The Pennsylvania State University

Electrical Engineering Department

University Park, Pennsylvania 16802

DIPOLE AND QUADRUPOLE SYNTHESIS OF ELECTRIC POTENTIAL FIELDS

Technical Report

NASA Grant No. NSG-3166

David G. Tilley

July 1979

The Pennsylvania State University

The Graduate School

Department of Electrical Engineering

Dipole and Quadrupole Synthesis of Electric Potential Fields

A Thesis in

Electrical Engineering

by

David George Tilley

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 1979

ABSTRACT

The accumulation of static charge on the surface of spacecraft can produce unknown potential fields that may cause error in the measurement of scientific data. Therefore methods for determining potential fields are being sought. This report describes a general technique for expanding an unknown potential field in terms of a linear summation of weighted dipole or quadrupole fields. Laplace's equation describes the unknown potential in regions that are devoid of electric charge when the nearby surface charge distribution is assumed to be independent of time. Therefore, general solutions to Laplace's equation can be generated by summing multipole fields (i.e., particular solutions to Laplace's equation) as long as the multipole locations where the multipole fields have singularities are not included in the spatial region of interest. The general solution approximating an unknown potential near a charged surface can be developed if suitable boundary conditions are available or if measurements of charged particle trajectories in the unknown potential can be obtained.

The classical boundary value problem in electrostatics can be described as finding the solution to Laplace's equation when the electric potential is known on a boundary enclosing a spatial domain of interest. In this report, computational methods are developed for the iterative addition of dipole fields until solutions to the classical boundary value problem can be obtained. Various solution potentials are compared inside the boundary with a more precise calculation of the potential to derive optimal schemes for locating the singularities of the dipole fields (e.g., dipoles should not be placed

too close to the boundary). Then, the problem of determining solutions to Laplace's equation on an unbounded domain as constrained by pertinent electron trajectory data is considered. The initial electron coordinates and velocities, as well as the final electron coordinates, comprise a set of constraints on the various schemes that are developed for dipole and quadrupole synthesis of approximations to the test potential. The various schemes are then compared in terms of convergence limits and rates and in terms of their accuracy on a finite domain. The report is concluded with a description of an electron-beam apparatus used for making trajectory measurements in a bell jar. The best computational schemes are used to synthesize approximations of a simulated test potential created in the bell jar.

TABLE OF CONTENTS

LIST OF TABLES

Page

Table

LIST OF FIGURES

ACKNOWLEDGMENTS

The author wishes to express his appreciation for the guidance and sense of direction provided by his thesis advisor, James W. Robinson. Dr. Robinson has also provided the motivation and inspiration required for the completion of this research report. The author also appreciates the efforts of William S. Adams, who originally introduced the author to this research opportunity. In addition, the computer time provided by Dr. Adams as Director of The Hybrid Computer Laboratory has helped defray the expense of this research.

The author also wishes to recognize the laboratory assistance of Michael Thompson and John Mitchell, who built the experimental apparatus and who are responsible for the data which has been collected. Especially, the efforts of Susan I. Wian, in the preparation of a typed copy of this thesis, have been invaluable and are greatly appreciated.

The research described in this report has been conducted in The Department of Electrical Engineering at The Pennsylvania State University and has been sponsored by The National Aeronautics and Space Administration, grant NSG-3166.

I. INTRODUCTION

The accumulation of electrostatic charge on the surface of a spacecraft produces an electric potential field in nearby spatial regions. Laplace's equation, $\nabla^2\psi = 0$, defines a class of functions which may represent this potential field everywhere outside the spacecraft (i.e., spatial regions having little charge). If the surface charge is known, the potential field ψ can be determined by an integration over the spacecraft of the field due to a point charge weighted with the known surface charge distribution. However, for many applications the surface charge is unknown and must be determined by solving Laplace's equation constrained by the pertinent boundary conditions that may be known. The potential fields of the ideal dipole, quadrupole, octapole, etc. are solutions of Laplace's equations in regions bounded away from the sites of these ideal poles. Therefore, the unknown potential ψ can be synthesized as a linear sum of multipole potential fields, with all point multipoles located outside the region of interest (ref. 1).

The classical boundary value problem may be considered for Laplace's equation in terms of a potential ϕ unknown in region R, but known at a finite number of points on a simple closed curve containing the region R. In two dimensions, the method of finite differences could be employed to solve for the unknown potential ϕ at the nodes of a grid covering region R subject to the known boundary values of the potential ψ (e.g., the specification of the potential at 4N points on a square boundary determines an approximation of the potential at N^2 points on the interior of the square boundary). Consider instead,

an iterative solution technique. Suppose some potential ϕ exists as an estimate of ψ. Then a measure of the discrepancy D between ψ and ϕ can be defined as the sum of the absolute differences between ψ and ϕ at the finite number of nodes on the boundary where ψ is known. The estimated potential ϕ can be said to be converging toward ψ if the discrepancy D is reduced upon successive applications of the iterative scheme which may include operations of rotation, shifting or scaling. However, discrepancy reduction and the shaping of the potential approximation will be primarily accomplished by the summation of multipole fields successively centered at points outside the boundary near locations where the discrepancy D is largest.

The expansion of potential fields is most often expressed as the linear sum of weighted multipole fields (e.g., one each of monopole, dipole, quadrupole, etc.) with all pole locations superimposed. This practice produces only one singularity in the spatial solution space. However, we have adopted the practice of allowing multipole locations along a line parallel to the surface and at a specified distance from it (i.e., usually all alike dipoles or quadrupoles). Thus, field shaping is facilitated but many field singularities are produced.

When considering the potential field existing near a charged surface, the potential may be unknown in an unbounded spatial region. Due to insufficient boundary conditions, additional constraints must be specified in order to approximate the potential on a finite spatial region. Actually, in mathematical terms, the region is bounded at infinity where the potential must equal zero. In order to obtain measurements of potential fields existing near charged surfaces,

consider an electron beam emanating from a source point (XO,YO) on a zero potential reference plane. If the reference plane were located far from the charged surface, it would approximate the boundary condition at infinity. Practically speaking, the reference plane can be moved to a finite and measurable distance from the charged surface while producing only minimal distortions of the potential field. If the spacecraft surface is negatively charged the electron beam will be deflected back toward the reference plane where the coordinates of a sink point (XT,YT) can be recorded. As seen in Figure 1, a displacement vector measured from the beam source point to the beam sink point represents one constraint on the family of functions that satisfy Laplace's equation. Suppose that a potential field, partially bounded by a charged surface and the reference plane, is to be determined based on measurements of many such displacement vectors associated with electron beams with known energies and initial directions. Suppose further that some estimate ϕ of the potential ψ exists on the region of interest. Estimated displacement vectors can be obtained from the initial electron beam data and the estimated potential by programming the laws of electron mechanics. These estimated displacements can be compared to those measured as data to define the error related to the electron trajectory constraints. A discrepancy DD can be defined as the sum of the absolute differences between the estimated displacement vectors and the measured displacement vectors. The estimated potential ϕ is said to be converging toward ψ if the discrepancy is reduced upon successive applications of some iterative scheme which may include scaling or rotating operations. However, major alterations in the estimated potential can be accomplished by adding the potential fields

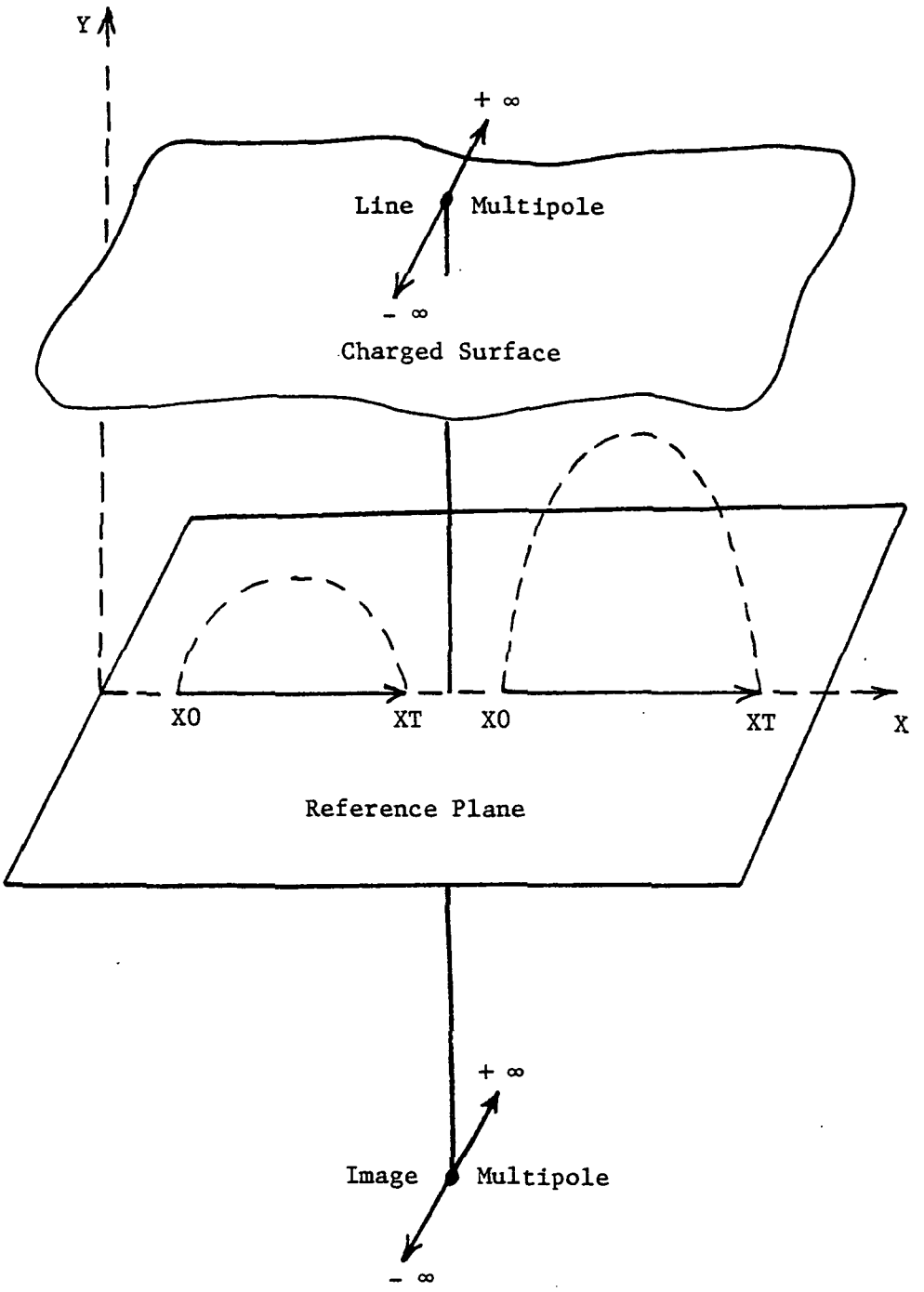


Figure 1. The Orientation of Ideal Line Multipoles with Respect to a Charged Surface and Reference Plane

of pairs of ideal multipoles. One multipole can be located above the charged surface at a given distance from the reference plane, while another multipole is located an equal distance below the reference plane. This imaging technique maintains a potential of zero on the reference plane while allowing for the shaping of the estimated potential in the region between the reference plane and the charged surface.

The nature and substance of the iterative schemes referred to in the above discussion are defined depending arbitrarily on the data that are available and the ingenuity of the programmer. A particular iterative scheme may have different convergence properties when applied to a number of different problems. A particular iterative scheme may have better convergence properties than other schemes when applied to a common problem. Furthermore, the selection of data defining a given problem may influence the convergence of a particular iterative scheme and the selection of an initial approximation may also influence the convergence toward a potential solution. Therefore, there are a broad range of topics that can be investigated and discussed relative to the synthesis of electric potentials by the iterative addition of ideal multipole fields. For the purposes of this report, the effects of boundary geometry, data selection and initial potential estimation will not be considered. Attention will be focused on the development of optimal iterative schemes based upon a well-defined model problem.

II. THE BOUNDARY VALUE PROBLEM

The classical boundary value problem in electrostatics may be stated as the problem of calculating values of a potential field existing on a spatial region devoid of electric charge (i.e., where Laplace's equation is valid) when given the value of the potential on a simple boundary enclosing the region of interest. When the potential is specified as a continuous function on the boundary, the problem can be solved by a separation of variables technique and the expansion of the potential in terms of a complete set of orthogonal eigenfunctions (ref. 2). The geometric shape of the boundary influences the choice of an appropriate coordinate system and the particular set of eigenfunctions used for the expansion of the potential field. The solution is continuous inside the boundary and the error depends on the number of terms used in the eigenfunction expansion.

In a practical situation, the value of the potential may be known only at a finite number of points on a boundary. The separation of variables and eigenfunction expansion method can be applied to obtain a continuous solution, if some approximation is used to interpolate a continuous functional value of the potential on the boundary from the finite number of known boundary values. The error of the solution depends on the approximation of the potential on the boundary as well as the number of terms in the expansion of the potential. The method of finite differences may also be used to approximate the potential inside a closed boundary when only a finite number of boundary values are known (ref. 3). The solution is developed by using Taylor series expansions to derive a finite number of linear potential equations in

terms of an equal number of unknowns (e.g., the potential values at all points on the grid enclosed by the boundary) and a finite number of knowns (e.g., the potential values at all points on the grid coinciding with the boundary). A solution may then be obtained by using Gaussian elimination procedures to solve the system of linear equations. A finite difference solution is not continuous since it is defined only at nodal points on the grid. Therefore, interpolation errors will exist in addition to Taylor approximation errors, both of which depend on the spacing of the grid lines. Boundaries with irregular shapes will be problematic; since either a very fine grid will be required, leading to lengthy computer calculations, or some boundary approximations will be required to accommodate a larger grid spacing.

A. Dipole Synthesis of Potential Fields

Consider an iterative scheme for summing the potential fields of ideal electric dipoles to develop solutions to Laplace's equation on a domain enclosed by a boundary where the potential is known at a finite number of points (i.e., Dirichlet boundary condition). An initial estimate of the potential satisfying Laplace's equation is an average of the known boundary values plus any linear terms suggested by the boundary values (e.g., $\phi_o(x,y) = a + bx + cy$). This estimated potential can be improved by the iterative addition of an ideal dipole outside the boundary and nearest the location exhibiting the largest discrepancy between estimated and known boundary values. The potential field of an ideal dipole is itself a solution to Laplace's equation. Hence, the linear sum of the fields due to many such electric dipoles is also a solution to Laplace's equation. We will assume to have synthesized a

particular solution to the boundary value problem when the sum of all the absolute discrepancies between estimated and known boundary potential values is reduced below some tolerable limit. Such a solution is not unique, since the choice of dipole locations is arbitrary. If the above procedure for the reduction of discrepancy is pursued indefinitely, ultimately one dipole of optimal magnitude will be located outside the boundary near each one of the points where a boundary value is given. Thus, the iterative scheme expands the potential as a linear sum of the fields associated with dipoles located outside the boundary. If the dipoles are allowed infinitely close to the boundary points, then each dipole magnitude will be chosen to reduce the discrepancy existing at one boundary location and have no effect on the discrepancies at the other boundary locations. Solutions developed in this manner are unacceptable since the field singularities introduced near the boundary produce only localized reduction of discrepancy. Consequently, the continuity of the potential solution on the boundary is jeopardized by placing dipoles too near the boundary itself. Therefore, we shall expect to synthesize solutions to the boundary value problem, if the dipoles are located well outside the bounded domain of interest. Convergence (e.g., discrepancy reduction) will be slower as dipoles are moved further from the boundary, but better approximations will result on the boundary between the pointwise specified values. Irregularly shaped boundaries can be easily accomodated by the iterative procedure for dipole synthesis. In addition, the estimated potential is defined in a continuous manner as the sum of dipole fields at all points on the bounded domain and no interpolation from a coordinate grid is required.

To define a specific class of boundary value problems, assume that the boundary S of region A in the two-dimensional r-θ plane can be represented by N points distributed around the origin with no two points having the same θ coordinate value. Assume also that the potential $\psi(r,\theta)$ satisfies Laplace's equation (i.e., $\nabla^2\psi = 0$) and is given at the same N points on the boundary

$$\psi(R_n,\theta_n) = \psi_n \; ; \; n = 1,2,3\cdots N \qquad \text{(II-1)}$$

In order to allow calculations of the potential on the interior region A, as well as the boundary S; construct a 2N x N coordinate grid with its origin at the geometric center or centroid of region A and including the N points of equation (II-1). Define the $2N^2$ points (r_m, θ_n) letting $r_m = R_m$ for $m = 1,2,3,\cdots N$ and $r_m = \frac{(m-N)R}{N}$ for $m = N+1, N+2,\cdots 2N$; where R is the radius of the smallest circle having its center at the origin of the coordinate grid which also encloses the region of interest.

In order to begin an iterative procedure for approximating the potential at points on the interior, given the potential at N points on the boundary, define an initial estimated potential ϕ_o existing as an approximation to ψ.

$$\phi_o(r_m,\theta_n) = 1/2[\psi_n^{max} - \psi_n^{min}] + 1/2[\psi_n^{max} - \psi_n^{min}]\, r_m \cos\theta_n \qquad \text{(II-2)}$$

This choice is arbitrarily based on the maximum and minimum values of the N known boundary potentials; it satisfies Laplace's equation (i.e., $\nabla^2\phi_o = 0$) and it guarantees that the difference between ψ and ϕ_o on the boundary is no larger than ψ_n^{max} minus ψ_n^{min}.

As a general procedure for improving the accuracy of the estimated potential on the boundary S, consider the iterative addition of dipole

potential fields $\phi_d(r_m,\theta_n)$ due to the ideal dipoles located at (r_d,θ_d) outside of the boundary S enclosing region A.

$$\phi_E(r_m,\theta_n) = \phi_o(r_m,\theta_n) + \sum_{d=1}^{E} \phi_d(r_m,\theta_n) \qquad \text{(II-3)}$$

The following algorithm is used for calculating in two dimensions (i.e., the r-θ coordinate system) the potential fields associated with ideal electric dipoles at (r_d,θ_d),

$$\phi_d(r_m,\theta_n) = \frac{A_d[r_d - r_m \cos(\theta_n-\theta_d)]}{r_d^2 + r_m^2 - 2r_d r_m \cos(\theta_n-\theta_d)}, \qquad \text{(II-4)}$$

as derived in Appendix A. We will assume that the estimated potential is a good approximation to ψ on the interior of region A, as well as on the boundary S, when a sufficient number of dipole potentials ϕ_d have been added to ϕ_o such that ϕ_E matches the boundary potential ψ_n at all points where the latter quantity is given.

$$\phi_E(R_n,\theta_n) = \psi(R_n,\theta_n) \;;\; n = 1,2,\cdots N \qquad \text{(II-5)}$$

The arbitrary constant $A_d = q\Delta r$ represents the charge magnitude q of an ideal dipole having separation Δr as measured along a radius extending from the centroid of region A to the location (r_d,θ_d) of the dipole.

For the purpose of calculating the error of the approximation of ψ by ϕ_E, define the discrepancy function D_n as the difference between ψ and ϕ_E at each of the N boundary locations where ψ is known.

$$D_n \equiv \psi(R_n,\theta_n) - \phi_E(R_n,\theta_n) \qquad \text{(II-6)}$$

The average discrepancy and average absolute discrepancy, as defined by equations (II-7) and (II-8), can be used as measures of the accuracy

of the estimated potential.

$$\bar{D} = \frac{1}{N} \sum_{n=1}^{N} D_n \tag{II-7}$$

$$\overline{|D|} = \frac{1}{N} \sum_{n=1}^{N} |D_n| \tag{II-8}$$

We are interested in developing iterative procedures for reducing $\bar{D}$ and $\overline{|D|}$. A uniform shift of the estimated potential can be used to reduce $\bar{D}$ to zero. Suppose the average discrepancy $\bar{D}$ exists for estimated potential ϕ_E. If a new estimated potential ϕ_E' is formed by adding the average discrepancy $\bar{D}$ to ϕ_E, then we wish to show that the average discrepancy $\bar{D}'$ for ϕ_E' is equal to zero.

$$\bar{D}' = \frac{1}{N} \sum_{n=1}^{N} D_n' = \frac{1}{N} \sum_{n=1}^{N} (\psi_n - \phi_n')$$

$$= \frac{1}{N} \sum_{n=1}^{N} (\psi_n - (\phi_n + \bar{D})) \tag{II-9}$$

Then by rearranging terms and recalling the definition of $\bar{D}$ expressed by equation (II-7), we derive the desired result.

$$\bar{D}' = \frac{1}{N} \sum_{n=1}^{N} (\psi_n - \phi_n) - \frac{1}{N} \sum_{n=1}^{N} \bar{D} = \bar{D} - \bar{D} = 0 \tag{II-10}$$

Thus, the estimated potential can be shifted so that its average discrepancy with known boundary values is reduced to zero.

The reduction of the average absolute discrepancy is not as easily accomplished. Consider the rotation of the estimated potential, with respect to the coordinate grid, as one operation for reducing $\overline{|D|}$ on the boundary. Define N rotated potentials $\phi_R(r_m, \theta_n)$; for $R = 1,2,\cdots N$ such that

$$\phi_R(r_m,\theta_n) = \phi(r_m,\theta_{n+R}) \text{ for } n = 1,2,\cdots N-R \qquad \text{(II-11)}$$

$$\phi_R(r_m,\theta_n) = \phi(r_m,\theta_{n+R-N}) \text{ for } n = N-R+1,\cdots N \qquad \text{(II-12)}$$

for all $m = 1,2,\cdots 2N$. Then one might choose, as the best estimated potential, the rotated potential yielding the lowest average absolute discrepancy and zero average discrepancy.

The average absolute discrepancy might also be reduced by scaling the estimated potential after the rotating and shifting operations. The following algorithm is proposed for scaling the estimated potential so as to reduce the average absolute discrepancy whenever the average discrepancy is zero.

$$\phi(r_m,\theta_n) = \bar{\phi} + \frac{\sum_{n=1}^{N} |\psi(R_n,\theta_n) - \bar{\psi}|}{\sum_{n=1}^{N} |\phi(R_n,\theta_n) - \bar{\phi}|} [\phi(r_m,\theta_n) - \bar{\phi}] \qquad \text{(II-13)}$$

$$\bar{\phi} = \frac{1}{N} \sum_{n=1}^{N} \phi(R_n,\theta_n) \qquad \text{(II-14)}$$

$$\bar{\psi} = \frac{1}{N} \sum_{n=1}^{N} \psi(R_n,\theta_n) \qquad \text{(II-15)}$$

After the rotation, shifting and scaling operations, the addition of ideal dipole potentials can be used to further reduce $\overline{|D|}$ and improve the approximation of ψ by ϕ. The definition of a specific iterative scheme depends on the general criteria that can be established for determining optimal locations and magnitudes for each successive electric dipole that is added near the boundary enclosing the region of interest.

B. Iterative Schemes for the Dipole Synthesis of Solutions to the Boundary Value Problem

VMODGR.F4 is the name of the main Fortran routine used for solving boundary value problems. Initially, subroutine DATA is called to set up the problem in the X-Y coordinate system corresponding to the user's particular reference frame. The subroutine asks the user for up to forty points representing the boundary and for the potential value at each of the points which must be specified in a clockwise sequence around any plane boundary. The main routine generates an r-θ coordinate grid with origin at the centroid of the region enclosed by the boundary.

The determination of a lower bound on the distance between the boundary and the dipole locations is the primary objective of the study. It is assumed that all dipoles will be best aligned if their axes, extending in the direction of charge separation, pass through the centroid of the bounded domain. Subroutine ADPOLE locates a dipole at the angular coordinate θ_d where the discrepancy D is greatest between the estimated potential ϕ and the boundary potential ψ. Then a parameter G is defined in subroutine ADPOLE so that all dipoles are located a fixed radial distance from the centroid. This distance

$$r_d = R(1+G/N) \qquad \text{(II-16)}$$

is calculated in terms of the number N of known boundary potentials and the largest radius R needed to construct a circle centered at the centroid and enclosing all of the region of interest. Dipoles will be allowed closer to the boundary as the value of the G parameter, specified by the user interactively with subroutine DATA, is allowed to decrease toward a limiting value of zero. Once a dipole location has

been determined, subroutine ADPOLE choses a dipole magnitude A_d that reduces the discrepancy at the boundary location nearest the dipole to approximately zero. Alternatively, subroutine TADPOLE may be called by the main routine to select dipole locations and magnitudes. In this subroutine, dipoles are also located at the angular coordinate θ_d corresponding to the greatest boundary discrepancy D_1 and the dipole axis is aligned with the centroid of the region of interest. However, for the TADPOLE scheme, the G parameter does not fix the radial coordinate for dipole location but specifies a minimum value that can be allowed. Whenever possible, a dipole magnitude A_d and dipole radius r_d are chosen simultaneously so that

$$\phi_d(R_d,\theta_d) = -D_1 \qquad \text{(II-17)}$$

and

$$\phi_d(R_c,\theta_c) = -\frac{D_1}{|D_1|}\, T\cdot|D|, \qquad \text{(II-18)}$$

where the coordinates (R_c,θ_c) represent the point on the boundary nearest the point (R_d,θ_d) where the discrepancy has a sign opposite from the sign of the largest discrepancy D_1. Whenever, the criterion expressed in equations (II-17) and (II-18) cannot be achieved without placing the dipole inside the minimum radius, as expressed by equation (II-16), r_d is chosen equal to this minimum radius on the basis of the G parameter specified in subroutine DATA. Thus, the T parameter, also entered interactively with subroutine DATA, balances discrepancy reduction at the boundary point where the discrepancy is maximum with the accompaning discrepancy increase at a nearby boundary location.

The main routine VMODGR.F4 first defines an initial estimated potential according to equation (II-2) and then calls subroutine

ROTATE to achieve discrepancy reduction by the rotation of the initial estimated potential relative to the coordinate grid. An iterative scheme, calling subroutines SHIFT, SCALE and either ADPOLE or TADPOLE; is then applied to accomplish the reduction of boundary discrepancies. Subroutines SHIFT and SCALE, listed in Appendix B, accomplish discrepancy reduction according to equations (II-10) and (II-13) respectively. The other Fortran routines described in this section are also listed in Appendix B.

The particular problem of a square boundary with a potential of 10 volts specified on three sides and 0 volts specified on the fourth side has been chosen to demonstrate the method of dipole synthesis and to investigate various criteria for choosing dipole locations in an iterative scheme. Figure 2 shows how twelve boundary points were selected at uniform angular intervals and how one interior point at ($r = .471$, $\theta = 90°$) was selected for comparing synthesized potentials with a precise series expansion which yields a value of 9.59 volts just inside the unit square boundary. Tables 1 and 2 show the potential values synthesized at the test point with the main routine VMODGR using subroutines ADPOLE and TADPOLE, respectively. The fields of ideal dipoles were added at each boundary point and at the test point in an iterative process, one dipole per iteration, until the sum of the absolute discrepancies was reduced below 10^{-5} volts. Note, in Table 1, that the estimated potential converges to its precise value at the test point as the dipoles are moved further from the boundary. However, the number of iterative computer calculations required is large. When dipoles are allowed closer to the boundary (i.e., smaller G parameter), convergence is much faster but errors in the estimated potential at the

Table 1

Potential Synthesis With Constant Dipole Radius

G	Iterations	Test Potential (volts)
1	65	10.42
2	73	10.08
4	109	9.83
2π	157	9.67
10	354	9.61
15	791	9.59
20	1718	9.59

Table 2

Potential Synthesis With Variable Dipole Radius, $G = 2\pi$

T	Iterations	Test Potential (volts)
0.01	157	9.67
0.30	157	9.67
0.50	161	9.64
0.70	169	9.59
0.90	214	9.59

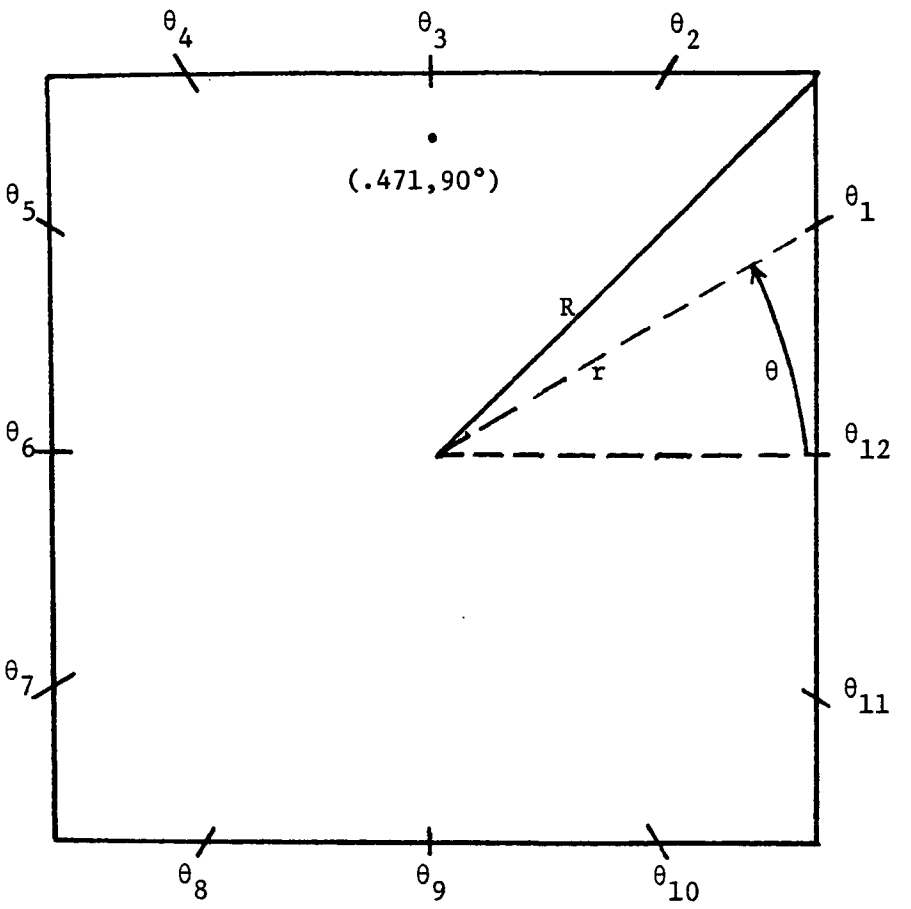


Figure 2. Polar Coordinate Frame of
Reference for The Boundary Value Problem

test point are substantial. Note, in Table 2, that the G parameter is set equal to 2π thereby establishing the minimum radial coordinate available for the location of point dipoles. The convergence of the estimated potential to the specified boundary values is fastest when the T parameter forces many dipoles close to the boundary (i.e., smaller values of T). The occasional placement of dipoles at greater distances from the boundary, using both the G and T parameters, provides a flexibility that allows compromise between the speed of the convergence and the accuracy of the estimated potential at interior locations. Thus, convergence with the required accuracy at the test point can be achieved with 791 iterations using the G factor alone and with 169 iterations using the G and T factors together.

