Determing the Conductivities of a Continuous Network By Discrete Network Approximations

Peter L. Staab
University of Utah

Stefan Treatman
M. I. T.

June 21, 2003

Abstract

In this paper, we discuss an algorithm for recovering conductivities of continuous materials by modelling with discrete networks. This procedure uses the Neumann to Dirichlet map of discrete networks and leads to our conjecture that there is a calculable link between discrete and continuous networks.

1 Introduction

Our objective is to use resistor networks to model continuous regions. We will restrict our discussion to square resistor networks, as in Curtis and Morrow[1]. A square network Ω_n consists of n^2 interior nodes, $4n$ boundary nodes, and $2n(n+1)$ resistors connecting the nodes such that each boundary node is connected to exactly one node. Ω_5 is shown in figure 1.

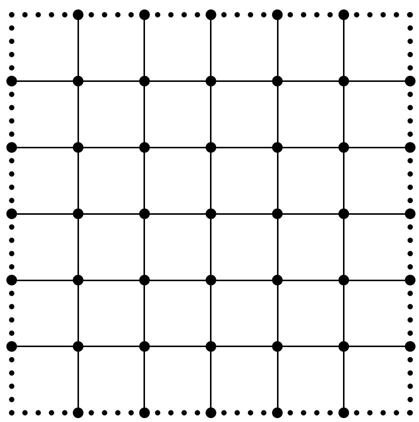

Figure 1

1.1 The Forward Neumann Problem and The Neumann to Dirichlet Map

The forward Neumann Problem is similar to the forward Dirichlet problem as briefly explained in [3]. The objective of the forward problem is to produce N: the Neumann to Dirichlet Map for an $n \times n$ network of resistors from $\Gamma = (\Omega_0, \Omega_1, \gamma)$ which is a network $\Omega = (\Omega_0, \Omega_1)$ and a function $\gamma : \Omega_1 \to R^+$ as explained in [3]. For each edge $\sigma = pq$ in Ω_1, the number $\gamma(\omega)$ is called the *conductance* of σ and $1/\gamma(\sigma)$ is the *resistance* of σ. The function γ on Ω_1 is called the *conductivity*. The means to get from Γ to N is by using Kirchoff's law (1) and Ohm's law (2).

$$\sum_{Q \sim P} \gamma(PQ)(u(P) - u(Q)) = 0 \tag{1}$$

$$I(PQ) = \gamma(PQ)(u(P) - u(Q)) \tag{2}$$

where $u(P)$ is the voltage at node P in Ω_0, $I(PQ)$ is the current passing through segment PQ, and $Q \sim P$ means that Q is to vary around P's neighbors, as in [3].

By setting up a Kirchoff's law equation for each interior node and an Ohm's law equation for each exterior node, a matrix, A, can be set up and solved for each set $\{u(P)\}$ for P, an interior node. But the matrix A is singular as explained in [1], therefore to elimate the problem, we remove one equation in A and arbitrarily set one voltage to zero. To solve the A matrix equation,

$$A\mathbf{u} = \mathbf{b}$$

the set $\{u(P)\}$ for P an exterior node is used for $\mathbf{b}$. This depends on the current on the boundary, $\partial\Omega$. The map N takes currents on $\partial\Omega$ and gives voltages on $\partial\Omega$. To produce a comprehendable N map and remembering that the total current in must equal the total current out, we choose as our basis elements the set $\{\phi_j\}, j = 1, \ldots, 4n$ where ϕ_j represents 1 amp entering at boundary node j and 1 amp coming out at boundary node $j+1$ with no current in or out the other boundary nodes. It is conveinent to think of outgoing currents as negative currents.

To create the N matrix, we used ϕ_j for the boundary current and solve the A matrix equation as explained above. Each ϕ_j will give the j^{th} column of the N matrix. Because the Neumann to Dirichlet map is not unique, (i.e. an arbitrary constant voltage will give zero current throughout the network), we use voltage differences as opposed to absolute voltages.

