 Regularizing Cauchy Continuation

Let $\mathbf{v}$ be any vector in $\mathbf{R}^{2n}$. We can stabilize the computation of $\mathbf{Mv}$ (and hence $\mathbf{M}^p\mathbf{v}$) as follows. First, decompose $\mathbf{v}$ as

$$\mathbf{v} = \sum_{k=1}^{n-1} c_k \mathbf{r}_k^1 + \sum_{k=1}^{n-1} d_k \mathbf{r}_k^2 + c_n \mathbf{r}_n^1 + d_n \mathbf{q} \tag{18}$$

where $\mathbf{q}$ is as defined in equation (16); it's easier to break out the $\mathbf{r}_n^1$ term separately. If we multiply by $\mathbf{M}^p$ for some $p \geq 1$ and use the above-discussed properties of the eigenvectors for $\mathbf{M}$ and equation (16) we obtain

$$\begin{aligned}\mathbf{M}^p\mathbf{v} &= \sum_{k=1}^{n-1} c_k\alpha_k^p\mathbf{r}_k^1 + \sum_{k=1}^{n-1} d_k\beta_k^p\mathbf{r}_k^2 + c_n\mathbf{M}^p\mathbf{r}_n^1 + d_n\mathbf{M}^p\mathbf{q} \\ &= \sum_{k=1}^{n-1} c_k\alpha_k^p\mathbf{r}_k^1 + \sum_{k=1}^{n-1} d_k\beta_k^p\mathbf{r}_k^2 + c_n\mathbf{q}_{p-1} + d_n\mathbf{q}_p \qquad (19)\end{aligned}$$

where $\mathbf{q}_{p-1}, \mathbf{q}_p$ are as defined by equation (17). Note that since $\alpha_k < 1$ for $1 \leq k \leq n-1$, the terms in the first summation above decay as $p \to \infty$. The vector $\mathbf{q}_p$ grows, but only arithmetically. The main offending terms are those involving β_k^p for $1 \leq k \leq n-1$, which grow exponentially in p since $\beta_k > 1$. Our regularized approach to computing $\mathbf{M}^p\mathbf{v}$ involves computing the coefficients α_k, β_k in the decomposition (18), then using (19) to approximate $\mathbf{M}^p\mathbf{v}$ but with the terms involving β_k^p modified to control growth.

Specifically, we reconstruct a regularized version $T_r(\mathbf{v}) \approx \mathbf{M}\mathbf{v}$ (note we are only taking the first power of $\mathbf{M}$ here) as

$$T_r(\mathbf{v}) = \sum_{k=1}^{n-1} c_k\alpha_k\mathbf{r}_k^1 + \sum_{k=1}^{n-1} d_k\tilde{\beta}_k\mathbf{r}_k^2 + c_n\mathbf{q} + d_n\mathbf{q}_1 \qquad (20)$$

where $r \in [0,1]$ is a chosen regularization parameter and

$$\tilde{\beta}_k = \frac{\beta_k}{(\beta_k - 1)r + 1}. \qquad (21)$$

Of course T_r simply denotes the linear operator on $\mathbf{v}$ defined by the right side of (20). When $r = 0$ we have $\tilde{\beta}_k = \beta_k$, so that $T_r(\mathbf{v}) = \mathbf{M}\mathbf{v}$ and the reconstruction is unregularized. But when $r = 1$ we have $\tilde{\beta}_k = 1$, maximum regularization. For $0 < r < 1$ we find $1 < \tilde{\beta}_k < \beta_k$. Higher powers of $\mathbf{M}$ are approximated as $\mathbf{M}^p\mathbf{v} \approx T_r^p(\mathbf{v})$. Thus when $r > 0$ the growth of β_k^p is damped, and at the extreme $r = 1$ we have no growth at all. We will illustrate the effectiveness of this regularization scheme below.

It's worth noting that it's easy to compute the c_k and d_k in the decomposition (18). If we dot both sides of (18) with $\mathbf{r}_m^1$ and use orthogonality we find

$$\begin{aligned}\mathbf{v}\cdot\mathbf{r}_m^1 &= c_m\mathbf{r}_m^1\cdot\mathbf{r}_m^1 + d_m\mathbf{r}_m^1\cdot\mathbf{r}_m^2 \\ &= c_m(\alpha_m^2+1)(v_m\cdot v_m) + d_m(\alpha_m\beta_m+1)(v_m\cdot v_m) \\ &= \frac{n}{2}\left(c_m(\alpha_m^2+1) + d_m(\alpha_m\beta_m+1)\right). \qquad (22)\end{aligned}$$

Similarly

$$\begin{aligned}\mathbf{v}\cdot\mathbf{r}_m^2 &= c_m\mathbf{r}_m^1\cdot\mathbf{r}_m^2 + d_m\mathbf{r}_m^2\cdot\mathbf{r}_m^2 \\ &= c_m(\alpha_m\beta_m+1)(v_m\cdot v_m) + d_m(\beta_m^2+1)(v_m\cdot v_m) \\ &= \frac{n}{2}\left(c_m(\beta_m\alpha_m+1) + d_m(\beta_m^2+1)\right). \qquad (23)\end{aligned}$$

The pair of linear equations (22)/(23) can be solved for c_m, d_m, for each $1 \leq m \leq n$.

3 Varying Backside Case

3.1 Model

In the case of a resistor grid with a backside that varies in height, we begin by making the following modeling assumptions:

- We know the width of the grid.
- Voltage and current are known on the frontside (graphically, the bottom).
- The sides of the grid are insulated.
- Each column has a certain "height," the row in which the column hits zero voltage. At this height and above it, the nodes in the column are set at 0 volts. (Perfect conductance in the grid above a column's height.)
- Current can leave through the zero volt nodes in any direction, i.e. they still affect the voltage of adjacent nodes.

For a visual example of a possible backside configuration, see Figure 5; note that the horizontal "side" resistors have been omitted from the figure. With bottom row as row 0 the column heights are 3,3,2,3, and 4, respectively. We should note that we still enforce the condition that each interior node is the average of its four neighbors. If column j has height p then we force $V_{j,k} = 0$ for $k \geq p$ in the system of linear equations that govern the network voltages.

The motivation for this model actually comes from steady-state thermal inverse problems, which are governed by essentially the same equations as the electrical conduction problem. In this case voltages are replaced by temperatures and we assume that the back surface of our sample is in contact with an environment which is at a constant temperature. We assume, without loss of generality, that this constant is zero. In the electrical context we would assume the corroded portion of the backside is held at ground, i.e., zero volts.

3.2 Algorithm

Given the frontside data on a grid (we assume it has no voids), how do we recover the variations in its backside? We can find this information out by use of an algorithm, but to do so, we will first need the following fundamental theorem:

Theorem 4. *If the voltage of every node on the front is greater (less) than 0, then the voltage of all nodes below the backside must be as well.*

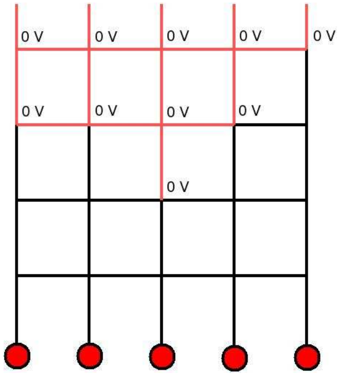

Figure 5: Heights: 3, 3, 2, 3, 4.

Proof. This follows from Theorem 2. First note that since the side boundaries are insulated, the nodes in the adjacent columns (nodes at locations $(1, j)$ and $(n-2, j)$) have the same voltage as the adjacent boundary nodes (locations $(0, j)$ or $(n-1, j)$). Thus if the minimum for the network occurs anywhere below the backside, it is attained at some interior node. If this is the case Theorem 2 (and the remarks after that theorem) show that the voltage on the network is identically zero, a contradiction. Thus the minimum must occur only at the backside, and hence all other nodes have positive voltage. A similar argument works for the case in which the frontside nodes have negative voltage. □

Therefore, if the front voltages are all greater than zero, we can propogate voltages through the grid, and if we see a node at zero volts, we know we've hit the backside. This allows us to easily create an algorithm to find the backside of a grid, but it requires that the front voltages are all positive. Luckily, it is quite reasonable to assume that we have control over the voltage on the front since we have access to the frontside (where we're taking data from in the first place).