Once convergence is achieved, VMODGR calls subroutine VGRAPH to plot equipotential lines as determined on the interior coordinate grid. Figures 3 and 4, corresponding to G parameter values of 2 and 2π respectively, show solutions to the boundary value problem for 20 points on the square boundary spaced at equal angular intervals. Figure 5 shows the potential solution for a G parameter of 2π and a T parameter value of 0.1. A comparison of Figures 4 and 5 reveals no major differences between the potential solutions generated by the two iterative schemes. However, the solution shown in Figure 5 was generated by using subroutine TADPOLE and required less than half the computer time needed when subroutine ADPOLE was used. Figure 6 shows the improvement in the solution potential resulting from the specification of 36 boundary values as compared to only 20 boundary values. It should be noted that the grid structure used inside the boundary results in linear interpolation errors in the drawing of equipotential lines. These

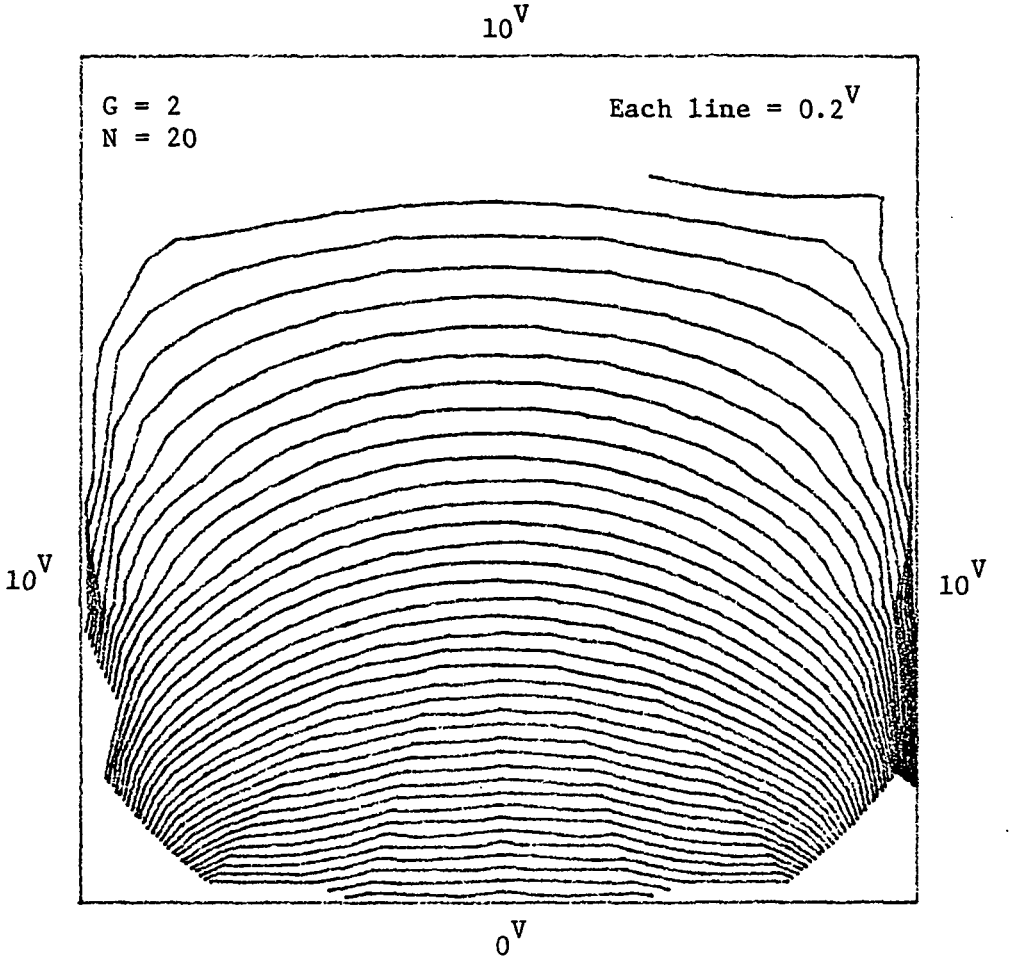


Figure 3. Equipotential Lines for Dipole Synthesis with G=2 and 20 angular grid lines

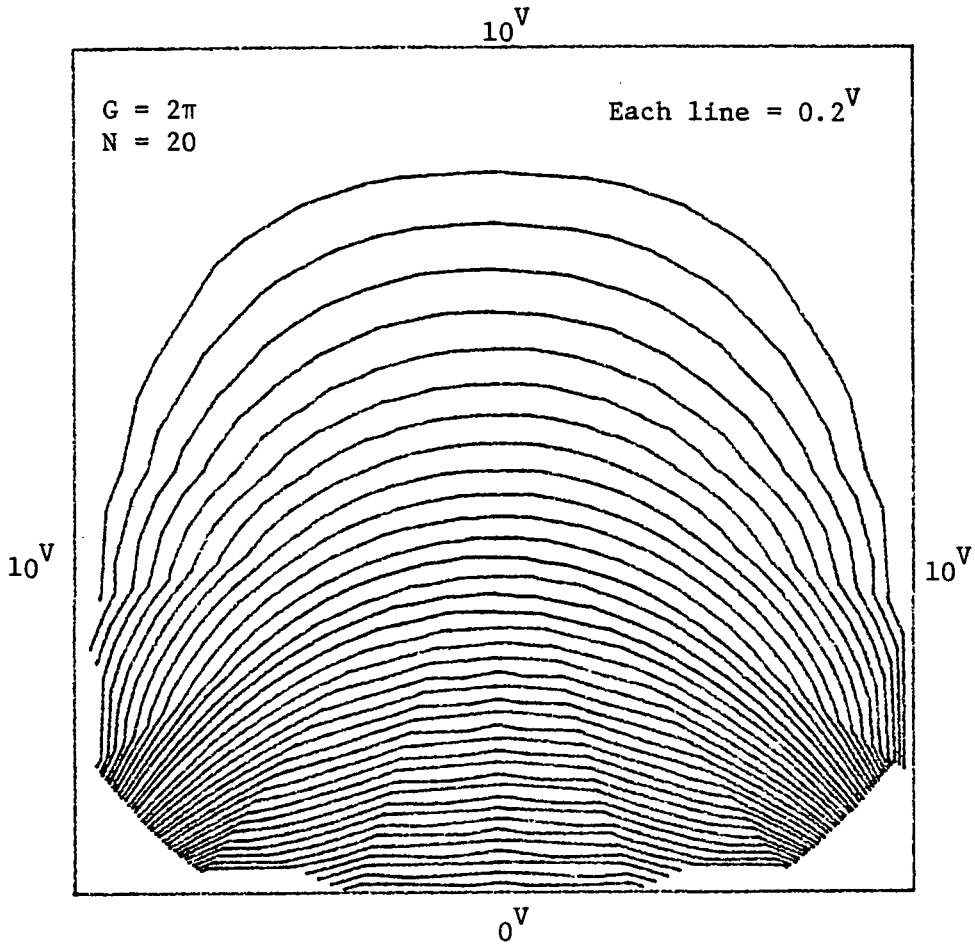


Figure 4. Equipotential Lines for Dipole Synthesis with G=2π and 20 angular grid lines

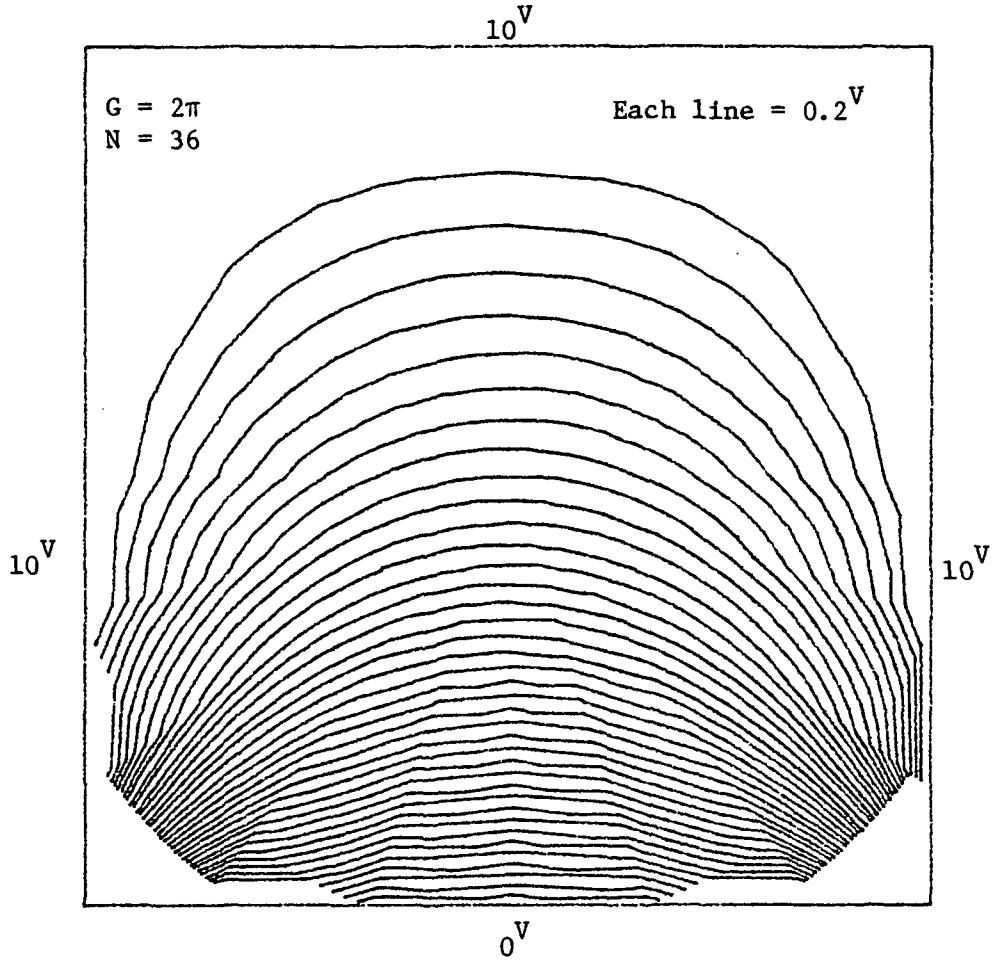


Figure 5. Equipotential Lines for Dipole Synthesis with G=2π, T=0.1 and 20 angular grid lines

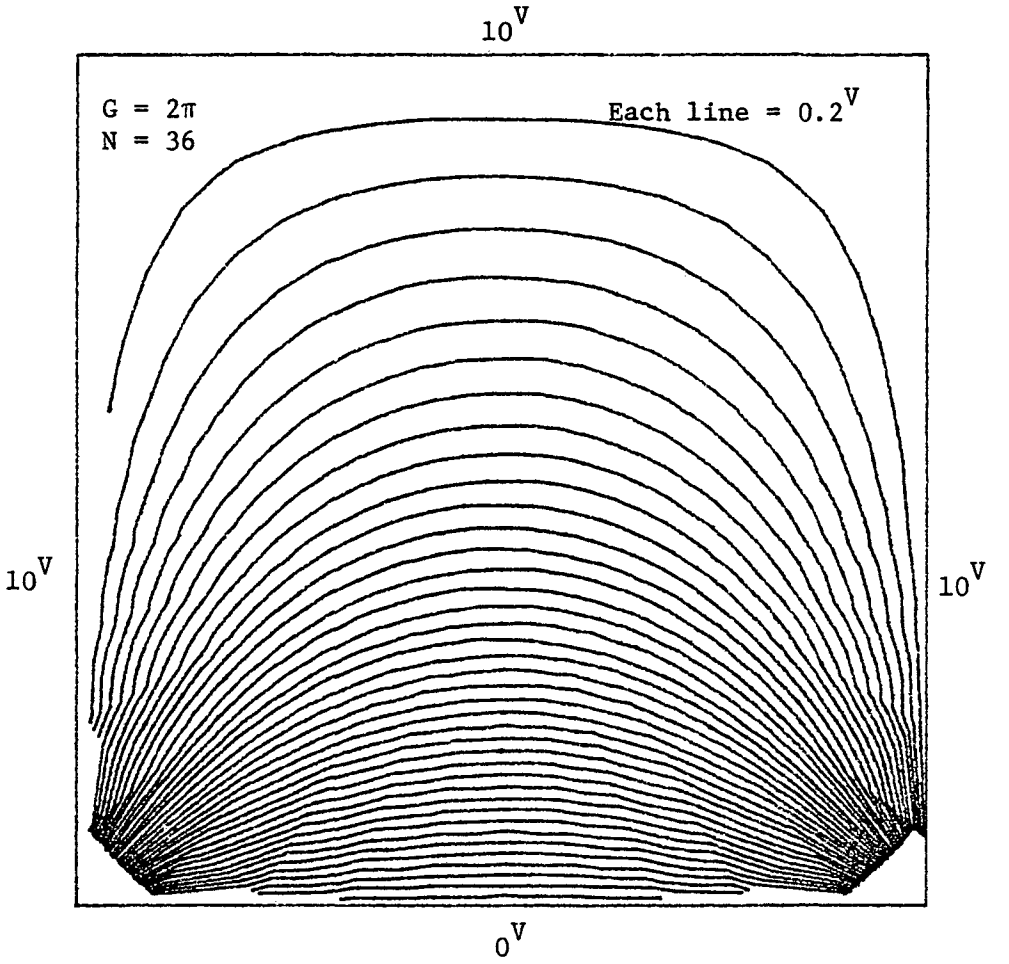


Figure 6. Equipotential Lines for Dipole Synthesis with G=2π, T=0.1 and 36 angular grid lines

errors are not a consequence of the procedure for potential field synthesis but result when an attempt is made to display a potential field based upon its value at a finite number of points. The potential display could be improved by a better interpolation procedure or by calculating the potential at more points. However, the accuracy of the potential estimate synthesized at a particular point of interest does not depend on the number of interior points considered. Therefore, potential estimates could be synthesized at as many, or as few, interior points as desired with accuracy depending on the number of boundary values specified.

III. THE DEFINITION OF A MODEL PROBLEM

The electric potential resulting from a distribution of surface charge can be shown to satisfy Laplace's equation everywhere except on the surface itself. Since the potential fields associated with ideal multipoles are also solutions of Laplace's equation, we assume that a linear sum of weighted multipole potentials can be used to construct a particular solution that approximates the potential field created by the surface charge distribution. This particular solution will have singularities at the spatial locations of the multipoles. Therefore, the spatial domain near a charged surface can be partitioned into two subspaces; one subspace defined as the region where a valid potential approximation is desired and another subspace reserved for the location of ideal multipoles. In the boundary value problem considered previously, the closed boundary separated the interior subspace (where solutions were desired) from the exterior subspace (where dipoles were located). In a general situation, the charged surface may only partially define the subspace where multipole location is allowed. In any event, an iterative scheme for summing multipole potentials to approximate the potential near a charged surface will depend on the geometry of the particular problem under consideration.

Furthermore, in contrast to the boundary value problem, the surface charge and surface potential may be unknown in the general problem. Trajectory data for electrons moving in an unknown potential field created by an unknown charge distribution can be used as constraints on the class of functions that approximate the potential field (ref. 4). Electron deflection measurements can be defined, in the absence of

boundary values, as a basis for calculating the discrepancy between the potential in the region probed by the electron trajectories and the multipole potential approximation generated by an iterative procedure. In such a case, the electron trajectories define a subspace where the potential approximation is desired and where ideal multipoles cannot be located. Therefore, a particular iterative scheme for determining the optimal locations of dipoles, quadrupoles or multipoles will be influenced by the electron trajectory data provided as part of a general problem.

To define a particular problem that can be used to compare the convergence properties of various iterative procedures, consider the potential field created by two uniformly biased fins in the vicinity of a zero potential ground plane. So that calculations may be performed in a two-dimensional space, assume that both the ground plane and the biased fins extend infinitely in the positive and negative z coordinate directions. Note, in Figure 7, that the fins begin at x-y coordinate values $(\frac{DX}{4},DY)$ and $(\frac{3DX}{4},DY)$ respectively and extend infinitely in the positive y dimension. Suppose the x-y coordinate plane can be mapped onto the x'-y' coordinate plane by shifting the origin by amounts DY and $\frac{DX}{2}$, by scaling the axes by factors of $\frac{DX}{\pi}$ and $\frac{DY}{\pi}$, and by rotation of the axes through 90°.

$$x' = \frac{\pi}{DY}\,(DY-y) \text{ and } y' = \frac{\pi}{DX}\,(x-\frac{DX}{2}) \qquad \text{(III-1)}$$

The conformal mapping of the complex z plane ($z = x' + iy'$) onto the complex w plane ($w = u + iv$) is

$$w = \sin^{-1}(e^{z}) \qquad \text{(III-2)}$$

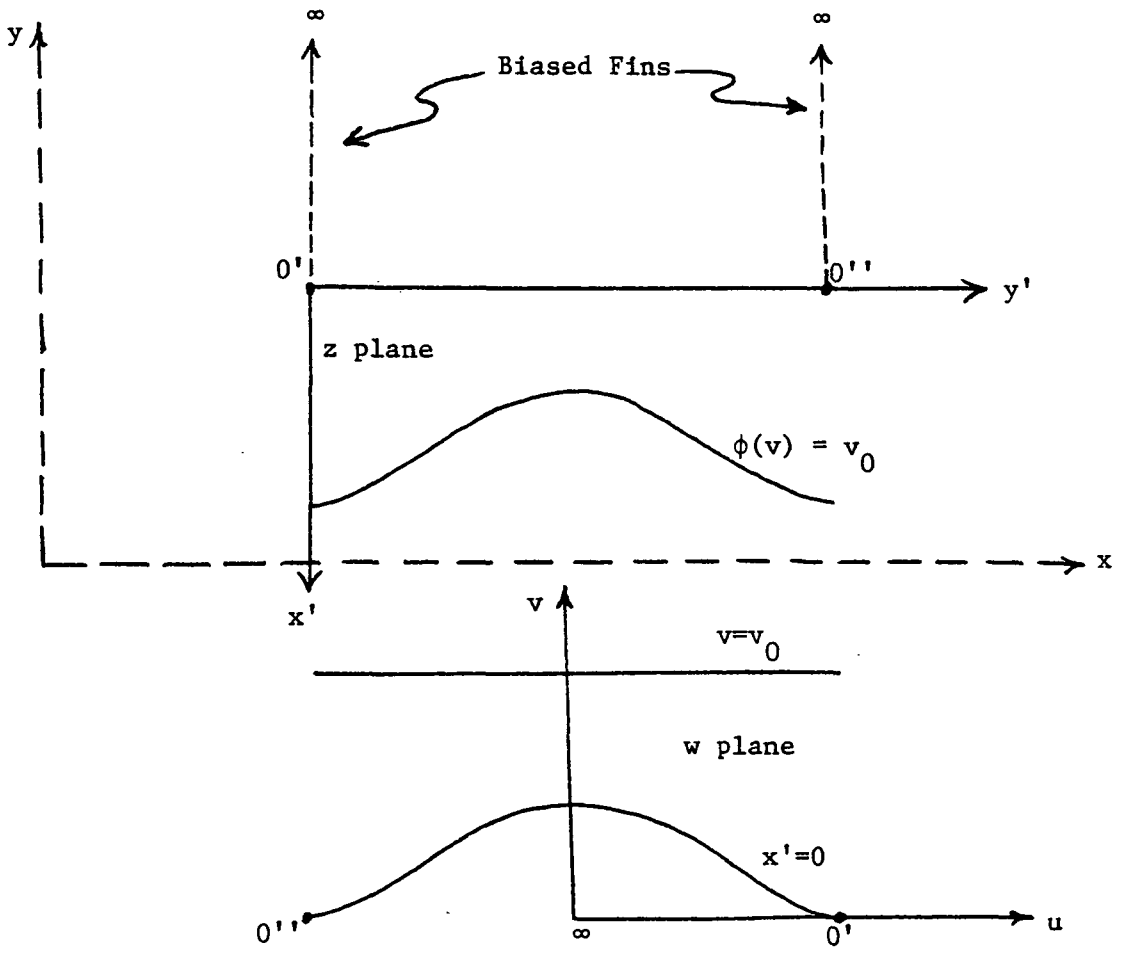


Figure 7. The Reference Frames of the Biased Fins and the Conformal Mapping $w=\sin^{-1}[\exp(z)]$

which collapses the infinite fins onto a finite interval (i.e., [0'',0']) along the u-coordinate axis (ref. 5). The value of this transformation lies in the fact that the potential field created by the biased fins in the x-y plane transforms to a potential with a uniform gradient in the u-v plane. Thus,

$$\phi(z)-C = v(z) = \mathrm{Im}[w] = \mathrm{Im}[\sin^{-1}(e^{z})] \qquad \text{(III-3)}$$

can be used with an appropriate expression for $\sin^{-1}(e^{z})$ to construct an algorithm

$$\phi(x',y')-C = \mathrm{Im}\{-i\cdot\ln[ie^{x'}\cdot e^{iy'} + (1-e^{2x'}\cdot e^{2iy'})^{1/2}]\} \qquad \text{(III-4)}$$

that can be used to calculate the biased fin potential at selected points in the x-y plane. The constant C is required so that the potential can be shifted enough to simulate the zero potential line $x' = DY$ (i.e., $y = 0$). In order to simplify the algebraic evaluation of the natural logarithm in equation (III-4), convert the complex argument of the square root to polar form.

$$r \equiv (1+e^{4x'}-2e^{2x'}\cdot\cos 2y')^{1/2} \qquad \text{(III-5)}$$

$$\theta \equiv \pi + \tan^{-1}\left[\frac{-e^{2x'}\cdot\sin 2y'}{1-e^{2x'}\cdot\cos 2y'}\right]. \qquad \text{(III-6)}$$

These expressions lead to the form

$$\phi(x',y',r,\theta) = \mathrm{Im}[-i\cdot\ln(ie^{x'}\cdot e^{iy'} + \sqrt{r}e^{i\theta/2})] + C. \qquad \text{(III-7)}$$

To further simplify the calculations, let

$$R \equiv [r+e^{2x'}-2\sqrt{r}e^{x'}\cdot\sin(y'-\theta/2)]^{1/2} \qquad \text{(III-8)}$$

and

$$\Omega \equiv \pi + \tan^{-1}\left[\frac{e^{x'}\cos y' + \sqrt{r}\sin\theta/2}{\sqrt{r}\cos\theta/2 - e^{x'}\sin y'}\right] \qquad \text{(III-9)}$$

to obtain

$$\phi(R)-C = \mathrm{Im}[-i\cdot\ln(Re^{i\Omega})] = \mathrm{Im}[-i\cdot\ln(R)\pm\Omega] = -\ln(R). \qquad \text{(III-10)}$$

Therefore, to find the potential anywhere in the x-y plane; we first calculate values of x' and y', then find the corresponding values of r and θ, and finally determine a value of R. The potential at point (x,y) is then equal to the natural logarithm of R plus an arbitrary constant.

This algorithm for calculating the potential field in the x-y coordinate space of the biased fins is implemented by main routine TEST.F4 (ref. Appendix C). When the additive constant is properly chosen, the potential falls to zero along the x axis within a tolerance of 1% of the fin potential if the spacing of the fins is not greater than the distance from the fins to the x axis by more than a factor of 1.8. Thus, the gound reference plane could be moved to within a distance DY = DX/3.6, when the spacing of the fins is DX/2, without altering the potential field of the biased fins by significant proportions. TEST.F4 calculates potential values on an 80 x 27 coordinate grid corresponding to a region DX by 1.4DY in the x-y plane. The I and J indices are used to identify points in a region with dimensions DX by DY according to

$$X(I) = (I-2)DX/80 \qquad \text{(III-11)}$$

and

$$Y(I) = (J-2)DY/20. \qquad \text{(III-12)}$$

These values are written into a storage file POTL.DAT which may be accessed by other routines that require the test potential. Table 3 lists values of the biased fin potential at 110 points within the spatial regtion of interest. TEST.F4 also includes a graphic subroutine, VPLOT, that utilizes a linear interpolation technique to plot equipotential lines in the space between the biased fins and the ground reference plane. Figure 8 shows the graphical plot generated by TEST.F4 with input parameters, DX = 3.6 and DY = 1.0.

Also consider six electron trajectories, emanating at six locations along the x axis, that probe the region between the fins before falling back to the reference plane. Assume that the initial coordinate values and the initial coordinate velocities are known as well as the final coordinate values of the six trajectories that serve as constraints placed upon approximations of the biased fin potential. In particular, let five trajectories emanate from locations (0.900,0), (1.305,0) , (1.710,0), (2.105,0) and (2.520,0) along the x axis and with an initial direction parallel to the y axis and with initial kinetic energy equal to 85% of the fin bias. Also let a scaling trajectory emanate from the location (0.855,0) on the x axis with an initial velocity in the x direction equal to 1.1 times the velocity in the y direction. The main routine DATA.F4 is used to determine the final coordinates of the six trajectories defined by the initial electron parameters and the test potential. DATA.F4 calls subroutine DVOGEL in an iterative manner to calculate successive electron steps through a test potential specified by reading file POTL.DAT. DVOGEL solves the electron's second order equations of motion according to a method proposed by the mathematician DeVogelaire (ref. 6). Electric fields

Table 3

Values of the Test Potential at 110 Selected Points

y-coordinate index (J)											
21	2.02	1.82	1.67	1.57	1.52	1.50	1.52	1.57	1.67	1.82	2.02
19	1.69	1.61	1.50	1.43	1.38	1.37	1.38	1.43	1.50	1.61	1.69
17	1.44	1.40	1.33	1.27	1.24	1.23	1.24	1.27	1.33	1.40	1.44
15	1.23	1.20	1.16	1.11	1.09	1.08	1.09	1.11	1.16	1.20	1.23
13	1.02	1.01	0.98	0.95	0.93	0.92	0.93	0.95	0.98	1.01	1.02
11	0.83	0.82	0.80	0.78	0.76	0.76	0.76	0.78	0.80	0.82	0.83
9	0.65	0.64	0.63	0.61	0.60	0.59	0.60	0.61	0.63	0.64	0.65
7	0.46	0.46	0.45	0.44	0.43	0.43	0.43	0.44	0.45	0.46	0.46
5	0.28	0.28	0.27	0.27	0.26	0.26	0.26	0.27	0.27	0.28	0.28
3	0.10	0.10	0.10	0.09	0.09	0.09	0.09	0.09	0.10	0.10	0.10
x-coordinate index (I)											
	22	26	30	34	38	42	46	50	54	58	62

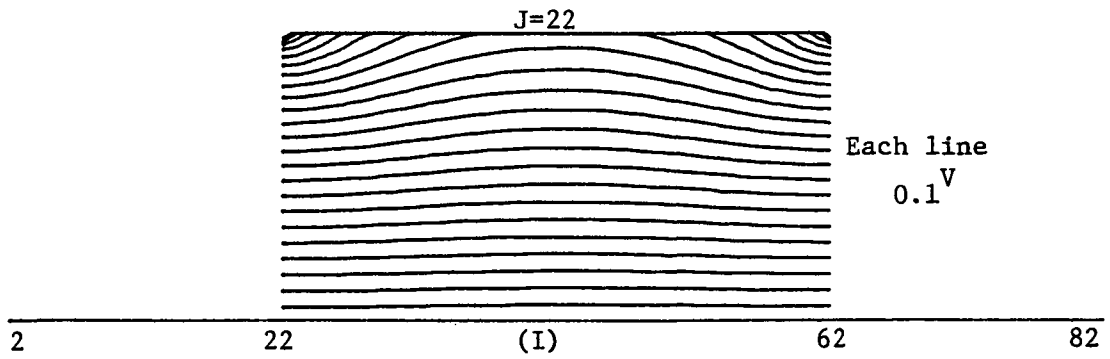


Figure 8. Equipotential Lines for the Test-Biased Fin Potential on The Domain $22 < I < 62$, $2 < J < 22$

required by subroutine DVOGEL at points between the grid lines are obtained by a call to subroutine NEWTON which implements a divided difference interpolation (ref. 7) of the electric fields based upon the grid potential values provided by POTL.DAT. The electric fields at the 16 grid points (i.e., a 4 x 4 minigrid) closest to the point where an interpolated electric field is required, are used to determine a spatial step size (not in excess of 0.01 times a typical diagonal measurement of the region of interest) used for the piecewise development of electron trajectories.

The object of routine DATA.F4 is the determination of the final coordinate values (0.901,0), (1.848,0), (1.824,0), (1.709,0) (2.086,0) and (2.691,0) for the six constraining trajectories. These final coordinate values, in addition to the initial coordinate values and velocities are written into file TRAJ.DAT which can be referenced by routines that approximate the potential field of the biased fins. DATA.F4 includes a graphics routine for plotting the constraining electron trajectories and calls subroutine VPLOT to superimpose equipotential lines representing the test potential as shown in Figure 9.

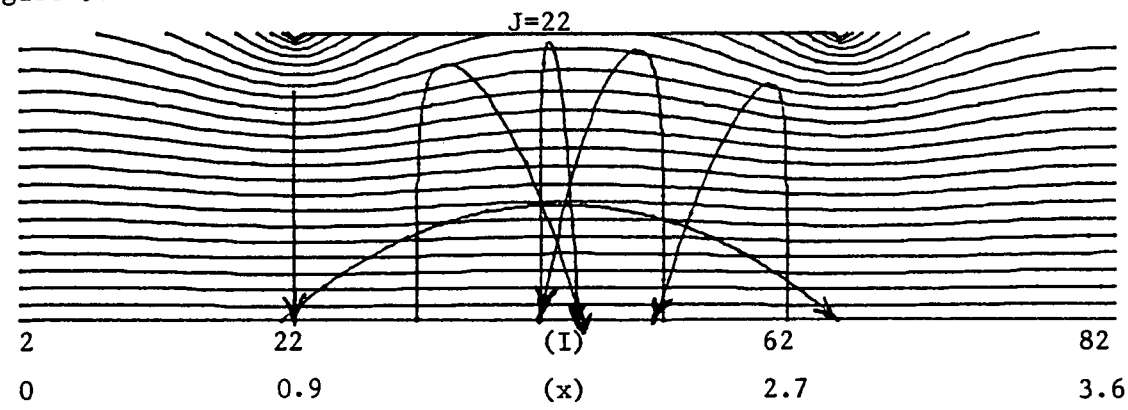


Figure 9. Equipotential Lines for the Test-Biased Fin Potential and Constraining Electron Trajectories on the Domain 2< I <82, 2< J <22

IV. DIPOLE SYNTHESIS OF ELECTROSTATIC POTENTIALS

Preliminary investigation was begun with the intention of developing methods for dipole synthesis of three-dimensional electrostatic potentials assuming a uniform potential in one coordinate direction. The problem can be considered two dimensional if a line dipole is constructed by letting two infinite lines of opposite charge approach each other in the limit of superposition. The result is a line dipole potential which is uniform in the coordinate direction parallel to the line dipole. Furthermore, in a plane perpendicular to the line dipole; a difference results from integration along the line charges producing a $1/r$ variation of the potential with distance from the line dipole, rather than the $1/r^2$ dependence associated with a point dipole.

The line dipole described above is to be used as the basic building block for synthesizing unknown potentials in a two-dimensional space that can be probed by electron beams. This space is bounded above by the charged surface that produces the potential of interest and bounded below by the zero potential reference plane of the electron beam apparatus. Since the potential must be zero on this boundary, line dipoles can be placed in pairs, one on either side of the boundary so as to be electrostatic images of each other. Figure 1 shows the space of interest, the zero-potential boundary and a line dipole pair. Notice that the pair of line dipoles is located outside the space of interest so that Laplace's equation will hold on the interior.

The model problem under consideration requires the determination of the magnitude, location and number of line dipole pairs that best approximate the potential existing in the space of interest resulting

from an unknown surface charge distribution. The potential in the region of interest is described by electron trajectory data including initial electron coordinates and velocities and final electron coordinates. The basic method begins with a uniform field approximation of the potential which can be modified by the iterative addition of line dipole potentials until a potential is created that reproduces the electron trajectory data.

To formulate an iterative scheme for the dipole synthesis of solutions to the model problem, suppose that the potential ψ exists as a result of the biasing of two fins relative to a zero potential reference plane. The last of the six trajectories listed in the statement of the model problem (i.e., the scaling trajectory) can be used for a uniform field approximation of ψ in order to develop an initial estimated potential ϕ_o. The equations of motion describing the scaling trajectory are

$$\frac{\partial^2 x}{\partial t^2} = \frac{q}{m}\frac{\partial \phi_o}{\partial x} = 0 \text{ and } \frac{\partial^2 y}{\partial t^2} = \frac{q}{m}\frac{\partial \phi_o}{\partial y} = \frac{q}{m} A_o. \qquad \text{(IV-1)}$$

A linear system of two equations

$$XT(6) = XO(6) + T \cdot VXO(6) \qquad \text{(IV-2)}$$

and

$$YT(6) = YO(6) + T \cdot VYO(6) + \frac{1}{2}\frac{q}{m} A_o \cdot T^2 \qquad \text{(IV-3)}$$

in two unknowns (i.e., the constants A_o and T) is developed by integrating equations (IV-1) twice and substituting the data for the scaling trajectory. When YT(6) = YO(6) = 0, the result is

$$A_o = \frac{2m}{q} \frac{VXO(6) \cdot VYO(6)}{[XT(6)-XO(6)]}. \qquad \text{(IV-4)}$$

Thus, an initial estimated potential, $\phi_o = A_o y$, can be calculated at all points in the region of interest based upon the scaling trajectory data and the electron charge to mass ratio $\frac{q}{m}$. The estimated potential ϕ can be synthesized from ϕ_o and a number of dipole fields (see Appendix A),

$$\phi_d(x,y) = A_d\left[\frac{y-y_d}{(x-x_d)^2 + (y-y_d)^2} + \frac{y+y_d}{(x-x_d)^2 + (y+y_d)^2}\right], \qquad \text{(IV-5)}$$

to improve the estimation of the biased fin potential,

$$\phi(x,y) = \phi_o(x,y) + \sum_d \phi_d(x,y), \qquad \text{(IV-6)}$$

according to some iterative scheme that specifies optimal dipole parameters A_d, x_d, y_d for meeting the constraints imposed by the electron trajectory data (i.e., trajectories 1-5). For each iteration of the scheme, estimated electron trajectories are developed from the initial trajectory data (i.e., XO(K), YO(K); K=1,2,...6) and the equations of motion (IV-1) substituting, for ϕ_o, the best estimate of the potential field $\phi(x,y)$ calculated from equation (IV-6). The coordinates XM(K) and YM(K) at the highest points on the estimated trajectories are recorded as well as the exit coordinates XE(K) and YE(K); for K=1,2,$\cdots$6. A discrepancy is defined

$$D(K) = \frac{[XT(K)-XE(K)]}{[XT(K)-XO(K)]} \; ; \; K=1,2,\cdots 5 \qquad \text{(IV-7)}$$

based upon estimated trajectory data XE(K) and actual trajectory data XT(K) and XO(K). In order to reduce the discrepancies, dipoles should be located between trajectories having discrepancies opposite in sign. Therefore, the trajectory with the largest absolute discrepancy

(e.g., $K=K_1$) is identified and the nearest trajectory having a discrepancy of opposite sign (e.g., $K=K_2$) is identified. The dipole x_d parameter,

$$x_d = \frac{1}{2}\,[XM(K_1) + XM(K_2)], \tag{IV-8}$$

is chosen midway between these two trajectories. The dipole y_d parameter is fixed, $y_d=1.4\cdot DY$, at a reasonable distance from the region of interest (e.g., as close to the region as possible without introducing local singularities, as discussed for the boundary value problem). Then a dipole magnitude A_d can be chosen which reduces the discrepancy of the K_1 trajectory to approximately zero.