We label the boundary nodes in a clockwise direction, starting in the left-most node of the north side (figure 2). N will be represented as a $4n$ by $4n$ matrix where $N_{i,j}$ is the potential difference between boundary node i and $i+1$ due to ϕ_j. (For clarification, $N_{4n,j}$ is the potential difference between boundary nodes $4n$ and 1 due to ϕ_j).

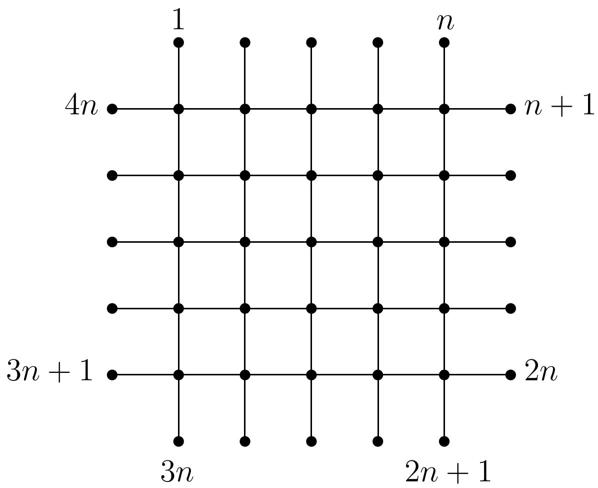

Figure 2

1.2 The Continuous Problem

The problem we wish to solve is the following. Four square pieces of conductive material are placed adjacent to one another to form a larger square as shown in figure 3, so that current is free to flow between them. We wish to recover the conductivity of each region solely from boundary information.

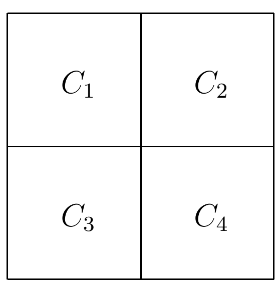

Figure 3

We would like to model this system of four regions with a network of resistors. For any n, we can superimpose Ω_n on the four regions and set the value of the resistivity of each resistor to be that of the quadrant it falls upon. Our first attempt to model the system with a discrete network would be the 2×2 network. We assign all the resistors in each quadrant the same resistivity. This is shown in figure 4a. For the 4×4 model, the crosshairs extend as shown in figure 4b.

As n increases, this "crosshair" model produces closer approximations to the continuous case. We will need a way to take boundary data on an arbitrary $n \times n$ network and recover the resistors within the network. This is the inverse Neumann problem. To make our computations comparable with one another, we also seek an effective way to reduce every size network to a 2×2 network.

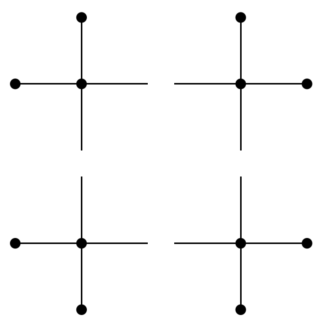

Figure 4a

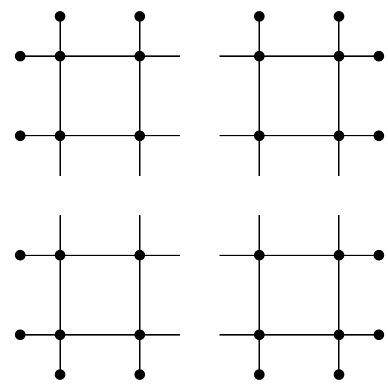

Figure 4b

2 The Inverse Neumann Problem

The inverse Neumann problem consists of calculating the values of the conductors in the network from the boundary data map N. The inverse Neumann problem is similar to the inverse Dirichlet problem as described in Curtis and Morrow[2]. Because the continuous case being modeled is for a group of only four continuous regions, the inverse problem described in this paper is for a 2×2 network, but may be extended further for larger systems.