With the above assumptions, it's easy to posit the following algorithm.

Algorithm (Varying Backside)**.** Assuming the model given earlier in this section and a strictly positive (or strictly negative) voltage on the front nodes, the following set of steps will recover the backside:

1. Use the voltage and current for the front nodes to determine the voltages on the first inner row.
2. Use the voltages found in rows $k-1$ and k and the methods described in Section 2.3 to calculate the voltages for row $k+1$.

3. If any of the nodes in the latest row are less than or equal to 0 (or some threshold, say $5 \cdot 10^{-3}$), set that node and all the above nodes in that column to voltage 0.

4. Repeat from Step 2 until we obtain a row that is all 0's.

5. The first row in which a column contains a node with zero voltage yields the height of that column.

We now proof that the algorithm does in fact work:

Proof. By Lemma 3, we know each row is uniquely determined by the two beneath it. From the model and Theorem 4, we will encounter a node with zero voltage if and only if the node is on the backside. Thus encountering a zero-voltage node in a column is a signal that all of the nodes above it are at zero volts as well. As long as we make sure to include the effect of these zeros as we continue to propogate, the methods behind propagation are still well-founded. We can continue in this fashion arbitrarily long, eventually resulting in a row containing all zeros. Thus we will have found the height of each column. □

For a visual example of the algorithm, see Figure 6.

We now have a sound mathematical foundation for our algorithm. However, the above proof relies on the assumption that we start with perfect data and compute without round-off error. This is an unlikely occurrence in the real world, and as we saw earlier with the instability of propagation, tiny errors in measurement can result in large errors in our results. In fact, the algorithm described above is highly susceptible to error. Take the following example of a ten-wide grid with 10 volts on each front node. Given the following column heights

$$\begin{bmatrix} 9 & 10 & 9 & 8 & 6 & 5 & 6 & 7 & 8 & 9 \end{bmatrix},$$

the voltage will induce a current on the front nodes, which we compute to an accuracy of 10 significant figures. If we feed this voltage and current information into the algorithm, it outputs a backside of

$$\begin{bmatrix} 9 & \infty & 9 & 8 & 6 & 5 & 6 & 7 & 9 & 9 \end{bmatrix},$$

which is fairly good.[1] However, if we introduce a tiny error of 0.01% to the reading of the voltage from the eighth node, things go much more poorly:

$$\begin{bmatrix} 9 & \infty & 8 & \infty & 6 & 6 & 6 & \infty & 6 & \infty \end{bmatrix}.$$

Clearly, an understanding of how to best control error in the algorithm is highly desirable.

[1] Note that an entry of ∞ means that the algorithm never finds a 0 (or negative) voltage node in the column, and so cannot determine the column's height.

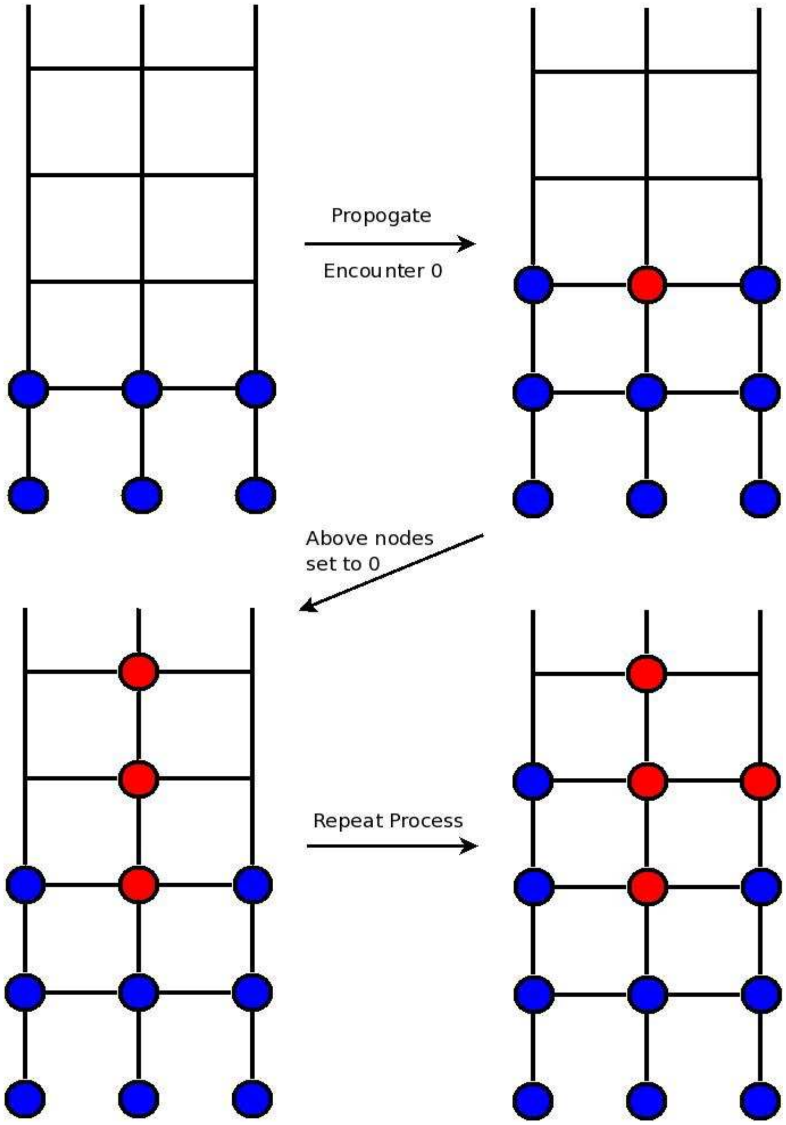

Figure 6: Blue indicates known nonzero voltage. — Red indicates 0 voltage.

3.3 Addressing Instability

There are several (interrelated) factors that influence the instability inherent in the process: the shape of the network (width versus height), the eigenvalues of the Cauchy propagation matrix $\mathbf{M}$, and the applied voltage.

The first and best way to address instability is the one we do not actually have control over: the shape of the grid. Since error propagates through repeated iteration of M and causes small initial errors to grow exponentially, the fewer times we have to iterate M, the better. Wide, short grids are the most easily recoverable because they require fewer iterations of M on the initial data. If, on the other hand, we try to recover a grid whose average height is considerably taller than its width, the chance of recovery goes to almost zero. This difficulty in recovering tall grids is fundamental to the nature of this inverse problem. Since minute errors can destroy the usefulness of the algorithm and it is impossible to take these measurements without error, there will always be a limit to how far we can "see" into the grid. If a grid is too tall for the accuracy of our measurement and computation, the algorithm will be unable to return anything useful. Thus, it is important to choose what kind of grids we apply the algorithm to with care. If we know the grid we are dealing with is taller than it is wide, the backside is most likely unrecoverable.

However, in the case where trying to recover a grid's shape is not a hopeless problem, there are some techniques to slightly improve the accuracy of the algorithm. We will start by looking at regularization through eigenvalues. The essential analysis has already been done in Section 2.6.3. We will begin by looking at how well this regularization works on a perfectly rectangular grid.

In section 2.5, we explored iterations of M on two vectors: $\vec{b_1}$ and a comparison vector with error, $\vec{b_1}^*$. Using the model we now have for a varying backside, we can see that $\vec{b_1}$ is equivalent to the boundary data given by a 5-wide (that is, five interior columns) grid that has a height of 8 in each column. (Note that $\vec{b_1}$ has ten entries—five for the 0th row voltages, five for the first row voltages, the latter computed using the input current.) If we were to use the algorithm on $\vec{b_1}$, this is exactly the result we would get. But what would be the result if we used the algorithm on $\vec{b_1}^*$? Using the regularization method given in 2.6.3 and a threshold[2] of $5 \cdot 10^{-3}$, we obtain a backside of

$$\begin{bmatrix} 8 & 8 & 8 & 8 & 8 \end{bmatrix}.$$

Not bad at all, especially when compared to the backside $\vec{b_1}^*$ would give without regularization:

$$\begin{bmatrix} 8 & 8 & \infty & 7 & \infty \end{bmatrix}.$$

What about a larger quantity of error in our initial measurements? Say, something ridiculous, like a random error of between 0 and 10% in each com-

[2] Obtaining perfect 0's on the backside is almost impossible due to roundoff error, so it is necessary to introduce a range of values that is acceptable. Thus, any value less than the threshold is considered a 0 for purposes of finding the backside.

ponent of $\vec{b_i}$, which we'll call $\vec{b_i}^{**}$:

$$\vec{b_1}^{**} = \begin{bmatrix} 0.875 & 0.917875 & 0.821625 & 0.8855 & .788375 & \\ & 0.964 & 1.029 & 0.946 & 0.939 & 0.970 \end{bmatrix}.$$

Let's begin at looking at what result we obtain from the algorithm acting on $\vec{b_i}^{**}$ without regularization:

$$\begin{bmatrix} 4 & \infty & 3 & \infty & 3 \end{bmatrix}.$$

Now compare that to using the algorithm with regularization:

$$\begin{bmatrix} 9 & 9 & 9 & 9 & 9 \end{bmatrix},$$

which, while not perfect, is still pretty astonishing.

