AMO - Advanced Modeling and Optimization, Volume 4, Number 3, 2002

An Electrostatic Model of Multiple Path Planning [1]

L. A. Reibling [2]
Department of Computer Science
Azusa Pacific University, Azusa, CA, USA

Abstract

A model for path planning based upon a natural phenomenon is presented. An electrostatic field is shown to have a similarity to path planning in generating multiple, alternative solution paths. Analysis of the electrostatic model results in a partial differential equation for the potential field and its boundary conditions which correspond to the path planning problem requirements. A finite difference approximation for computing the numerical solution to the partial differential equation is also derived. This finite difference approximation is the basis for a neural network architecture designed to compute the numerical solution of the potential field. Gradient descent over the potential field produces the multiple path solutions.

Keywords: multiple path planning, route planning, artificial potential field, massively parallel architecture, natural parallelism

I. Introduction

This paper describes a new model for path planning problems[21]. The basis for this new model of path planning recognizes that certain computational problems can be described by physical analogies in nature. This new path planning model uses an analogy of electromagnetic field theory for the mathematical model. The solution to the application problem is provided by a partial differential equation in field theory.

The model taken from electromagnetic phenomena is based on electrostatics. The electrostatic model is used to compute multiple, parallel paths which avoid regions of high cost. The

[1]Support provided through Independent Research and Development project funding from Smiths Industries Aerospace & Defense Systems, Inc., and additional support provided through an Accomplished Scholar's Award grant funded by Azusa Pacific University.

[2]Correspondence: Lyle A. Reibling, Department of Computer Science, Azusa Pacific University, 901 E. Alosta Ave., Azusa, CA 91702 USA, Email: reibling@apu.edu

mathematical model describes paths through a region containing a variable cost function as a problem in mathematical physics. One such problem is finding the distribution of current flow through a nonuniform conducting media such as a plate of nonhomogeneous resistive material. The cost function for the path planning problem is analogous to the nonuniform resistivity of the media. The solution to the Laplacian partial differential equation is a potential field in the media. Paths are computed which are orthogonal to the equipotential contours of this potential field.

II. Background

Problems may be defined as state spaces. The search algorithm solves the problem by finding the sequence of operators which converts the initial state into the final state. This sequence is the problem solution, and is described as a path through the problem space from the initial state to the final state. In problem spaces which correspond to physical space, the resulting path represents a physical route.

Trajectory generation for aircraft, unmanned guided vehicles, and other robotic systems is a well studied application of path planning search. It has been mapped into a multidimensional grid representation with each cell of the grid containing the cost of flying through that cell. The trajectory optimization problem then is to find the minimum cost path through the grid by searching. An optimal solution to a problem is obtained when the search results in the best possible answer. But in many cases, the best solution may be too costly or prohibitive to find, so reasonable solutions are suitable alternatives. Good approximations to optimal solutions often serve well as satisfactory results.

Much research regarding combinatorial search algorithms[2, 18, 26] have focused on traditional algorithm development and analysis for sequential processors. Some research[1, 11, 15] has addressed converting these algorithms into parallelized versions for execution on multiprocessors. Those approaches also begin with the serial perspective on algorithm design. This research will show a unique approach to searching that does not begin with a serial viewpoint, but rather a truly parallel model which encompasses the problem space representation as well as the algorithmic model.

III. The Model

The electrostatic model is a physical model in nature which is used for the natural analogy of parallel path planning. The natural analogy contains corresponding entities between the physical model and the abstract path planning model. The paths of the path planning model

correspond to the current flux lines of the electrostatic model. These flux lines describe the distribution of current flow. The variable cost region of the path planning model corresponds to the nonuniform conducting medium of the electrostatic model. A plate of nonhomogeneous resistive material is an example of such a medium. With this analogy, finding good paths is like describing the manner in which current flows through the medium. The solution of the distribution of current flow can be used as a corresponding solution for path planning.

When a potential difference is applied at two different points in a resistive media, current flows along paths between the points where the different potentials are applied. The point with the higher potential is the source point, and the point with the lower potential is the sink point. The paths of current flow are determined by the resistivity of the media and the locations of the source and sink points. The distribution of current flow along these paths is optimized by nature in some fashion. Current path distribution is less dense through regions where the media has a higher concentration of resistivity. Conversely, regions with more conductivity (lower resistivity) have a greater density of current distribution.

Current flow is along flux lines which correspond to the current density field. The current density field is continuous, so there are an infinite number of flux lines from the source point to the sink point. From the source to the sink, the current flux lines can be traced along the direction specified by the current density vector field. (Just as magnetic flux lines can be seen between the poles of a bar magnet by sprinkling metal filings upon a sheet of paper placed on top of the magnet.)

In the electrostatic model for path planning, several analogies to nature are defined. The variable cost function for the search problem is represented by the nonuniform resistivity of the media. This media has the same dimensionality as the problem space. The range of each dimension of the problem space defines the extent of a grid upon which a spatial sampling is performed for the architecture. Time may be explicitly represented as one of the dimensions, if the cost function is a function of time. The source and sink locations represent the start and goal nodes, respectively of the search space.

The mathematical model is based upon electric field theory[19]. The electric field is a vector field which is defined as

$$\boldsymbol{E} = -\nabla\phi \tag{1}$$

where ϕ is the scalar potential field. The current density field is also a vector field defined as

$$\boldsymbol{j} = \sigma\boldsymbol{E} \tag{2}$$

where σ is the conductivity of the media. From the definition for current density and its divergence in steady state conditions the N- dimensional electric field potential $\phi \equiv$

$\phi(x_1, x_2, \ldots, x_N)$ with x_i as orthogonal coordinates of a nonuniform conductive media can be derived[21] as the solution to the second order partial differential equation

$$\nabla^2\phi + \rho\nabla\sigma \cdot \nabla\phi = 0 \tag{3}$$

where $\rho \equiv \rho(x_1, x_2, \ldots, x_N)$ is the nonuniform resistivity of the media, and $\sigma = \rho^{-1}$. The derivation of Equation 3 is shown in Appendix A. The cost function is modeled by the resistivity of the conducting media. Expanding the gradient and divergence operators in the two dimensions x, y results in the following second order partial differential equation

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{1}{\sigma}\frac{\partial\sigma}{\partial x}\frac{\partial\phi}{\partial x} + \frac{1}{\sigma}\frac{\partial\sigma}{\partial y}\frac{\partial\phi}{\partial y} = 0 \tag{4}$$

This is expressed more conveniently using subscript notation for partial derivatives as

$$\phi_{xx} + \phi_{yy} + \rho\sigma_x\phi_x + \rho\sigma_y\phi_y = 0 \tag{5}$$

By solving Equation 5 for ϕ and substituting into Equation 1, the current flow vector field, $\boldsymbol{j}$, can be found by substituting Equation 1 into Equation 2.