To begin with, the corner resistors are calculated by placing zero current everywhere except the i^{th} and $(i+1)^{th}$ nodes which must be around a corner. A current of 1 is sent in at the i^{th} node and -1 at the $(i+1)^{th}$ node. A voltage of 0 on the face opposite the $(i+1)^{th}$ node is set as shown in figure 5 for $i = 2$. For clarification, currents are shown in parentheses, while voltages are not.

By harmonic continuation as in [2], a zero voltage occurs at the interior node that connnects both the i^{th} and $(i+1)^{th}$ node. To calculate the two resistors the values of the voltage are needed at the i^{th} and $(i+1)^{th}$ boundary nodes. This is obtained directly from the N map. For the $i = 2$ case, the voltages for the i^{th} and $(i+1)^{th}$ nodes are read from the 2^{nd} columns of the N matrix. $V_2 = -N_{1,2}$ and $V_3 = N_{3,2}$.

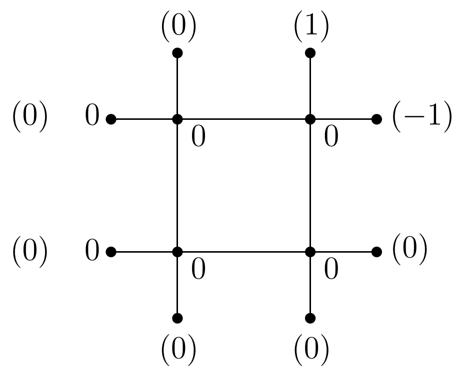

Figure 5

The other three corners are found in a similar fashion.

To calculate the four interior resistors, an important corner relationship (3) and other similar properties (for the other corners) of the N matrix are used.

$$N_{2,j} + \beta N_{2,j} = 0 \text{for} j = 5, 7 \tag{3}$$

(This is similar to the corner relationships of the Λ matrix for the Dirichlet to Neumann map in Curtis and Morrow[1], but differs slightly because of the underlying basis of the matrix. The ϕ_j for N is (0,0,...,0,1,-1,0,0,...,0) whereas the basis for Λ is (0,0,...,0,1,0,0,...,0).)

After calculating the β from equation (3), the voltage V (see figure 6) at the non zero interior node can be found along with the values of the two interior conductors touching that node. A simple rotation finds the remaining two conductor values.

Overall, given the current to voltage boundary information for any 2×2 resistor network, Ω_2, the inverse Neumann problem solves for the twelve resistors (eight exterior and four interior).

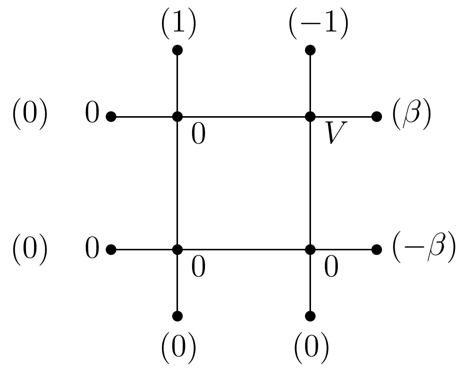

Figure 6

3 The Condensation Process

Since we wish to model the continuous case with a 2 by 2 network, we require a process to reduce the $4n$ by $4n$ N matrix of an arbitrary n by n resistor network to an 8 by 8 matrix. This process is called condensation. We produce the desired 8 by 8 matrix through the following steps.

The n by n network to be condensed has four faces, with each face divided into halves. Label the half-faces from 1 to 8 in a counter-clockwise direction as shown in figure 7.