It seems that regularization works extremely well when we are dealing with a perfectly rectangular grid. But what about in the case of a varying backside? Perhaps there is something about how the regularization works that makes it very good at seeing through error on a perfectly rectangular grid, but not nearly as good when the backside has variations. Let us consider a 4-wide grid (4 interior columns) with column heights of

$$\begin{bmatrix} 6 & 5 & 4 & 5 \end{bmatrix}$$

and 1 volt on each of the front nodes. This induces a boundary data vector of

$$\vec{b_1} = \begin{bmatrix} 0.781639 & 0.777847 & 0.773377 & 0.772440 & \\ & 1 & 1 & 1 & 1 \end{bmatrix}.$$

We start by applying the algorithm on $\vec{b_1}$ without any regularization and no noise in the data. With an threshold of $5 \cdot 10^{-3}$, the algorithm returns heights of

$$\begin{bmatrix} 6 & 5 & 4 & 5 \end{bmatrix},$$

which is not too surprising since there was no error in $\vec{b_1}$ and none of the heights were very large. Now let's look at what the algorithm returns when we apply maximum regularization, by taking $r = 1$ in equation (21). We obtain heights

$$\begin{bmatrix} 5 & 5 & 5 & 5 \end{bmatrix}.$$

Clearly this is an issue. Even with no error present and an easily recoverable backside, applying regularization gives an incorrect result. Not only that, but it seems applying regularization can cause the algorithm to instead return the average height over the entire backside for each column's height.

In fact, this averaging property of the regularization process seems to occur universally, no matter what sort of backside we try to recover. While the regularization process is able to significantly decrease damage caused by propagation of initial error, it does so by sacrificing almost all of the information contained in the $\vec{r_i^2}$ eigenvectors. Apparently, the information for the backside's average height is contained in the rest of the modified eigenspace, whereas the $\vec{r_i^2}$ eigenvectors contain the differences in columns. Thus, it makes perfect sense that a perfectly rectangular grid is highly compatible with regularization, but a grid with a varying backside is not. We clearly need to fine-tune our regularization process.

The upshot is that while $r = 0$ results in instability, the choice $r = 1$ obliterates essential information in the data. Clearly some compromise is needed. In the examples that follow we simply choose r by trial and error, but it's clear the further research on the optimal choice for r is needed. The best choice will certainly depend on the grid width and height and noise level, and perhaps also some a priori information about the "likely" shape of the backside. This leads us to our last influential factor in the stability of this procedure.

The input voltage chosen can have a large effect on our ability to recover the backside. The obvious extreme case is to use identically zero input voltages, which makes it impossible to see anything. We now examine the possibility of choosing our input voltages based on hypothetical backside configurations. Say we do not know the exact configuration of our grid, but we are confident that it is one of two configurations: G_1 or G_2. To find out which grid configuration we have, it would behoove us to find a "maximally differentiating voltage:" a voltage that will induce a current in our grid that allows us to best determine if we are working with G_1 or G_2.

Begin by noting that given a grid G and an initial voltage vector $\vec{a_0}$ on the frontside of the grid, a series of linear equations automatically determines the corresponding input current vector $\vec{i}$ on the frontside. Furthermore, by Lemma 3, if we have two distinct grids G_1 and G_2, then given the same $\vec{a_0}$, the grids must yield unique current vectors $\vec{i_1}$ and $\vec{i_2}$. Thus, if we can find an $\vec{a_0}$ such that $\vec{i_1}$ and $\vec{i_2}$ are as "different" as possible, when we apply $\vec{a_0}$ to an unknown grid, it should help us to determine if we are working with G_1 or G_2. For our purposes, we will measure difference as the mean squared difference between $\vec{i_1}$ and $\vec{i_2}$. In summary, we wish to find an $\vec{a_0}$ that maximizes the quantity $||\vec{i_1} - \vec{i_2}||^2$ (the usual Pythagorean norm in $\mathbb{R}^n$). Furthermore, since we could arbitrarily increase the size of this quantity by choosing ever larger magnitudes for $\vec{a_0}$, we will impose the added requirement that $||\vec{a_0}|| = 1$.

To find this maximally differentiating voltage, it is important to notice that since $\vec{i}$ is determined entirely by linear equations, there exists a " voltage-to-current matrix" $\mathbf{C}$ such that for any input voltage $\vec{a_0}$, we have $\vec{i} = \mathbf{C}\vec{a_0}$. Let $\mathbf{C}_1$ and $\mathbf{C}_2$ denote the voltage-to-current matrices for grids G_1 and G_2, respectively. Using these matrices and our requirement that $||\vec{a_0}|| = 1$, we can rephrase our goal as finding an initial voltage vector $\vec{a_0}$ that maximizes the quantity

$$\frac{||\mathbf{C}_1\vec{a_0} - \mathbf{C}_2\vec{a_0}||^2}{||\vec{a_0}||^2}. \tag{24}$$

Let $\mathbf{D} = \mathbf{C}_1 - \mathbf{C}_2$ and notice that

$$||\mathbf{C}_1\vec{a_0} - \mathbf{C}_2\vec{a_0}||^2 = ||\mathbf{D}\vec{a_0}||^2 = (\mathbf{D}\vec{a_0})^T\,\mathbf{D}\vec{a_0} = \vec{a_0}^T(\mathbf{D}^T\mathbf{D})\vec{a_0} \tag{25}$$

where "T" denotes the transpose operator. Thus, we can once again rephrase our goal: we are trying to find $\vec{a_0}$ such that it maximizes the quantity

$$\frac{\vec{a_0}^T(\mathbf{D}^T\mathbf{D})\vec{a_0}}{||\vec{a_0}||^2}. \tag{26}$$

But this is just the Rayleigh quotient of the matrix $(\mathbf{D}^T\mathbf{D})$. A standard result in linear algebra (see, for example, [4], Section 4.2.) shows that the Rayleigh quotient reaches its maximum value when $\vec{a_0}$ is the eigenvector corresponding to the largest eigenvalue of $\mathbf{D}^T\mathbf{D}$. Thus, to find our maximally differentiating voltage $\vec{a_0}$, we simply need to find the eigenvector corresponding to the largest eigenvalue of the matrix $\mathbf{D}^T\mathbf{D}$, which can be done by a variety of numerical methods.

Note that if necessary we can actually compute the voltage-to-current matrix $\mathbf{C}$ for any network by using our ability to solve the forward problem, as follows. It's easy to see that (since the forward problem is linear), if we set $\vec{a_0} = \vec{e_k} = [0, \ldots, 0, 1, 0, \ldots, 0]^T$ (the k-th standard basis vector) and solve the forward problem then the resultant $\vec{i}$ is equal to the k-th column of C. From there, it is a simple matter to assemble C from these columns.

Now that we can find a maximally differentiating voltage $\vec{a_0}$ given two hypothetical grids G_1 and G_2, how do we use this in conjunction with the algorithm? One approach is to let G_1 denote the undamaged or "starting" configuration that the grid should have, and let G_2 be a grid dictated by experience with the type of damage encountered. The resulting optimal input voltage should give good images of any backside damage similar to past damage.