Toward this end, consider the integration of the equations of motion (IV-1), with ϕ replacing ϕ_o, for electrons following the K_1 estimated trajectory through the estimated potential. If the time T represents the time required for electrons to traverse this trajectory, then the results of the integration are

$$XE(K_1) = XO(K_1) + T\cdot VXO(K_1) + \frac{1}{2}\frac{q}{m}\int_0^T \int_0^{t'} \frac{\partial\phi}{\partial x}\, dt\, dt'. \tag{IV-9}$$

and

$$YE(K_1) = YO(K_1) + T\cdot VYO(K_1) + \frac{1}{2}\frac{q}{m}\int_0^T \int_0^{t'} \frac{\partial\phi}{\partial y}\, dt\, dt'. \tag{IV-10}$$

The K_1 trajectory consists of a number of points whose coordinates have been determined by a stepwise solution of the equations of motion. The electric fields, $\frac{\partial\phi}{\partial x}$ and $\frac{\partial\phi}{\partial y}$, at these points were calculated along with the time steps for electron motion between the points in the development of the K_1 trajectory. Therefore, the integrals in equations (IV-9) and (IV-10) can be replaced by a pointwise summation of the electric fields

over the N points of the K_1 trajectory using trapezoidal integration to develop

$$XI(K_1) \equiv \int_0^T \int_0^{t'} \frac{\partial \phi}{\partial x} \, dt \, dt' = \frac{1}{2} \sum_{n=1}^{N} (S_{n-1} + S_n) \Delta t_n \tag{IV-11}$$

where

$$S_n = \frac{1}{2} \sum_{i=1}^{n} [(\frac{\partial \phi}{\partial x})_{i-1} + (\frac{\partial \phi}{\partial x})_i] \Delta t_i \tag{IV-12}$$

and

$$YI(K_1) \equiv \int_0^T \int_0^{t'} \frac{\partial \phi}{\partial y} \, dt \, dt' = \frac{1}{2} \sum_{n=1}^{N} (R_{n-1} + R_n) \Delta t_n \tag{IV-13}$$

where

$$R_n = \frac{1}{2} \sum_{i=1}^{n} [(\frac{\partial \phi}{\partial y})_{i-1} + (\frac{\partial \phi}{\partial y})_i] \Delta t_i . \tag{IV-14}$$

Substituting the approximation of equation (IV-13) into equation (IV-10), we have for $YE(K_1) = YO(K_1) = 0$,

$$T = - \frac{1}{2} \frac{q}{m} \frac{YI(K_1)}{VYO(K_1)} . \tag{IV-15}$$

Substituting the approximation of equation (IV-11) into equation (IV-9), we have

$$XE(K_1) = XO(K_1) + T \cdot VXO(K_1) + \frac{1}{2} \frac{q}{m} XI(K_1) . \tag{IV-16}$$

Now, we wish to add a dipole potential ϕ_d to the estimated potential ϕ, altering the K_1 trajectory only slightly and shifting the exit

coordinate from $XE(K_1)$ to the data coordinate $XT(K_1)$. The equations of motion for this new estimated potential are

$$\frac{\partial^2 x}{\partial t^2} = \frac{q}{m}\left(\frac{\partial \phi}{\partial x} + \frac{\partial \phi_d}{dx}\right) \text{ and } \frac{\partial^2 y}{\partial t^2} = \frac{q}{m}\left(\frac{\partial \phi}{\partial y} + \frac{\partial \phi_d}{\partial y}\right). \tag{IV-17}$$

Integration of the first of these equations (IV-17), yields

$$XT(K_1) = XO(K_1) + T\cdot VXO(K_1) + \frac{1}{2}\frac{q}{m}\int_0^T \int_0^{t'} \left(\frac{\partial \phi}{\partial x} + \frac{\partial \phi_d}{\partial x}\right) dt\, dt'. \tag{IV-18}$$

When trajectory K_1 is shifted only slightly, the integral in equation (IV-18) can be replaced by the summation of electric fields over the same N points used to generate equation (IV-16). Hence,

$$XT(K_1) = XO(K_1) + T\cdot VXO(K_1) + \frac{1}{2}\frac{q}{m}\left[XI(K_1) + A_d \cdot VI(K_1)\right], \tag{IV-19}$$

where

$$VI(K_1) \equiv \frac{1}{2}\sum_{n=1}^{N}(V_{n-1} + V_n)\Delta t_n \tag{IV-20}$$

and where

$$V_n \equiv \frac{1}{2}\sum_{i=1}^{n}\frac{1}{A_d}\left[\left(\frac{\partial \phi_d}{\partial x}\right)_{i-1} + \left(\frac{\partial \phi_d}{\partial x}\right)_i\right]\Delta t_i. \tag{IV-21}$$

Subtracting equation (IV-16) from equation (IV-20), we may solve for the dipole magnitude

$$A_d = \frac{2m}{q}\left[\frac{XT(K_1) - XE(K_1)}{VI(K_1)}\right]. \tag{IV-22}$$

After the addition of a dipole with magnitude A_d at coordinates (x_y, y_d), one iteration of the scheme for discrepancy reduction is

completed by scaling the dipole modified estimated potential,

$$\phi'(x,y) = SF \cdot \phi(x,y). \qquad \text{(IV-23)}$$

The scale factor,

$$SF = \frac{XE(6) - XO(6)}{XT(6) - XO(6)}, \qquad \text{(IV-24)}$$

is calculated using the exit coordinate XE(6) obtained for the stepwise development of the scaling trajectory in the dipole modified estimated potential.

Appendix D lists the main Fortran routine PTENT.F4 which implements the iterative routine described by equations (IV-1) through (IV-24). The user has the option of entering electron trajectory data by reading from file TRAJ.DAT or by answering pertinent questions in an interactive user mode. Table 4 shows the reduction of discrepancy and and of error estimates obtained for 15 iterations of the routine. Note that these quantities do not decrease monotonically and limits are reached after about 8 iterations of the routine. Figure 10 shows equipotential lines drawn with a linear interpolation subroutine of PTENT similar to the VPLOT subroutine described for the boundary value problem. The solution to the model problem depicted in Figure 10 can be compared to the test potential in Figure 8.

Appendix E contains a listing of the main Fortran routine RTENT.F4 which is also an iterative routine for the dipole synthesis of solutions to the model problem. RTENT.F4 differs from PTENT.F4 only in the methods used for determining the dipole x_d, y_d and A_d factors. For this new scheme, the x_d parameter can be calculated by equation (IV-8) when K_1 is identified as the trajectory having the largest positive discrepancy and K_2 is identified as the trajectory having the largest

Table 4

Error Estimates for Dipole Synthesis with PTENT.F4

Iteration	Discrepancy, DD $\sum_{i=1}^{5} \lvert XT(K_i) - XE(K_i) \rvert$	Average Error (%)	Average Absolute Error (%)	Maximum Absolute Error (%)
0	1.51	4.6	5.8	20.0
1	1.29	3.5	5.0	21.6
2	1.28	1.9	4.6	20.0
3	1.33	1.7	3.8	16.8
4	0.92	3.7	4.4	18.3
5	0.92	3.0	4.2	18.1
6	0.70	2.3	2.6	14.4
7	0.45	2.9	3.7	17.9
8	0.42	1.7	3.5	16.8
9	0.45	3.0	3.7	17.9
10	0.43	1.8	3.4	16.8
11	0.45	3.0	3.7	17.9
12	0.43	1.9	3.4	16.9
13	0.45	3.1	3.7	17.9
14	0.43	2.0	3.4	16.9
15	0.45	3.1	3.8	18.0

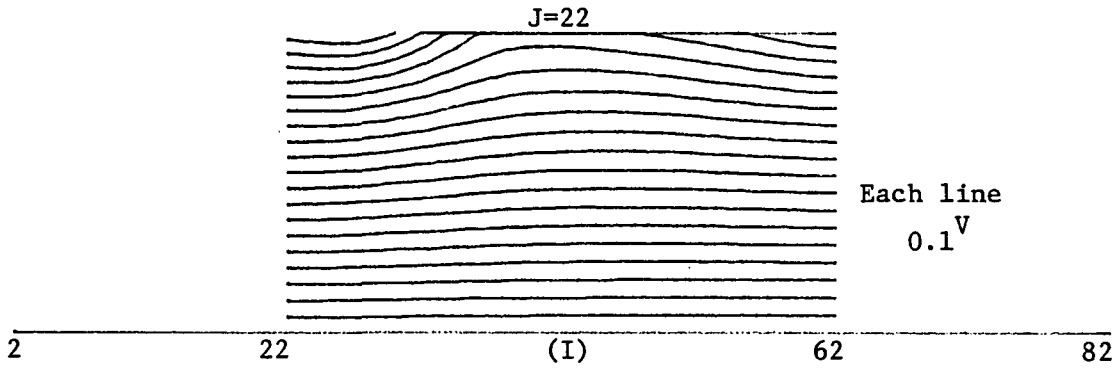


Figure 10. Solution to the Model Problem Synthesized with PTENT.F4

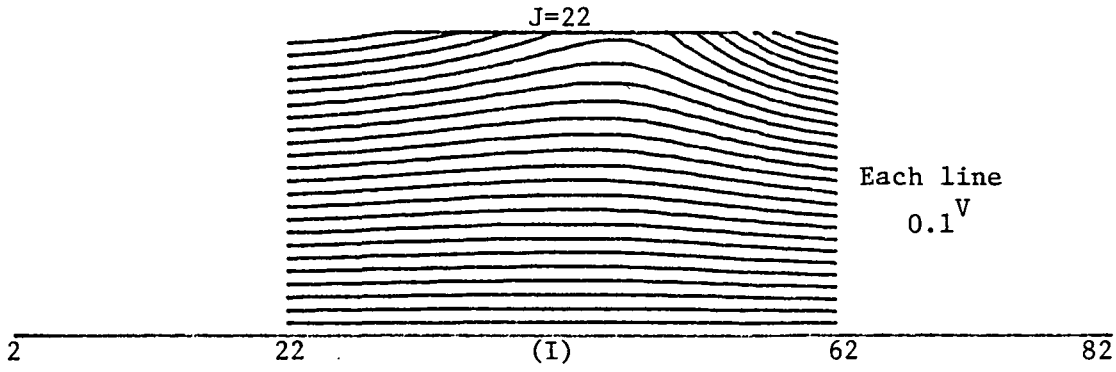


Figure 11. Solution to the Model Problem Synthesized with RTENT.F4

negative discrepancy. The y_d parameter is arrived at by a trial and error procedure designed to reduce the discrepancies of both the K_1 and K_2 trajectories. This is accomplished by finding a minimum value of

$$\frac{D(K_1)}{D(K_2)} - \frac{\frac{\partial \phi_d}{\partial x}[XM(K_1), YM(K_1)]}{\frac{\partial \phi_d}{\partial x}[XM(K_2), YM(K_2)]} \qquad \text{(IV-25)}$$

as y_d is allowed to run through a range of values. When the values of x_d and y_d have been chosen, a value of A_d is chosen to reduce the discrepancy of the K_1 trajectory (or the K_2 trajectory if its discrepancy is larger in absolute magnitude than the K_1 trajectory) to approximately zero as described by equations (IV-9) through (IV-22). Table 5 lists the error estimates for 15 iterations of the RTENT.F4 routine as compared to Table 4. Note that error estimates are larger for RTENT.F4 than for PTENT.F4 and that the reduction of discrepancy is slower. However, RTENT.F4 consistently underestimates the biased fin potential while PTENT.F4 consistently yields an overestimation. Therefore, upper and lower bounds can be established for the solution of the model problem. Figure 11 shows the estimated potential generated by RTENT.F4 which can be compared to Figures 10 and 8 to appreciate the accuracy of these iterative dipole routines.

Tables 6, 7 and 8 show the potential error estimates at 110 selected locations in the region of interest for 5, 10 and 15 iterations of routine RTENT.F4; respectively. First note that the largest errors occur at the edges of the region of interest in spatial regions unprobed by the electron trajectories. Note also that the errors at

Table 5

Error Estimates for Dipole Synthesis with RTENT.F4

Iterations	Discrepancy, DD $\sum_{i=1}^{5} \lvert XT(K_i) - XE(K_i) \rvert$	Average Error (%)	Average Absolute Error (%)	Maximum Absolute Error (%)
0	1.51	-4.6	5.8	20.0
1	1.20	-3.6	4.5	17.4
2	1.13	-3.5	4.8	18.9
3	1.08	-3.9	4.9	19.1
4	0.99	-3.9	4.5	18.3
5	0.95	-2.8	4.0	16.4
6	0.95	-2.3	3.9	17.6
7	0.93	-4.2	4.7	19.2
8	0.91	-3.8	4.5	18.8
9	0.73	-2.2	3.3	16.2
10	0.66	-3.3	3.8	17.9
11	0.67	-2.7	3.5	17.3
12	0.69	-2.0	3.3	16.7
13	0.64	-3.6	4.0	18.2
14	0.69	-3.2	3.7	17.7
15	0.64	-2.3	3.4	17.0

Table 6

Error Estimates for Dipole Synthesis with Routine RTENT.F4

(5 dipole pairs)

Y-COORDINATE INDEX (J)											
21	-13.6	- 3.3	3.8	5.8	5.2	3.9	2.6	0.6	- 3.5	-10.2	-18.0
19	- 7.9	- 3.4	1.9	4.2	4.2	3.3	1.8	-0.5	- 4.3	- 9.3	-12.4
17	- 5.1	- 2.8	0.9	3.0	3.3	2.6	1.1	-1.2	- 4.5	- 8.1	- 9.5
15	- 3.5	- 2.2	0.3	2.1	2.6	1.9	0.5	-1.7	- 4.5	- 7.1	- 7.7
13	- 2.5	- 1.8	-0.4	1.4	1.9	1.4	0.0	-2.0	- 4.4	- 6.4	- 6.6
11	- 2.0	- 1.7	-0.3	0.9	1.4	0.9	-0.4	-2.3	- 4.4	- 5.9	- 5.9
9	- 1.9	- 1.7	-0.7	0.4	0.9	0.5	-0.7	-2.5	- 4.5	- 5.8	- 5.7
7	- 2.4	- 2.3	-1.3	-0.1	0.5	0.2	-1.1	-2.9	- 4.9	- 6.1	- 6.1
5	- 4.2	- 3.9	-2.6	-1.0	-0.0	-0.1	-1.5	-3.7	- 6.1	- 7.6	- 7.7
3	-13.3	-12.3	-9.1	-4.8	-1.3	-0.4	-2.8	-7.4	-12.3	-15.7	-16.4
	22	26	30	34	38	42	46	50	54	58	62

X-COORDINATE INDEX (I)

Table 7

Error Estimates for Dipole Synthesis with Routine RTENT.F4

(10 dipole pairs)

Y-COORDINATE INDEX (J)											
21	-12.4	- 1.5	4.2	3.6	2.0	1.6	1.4	-0.0	- 3.6	-10.1	-17.9
19	- 6.8	- 2.2	2.0	2.6	1.8	1.3	0.6	-1.2	- 4.5	- 9.3	-12.3
17	- 4.2	- 2.0	0.8	1.8	1.5	0.8	-0.1	-1.9	- 4.8	- 8.2	- 9.4
15	- 2.8	- 1.7	0.2	1.1	1.0	0.4	-0.6	-2.4	- 4.9	- 7.2	- 7.7
13	- 2.0	- 1.5	-0.2	0.6	0.6	0.0	-1.0	-2.7	- 4.8	- 6.5	- 6.6
11	- 1.5	- 1.4	-0.5	0.2	0.2	-0.3	-1.4	-3.0	- 4.8	- 6.1	- 6.0
9	- 1.5	- 1.5	-0.8	-0.3	-0.1	-0.6	-1.7	-3.2	- 4.9	- 6.0	- 5.7
7	- 2.0	- 2.1	-1.5	-0.7	-0.5	-0.9	-2.0	-3.6	- 5.3	- 6.3	- 6.1
5	- 3.8	- 3.8	-2.8	-1.6	-0.9	-1.1	-2.4	-4.0	- 6.5	- 7.8	- 7.7
3	-12.9	-12.2	-9.3	-5.4	-2.2	-1.4	-3.6	-8.0	-12.7	-15.7	-16.5
	22	26	30	34	38	42	46	50	54	58	62

X-COORDINATE INDEX (I)

Table 8

Error Estimates for Dipole Synthesis with Routine RTENT.F4

(15 dipole pairs)

Y-COORDINATE INDEX (J)											
21	-11.4	- 0.3	5.3	4.5	2.7	2.2	2.1	0.8	- 2.8	- 9.3	-17.0
19	- 5.7	- 1.1	3.1	3.5	2.6	2.0	1.3	-0.4	- 3.7	- 8.4	-11.3
17	- 3.1	- 0.9	1.9	2.7	2.3	1.6	0.6	-1.1	- 4.0	- 7.3	- 8.5
15	- 1.6	- 0.6	1.2	2.0	1.9	1.2	0.1	-1.6	- 4.0	- 6.3	- 6.7
13	- 0.9	- 0.4	0.8	1.5	1.5	0.9	-0.2	-1.9	- 3.9	- 5.6	- 5.6
11	- 0.4	- 0.3	0.5	1.1	1.1	0.5	-0.6	-2.2	- 3.9	- 5.2	- 5.0
9	- 0.4	- 0.5	-0.1	0.7	0.8	0.2	-0.9	-2.4	- 4.0	- 5.0	- 4.8
7	- 0.9	- 1.0	-0.5	0.2	0.4	-0.0	-1.2	-2.8	- 4.4	- 5.4	- 5.1
5	- 2.7	- 2.8	-1.9	-0.7	-0.0	-0.3	-1.6	-3.6	- 5.7	- 6.9	- 6.8
3	-12.0	-11.3	-8.4	-4.5	-1.3	-0.5	-2.8	-7.2	-11.8	-15.0	-15.6
	22	26	30	34	38	42	46	50	54	58	62

X-COORDINATE INDEX (I)

nearly all of the selected locations decrease monotonically from the fifth to the tenth to the fifteenth iteration.

V. QUADRUPOLE SYNTHESIS OF ELECTROSTATIC POTENTIALS

The two iterative routines for dipole field synthesis of electric potentials yield good approximations of the biased fin potential. However, the addition of dipole fields individually chosen to reduce trajectory discrepancies may result in slow convergence and lengthy computer runs (e.g., the 15 iterations appearing in Table 4 required 30' 38" of CPU time). Efforts to speed the convergence of the synthesis process by adding dipole arrays, rather than single dipoles, have been unsuccessful. Note, in Figure 12, that the x-component of the electric field associated with the line dipole is not symmetric about the axis of the line dipole. The five curves of Figure 12 represent the electric field in the x-y reference plane of the biased fins (see Figure 7) along a line midway between the reference plane (y=0) and the biased fins (y=DY), as the result of a line dipole and its image located at five perpendicular distances from the reference plane. Note also that the dipole has equal and opposite effects at two spatially distinct locations in the region of interest. These characteristics of the dipole field are problematic when considering the placement of several line dipoles simultaneously in an effort to reduce the discrepancies associated with several electron trajectories. Consequently, the optimal placements of dipole array elements are not easily deduced since it is difficult to associate discrepancy reduction for one trajectory with a particular element of the dipole array.

A. Quadrupole Synthesis of Solutions to the Model Problem

Although the difficulties associated with synthesizing potential fields with dipole arrays may not be insurmountable, we have chosen to investigate

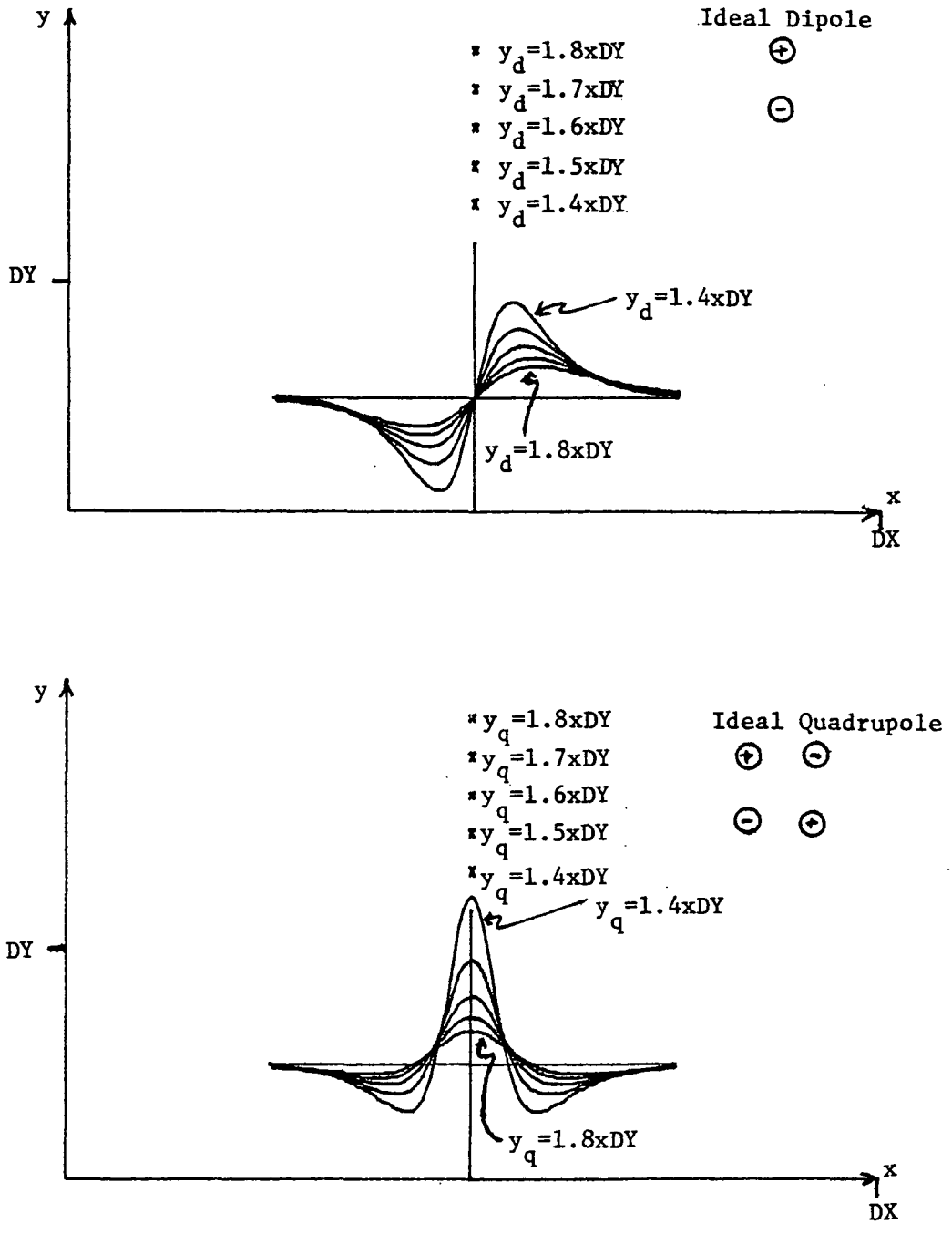


Figure 12. Electric Fields for Five Dipole and Quadrupole Locations along a Line inside the Region of Interest

the quadrupole synthesis of electric potentials. In analogy to the ideal line dipole, consider the ideal line quadrupole consisting of four lines of alternating charge extending infinitely in a direction perpendicular to the two dimensional region of interest. Figure 12 shows these four lines of charge which are allowed to approach each other in the limit of superposition to derive the potential field of the ideal line quadrupole. Since, we are interested in the synthesis of potentials which fall to zero along a reference plane, an image line quadrupole is simultaneously located an equal distance from the reference plane but on the side opposite the location of the line quadrupole itself. The potential field, as derived in Appendix A, due to a line dipole at (x_q,y_q) and its image at $(x_q,-y_q)$ is

$$\phi_q(x,y)=A_q(x-x_q)\left[\frac{(y+y_q)}{[(x-x_q)^2+(y+y_q)^2]^2}+\frac{(y-y_q)}{[(x-x_q)^2+(y-y_q)^2]^2}\right]. \quad (V\text{-}1)$$

Notice that this potential vanishes along the reference plane (y=0) regardless of the values of the quadrupole parameters A_q, x_q, and y_q. In analogy with dipole synthesis, the line quadrupole potential can be used for the synthesis of unknown potentials existing in a two dimensional space of the biased fins (i.e., along a line, y=constant, midway between the fins and the reference plane) as the result of an ideal line quadrupole and its image located at the same five perpendicular distances that were considered for the ideal dipole field of Figure 12. In comparing these two diagrams, notice that the quadrupole field is symmetric about its axis. Furthermore, the quadrupole field is largest

on its axis, rather than at locations on either side of the axis as was found for the dipole field. These quadrupole field characteristics facilitate the definition of iterative schemes for the synthesis of unknown potentials. In particular, a number of ideal quadrupoles can be located simultaneously with that quadrupole which is placed directly over each trajectory being primarily responsible for the reduction of discrepancy associated with that trajectory. However, before considering the topic of potential synthesis using the fields of quadrupole arrays, the synthesis of electric potentials by the iterative addition of single quadrupole fields will be considered. This will provide a basis for comparing dipole and quadrupole methods for the synthesis of solutions to Laplace's equation subject to the constraints imposed by electron trajectory data.

To formulate an iterative scheme for quadrupole synthesis of solutions to the model problem, suppose that the potential ψ exists as the result of the biasing of the fins relative to a zero potential reference plane. An initial estimated potential ϕ_o can be defined as the uniform-field approximation of ψ which satisfies the data for the last of the six trajectories listed in the statement of the model problem (i.e., the scaling trajectory). In analogy with equations IV-1) to (IV-4), an initial estimated potential $\phi_o = A_o y$, can be calculated at any point in the region of interest based upon the data for the scaling trajectory and the electron charge to mass ratio. After determining the uniform field approximation of the estimated potential, the estimated potential can be modified according to an iterative scheme that adds quadrupole fields $\phi_q(x,y)$ to $\phi_o(x,y)$ in order to improve the approximation of the biased fin potential $\psi(x,y)$.

$$\phi(x,y) = \phi_o(x,y) + \sum_{q=1}^{Q} \phi_q(x,y) \qquad (V\text{-}2)$$

For each iteration of the scheme, estimated electron trajectories in the estimated potential field are developed from a knowledge of initial electron coordinates and velocities for the first five electron trajectories listed as data for the model problem. The coordinates (XM(K), YM(K)) at the highest points on the estimated electron trajectories are recorded as well as the exit coordinates (XE(K), YE(K)) for K = 1, 2, 3, 4, 5. Then trajectory discrepancies,

$$D(K) = XT(K) - XE(K), \qquad (V\text{-}3)$$

are defined for each trajectory as the difference between the final x-coordinate XT(K) listed as data and the final x-coordinate XE(K) found for the estimated trajectories in the estimated potential. The sum of the absolute discrepancy values

$$DD = \sum_{K=1}^{5} |D(K)| \qquad (V\text{-}4)$$

is calculated in order to obtain a quantitative measure of the error in the approximation of $\psi(x,y)$ by $\phi(x,y)$. The object of each iteration is the reduction of DD by the addition of an ideal line quadrupole and its image with a magnitude and spatial location that is optimal in some regard. In particular, let the quadrupole be located a fixed distance from the region of interest (e.g., $y_q = 1.4DY$) and let the trajectory K_1 having the largest absolute discrepancy, $|D(K_1)|$, define the x-coordinate location of the quadrupole (e.g., $x_q = XM(K_1)$). Then a quadrupole magnitude A_q can be chosen which reduces the largest absolute trajectory discrepancy to approximately zero. Toward this end, consider the equations of motion for electrons following the estimated trajectory K_1 through the estimated potential $\phi(x,y)$.

$$\frac{\partial^2 x}{\partial t^2} = \frac{q}{m}\frac{\partial \phi}{\partial x} \; ; \; \frac{\partial^2 y}{\partial t^2} = \frac{q}{m}\frac{\partial \phi}{\partial y} \tag{V-5}$$

The integration of these equations over the time T required for electrons traveling along trajectory K_1 yields

$$XE(K_1) = XO(K_1) + T \cdot VXO(K_1) + \frac{1}{2}\frac{q}{m}\int_0^T \int_0^{t'} \frac{\partial \phi}{\partial x}\, dt \; dt'. \tag{V-6}$$

and

$$YE(K_1) = YO(K_1) + T \cdot VYO(K_1) + \frac{1}{2}\frac{q}{m}\int_0^T \int_0^{t'} \frac{\partial \phi}{\partial y}\, dt \; dt'. \tag{V-7}$$

Recall that the trajectory K_1 consists of a number of points whose coordinates were determined by a stepwise solution of the equations of motion. Furthermore, the electric fields $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial y}$ were determined at these points by interpolation from a grid in order to proceed with the stepwise development of the trajectory. Since these electric fields and corresponding coordinate values are available, the integrals in equations (V-6) and (V-7) may be replaced by a pointwise summation over the trajectory K_1. Using the trapezoidal rule for numerical integration, we obtain

$$VXI(K_1) \equiv \int_0^T \int_0^{t'} \frac{\partial \phi}{\partial x}\, dt \; dt' = \sum_{n=1}^{N} \frac{1}{2}(S_n + S_{n-1})\Delta t_n \tag{V-8}$$

where

$$S_n = \sum_{i=1}^{n} \frac{1}{2}\left[\left(\frac{\partial \phi}{\partial x}\right)_i + \left(\frac{\partial \phi}{\partial x}\right)_{i-1}\right]\Delta t_i. \tag{V-9}$$

Similarly, we obtain

$$VYI(K_1) \equiv \int_0^T \int_0^{t'} \frac{\partial \phi}{\partial y}\, dt\ dt' = \sum_{n=1}^{N} \frac{1}{2}(V_n + V_{n-1})\Delta t_n \qquad \text{(V-10)}$$

where

$$V_n = \sum_{i=1}^{n} \frac{1}{2}\left[\left(\frac{\partial \phi}{\partial y}\right)_i + \left(\frac{\partial \phi}{\partial y}\right)_{i-1}\right]\Delta t_i \qquad \text{(V-11)}$$

and N represents the number of time steps Δt_i used in the development of trajectory K_1. Substituting these expressions for the integrals in equation (V-6) and (V-7), we obtain estimates of the time T required for electrons following trajectory K_1 and the electron exit coordinate $XE(K_1)$. With $YE(K_1) = YO(K_1) = 0$, we have

$$T = -\frac{1}{2}\frac{q}{m}\frac{VYI(K_1)}{VYO(K_1)} \qquad \text{(V-12)}$$

and

$$XE(K_1) = XO(K_1) + T \cdot VXO(K_1) + \frac{1}{2}\frac{q}{m} VXI(K_1). \qquad \text{(V-13)}$$

Notice that $T > 0$ for negatively charged electrons.

We wish to add a quadrupole field $\phi(x,y)$ altering the trajectory K_1 only slightly and shifting the exit coordinate from its estimated value $XE(K_1)$ to the data value $XT(K_1)$. The equations of motion for electrons in this new potential field are

$$\frac{\partial^2 x}{\partial t^2} = \frac{q}{m}\left(\frac{\partial \phi}{\partial x} + \frac{\partial \phi_q}{\partial x}\right) \text{ and } \frac{\partial^2 y}{\partial t^2} = \frac{q}{m}\left(\frac{\partial \phi}{\partial y} + \frac{\partial \phi_q}{\partial y}\right). \qquad \text{(V-14)}$$

The integration of the first of these equations yields

$$XT(K_1^*) + XO(K_1^*) + T^* \cdot VXO(K_1^*) + \frac{1}{2}\frac{q}{m}\int_0^{T^*}\int_0^{t'}\left(\frac{\partial \phi}{\partial x} + \frac{\partial \phi_q}{\partial x}\right)dt\ dt'. \qquad \text{(V-15)}$$

When K_1^* trajectory is not too different from the K_1 trajectory, the time T^* is not much different from the time T and the integral expression in equation (V-15) can be evaluated by summation of electric fields over the same N points used for the evaluation of the integral in equation (V-6). Therefore,

$$XT(K_1^*) \simeq XO(K_1) + T \cdot VXO(K_1) + \frac{1}{2}\frac{q}{m}[VXI(K_1) + A_q\, VI(K_1)] \qquad (V\text{-}16)$$

where

$$VI(K_1) = \sum_{n=1}^{N} \frac{1}{2}(V_{q_n} + V_{q_{n-1}})\Delta t_n \qquad (V\text{-}17)$$

and

$$V_{q_n} = \sum_{i=1}^{n} \frac{1}{2A_q}\left[\left(\frac{\partial \phi_q}{\partial x}\right)_i + \left(\frac{\partial \phi_q}{\partial x}\right)_{i-1}\right]\Delta t_i. \qquad (V\text{-}18)$$

Subtracting equation (V-13) from equation (V-16), and solving for A_q, we find

$$A_q = \frac{2m}{q}\left[\frac{XT(K_1^*) - XE(K_1)}{VI(K_1)}\right]. \qquad (V\text{-}19)$$

This expression for A_q can be used to calculate the magnitude of the ideal line quadrupole which, when located above the trajectory exhibiting the largest discrepancy, reduces that discrepancy approximately to zero.

After a quadrupole field with the appropriate magnitude is added to the estimated potential, a scaling operation is executed to complete one iteration of the scheme. The estimated potential is scaled by an amount sufficient to reduce the discrepancy of the scaling trajectory approximately to zero. To derive an algorithm for this purpose, we

assume that electrons following the scaling trajectory experience no net force in the x-direction. More specifically, this implies that

$$XE(6) = XO(6) + T \cdot VXO(6) \tag{V-20}$$

with

$$T = -\frac{1}{2}\frac{q}{m}\frac{VYI(6)}{VYO(6)}. \tag{V-21}$$

Then, scaling the potential by a factor SF, we shift the exit coordinate to a value

$$XT(6) = XO(6) + T^{*} \cdot VXO(6) \tag{V-22}$$

where

$$T^{*} = -\frac{1}{2}\frac{q}{m} SF \frac{VYI(6)}{VYO(6)} = SF \cdot T \tag{V-23}$$

Therefore, using equations (V-20) to (V-22) to solve equation (V-23) for the scale factor, we obtain

$$SF = \frac{T^{*}}{T} = \frac{XT(6) - XO(6)}{XE(6) - XO(6)}. \tag{V-24}$$

Recall that this same algorithm was used for scaling the estimated potential when solving the model problem by dipole synthesis.