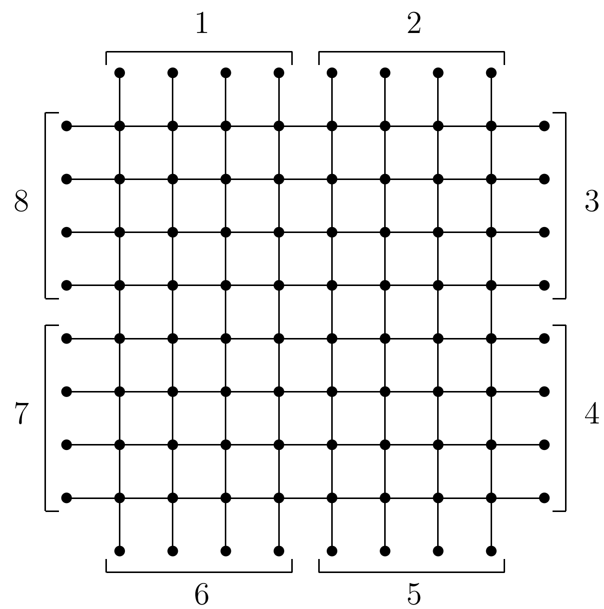

Figure 7

Let Ψ_j represent a current of 1 evenly distributed over the boundary nodes in half-face j and -1 evenly distributed over the nodes of half-face $j+1$ with zero currents everywhere else. (Again, if $j = 8$, then $j+1$ wraps around to half-side 1).

Figure 8 shows current flow due to Ψ_4 on an 8 by 8 network with zero current on the north and west faces.

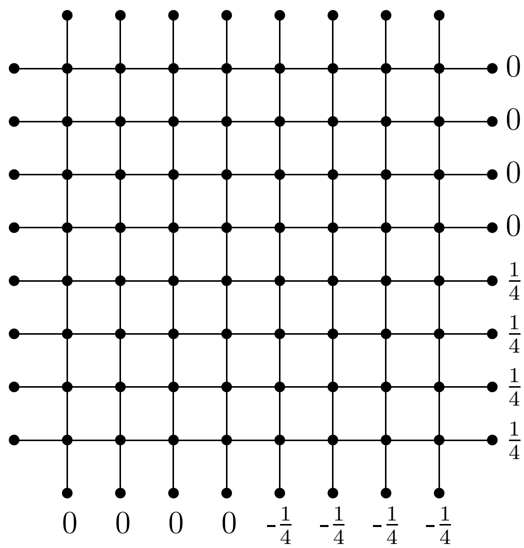

Figure 8

Given the distribution of currents according to Ψ_j, and the original $n \times n$ N matrix which gives potential differences across all of the nodes of the boundary, we solve for the 64 entries of the condensed matrix, $\tilde{N}$ by defining $\tilde{N}_{i,j}$ to be the average potential difference between half-side i and half-face $i+1$ due to Ψ_j. (Again, half-face 8 is adjacent to 1).

Because the condensation from arbitray $n \times n$ matrices to an 8×8 matrix detroys many of the properties of the N matrix, differing algorithms for the inverse problem will produce differing conductor values. After testing different algorithms, we found that no one algorithm was better than another at reproducing all of the conductors. We used a combination of algorithms to get a "best" value for the conductors.

3.1 C: The Conductance Map from R^4 to R^{12}

With the algoritms for producing the forward Neumann problem, the condensation process and the inverse problem, we can fuse the three to produce another map. The forward problem takes the values of the resistors of an $n \times n$ network to prduce an $4n \times 4n$ N matrix. Instead of knowing the value of each of the $2n(n+1)$ resistors, we take the values of each quadrant as explained in the introduction to produce the N matrix. The $4n \times 4n$ matrix produced is used to form a new, condensed 8×8

matrix as explained above. The last step takes this 8×8 matrix and produces 12 values, one for each resistor in the $n = 2$ case.

This map C can be throught of as a map from the original 4 quadrant values, or a vector in R^4 to the reconstructed 12 resistor values or a vector in R^{12}. Originally, we hoped to discover properties about C, but with a Newton method convergence algorithm used instead, we were able to discover plenty without such knowledge of the actually map.

