Applications of Iterative Schemes: The Dirichlet Problem and the Search for Current Sources in Electrical Networks

Nancy L. McNally

August 16,1991

Abstract

I discuss the discrete form of the Dirichlet problem in two dimensions and consider three iterative methods to solve the system of equations arising from this problem. Then I discuss the use of the most efficient of these iterative schemes in an attempt to locate current sources of unit strength in a square network of resistors. Finally, I discuss some shortcomings associated with my method of solving this last problem.

I. The Dirichlet Problem

Consider a square network of resistors in the plane. The network is made up of resistors joined together at nodes to form a lattice. Each resistor has conductance γ, where conductance is defined to be the reciprocal of resistance. The typical network under discussion in this paper has m boundary nodes on a side and m^2 interior nodes. I also difine rim nodes as those interior nodes having at least one neighbor which is a boundary node. The typical square network I am considering has 4m–4 rim nodes. Also, for convenience, all conductances are taken to be 1 throughout this paper. A typical network is shown in figure 1.

The discrete Dirichlet problem for this network (discussed simply in Snell and Doyle (1)) consists of finding a potential function u = u(x,y) which satisfies the discrete form of Laplace's equation (to be discussed below) at each node in the network with the restriction that u(x,y) = ϕ(x,y) on the boundary of the network; ϕ(x,y) is a prescribed vector of boundary potentials. In the continuous case, solutions to the Dirichlet problem satisfy Laplace's equation:

$$u_{xx} + u_{yy} = 0.$$

When this equation is modified to be used in discrete (electrical network) problems such as those in Curtis and Morrow (2), it takes the form:

$$[\sum_{q \in N(p)} \gamma(pq)]u(p) = \sum_{q \in N(p)} \gamma(pq)u(q)$$

where u(p) is the potential at node p, both sums are taken over the nodes q which are neighbors of p, and γ(pq) is the conductance of the resistor joining node p to its neighbor q.

For a given square network with m boundary nodes on a side, we get m^2 equations of the form (2), one equation per interior node. The unknowns in these equations are the potentials at each interior node p due to a specified set of boundary potentials ϕ. The solution of this system of linear equations is the solution to the discrete version of the Dirichlet problem. This system of equations may be written in matrix form:

$$Au = b,$$

where u is a vector of interior potentials, b is a vector determined by ϕ and the conductances, and A is what I will call the Kirchoff matrix since each row of it comes from an application of (2), which is a statement of Kirchoff's Law.

II. Iterative Methods for Solving Systems of Linear Equations

The iterative schemes discussed below were studied as an alternative to using linear algebra FORTRAN subroutines to solve the linear system Au = b. These schemes have little if any advantage over the FORTRAN subroutines when the system to be solved is small. However, for very large systems, these methods prove to be more efficient.

A. The Jacobi Iterative Method

In this iterative method (as in the other methods to follow), an initial guess u_0 is given. Once the initial guess is given, the Jacobi method computes the first approximation u_1 to the true solution u using only the values u_0. Once the vector u_1 is computed, the second approximation u_2 is computed using only the values in u_1. This process is repeated until $\parallel u_k - u \parallel < \epsilon$ where ϵ is a pre-determined (small) number.

Now, let me be more specific. The Kirchoff matrix A may be written as the sum of a diagonal matrix D and a non-diagonal matrix N (i.e., the diagonal entries of N are all 0 and the off-diagonal entries of N are the off-diagonal entries of A):

$$A = D + N$$

Then the Jacobi iteration is given in matrix form by:

$$u_{k+1} = D^{-1}(b - Nu_k)$$

According to G. D. Smith (3), since the Kirchoff matrix is diagonally dominant, the u_k produced by the Jacobi method will converge to the true solution u.

The Jacobi method has two major drawbacks that can be overcome using another iterative scheme called Gauss–Seidel. The first drawback is that all of the entries of u computed in the kth iteration must be saved in order to compute the (k + 1)st iteration. The second drawback is that the Jacobi method converges to the true solution u very slowly. Again, both these problems can be overcome with the next iterative scheme: Gauss–Seidel.