Since the electric field and current density field result from the gradient of a scalar field, they are conservative fields, and the field lines emanate from source charges and terminate on sink charges[25]. These field lines represent possible solutions to the search problem as multiple paths. These curved lines then correspond to multiple paths of the corresponding path planning problem. These multiple path solutions do not include all of the possible solution paths in the exponential search space. In using the field lines as paths, the solution space is restricted to include only those paths which do not share any path segments. The multiple paths will not split from any common path or join together into a single path. The multiple paths all originate at the start state as unique alternatives.

The field lines for the electrostatic model are used as the path definitions. The field lines are computed once the electrostatic potential field ϕ is derived for the problem from the solution to Equation 5. The current density vector field $\boldsymbol{j}$ of Equation 2 is used to trace the field lines. This vector field is the tangential field for the solution. There are an infinite number of field lines which leave the source point and enter the sink point. At every point in the potential field with the exception of the source and sink, the vector field (the electric field vector) is defined. From the source point, an angle (initial heading at the start node) is selected from which to trace out the field line. Using a small incremental path step, the path progresses along this direction. From this point on, the gradient vector can be computed which will define the direction of the path descent over the potential field surface, obtaining the path as the field line. Gradient descent over the potential field will not be troubled with

local minima since the potential field cannot have maximum or minimum values except at the source and sink locations[24].

In addition to the source and sink values included as boundary conditions described previously, it is necessary to establish boundary conditions along the edge of the grid for the problem. These boundary conditions will determine what happens to the paths where they approach the edge of the grid. The grid itself is assumed to be large enough to include a sufficient portion of the path planning area in which multiple solutions are needed. Since all of the paths should be continuous from start to goal within this planning area, it is desired that none of the solution paths run off the grid. Thus, asymptotically near the edge of the grid, each path should approach a parallel course to the edge.

For the field lines to be asymptotically parallel at the edge of the problem grid, it is necessary to establish the equipotential contours as normal to the grid edges as they approach the edge. This is accomplished by setting up the following boundary conditions:

$$\left.\frac{\partial \phi}{\partial y}\right|_{(y=y_{\min}, y=y_{\max})} = 0 \tag{6}$$

and

$$\left.\frac{\partial \phi}{\partial x}\right|_{(x=x_{\min}, x=x_{\max})} = 0 \tag{7}$$

The first boundary condition (Equation 6), causes the field lines, $\boldsymbol{E}$, to be parallel to the unit vector, $\boldsymbol{x}$, since along the top and bottom edges of the grid the y-component of the potential gradient is zero. Similarly, the second boundary condition (Equation 7) causes parallel solutions of $\boldsymbol{E}$ along the left and right edges of the grid because the x-component of the potential gradient is zero also.

IV. The Architecture

This section develops the architectural design of a neural network[21] that computes the numerical solution to the Laplacian partial differential equation. The architecture implements a finite difference approximation[6] to compute a numerical solution to the scalar partial differential equation. The numerical approximation defines an artificial neural network processing unit and its weighted interconnections to a selected set of neighboring processing units. The approximation templates of the difference formula define the connection strengths.

Appendix B contains the details of the finite difference approximation which computes the numerical solution to the Laplacian equation. The approximation equation for the potential field ϕ at grid location P is

$$U_P = \frac{1}{4}\left[\left(1+\frac{C_x}{2}\right)U_{Q_1} + \left(1+\frac{C_y}{2}\right)U_{Q_2} + \left(1-\frac{C_x}{2}\right)U_{Q_3} + \left(1-\frac{C_y}{2}\right)U_{Q_4}\right] \tag{8}$$

where $C_z = \sigma_z/\sigma$, and $Q_i, i = 1,2,3,4$ are the four nearest neighbors of the point P in the grid.

This artificial neural network architecture uses a computational grid. Each point of the grid is a processor, and the links of the grid consist of the links between the processors. The mapping of the finite difference approximation into this grid is illustrated in Figure 1. The block diagram which shows the neuron model definition which implements the five point approximation of Equation 8 is also shown in Figure 1.

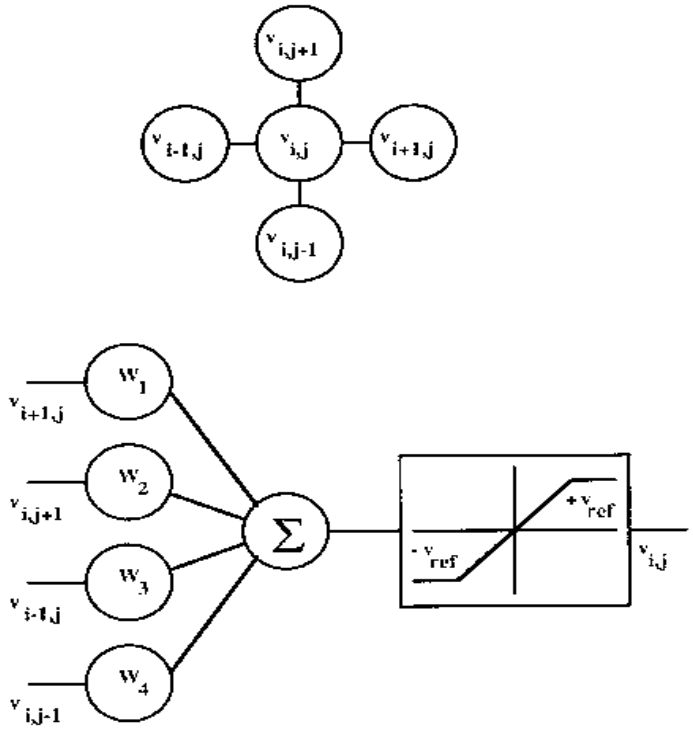

Figure 1: *Neuron Model for the 5-pt. Finite Difference Approximation*

The neuron model is defined as consisting of an input function and output function. The input function u_o computes the weighted sum of the interconnections from the other neurons,

$$u_o = \sum_{i=1}^{n} W_i V_i \tag{9}$$

and the output function $v_{i,j}$ computes the linear ramp between the saturation voltages $-V_{ref}$, $+V_{ref}$ of the operational amplifier.

The neural network is used to perform a parallel computation of the scalar potential field ϕ at every spatial point. Thus, the network computes a numerical solution to Equation 5. The architecture of the neural network must solve an elliptic partial differential equation. There are many numerical methods for solving partial differential equations. In this research, finite

differences were employed to solve the potential field Equation 5. Such an approximation is the five-point formula defined in Equation 8. For the neural network implementation, the neuron input function $u_{i,j}$ is defined as

$$u_{i,j} = \frac{1}{4}\left[(1+\frac{\sigma_x}{2\sigma})v_{i+1,j} + (1+\frac{\sigma_y}{2\sigma})v_{i,j-1} + (1-\frac{\sigma_x}{2\sigma})v_{i-1,j} + (1-\frac{\sigma_y}{2\sigma})v_{i,j-1}\right] \qquad (10)$$

The synaptic weights implement the non-uniformity of the cost functions (the coefficients of Equation 10 which contain σ_x and σ_y). The weights to neuron(i,j) are shown in Table 1.