The main routine QPOLE.F4 implements the iterative scheme described above for the quadrupole synthesis of solutions to the model problem. QPOLE.F4 is similar to the dipole routines PTENT.F4 and RTENT.F4, particularly in regard to the iterative procedures used for scaling the estimated potential and for determining an optimal pole magnitude after a pole location has been specified. The major difference between the dipole and quadrupole routines is the criteria used for selecting optimal pole locations. Table 9 lists error

Table 9

Error Estimates for Quadrupole Synthesis with Routine QPOLE.F4

Iteration	Discrepancy (DD)	Average Error (%)	Average Absolute Error (%)	Maximum Absolute Error (%)
0	1.51	-4.6	5.8	20.0
1	0.99	-3.4	4.5	19.0
2	1.00	-1.4	4.1	17.5
3	0.97	-1.9	4.1	17.8
4	0.60	-1.5	2.7	16.2
5	0.56	-1.7	2.7	16.5
6	0.55	-2.0	2.7	16.7
7	0.48	-2.5	3.1	17.6
8	0.44	-2.9	3.3	18.1
9	0.51	-0.9	3.0	16.3
10	0.49	-1.4	3.0	16.8
11	0.47	-2.0	3.0	17.2
12	0.45	-2.5	3.1	17.7
13	0.43	-3.1	3.4	18.1
14	0.22	-1.5	1.9	13.5
15	0.20	-1.7	2.0	13.6

estimates obtained after 15 iterations of the scheme for quadrupole synthesis which can be compared to the equivalent error estimates for dipole synthesis appearing in Tables 4 and 5. Note that in 15 iterations of the quadrupole routine, the discrepancy is reduced to a value considerabley lower than the discrepancy values achieved with 15 iterations of the dipole schemes. This discrepancy reduction for the quadrupole scheme is accomplished in 30'32" of CPU time representing only a small improvement in the amount of time expended per iteration. Although convergence limits are not reached in 15 iterations of the QPOLE.F4 routine, the indication is that the method would ultimately converge to a better approximation of the biased fin potential than solutions obtained with the dipole methods. Tables 10, 11 and 12 show the error percentages, after 5, 10 and 15 iterations respectively, associated with the estimated potential at 110 points chosen arbitrarily from a much finer grid (i.e., the interpolation of electric fields with subroutine NEWTON requires a finer grid spacing) in the reference frame of the biased fins. Notice that the error percentages for points near spatial regions probed by the electron trajectories (e.g., 30<I<54 and 5<J<17) decrease with successive iterations of the quadrupole scheme. The improvement of the estimated potential in these regions is accomplished while a degradation in the accuracy of the estimated potential results for points further from the space probed by the electron trajectories. This result for quadrupole synthesis of solutions to the model problem can be compared to similar results for dipole synthesis which have been discussed previously with references to Tables 4, 5 and 6. Figure 13 shows equipotential lines for the QPOLE.F4 routine which can be compared to Figures 8, 10 and 11 depicting the test

Table 10

Estimated Potential Error Percentages for Quadrupole Synthesis (5 Iterations of Routine QPOLE.F4)

Y COORDINATE INDEX (J)	22	26	30	34	38	42	46	50	54	58	62
21	-16.5	- 6.8	0.9	5.0	5.5	2.9	0.4	1.1	-0.4	- 1.6	- 5.7
19	-10.8	- 6.4	- 0.6	3.0	3.9	2.5	1.0	0.6	-0.3	- 1.4	- 1.8
17	- 7.9	- 5.5	- 1.4	1.7	2.7	2.1	1.1	0.5	-0.2	- 0.8	- 0.2
15	- 6.2	- 4.8	- 1.8	0.8	1.9	1.7	1.1	0.5	-0.0	- 0.2	0.6
13	- 5.1	- 4.2	- 2.0	0.1	1.2	1.3	0.9	0.4	0.1	0.1	1.1
11	- 4.5	- 3.9	- 2.2	-0.4	0.7	1.0	0.8	0.4	0.1	0.3	1.3
9	- 4.3	- 3.9	- 2.5	-0.8	0.3	0.7	0.6	0.2	-0.1	0.1	1.1
7	- 4.7	- 4.4	- 3.0	-1.3	-0.1	0.5	0.3	-0.1	-0.5	- 0.4	0.5
5	- 6.4	- 6.0	- 4.3	-2.2	-0.5	0.2	-0.0	-0.9	-1.8	- 2.1	- 1.4
3	-15.3	-14.2	-10.7	-5.9	-1.8	-1.8	-1.3	-4.7	-8.3	-10.7	-10.8

X COORDINATE INDEX (I)

Table 11

Estimated Potential Error Percentages for Quadrupole Synthesis (10 Iterations of Routine QPOLE.F4)

Y COORDINATE INDEX (J)	22	26	30	34	38	42	46	50	54	58	62
21	-16.8	- 7.1	0.1	3.5	4.4	4.0	2.6	2.9	0.7	- 0.9	- 5.3
19	-11.1	- 6.8	- 1.3	2.0	3.3	3.3	2.6	2.1	0.8	- 0.7	- 1.3
17	- 8.3	- 6.0	- 2.0	1.0	2.4	2.7	2.4	1.8	0.9	- 0.0	0.3
15	- 6.5	- 5.2	- 2.3	0.2	1.7	2.2	2.1	1.6	0.9	0.5	1.2
13	- 5.5	- 4.7	- 2.5	-0.3	1.2	1.8	1.8	1.4	1.0	0.9	1.7
11	- 4.9	- 4.3	- 2.6	-0.7	0.7	1.4	1.5	1.3	1.0	1.0	1.8
9	- 4.7	- 4.3	- 2.8	-1.1	0.4	1.1	1.3	1.0	0.8	0.9	1.7
7	- 5.1	- 4.8	- 3.3	-1.5	0.0	0.9	1.0	0.7	0.3	0.3	1.1
5	- 6.7	- 6.3	- 4.6	-2.4	-0.4	0.6	0.6	-0.2	-1.1	- 1.4	- 0.8
3	-15.6	-14.5	-11.1	-6.1	-1.7	0.4	-0.7	-3.9	-7.6	-10.1	-10.3

X COORDINATE INDEX (I)

Table 12

Estimated Potential Error Percentages for Quadrupole Synthesis (15 Iterations of Routine QPOLE.F4)

Y COORDINATE INDEX (J)	22	26	30	34	38	42	46	50	54	58	62
21	-12.4	- 1.6	2.7	-0.6	-1.4	-0.1	0.3	1.8	0.4	- 0.8	- 5.0
19	- 7.1	- 2.7	0.4	-0.4	-0.9	-0.2	0.4	1.0	0.3	- 0.7	- 1.1
17	- 4.7	- 2.7	- 0.7	-0.6	-0.7	-0.3	0.3	0.6	0.3	- 0.2	0.5
15	- 3.3	- 2.5	- 1.2	-0.8	-0.6	-0.3	0.1	0.4	0.3	0.3	1.3
13	- 2.6	- 2.4	- 1.5	-1.0	-0.7	-0.3	0.0	0.2	0.3	0.6	1.7
11	- 2.2	- 2.3	- 1.7	-1.2	-0.8	-0.4	-0.1	0.1	0.2	0.7	1.8
9	- 2.2	- 2.5	- 2.0	-1.4	-0.9	-0.5	-0.3	-0.1	0.0	0.5	1.7
7	- 2.7	- 3.0	- 2.5	-1.8	-1.1	-0.7	-0.5	-0.5	-0.4	- 0.1	1.0
5	- 4.5	- 4.7	- 3.9	-2.6	-1.5	-0.8	-0.8	-1.3	-1.8	- 1.8	- 0.9
3	-13.6	-13.0	-10.3	-6.3	-2.7	-1.0	-2.0	-5.0	-8.3	-10.4	-10.4

X COORDINATE INDEX (I)

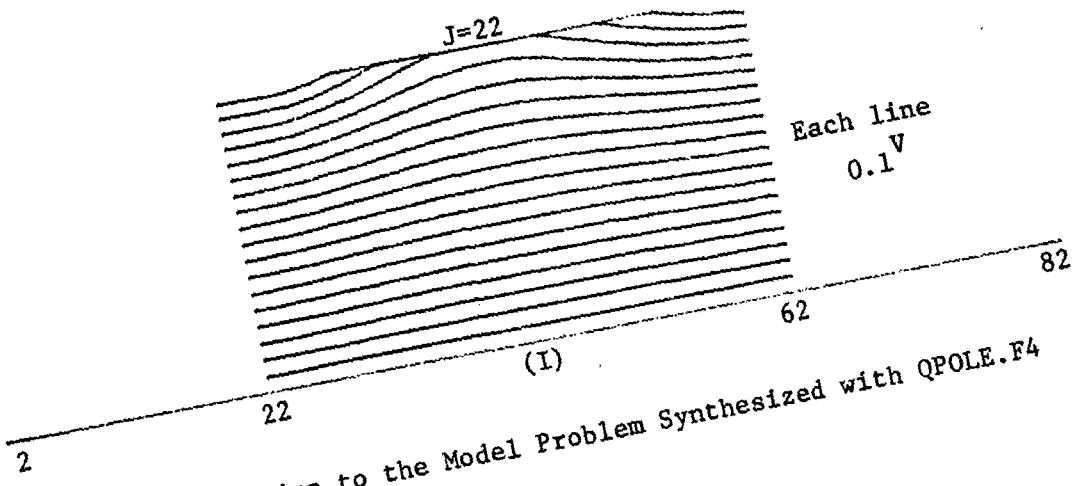


Figure 13. Solution to the Model Problem Synthesized with QPOLE.F4

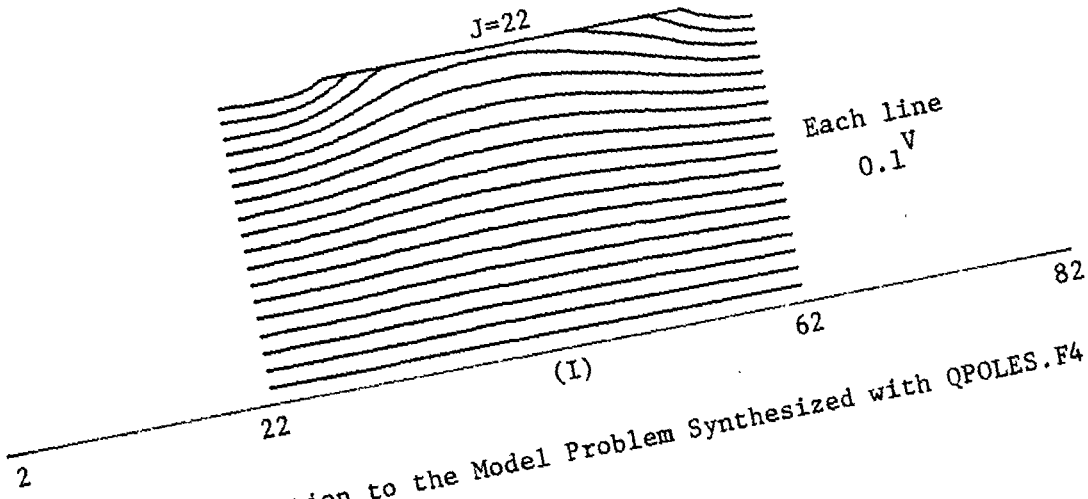


Figure 14. Solution to the Model Problem Synthesized with QPOLES.F4

potential and the dipole synthesized solutions.

B. Potential Synthesis with Quadrupole Arrays

In order to speed the process of discrepancy reduction, consider the placement of five ideal line quadrupoles upon each iteration of a new scheme for quadrupole synthesis of solutions to the model problem. The procedure for calculating an initial estimated potential, $\phi_o(x,y)$, based upon a uniform field approximation and the data for the scaling trajectory (i.e., K=6) is unchanged and described by equations IV-1) through (IV-4). The estimated potential can then be modified by an iterative scheme that locates five quadrupoles simultaneously in order to achieve discrepancy reduction for all five of the electron trajectories. Equation (V-25) depicts this situation

$$\phi(x,y) = \phi_o(x,y) + \sum_{i=1}^{N} \sum_{K=1}^{5} \phi_K(x,y) \qquad \text{(V-25)}$$

where $\phi_K(x,y)$ is given by equation (V-1) and the i index runs up to the total number of iterations that are needed for adequate discrepancy reduction. In developing this scheme, we expect that each iteration will require more CPU time since five quadrupoles are located as compared to only one with the old scheme. However, we hope that fewer iterations will be required to achieve a given reduction in discrepancy and that the total CPU time expended will decrease. To define the new scheme, let all five quadrupoles be located with a y-coordinate value of 1.4DY. Let the x-coordinate values of the quadrupole locations be equal to the x-coordinates corresponding to the highest points on the five estimated trajectories that have been developed for the estimated potential.

$$x_K = XM(K) \; ; \; 1,2,3,4,5 \tag{V-26}$$

Once the five quadrupole locations have been chosen, we seek five quadrupole magnitudes (i.e., A_K;K = 1, 2, 3, 4, 5) which reduce the discrepancies associated with all five trajectories to approximately zero. Toward this end, consider the equations of motion for electrons moving in the estimated potential $\phi(x,y)$. Then, in analogy with equations (V-5) through (V-13), an equation for each estimated trajectory can be determined.

$$XE(K) = XO(K) + T \cdot VXO(K) + \frac{1}{2}\frac{q}{m} VXI(K) \; ; \; K = 1,2,3,4,5 \tag{V-27}$$

Now, we wish to add five quadrupole fields to the estimated potential so as to alter the five estimated trajectories only slightly and shifting the estimated exit coordinates (XE(K) ; K = 1, 2, 3, 4, 5) over to the exit coordinates (XT(K) ; K = 1, 2, 3, 4, 5) measured as data. The equations of motion in this new estimated potential are

$$\frac{\partial^2 x}{\partial t^2} = \frac{q}{m}\left(\frac{\partial \phi}{\partial x} + \sum_{K=1}^{5} \frac{\partial \phi_K}{\partial x}\right) \tag{V-28}$$

and

$$\frac{\partial^2 y}{\partial t^2} = \frac{q}{m}\left(\frac{\partial \phi}{\partial y} + \sum_{K=1}^{5} \frac{\partial \phi_K}{\partial x}\right). \tag{V-29}$$

An integration of the equations of motion subject to the initial and final conditions available as trajectory data results in five equations describing the electron trajectories in a new estimated potential. Thus,

$$XT(K) = XO(K) + T \cdot VXO(K) + \frac{1}{2}\frac{q}{m}\left[VXI(K) + \sum_{k=1}^{5} A_K \cdot VI(K)\right], \tag{V-30}$$

with VI(K) defined by equations (V-17) and (V-18) and where

K = 1, 2, 3, 4, 5. The subtraction of equation (V-27) from equation (V-30) yields,

$$XT(K) - XE(K) = \frac{1}{2}\frac{q}{m}\sum_{K=1}^{5} A_K VI(K) \; ; \; K = 1,2,3,4,5. \qquad \text{(V-31)}$$

Equation (V-31) represents a linear system of five equations in five unknowns (i.e., A_K;K = 1, 2, 3, 4, 5) which can be solved by gaussian elimination procedures to determine the magnitude of the five ideal line quadrupoles. When the potential fields associated with these quadrupoles are added to the estimated potential, a new estimated potential is synthesized. Then the discrepancies associated with the new estimated trajectories, developed in the new estimated potential, should individually be approximately equal to zero. Thus, complete discrepancy reduction could be accomplished with a single application of the new scheme. However, the scheme involves several approximations which produce a less than total reduction of discrepancy. For example, the time T in equations (V-27) and (V-30), as well as the VXI(K) terms, depend upon the old and new estimated potentials respectively. Therefore, the terms involving these expressions will not cancel exactly as indicated in the derivation of equation (V-31). The approximation as implied by equations (V-27) and (V-30) is that the five electron trajectories are not altered greatly by the addition of five quadrupoles for the reduction of discrepancy. Therefore, the errors introduced by these approximations will result in a less than total reduction of discrepancy for any one iteration of the scheme. However, on successive iterations, the approximations will produce lesser errors and for a suitable initial potential estimate ϕ_o, the method of

successive approximation will result in a convergent iterative scheme.

The main routine QPOLES.F4 (Appendix E) implements the method described above for the synthesis of solutions to the model problem. Table 13 shows the error estimates generated at 110 points on a grid located between the biased fins and above the zero potential reference plane. These results can be compared to Tables 10, 11 and 12 which list the results of routine QPOLE.F4 also used for quadrupole synthesis of solutions to the model problem. Note that the single pole scheme results in solutions that are scaled better than the solutions synthesized by the addition of five poles simultaneously. However, discrepancy reduction is much more rapid for the latter method. A reduction in discrepancy from 1.5 to 0.4 can be accomplished with a single iteration of the QPOLES.F4 scheme requiring 3'24" of computer processing time (CPU time) as compared to 14 iterations of the QPOLE.F4 scheme requiring 28'29". Although 4 iterations of the method have been listed, the reduction of discrepancy and of approximation errors reaches a limit after only 2 iterations. This is in contrast to the single pole scheme where convergence limits had not been reached after 15 iterations. Figure 14 shows equipotential lines for the QPOLES.F4 routine.

Although convergence is rapid for the main routine QPOLES.F4, the synthesized solution to the model problem underestimates the biased fin potential at 109 of the 110 points considered. The scaling procedure itself is unchanged from the successful procedure used previously for the single quadrupole and dipole routines. The difficulty seems to be manifested in balancing the strategies of field scaling and field shaping depending on whether the addition of quadrupoles, reducing discrepancies for the first five trajectories, has a sufficiently detrimental effect on the discrepancy of the scaling trajectory. To possibly save

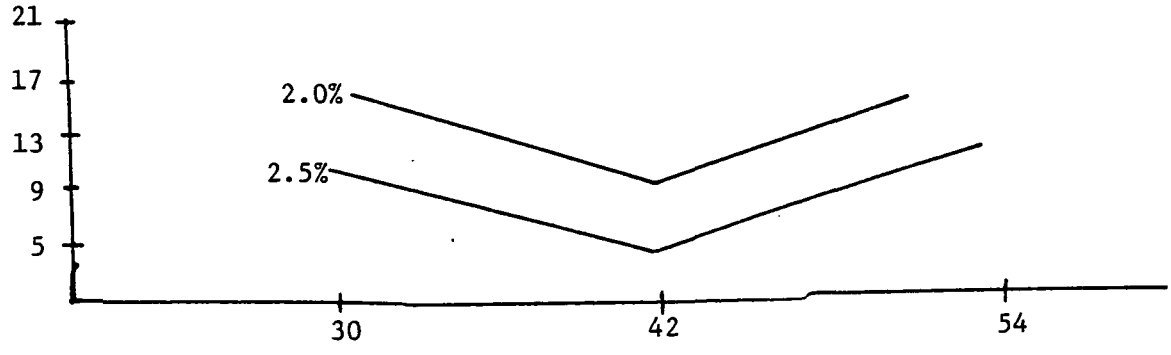
Table 13

Error Estimates for Routine QPOLES.F4 with Step Size d
(0.013 < d < 0.025) and Scale Tolerance of 1.0%

Iterations	Discrepancy (DD)	Scale Error (%)	Average Error (%)	Average Absolute Error (%)	Maximum Absolute Error (%)
0	1.51	0.05	-4.6	5.8	20.0
1	0.38	0.02	-3.2	3.3	14.0
2	0.08	0.16	-3.3	3.3	13.7
3	0.05	0.02	-3.2	3.3	14.1
4	0.06	0.22	-3.3	3.3	14.5

Y COORDINATE INDEX (J)												
	21	-12.4	- 2.2	1.3	-1.3	-1.5	-1.4	-1.3	-1.1	- 2.5	- 3.7	- 8.8
	19	- 7.3	- 3.3	- 0.6	-1.2	-1.5	-1.5	-1.4	-1.6	- 2.5	- 3.8	- 4.7
	17	- 5.0	- 3.3	- 1.5	-1.4	-1.5	-1.6	-1.6	-1.9	- 2.5	- 3.3	- 3.0
	15	- 3.8	- 3.2	- 2.0	-1.7	-1.7	-1.7	-1.8	-2.0	- 2.5	- 2.8	- 2.0
	13	- 3.1	- 3.0	- 2.3	-1.9	-1.8	-1.9	-2.1	-2.4	- 2.4	- 2.4	- 1.4
	11	- 2.8	- 3.0	- 2.5	-2.1	-1.9	-1.9	-2.0	-2.2	- 2.4	- 2.3	- 1.3
	9	- 2.8	- 3.1	- 2.8	-2.4	-2.1	-2.0	-2.2	-2.4	- 2.6	- 2.4	- 1.4
	7	- 3.4	- 3.6	- 3.3	-2.8	-2.3	-2.2	-2.4	-2.8	- 3.1	- 2.9	- 2.0
	5	- 5.1	- 5.4	- 4.7	-3.6	-2.7	-2.4	-2.7	-3.5	- 4.3	- 4.6	- 3.9
	3	-14.2	-13.7	-11.0	-7.2	-3.9	-2.6	-3.9	-7.2	-10.7	-13.0	-13.0
		22	26	30	34	38	42	46	50	54	58	62

X COORDINATE INDEX (I)



time, a decision to scale or not is made by comparing the scale error (i.e., discrepancy) of this trajectory with an arbitrary scale tolerance. Tables 14, 13 and 15 show error estimates for the synthesis of solutions to the model problem for scale tolerances of 5.0%, 1.0% and 0.1% respectively. Note that a scale tolerance of 0.1% is preferable. Decreasing the scale tolerance further will not improve the accuracy of the solution, since the routine already scales at every opportunity with a scale tolerance of 0.1%. Thus, invoking the scaling operation after each shaping operation facilitates the reduction of discrepancy and produces a better approximation of the biased fin potential. However, the synthesized solution to the model problem still underestimates the biased fin potential at 109 of the 110 points considered. Thus, field shaping by the addition of five quadrupoles simultaneously results in a rapid convergence at the expense of accurate scaling.

Recall that the terms VI(K) and VXI(K) in equations (V-17) and (V-8) are calculated by summing terms generated by the trapezoidal rule for integration over the estimated electron trajectories. The accuracy of this integration depends upon the number of trapezoidal terms as determined by the size of the time step used for tracking electrons through a potential field. Subroutine DVOGEL monitors the trajectory arc length corresponding to a particular time step to determine the acceptance or rejection of that time step. Whenever the arc length surpasses some upper limit, the time step is reduced and conversely whenever the arc length falls below some lower limit, the time step is increased. Table 16 shows x and y coordinate values at points along a particular electron trajectory as well as the electric field at those points for an ideal line quadrupole that will be added

Table 14

Error Estimates for Routine QPOLES.F4 with Scale Tolerance of 5.0%

Iterations	Discrepancy (DD)	Scale Error (%)	Average Error (%)	Average Absolute Error (%)	Maximum Absolute Error (%)
0	1.51	0.05	-4.6	5.8	20.0
1	0.62	2.78	-7.0	7.0	17.4
2	0.26	2.64	-6.9	6.9	17.3
3	0.05	2.51	-6.8	6.8	17.3
4	0.03	2.51	-6.7	6.7	17.6

Y COORDINATE INDEX (J)											
21	-16.2	- 6.5	- 3.0	- 4.3	-3.2	-3.0	-3.7	- 3.7	- 5.8	- 8.4	-13.7
19	-11.3	- 7.4	- 4.5	- 4.2	-3.7	-3.5	-3.9	- 4.5	- 6.0	- 8.2	- 9.5
17	- 9.0	- 7.3	- 5.3	- 4.5	-4.0	-3.9	-4.2	- 4.9	- 6.1	- 7.5	- 7.6
15	- 7.8	- 7.1	- 5.7	- 4.8	-4.3	-4.2	-4.5	- 5.1	- 6.1	- 6.9	- 6.5
13	- 7.1	- 6.8	- 5.9	- 5.1	-4.6	-4.5	-4.7	- 5.3	- 6.0	- 6.5	- 5.9
11	- 6.7	- 6.7	- 6.1	- 5.3	-4.8	-4.7	-4.9	- 5.5	- 6.1	- 6.3	- 5.7
9	- 6.7	- 6.9	- 6.3	- 5.6	-5.1	-4.9	-5.2	- 5.7	- 6.2	- 6.4	- 5.7
7	- 7.2	- 7.4	- 6.8	- 6.0	-5.4	-5.1	-5.4	- 6.0	- 6.7	- 6.9	- 6.3
5	- 8.9	- 9.0	- 8.1	- 6.8	-5.7	-5.3	-5.8	- 6.8	- 7.9	- 8.5	- 8.0
3	-17.6	-17.0	-14.2	-10.3	-6.9	-5.6	-6.9	-10.3	-14.0	-16.5	-16.8
X (J)	22	26	30	34	38	42	46	50	54	58	62

X COORDINATE INDEX (I)

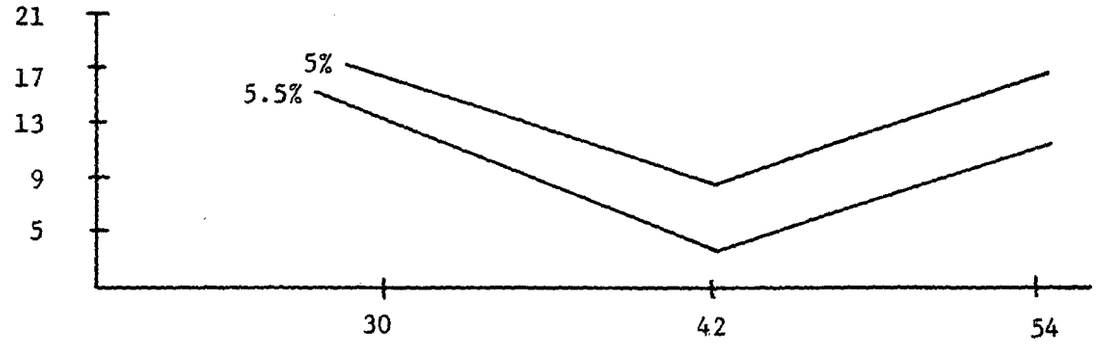


Table 15

Error Estimates for Routine QPOLES.F4 with Scale Tolerance of 0.1%

Iterations	Discrepancy (DD)	Scale Error (%)	Average Error (%)	Average Absolute Error (%)	Maximum Absolute Error (%)
0	1.51	0.05	-4.6	5.8	25.4
1	0.38	0.02	-3.2	3.3	17.4
2	0.06	0.00	-3.1	3.1	14.1
3	0.04	0.01	-3.0	3.0	14.2
4	0.02	0.01	-3.0	3.0	14.3

Y COORDINATE INDEX (J)												
	21	-12.3	- 1.6	1.9	-0.9	-1.3	-1.2	-1.1	-0.8	- 2.2	- 3.3	- 8.4
	19	- 7.1	- 2.8	- 0.1	-0.8	-1.2	-1.2	-1.2	-1.3	- 2.2	- 3.4	- 4.3
	17	- 4.8	- 2.9	- 1.1	-1.0	-1.2	-1.3	-1.3	-1.6	- 2.2	- 2.9	- 2.6
	15	- 3.6	- 2.8	- 1.6	-1.3	-1.3	-1.4	-1.5	-1.7	- 2.2	- 2.4	- 1.7
	13	- 2.9	- 2.7	- 1.9	-1.5	-1.5	-1.5	-1.6	-1.8	- 2.1	- 2.1	- 1.2
	11	- 2.6	- 2.7	- 2.2	-1.8	-1.6	-1.6	-1.7	-1.9	- 2.1	- 2.0	- 1.0
	9	- 2.6	- 2.9	- 2.5	-2.1	-1.8	-1.7	-1.9	-2.1	- 2.3	- 2.1	- 1.1
	7	- 3.2	- 3.4	- 3.1	-2.5	-2.0	-1.9	-2.1	-2.4	- 2.8	- 2.6	- 1.7
	5	- 5.0	- 5.1	- 4.4	-3.3	-2.4	-2.1	-2.4	-3.2	- 4.0	- 4.3	- 3.6
	3	-14.0	-13.4	-10.7	-6.9	-3.6	-2.3	-3.6	-6.9	-10.4	-12.7	-12.8
		22	26	30	34	38	42	46	50	54	58	62

X COORDINATE INDEX (I)

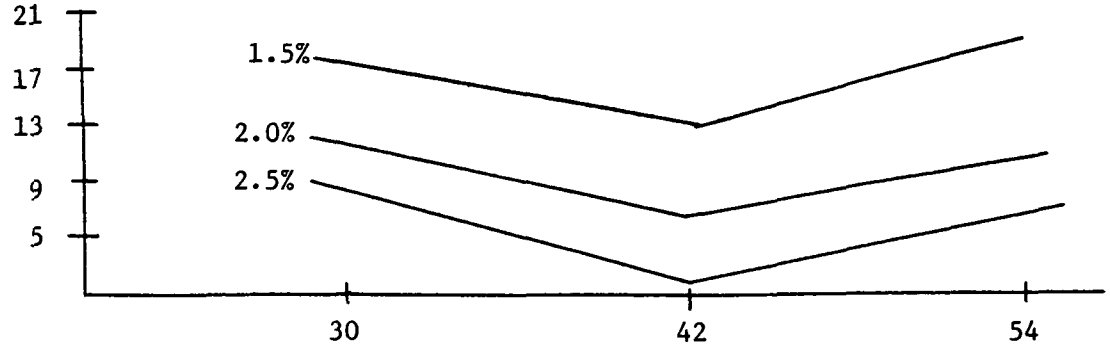


Table 16

Quadrupole Electric Fields at Points Defining Estimated Electron Trajectories in the Uniform Field Potential

	STEP SIZE					
	.008 < d <.025		.013 < d < .025		.013 < d ≤ .050	
STEP	Y	E_x	Y	E_x	Y	E_x
	0.00	0.00	0.00	0.00	0.00	0.00
	0.03	-0.05	0.05	-0.08	0.05	-0.08
	0.07	-0.11	0.10	-0.16	0.10	-0.16
	0.10	-0.16	0.15	-0.24	0.15	-0.24
	0.13	-0.21	0.20	-0.33	0.20	-0.33
	0.16	-0.27	0.24	-0.42	0.24	-0.42
	0.20	-0.33	0.29	-0.54	0.29	-0.54
	0.23	-0.40	0.34	-0.66	0.34	-0.66
	0.26	-0.46	0.39	-0.80	0.39	-0.80
	0.30	-0.54	0.44	-0.97	0.44	-0.97
	0.33	-0.62	0.49	-1.17	0.49	-1.17
	0.36	-0.71	0.54	-1.41	0.54	-1.41
	0.39	-0.81	0.59	-1.73	0.59	-1.73
	0.42	-0.92	0.64	-2.13	0.64	-2.13
	0.46	-1.05	0.69	-2.65	0.69	-2.65
	0.49	-1.19	0.74	-3.33	0.74	-3.33
	0.53	-1.35	0.78	-4.18	0.78	-4.18
	0.56	-1.54	0.83	-5.43	0.83	-5.43
	0.59	-1.76	0.88	-7.13	0.88	-7.13
	0.62	-2.01	0.82	-5.03	0.82	-5.03
	0.65	-2.31	0.73	-3.30	0.65	-2.24
	0.69	-2.68	0.64	-2.18	0.48	-1.13
	0.72	-3.08	0.55	-1.50	0.30	-0.56
	0.75	-3.57	0.46	-1.06	0.12	-0.19
	0.78	-4.17	0.38	-0.75		
	0.82	-4.94	0.29	-0.53		
	0.85	-5.89	0.20	-0.33		
	0.88	-7.04	0.11	-0.17		
	0.84	-5.73	0.02	-0.04		
	0.75	-3.57				
	0.66	-2.38				
	0.57	-1.65				
	0.48	-1.14				
	0.40	-0.82				
	0.30	-0.56				
	0.21	-0.36				
	0.12	-0.20				
	0.04	-0.06				

for field shaping. This trajectory shows no variation in the x coordinate since the estimated potential field used to generate the trajectory from the initial electron data is the uniform field approximation developed as the first operation in all of the iterative routines that have been discussed. Note that decreasing the lower limit on the step size results in more steps as the electron moves against the potential gradient (increasing y coordinate). Note also that increasing the upper limit on the step size results in fewer steps as the electron moves with the potential gradient (decreasing y coordinate). Tables 17 and 18 when compared to Table 15 show that the three different bounds on the step size result in synthesized solutions to the model problem which are only slightly different from each other.

It is interesting to consider the effects of erroneous measurements of initial electron energy or initial electron direction. Tables 19 and 20 show error estimates for the approximations of the biased fin potential which result from erroneous measurements of electron initial energies of 3% and 6% respectively. Note that this type of error affects the field scaling operation but has little effect on the field shaping operations. Tables 21 and 22 show error estimates for approximations of the biased fin potential resulting from erroneous measurements of electron initial directions of 1° and 3° respectively. Note that this type of error severely affects field shaping operations as well as the scaling operations.