Another approach is to take G_1 as above and obtain G_2 as follows: put a voltage on the grid (something simple, say $[1, \ldots, 1]$), take the resultant current, and then feed this starting information into the algorithm. The algorithm will result in a backside which, while unlikely to be correct, at least gives us a vague idea of what the backside might look like. We use this reconstruction in place of G_2, and then use these two grids to find a maximally differentiating voltage. We can even continue in such a manner, repeatedly using the newly generated $\vec{a_0}$ vectors to refine our educated guess and make a new hypothetical grid.

While regularization and maximally differentiating voltages are useful tools on their own, it is best to combine them. By using small quantities of regularization in tandem with repeated use of the methods given in the above paragraph, the depth of accuracy in the grid could normally be improved by one or two nodes. It is quite possible that this could still be improved further, but unlikely that very significant improvements could be made. Ultimately though, the ability to recover a grid's backside comes down to the shape of the grid. As the grid becomes taller and taller for a fixed width, the amount of noise required to destroy the initial data becomes less and less. The most important thing to keep in mind when addressing instability is whether or not the problem is realistically solvable.

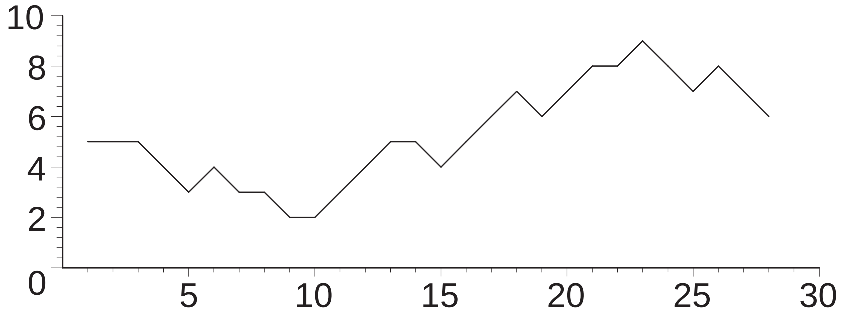

Figure 7: Column height versus column index.

3.4 Example

Let G denote a network grid with width 30 (28 internal node columns) and a maximum column height of 10 (the bottom or zeroth row, up to row 9). We impose column heights for columns 1 through 28 of

$$5, 5, 5, 4, 3, 4, 3, 3, 2, 2, 3, 4, 5, 5, 4, 5, 6, 7, 6, 7, 8, 8, 9, 8, 7, 8, 7, 6$$

In Figure 7 we show a plot of the column heights versus column index for $1 \leq k \leq 28$. We impose an input voltage of 1.0 volts on each node along the frontside of the network, then solve the forward problem. We print out the input voltages and (solved for) input currents along the frontside, but to each number we add a uniformly distributed random error in the range $[-10^{-4}, 10^{4}]$. This becomes the input data to our inverse solver.

When we run the inverse solver with no regularization and threshold 0.005 we obtain column heights

$$5, 7, 5, 5, 3, 5, 3, 3, 2, 2, 3, 4, 6, 5, 4, 6, 6, 7, \infty, 6, \infty, 8, 7, \infty, 7, \infty, 6, 7.$$

A plot showing these columns and the original true column heights is shown in Figure 8, with the infinity height columns shown as height 10.

When we regularize with regularization parameter $r = 1$ we obtain column heights

$$5, 4, 4, 4, 4, 4, 4, 4, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5.$$

A plot showing these columns and the original true column heights is shown in Figure 9, with the infinity height columns shown as height 10.

It should be clear that neither extreme is ideal. Some compromise is needed. In Figure 10 we show the reconstruction obtain with $r = 0.1$. The situation is considerably improved; the exception is when the backside is far from the frontside, as in columns 21 to 27—here, the backside is “in the dark” and hence the estimate is relatively unstable. No choice for r will stabilize this portion of the estimate (without simply making the region flat.)

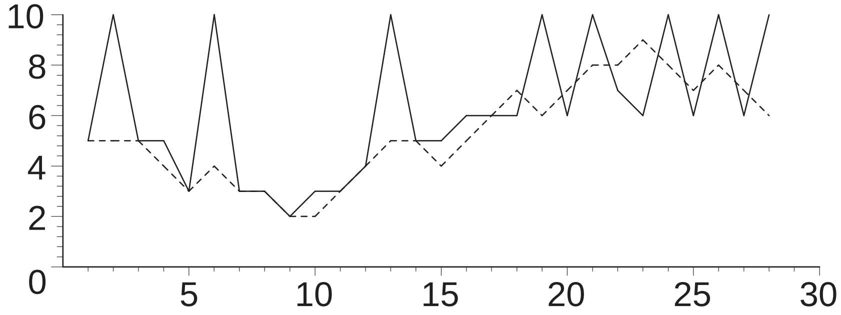

Figure 8: Column height versus column index, recovered (solid) and true (dashed), no regularization.

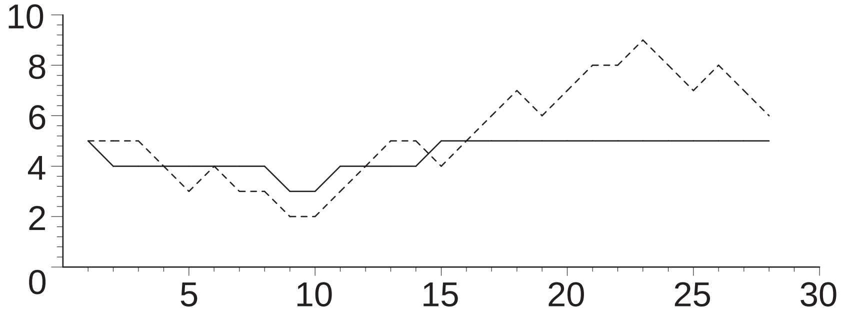

Figure 9: Column height versus column index, recovered (solid) and true (dashed), maximum regularization.

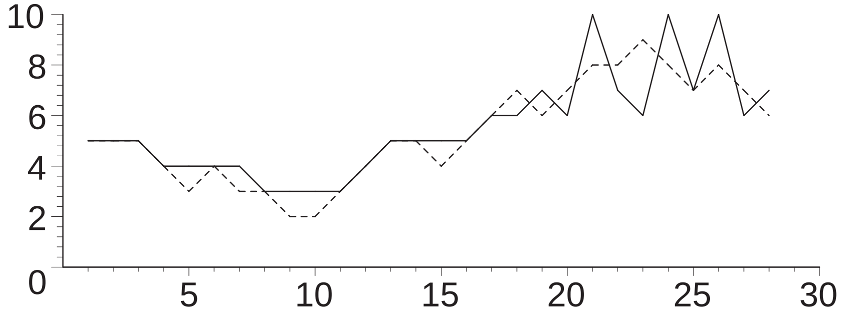

Figure 10: Column height versus column index, recovered (solid) and true (dashed), $r = 0.1$.

Procedures for choosing regularization parameters abound in the inverse problems literature, and further investigation into automatic parameter selection for this inverse problem would be a good topic for a future REU group.

4 The Missing Nodes Case

In this section, we deal with the problem of finding a missing interior node. The goal is to pass electrical current (or later, heat flux) through a discrete network of resistors, then measure the voltages (or temperatures) on all or part of the outer boundary. Using this information, we then determine the location of the missing node(s) within the resistor network. One could use several input currents and measured voltages to accomplish this task, but we focus on the case in which only one input current-boundary voltage pair is available.

4.1 Necessary Assumptions and Theorems

The following are assumptions necessary in the formulation of the problem.

1. If a node is missing, the resistors immediately neighboring it are assumed to be disconnected at the node. Effectively, these resistors now have infinite resistance, hence no current flows through them.

2. Apart from the above mentioned resistors, all other resistors have a resistance of 1Ω.

We next state the theorems necessary in order to formulate the algorithm for this problem.