Table 1: *Neural Network Weight Definitions for the 5-pt. Approximation*

Weight	From Neuron	Value
W_1	$(i+1, j)$	$(1+\sigma_x/2\sigma)/4$
W_2	$(i, j+1)$	$(1+\sigma_y/2\sigma)/4$
W_3	$(i-1, j)$	$(1-\sigma_x/2\sigma)/4$
W_4	$(i, j-1)$	$(1-\sigma_y/2\sigma)/4$

The non-linear neuron output function $v_{i,j}$ is defined as

$$v_{i,j} = \begin{cases} -V_{ref} & : \quad u_{i,j} < -V_{ref} \\ u_{i,j} & : \quad -V_{ref} \leq u_{i,j} \leq +V_{ref} \\ +V_{ref} & : \quad u_{i,j} > +V_{ref} \end{cases} \qquad (11)$$

The entire search space is constructed by replicating this portion of the neural network over the entire grid. The output of the source and sink neurons which correspond to the start and goal nodes are clamped to $+V_{ref}$ and $-V_{ref}$, respectively. Since the clamping of the source and sink to the maximum and minimum (respectively) potential value occurs on the boundary of the problem, it is appropriate to use these clamped values as the limits of the neuron output function (Equation 11), since all potential values inside the boundary of the problem are guaranteed to lie between these limits.

The electrostatic model neural network becomes an element in a system which can be applied to the generation of physical trajectories. Such a trajectory generator is shown as a block diagram in Figure 2. Inputs to the neural network are the start and goal coordinates for the trajectory, and the explicit cost function values for each cell of the problem grid. The network settling is checked against an input parameter for convergence. This check tests if the sum-squared difference of all the neuron outputs between successive iterations is less than the input parameter. When the neural network settles to its stable state, the neuron outputs define the potential surface solution for the electrostatic model. This surface is provided to the gradient descent procedure, along with the start and goal coordinates, and a discrete signal indicating that the network outputs are settled. The descent procedure

also receives a direction angle as an external input. This angle is the heading at the start coordinate to use in initializing the descent procedure. The procedure then outputs the list of trajectory coordinates which are obtained from the descent starting with this heading. The accumulated cost along this trajectory is also provided with the list of coordinates.

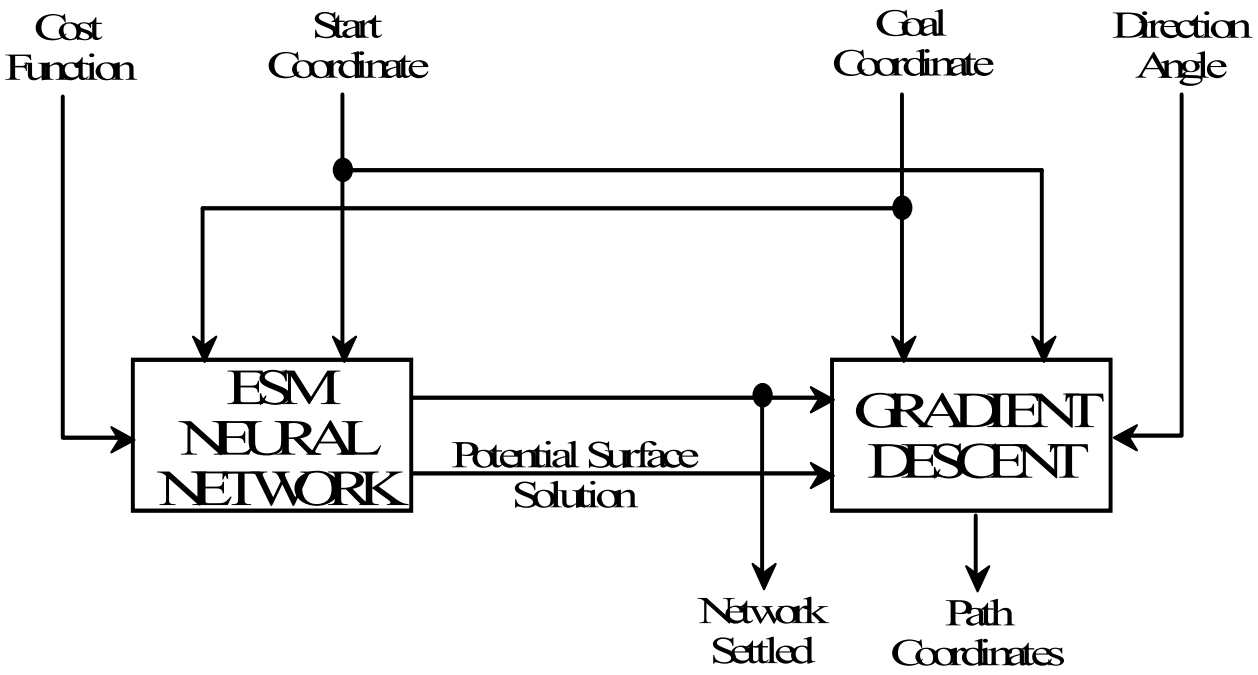

Figure 2: *System Block Diagram of the Electrostatic Model Neural Network*

Once the neural network has settled, new direction angles may be input to direct the trajectory generator to supply a new list of trajectory coordinates. This is used in the trajectory generation system which is illustrated in Figure 3. The trajectory generator of Figure 2 in contained in this system as shown in the block diagram. The current aircraft heading is supplied to a trajectory analysis function as an initial heading. The current aircraft location is supplied to the trajectory generator as the start coordinate. The trajectory analysis function passes the initial heading as the current direction angle to the neural network. When the network is settled, the trajectory coordinates and the final accumulated cost of the trajectory is input to the analysis function. This function can then adjust the direction angle and receive a new set of trajectory coordinates and accumulated cost from the neural network, without a re-settling of the network, since the cost, start, and goal conditions have not been changed. This allows the analysis function to employ comparison techniques to converge on the minimum cost trajectory as a function of heading. An example of techniques to use could be a binary search over heading, or a branch and bound search over heading. When the analysis function finds the best trajectory according to its search criteria,

it signals that the trajectory has been found and the corresponding trajectory coordinates are output.