Table 17

Error Estimates for Routine QPOLES.F4 with Step Size d

(0.008 < d < 0.025)

Iterations	Discrepancy (DD)	Scale Error (%)	Average Error (%)	Average Absolute Error (%)	Maximum Absolute Error (%)
0	1.51	0.05	-4.6	5.8	20.0
1	0.38	0.02	-3.2	3.3	14.1
2	0.09	0.22	-3.4	3.5	13.9
3	0.04	0.18	-3.3	3.4	14.1
4	0.02	0.21	-3.3	3.4	14.3

Y COORDINATE INDEX (J)												
	21	-12.7	- 2.1	1.5	-1.1	-1.4	-1.3	-1.3	-1.1	- 2.6	- 3.8	- 8.9
	19	- 7.6	- 3.3	- 0.5	-1.0	-1.4	-1.4	-1.4	-1.6	- 2.6	- 3.8	- 4.8
	17	- 5.3	- 3.4	- 1.5	-1.3	-1.4	-1.5	-1.6	-1.9	- 2.6	- 3.3	- 3.0
	15	- 4.0	- 3.2	- 2.0	-1.6	-1.6	-1.6	-1.7	-2.0	- 2.5	- 2.8	- 2.1
	13	- 3.3	- 3.1	- 2.3	-1.8	-1.7	-1.7	-1.9	-2.1	- 2.5	- 2.5	- 1.6
	11	- 3.0	- 3.1	- 2.5	-2.1	-1.9	-1.9	-2.0	-2.2	- 2.5	- 2.3	- 1.4
	9	- 3.0	- 3.2	- 2.8	-2.4	-2.1	-2.0	-2.1	-2.4	- 2.6	- 2.4	- 1.5
	7	- 3.6	- 3.8	- 3.4	-2.8	-2.3	-2.2	-2.4	-2.8	- 3.1	- 3.0	- 2.1
	5	- 5.3	- 5.5	- 4.7	-3.6	-2.7	-2.3	-2.7	-3.5	- 4.4	- 4.7	- 3.9
	3	-14.4	-13.8	-11.0	-7.2	-3.9	-2.5	-3.9	-7.2	-10.7	-13.0	-13.1
		22	26	30	34	38	42	46	50	54	58	62

X COORDINATE INDEX (I)

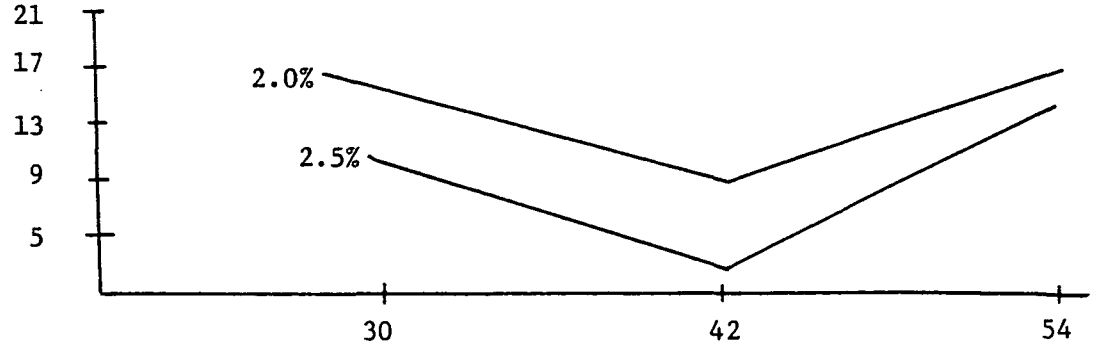


Table 18

Error Estimates for Routine QPOLES.F4 with Step Size d

(0.013 < d < 0.050)

Iterations	Discrepancy (DD)	Scale Error (%)	Average Error (%)	Average Absolute Error (%)	Maximum Absolute Error (%)
0	1.51	0.43	-6.9	6.9	17.4
1	1.01	0.01	-3.2	3.3	14.2
2	0.85	0.02	-3.9	4.3	17.3
3	0.28	0.03	-3.1	3.3	14.8
4	0.08	0.37	-3.3	3.4	15.1

Y COORDINATE INDEX (J)												
	21	-13.7	- 3.3	0.2	-2.0	-1.9	-0.6	-0.5	1.1	- 1.9	- 4.2	- 7.8
	19	- 8.6	- 4.4	- 1.5	-1.8	-1.8	-0.9	-0.6	-0.2	- 1.9	- 3.6	- 3.9
	17	- 6.3	- 4.4	- 2.4	-2.0	-1.8	-1.2	-0.9	-0.9	- 1.8	- 2.8	- 2.2
	15	- 5.0	- 4.2	- 2.9	-2.2	-1.9	-1.4	-1.2	-1.2	- 1.8	- 2.2	- 1.3
	13	- 4.2	- 4.0	- 3.1	-2.4	-2.0	-1.6	-1.4	-1.4	- 1.7	- 1.8	- 0.8
	11	- 3.9	- 3.9	- 3.3	-2.6	-2.1	-1.8	-1.6	-1.6	- 1.8	- 1.6	- 0.6
	9	- 3.9	- 4.1	- 3.6	-2.9	-2.3	-1.9	-1.8	-1.8	- 2.0	- 1.7	- 0.7
	7	- 4.4	- 4.6	- 4.1	-3.3	-2.5	-2.1	-2.0	-2.2	- 2.4	- 2.3	- 1.8
	5	- 6.2	- 6.3	- 5.4	-4.1	-2.9	-2.3	-2.3	-3.0	- 3.7	- 3.9	- 3.2
	3	-15.1	-14.5	-11.7	-7.7	-4.1	-2.5	-3.6	-6.6	-10.1	-12.4	-12.4
		22	26	30	34	38	42	46	50	54	58	62

X COORDINATE INDEX (I)

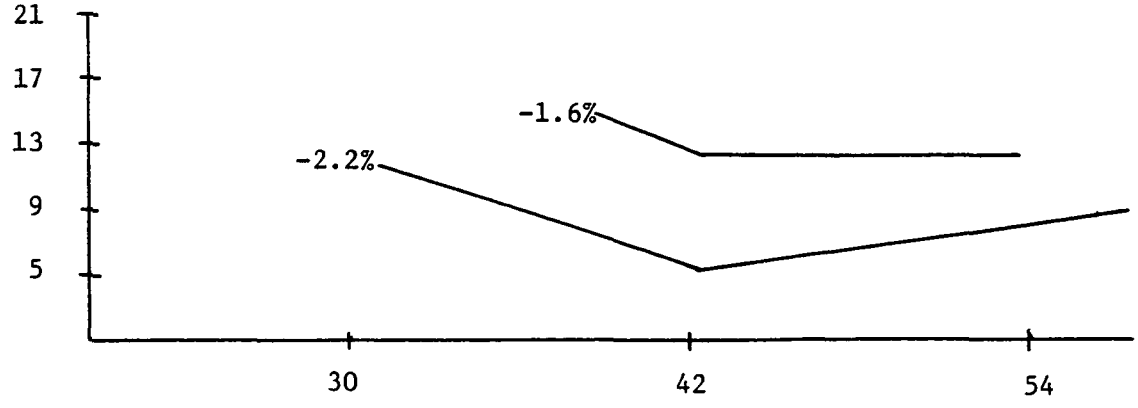


Table 19

Error Estimates for Routine QPOLES.F4 with 3% Error in Beam Energy

Iterations	Discrepancy (DD)	Scale Error (%)	Average Error (%)	Average Absolute Error (%)	Maximum Absolute Error (%)
0	1.51	0.05	-1.7	5.0	17.6
1	0.36	0.02	-0.1	1.8	10.9
2	0.09	0.18	-0.2	1.6	10.7
3	0.03	0.17	-0.2	1.6	11.0
4	0.01	0.18	-0.2	1.6	11.1

Y COORDINATE INDEX (J)												
	21	- 8.9	2.7	5.7	1.4	0.5	0.7	0.5	1.0	0.2	0.0	- 4.7
	19	- 3.6	1.0	3.4	1.8	0.9	0.7	0.6	0.7	0.4	- 0.2	- 0.7
	17	- 1.3	0.6	2.2	1.7	1.0	0.7	0.7	0.6	0.4	0.3	0.9
	15	- 0.1	0.6	1.6	1.5	1.0	0.8	0.7	0.6	0.6	0.7	1.8
	13	0.5	0.7	1.2	1.2	1.0	0.8	0.7	0.6	0.6	1.0	2.2
	11	0.8	0.6	0.9	1.0	0.9	0.8	0.6	0.6	0.6	1.1	2.4
	9	0.7	0.4	0.6	0.7	0.8	0.7	0.5	0.4	0.5	1.0	2.2
	7	0.1	- 0.3	-2.7	0.3	0.6	0.6	0.4	0.1	0.0	0.4	1.6
	5	- 1.8	- 2.0	-1.4	-0.5	0.2	0.4	0.1	-0.7	-1.3	- 1.3	- 0.3
	3	-11.1	-10.6	-8.0	-4.3	-1.0	0.2	-1.2	-4.4	-7.8	-10.0	-10.0
		22	26	30	34	38	42	46	50	52	58	62

X COORDINATE INDEX (I)

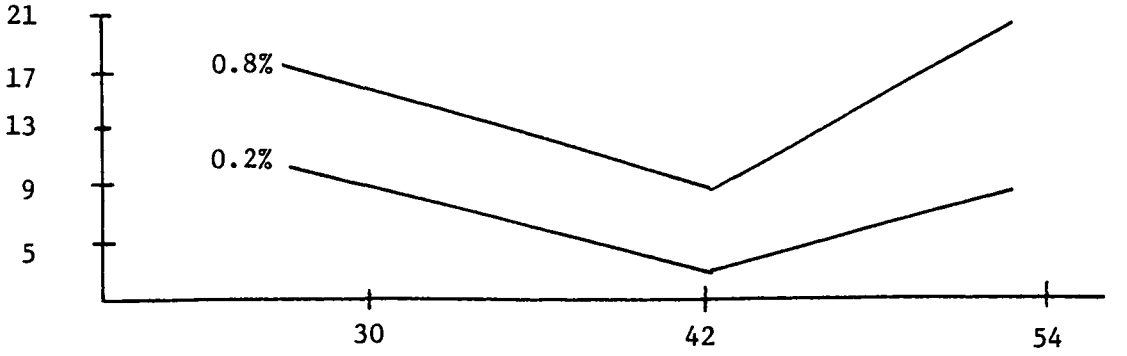


Table 20

Error Estimates for Routine QPOLES.F4 with 6% Error in Beam Energy

Iterations	Discrepancy (DD)	Scale Error (%)	Average Error (%)	Average Absolute Error (%)	Maximum Absolute Error (%)
0	1.51	0.05	1.2	5.2	15.1
1	0.38	0.02	2.7	3.8	8.8
2	0.07	0.16	2.6	3.6	8.5
3	0.04	0.14	2.7	3.7	8.8
4	0.02	0.15	2.7	3.7	9.0

Y COORDINATE INDEX (J)												
	21	-7.2	4.1	7.8	4.9	4.6	4.7	4.9	5.1	3.6	2.3	-3.2
	19	-1.7	2.8	5.7	5.0	4.7	4.7	4.7	4.5	3.5	2.2	1.2
	17	0.7	2.6	4.6	4.8	4.6	4.6	4.5	4.2	3.5	2.7	3.0
	15	2.0	2.8	4.0	4.5	4.5	4.5	4.4	4.1	3.6	3.3	4.0
	13	2.7	3.0	3.8	4.2	4.4	4.4	4.2	4.0	3.6	3.6	4.6
	11	3.1	3.0	3.5	4.0	4.2	4.2	4.1	3.8	3.6	3.8	4.8
	9	3.0	2.8	3.2	3.7	4.0	4.1	3.9	3.7	3.4	3.6	4.7
	7	2.5	2.2	2.6	3.3	3.7	3.9	3.7	3.3	3.0	3.1	4.0
	5	0.6	0.4	1.2	2.4	3.1	3.7	3.1	2.5	1.6	1.3	2.9
	3	-9.0	-8.9	-5.5	-1.5	2.1	3.5	2.1	-1.4	-5.1	-7.6	-7.7
		22	26	30	34	38	42	46	50	54	58	62

X COORDINATE INDEX (I)

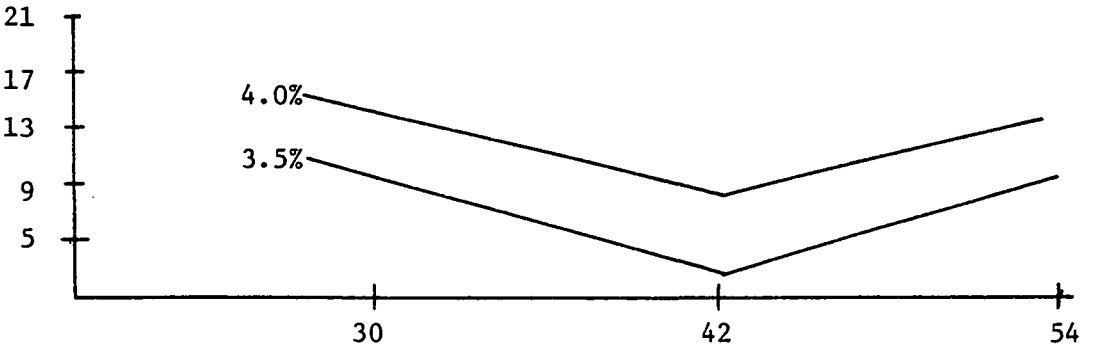


Table 21

Error Estimates for Routine QPOLES.F4
with 1% Error in Initial Electron Direction

Iterations	Discrepancy	Scale Error	Average Error	Average Absolute Error	Maximum Absolute Error
	(DD)	(%)	(%)	(%)	(%)
0	1.57	0.05	-4.9	6.0	20.3
1	0.43	0.01	-3.7	3.8	-15.0
2	0.11	0.37	-4.0	4.0	-15.7
3	0.06	0.38	-3.9	3.9	-16.2
4	0.03	0.44	-4.0	4.0	-16.5

Y COORDINATE INDEX											
21	-16.1	- 4.7	0.1	-1.5	-1.9	-1.7	-1.9	-1.4	- 2.9	- 3.4	- 6.3
19	-10.8	- 5.8	- 2.0	-1.8	-2.0	-1.9	-1.9	-1.9	- 2.6	- 3.1	- 2.7
17	- 8.3	- 5.8	- 3.1	-2.3	-2.1	-2.0	-2.0	-2.0	- 2.4	- 2.4	- 1.3
15	- 6.8	- 5.6	- 3.7	-2.7	-2.3	-2.1	-2.1	-2.1	- 2.2	- 1.9	- 0.6
13	- 6.0	- 5.4	- 4.0	-3.0	-2.5	-2.3	-2.2	-2.1	- 2.1	- 1.6	- 0.2
11	- 5.6	- 5.3	- 4.3	-3.3	-2.7	-2.4	-2.3	-2.2	- 2.1	- 1.5	- 0.1
9	- 5.5	- 5.4	- 4.6	-3.6	-3.0	-2.6	-2.4	-2.4	- 2.2	- 1.6	- 0.3
7	- 6.0	- 5.9	- 5.1	-4.1	-3.2	-2.7	-2.6	-2.7	- 2.7	- 2.2	- 1.0
5	- 7.7	- 7.5	- 6.4	-4.9	-3.6	-2.9	-2.9	-3.5	- 3.9	- 3.9	- 2.9
3	-16.5	-15.6	-12.6	-8.5	-4.8	-3.1	-4.1	-7.1	-10.3	-12.3	-12.2
	22	26	30	34	38	42	46	50	54	58	62

X COORDINATE INDEX (I)

(J)

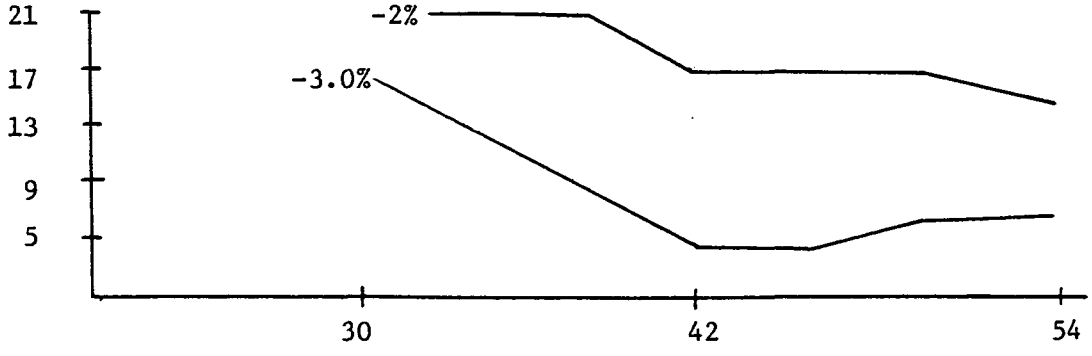


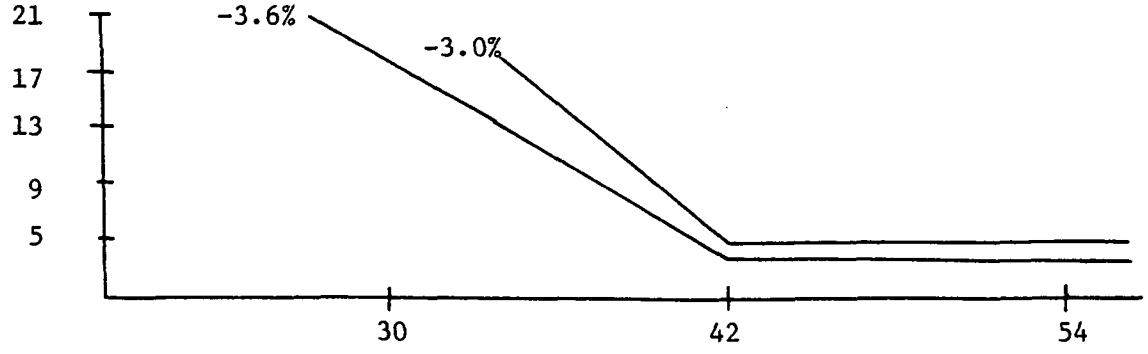
Table 22

Error Estimates for Routine QPOLES.F4
with 3% Error in Initial Electron Direction

Iterations	Discrepancy (DD)	Scale Error (%)	Average Error (%)	Average Absolute Error (%)	Maximum Absolute Error (%)
0	1.83	0.05	-6.0	6.7	21.2
1	0.58	0.04	-4.9	5.0	20.1
2	0.21	0.80	-5.6	5.6	21.9
3	0.13	0.05	-4.3	4.7	21.6
4	0.07	0.31	-4.5	5.1	-22.3

Y COORDINATE INDEX (J)												
	21	-22.3	- 9.6	- 2.3	- 1.6	-2.4	-2.6	-2.7	-2.6	-2.9	-2.1	-0.1
	19	-16.2	-10.3	- 4.6	- 2.8	-2.7	-2.7	-2.6	-2.4	-2.1	-0.8	2.7
	17	-13.1	-10.0	- 5.8	- 3.7	-3.1	-2.8	-2.5	-2.1	-1.4	0.2	3.5
	15	-11.3	- 9.4	- 6.5	- 4.4	-3.4	-2.9	-2.5	-1.9	-1.0	0.8	3.8
	13	-10.1	- 9.0	- 6.8	- 4.9	-3.7	-3.0	-2.4	-1.7	-0.6	1.2	3.9
	11	- 9.5	- 8.7	- 7.0	- 5.3	-4.0	-3.1	-2.5	-1.7	-0.5	1.3	3.8
	9	- 9.2	- 8.8	- 7.3	- 5.6	-4.3	-3.3	-2.5	-1.7	-0.6	1.1	3.4
	7	- 9.6	- 9.2	- 7.8	- 6.0	-4.5	-3.4	-2.7	-2.0	-1.0	0.5	2.6
	5	-11.1	-10.7	- 9.0	- 6.9	-4.9	-3.6	-3.0	-2.7	-2.3	-1.2	0.6
	3	-19.5	-18.5	-15.1	-10.4	-6.1	-3.8	-4.2	-6.3	-8.7	-9.9	-9.0
		22	26	30	34	38	42	46	50	54	58	62

X COORDINATE INDEX (I)



VI. EXPERIMENTAL RESULTS

The biased fin potential referred to in sections III, IV and V of this report has been a conceptual entity rather than a physical entity. The same can be said for the electron trajectories that are used to measure and probe the biased fin potential. In this section, however, we will describe a laboratory experiment that was conducted to test the computer codes that have been developed for the synthesis of approximations to the biased fin potential.

The potential developed in section III was proposed to approximate the field existing near an infinite number of parallel-planar fins, all of which extended infinitely in both parallel directions. Thus, the conceptual system extended infinitely in three spatial directions. However, the physical system was constructed inside a cylindrical bell jar and consisted of six parallel plates approximately five inches high and were biased with -100 volts at a distance of 1.7 inches from a zero potential ground plane. Due to the finite size of the system, we expect that the physical potential field existing inside the evacuated bell jar will be somewhat different from the biased fin potential as derived in section III. In spite of this dimensional limitation, we hoped to use the computer codes described in section V to process electron trajectory data and develop a synthesized potential field for comparision with the biased fin potential plotted in Figure 8.

An electron beam was generated by an electron gun consisting of a wire filament inside a cylindrical aluminum can having a pinhole at the center of one end. The electron gun was positioned below a grounded wire mesh (i.e., the reference plane) so that the electron beam emanating

from the can would pass through the wire mesh on a trajectory that ultimately returned to the wire mesh due to the deflection of the electron beam by the negatively charged fins. The source and sink points in the reference plane were recorded for six trajectories as listed in Table 23. Although the dimensions of the physical system (e.g., 3.17 inches by 1.7 inches) are different from those of the conceptual system (e.g., 1.8 units by 1 unit), it is apparent that the trajectories listed as physical data do not correspond to those trajectories defined conceptually in the statement of the model problem. Thus, it is interesting to speculate what the electron trajectories might have looked like in the conceptual biased fin potential given the initial electron parameters measured as physical data. Figure 15 shows projected trajectories developed for an assumption of the biased fin potential as expressed in equation (III-10) and for the initial electron velocities listed in Table 23. Note that the sink points along the reference plane do not correspond to the sink points measured as physical data. Therefore, we conclude that the physical potential field of the biased fins must be somewhat different from the conceptualized potential. With the knowledge that the unknown potential field in the bell jar is similar to, but not identical to, the conceptualized biased fin potential; it is appropriate to apply the computerized methods described in section V in order to synthesize approximations of the unknown potential. Figure 16 shows the results obtained with five iterations of the QPOLE.F4 routine using all the data that appears in Table 23. Note that this solution indicated a lower potential midway between the fins when compared to the potential field of Figure 8. This may well be the result of approximating an infinite fin height (measured

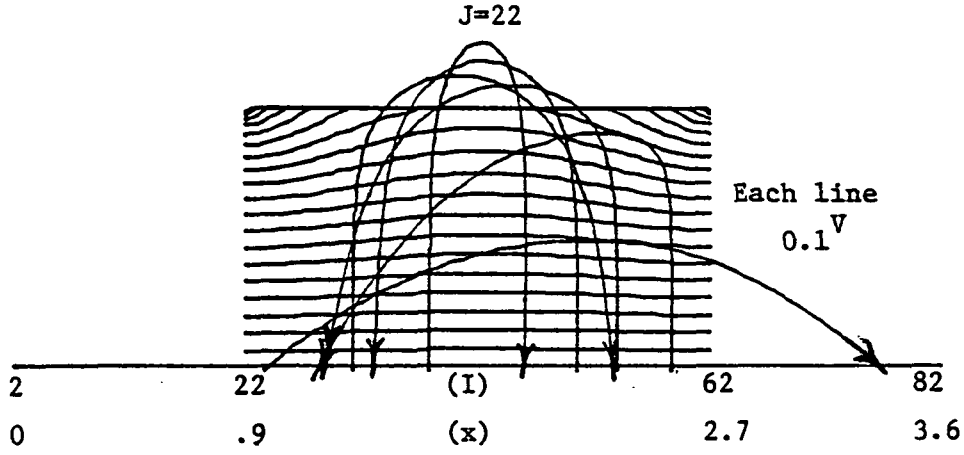


Figure 15. Trial Electron Trajectories for Experimental Electron Data and the Ideal Biased Fin Potential

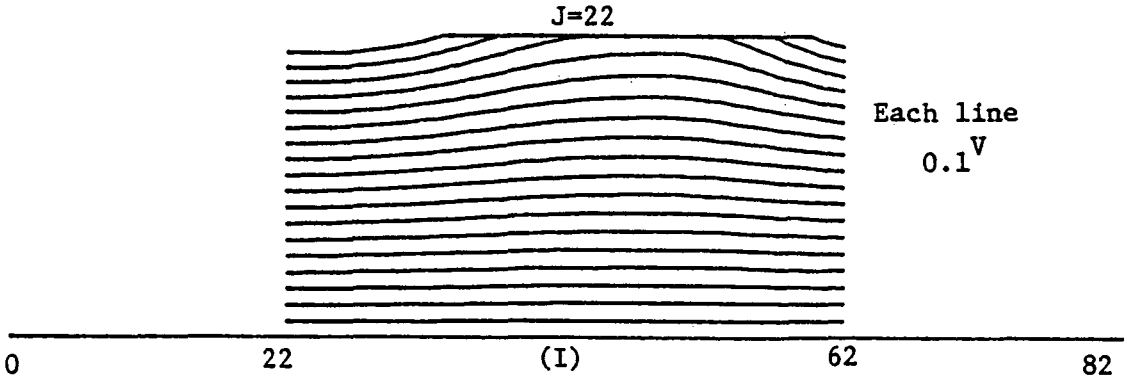


Figure 16. Approximation of the Experimental Potential Synthesized with QPOLE.F4

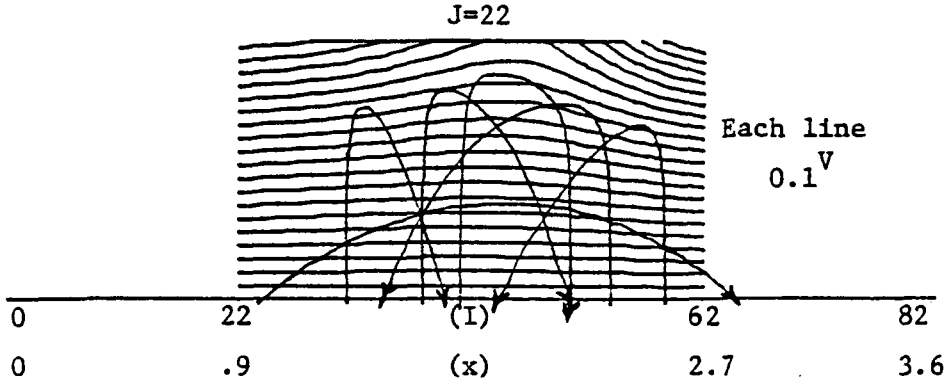


Figure 17. Approximation of the Experimental Potential Synthesized with QPOLES.F4

perpendicular to the reference plane) with a fin only 5 inches high.

Figure 17 shows the results obtained when only four of the trajectories (i.e., K = 2,3,4 and 6) are used with the QPOLES.F4 routine. Attempts to use more of the data in Table 23 caused instabilities in the method preventing the synthesis of approximations to the potential field existing in the bell jar. Thus, by concentrating only on the interior trajectories, a potential field can be created which reproduces the sink points of those interior trajectories. However, as seen in Figure 17, the sink points for the K=1 and K=5 trajectories are in error since the data for these trajectories is not used.

Thus, we have demonstrated that the quadrupole methods developed for an idealized biased fin potential and a particular set of trajectory data can be used to synthesize approximations of the similar potential field existing in the bell jar by using a different set of trajectory data. We see that the choice of data for a particular scheme does affect the convergence properties of that scheme, to the extent that no solution is a possibility. Therefore, we conclude that data selection is important for the application of these iterative routines. Conversely, the definition of a particular iterative scheme may depend heavily on the data that can be obtained.

Table 23

Experimental Electron Trajectory Data

Trajectory	Source Points		Source Velocities		Sink Points	
K	XO(K) (inches)	YO(K) (inches)	VXO(K) $(\frac{m}{sec})$	VYO(K) $(\frac{m}{sec})$	XT(K) (inches)	YT(K) (inches)
1	0.73	0.0	0.0	5.5×10^6	2.72	0.0
2	1.25	0.0	0.0	5.5×10^6	2.29	0.0
3	2.27	0.0	0.0	5.5×10^6	1.52	0.0
4	2.54	0.0	0.0	5.5×10^6	0.96	0.0
5	2.91	0.0	0.0	5.5×10^6	0.33	0.0
6	0.12	0.0	3.4×10^6	3.4×10^6	3.29	0.0

VII. CONCLUSIONS

The synthesis of electric potentials from the fields of electric dipoles and quadrupoles has been considered as a method for approximating unknown potential fields in two dimensions. Although we were most interested in determining potential fields on an unbounded spatial domain; computational methods were developed first for the classical boundary value problem in electrostatics, since other well-known methods exist for its solution.

Several iterative schemes for the dipole synthesis of solutions to a square boundary value problem were developed and their accuracy was compared to a series expansion solution. We found that as the dipole singularities were moved farther away from the boundary, the synthesized solutions became more accurate at the expense of more computation (i.e., more iterations of a scheme were required for convergence to a desired limit). Additionally, procedures for scaling, shifting and rotating the estimated potential were developed to assist in the solution of the boundary value problem. The rotational procedure was found to be of assistance only in determining an initial estimated potential that best matched the known boundary values. The scaling and shifting procedures were found to be helpful immediately after each successive addition of a dipole field. The dipole fields were chosen to reduce the discrepancy between a known potential value and its estimated value at the point on the boundary where the discrepancy was largest. This scheme showed good convergence properties. We found that convergence could be hastened by considering potential discrepancy reduction at two boundary points simultaneously and then allowing the

location of dipole field singularities at variable distances from the boundary. Extensions of this work on the boundary value problem could be interesting if quadrupole synthesis is explored and if other boundary shapes are considered.

Both dipole and quadrupole methods have been developed for the synthesis of unknown potential fields in two dimensions of an unbounded spatial domain. Our first step was to introduce the concept of a reference plane existing near the surface charge distribution responsible for the potential field existing in some spatial region of interest. The reference plane represents an artificial boundary condition, since we have required a potential of zero on it in order to allow the presence of a measuring device (i.e., an electron beam probe) in the vicinity of the charged surface. Charged particles emanate from the reference plane, they are deflected by the potential field of a charge distribution and fall back to the reference plane. The energy of the charged particle and it's initial and final coordinate values have been used as constraints upon acceptable dipole and quadrupole synthesized approximations to the potential resulting from the charge distribution. Dipole and quadrupole fields have been added as image pairs to insure a zero potential on the reference plane.

Basic schemes have been developed for the dipole synthesis of electric potentials as constrained by electron trajectory data. A uniform field approximation is determined so as to meet the constraining data for a particular trajectory called the scaling trajectory. Then a dipole field is added in order to meet the constraints of two other trajectories while keeping the dipole field singularity at a fixed distance from the reference plane. A potential shifting operation has

not been required since the dipole fields have been added in a manner that kept the potential at zero on the reference plane. However, potential scaling has been found to be beneficial after the addition of each dipole field in order to meet the constraints of the scaling trajectory. Convergence limits for this scheme have been adequate, although for the model problem, the synthesized solution overestimated the test potential. We found that an underestimation of the test potential could be obtained by altering the criterion used for determining which two sets of trajectory constraints are satisfied when adding a dipole field. Therefore, we were able to establish bounds on the unknown potential created by the surface charge distribution of the model problem. We also found that convergence to a desired accuracy could be hastened by allowing the location of the dipole field singularities at a variable distance from the reference plane.

Quadrupole fields were also used for synthesis of solutions to the model problem. When used in the same way as the dipole fields, we found that better convergence limits were achieved with about the same amount of computation time. In addition, we were able to develop procedures for the addition of one quadrupole field for each trajectory during a single iteration of a scheme designed to meet the constraints of all the trajectories, excepting the scaling trajectory, simultaneously. This scheme reduced by a factor of 5 (i.e., the number of trajectories considered), the number of iterations and the amount of computational time required to obtain a desired accuracy. Future possibilities for extending this work include the development of schemes that combine the scaling operation and the quadrupole synthesis operation into a single operation which meets the constraints for all of the the trajectories.

This would not alter the speed of the procedures appreciably but might improve the scale accuracy. Also of interest are schemes which combine the dipole and quadrupole methods for potential field synthesis.

Finally, a test potential was created in the laboratory to test the computer codes developed for the synthesis of unknown potentials. An electron-beam probe was used to collect constraining electron trajectory data. Potentials were synthesized which seemed like reasonable approximations considering that the biased-fin potential that we attempted to create in the laboratory was undoubtably altered somewhat by the finite size of the physical apparatus. We learned that data selection is important for the best determination of unknown potentials when applying the particular codes that we have developed. Therefore, the challenge in the area of dipole and quadrupole field synthesis lies in the determination of optimal schemes depending on whatever data are available.

References

1. W. Panofsky and M. Phillips, Classical Electricity and Magnetism, p. 15, Addison-Wesley Inc., Reading, Massachusetts, 1955.

2. J. Marion, Classical Electromagnetic Radiation, pp. 53-93, Academic Press Inc., New York and London, 1965.

3. E. Isaacson and H. Keller, Analysis of Numerical Methods, Wiley, New York, 1966.

4. W. Black and J. Robinson, "Measuring Rotationally Symmetric Potential Profiles with an Electron-Beam Probe," Journal of Applied Physics, Vol. 45, No. 6, pp. 2497-2501, June, 1974.

5. H. Kober, Dictionary of Conformal Representations, p. 118, Dover Press Inc., New York, 1957.

6. A. Booth, Numerical Methods, 3rd Edition, p. 71, Butterworths, London, 1966.

7. J. Steffensen, Interpolation, pp. 14-34, 203-224, Chelsea Publishing Company, New York, 1950.

8. Chemical Rubber Company, Handbook of Chemistry and Physics, 59th Edition, p. A-54, eq. 165, West Palm Beach, Florida, 1978.