B. The Gauss-Seidel Iterative Scheme

In this iterative method, the (k+1)st values in the vector u_{k+1} are used as soon as they are computed. This both speeds up the convergence of the iterations to the true solution u and makes it unnecssary to store any of the values computed on the kth iteration. For example, consider the 4 x 4 system of equations below, where the (k+1)st iteration is being computed:

$$\begin{aligned} {u_1}^{k+1} &= a_{11}^{-1}(b_1 - a_{12}{u_2}^k - a_{13}{u_3}^k - a_{14}{u_4}^k) \\ {u_2}^{k+1} &= a_{22}^{-1}(b_2 - a_{21}{u_1}^{k+1} - a_{23}{u_3}^k - a_{24}{u_4}^k) \end{aligned}$$

$$u_3{}^{k+1} = a_{33}^{-1}(b_3 - a_{31}u_1{}^{k+1} - a_{32}u_2{}^{k+1} - a_{34}u_4{}^{k})$$
$$u_4{}^{k+1} = a_{44}^{-1}(b_4 - a_{41}u_1{}^{k+1} - a_{42}u_2{}^{k+1} - a_{43}u_3{}^{k+1})$$

Notice that the equation for the (k+1)st value of u_2 uses the just computed (k+1)st value of u_1 but uses the kth values of u_3 and u_4 since the (k+1)st for these variables have not been re–computed yet.

The Gauss–Seidel iteration method may be expressed in the general case of m equations as follows:

$$u_i^{k+1} = a_{ii}^{-1}(b_i - \sum_{j=1}^{i-1} a_{ij}u_j^{k+1} - \sum_{j=i+1}^{m} a_{ij}u_j^{k})$$

Since the Kirchoff matrix A is symmetric and positive–definite, by theorem 2.4, Johnson and Riess (4), the Gauss–Seidel iterative scheme will converge to the true solution u.

C. Gauss–Seidel with Successive Over–Relaxation

While the Gauss–Seidel method converges fairly rapidly even for large systems, it is possible to speed up the convergence (and therefore the computing time) of this method by introducing what is called a relaxaion parameter, ω.

Before ω is defined, let me first write an typical equation from the system (3) in the form of equation (2) as follows:

$$4u_i^{k+1} = 1u_{i-1}^{k+1} + 1u_{i+1}^{k} + 1u_{i+m}^{k} + 1u_{i-m}^{k+1}$$

where the coefficients of the $u's$ are specific to the case I am considering, namely all conductances are equal to 1. It should be noted that the above equation is valid for all interior nodes except rim nodes; for rim nodes the equation is slightly modified since some of the terms are actually known (boundary) values. This equation can be re–written:

$$u_i^{k+1} = u_i^{k} + [(u_{i-1}^{k+1} + u_{i+1}^{k} + u_{i+m}^{k} + u_{i-m}^{k+1} - 4u_i^{k})(.25)]$$

The expression in brackets can be thought of as a correction factor in going from the kth iteration to the (k+1)st iteration. The idea behind SOR is to multiply this correction factor by a constant ω (the relaxation parameter) in order to speed up the convergence of the iterations to the true solution. I make no attempt here to describe more details of how to choose ω but I give the expression I used for ω in my FORTRAN programs; this expression was found in Johnson and Riess (4):

$$\omega = \frac{2}{1 + sin(\pi/n)}$$

where n is the number of eqations in the Kirchoff matrix. The rates of convergence for these three iterative schemes are shown in figures 2 through 4.

III. Finding Current Sources: The Forward Problem

Once I was able to solve the system (3) using Gauss–Seidel with SOR , I began to use it to determine the potential at each rim node due to a given configuration of current sources of unit strength when all the boundary nodes were grounded (i.e., $\phi \equiv 0$). These configurations had several resrictions on them. No configuration of sources included any boundary nodes or rim nodes, all configuraions had nine or fewer current sources, all configurations were x–convex, and again, the magnitude of each current source was identically 1. A region is x–convex if along any row of the network there are no inner nodes which are not sources between two nodes which are current sources. Figure 5 shows a typical configuration.

The Dirichlet problem associated with this problem has two basic differences from the Dirichlet problem discussed in section I. A minor and obvious difference is that all the entries of the vector b in the system $Au = b$ are 0 because $\phi \equiv 0$. The second difference is in the form of the equations in the linear system (3) which correspond to inner nodes at which there is a current source. Instead of being of the form of equation (2), the discrete form of Laplace's equation, these equations are of the form:

$$[\sum_{q \in N(p)} \gamma(pq)]u(p) = 1 + \sum_{q \in N(p)} \gamma(pq)u(q)$$

where the added term of 1 on the right hand side accounts for the current sources of unit strength. This equation is a discrete form of Poisson's equation:

$$u_{xx} + u_{yy} = h(x, y)$$

In the case under discussion, $h(x, y) \equiv 1$. The equations in the system (3) corresponding to the inner nodes with no current sources are still of the form of equation (2), as they are in the discrete Dirichlet problem discussed in section I.

Again, the Gauss–Seidel method with SOR was used to solve the linear system arising from this problem because it was exceptionally fast and because it is easy to program.