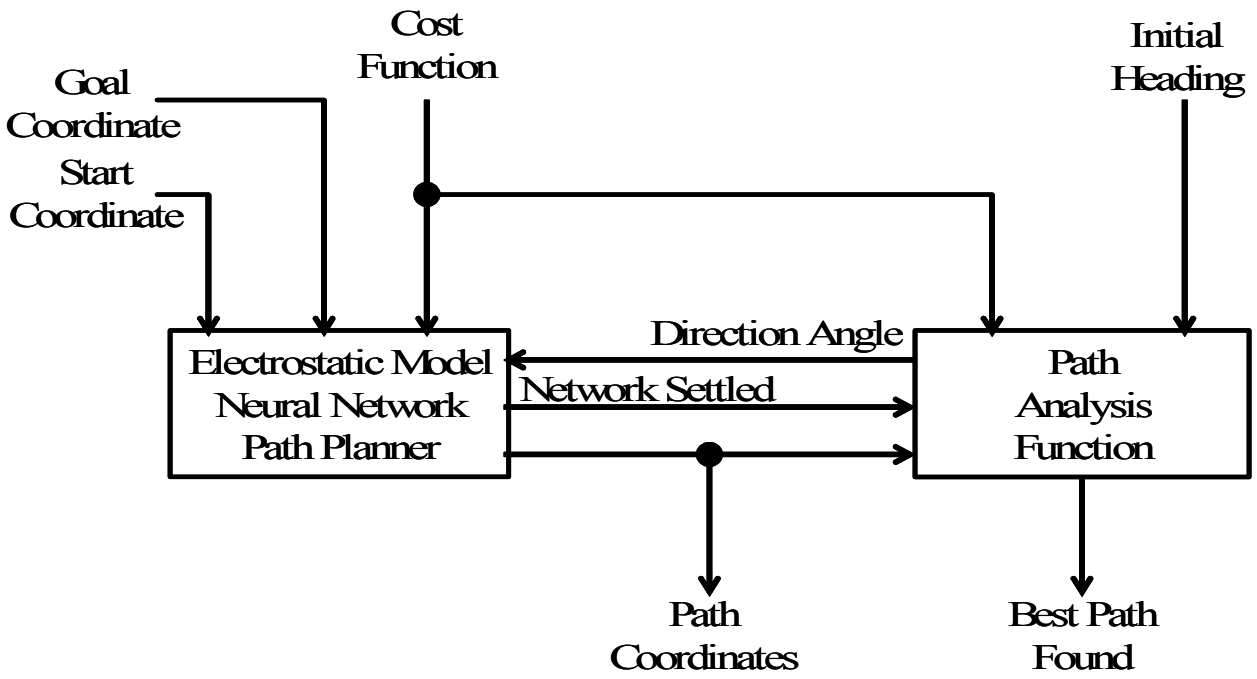

Figure 3: *Block Diagram of a Trajectory Generation System*

V. Results & Discussion

The paths are extracted from the neural network once it settles. The network computes the scalar potential field, as described previously. The final values of the nodes represent a surface with a global maximum at the start node and a global minimum at the goal node. A direction is selected for a path at the start node. A gradient descent algorithm is performed over the potential surface to extract the path for that direction with bivariate interpolation of the potential values used between the grid points. The algorithm computes the new location, new accumulated cost, and new gradient angle. Each pass through the algorithm generates another point along the path. This algorithm is repetatively invoked until the entire path from the start node to the goal node is extracted from the gradient descent of the potential surface. This will have a complexity of $O(L)$, where L is the length of the path.

The neural network computes a finite difference approximation, which can be expressed in matrix form. The matrix equation for the neural network is

$$\boldsymbol{u}^{(k+1)} = \boldsymbol{A}\boldsymbol{u}^{(k)} \tag{12}$$

where $\boldsymbol{u}^{(j)}$ is the j^{th} update to the $(n^2 \times 1)$ neuron vector

$$\boldsymbol{u} = \begin{bmatrix} u_{1,1} \\ u_{1,2} \\ \vdots \\ u_{1,n} \\ u_{2,1} \\ \vdots \\ u_{n,n} \end{bmatrix} \tag{13}$$

$\boldsymbol{A}$ is the tri-diagonal square $(n^2 \times n^2)$ matrix defined as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{T} & \frac{1}{2}\boldsymbol{B} & & 0 \\ \boldsymbol{Y_N} & \boldsymbol{T} & \boldsymbol{Y_P} & \\ \ddots & \ddots & \ddots & \\ & & & \boldsymbol{Y_P} \\ 0 & & \frac{1}{2}\boldsymbol{B} & \boldsymbol{T} \end{bmatrix} \tag{14}$$

along with the square $(n \times n)$ matricies

$$\boldsymbol{B} = \boldsymbol{I} \tag{15}$$

$$\boldsymbol{T} = \begin{bmatrix} 0 & 1/2 & & 0 \\ (1-\frac{C_x}{2})/4 & 0 & (1+\frac{C_x}{2})/4 & \\ \ddots & \ddots & \ddots & \\ & (1-\frac{C_x}{2})/4 & 0 & (1+\frac{C_x}{2})/4 \\ 0 & & 1/2 & 0 \end{bmatrix} \tag{16}$$

$$\boldsymbol{Y_P} = \begin{bmatrix} (1+\frac{C_y}{2})/4 & & & 0 \\ & (1+\frac{C_y}{2})/4 & & \\ & & \ddots & \\ 0 & & & (1+\frac{C_y}{2})/4 \end{bmatrix} \tag{17}$$

$$\boldsymbol{Y_N} = \begin{bmatrix} (1-\frac{C_y}{2})/4 & & & 0 \\ & (1-\frac{C_y}{2})/4 & & \\ & & \ddots & \\ 0 & & & (1-\frac{C_y}{2})/4 \end{bmatrix} \tag{18}$$

Each iteration of Equation 12 requires $O(n^4)$ operations on a Von Neumann processor, but only a single cycle on the neural network. The neural network implements the five point finite difference approximation as a point iterative Jacobi[14] algorithm (all points are computed simultaneously in parallel, however). Lapidus and Pinder show[14] that the rate of convergence of the iterations of Equation 12 is inversely proportional to the square

of the discretization of the grid $(1/n)$. Therefore, the convergence of the neural network approximation is $O(n^2)$. The complexity of the algorithm is

$$O(n^2) + O(\text{all paths}) = O(n^2) + \text{no. of paths} \cdot O(\text{path length}) \tag{19}$$

The $O(\text{path length})$ is $O(n^2)$ since the worst case length includes all nodes of the neural network. The number of paths computed is linear in heading at the start node, so the overall complexity of the algorithm is $O(n^2)$.

Experimental verification of the solution and performance characteristics of the massive parallel architectures was accomplished using simulations on a VAX and a Connection Machine. The results and performance of the neural network which implements the electrostatic model is reviewed, including the simulations on the Connection Machine. The solutions generated by the massively parallel architecture have been checked against known admissible algorithm results.

The results using the electrostatic model show an average difference of eight percent in computing the best cost paths relative to sequential best first search. The analog circuit simulation on the 16k CM-2 Connection Machine resulted in a real-time performance projection of 18 milliseconds for a test case whose network had 3,600 neurons.

Several test cases were used to experiment with the architecture simulations in this research.

Many neural network investigations have used at most 100 neurons in their test cases. This test set included networks with several thousand neurons. Figure 4 shows an example of the multiple paths computed by the architecture along with the benchmark optimal path. The contour plot depicts the cost function of the test case. In the figure, seven paths are illustrated which are separated at the start node by a 45 degree direction angle.

Figure 5 illustrates the solution cost from the simulated architecture. The path cost shown is that of the least cost path over 360 headings around the start node with a one degree increment. At the end of each cycle of the simulated architecture, the path defined by the flux line at each of the 360 angles was computed and the cost of the path accumulated. The plotted cost is the minimum of all of these 360 possible paths at each pass of the simulation. This figure illustrates the convergence of the minimum cost path to asymptotic values for the electrostatic model.