APPENDIX A

Calculating Potential Fields for Ideal Line Dipoles and Quadrupoles

To determine the potential field ϕ_d existing near an ideal line dipole located at point (x_d,y_d), consider a line of point dipoles extending to infinity in the positive and negative z-directions as in Figure A1. At any given distance z_1 from the x-y plane, there exists an electric dipole contributing a potential field,

$$\phi_d(x,y,z_1) = A_d \frac{\cos \beta(x,y)}{b^2(x,y,z_1)}, \tag{A-1}$$

at all points in the x-y plane (ref. 1). The electric potential at an arbitrary point (x,y) due to the ideal line dipole

$$\phi_d(x,y) = \int_{-\infty}^{\infty} \phi_d(x,y,z_1)dz_1 \tag{A-2}$$

is found by integrating the potential field due to an electric dipole at distance z_1 from the x-y plane over all values of z_1 from $-\infty$ to $+\infty$. Note that

$$b^2(x,y,z_1) = z_1^2 + (x-x_d)^2 + (y-y_d)^2 \tag{A-3}$$

and

$$\cos \beta(x,y) = \frac{(y-y_d)}{[z_1^2 + (x-x_d)^2 + (y-y_d)^2]^{1/2}}. \tag{A-4}$$

Therefore,

$$\phi_d(x,y) = A_d(y-y_d) \int_{\infty}^{\infty} \frac{dz_1}{[z_1^2 + (x-x_d)^2 + (y-y_d)^2]^{3/2}}. \tag{A-5}$$

Evaluating the integral, we find

$$\phi_d(x,y) = \frac{A_d(y-y_d)}{(x-x_d)^2 + (y-y_d)^2} \tag{A-6}$$

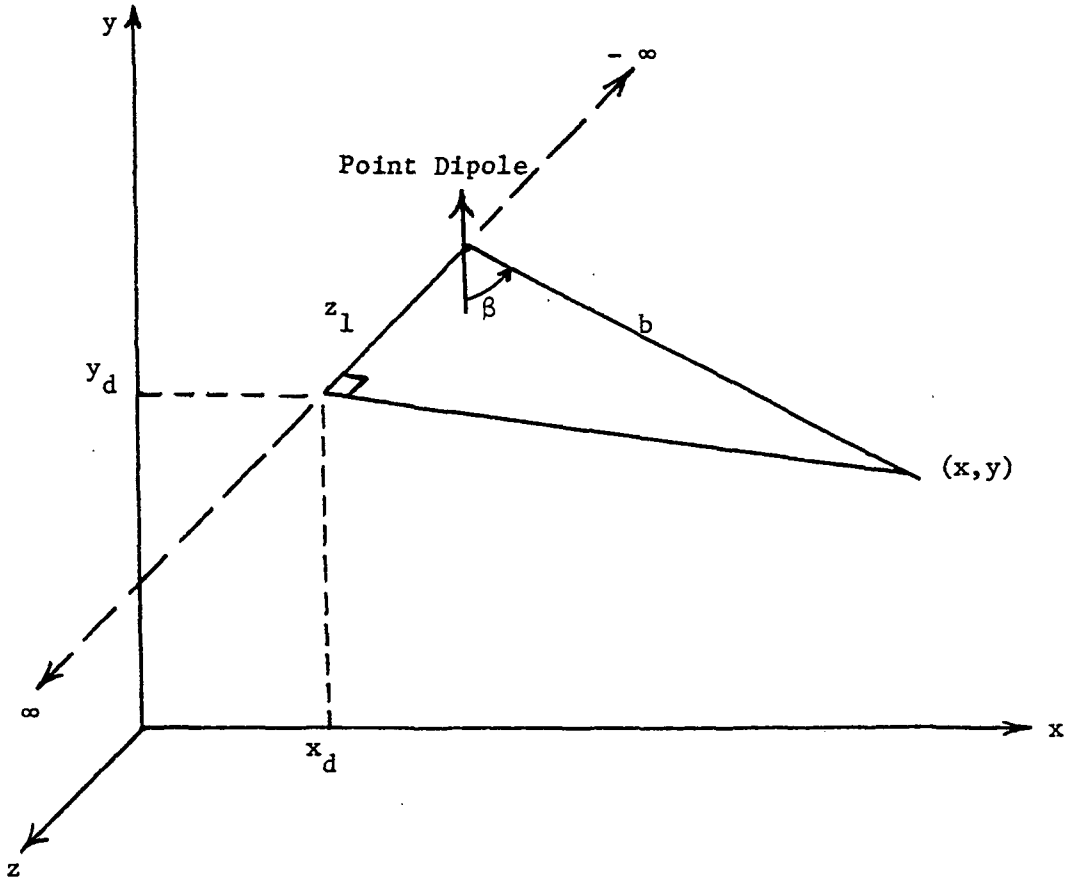


Figure A1. X-Y Reference Frame for the Ideal Line Dipole Consisting of an Infinite Number of Ideal Point Dipoles

with the aid of CRC Integral Tables (ref. 8). Then, if an image line dipole is also present at point $(x_d, -y_d)$, the combined electric potential at point (x,y) is

$$\phi_d(x,y) = A_d\left[\frac{(y-y_d)}{(x-x_d)^2 + (y-y_d)^2} + \frac{(y+y_d)}{(x-x_d)^2 + (y+y_d)^2}\right]. \quad \text{(A-7)}$$

In a similar manner, the potential ϕ_q existing near an ideal line quadrupole may be considered as an integral of the field due to a point quadrupole.

$$\phi_q(x,y,z_1) = a_q \frac{\sin \beta(x,y) \cos \beta(x,y)}{b^3(x,y,z_1)} \quad \text{(A-8)}$$

located a distance z_1 from the x-y plane, over all values from $-\infty$ to ∞.

$$\phi_q(x,y) = \int_{-\infty}^{\infty} \phi_q(x,y,z_1)dz_1 \quad \text{(A-9)}$$

Using equations (A-3) and (A-4), along with

$$\sin \beta(x,y) = \frac{(x-x_d)}{[z_1^2 + (x-x_d)^2 + (y-t_d)^2]1/2}, \quad \text{(A-10)}$$

we obtain the integral expression;

$$\phi_q(x,y) = a_q(x-x_d)(y-y_d) \int_{-\infty}^{\infty} \frac{dz_1}{[z_1^2 + (x-x_d)^2 + (y-y_d)^2]5/2}. \quad \text{(A-11)}$$

Evaluating the integral, we find

$$\phi_q(x,y) = \frac{A_q(x-x_d)(y-y_d)}{[(x-x_d)^2 + (y-y_d)^2]2}, \quad \text{(A-12)}$$

where $A_q = \frac{2}{3} a_q$. Then, if an image line dipole is also present at point $(x_d, -y_d)$, the combined electric quadrupole potential at point (x,y) is

$$\phi_q(x,y) = A_q(x-x_d) \left[\frac{(y-y_d)}{[(x-x_d)^2 + (y-y_d)^2]^2} + \frac{(y+y_d)}{[(x-x_d)^2 + (y+y_d)^2]^2}\right]. \quad \text{(A-13)}$$

For an r-θ coordinate system, consider the potential field due to an ideal line dipole intersecting the r-θ plane at the point (r_d,θ_d) and oriented so that the axes of all the point dipoles comprising the line dipole extend through points on a perpendicular line at the origin of the r-θ plane. This situation is shown in Figure A2. The point dipole located a distance z_1 from the r-θ plane contributes a potential field

$$\phi_d(r,\theta,z_1) = A_d \frac{\cos \beta(r,\theta)}{b^2(r,\theta,z_1)}, \quad \text{(A-14)}$$

at all points (r_m,θ_n) in the r-θ plane. The electric potential due to the entire line dipole

$$\phi_d(r,\theta) = \int_{-\infty}^{\infty} \phi_d(r,\theta,z_1)dz_1 \quad \text{(A-15)}$$

is found by integrating equation (A-14) for all point dipoles located along the line (i.e. $-\infty < z_1 < \infty$).

Applying the Pythagorean Theorem, $b^2 = z_1^2 + d^2$, and the law of cosines,

$$d^2 = r^2 + r_d^2 - 2rr_d\cos(\theta_d-\theta), \quad \text{(A-16)}$$

and recognizing in Figure A2 that

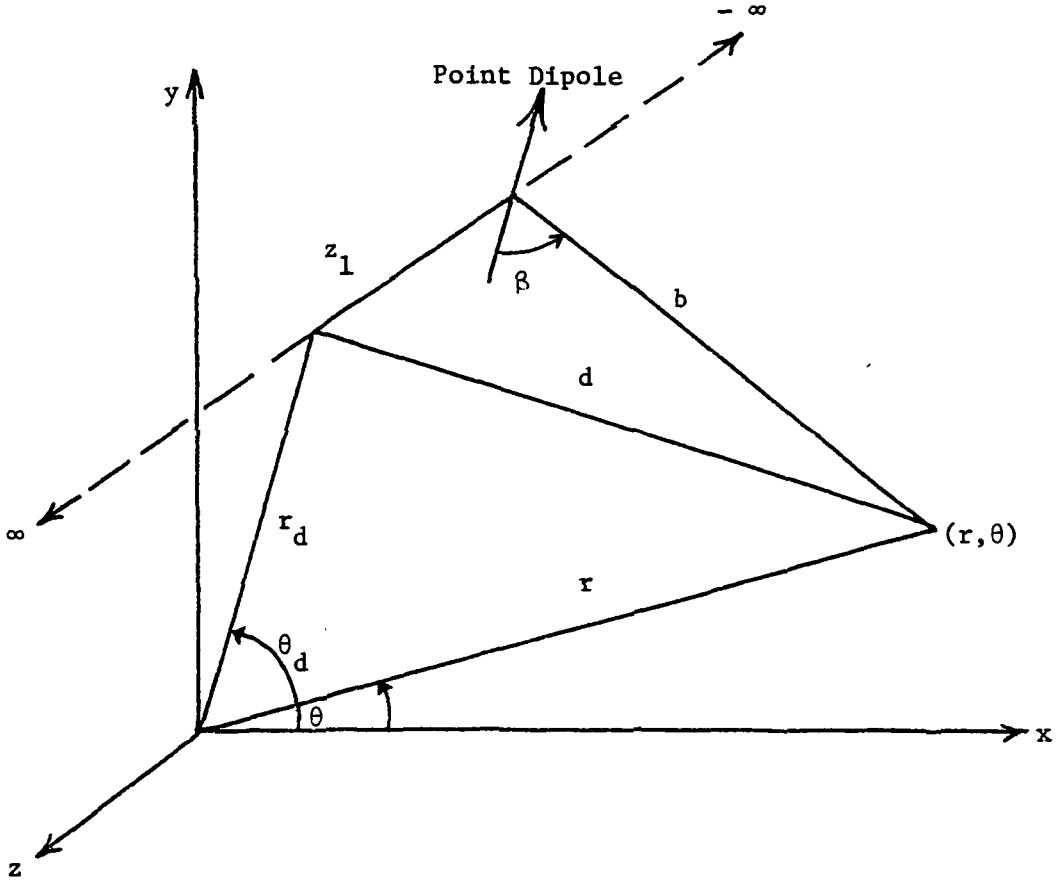


Figure A2. R-θ Reference Frame for The Ideal Line Dipole Consisting of an Infinite Number of Ideal Point Dipoles

$$\cos \beta = \frac{r_d - r \cos(\theta_d - \theta)}{b}; \tag{A-17}$$

we derive

$$\phi_d(r,\theta) = \int_{-\infty}^{\infty} \frac{A_d[r_d - r \cos(\theta_d - \theta)]}{(d^2 + z_1^2)^{3/2}} dz_1. \tag{A-18}$$

Then the integral may be evaluated with the result

$$\phi_d(r,\theta) = \frac{A_d[r_d - r \cos(\theta_d - \theta)]}{r_d^2 + r^2 - 2rr_d \cos(\theta_d - \theta)}. \tag{A-19}$$

APPENDIX B

Fortran Routines for Dipole Synthesis of Solutions to The Boundary Value Problem

```
cROUTINE VMODGR.F4
      IMPLICIT REAL *8(A-H,O-Z)
      COMMON/MVD/XB(40),YB(40),PPHI(40),IB,F,S
      COMMON/ALL/PSI(80,40),PHI(40),M,N
      COMMON/MVPA/R(80),THETA(40)

      IT=0
      CALL DATA
      XMAX=-1.0D38
      XMIN=1.0D38
      YMAX=-1.0D38
      YMIN=1.0D38
      XB(IB+1)=XB(1)
      YB(IB+1)=YB(1)
      XL=0.0
      YL=0.0
      AL=0.0
      DO 120 K=1,IB
      IF(XB(K).GT.XMAX) XMAX=XB(K)
      IF(XB(K).LT.XMIN) XMIN=XB(K)
      IF(YB(K).GT.YMAX) YMAX=YB(K)
      IF(YB(K).LT.YMIN) YMIN=YB(K)
      IF(XB(K+1).NE.XB(K)) GO TO 100
      XL=XL+XB(K)*DABS(YB(K+1)-YB(K))
      YL=YL+(YB(K+1)+YB(K))*0.5*DABS(YB(K+1)-YB(K))
      AL=AL+DABS(YB(K+1)-YB(K))
      GO TO 120
100   IF(YB(K+1).NE.YB(K)) GO TO 110
      XL=XL+(XB(K+1)+XB(K))*0.5*DABS(XB(K+1)-XB(K))
      YL=YL+YB(K)*DABS(XB(K+1)-XB(K))
      AL=AL+DABS(XB(K+1)-XB(K))
      GO TO 120
110   TANB=(YB(K+1)-YB(K))/(XB(K+1)-XB(K))
      XL=XL+(1+TANB**2)**0.5*0.5*(XB(K+1)**2-XB(K)**2)
      YL=YL+(1+1/TANB**2)**0.5*0.5*(YB(K+1)**2-YB(K)**2)
      AL=AL+((XB(K+1)-XB(K))**2+(YB(K+1)-YB(K))**2)**0.5
120   CONTINUE
      XO=XL/AL
      YO=YL/AL
      TYPE 125,XO,YO
125   FORMAT(' XO=',D,'YO=',D)
      PHIMAX=-1.0D38
      PHIMIN=1.0D38
      RMAX=0.0
      PI=4*DATAN(1.D0)
      DO 290 J=1,N
      JK=0
      THETA(J)=2*PI*J/N
160   IF(DABS(DSIN(THETA(J))).LE.1.D-10) GO TO 170
      IF(DABS(DCOS(THETA(J))).LE.1.D-10) GO TO 180
      TANJ=DSIN(THETA(J))/DCOS(THETA(J))
      GO TO 190
170   JK=1
```

```
      GO TO 190
180   JK=-1
190   DO 280 K=1,IB
      THETA1=DATAN2(YB(K+1)-Y0,XB(K+1)-X0)
      THETA2=DATAN2(YB(K)-Y0,XB(K)-X0)
      IF(THETA1.GE.0.) GO TO 195
      THETA1=THETA1+2*PI
195   IF(THETA2.GE.0.) GO TO 197
      THETA2=THETA2+2*PI
197   IF(THETA1.GT.THETA2) GO TO 198
      IF(THETA(J).LE.THETA1) GO TO 280
      IF(THETA(J).GT.THETA2) GO TO 280
      GO TO 888
198   IF(THETA(J).LT.PI) GO TO 199
      IF(THETA(J).LE.THETA1) GO TO 280
      GO TO 888
199   IF(THETA(J).GT.THETA2) GO TO 280
888   IF(DABS(XB(K)-XB(K+1)).LE.1.D-10) GO TO 230
      IF(DABS(YB(K)-YB(K+1)).LE.1.D-10) GO TO 240
      IF(JK) 210,220,200
200   Y=Y0
      X=XB(K)-(YB(K)-Y0)*(XB(K)-XB(K+1))/(YB(K)-YB(K+1))
      GO TO 270
210   X=X0
      Y=YB(K)-(XB(K)-X0)*(YB(K)-YB(K+1))/(XB(K)-XB(K+1))
      GO TO 270
220   YN=YB(K)*(XB(K)-XB(K+1))/(YB(K)-YB(K+1))-XB(K)+X0+Y0/TANJ
      YD=(XB(K)-XB(K+1))/(YB(K)-YB(K+1))
      Y=YN/YD
      X=X0+(Y-Y0)/TANJ
      GO TO 270
230   X=XB(K)
      IF(JK) 280,236,232
232   Y=Y0
      GO TO 270
236   Y=Y0+(X-X0)*TANJ
      GO TO 270
240   Y=YB(K)
      IF(JK) 244,246,280
244   X=X0
      GO TO 270
246   X=X0+(Y-Y0)/TANJ
270   R(J)=(X**2+Y**2)**0.5
      IF(R(J).GT.RMAX) RMAX=R(J)
      PHI(J)=PPHI(K)
      IF(PHI(J).LT.PHIMIN) PHIMIN=PHI(J)
      IF(PHI(J).GT.PHIMAX) PHIMAX=PHI(J)
280   CONTINUE
290   CONTINUE
      M=2*N
      IN=N+1
      DO 291 I=IN,M
      R(I)=(I-N)*RMAX/(M-N)
```

```
291     CONTINUE
        DO 295 J=1,N
        DO 294 I=1,M
        PSI(I,J)=(PHIMAX-PHIMIN)*R(I)*DCOS(THETA(J))/(2*RMAX)+(PHIMAX+PHI
     1           MIN)/2
294     CONTINUE
295     CONTINUE
        CALL ROTATE
        GO TO 301
300     CALL SHIFT
301     CALL SCALE
        DAN=0.0
        DO 310 J=1,N
        DAN=DAN+DABS(PHI(J)-PSI(J,J))
310     CONTINUE
        IF(DAN.LE.0.00001) GO TO 320
        CALL TADPOLE(F,S)
        IT=IT+1
        GO TO 300
320             PAUSE
       CALL INITT
       CALL DWINDO(SNGL(XMIN),SNGL(XMAX),SNGL(YMIN),SNGL(YMAX))
        CALL TWINDO(0,1023,0,780)
        CALL ERASE
        CALL VGRAPH(XO,YO,PHIMIN,PHIMAX)
        PAUSE
        CALL PRINT(IT)
        STOP
330     TYPE 340
340     FORMAT(' THE POINT XO,YO LIES ON THE BOUNDARY')
        CALL FINITT
        STOP
        END
```

```
      SUBROUTINE DATA
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/MVD/XB(40),YB(40),PPHI(40),IB,F,S
      COMMON/ALL/PSI(80,40),PHI(40),M,N
      TYPE 1
 1    FORMAT(' ENTER NUMBER OF ANGULAR GRID LINES')
      ACCEPT 2,N
 2    FORMAT(I)
      TYPE 3
 3    FORMAT(' ENTER NUMBER OF STRAIGHT BOUNDARY SEGMENTS')
      ACCEPT 2,IB
      DO 8 K=1,IB
      TYPE 4,K
 4    FORMAT(' ENTER X-COORD. AT BEGINNING OF SEGMENT ',I3)
      ACCEPT 5,XB(K)
 5    FORMAT(D)
      TYPE 6,K
 6    FORMAT(' ENTER Y-COORD. AT BEGINNING OF SEGMENT ',I3)
      ACCEPT 5,YB(K)
      TYPE 7,K
 7    FORMAT(' ENTER POTENTIAL ON BOUNDARY SEGMENT ',I3)
      ACCEPT 5,PPHI(K)
 8    CONTINUE
      TYPE 28
28    FORMAT(' ENTER DIPOLE G PARAMETER')
      ACCEPT 5,F
      TYPE 29
29    FORMAT(' ENTER DIPOLE T PARAMETER')
      ACCEPT 5,S
      RETURN
      END
```

```
      SUBROUTINE ADPOLE(T)
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/ALL/PSI(80,40),PHI(40),M,N
      COMMON/MVPA/R(80),THETA(40)
      DIMENSION D(45)
      DAN=0.0
      DMAX=0.0
      DO 41 J=1,N
      D(J)=PHI(J)-PSI(J,J)
      DAN=DAN+DABS(D(J))
      IF(DABS(D(J)).LE.DMAX) GO TO 41
      DMAX=DABS(D(J))
      JD=J
 41   CONTINUE
      AVDAN=DAN/N
      KL=0
      JN=JD+N
      DO 44 J=JD,JN
      IF(J.GT.N) GO TO 42
      JL=J
      GO TO 43
 42   JL=J-N
 43   IF(D(JL)/D(JD).GT.0.0) GO TO 44
      KL=KL+1
      IF(KL.EQ.1) L=JL
      LL=JL
 44   CONTINUE
      IF(L.GT.JD) GO TO 45
      IL=JD-L
      GO TO 46
 45   IL=L-JD
 46   IF(LL.GT.JD) GO TO 805
      ILL=JD-LL
      GO TO 806
805   ILL=LL-JD
806   IF(IL.GT.ILL) GO TO 807
      JC=L
      GO TO 47
807   JC=LL
 47   A=DABS(D(JD))-T*AVDAN
      B=R(JC)*DCOS(THETA(JC)-THETA(JD))*(2*T*AVDAN-DABS(D(JD)))-
     1 R(JD)*DABS(D(JD))
      C=R(JD)*R(JC)*DCOS(THETA(JC)-THETA(JD))*DABS(D(JD))-
     1 R(JC)**2*T*AVDAN
      IF(B**2-4*A*C.LT.0.0) GO TO 60
      IF(B.LT.0.0) GO TO 48
      RD1=(2*C)/(-B-(B**2-4*A*C)**0.5)
      RD2=(-B-(B**2-4*A*C)**0.5)/(2*A)
      GO TO 49
 48   RD1=(-B+(B**2-4*A*C)**0.5)/(2*A)
      RD2=(2*C)/(-B+(B**2-4*A*C)**0.5)
 49   IF(RD1.LE.R(JD)) GO TO 50
      RD=RD1
```

```
      GO TO 51
 50   IF(RD2.LE.R(JD)) GO TO 61
      RD=RD2
 51   IF(RD.LT.R(JD)*(1.0+6.2835/N)) RD=R(JD)*(6.2835/N+1.0)
      AD=(RD-R(JD))*D(JD)
      TYPE 515,RD,THETA(JD)
515   FORMAT('   RD=',D,'   THETA=',D)
      M=2*N
      DO 53 J=1,N
      DO 52 I=1,M
      PSI(I,J)=PSI(I,J)+AD*(RD-R(I)*DCOS(THETA(J)-THETA(JD)))/(RD**2
     1          +R(I)**2-2*RD*R(I)*DCOS(THETA(J)-THETA(JD)))
 52   CONTINUE
 53   CONTINUE
      RETURN
 60   TYPE 62
 62   FORMAT(' DIPOLE RADIUS INCLUDES IMAGINARY ROOT')
      RETURN
 61   TYPE 63
 63   FORMAT(' DIPOLE LOCATED INSIDE BOUNDARY')
      RETURN
      END
```

```
      SUBROUTINE TADPOLE(G,T)
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/ALL/PSI(80,40),PHI(40),M,N
      COMMON/MVPA/R(80),THETA(40)
      DIMENSION D(45)
      DAN=0.0
      DMAX=0.0
      DO 41 J=1,N
      D(J)=PHI(J)-PSI(J,J)
      DAN=DAN+DABS(D(J))
      IF(DABS(D(J)).LE.DMAX) GO TO 41
      DMAX=DABS(D(J))
      JD=J
 41   CONTINUE
      AVDAN=DAN/N
      KL=0
      JN=JD+N
      DO 44 J=JD,JN
      IF(J.GT.N) GO TO 42
      JL=J
      GO TO 43
42    JL=J-N
43    IF(D(JL)/D(JD).GT.0.0) GO TO 44
      KL=KL+1
      IF(KL.EQ.1) L=JL
      LL=JL
44    CONTINUE
      IF(L.GT.JD) GO TO 45
      IL=JD-L
      GO TO 46
45    IL=L-JD
46    IF(LL.GT.JD) GO TO 805
      ILL=JD-LL
      GO TO 806
805   ILL=LL-JD
806   IF(IL.GT.ILL) GO TO 807
      JC=L
      GO TO 47
807   JC=LL
47    A=DABS(D(JD))-T*AVDAN
      B=R(JC)*DCOS(THETA(JC)-THETA(JD))*(2*T*AVDAN-DABS(D(JD)))-
     1 R(JD)*DABS(D(JD))
      C=R(JD)*R(JC)*DCOS(THETA(JC)-THETA(JD))*DABS(D(JD))-
     1 R(JC)**2*T*AVDAN
      IF(B**2-4*A*C.LT.0.0) GO TO 60
      IF(B.LT.0.0) GO TO 48
      RD1=(2*C)/(-B-(B**2-4*A*C)**0.5)
      RD2=(-B-(B**2-4*A*C)**0.5)/(2*A)
      GO TO 49
 48   RD1=(-B+(B**2-4*A*C)**0.5)/(2*A)
      RD2=(2*C)/(-B+(B**2-4*A*C)**0.5)
 49   IF(RD1.LE.R(JD)) GO TO 50
      RD=RD1
```

```
      GO TO 51
 50   IF(RD2.LE.R(JD)) GO TO 61
      RD=RD2
51    IF(RD.LT.R(JD)*(1.0+G/N)) RD=R(JD)*(G/N+1.0)
      AD=(RD-R(JD))*D(JD)
      TYPE 515,RD,THETA(JD)
515   FORMAT('   RD=',D,'   THETA=',D)
      M=2*N
      DO 53 J=1,N
      DO 52 I=1,M
      PSI(I,J)=PSI(I,J)+AD*(RD-R(I)*DCOS(THETA(J)-THETA(JD)))/(RD**2
     1          +R(I)**2-2*RD*R(I)*DCOS(THETA(J)-THETA(JD)))
 52   CONTINUE
 53   CONTINUE
      RETURN
 60   TYPE 62
 62   FORMAT(' DIPOLE RADIUS INCLUDES IMAGINARY ROOT')
      RETURN
 61   TYPE 63
 63   FORMAT(' DIPOLE LOCATED INSIDE BOUNDARY')
      RETURN
      END
```

```
      SUBROUTINE SHIFT
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ALL/PSI(80,40),PHI(40),M,N
      DIMENSION QPSI(40)
      DON=0.0
      DO 85 J=1,N
      DON=DON+PHI(J)-PSI(J,J)
85    CONTINUE
      ADON=DON/N
      DO 88 I=1,M
      DO 86 J=1,N
      QPSI(J)=PSI(I,J)
86    CONTINUE
      DO 87 J=1,N
      PSI(I,J)=QPSI(J)+ADON
87    CONTINUE
88    CONTINUE
      RETURN
      END
 49   IF(RD1.LE.R(JD)) GO TO 50
      RD=RD1
      GO TO 51
 50   IF(RD2.LE.R(JD)) GO TO 61
      RD=RD2
51    IF(RD.LT.R(JD)*(1.0+6.2835/N)) RD=R(JD)*(6.2835/N+1.0)
      AD=(RD-R(JD))*D(JD)
      TYPE 515,RD,THETA(JD)
515   FORMAT('   RD=',D,'   THETA=',D)
      M=2*N
      DO 53 J=1,N
      DO 52 I=1,M
      PSI(I,J)=PSI(I,J)+AD*(RD-R(I)*DCOS(THETA(J)-THETA(JD)))/(RD**2
     1          +R(I)**2-2*RD*R(I)*DCOS(THETA(J)-THETA(JD)))
 52   CONTINUE
 53   CONTINUE
      RETURN
 60   TYPE 62
 62   FORMAT(' DIPOLE RADIUS INCLUDES IMAGINARY ROOT')
      RETURN
 61   TYPE 63
 63   FORMAT(' DIPOLE LOCATED INSIDE BOUNDARY')
      RETURN
      END
      SUBROUTINE VGRAPH(XO,YO,PHIMIN,PHIMAX)
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/MVD/XB(40),YB(40),PPHI(40),IB,S
      COMMON/ALL/PSI(80,40),PHI(40),M,N
      COMMON/MVPA/R(80),THETA(40)
      DIMENSION X(190),Y(190)
      CALL MOVEA(SNGL(XB(IB)),SNGL(YB(IB)))
      DO 500 K=1,IB
      CALL DRAWA(SNGL(XB(K)),SNGL(YB(K)))
500   CONTINUE
```

```
      PI=4*DATAN(1.D0)
      MR=N+1
      MN=N+2
      DO 595 K=1,39
      V=PHIMIN+K*(PHIMAX-PHIMIN)/40
      L=0
      DO 515 J=1,N
      DO 510 I=MN,M
      IF(R(I).GT.R(J)) GO TO 515
      IF((PSI(I-1,J)-V)*(PSI(I,J)-V).GT.0.0) GO TO 510
      L=L+1
      RR=R(I)+(V-PSI(I,J))*(R(I)-R(I-1))/(PSI(I,J)-PSI(I-1,J))
      X(L)=RR*DCOS(THETA(J))
      Y(L)=RR*DSIN(THETA(J))
510   CONTINUE
515   CONTINUE
      THETA(N+1)=THETA(1)
      DO 525 I=MR,M
      PSI(I,N+1)=PSI(I,1)
      DO 520 J=2,MR
      JJ=J
      IF(JJ.EQ.N+1) JJ=1
      IF(R(I).GT.R(JJ)) GO TO 520
      IF(R(I).GT.R(J-1)) GO TO 520
      IF((PSI(I,J-1)-V)*(PSI(I,J)-V).GT.0.0) GO TO 520
      L=L+1
      TT=THETA(J)+(V-PSI(I,J))*2*PI/((PSI(I,J)-PSI(I,J-1))*N)
      X(L)=R(I)*DCOS(TT)
      Y(L)=R(I)*DSIN(TT)
520   CONTINUE
525   CONTINUE
      LMAX=L
      DG=0.0
      DO 570 L=1,LMAX
      IF((X0-X(L))**2+(Y0-Y(L))**2.LT.DG) GO TO 570
      DG=(X0-X(L))**2+(Y0-Y(L))**2
      LY=L
570   CONTINUE
      XX=X(1)
      YY=Y(1)
      X(1)=X(LY)
      Y(1)=Y(LY)
      X(LY)=XX
      Y(LY)=YY
      CALL MOVEA(SNGL(X(1)),SNGL(Y(1)))
      LL=1
535   LL=LL+1
      DSQ=1.0D38
      DO 540 L=LL,LMAX
      DSQL=(X(L)-X(LL-1))**2+(Y(L)-Y(LL-1))**2
      IF(DSQL.GT.DSQ) GO TO 540
      DSQ=DSQL
      LDMIN=L
```

```
540     CONTINUE
        XX=X(LL)
        YY=Y(LL)
        X(LL)=X(LDMIN)
        Y(LL)=Y(LDMIN)
        X(LDMIN)=XX
        Y(LDMIN)=YY
        CALL DRAWA(SNGL(X(LL)),SNGL(Y(LL)))
        IF(LL.EQ.LMAX) GO TO 595
        GO TO 535
595     CONTINUE
        RETURN
        END
```

```
      SUBROUTINE SCALE
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/ALL/PSI(80,40),PHI(40),M,N
      AVPHI=0.0
      AVPSI=0.0
      DO 80 J=1,N
      AVPHI=AVPHI+PHI(J)
      AVPSI=AVPSI+PSI(J,J)
 80   CONTINUE
      AVPHI=AVPHI/N
      AVPSI=AVPSI/N
      ABPHI=0.0
      ABPSI=0.0
      DO 81 J=1,N
      ABPHI=ABPHI+DABS(PHI(J)-AVPHI)
      ABPSI=ABPSI+DABS(PSI(J,J)-AVPSI)
 81   CONTINUE
      M=2*N
      DO 83 J=1,N
      DO 82 I=1,M
      IF(ABPSI.EQ.0.0) GO TO 82
      PSI(I,J)=(PSI(I,J)-AVPSI)*ABPHI/ABPSI+AVPSI
 82   CONTINUE
 83   CONTINUE
      RETURN
      END
 49   IF(RD1.LE.R(JD)) GO TO 50
      RD=RD1
      GO TO 51
 50   IF(RD2.LE.R(JD)) GO TO 61
      RD=RD2
 51   IF(RD.LT.R(JD)*(1.0+6.2835/N)) RD=R(JD)*(6.2835/N+1.0)
      AD=(RD-R(JD))*D(JD)
      TYPE 515,RD,THETA(JD)
515   FORMAT('   RD=',D,'   THETA=',D)
      M=2*N
      DO 53 J=1,N
      DO 52 I=1,M
      PSI(I,J)=PSI(I,J)+AD*(RD-R(I)*DCOS(THETA(J)-THETA(JD)))/(RD**2
     1          +R(I)**2-2*RD*R(I)*DCOS(THETA(J)-THETA(JD)))
 52   CONTINUE
 53   CONTINUE
      RETURN
 60   TYPE 62
 62   FORMAT(' DIPOLE RADIUS INCLUDES IMAGINARY ROOT')
      RETURN
 61   TYPE 63
 63   FORMAT(' DIPOLE LOCATED INSIDE BOUNDARY')
      RETURN
      END
      SUBROUTINE VGRAPH(XO,YO,PHIMIN,PHIMAX)
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/MVD/XB(40),YB(40),PPHI(40),IB,S
```

```
      COMMON/ALL/PSI(80,40),PHI(40),M,N
      COMMON/MVPA/R(80),THETA(40)
      DIMENSION X(190),Y(190)
      CALL MOVEA(SNGL(XB(IB)),SNGL(YB(IB)))
      DO 500 K=1,IB
      CALL DRAWA(SNGL(XB(K)),SNGL(YB(K)))
500   CONTINUE
      PI=4*DATAN(1.D0)
      MR=N+1
      MN=N+2
      DO 595 K=1,39
      V=PHIMIN+K*(PHIMAX-PHIMIN)/40
      L=0
      DO 515 J=1,N
      DO 510 I=MN,M
      IF(R(I).GT.R(J)) GO TO 515
      IF((PSI(I-1,J)-V)*(PSI(I,J)-V).GT.0.0) GO TO 510
      L=L+1
      RR=R(I)+(V-PSI(I,J))*(R(I)-R(I-1))/(PSI(I,J)-PSI(I-1,J))
      X(L)=RR*DCOS(THETA(J))
      Y(L)=RR*DSIN(THETA(J))
510   CONTINUE
515   CONTINUE
      THETA(N+1)=THETA(1)
      DO 525 I=MR,M
      PSI(I,N+1)=PSI(I,1)
      DO 520 J=2,MR
      JJ=J
      IF(JJ.EQ.N+1) JJ=1
      IF(R(I).GT.R(JJ)) GO TO 520
      IF(R(I).GT.R(J-1)) GO TO 520
      IF((PSI(I,J-1)-V)*(PSI(I,J)-V).GT.0.0) GO TO 520
      L=L+1
      TT=THETA(J)+(V-PSI(I,J))*2*PI/((PSI(I,J)-PSI(I,J-1))*N)
      X(L)=R(I)*DCOS(TT)
      Y(L)=R(I)*DSIN(TT)
520   CONTINUE
525   CONTINUE
      LMAX=L
      DG=0.0
      DO 570 L=1,LMAX
      IF((X0-X(L))**2+(Y0-Y(L))**2.LT.DG) GO TO 570
      DG=(X0-X(L))**2+(Y0-Y(L))**2
      LY=L
570   CONTINUE
      XX=X(1)
      YY=Y(1)
      X(1)=X(LY)
      Y(1)=Y(LY)
      X(LY)=XX
      Y(LY)=YY
      CALL MOVEA(SNGL(X(1)),SNGL(Y(1)))
      LL=1
```

```
535     LL=LL+1
        DSQ=1.0D38
        DO 540 L=LL,LMAX
        DSQL=(X(L)-X(LL-1))**2+(Y(L)-Y(LL-1))**2
        IF(DSQL.GT.DSQ) GO TO 540
        DSQ=DSQL
        LDMIN=L
540     CONTINUE
        XX=X(LL)
        YY=Y(LL)
        X(LL)=X(LDMIN)
        Y(LL)=Y(LDMIN)
        X(LDMIN)=XX
        Y(LDMIN)=YY
        CALL DRAWA(SNGL(X(LL)),SNGL(Y(LL)))
        IF(LL.EQ.LMAX) GO TO 595
        GO TO 535
595     CONTINUE
        RETURN
        END
```