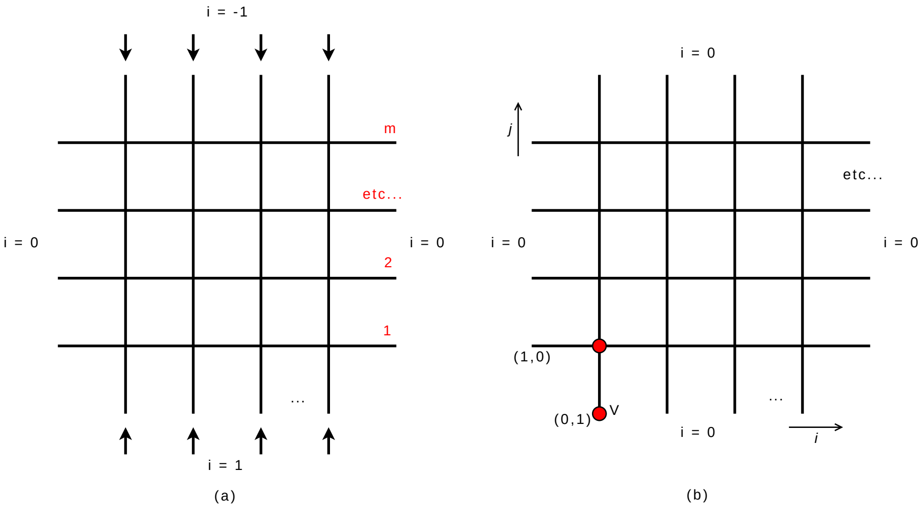

Figure 11: G and $f-g$

Lemma 5. *Consider a fully rectangular $m \times n$ resistor grid G with no missing nodes, input current of 1 (amp) through every resistor from the bottom, input current of -1 amp through every resistor from the top and insulated sides (Figure 11(a)). (Note that an input current of -1 means we are actually pulling current OUT of the top.) Then the nodal voltages are uniquely determined up to an additive constant. Moreover, the voltages are constant across horizontal rows.*

Proof. Choose any real number V and assign a voltage of $V + m + 1$ to each node on the bottom (zero) row. Assign voltages to the nodes in rows 1 to $m - 1$ as

$$\begin{aligned}
\text{Row 1} &= V + m \\
\text{Row 2} &= V + m - 1 \\
&\cdots \\
\text{Row m} &= V + 1 \\
\text{Top nodes} &= V
\end{aligned}$$

It's easy to check that the net current into each node is zero and that Ohm's Law is obeyed, hence this is a solution. Precisely one amp flows through all vertical resistors, and no current flows through the horizontal resistors.

To see that all solutions are of this form (constant voltage across each row), let $f_{j,k}$ and $g_{j,k}$ be two different solutions on the grid G. Then $f - g$ (that is, voltage $f_{j,k} - g_{j,k}$ at node (j, k)) is also a solution and is as shown in Figure 11(b). Note that $f - g$ has zero input current on all exterior resistors. From the Maximum Principle, Theorem 2, we know that the maximum value of the solution $f - g$ is attained on the boundary of the grid. Suppose, for example (and without loss of generality), that this voltage occurs at node $(1, 0)$ (as illustrated) and equals some constant B. But since there is no current in the attached resistor, the voltage at node $(1, 1)$ would also be B, and the maximum is attained at an interior node. From the Maximum principle the solution throughout the grid must be constant, that is $f_{j,k} - g_{j,k} = C$, which proves the lemma. □

Remark 2. *It's worth noting that the above argument for uniqueness up to an additive constant applies to any situation in which we specify only input currents, but no voltages. Moreover, since all solutions in this situation differ only by an additive constant we can normalize any solution by requiring that the voltages around the outer nodes sum to zero. It's easy to see this uniquely determines the solution. We will assume this has been done.*

Theorem 6. *Consider a perfectly rectangular resistor grid G_1 with no missing nodes, input current of 1 amp through every resistor from the bottom, input current of -1 amp through every resistor from the top, and insulated sides. Let G_2 be a rectangular resistor grid which is identical to G_1 except for a single missing interior node (see Figure 12). We assume the same current is input to G_2. Then it is impossible for G_1 and G_2 to have identical boundary voltages along*

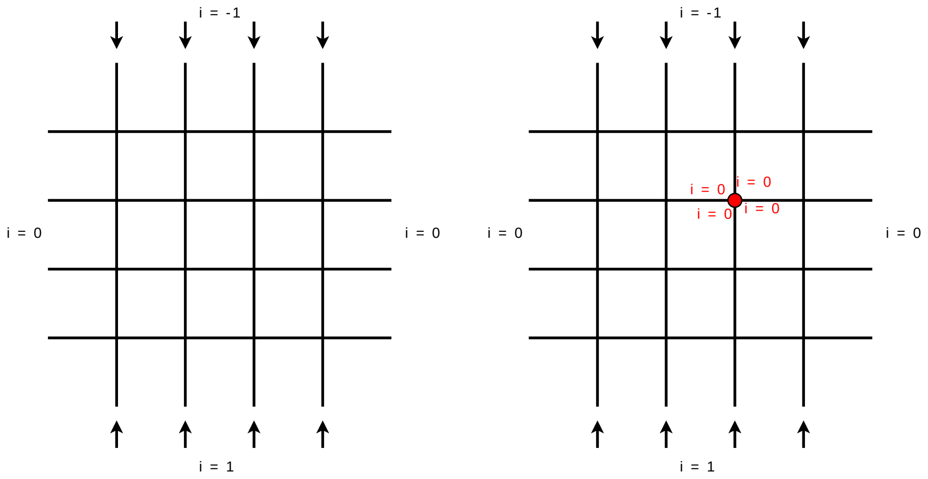

Figure 12: Resistor grids G_1 and G_2

the frontside (bottom row) or backside (top row), assuming the normalization in Remark 2.

Proof. We use proof by contradiction. Suppose the missing node for G_2 is in row $r \geq 1$, say at position (p, r), and suppose G_1 and G_2 have identical Cauchy data on the frontside. We can continue the Cauchy data into both networks using the techniques of Section 2.3 and see that both grids must have the same nodal voltages up to row $r - 1$. Now the node at position $(p, r - 1)$ in G_1 has 1 amp of current entering from below, no current flowing in from the horizontal resistors, and 1 amp of current exiting through the vertical resistor above. Since all nodes in G_2 up to row $r-1$ have the same voltages as G_1 we may also conclude that the node $(p, r-1)$ in G_2 has 1 amp of current entering from below, no current flowing in from the horizontal resistors. But since the resistor from $(p, r - 1)$ to (p, r) is disconnected, there is nowhere for the current to go, violating conservation of charge. We conclude that G_1 and G_2 cannot have the same voltage data along the bottom row. □

Theorem 6 guarantees that if have a network with a single missing node, a specific type of input current will allow us to distinguish this from the graph in which no nodes are missing. It's easy to see that the proof of the theorem generalizes to a graph with more than one missing node. What is more difficult, however, is to determine whether or not the missing node(s) are uniquely determined by this (or some other) specific input current patterns. This would be a good topic for future work.

4.2 The Missing Node Algorithm

In this section we examine an algorithm for locating a single missing node in a rectangular network. The algorithm for this problem uses the Mean Value Property in order to determine the location of the missing node in the discrete network of resistors. It is quite similar to the varying backside algorithm, in that we continue Cauchy data in from one or more sides of the network. The location of the missing nodes are identified via the indicative current.

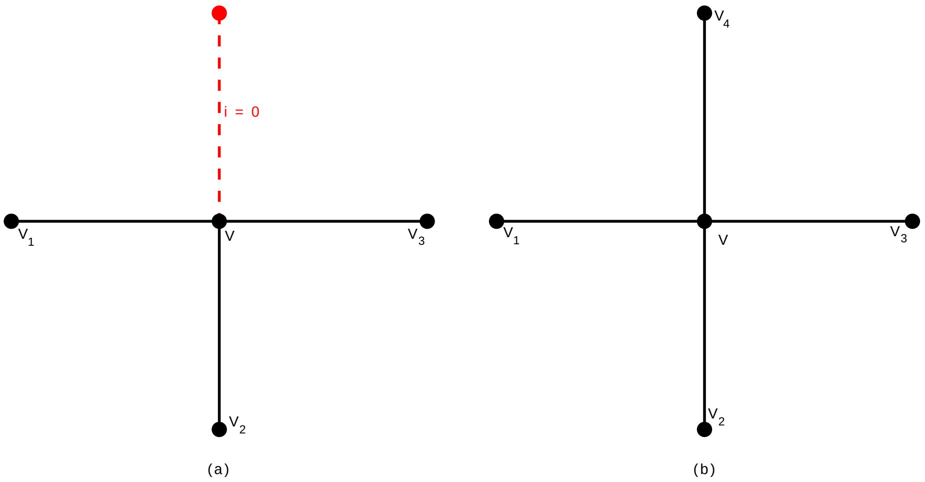

Figure 13: Comparing two systems of nodes

Consider Figure 13(a) with nodal voltages as indicated. By the Mean Value Property,

$$V = \frac{1}{3}(V_1 + V_2 + V_3) \tag{27}$$

However, we are running the algorithm on a network in which we do not know where the missing node is. Hence, the algorithm would calculate the voltage V_4 as though it is attached by a resistor of 1Ω to the center node (as in Figure 13(b)), using the equation:

$$V = \frac{1}{4}(V_1 + V_2 + V_3 + V_4) \tag{28}$$

Note that equations (27) and (28) are equivalent if and only if $V_4 = V$. We arrive at the conclusion that there is no current flow in the resistor that connects these two nodes. The fact that there is no current flow in the resistor is the indicator of a missing node at the position directly above that resistor.