One of the features of the Electrostatic Model for path planning is the generation of multiple paths in the solution space. Once the potential surface is computed by the neural network, gradient descent from the start node over the surface extracts the path. The initial direction angle, which is given as a heading, determines a unique path to the goal node. For all of the test cases, a one degree increment was used at the start node heading to create the

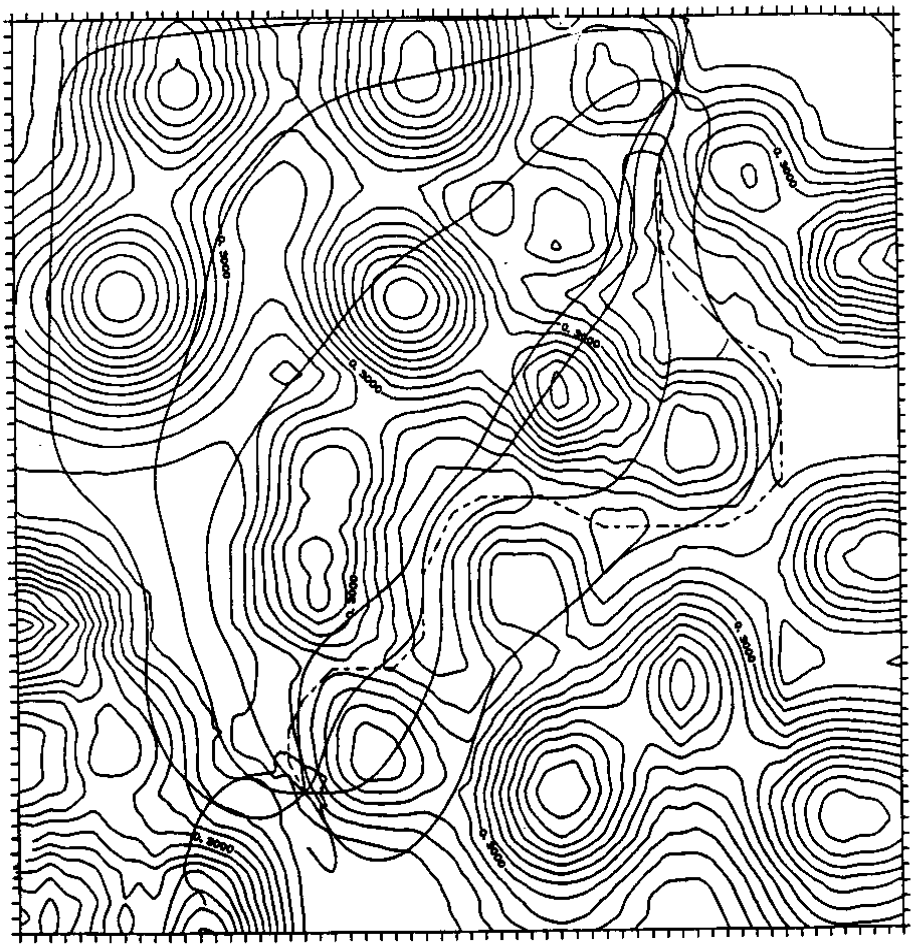

Figure 4: *Multiple Path Results using the Electrostatic Model*

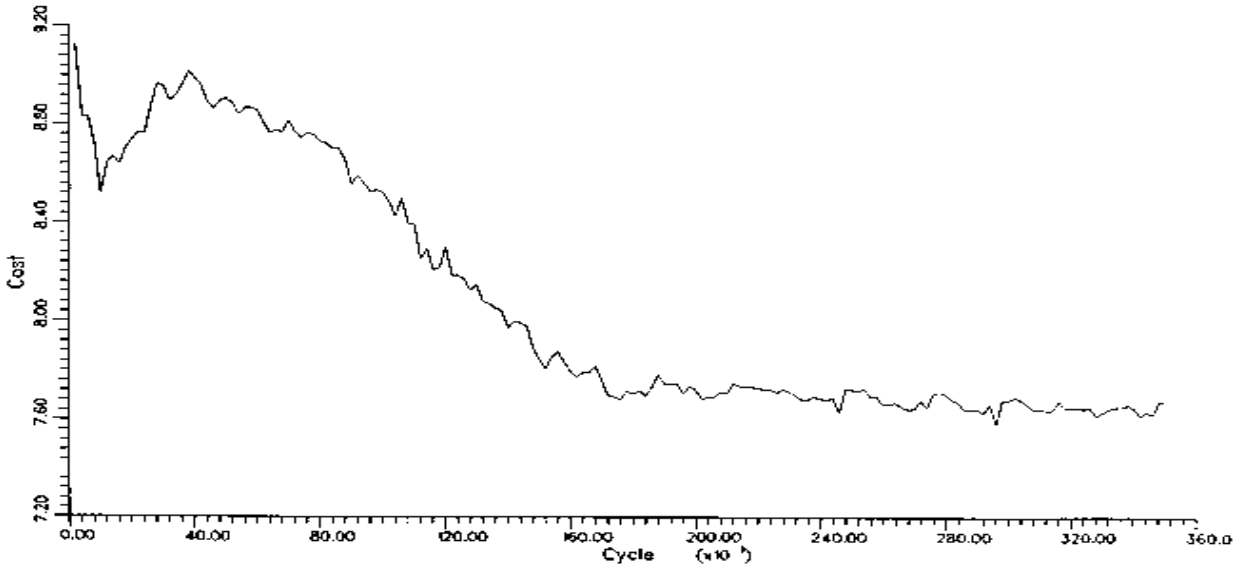

Figure 5: *Least Cost Path Value Found During the Neural Network Settling*

unique paths. This section reviews the results of all the test cases for their multiple paths.

Table 2 shows the best ten paths for the example case. The costs of the first 10 paths had a range of 0.121 which is approximately 1.6 percent of the best path cost of 0.7672. The algorithm computed a total of 320 paths for the test case. The costs of these paths are plotted in sorted order from the best value to the worst value in Figure 6. The figure illustrates a linear cost increase of only 0.027 units per path up through the first 215 paths which were generated. This is a mere 0.35 percent increase in the cost per path. Of the 215 paths generated for the test case, these 215 nearly linearly increasing path values account for 67 percent of the solutions which were generated by the algorithm.

Similar results held for all of the test cases. The test cases showed that the increase in path cost values for the multiple paths starts out linearly, followed by a more dramatic rise in the path cost values. Over the linear region of these path cost values, the slope of the linear increase averaged only 0.22 percent of the best path value per new path. This linear portion of the multiple paths averaged to be 63 percent of the multiple paths generated over all of the test cases.