APPENDIX C

Fortran Routines for Generating the Biased Fin Potential and the Associated Electron Trajectory Data

```
cROUTINE TEST.F4
      IMPLICIT REAL *8 (A-H,O-Z)
      COMPLEX Z,ZZ
      DIMENSION PHI(82,22),XX(200),YY(200),X(82),Y(22)

      PI=4*DATAN(1.0D0)
      TYPE 10
10    FORMAT(' ENTER X DIMENSION=',$)
      ACCEPT 20,DX
20    FORMAT(D)
      TYPE 30
30    FORMAT(' ENTER Y DIMENSION=',$)
      ACCEPT 20,DY
      X(1)=-DX/80
      DO 40 I=2,82
      X(I)=(I-2)*DX/80
40    CONTINUE
      Y(1)=-DY/20
      DO 50 J=2,22
      Y(J)=(J-2)*DY/20
50    CONTINUE
      S=PI/1.8
      PSI=DLOG(-DEXP(S)+DSQRT(1+DEXP(2*S)))
      TYPE 55,PSI
55    FORMAT(D)
      DO 70 I=22,41
      DO 60 J=2,22
      S=PI*(1-Y(J)/DY)/1.8
      T=PI*(2*X(I)/DX-0.5)
      Z=CMPLX(SNGL(S),SNGL(T))
      ZZ=CSQRT(1-CEXP(2*Z))
      D=REAL(ZZ)
      E=AIMAG(ZZ)
      ZZ=CEXP(Z)
      DD=-AIMAG(ZZ)
      EE=REAL(ZZ)
      ZZ=CMPLX(SNGL(D+DD),SNGL(E+EE))
      ZZ=CLOG(ZZ)
      PHI(I,J)=-REAL(ZZ)-PSI
60    CONTINUE
70    CONTINUE
      DO 75 J=2,22
      S=PI*(1-Y(J)/DY)/1.8
      PHI(42,J)=DLOG(-DEXP(S)+DSQRT(1+DEXP(2*S)))-PSI
75    CONTINUE
      DO 90 K=2,21
      DO 80 J=2,22
      PHI(K,J)=PHI(44-K,J)
80    CONTINUE
90    CONTINUE
      DO 110 L=3,42
      DO 100 J=2,22
      PHI(40+L,J)=PHI(L,J)
```

```
100     CONTINUE
110     CONTINUE
        CALL INITT
        CALL DWINDO(SNGL(X(2)),SNGL(X(82)),SNGL(Y(2)),SNGL(Y(22)))
        II=INT(SNGL(780*DY/DX))
        CALL TWINDO(0,780,0,II)
        CALL MOVEA(SNGL(X(2)),SNGL(Y(2)))
        CALL DRAWA(SNGL(X(82)),SNGL(Y(2)))
        PHIMIN=1.0D38
        PHIMAX=-1.0D38
        DO 140 I=2,82
        DO 130 J=2,22
        IF(PHI(I,J).GT.PHIMIN) GO TO 125
        PHIMIN=PHI(I,J)
125     IF(PHI(I,J).LT.PHIMAX) GO TO 130
        PHIMAX=PHI(I,J)
130     CONTINUE
140     CONTINUE
        TYPE 290,PHIMIN,PHIMAX
290     FORMAT(' PHIMIN=',D12.6,'   PHIMAX=',D12.6)
        K=0
142     K=K+1
        V=PHIMIN+K*0.2
        L=0
        DO 160 J=2,22
        DO 150 I=3,82
        IF((PHI(I-1,J)-V)*(PHI(I,J)-V).GE.0.0) GO TO 150
        L=L+1
        XX(L)=X(I-1)+(X(I)-X(I-1))*(V-PHI(I-1,J))/(PHI(I,J)-PHI(I-1,J))
        YY(L)=Y(J)
150     CONTINUE
160     CONTINUE
        DO 180 I=2,82
        DO 170 J=3,22
        IF((PHI(I,J-1)-V)*(PHI(I,J)-V).GE.0.0) GO TO 170
        L=L+1
        XX(L)=X(I)
        YY(L)=Y(J-1)+(Y(J)-Y(J-1))*(V-PHI(I,J-1))/(PHI(I,J)-PHI(I,J-1))
170     CONTINUE
180     CONTINUE
        LM=L
        XMIN=X(82)
        DO 190 L=1,LM
        IF(XX(L).GT.XMIN) GO TO 190
        XMIN=XX(L)
        LMIN=L
190     CONTINUE
        XS=XX(1)
        YS=YY(1)
        XX(1)=XX(LMIN)
        YY(1)=YY(LMIN)
        XX(LMIN)=XS
        YY(LMIN)=YS
```

```
      CALL MOVEA(SNGL(XX(1)),SNGL(YY(1)))
      LL=1
195   LL=LL+1
      DSQ=1.0D38
      DO 210 L=LL,LM
      DSQL=(XX(L)-XX(LL-1))**2+(YY(L)-YY(LL-1))**2
      IF(DSQL.GT.DSQ) GO TO 210
      DSQ=DSQL
      LDM=L
210   CONTINUE
      XS=XX(LL)
      YS=YY(LL)
      XX(LL)=XX(LDM)
      YY(LL)=YY(LDM)
      XX(LDM)=XS
      YY(LDM)=YS
      CALL DRAWA(SNGL(XX(LL)),SNGL(YY(LL)))
      IF(LL.EQ.LM) GO TO 220
      GO TO 195
220   IF(V+0.2.LE.PHIMAX) GO TO 142
      PAUSE
      OPEN(UNIT=1,FILE='POTL.DAT')
      WRITE (1,20) X(1)
      WRITE (1,20) Y(1)
      DO 240 I=2,82
      WRITE(1,20) X(I)
      DO 230 J=2,22
      IF(I.GT.2) GO TO 222
      WRITE(1,20) Y(J)
222   WRITE(1,20) PHI(I,J)
225   FORMAT(D12.8)
230   CONTINUE
240   CONTINUE
      STOP
      END
```

```
      cROUTINE DATA.F4
        IMPLICIT REAL *8(A-H,O-Z)
        COMMON/COORD/PHI(82,22),XX(82),YY(22),DX,DY
        COMMON/DATA/XO(50),YO(50),VXO(50),VYO(50),XT(50),YT(50)
        COMMON/SAVE/X(250),Y(250),FX(250),FY(250),DT(250),KT
        OPEN(UNIT=1,FILE='POTL.DAT')
        READ(1,50) XX(1)
        READ(1,50) YY(1)
        DO 200 I=2,82
        READ (1,50) XX(I)
 50     FORMAT(D)
        DO 100 J=2,22
        IF(I.GT.2) GO TO 80
        READ(1,50) YY(J)
 80     READ(1,50) PHI(I,J)
100     CONTINUE
200     CONTINUE
        CALL INITT
        CALL DWINDO(SNGL(XX(1)),SNGL(XX(82)),SNGL(YY(1)),SNGL(YY(22)))
        DX=XX(82)-XX(2)
        DY=YY(22)-YY(2)
        II=INT(SNGL(780*DY/DX))
        CALL TWINDO(0,780,0,II)
        K=0
300     K=K+1
        XO(K)=XX(13+K*9)
        YO(K)=0.0
        CALL MOVEA(SNGL(XO(K)),SNGL(YO(K)))
        VXO(K)=1.0
        VYO(K)=7.35D5
        KT=0
        DT(1)=DY/(100*VYO(K))
400     KT=KT+1
        CALL DVOGEL(K)
        CALL DRAWA(SNGL(X(KT+1)),SNGL(Y(KT+1)))
        IF(Y(KT+1).LE.0.0) GO TO 500
        GO TO 400
500     XT(K)=X(KT+1)
        YT(K)=0.0
        IF(K.EQ.5) GO TO 600
        GO TO 300
600     K=K+1
        XO(K)=XX(21)
        YO(K)=0.0
        CALL MOVEA(SNGL(XO(K)),SNGL(YO(K)))
        VXO(K)=5.5D5
        VYO(K)=5.0D5
        KT=0
        DT(1)=DY/(100*VYO(K))
700     KT=KT+1
        CALL DVOGEL(K)
        CALL DRAWA(SNGL(X(KT+1)),SNGL(Y(KT+1)))
        IF(Y(KT+1).LE.0.0) GO TO 800
```

```
        GO TO 700
800     XT(K)=X(KT+1)
        YT(K)=0.0
        OPEN(UNIT=1,FILE='TRAJ.DAT')
        IL=6
        WRITE(1,1500) IL
        DO 1400 K=1,6
        WRITE(1,1600) XO(K),VXO(K)
        WRITE(1,1600) YO(K),VYO(K)
        WRITE(1,1600) XT(K),YT(K)
1400    CONTINUE
1500    FORMAT(I)
1600    FORMAT(2D)
        CALL VPLOT
        STOP
        END
        SUBROUTINE NEWTON(X,Y,EX,EY,PSI,IK)
        IMPLICIT REAL *8(A-H,O-Z)
        COMMON/COORD/PHI(82,22),XX(82),YY(22),DX,DY
        DIMENSION FM(4,4,4),FMN(4,4,4),XM(4),XM1(4),YN(4),YN1(4)
        IF(X.GT.XX(82)) GO TO 220
        IF(X.LT.XX(1)) GO TO 220
        IF(Y.GT.YY(22)) GO TO 220
        IF(Y.LT.YY(1)) GO TO 220
        HX=DX/30.0
        HY=DY/20.0
        I=0
 10     I=I+1
        IF(X.GT.XX(I)) GO TO 10
        IF(I.EQ.2) GO TO 12
        IB=I-2
        GO TO 14
12      IB=I-1
14      IF(I.EQ.82) GO TO 16
        IE=I+1
        GO TO 18
16      IE=I
18      J=0
 20     J=J+1
        IF(Y.GT.YY(J)) GO TO 20
        IF(J.EQ.2) GO TO 25
        JB=J-2
        GO TO 30
 25     JB=J-1
 30     IF(J.EQ.22) GO TO 35
        JE=J+1
        GO TO 40
 35     JE=J
 40     MM=IE-IB+1
        XM(1)=1.0
        XM1(1)=0.0
        DO 70 M=2,MM
        IM=IB+M-2
```

```
      XM(M)=XM(M-1)*(X-XX(IM))
      XM1(M)=0.0
      DO 60 I=2,M
      X1=1.0
      DO 50 K=2,M
      IF(K.EQ.I) GO TO 50
      X1=X1*(X-XX(IB+K-2))
 50   CONTINUE
      XM1(M)=XM1(M)+X1
 60   CONTINUE
 70   CONTINUE
      NN=JE-JB+1
      YN(1)=1.0
      YN1(1)=0.0
      DO 100 N=2,NN
      JN=JB+N-2
      YN(N)=YN(N-1)*(Y-YY(JN))
      YN1(N)=0.0
      DO 90 J=2,N
      Y1=1.0
      DO 80 L=2,N
      IF(L.EQ.J) GO TO 80
      Y1=Y1*(Y-YY(JB+L-2))
 80   CONTINUE
      YN1(N)=YN1(N)+Y1
 90   CONTINUE
100   CONTINUE
      DO 110 M=1,MM
      DO 105 N=1,NN
      FM(1,M,N)=PHI(IB+M-1,JB+N-1)
105   CONTINUE
110   CONTINUE
      DO 140 L=2,NN
      DO 130 N=L,NN
      DO 120 M=1,MM
      FM(L,M,N)=(FM(L-1,M,N-1)-FM(L-1,M,N))/(HY*(1-L))
120   CONTINUE
130   CONTINUE
140   CONTINUE
      DO 160 M=1,MM
      DO 150 N=1,NN
      FMN(1,M,N)=FM(N,M,N)
150   CONTINUE
160   CONTINUE
      DO 190 K=2,MM
      DO 180 M=K,MM
      DO 170 N=1,NN
      FMN(K,M,N)=(FMN(K-1,M-1,N)-FMN(K-1,M,N))/(HX*(1-K))
170   CONTINUE
180   CONTINUE
190   CONTINUE
      EX=0.0
      EY=0.0
```

```
      PSI=0.0
      DO 210 N=1,NN
      DO 200 M=1,MM
      EX=EX-XM1(M)*YN(N)*FMN(M,M,N)
      EY=EY-YN1(N)*XM(M)*FMN(M,M,N)
      PSI=PSI+XM(M)*YN(N)*FMN(M,M,N)
200   CONTINUE
210   CONTINUE
      GO TO 230
220   IK=-1
230   RETURN
      END
      SUBROUTINE DVOGEL(K)
      IMPLICIT REAL *8(A-H,O-Z)
      COMMON/COORD/PHI(82,22),XX(82),YY(22),DX,DY
      COMMON/DATA/X0(50),Y0(50),VX0(50),VY0(50),XT(50),YT(50)
      COMMON/SAVE/X(250),Y(250),FX(250),FY(250),DT(250),KT
      QM=1.76D11
      DIM=DSQRT(DX**2+DY**2)
      IK=0
      IF(KT.GT.1) GO TO 200
      VX=VX0(K)
      VY=VY0(K)
      X(1)=X0(K)
      Y(1)=Y0(K)
100   CALL NEWTON(X(KT),Y(KT),EX,EY,PSI,IK)
      IF(IK.NE.0) GO TO 400
      AX=QM*EX
      AY=QM*EY
      XH=X(KT)-(0.5*VX-0.125*AX*DT(KT))*DT(KT)
      YH=Y(KT)-(0.5*VY-0.125*AY*DT(KT))*DT(KT)
      CALL NEWTON(XH,YH,EX,EY,PSI,IK)
      IF(IK.NE.0) GO TO 400
      AXH=QM*EX
      AYH=QM*EY
200   XH=X(KT)+(0.5*VX+(4.0*AX-AXH)*DT(KT)/24.0)*DT(KT)
      YH=Y(KT)+(0.5*VY+(4.0*AY-AYH)*DT(KT)/24.0)*DT(KT)
      CALL NEWTON(XH,YH,EX,EY,PSI,IK)
      IF(IK.NE.0) GO TO 400
      AXH=QM*EX
      AYH=QM*EY
      X(KT+1)=X(KT)+(VX+(AX+2.0*AXH)*DT(KT)/6.0)*DT(KT)
      Y(KT+1)=Y(KT)+(VY+(AY+2.0*AYH)*DT(KT)/6.0)*DT(KT)
      AAX=AX
      AAY=AY
      CALL NEWTON(X(KT+1),Y(KT+1),EX,EY,PSI,IK)
      IF(IK.NE.0) GO TO 400
      AX=QM*EX
      AY=QM*EY
      FX(KT)=EX
      FY(KT)=EY
      A1SQ=AAX**2+AAY**2
      DELD=DSQRT((X(KT+1)-X(KT))**2+(Y(KT+1)-Y(KT))**2)
```

```
      IF(DELD.GT.DIM/80) GO TO 320
      IF(DELD.LT.DIM/400) GO TO 330
      IF(A1SQ.LT.1.D-6) GO TO 300
280   VX=VX+DT(KT)*(AX+4.0*AXH+AAX)/6.0
      VY=VY+DT(KT)*(AY+4.0*AYH+AAY)/6.0
      DT(KT+1)=DT(KT)
      GO TO 440
300   V1SQ=VX**2+VY**2
      DVSQ=A1SQ*DT(KT)**2
      IF(V1SQ.GT.DVSQ*1.D4) GO TO 280
      IK=1
      GO TO 410
320   DT(KT)=DT(KT)/1.5
      GO TO 100
330   DT(KT)=DT(KT)*1.5
      GO TO 100
400   TYPE 420
      GO TO 440
410   TYPE 430
420   FORMAT(' ELECTRON HAS ESCAPED POTENTIAL REGION
     1  INCLUDED IN X-Y COORDINATE GRID')
430   FORMAT(' PARTICLE VELOCITY AND ACCELERATION BOTH
     1  EQUAL ZERO')
440   RETURN
      END
      SUBROUTINE VPLOT
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/COORD/PHI(82,22),XX(82),YY(22),DX,DY
      DIMENSION X(180),Y(180)
      CALL INITT
      CALL DWINDO(SNGL(XX(1)),SNGL(XX(82)),SNGL(YY(1)),SNGL(YY(22)))
      II=INT(SNGL(780*DY/DX))
      CALL TWINDO(0,780,0,II)
      CALL MOVEA(SNGL(XX(2)),SNGL(YY(2)))
      CALL DRAWA(SNGL(XX(82)),SNGL(YY(2)))
      PHIMIN=1.0D38
      PHIMAX=-1.0D38
      DO 10 I=2,82
      DO 5 J=2,22
      IF(PHI(I,J).GT.PHIMIN) GO TO 2
      PHIMIN=PHI(I,J)
 2    IF(PHI(I,J).LT.PHIMAX) GO TO 5
      PHIMAX=PHI(I,J)
 5    CONTINUE
10    CONTINUE
      K=0
12    K=K+1
      V=PHIMIN+K*0.2
      L=0
      DO 20 J=2,22
      DO 15 I=3,82
      IF((PHI(I-1,J)-V)*(PHI(I,J)-V).GE.0.0) GO TO 15
      L=L+1
```

```
      X(L)=XX(I-1)+(XX(I)-XX(I-1))*(V-PHI(I-1,J))/(PHI(I,J)-
     1  PHI(I-1,J))
      Y(L)=YY(J)
15    CONTINUE
20    CONTINUE
      DO 30 I=2,82
      DO 25 J=3,22
      IF((PHI(I,J-1)-V)*(PHI(I,J)-V).GE.0.0) GO TO 25
      L=L+1
      X(L)=XX(I)
      Y(L)=YY(J-1)+(YY(J)-YY(J-1))*(V-PHI(I,J-1))/(PHI(I,J)-
     1  PHI(I,J-1))
25    CONTINUE
30    CONTINUE
      LMAX=L
      XMIN=XX(82)
      DO 35 L=1,LMAX
      IF(X(L).GT.XMIN) GO TO 35
      XMIN=X(L)
      LMIN=L
35    CONTINUE
      XS=X(1)
      YS=Y(1)
      X(1)=X(LMIN)
      Y(1)=Y(LMIN)
      X(LMIN)=XS
      Y(LMIN)=YS
      CALL MOVEA(SNGL(X(1)),SNGL(Y(1)))
      LL=1
40    LL=LL+1
      DSQ=1.0D38
      DO 45 L=LL,LMAX
      DSQL=(X(L)-X(LL-1))**2+(Y(L)-Y(LL-1))**2
      IF(DSQL.GT.DSQ) GO TO 45
      DSQ=DSQL
      LDMIN=L
45    CONTINUE
      XS=X(LL)
      YS=Y(LL)
      X(LL)=X(LDMIN)
      Y(LL)=Y(LDMIN)
      X(LDMIN)=XS
      Y(LDMIN)=YS
      CALL DRAWA(SNGL(X(LL)),SNGL(Y(LL)))
      IF(LL.EQ.LMAX) GO TO 50
      GO TO 40
50    IF(V+0.2.LE.PHIMAX) GO TO 12
      CALL FINITT(800,500)
      RETURN
      END
```