Below, we show an example of how the algorithm works. In Figure 14(a), we begin with the Cauchy Data at the bottom row nodes, as well as the assumption that all other boundary nodes are insulated. The next row of voltages are obtained by Ohm's Law (Figure 14(b)), followed by using the Mean Value

Property for the subsequent rows of nodes(Figure 14(c))—this is simply the continuation of the Cauchy data as in Section 2.3; it can also be regularized as previously discussed. Note that the voltage of the missing node is calculated to be the same as that of the node directly below it, causing the resistor between them to have zero current flow (Figure 14(d)). At this point we know which node is missing. If we make the a priori assumption that the network is missing only one node, we are done.

If, however, we believe that more than one node is missing then we can continue the algorithm. The algorithm is structured such that the missing node is assigned a value for its voltage, even though this voltage should be 'undefined'. As a result, this value is used in the algorithm to calculate voltages of other nodes through the Mean Value Property, which as a result are incorrect. We examine this a bit in the next section.

4.3 Limitations to the Algorithm

4.3.1 The "Cone of Ignorance"

As can be seen in Figure 15(a), since voltage V_3 is undefined, the value of V_4 should be calculated as follows:

$$V = \frac{1}{3}(V_1 + V_2 + V_4)$$

$$\Rightarrow V_4 = 3V - V_1 - V_2$$

However, due to the assignment of a voltage to V_3, and the algorithm's inability to account for the missing node, the value of V_4 is instead calculated:

$$V = \frac{1}{4}(V_1 + V_2 + V_3 + V_4)$$

$$\Rightarrow V_4 = 4V - V_1 - V_2 - V_3$$

In this way, the value of V_4 and the voltages of the two nodes next to it would be affected by the assigned voltage value of V_3, and this error is propagated in a cone that spreads out from the missing node with a 45° angle (Figure 15(b)), which we dub the "cone of ignorance". In order the algorithm to continue properly, some modification is needed, a topic for future REU work.

4.3.2 Instability

As previously discussed, the algorithm for marching Cauchy data magnifies any error, including round-off error, to the extent that a missing node in the 6th or more row into the network is virtually undetectable with moderate (0.1 percent and larger) noise. Regularization can be applied to this problem with certain constraints. Firstly, too much regularization causes important information to be lost, including the encoded information that would identify the location of the missing node. Secondly, numerical studies have shown that how far in the row which has a missing node is located in the resistor network directly relates to the degree of regularization that should be used. However, since we do not know initially which row contains the missing node, we can only guess at the degree of regularization that should be used, and different degrees can provide different conclusions to the problem. Again, this issue would benefit from further analysis.

5 Using Heat

The use of heat for imaging has drawn significant interest over the past couple of decades. The quasi-steady-state forward and inverse problem, in both the continuous and discrete (network) versions have many similarities to the electrical case. We will focus on the discrete case, and generalize some of the results for electrical networks.

Due to the nature of electricity, there are certain arrangements of missing nodes within a resistor network that cannot be accurately identified by passing an electric current through the network. Consider a rectangular network with three missing nodes as shown in Figure 16(a). By our assumptions, this would imply that all the highlighted resistors (see Figure 16(b)) would have not have any current flowing through them. However, since current cannot be stored within a node or a resistor, no current will flow through resistor A, resulting in node V having zero current flowing to it from any resistor. As such, the network would have the same Cauchy data as though node V was also missing.

Such a problem can be overcome by using heat flow. Since heat can be stored within resistors (in our model), there would have been some heat flowing through resistor A, resulting in a differing Cauchy data that would have distinguished the two networks. In this section, we will first examine the heat model and equations necessary for the formulation of the algorithm, followed by a run-through of the algorithm itself.

5.1 The Heat Equation on a Grid

We begin by modeling the behavior of a "thermal resistor" in the quasi-steady-state (periodic response) case. Consider a thermal resistor that stretches along the x axis from $x = 0$ to $x = 1$, with thermal diffusivity and conductivity taken to be 1 for simplicity. Let the temperature in the resistor at position x, time t be given by $u(x, t)$. We assume u satisfies the usual heat equation $u_t - u_{xx} = 0$ inside the resistor, and that u has boundary and initial condition

$$\begin{aligned} u_t - u_{xx} &= 0, \qquad 0 < x < 1 \\ u(0,t) &= h(t) \\ u_x(0,t) &= -g(t) \\ u(x,0) &= f(x) \end{aligned}$$

where h, g are the boundary conditions and f is the initial temperature. Note that $g(t)$ is the input heat flux at $x = 0$, while $h(t)$ is the temperature.

Now suppose we know that $u(x,t)$ is periodic in t with natural frequency ω. Let $h(t) = He^{i\omega t}$, $g(t) = Ge^{i\omega t}$ where G, H are some (probably complex) constants, and suppose that $u(x,t) = U(x)e^{i\omega t}$, where $U(x)$ may be complex-valued. Then we find that

$$\begin{aligned} i\omega U(x) - U''(x) &= 0, \qquad 0 < x < 1 \\ U(0) &= H \\ U'(0) &= -G \end{aligned}$$

and the initial condition becomes irrelevant. Here G carries the interpretation of input heat flux at $x = 0$ and H is the temperature. The previous problem has now become a second order ordinary differential equation that can be easily solved to find

$$U(x) = H\cosh(\alpha x) - \frac{G}{\alpha}\sinh(\alpha x), \quad \text{where } \alpha = \sqrt{i\omega} \tag{29}$$

5.2 The Algorithm

Using equation (29) above, we can now formulate an algorithm that can be used to locate the missing nodes.

5.2.1 Assumptions

We begin by listing the assumptions that are used by the algorithm.

1. In the application of equation (29) to each resistor (as in Figure 17), the center node is taken to be $x = 1$, while each "outside" node is at $x = 0$ for its respective resistor. This is necessary, as $U_k(1)$ must be the same value for $k = 1, \ldots, 4$.

2. If a node is missing, we consider all the resistors attached to it to be disconnected from the system, i.e. there is no heat flux flowing into them from the neighboring nodes (so they are disconnected at the $x = 0$ end).

3. No heat can be stored within a node, although it can be stored in a resistor.

5.2.2 The Propagation Equation

In order to formulate the algorithm, we must first calculate the propagation equation, similar to that of the Mean Value Property that was used in the electrical version of the problem. With the labeling of Figure 17 we use G_k and H_k, $1 \leq k \leq 4$, to denote the corresponding quantities G and H from equation (29). We use Q to denote the temperature of the center node. Then

$$\sum_{k=1}^{4} U_k(1) = 4Q, \tag{30}$$

$$\sum_{k=1}^{4} U_k'(1) = 0 \tag{31}$$

Equation (30) is derived from the fact that for all $k = 1, \ldots, 4$, $U_k(1) = Q$. Equation (31) is derived from the assumption that no heat can be stored within a node, hence the net heat flux into a node is 0.