Assessment of the data from all of the test cases showed that the value of the paths generated by the neural network had an average difference of 8.3 percent over the value of the optimal path for the test case. Even if 8.3 percent is a higher penalty for using the neural

Table 2: *Costs and Headings for 10 Best Path Values of an Example Case*

RANK	Path Cost Value	Heading
1	7.672	92
2	7.678	96
3	7.694	95
4	7.701	93
5	7.703	94
6	7.725	89
7	7.729	97
8	7.742	90
9	7.790	91
10	7.793	99

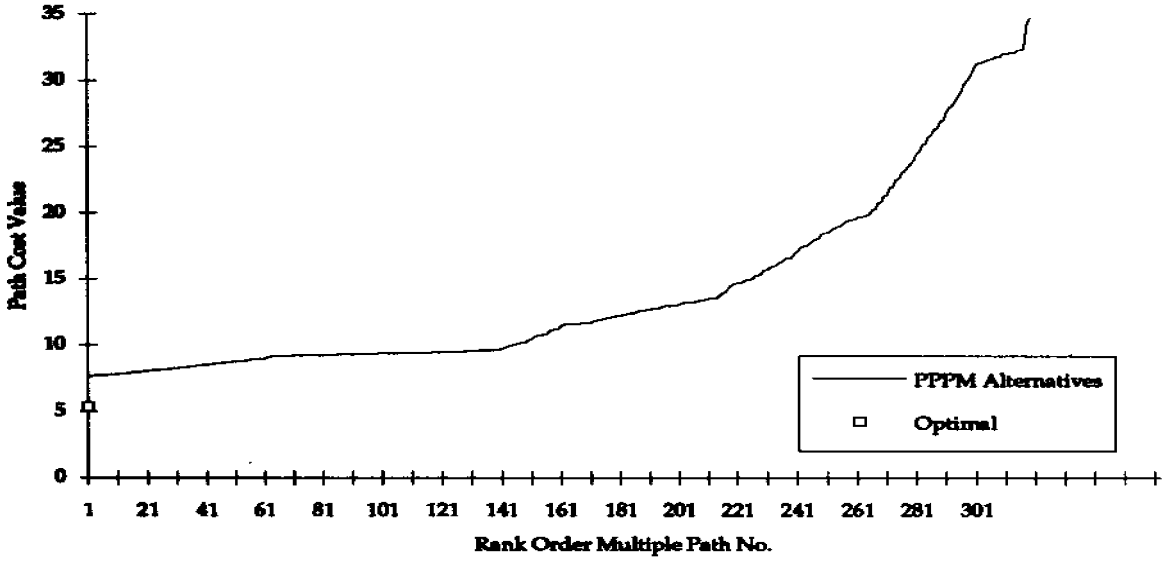

Figure 6: *Plot of Multiple Path Values Ordered by Cost*

network in the path planning application, the benefits of the multiple paths generated are the compensation of the method. The data revealed a significant effect from the multiple path generation. This effect was that the costs of the top-ranked paths generated by the neural network increased linearly for at least the first half of the alternative paths. Furthermore, the rate of increase averaged to be only 0.22 percent of value of the least cost path found by the neural network. So although the value of the least cost path may be higher than the optimal path (the optimization function is derived in Appendix C), the architecture will generate many additional paths which do not significantly degrade in cost.

Other solution models have been based on the notion of a potential field to represent the cost function. These methods are known as potential field method or artificial potential fields. The potential field method as a path planning system was first introduced in 1985[12]. In these path-finding solutions, obstacles in a problem space are assigned repulsive potential fields, which then affect the movement of the object as it approaches them. The object to be moved and the obstacles would be assigned similar charges, thereby repulsing each other as they drew nearer. The destination point would be assigned an opposite charge, attracting the moving object to its final place of rest. Possible solution paths fall along potential valleys that occur between the obstacles. From these possible paths, one is chosen according to certain cost functions, which may be associated with distance, cumulative potential, or even both.

This method has many advantages, including speed and wide clearance of obstacles, which make it very useful in real-time systems. In fact, the main application of this method is geared towards robot arm manipulation or robot navigation. There are several disadvantages however, about which many have written in the years since this approach was published. Many of these address the problem of local minima which result in states of rest not at the destination. These can occur at a number of places in the problem space, most likely in concave obstacles or narrow passages.

The approach to path-finding described in this article is fundamentally quite different. In this approach, instead of assigning potential fields to the obstacles themselves, the only points with pre-assigned potential are the source and destination points. The problem area is then represented as a resistivity plane - simulating a plate with non-uniform resistance. Obstacles in the problem space are mapped to areas of high resistivity. In accordance with laws of physics, the difference in the potentials of the source and destination points creates a current that flows from the higher potential to the lower. This current flows along parallel paths of least resistance, avoiding the areas of high resistivity (obstacles). One aspect of this approach that sets it apart from other potential field methods is that paths are complete. There are no local minima to worry about. Each parallel solution path goes from the source

to the destination, just as in nature current flux field flows between its source and sink. As this research shows in various test cases, choosing the optimal path from among these parallel paths results in a solution very close to the optimal path found by complete (as opposed to approximate) algorithms.

How does the potential field method (PFM), or artificial potential field (APF), compare to the electrostatic model used in the path planning model of this research? The solution to the nonhomogeneous Laplacian of the ESM (Electrostatic Model) is a potential field. The field in fact is the electrostatic potential. This field governs the current density field via Equation 2 and Equation 5. The current flux is orthogonal to the current density field is used to represent the path solutions.

The PFM/APF approach sets up charge distributions for the planning obstacles that in tern exert an attractive or repulsive force on a particle in motion through the potential field. Thus the field is pre-existing based on the static locations of the charge distribution and exerts a local force to a moving particle. The ESM by contrast uses the resistivity distribution to set up a potential field solution unique to the location of the source and sink (start and goal nodes) locations.

VI. Conclusions

One of the most difficult tasks of this research was to develop a model of path planning search which had a true parallelism inherent in the model. This interest came about since strictly sequential algorithms were not directly applicable to artificial neural network architectures. Parallelized versions were not of interest either since they are targeted for loosely coupled multiprocessor systems. The path planning model required the feasibility of a more direct use on massively parallel architectures such as artificial neural networks.

The second interest, and perhaps a more compelling requirement for path planning, was a desire for the resulting system to produce alternative sub-optimal solutions. These alternative paths provide choices at the starting node to the user of the path planning system. For various operational considerations, users often desire alternative choices when employing automated planning systems. But this is a very difficult proposition for classical sequential search algorithms. The principle of optimality does not support the consideration of simultaneous, sub-optimal decisions.

It was at this point that the electrostatic model was conjectured to satisfy these requirements for a directly parallelized model and alternative sub-optimal paths. The idea was simple enough — nature "automatically" distributes electrical current through conducting material in an optimal manner, although this optimization criterion is different from that

for a least cost path. The paths are also parallel alternatives which have an incremental variation between the adjacent paths. Neighboring (adjacent) paths are defined using the theory of the calculus of variations. When adjusted by a "variational" increment, a new path neighboring the original path[3, 13] is defined. They remain adjacent to each other in a variational sense, preserving the alternative choices for the user. Thus, the electrostatic model was perceived as a viable approach to modeling this path planning problem.

Could the current flux lines then be used as the alternate paths? The result is that they can as was shown. These alternative paths make intuitive sense. They avoid regions of high cost and prefer regions of low cost.