3.2 Stretch Factor

Starting with a 2×2 network in which the conductivitry of each crosshair is 1 (see figure 4a), we produce an 8×8 N matrix. We perform the previously described function C to retrieve the values of each of the twelve resistors. Because of the symmetry of the starting vector (1,1,1,1), all boundary reistors are found to be equal. We repeat this process for all even values of n. In each case, the four conductor values returned are all equal. We take the ratio of the returned value for the $n \times n$ case to that of the 2×2 case. As n increase, we produce a sequence of ratios which is known to converge since it is achieved through a finite-differences approximation to the forward Neumann problem. This limiting ratio will be used later and will be referred to as the "stretch factor". The reason the returned values for the conductivities in each quadrant increases with n is that the finer the grid becomes, the easier it is for the current to pass across the corner, where the two sides come very close together.

Because more current flows through this region, the inverse process returns higher values for the conductivities. It interprets the increase in current to be due to more conductive regions. In other words, it "thinks" the region has a higher conductivity than it really does. We make an approximate correction later by dividing corner conductivities by the stretch factor.

4 Solving the Continuous Inverse Problem Using Discrete Methods

Given four continuous materials arranged as shown previously and each quadrant with a constant conductivity, we seek a method to recover the value of the four conductors,

(c_1, c_2, c_3, c_4). By evenly distrbuting a current of 1 over a half-face and taking out a current of 1 over the clockwise-adjacent half-face, insulating all other half-faces, one can determine the integrated potentials over each half-face. The N matrix for this system is thus produced as follows. ψ_j represents an average current of 1 over half-side j and an average current of -1 over its clockwise-adjacent half-face. (There are eight half-faces). $N_{i,j}$ is then the difference between the average integrated potential over half-face i and that of half-face $i+1$ due to ψ_j. This creates an 8×8 N matrix for a <u>continuous</u> system. Our earlier inverse solver takes 8×8 matrix and produces twelve numbers—one for each resistor value— mong those twelve are four values we use as approximations for the conductivities of each quadrant. These values are incorrect, but we divide through by the stretch factor and produce four values for the c_i's that we hope are close to the actual values. These new c_i's will act as our initial guess in the Levenberg-Marquardt algorithm.

4.1 The Levenberg-Marquardt Algorithm

We use the Levenberg-Marquardt algorithm, a Newton method-type algorithm, to find the solution to the function, C. Using the conductance map C from R^4 to R^{12}, we want to be able to recover the four conductors (c_1, c_2, c_3, c_4) from a guess at the four conductors. The guess is made from the 8×8 N matrix from continuous data and the stretch factor calculated. The subroutine SNLSE in CMLIB performs iterations of the algorithm which takes the initial guess and gradually alters them until they are within some specified epsilon of the actual four conductivities.

4.2 Experimental Results

To demonstrate the process in action, we began by setting all conductors in the northwest quadrant of a 10×10 network to 1000, the northeast quadrant to 1, the southwest corner to 1.1 and the southeast quadrant to 0.001. These should be taken as the <u>true</u> values of (c_1, c_2, c_3, c_4). We produced a 40×40 N matrix by the forward Neumann map. This matrix was condensed to an 8×8 matrix as prescribed and then run through the inverse solver. Using 2.91 as our stretch factor, we divided the corner conductivites by 2.91 to prodcue our initial guess for (c_1, c_2, c_3, c_4). Our guess turned out to be (1493.14,1.12255,1.23490,7.51977e-04). After several iterations, the subroutine returned our "best" guess for the conductivites to be (1000.000, 1.00000,1.10000,1.00000e-03). This is consistant to the real values to 6 significant figures.

5 Conclusion

The claim that the Levenberg-Marquardt algorithm as used by SNLSE can recover the conductivities of continuous materials is based on the limiting arguement of the finite-differences approximation and numerous test cases in which we used data from a 10×10 network as data from a contnuous system. Our stretch factor was simply the ratio of the 10×10 network that was involved in the sequence that was used to produce the original stretch factor. The process resulted in the recovery of the actual four conductivites in every case we attempted to run.

References

[1] E. B. Curtis and J. A. Morrow, *The Inverse Conductivity Problem.*

[2] E. B. Curtis and J. A. Morrow, *The Dirichlet to Neumann Map for a Resistor Network.*

[3] E. B. Curtis and J. A. Morrow, *Determining the Resistors in a Network*