Now, if we make use of equations (31) and (29) we obtain

$$\sum_{k=1}^{4} U_k'(1) = \alpha \sinh(\alpha) \sum_{k=1}^{4} H_k - \cosh(\alpha) \sum_{k=1}^{4} G_k = 0$$

which yields, after a bit of rearrangement,

$$\sum_{k=1}^{4} G_k = \frac{\alpha \sinh(\alpha)}{\cosh(\alpha)} \sum_{k=1}^{4} H_k. \tag{32}$$

If we make use of equations (30) and (29) we obtain

$$\sum_{k=1}^{4} U_k(1) = \cosh(\alpha) \sum_{k=1}^{4} H_k - \frac{\sinh(\alpha)}{\alpha} \sum_{k=1}^{4} G_k = 4Q \tag{33}$$

Use equation (32) to substitute out the sum of the G_k in (33) and find

$$\cosh(\alpha) \sum_{k=1}^{4} H_k - \left(\frac{\sinh(\alpha)}{\alpha}\right) \left(\frac{\alpha \sinh(\alpha)}{\cosh(\alpha)} \sum_{k=1}^{4} H_k\right) = 4Q$$

which becomes, after some simplification,

$$\left[\frac{\cosh^2(\alpha) - \sinh^2(\alpha)}{\cosh(\alpha)}\right] \sum_{k=1}^{4} H_k = 4Q$$

By the property $\cosh^2(\alpha) - \sinh^2(\alpha) = 1$, we obtain the propagation equation:

$$Q = \frac{1}{4\cosh(\alpha)}(H_1 + H_2 + H_3 + H_4) \tag{34}$$

which relates the temperature at the central node to those of the surrounding nodes. Note the case $\alpha = 0$ (equivalent to $\omega = 0$) corresponds to the usual steady-state case, that is, the electrical conduction case.

As in the steady-state case, equation (34) can be used to continue Cauchy data into the network, and a missing node is identified by noting that a certain resistor has zero flux. The details are given below.

5.2.3 Indicative Heat Flux

For a node with three neighboring nodes attached to it via resistors (as in Figure 18), computations entirely analogous to those above show that Q should be calculated as

$$Q = \frac{1}{3\cosh(\alpha)}(H_1 + H_2 + H_3) \tag{35}$$

Note that equation (35) is equivalent to equation (34) if and only if $H_4 = Q\cosh(\alpha)$.

Substituting this value of H_4 into equation (29) and its derivative results in the following:

$$U_4(1) = Q\cosh^2(\alpha) - \frac{G_4}{\alpha}\sinh(\alpha) = Q \tag{36}$$

$$U_4'(1) = Q\alpha\cosh(\alpha)\sinh(\alpha) - G_4\cosh(\alpha) \tag{37}$$

From (36) and (37), we see that

$$\begin{aligned}\frac{G_4}{\alpha}\sinh(\alpha) &= Q\left(\cosh^2(\alpha) - 1\right)\\ &= Q\sinh^2(\alpha)\end{aligned}$$

From this we have

$$\begin{aligned}U_4'(1) &= Q\alpha\cosh(\alpha)\sinh(\alpha) - \frac{Q\alpha\sinh^2(\alpha)}{\sinh(\alpha)}\cosh(\alpha)\\ &= Q\alpha\cosh(\alpha)\sinh(\alpha) - Q\alpha\sinh(\alpha)\cosh(\alpha)\\ &= 0 \qquad (38)\end{aligned}$$

Equation (38) shows that a missing node is indicated by the fact that there is no heat flux flowing in the resistor between the missing node and entering the node directly below it. This fact is used to help us identify the location of the missing node, using exactly the same procedure as in the steady-state case.

5.2.4 Limitations

The limitations of the heat algorithm are identical to that of the electrical algorithm. Since the algorithms essentially work in the same way, the "Cone of Ignorance" is still in effect, resulting in any missing node within that cone to be undetectable.

The heat algorithm is also very unstable, even more so than that of the electrical algorithm, in the case that the driving frequency ω is very large. This is due to the eigenvalues of the heat propagation matrix. The propagation matrix M for the Cauchy data is similar to the steady-state case, namely

$$\mathbf{M} = \begin{bmatrix}\mathbf{A} & -\mathbf{I}\\ \mathbf{I} & 0\end{bmatrix}$$

but where the matrix $\mathbf{A}$ is (in, for example, the 4 × 4 case)

$$\mathbf{A} = \begin{bmatrix} 4\cosh(\alpha) - \frac{1}{\cosh(\alpha)} & -1 & 0 & 0 \\ -1 & 4\cosh(\alpha) & -1 & 0 \\ 0 & -1 & 4\cosh(\alpha) & -1 \\ 0 & 0 & -1 & 4\cosh(\alpha) - \frac{1}{\cosh(\alpha)} \end{bmatrix}$$

Remember that $\alpha = \sqrt{i\omega}$. As $\omega \to \infty$, the complex quantity $4\cosh(\alpha)$ has magnitude approximately $2e^{\sqrt{\omega/2}}$. The matrix $\mathbf{A}$ becomes effectively diagonal, with eigenvalues of magnitudes near $2e^{\sqrt{\omega/2}}$. The eigenvalues for $\mathbf{M}$ are, as in the steady-state case, given as roots of $\mu + \frac{1}{\mu} = \lambda$ where λ is an eigenvalue for $\mathbf{A}$. As such, some of the μ will also become extremely large as ω grows. Indeed, in the steady-state case ($\omega = 0$) the eigenvalues for $\mathbf{M}$ were bounded above in magnitude by 6, but here there is no such bound. Hence, any error will be magnified far more greatly by the heat propagation matrix as compared to the electric propagation matrix, causing the heat algorithm to be much more instable than the electric propagation matrix.

6 Conclusion

In our research, we have found that the algorithms formulated to solve for the varying backside case and for the missing nodes case work well under perfect conditions. Unfortunately, seeing as it is impossible for us to obtain data that is without any error, the algorithms in their most naive form fail due to their instability. Efforts to counter this instability have succeeded only marginally, and more work can be done in trying to overcome this limitations of the algorithm.

The work discussed in this paper can be extended to the continuous case. An example of a real-world continuous case would be that of a blast furnace, where the hot molten ore within the blast furnace is capable of corroding into the wall of the blast furnace. Workers would need to know when the wall has been corroded so thin that it is in danger of cracking completely, and exposing them to danger. Another example would be that of a solid object that might have had air bubbles introduced during the manufacturing procedure, creating a crack or a void within it.

The varying backside case is the discrete approximation to the first example, where the resistor grid represents the cross-sectional area of the wall, and the varying backside is caused by the corrosion of the hot molten ore. The missing nodes case would be the discrete approximation to the second example, in which the resistor grid is the cross-sectional area of the object where the crack is located, with the missing node representing the area where the crack is.

Numerical tests have been run on a 30 × 10 grid for both cases, and results came out to be accurate up to 85%. More work can be done in this area, as well as proving that the discrete model used in this research can indeed be used to approximate the continuous model.

Further research questions for future REU groups include the following:

1. Looking at how corrosion increases resistance, rather than simple removing nodes completely or causing the nodes to have $0V$.

2. How results change with increasing node resolution, i.e. working with the same depth but increasing the number of nodes there are within that depth.

3. How results can be improved in the varying backside case using imaging from the sides of the boundary as well as from the bottom.

4. Overcoming the limitation of the "Cone of Ignorance" in the algorithm for the missing nodes case, so that multiple missing node can be found.

There are many other extensions and variations of this problem that can be undertaken in further research, and the above are simply several examples that can be used. We feel that this problem can have many real-world applications, and further research should be done to tackle these problems.

References

[1] K. Bryan and L. Caudill, “Stability and Reconstruction for an Inverse Problem for the Heat Equation,” *Inverse Problems*, (14), 1998, pp. 1429-1453.

[2] E. Curtis and J. Morrow, “Inverse Problems for Electrical Networks”, World Scientific Publishing Company, 2000.

[3] T. Fredman, Boundary Identification in the Domain of a Parabolic PDE, Ph.D. Thesis, Abo Akademi, Turku Finland, 2008.

[4] R. Horn and C. Johnson, “Matrix Analysis”, Cambridge University Press, 1985.

[5] M. Cheney and D. Isaacson, “Issues in Impedance Imaging,” *IEEE Computational Science and Engineering* Volume 2, Issue 4, 1995, pp. 53-62.