References

[1] Jeffrey Michael Abram. *Shortest-Path Algorithms with Decentralized Information and Communication Requirements.* DSc.thesis, Washington University, 1981.

[2] A. Barr and E.A. Feigenbaum. *Handbook of Artificial Intelligence.* Kaufmann, 1981.

[3] Gilbert A. Bliss. *Lectures on the Calculus of Variations.* University of Chicago Press, 1946.

[4] Oskar Bolza. *Lectures on the Calculus of Variations.* Dover Publications, 1961.

[5] C. Carathéodory. *Calculus of Variations and Partial Differential Equations of the First Order.* Holden-Day, 1965.

[6] Lothar Collatz. *The Numerical Treatment of Differential Equations.* Springer-Verlag, New York, NY, third edition, 1966.

[7] R. Courant and D. Hilbert. *Methods of Mathematical Physics.* Interscience Publishers, 1953.

[8] Richard P. Feynman. *The Feynman Lecture Series on Physics.* Addison-Wesley Publishing Co., 1963.

[9] George E. Forsythe and Wolfgang R. Wasow. *Finite-Difference Methods for Partial Differential Equations.* John Wiley & Sons, Inc., New York, NY, 1960.

[10] L. Fox. *Numerical Solution of Ordinary and Partial Differential Equations.* Pergamon Press, Oxford, England, 1962.

[11] Jun Gu. *Parallel Algorithms and Architectures for Very Fast AI Search.* PhD thesis, University of Utah, 1989.

[12] O. Khatib. "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots" *1985 IEEE International Conference on Robotics and Automation.* March 25-28, 1985. St. Louis, pp. 500-505.

[13] C. Lanczos. *The Variational Principles of Mechanics.* Dover Publications, Inc., New York, NY, fourth edition, 1970.

[14] Leon Lapidus and George F. Pinder. *Numerical Solution of Partial Differential Equations in Science and Engineering.* John Wiley & Sons, Inc., New York, NY, 1982.

[15] Juo-Jie Li. *Parallel Processing of Combinatorial Search Problems.* PhD thesis, Purdue University, 1985.

[16] James Clerk Maxwell. *A Treatise On Electricity and Magnetism.* Volume 1, Dover Publications, Inc., New York, NY, 1954. An unabridged and unaltered republication of the third edition of 1891.

[17] A. R. Mitchell and D. F. Griffiths. *The Finite Difference Method in Partial Differential Equations.* John Wiley & Sons, Inc., New York, NY, 1980.

[18] N.J. Nilsson. *Principles of Artificial Intelligence.* Tioga, 1980.

[19] Allen Nussbaum. *ELECTROMAGNETIC THEORY for Engineers and Scientists.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1965.

[20] Robert Plonsey and Robert E. Collin. *Principles and Applications of Electromagnetic Fields.* McGraw-Hill, 1961.

[21] L. A. Reibling. *A Massively Parallel Architecture Design for Path Planning Applications.* Ph.D.thesis, Michigan State University, 1992.

[22] Mario G. Salvadori and Melvin L. Baron. *Numerical Methods in Engineering.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1961.

[23] G. D. Smith. *Numerical Solution of Partial Differential Equations.* Oxford University Press, New York, NY, 1965.

[24] William R. Smythe. *Static and Dynamic Electricity.* McGraw-Hill, 1950.

[25] Ernst Weber. *Electromagnetic Fields. Volume I — Mapping of Fields,* John Wiley & Sons, Inc., New York, NY, 1950.

[26] Patrick H. Winston. *Artificial Intelligence.* Addison-Wesley Publishing Co., 2nd edition, 1984.

Appendix A
Derivation of the Laplacian Equation

This appendix reviews those aspects of electric field theory which are pertinent to the model. For a more extensive explanation of electromagnetic phenomenon, see [16, 19, 25]. The following analysis will be used to derive Equation 3.

The divergence of the current density $\boldsymbol{j}$ is defined as the charge (ρ) rate of change

$$\nabla \cdot \boldsymbol{j} = -\frac{\partial \rho}{\partial t} \tag{20}$$

In the steady state condition, no charge distribution changes occur

$$\frac{\partial \rho}{\partial t} = 0 \tag{21}$$

Thus, in the steady state condition,

$$\nabla \cdot \boldsymbol{j} = 0 \tag{22}$$

Substitution of Equation 2 into Equation 22 gives

$$\nabla \cdot (\sigma \boldsymbol{E}) = 0 \tag{23}$$

or, equivalently through a vector identity that

$$\nabla \sigma \cdot \boldsymbol{E} + \sigma \nabla \cdot \boldsymbol{E} = 0 \tag{24}$$

Substituting Equation 1 into Equation 24 yields

$$\nabla \sigma \cdot (-\nabla \phi) + \sigma \nabla \cdot (-\nabla \phi) = 0 \tag{25}$$

After dividing the equation by $(-\sigma)$, the equation for the electric field potential of a nonuniform conductive media is

$$\nabla^2 \phi + \frac{1}{\sigma} \nabla \sigma \cdot \nabla \phi = 0 \tag{26}$$

Equation 26 is called the Laplacian Equation.

Appendix B
Defining the Finite Difference Approximation

It is desired that the general second-order partial differential equation

$$L(u) = A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} + D\frac{\partial u}{\partial x} + E\frac{\partial u}{\partial y} + Fu = 0 \tag{27}$$

be approximated[9, 10, 14, 17, 22, 23] on a unit grid by

$$L_1(u) \approx \alpha_o u_P - \sum_{i=1}^{n} \alpha_i u_{Q_i} \tag{28}$$

with the n neighbors of P being $Q_1, \ldots, Q_n$, where $Q_i = (x+\xi_i, y+\eta_i)$. The Taylor series about $P = (x,y)$ in two variables is

$$\begin{aligned} f(x+ih, y+jk) &= f(x,y) + ih\frac{\partial}{\partial x}f(x,y) + jk\frac{\partial}{\partial y}f(x,y) \\ &+ \frac{(ih)^2}{2}\frac{\partial^2}{\partial x^2}f(x,y) + ijhk\frac{\partial^2}{\partial x \partial y}f(x,y) \\ &+ \frac{(jk)^2}{2}\frac{\partial^2}{\partial y^2}f(x,y) + \ldots \end{aligned} \tag{29}$$

Let the neighborhood of P be defined as the n points (an $n+1$ point approximation template) $Q_i = \{(x+\xi_i, y+\eta_i)\}$, where ξ_i, η_i are integers ($h = k = i$ for the unit grid). The expansion of $u(x+\xi_i, y+\eta_i)$ about P is

$$\begin{aligned} u(x+\xi_i, y+\eta_i) = U_{Q_i} &= u(x,y)|_P + \xi_i \left.\frac{\partial u}{\partial x}\right|_P + \eta_i \left.\frac{\partial u}{\partial y}\right|_P \\ &+ \left.\frac{\xi_i^2}{2}\frac{\partial^2 u}{\partial x^2}\right|_P + \xi_i\eta_i \left.\frac{\partial^2 u}{\partial x \partial y}\right|_P \\ &+ \left.\frac{\eta_i^2}{2}\frac{\partial^2 u}{\partial y^2}\right|_P + \ldots \end{aligned} \tag{30}$$