IV. Finding Current Sources: The Inverse Problem

The forward problem discussed in section III is simply to find the rim potentials which arise from a given configuration of unit strength current sources. The inverse problem in this case is that of finding the locations of the current sources inside the network, given only a vector of rim potentials due to those sources. Of course, along with the given rim potentials, I also assume that the region of sources to be found is "valid" ,i.e., has all the restrictions described in section III. I now discuss the first halting steps in the search for an algorithm to solve this problem. The basic idea offered here was suggested to me by Professor Curtis.

Starting with a vector of rim potentials which comes from a valid region of current sources, I first find the total current flowing out of the network due to this region. This total current is just the sum of the currents exiting the network at each boundary node. These currents are found using Ohm's Law, ΔV= IR. Since for each boundary resistor I know the boundary and rim potentials and the conductance ($1 \equiv \gamma = 1/R$), the current flowing out of the network at each boundary node is just

$$\Delta V = rimpotential - boundarypotential.$$

But since each boundary potential is identically 0, I have that the current at each boundary node is just the value of the rim potential adjacent to that node. Since all the current sources are of unit strength, once I know the total current leaving the network due to a configuration of sources, I immediately know the number of current sources to be located.

Once I know the number of current sources, I begin essentially to make guesses about the location and configuration of these sources. I start by guessing a set of sources (always with the restrictions outlined in section III) at the center of the network and comparing the rim potentials due to the guessed region with the actual given rim potentials. Specifically, I compute a vector *error*; each entry in *error* is the difference between the rim potential due to my guessed region of sources and the rim potential due to the actual region of sources:

$$error(i) = guessedrimpotential(i) - actualrimpotential(i)$$

where the index i refers to the ith rim node.

I make my next guess/approximation to the true region of sources by considering the locations on the rim of the network of the maximum and minimum values of the vector *error*. To make this discussion concrete, consider figure 6.

Suppose the region shown in figure 6 is one of my guesses and that the maximum value of *error* occurs at rim node i while the minimum value of *error* occurs at rim node j, as shown. Since the rim potential at location i is too large and the rim potential at location j is too small, my next guess would be to move the source at the bottom right corner of the region (the source nearest rim node i) to the first position above rim node j which will make the region x–convex. This position is circled in figure 6 (only the 6 nodes marked with an x are part of the guessed region; the circled node will be part of the next guess). So the region of my next guess would be that shown in figure 7. Then I examine the *error* vector due to this guess and repeat the guessing procedure (moving only one source per guess) until the entries in $error \approx 0$ (they never equal to 0 because of round–off error).

V. Shortcomings

In this section I discuss several major drawbacks: one concerning the formulation of the inverse problem just described and some others concerning the "algorithm" for solving it.

The first shortcoming is by far the most important, and this is that I offer no theoretical proof that each valid region of sources produces a different vector of rim potentials. I am fully aware that unless this is proved, the inverse problem as I have stated it cannot be shown to have a unique solution.

Now, the algorithm described in section IV for solving the inverse problem (under the assumption that it has a unique solution) has several problems. The first of these is that it sometimes occurred that the algorithm, if strictly applied, lead me to keep moving a source back and forth between the same two locations. That is, the error vector from one guess would lead me to move a source to a new location, but the error vector with the source in the new location would lead me to move the source back to its previous position. Professors Curtis and Morrow suggested that linear programming techniques could be used to avoid this problem of "cycling"; I did not implement this suggestion.

A second problem with this algorithm is that the error vector associated with a guessed configuration of current sources did not always tell me unambiguously which source to move on the next guess or to which location it should be moved. The situation shown in figure 6 is idealized in this sense; there is no ambiguity in the case illustrated there. However, it often happened that the largest entry in the error vector occurred at a rim node on a row (or column) of the network which had no sources in it. A similar situation frequently occurred with the smallest entry in the error vector. In these cases I chose to move the source nearest (fewest resistor segments away from) the rim node corresponding to the maximum value of the error vector. This source was moved to the node nearest the rim node corresponding to the minimum entry of *error* for which the region would still be x–convex.

One further suggestion made by Professors Curtis and Morrow for improving the given algorithm was to compute the center of mass of the desired region by using the given rim potentials resulting from it. If the center of mass were known, the initial guess in the algorithm could be distributed around this point, thus making the guesses more accurate from the beginning; unfortunately, I had no opportunity to use this suggestion either.

References

1. Snell, J. L. and Doyle, P. G.; Random Walks and Electrical Networks, Mathematical Association of America, 1982.
2. Curtis, E. B. and Morrow, J. A.; Determining the resistors in a network, SIAM J. Appl. Math, 50(3), pp. 918–930, 1990.
3. Smith, G. D.; Numerical Solution of Partial Differential Equations:

Finite Difference Methods, Third Edition, Clarendon Press, Oxford, 1985.
4. Johnson, L. W. and Riess, R. D.; Numerical Analysis, Second Edition, Addison–Wesley Publishing Company, Philippines, 1982.