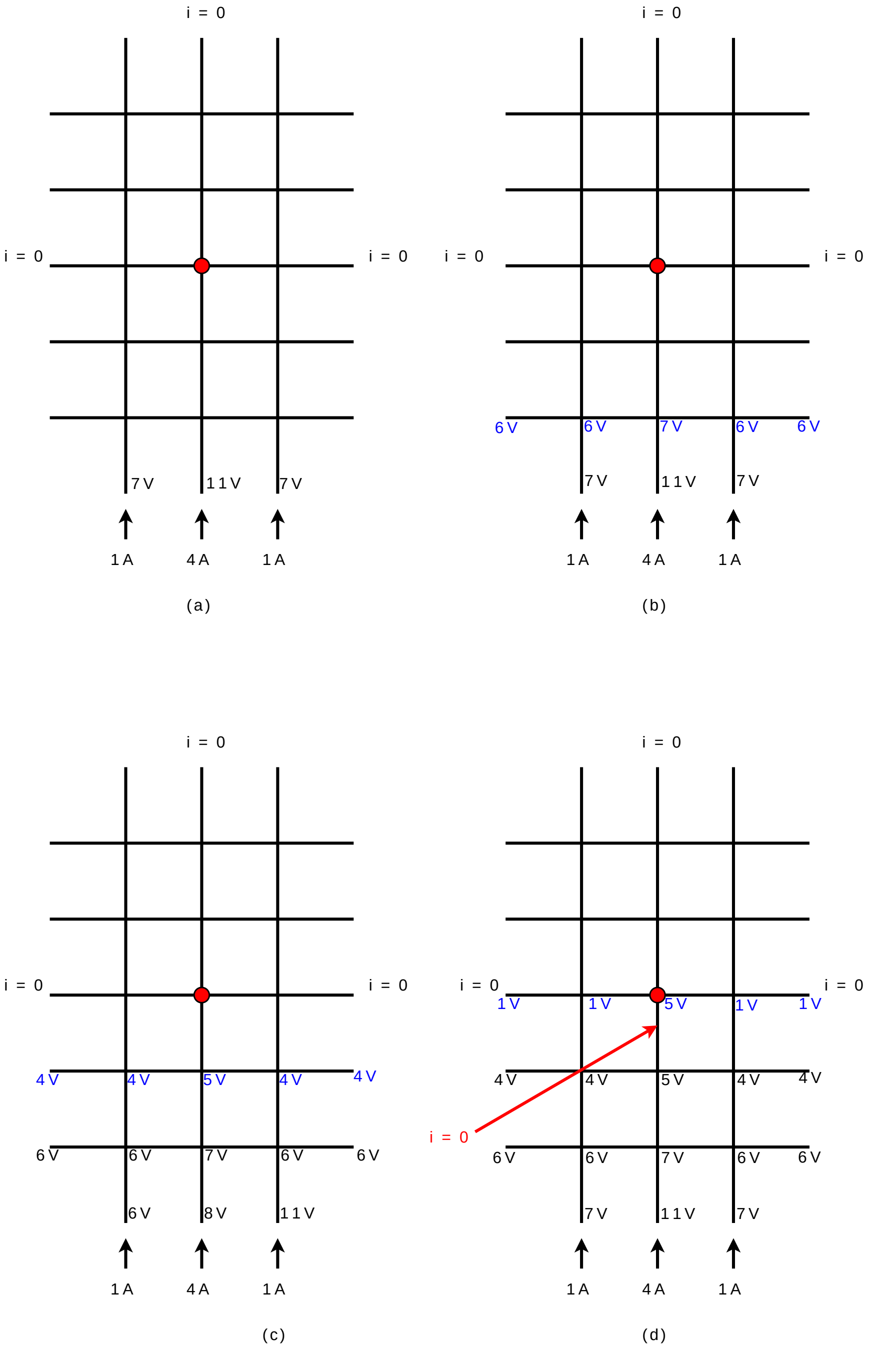

Figure 14: Example of Algorithm

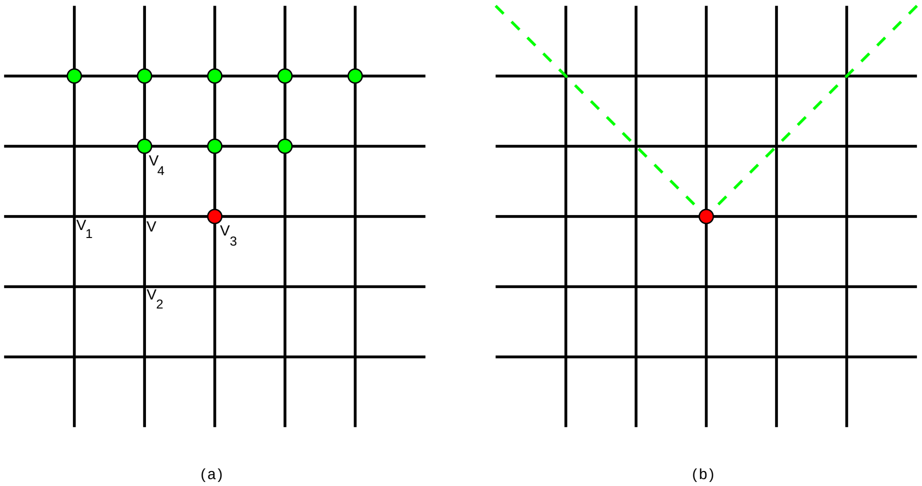

Figure 15: Cone of Ignorance

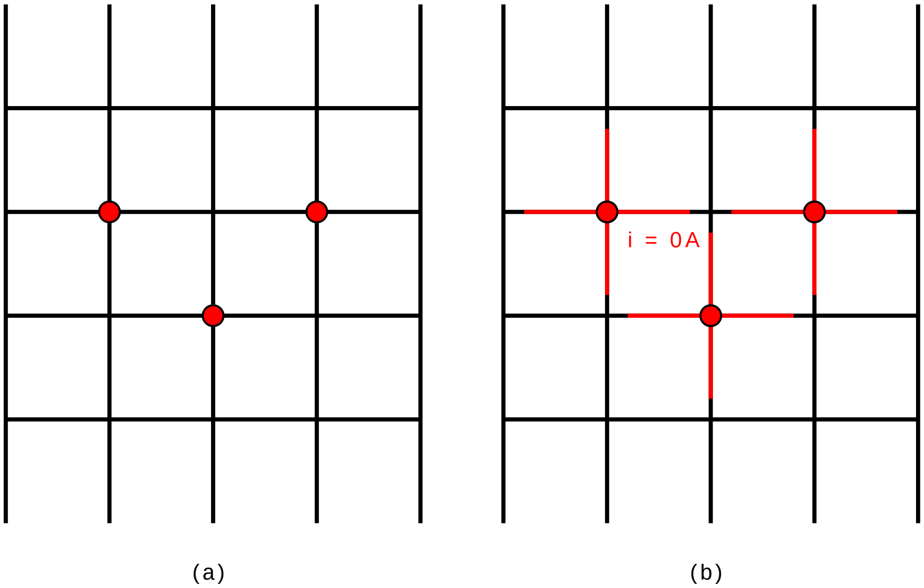

Figure 16: Node Arrangement Undetectable by Electricity

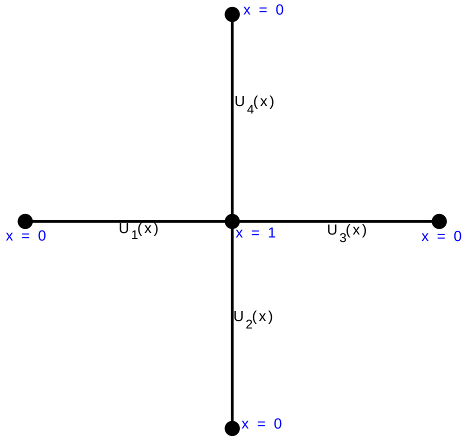

Figure 17: Node representation for Heat

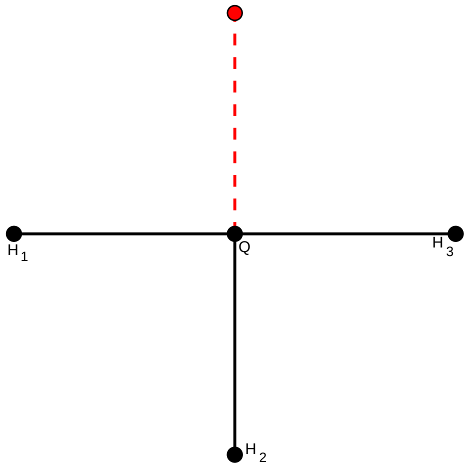

Figure 18: System with one missing node (in red)