So Equation 28 upon substitution by Equation 30 and collection of common terms becomes

$$\begin{aligned} L_1(u) = 0 &= \left.\frac{\partial^2 u}{\partial x^2}\right|_P \frac{1}{2}\sum_{i=1}^{n}\alpha_i\xi_i^2 + \left.\frac{\partial^2 u}{\partial x \partial y}\right|_P \sum_{i=1}^{n}\alpha_i\xi_i\eta_i + \left.\frac{\partial^2 u}{\partial y^2}\right|_P \frac{1}{2}\sum_{i=1}^{n}\alpha_i\eta_i^2 \\ &+ \left.\frac{\partial u}{\partial x}\right|_P \sum_{i=1}^{n}\alpha_i\xi_i + \left.\frac{\partial u}{\partial y}\right|_P \sum_{i=1}^{n}\alpha_i\eta_i + u_P\left(\sum_{i=1}^{n}\alpha_i - \alpha_0\right) \end{aligned} \tag{31}$$

Substituting the coefficients of Equation 31 for those of Equation 27 gives the system of equations

$$\sum_{i=1}^{n}\alpha_i - \alpha_0 = F \tag{32}$$

$$\sum_{i=1}^{n}\alpha_i\eta_i = E \tag{33}$$

$$\sum_{i=1}^{n}\alpha_i\xi_i = D \tag{34}$$

$$\sum_{i=1}^{n}\alpha_i\eta_i^2 = 2C \tag{35}$$

$$\sum_{i=1}^{n} \alpha_i \xi_i \eta_i = B \tag{36}$$

$$\sum_{i=1}^{n} \alpha_i \xi_i^2 = 2A \tag{37}$$

The solution of the coefficients α_i of the above system of equations allows approximating the value of a grid point by solving Equation 28 for u_P

$$u_P = \frac{1}{\alpha_0} \sum_{i=1}^{n} \alpha_i u_{Q_i} \tag{38}$$

For the nonhomogeneous equation, set the quantities $C_x = \sigma_x/\sigma$ and $C_y = \sigma_y/\sigma$, resulting in

$$L_1(\phi) = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + C_x \frac{\partial \phi}{\partial x} + C_y \frac{\partial \phi}{\partial y} = 0 \tag{39}$$

and the coefficients of Equation 27 are $A = 1, B = 0, C = 1, D = C_x, E = C_y,$ and, $F = 0$. It is desired to derive a five-point finite difference formula. The coefficients are as shown in Table 3. Substituting the values of Table 3 into the system defined by Equation 32 through

Table 3: *Coordinates of the 5-pt. Finite Difference Formula*

i	ξ_i	η_i
1	1	0
2	0	1
3	-1	0
4	0	-1

Equation 37, yields the following system of equations

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = \alpha_0 \tag{40}$$

$$\alpha_2 - \alpha_4 = C_y \tag{41}$$

$$\alpha_1 - \alpha_3 = C_x \tag{42}$$

$$\alpha_2 + \alpha_4 = 2 \tag{43}$$

$$\alpha_1 + \alpha_3 = 2 \tag{44}$$

The solution to this set of equations is

$$\alpha_1 = 1 + \frac{C_x}{2} \tag{45}$$

$$\alpha_2 = 1 + \frac{C_y}{2} \tag{46}$$

$$\alpha_3 = 1 - \frac{C_x}{2} \tag{47}$$

$$\alpha_4 = 1 - \frac{C_y}{2} \tag{48}$$

$$\alpha_0 = 4 \tag{49}$$

Substituting Equation 45 through Equation 49 into Equation 28 and solving for U_P gives the finite difference formula

$$U_P = \frac{1}{4}\left[\left(1+\frac{C_x}{2}\right)U_{Q_1} + \left(1+\frac{C_y}{2}\right)U_{Q_2} + \left(1-\frac{C_x}{2}\right)U_{Q_3} + \left(1-\frac{C_y}{2}\right)U_{Q_4}\right] \tag{50}$$

Appendix C
Optimization Analysis

In this section, the analysis of the minimization performed by the electrostatic model is reviewed. The analysis uses the calculus of variations [3, 4, 5, 7]. The partial differential equation for the electrostatic solution to the nonuniformly conductive media (Equation 5) is derivable using the calculus of variations. The derivation shows the corresponding minimization function for the electrostatic model.

The objective of the calculus of variations is to find the extremal surface $u(x,y)$ which minimizes the variational integral

$$\iint I(u,x,y,u_x,u_y)\,dx\,dy \tag{51}$$

This is satisfied when the Euler Equation is found:

$$\frac{\partial I}{\partial u} - \frac{\partial}{\partial x}\frac{\partial I}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial I}{\partial u_y} = 0 \tag{52}$$

This gives a second order partial differential equation in the cases of interest.

To apply the calculus of variations to the electrostatic model, set the function to minimize to

$$I = \boldsymbol{j}\cdot\boldsymbol{E} \tag{53}$$

This right side is by definition

$$\boldsymbol{j}\cdot\boldsymbol{E} = \sigma(x,y)(\phi_x^2+\phi_y^2) \tag{54}$$

so the function is

$$I(\phi,x,y,\phi_x,\phi_y) = \sigma(x,y)(\phi_x^2+\phi_y^2) \tag{55}$$

From this definition

$$\frac{\partial I}{\partial \phi} = 0 \tag{56}$$

$$\frac{\partial I}{\partial \phi_x} = 2\sigma\phi_x \tag{57}$$

$$\frac{\partial I}{\partial \phi_y} = 2\sigma\phi_y \tag{58}$$

$$\frac{\partial}{\partial x}\left(\frac{\partial I}{\partial \phi_x}\right) = 2\sigma_x\phi_x + 2\sigma\phi_{xx} \tag{59}$$

$$\frac{\partial}{\partial y}\left(\frac{\partial I}{\partial \phi_y}\right) = 2\sigma_y\phi_y + 2\sigma\phi_{yy} \tag{60}$$

The Euler Equation (Equation 52) for two dimensions is

$$\frac{\partial I}{\partial \phi} - \frac{\partial}{\partial x}\left(\frac{\partial I}{\partial \phi_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial I}{\partial \phi_y}\right) = 0 \tag{61}$$

Substitution of Equation 56, Equation 59, and Equation 60 into Equation 61 yields

$$-2\sigma_x\phi_x - 2\sigma\phi_{xx} - 2\sigma_y\phi_y - 2\sigma\phi_{yy} = 0$$

Dividing through by -2σ gives the final result:

$$\phi_{xx} + \phi_{yy} + \frac{1}{\sigma}\sigma_x\phi_x + \frac{1}{\sigma}\sigma_y\phi_y = 0 \tag{62}$$

This shows that the partial differential equation (Equation 62) minimizes the energy loss due to heat loss ($\boldsymbol{j} \cdot \boldsymbol{E}$) in a nonuniform conducting medium[8, 20].