APPENDIX D

Fortran Routines for Dipole Synthesis of Solutions to The Model Problem

```
cROUTINE PTENT.F4
      IMPLICIT REAL *8(A-H,O-Z)
      COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
      COMMON/DATA/XO(50),YO(50),VXO(50),VYO(50),XT(50),YT(50)
      COMMON/SAVE/X(110),Y(110),DT(250),KT
      DIMENSION XS(110),YS(110),SX(110),SY(110),TD(110),DDX(50)
     1  ,YN(50),XN(50)
      EXTERNAL TEKHAN,PLTHAN,SCPHAN
      CALL INITTG
      TYPE 1
    1 FORMAT(' ENTER X AND Y DIMENSIONS OF POTENTIAL REGION',$)
      ACCEPT 2,DX,DY
    2 FORMAT(2D)
      TYPE 19
   19 FORMAT(' ENTER 1 TO INPUT DATA FROM FILE TRAJ.DAT, ENTER 0
     1  TO INPUT DATA MANUALLY',$)
      ACCEPT 4, NT
      IF(NT.EQ.1) GO TO 140
      TYPE 3
    3 FORMAT(' ENTER NUMBER OF ELECTRON TRAJECTORIES',$)
      ACCEPT 4,KK
    4 FORMAT(I)
      DO 8 K=1,KK
      TYPE 5
    5 FORMAT(' ENTER INITIAL X COORDINATE AND VELOCITY',$)
      ACCEPT 2,XO(K),VXO(K)
      TYPE 6
    6 FORMAT(' ENTER INITIAL Y COORDINATE AND VELOCITY',$)
      ACCEPT 2,YO(K),VYO(K)
      TYPE 7
    7 FORMAT(' ENTER FINAL X AND Y COORDINATES',$)
      ACCEPT 2,XT(K),YT(K)
      type 67,k,xO(k),xt(k)
   67 format(i2,2d12.6)
    8 CONTINUE
    9 EQ=1.60210D-19
      EM=9.1091D-31
      QM=EQ/EM
      AO=2*VXO(KK)*VYO(KK)/(QM*(XT(KK)-XO(KK)))
      TYPE 15,AO
   15 FORMAT(' AO=',D)
      XX(1)=-DX/80
      XX(2)=0.0
      DO 20 I=3,82
      XX(I)=XX(I-1)+DX/80
   20 CONTINUE
      YY(1)=-DY/20
      YY(2)=0.0
      DO 30 J=3,32
      YY(J)=YY(J-1)+DY/20
   30 CONTINUE
   32 DO 50 I=1,82
      DO 40 J=1,32
```

```
      PHI(I,J)=A0*YY(J)
40    CONTINUE
50    CONTINUE
      PAUSE
      IT=0
60    DD=0.0
      D=0.0
      TYPE 65,IT
65    FORMAT(' ITERATION=',I4)
      IT=IT+1
      DO 90 K=1,KK-1
      YN(K)=0.0
      KT=0
      TYPE 82,X0(K),Y0(K)
      DT(1)=DY/(100*VY0(K))
70    KT=KT+1
      CALL DVOGEL(K)
      IF(Y(KT+1).LT.YN(K)) GO TO 75
      YN(K)=Y(KT+1)
      XN(K)=X(KT+1)
75    IF(Y(KT+1).GT.Y0(K)) GO TO 70
80    TYPE 82,X(KT+1),Y(KT+1)
      DDX(K)=(XT(K)-X(KT+1))/(XT(K)-X0(K))
82    FORMAT(' X=',D12.4,'    Y=',D12.4)
      DD=DD+DABS(XT(K)-X(KT+1))
      IF(D.GT.DABS(DDX(K))) GO TO 90
      D=DABS(DDX(K))
      XM=X(KT+1)
      KM=K
      DO 85 KI=1,KT+1
      XS(KI)=X(KI)
      YS(KI)=Y(KI)
      SX(KI)=FX(KI)
      SY(KI)=FY(KI)
      TD(KI)=DT(KI)
85    CONTINUE
90    CONTINUE
      TYPE 95,D
      TYPE 95,DD
95    FORMAT(D)
      IF(DD.LT.0.01*DX) GO TO 130
      K=KM
180   K=K+1
      IF(K.EQ.KK) GO TO 190
      IF(DDX(K)*DDX(K-1).GT.0.0) GO TO 180
      LP=K-KM
      GO TO 200
190   LP=KK-KM
200   K=KM
210   K=K-1
      IF(K.EQ.0) GO TO 220
      IF(DDX(K)*DDX(K+1).GT.0.0) GO TO 210
      LN=K-KM
```

```
        GO TO 230
220     LN=-KM
230     IF(LP.LT.-LN) GO TO 240
        IF(LP.GT.-LN) GO TO 235
        IF(DABS(DDX(KM+LP)).GT.DABS(DDX(KM+LN))) GO TO 240
235     L=LN
        GO TO 250
240     L=LP
250     IF(L.EQ.-KM) GO TO 260
        IF(L.EQ.KK-KM) GO TO 270
        XL=0.5*(XN(KM+L)+XN(KM+L-1))
        GO TO 300
260     XL=1.5*XN(1)-0.5*XN(2)
        GO TO 300
270     XL=1.5*XN(KK-1)-0.5*XN(KK-2)
300     YL=1.4*DY
        VXII=0.0
        DVXII=0.0
        VXXI=0.0
        DO 98 I=1,KT
        DVXI=DVXII
        DVXII=2*(XS(I)-XL)*((YL-YS(I))/((XS(I)-XL)**2
     1  +(YL-YS(I))**2)**2-(YS(I)+YL)/((XS(I)-XL)**2
     2  +(YS(I)+YL)**2)**2)
        VXI=VXII
        VXII=VXI+0.5*(DVXII+DVXI)*TD(I)
        VXXI=VXXI+0.5*(VXII+VXI)*TD(I)
 98     CONTINUE
        AL=(XM-XT(KM))/(QM*VXXI)
        TYPE 95,AL
        IF(AL.LT.-0.1) GO TO 100
        IF(AL.GT.0.1) GO TO 99
        GO TO 101
 99     AL=0.1
        GO TO 101
100     AL=-0.1
101     TYPE 105,AL,XL,YL
105     FORMAT(' AL=',D12.5,' XL=',D12.5, ' YL=',D12.5)
        PAUSE
        DO 120 I=2,82
        DO 110 J=2,32
        PHI(I,J)=PHI(I,J)+AL*((YY(J)+YL)/((XX(I)-XL)**2
     1  +(YY(J)+YL)**2)-(YL-YY(J))/((XX(I)-XL)**2
     2  +(YL-YY(J))**2))
110     CONTINUE
120     CONTINUE
        KT=0
        DT(1)=DY/(100*VY0(KK))
121     KT=KT+1
        CALL DVOGEL(KK)
        IF(Y(KT+1).GT.Y0(KK)) GO TO 121
        SF=(X(KT+1)-X0(KK))/(XT(KK)-X0(KK))
        TYPE 122,SF
```

```
122     FORMAT(' SF=',D12.6)
        DO 125 I=2,82
        DO 124 J=2,32
        PHI(I,J)=SF*PHI(I,J)
124     CONTINUE
125     CONTINUE
        GO TO 60
130     DO 134 I=10,70
        TYPE 132, PHI(I,3),PHI(I,5),PHI(I,7),PHI(I,9),PHI(I,11),
     1  PHI(I,13),PHI(I,15),PHI(I,17),PHI(I,19),PHI(I,21)
132     FORMAT(' ',10D12.5)
134     CONTINUE
        CALL VPLOT
        GO TO 160
140     OPEN(UNIT=1,FILE='TRAJ.DAT')
        READ (1,4) KK
        DO 150 K=1,KK
        READ(1,2) X0(K),VX0(K)
        READ(1,2) Y0(K),VY0(K)
        READ(1,2) XT(K),YT(K)
        type 155,x0(k),xt(k)
155     format(' x0=',d12.6 ,'    xt=',d12.6)
150     CONTINUE
        GO TO 9
160     STOP
        END
        SUBROUTINE NEWTON(X,Y,EX,EY,PSI,IK)
        IMPLICIT REAL *8(A-H,O-Z)
        COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
        DIMENSION FM(4,4,4),FMN(4,4,4),XM(4),XM1(4),YN(4),YN1(4)
  5     FORMAT(2D12.6)
        IF(X.GT.XX(82)) GO TO 220
        IF(X.LT.XX(1)) GO TO 220
        IF(Y.GT.YY(32)) GO TO 220
        IF(Y.LT.YY(1)) GO TO 225
        HX=DX/80.0
        HY=DY/20.0
        I=0
 10     I=I+1
        IF(X.GT.XX(I)) GO TO 10
        IF(I.LE.3) GO TO 12
        IB=I-2
        GO TO 14
12      IB=2
14      IF(I.EQ.82) GO TO 16
        IE=I+1
        GO TO 18
16      IE=I
18      J=0
 20     J=J+1
        IF(Y.GT.YY(J)) GO TO 20
        IF(J.LE.3) GO TO 25
        JB=J-2
```

```
      GO TO 30
25    JB=2
30    IF(J.LE.32) GO TO 35
      JE=J+1
      GO TO 40
35    JE=32
40    MM=IE-IB+1
      XM(1)=1.0
      XM1(1)=0.0
      DO 70 M=2,MM
      IM=IB+M-2
      XM(M)=XM(M-1)*(X-XX(IM))
      XM1(M)=0.0
      DO 60 I=2,M
      X1=1.0
      DO 50 K=2,M
      IF(K.EQ.I) GO TO 50
      X1=X1*(X-XX(IB+K-2))
50    CONTINUE
      XM1(M)=XM1(M)+X1
60    CONTINUE
70    CONTINUE
      NN=JE-JB+1
      YN(1)=1.0
      YN1(1)=0.0
      DO 100 N=2,NN
      JN=JB+N-2
      YN(N)=YN(N-1)*(Y-YY(JN))
      YN1(N)=0.0
      DO 90 J=2,N
      Y1=1.0
      DO 80 L=2,N
      IF(L.EQ.J) GO TO 80
      Y1=Y1*(Y-YY(JB+L-2))
80    CONTINUE
      YN1(N)=YN1(N)+Y1
90    CONTINUE
100   CONTINUE
      DO 110 M=1,MM
      DO 105 N=1,NN
      FM(1,M,N)=PHI(IB+M-1,JB+N-1)
105   CONTINUE
110   CONTINUE
      DO 140 L=2,NN
      DO 130 N=L,NN
      DO 120 M=1,MM
      FM(L,M,N)=(FM(L-1,M,N-1)-FM(L-1,M,N))/(HY*(1-L))
120   CONTINUE
130   CONTINUE
140   CONTINUE
      DO 160 M=1,MM
      DO 150 N=1,NN
      FMN(1,M,N)=FM(N,M,N)
```

```
150     CONTINUE
160     CONTINUE
        DO 190 K=2,MM
        DO 180 M=K,MM
        DO 170 N=1,NN
        FMN(K,M,N)=(FMN(K-1,M-1,N)-FMN(K-1,M,N))/(HX*(1-K))
170     CONTINUE
180     CONTINUE
190     CONTINUE
        EX=0.0
        EY=0.0
        PSI=0.0
        DO 210 N=1,NN
        DO 200 M=1,MM
        EX=EX-XM1(M)*YN(N)*FMN(M,M,N)
        EY=EY-YN1(N)*XM(M)*FMN(M,M,N)
        PSI=PSI+XM(M)*YN(N)*FMN(M,M,N)
200     CONTINUE
210     CONTINUE
        GO TO 230
220     IK=-1
        GO TO 230
225     IK=2
230     RETURN
        END
        SUBROUTINE DVOGEL(K)
        IMPLICIT REAL *8(A-H,O-Z)
        COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
        COMMON/DATA/XO(50),YO(50),VXO(50),VYO(50),XT(50),YT(50)
        COMMON/SAVE/X(110),Y(110),DT(250),KT
        DIM=DSQRT(DX**2+DY**2)
        QM=1.76D11
        IK=0
        IF(KT.GT.1) GO TO 200
        VX=VXO(K)
        VY=VYO(K)
        X(1)=XO(K)
        Y(1)=YO(K)
100     CALL NEWTON(X(KT),Y(KT),EX,EY,PSI,IK)
101     FORMAT(' EX=',D12.4,'EY=',D12.4)
        IF(IK.NE.0) GO TO 400
        AX=QM*EX
        AY=QM*EY
        XH=X(KT)-(0.5*VX-0.125*AX*DT(KT))*DT(KT)
        YH=Y(KT)-(0.5*VY-0.125*AY*DT(KT))*DT(KT)
        CALL NEWTON(XH,YH,EX,EY,PSI,IK)
        IF(IK.NE.0) GO TO 400
        AXH=QM*EX
        AYH=QM*EY
200     XH=X(KT)+(0.5*VX+(4.0*AX-AXH)*DT(KT)/24.0)*DT(KT)
        YH=Y(KT)+(0.5*VY+(4.0*AY-AYH)*DT(KT)/24.0)*DT(KT)
        CALL NEWTON(XH,YH,EX,EY,PSI,IK)
        IF(IK.NE.0) GO TO 400
```

```
      AXH=QM*EX
      AYH=QM*EY
      X(KT+1)=X(KT)+(VX+(AX+2.0*AXH)*DT(KT)/6.0)*DT(KT)
      Y(KT+1)=Y(KT)+(VY+(AY+2.0*AYH)*DT(KT)/6.0)*DT(KT)
260   FORMAT(2I)
      AAX=AX
      AAY=AY
      CALL NEWTON(X(KT+1),Y(KT+1),EX,EY,PSI,IK)
      IF(IK.NE.0) GO TO 400
      AX=QM*EX
      AY=QM*EY
      A1SQ=AAX**2+AAY**2
      IF(A1SQ.LT.1.D-6) GO TO 300
      DELD=DSQRT((X(KT+1)-X(KT))**2+(Y(KT+1)-Y(KT))**2)
275   FORMAT(' DELD=',D12.6,' DIM=',D12.6)
      IF(DELD.GT.DIM/40) GO TO 320
      IF(DELD.LT.DIM/120) GO TO 330
280   VX=VX+DT(KT)*(AX+4.0*AXH+AAX)/6.0
      VY=VY+DT(KT)*(AY+4.0*AYH+AAY)/6.0
      DT(KT+1)=DT(KT)
      GO TO 440
300   V1SQ=VX**2+VY**2
      DVSQ=A1SQ*DT(KT)**2
      IF(V1SQ.GT.DVSQ*1.D4) GO TO 280
      IK=1
      GO TO 410
320   DT(KT)=DT(KT)/1.1
      GO TO 100
330   DT(KT)=DT(KT)*1.1
      GO TO 100
400   IF(IK.EQ.-1) TYPE 420
      GO TO 440
410   TYPE 430
420   FORMAT(' ELECTRON HAS ESCAPED POTENTIAL REGION
     1  INCLUDED IN X-Y COORDINATE GRID')
430   FORMAT(' PARTICLE VELOCITY AND ACCELERATION BOTH
     1  EQUAL ZERO')
440   RETURN
      END
      SUBROUTINE VPLOT
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
      DIMENSION X(250),Y(250)
      CALL INITT
      CALL SELINI
      CALL DWINDO(SNGL(XX(1)),SNGL(XX(82)),SNGL(YY(1)),SNGL(YY(22)))
      II=INT(SNGL(780*DY/DX))
      CALL TWINDO(0,780,0,II)
      CALL MOVEA(SNGL(XX(2)),SNGL(YY(2)))
      CALL DRAWA(SNGL(XX(82)),SNGL(YY(2)))
      PHIMIN=1.0D38
      PHIMAX=-1.0D38
      DO 10 I=2,82
```

```
      DO 5 J=2,22
      IF(PHI(I,J).GT.PHIMIN) GO TO 2
      PHIMIN=PHI(I,J)
 2    IF(PHI(I,J).LT.PHIMAX) GO TO 5
      PHIMAX=PHI(I,J)
 5    CONTINUE
10    CONTINUE
      K=0
12    K=K+1
      V=PHIMIN+K*0.2
      L=0
      DO 20 J=2,22
      DO 15 I=3,82
      IF((PHI(I-1,J)-V)*(PHI(I,J)-V).GE.0.0) GO TO 15
      L=L+1
      X(L)=XX(I-1)+(XX(I)-XX(I-1))*(V-PHI(I-1,J))/(PHI(I,J)-
     1 PHI(I-1,J))
      Y(L)=YY(J)
15    CONTINUE
20    CONTINUE
      DO 30 I=2,82
      DO 25 J=3,22
      IF((PHI(I,J-1)-V)*(PHI(I,J)-V).GE.0.0) GO TO 25
      L=L+1
      X(L)=XX(I)
      Y(L)=YY(J-1)+(YY(J)-YY(J-1))*(V-PHI(I,J-1))/(PHI(I,J)-
     1 PHI(I,J-1))
25    CONTINUE
30    CONTINUE
      LMAX=L
      XMIN=XX(82)
      DO 35 L=1,LMAX
      IF(X(L).GT.XMIN) GO TO 35
      XMIN=X(L)
      LMIN=L
35    CONTINUE
      XS=X(1)
      YS=Y(1)
      X(1)=X(LMIN)
      Y(1)=Y(LMIN)
      X(LMIN)=XS
      Y(LMIN)=YS
      CALL MOVEA(SNGL(X(1)),SNGL(Y(1)))
      LL=1
40    LL=LL+1
      DSQ=1.0D38
      DO 45 L=LL,LMAX
      DSQL=(X(L)-X(LL-1))**2+(Y(L)-Y(LL-1))**2
      IF(DSQL.GT.DSQ) GO TO 45
      DSQ=DSQL
      LDMIN=L
45    CONTINUE
      XS=X(LL)
```

```
      YS=Y(LL)
      X(LL)=X(LDMIN)
      Y(LL)=Y(LDMIN)
      X(LDMIN)=XS
      Y(LDMIN)=YS
      CALL DRAWA(SNGL(X(LL)),SNGL(Y(LL)))
      IF(LL.EQ.LMAX) GO TO 50
      GO TO 40
50    IF(V+0.2.LE.PHIMAX) GO TO 12
      CALL FINITT(800,500)
      RETURN
      END
```

```
cROUTINE RTENT.F4
      IMPLICIT REAL *8(A-H,O-Z)
      COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
      COMMON/DATA/XO(50),YO(50),VXO(50),VYO(50),XT(50),YT(50)
      COMMON/SAVE/X(110),Y(110),DT(250),KT
      DIMENSION XS(110),YS(110),SX(110),SY(110),TD(110),DDX(50),XM(50)
     1  ,YN(50),XN(50)
      EXTERNAL TEKHAN,PLTHAN,SCPHAN
      CALL INITTG
      TYPE 1
 1    FORMAT(' ENTER X AND Y DIMENSIONS OF POTENTIAL REGION',$)
      ACCEPT 2,DX,DY
 2    FORMAT(2D)
      TYPE 19
19    FORMAT(' ENTER 1 TO INPUT DATA FROM FILE TRAJ.DAT, ENTER 0
     1  TO INPUT DATA MANUALLY',$)
      ACCEPT 4, NT
      IF(NT.EQ.1) GO TO 140
      TYPE 3
 3    FORMAT(' ENTER NUMBER OF ELECTRON TRAJECTORIES',$)
      ACCEPT 4,KK
 4    FORMAT(I)
      DO 8 K=1,KK
      TYPE 5
 5    FORMAT(' ENTER INITIAL X COORDINATE AND VELOCITY',$)
      ACCEPT 2,XO(K),VXO(K)
      TYPE 6
 6    FORMAT(' ENTER INITIAL Y COORDINATE AND VELOCITY',$)
      ACCEPT 2,YO(K),VYO(K)
      TYPE 7
 7    FORMAT(' ENTER FINAL X AND Y COORDINATES',$)
      ACCEPT 2,XT(K),YT(K)
      type 67,k,xO(k),xt(k)
67    format(i2,2d12.6)
 8    CONTINUE
 9    EQ=1.60210D-19
      EM=9.1091D-31
      QM=EQ/EM
      AO=2*VXO(KK)*VYO(KK)/(QM*(XT(KK)-XO(KK)))
      TYPE 15,AO
15    FORMAT(' AO=',D)
      XX(1)=-DX/80
      XX(2)=0.0
      DO 20 I=3,82
      XX(I)=XX(I-1)+DX/80
20    CONTINUE
      YY(1)=-DY/20
      YY(2)=0.0
      DO 30 J=3,32
      YY(J)=YY(J-1)+DY/20
30    CONTINUE
32    DO 50 I=1,82
      DO 40 J=1,32
```

```
      PHI(I,J)=A0*YY(J)
40    CONTINUE
50    CONTINUE
      PAUSE
      IT=0
60    DD=0.0
      D=0.0
      TYPE 65,IT
65    FORMAT(' ITERATION=',I4)
      IT=IT+1
      DO 90 K=1,KK-1
      YN(K)=0.0
      KT=0
      TYPE 82,X0(K),Y0(K)
      DT(1)=DY/(100*VY0(K))
70    KT=KT+1
      CALL DVOGEL(K)
      IF(Y(KT+1).LT.YN(K)) GO TO 75
      YN(K)=Y(KT+1)
      XN(K)=X(KT+1)
75    IF(Y(KT+1).GT.Y0(K)) GO TO 70
80    TYPE 82,X(KT+1),Y(KT+1)
      DDX(K)=(XT(K)-X(KT+1))/DABS(XT(K)-X0(K))
82    FORMAT(' X=',D12.4,'    Y=',D12.4)
      DD=DD+DABS(XT(K)-X(KT+1))
      IF(D.GT.DABS(DDX(K))) GO TO 90
      D=DABS(DDX(K))
      XM(K)=X(KT+1)
      KM=K
      DO 85 KI=1,KT+1
      XS(KI)=X(KI)
      YS(KI)=Y(KI)
      SX(KI)=FX(KI)
      SY(KI)=FY(KI)
      TD(KI)=DT(KI)
85    CONTINUE
90    CONTINUE
      TYPE 95,D
      TYPE 95,DD
95    FORMAT(D)
      IF(DD.LT.0.01*DX) GO TO 130
      k=km
180   k=k+1
      if(k.eq.kk) go to 220
      if(ddx(k)*ddx(km).gt.0.0) go to 180
190   k=k+1
      if(k.eq.kk) go to 200
      if(ddx(k)*ddx(km).gt.0.0) go to 200
      if(dabs(ddx(k)).lt.dabs(ddx(k-1))) go to 200
      go to 190
200   lp=k-1-km
      go to 230
220   lp=kk
```

```
230      k=km
240      k=k-1
         if(k.eq.0) go to 270
         if(ddx(k)*ddx(km).gt.0.0) go to 240
250      k=k-1
         if(k.eq.0) go to 260
         if(ddx(k)*ddx(km).gt.0.0) go to 260
         if(dabs(ddx(k)).lt.dabs(ddx(k+1))) go to 260
         go to 250
260      ln=k+1-km
         go to 280
270      ln=-kk
280      if(lp.ne.kk) go to 340
         if(ln.ne.-kk) go to 340
         DDN=0.0
         DO 285 K=1,KM
         DDN=DDN+DDX(K)
285      CONTINUE
         DDP=0.0
         DO 290 K=KM,KK-1
         DDP=DDP+DDX(K)
290      CONTINUE
         IF(DABS(DDN).GT.DABS(DDP)) GO TO 300
         XL=XN(1)-0.5*(XN(2)-XN(1))
         GO TO 365
300      XL=XN(KK-1)+0.5*(XN(KK-1)-XN(KK-2))
         GO TO 365
340      IF(LP.GT.-LN) GO TO 350
         L=LP
         GO TO 360
350      L=LN
360      XL=XN(KM)+0.5*(XN(KM+L)-XN(KM))
365      YL=(1+0.1)*DY
         DDE=1.0D30
370      YL=YL+0.05*DY
         D1=((YN(KM)-YL)**3-3*(YN(KM)-YL)*(XN(KM)-XL)**2)/
     1   ((XN(KM)-XL)**2+(YN(KM)-YL)**2)**3
         D2=((YN(KM)+YL)**3-3*(YN(KM)+YL)*(XN(KM)-XL)**2)/
     1   ((XN(KM)-XL)**2+(YN(KM)+YL)**2)**3
         DDA=DDE
         DDE=DABS(D1-D2)
         IF(DDE.LT.DDA) GO TO 370
         VXII=0.0
         DVXII=0.0
         VXXI=0.0
         DO 98 I=1,KT
         DVXI=DVXII
         DVXII=2*(XS(I)-XL)*((YL-YS(I))/((XS(I)-XL)**2
     1   +(YL-YS(I))**2)**2-(YS(I)+YL)/((XS(I)-XL)**2
     2   +(YS(I)+YL)**2)**2)
         VXI=VXII
         VXII=VXI+0.5*(DVXII+DVXI)*TD(I)
         VXXI=VXXI+0.5*(VXII+VXI)*TD(I)
```

```
 98     CONTINUE
        AL=(XM(KM)-XT(KM))/(QM*VXXI)
        TYPE 95,AL
        if(al.lt.-0.1) go to 100
        if(al.gt.0.1) go to 99
        GO TO 101
 99     AL=0.1
        GO TO 101
100     AL=-0.1
101     TYPE 105,AL,XL,YL
105     FORMAT(' AL=',D12.5,' XL=',D12.5, ' YL=',D12.5)
        DO 120 I=2,82
        DO 110 J=2,32
        PHI(I,J)=PHI(I,J)+AL*((YY(J)+YL)/((XX(I)-XL)**2
     1  +(YY(J)+YL)**2)-(YL-YY(J))/((XX(I)-XL)**2
     2  +(YL-YY(J))**2))
110     CONTINUE
120     CONTINUE
        KT=0
        DT(1)=DY/(100*VY0(KK))
121     KT=KT+1
        CALL DVOGEL(KK)
        IF(Y(KT+1).GT.Y0(KK)) GO TO 121
        SF=(X(KT+1)-X0(KK))/(XT(KK)-X0(KK))
        TYPE 122,SF
122     FORMAT(' SF=',D12.6)
        DO 125 I=2,82
        DO 124 J=2,32
        PHI(I,J)=SF*PHI(I,J)
124     CONTINUE
125     CONTINUE
        GO TO 60
130     DO 134 I=10,70
        TYPE 132, PHI(I,3),PHI(I,5),PHI(I,7),PHI(I,9),PHI(I,11),
     1  PHI(I,13),PHI(I,15),PHI(I,17),PHI(I,19),PHI(I,21)
132     FORMAT(' ',10D12.5)
134     CONTINUE
        CALL VPLOT
        GO TO 160
140     OPEN(UNIT=1,FILE='TRAJ.DAT')
        READ (1,4) KK
        DO 150 K=1,KK
        READ(1,2) X0(K),VX0(K)
        READ(1,2) Y0(K),VY0(K)
        READ(1,2) XT(K),YT(K)
        type 155,x0(k),xt(k)
155     format(' x0=',d12.6 ,'    xt=',d12.6)
150     CONTINUE
        GO TO 9
160     STOP
        END
        SUBROUTINE NEWTON(X,Y,EX,EY,PSI,IK)
        IMPLICIT REAL *8(A-H,O-Z)
```

```
      COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
      DIMENSION FM(4,4,4),FMN(4,4,4),XM(4),XM1(4),YN(4),YN1(4)
 5    FORMAT(2D12.6)
      IF(X.GT.XX(82)) GO TO 220
      IF(X.LT.XX(1)) GO TO 220
      IF(Y.GT.YY(32)) GO TO 220
      IF(Y.LT.YY(1)) GO TO 225
      HX=DX/80.0
      HY=DY/20.0
      I=0
10    I=I+1
      IF(X.GT.XX(I)) GO TO 10
      IF(I.LE.3) GO TO 12
      IB=I-2
      GO TO 14
12    IB=2
14    IF(I.EQ.82) GO TO 16
      IE=I+1
      GO TO 18
16    IE=I
18    J=0
20    J=J+1
      IF(Y.GT.YY(J)) GO TO 20
      IF(J.LE.3) GO TO 25
      JB=J-2
      GO TO 30
25    JB=2
30    IF(J.EQ.32) GO TO 35
      JE=J+1
      GO TO 40
35    JE=J
40    MM=IE-IB+1
      XM(1)=1.0
      XM1(1)=0.0
      DO 70 M=2,MM
      IM=IB+M-2
      XM(M)=XM(M-1)*(X-XX(IM))
      XM1(M)=0.0
      DO 60 I=2,M
      X1=1.0
      DO 50 K=2,M
      IF(K.EQ.I) GO TO 50
      X1=X1*(X-XX(IB+K-2))
50    CONTINUE
      XM1(M)=XM1(M)+X1
60    CONTINUE
70    CONTINUE
      NN=JE-JB+1
      YN(1)=1.0
      YN1(1)=0.0
      DO 100 N=2,NN
      JN=JB+N-2
      YN(N)=YN(N-1)*(Y-YY(JN))
```

```
      YN1(N)=0.0
      DO 90 J=2,N
      Y1=1.0
      DO 80 L=2,N
      IF(L.EQ.J) GO TO 80
      Y1=Y1*(Y-YY(JB+L-2))
 80   CONTINUE
      YN1(N)=YN1(N)+Y1
 90   CONTINUE
100   CONTINUE
      DO 110 M=1,MM
      DO 105 N=1,NN
      FM(1,M,N)=PHI(IB+M-1,JB+N-1)
105   CONTINUE
110   CONTINUE
      DO 140 L=2,NN
      DO 130 N=L,NN
      DO 120 M=1,MM
      FM(L,M,N)=(FM(L-1,M,N-1)-FM(L-1,M,N))/(HY*(1-L))
120   CONTINUE
130   CONTINUE
140   CONTINUE
      DO 160 M=1,MM
      DO 150 N=1,NN
      FMN(1,M,N)=FM(N,M,N)
150   CONTINUE
160   CONTINUE
      DO 190 K=2,MM
      DO 180 M=K,MM
      DO 170 N=1,NN
      FMN(K,M,N)=(FMN(K-1,M-1,N)-FMN(K-1,M,N))/(HX*(1-K))
170   CONTINUE
180   CONTINUE
190   CONTINUE
      EX=0.0
      EY=0.0
      PSI=0.0
      DO 210 N=1,NN
      DO 200 M=1,MM
      EX=EX-XM1(M)*YN(N)*FMN(M,M,N)
      EY=EY-YN1(N)*XM(M)*FMN(M,M,N)
      PSI=PSI+XM(M)*YN(N)*FMN(M,M,N)
200   CONTINUE
210   CONTINUE
      GO TO 230
220   IK=-1
      GO TO 230
225   IK=2
230   RETURN
      END
      SUBROUTINE DVOGEL(K)
      IMPLICIT REAL *8(A-H,O-Z)
      COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
```

```
      COMMON/DATA/XO(50),YO(50),VXO(50),VYO(50),XT(50),YT(50)
      COMMON/SAVE/X(110),Y(110),DT(250),KT
      DIM=DSQRT(DX**2+DY**2)
      QM=1.76D11
      IK=0
      IF(KT.GT.1) GO TO 200
      VX=VXO(K)
      VY=VYO(K)
      X(1)=XO(K)
      Y(1)=YO(K)
100   CALL NEWTON(X(KT),Y(KT),EX,EY,PSI,IK)
101   FORMAT(' EX=',D12.4,'EY=',D12.4)
      IF(IK.NE.0) GO TO 400
      AX=QM*EX
      AY=QM*EY
      XH=X(KT)-(0.5*VX-0.125*AX*DT(KT))*DT(KT)
      YH=Y(KT)-(0.5*VY-0.125*AY*DT(KT))*DT(KT)
      CALL NEWTON(XH,YH,EX,EY,PSI,IK)
      IF(IK.NE.0) GO TO 400
      AXH=QM*EX
      AYH=QM*EY
200   XH=X(KT)+(0.5*VX+(4.0*AX-AXH)*DT(KT)/24.0)*DT(KT)
      YH=Y(KT)+(0.5*VY+(4.0*AY-AYH)*DT(KT)/24.0)*DT(KT)
      CALL NEWTON(XH,YH,EX,EY,PSI,IK)
      IF(IK.NE.0) GO TO 400
      AXH=QM*EX
      AYH=QM*EY
      X(KT+1)=X(KT)+(VX+(AX+2.0*AXH)*DT(KT)/6.0)*DT(KT)
      Y(KT+1)=Y(KT)+(VY+(AY+2.0*AYH)*DT(KT)/6.0)*DT(KT)
260   FORMAT(2I)
      AAX=AX
      AAY=AY
      CALL NEWTON(X(KT+1),Y(KT+1),EX,EY,PSI,IK)
      IF(IK.NE.0) GO TO 400
      AX=QM*EX
      AY=QM*EY
      A1SQ=AAX**2+AAY**2
      IF(A1SQ.LT.1.D-6) GO TO 300
      DELD=DSQRT((X(KT+1)-X(KT))**2+(Y(KT+1)-Y(KT))**2)
275   FORMAT(' DELD=',D12.6,' DIM=',D12.6)
      IF(DELD.GT.DIM/40) GO TO 320
      IF(DELD.LT.DIM/120) GO TO 330
280   VX=VX+DT(KT)*(AX+4.0*AXH+AAX)/6.0
      VY=VY+DT(KT)*(AY+4.0*AYH+AAY)/6.0
      DT(KT+1)=DT(KT)
      GO TO 440
300   V1SQ=VX**2+VY**2
      DVSQ=A1SQ*DT(KT)**2
      IF(V1SQ.GT.DVSQ*1.D4) GO TO 280
      IK=1
      GO TO 410
320   DT(KT)=DT(KT)/1.1
      GO TO 100
```

```
330     DT(KT)=DT(KT)*1.1
        GO TO 100
400     IF(IK.EQ.-1) TYPE 420
        GO TO 440
410     TYPE 430
420     FORMAT(' ELECTRON HAS ESCAPED POTENTIAL REGION
     1  INCLUDED IN X-Y COORDINATE GRID')
430     FORMAT(' PARTICLE VELOCITY AND ACCELERATION BOTH
     1  EQUAL ZERO')
440     RETURN
        END
        SUBROUTINE VPLOT
        IMPLICIT REAL *8 (A-H,O-Z)
        COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
        DIMENSION X(250),Y(250)
        CALL INITT
        CALL SELINI
        CALL DWINDO(SNGL(XX(1)),SNGL(XX(82)),SNGL(YY(1)),SNGL(YY(22)))
        II=INT(SNGL(780*DY/DX))
        CALL TWINDO(0,780,0,II)
        CALL MOVEA(SNGL(XX(2)),SNGL(YY(2)))
        CALL DRAWA(SNGL(XX(82)),SNGL(YY(2)))
        PHIMIN=1.0D38
        PHIMAX=-1.0D38
        DO 10 I=2,82
        DO 5 J=2,22
        IF(PHI(I,J).GT.PHIMIN) GO TO 2
        PHIMIN=PHI(I,J)
 2      IF(PHI(I,J).LT.PHIMAX) GO TO 5
        PHIMAX=PHI(I,J)
 5      CONTINUE
10      CONTINUE
        K=0
12      K=K+1
        V=PHIMIN+K*0.2
        L=0
        DO 20 J=2,22
        DO 15 I=3,82
        IF((PHI(I-1,J)-V)*(PHI(I,J)-V).GE.0.0) GO TO 15
        L=L+1
        X(L)=XX(I-1)+(XX(I)-XX(I-1))*(V-PHI(I-1,J))/(PHI(I,J)-
     1  PHI(I-1,J))
        Y(L)=YY(J)
15      CONTINUE
20      CONTINUE
        DO 30 I=2,82
        DO 25 J=3,22
        IF((PHI(I,J-1)-V)*(PHI(I,J)-V).GE.0.0) GO TO 25
        L=L+1
        X(L)=XX(I)
        Y(L)=YY(J-1)+(YY(J)-YY(J-1))*(V-PHI(I,J-1))/(PHI(I,J)-
     1  PHI(I,J-1))
25      CONTINUE
```

```
30      CONTINUE
        LMAX=L
        XMIN=XX(82)
        DO 35 L=1,LMAX
        IF(X(L).GT.XMIN) GO TO 35
        XMIN=X(L)
        LMIN=L
35      CONTINUE
        XS=X(1)
        YS=Y(1)
        X(1)=X(LMIN)
        Y(1)=Y(LMIN)
        X(LMIN)=XS
        Y(LMIN)=YS
        CALL MOVEA(SNGL(X(1)),SNGL(Y(1)))
        LL=1
40      LL=LL+1
        DSQ=1.0D38
        DO 45 L=LL,LMAX
        DSQL=(X(L)-X(LL-1))**2+(Y(L)-Y(LL-1))**2
        IF(DSQL.GT.DSQ) GO TO 45
        DSQ=DSQL
        LDMIN=L
45      CONTINUE
        XS=X(LL)
        YS=Y(LL)
        X(LL)=X(LDMIN)
        Y(LL)=Y(LDMIN)
        X(LDMIN)=XS
        Y(LDMIN)=YS
        CALL DRAWA(SNGL(X(LL)),SNGL(Y(LL)))
        IF(LL.EQ.LMAX) GO TO 50
        GO TO 40
50      IF(V+0.2.LE.PHIMAX) GO TO 12
        CALL FINITT(800,500)
        RETURN
        END
```

APPENDIX E

Fortran Routines for Quadrupole Synthesis of Solutions to The Model Problem

```
     cROUTINE QPOLE.F4
       IMPLICIT REAL *8(A-H,O-Z)
       COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
       COMMON/DATA/XO(50),YO(50),VXO(50),VYO(50),XT(50),YT(50)
       COMMON/SAVE/X(110),Y(110),DT(250),KT
       DIMENSION XS(110),YS(110),TD(110),DDX(50)
     1  ,YN(50),XN(50)
       EXTERNAL TEKHAN,PLTHAN,SCPHAN
       CALL INITTG
       TYPE 1
  1    FORMAT(' ENTER X AND Y DIMENSIONS OF POTENTIAL REGION',$)
       ACCEPT 2,DX,DY
  2    FORMAT(2D)
       TYPE 19
 19    FORMAT(' ENTER 1 TO INPUT DATA FROM FILE TRAJ.DAT, ENTER 0
     1  TO INPUT DATA MANUALLY',$)
       ACCEPT 4, NT
       IF(NT.EQ.1) GO TO 140
       TYPE 3
  3    FORMAT(' ENTER NUMBER OF ELECTRON TRAJECTORIES',$)
       ACCEPT 4,KK
  4    FORMAT(I)
       DO 8 K=1,KK
       TYPE 5
  5    FORMAT(' ENTER INITIAL X COORDINATE AND VELOCITY',$)
       ACCEPT 2,XO(K),VXO(K)
       TYPE 6
  6    FORMAT(' ENTER INITIAL Y COORDINATE AND VELOCITY',$)
       ACCEPT 2,YO(K),VYO(K)
       TYPE 7
  7    FORMAT(' ENTER FINAL X AND Y COORDINATES',$)
       ACCEPT 2,XT(K),YT(K)
       type 67,k,xO(k),xt(k)
 67    format(i2,2d12.6)
  8    CONTINUE
  9    EQ=1.60210D-19
       EM=9.1091D-31
       QM=EQ/EM
       AO=2*VXO(KK)*VYO(KK)/(QM*(XT(KK)-XO(KK)))
       TYPE 15,AO
 15    FORMAT(' AO=',D)
       XX(1)=-DX/80
       XX(2)=0.0
       DO 20 I=3,82
       XX(I)=XX(I-1)+DX/80
 20    CONTINUE
       YY(1)=-DY/20
       YY(2)=0.0
       DO 30 J=3,32
       YY(J)=YY(J-1)+DY/20
 30    CONTINUE
 32    DO 50 I=1,82
       DO 40 J=1,32
```

```
      PHI(I,J)=A0*YY(J)
40    CONTINUE
50    CONTINUE
      PAUSE
      IT=0
60    DD=0.0
      D=0.0
      TYPE 65,IT
65    FORMAT(' ITERATION=',I4)
      IT=IT+1
      DO 90 K=1,KK-1
      YN(K)=0.0
      KT=0
      TYPE 82,XO(K),YO(K)
      DT(1)=DY/(100*VYO(K))
70    KT=KT+1
      CALL DVOGEL(K)
      IF(Y(KT+1).LT.YN(K)) GO TO 75
      YN(K)=Y(KT+1)
      XN(K)=X(KT+1)
75    IF(Y(KT+1).GT.YO(K)) GO TO 70
80    TYPE 82,X(KT+1),Y(KT+1)
      DDX(K)=(XT(K)-X(KT+1))/(XT(K)-XO(K))
82    FORMAT(' X=',D12.4,'    Y=',D12.4)
      DD=DD+DABS(XT(K)-X(KT+1))
      IF(D.GT.DABS(DDX(K))) GO TO 90
      D=DABS(DDX(K))
      XM=X(KT+1)
      KM=K
      DO 85 KI=1,KT+1
      XS(KI)=X(KI)
      YS(KI)=Y(KI)
      TD(KI)=DT(KI)
85    CONTINUE
90    CONTINUE
      TYPE 95,D
      TYPE 95,DD
95    FORMAT(D)
      IF(DD.LT.0.01*DX) GO TO 126
      XL=XN(KM)
      YL=1.4*DY
      VXII=0.0
      DVXII=0.0
      VXXI=0.0
      DO 98 I=1,KT
      DVXI=DVXII
      DVXII=((YS(I)+YL)**3-3*(YS(I)+YL)*(XS(I)-XL)
     1**2)/((XS(I)-XL)**2+(YS(I)+YL)**2)**3
     2+((YS(I)-YL)**3-3*(YS(I)-YL)*(XS(I)-XL)**2
     3)/((XS(I)-XL)**2+(YS(I)-YL)**2)**3
      VXI=VXII
      VXII=VXI+0.5*(DVXII+DVXI)*TD(I)
      VXXI=VXXI+0.5*(VXII+VXI)*TD(I)
```

```
 98     CONTINUE
        AL=(XM-XT(KM))/(QM*VXXI)
        TYPE 95,AL
        IF(AL.LT.-0.1) GO TO 100
        IF(AL.GT.0.1) GO TO 99
        GO TO 101
 99     AL=0.1
        GO TO 101
100     AL=-0.1
101     TYPE 105,AL,XL,YL
105     FORMAT(' AL=',D12.5,' XL=',D12.5, ' YL=',D12.5)
        DO 120 I=2,82
        DO 110 J=2,32
        PHI(I,J)=PHI(I,J)+AL*(XX(I)-XL)*((YY(J)+YL)/((
     1  XX(I)-XL)**2+(YY(J)+YL)**2)**2+(YY(J)-YL)/
     2  ((XX(I)-XL)**2+(YY(J)-YL)**2)**2)
110     CONTINUE
120     CONTINUE
        KT=0
        DT(1)=DY/(100*VY0(KK))
        TYPE 82,X0(KK),Y0(KK)
121     KT=KT+1
        CALL DVOGEL(KK)
        IF(Y(KT+1).GT.Y0(KK)) GO TO 121
        TYPE 82,X(KT+1),Y(KT+1)
        SF=(X(KT+1)-X0(KK))/(XT(KK)-X0(KK))
        TYPE 122,SF
122     FORMAT(' SF=',D12.6)
        DO 125 I=2,82
        DO 124 J=2,32
        PHI(I,J)=SF*PHI(I,J)
124     CONTINUE
125     CONTINUE
        GO TO 60
126     DO 128 I=10,70
        TYPE 129 ,PHI(I,3),PHI(I,5),PHI(I,7),PHI(I,9),PHI(I,11)
     1  ,PHI(I,13),PHI(I,15),PHI(I,17),PHI(I,19),PHI(I,21)
128     CONTINUE
129     FORMAT(10D12.4)
130     CALL VPLOT
        GO TO 160
140     OPEN(UNIT=1,FILE='TRAJ.DAT')
        READ (1,4) KK
        DO 150 K=1,KK
        READ(1,2) X0(K),VX0(K)
        READ(1,2) Y0(K),VY0(K)
        READ(1,2) XT(K),YT(K)
        type 155,x0(k),xt(k)
155     format(' x0=',d12.6 ,'    xt=',d12.6)
150     CONTINUE
        GO TO 9
160     STOP
        END
```

```
      SUBROUTINE NEWTON(X,Y,EX,EY,PSI,IK)
      IMPLICIT REAL *8(A-H,O-Z)
      COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
      DIMENSION FM(4,4,4),FMN(4,4,4),XM(4),XM1(4),YN(4),YN1(4)
    5 FORMAT(2D12.6)
      IF(X.GT.XX(82)) GO TO 220
      IF(X.LT.XX(1)) GO TO 220
      IF(Y.GT.YY(22)) GO TO 220
      IF(Y.LT.YY(1)) GO TO 225
      HX=DX/80.0
      HY=DY/20.0
      I=0
   10 I=I+1
      IF(X.GT.XX(I)) GO TO 10
      IF(I.LE.3) GO TO 12
      IB=I-2
      GO TO 14
   12 IB=2
   14 IF(I.EQ.82) GO TO 16
      IE=I+1
      GO TO 18
   16 IE=I
   18 J=0
   20 J=J+1
      IF(Y.GT.YY(J)) GO TO 20
      IF(J.LE.3) GO TO 25
      JB=J-2
      GO TO 30
   25 JB=2
   30 IF(J.EQ.32) GO TO 35
      JE=J+1
      GO TO 40
   35 JE=J
   40 MM=IE-IB+1
      XM(1)=1.0
      XM1(1)=0.0
      DO 70 M=2,MM
      IM=IB+M-2
      XM(M)=XM(M-1)*(X-XX(IM))
      XM1(M)=0.0
      DO 60 I=2,M
      X1=1.0
      DO 50 K=2,M
      IF(K.EQ.I) GO TO 50
      X1=X1*(X-XX(IB+K-2))
   50 CONTINUE
      XM1(M)=XM1(M)+X1
   60 CONTINUE
   70 CONTINUE
      NN=JE-JB+1
      YN(1)=1.0
      YN1(1)=0.0
      DO 100 N=2,NN
```

```
      JN=JB+N-2
      YN(N)=YN(N-1)*(Y-YY(JN))
      YN1(N)=0.0
      DO 90 J=2,N
      Y1=1.0
      DO 80 L=2,N
      IF(L.EQ.J) GO TO 80
      Y1=Y1*(Y-YY(JB+L-2))
 80   CONTINUE
      YN1(N)=YN1(N)+Y1
 90   CONTINUE
100   CONTINUE
      DO 110 M=1,MM
      DO 105 N=1,NN
      FM(1,M,N)=PHI(IB+M-1,JB+N-1)
105   CONTINUE
110   CONTINUE
      DO 140 L=2,NN
      DO 130 N=L,NN
      DO 120 M=1,MM
      FM(L,M,N)=(FM(L-1,M,N-1)-FM(L-1,M,N))/(HY*(1-L))
120   CONTINUE
130   CONTINUE
140   CONTINUE
      DO 160 M=1,MM
      DO 150 N=1,NN
      FMN(1,M,N)=FM(N,M,N)
150   CONTINUE
160   CONTINUE
      DO 190 K=2,MM
      DO 180 M=K,MM
      DO 170 N=1,NN
      FMN(K,M,N)=(FMN(K-1,M-1,N)-FMN(K-1,M,N))/(HX*(1-K))
170   CONTINUE
180   CONTINUE
190   CONTINUE
      EX=0.0
      EY=0.0
      PSI=0.0
      DO 210 N=1,NN
      DO 200 M=1,MM
      EX=EX-XM1(M)*YN(N)*FMN(M,M,N)
      EY=EY-YN1(N)*XM(M)*FMN(M,M,N)
      PSI=PSI+XM(M)*YN(N)*FMN(M,M,N)
200   CONTINUE
210   CONTINUE
      GO TO 230
220   IK=-1
      GO TO 230
225   IK=2
230   RETURN
      END
      SUBROUTINE DVOGEL(K)
```

```
      IMPLICIT REAL *8(A-H,O-Z)
      COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
      COMMON/DATA/X0(50),Y0(50),VX0(50),VY0(50),XT(50),YT(50)
      COMMON/SAVE/X(110),Y(110),DT(250),KT
      DIM=DSQRT(DX**2+DY**2)
      QM=1.76D11
      IK=0
      IF(KT.GT.1) GO TO 200
      VX=VX0(K)
      VY=VY0(K)
      X(1)=X0(K)
      Y(1)=Y0(K)
100   CALL NEWTON(X(KT),Y(KT),EX,EY,PSI,IK)
101   FORMAT(' EX=',D12.4,'EY=',D12.4)
      IF(IK.NE.0) GO TO 400
      AX=QM*EX
      AY=QM*EY
      XH=X(KT)-(0.5*VX-0.125*AX*DT(KT))*DT(KT)
      YH=Y(KT)-(0.5*VY-0.125*AY*DT(KT))*DT(KT)
      CALL NEWTON(XH,YH,EX,EY,PSI,IK)
      IF(IK.NE.0) GO TO 400
      AXH=QM*EX
      AYH=QM*EY
200   XH=X(KT)+(0.5*VX+(4.0*AX-AXH)*DT(KT)/24.0)*DT(KT)
      YH=Y(KT)+(0.5*VY+(4.0*AY-AYH)*DT(KT)/24.0)*DT(KT)
      CALL NEWTON(XH,YH,EX,EY,PSI,IK)
      IF(IK.NE.0) GO TO 400
      AXH=QM*EX
      AYH=QM*EY
      X(KT+1)=X(KT)+(VX+(AX+2.0*AXH)*DT(KT)/6.0)*DT(KT)
      Y(KT+1)=Y(KT)+(VY+(AY+2.0*AYH)*DT(KT)/6.0)*DT(KT)
260   FORMAT(2I)
      AAX=AX
      AAY=AY
      CALL NEWTON(X(KT+1),Y(KT+1),EX,EY,PSI,IK)
      IF(IK.NE.0) GO TO 400
      AX=QM*EX
      AY=QM*EY
      A1SQ=AAX**2+AAY**2
      IF(A1SQ.LT.1.D-6) GO TO 300
      DELD=DSQRT((X(KT+1)-X(KT))**2+(Y(KT+1)-Y(KT))**2)
275   FORMAT(' DELD=',D12.6,' DIM=',D12.6)
      IF(DELD.GT.DIM/40) GO TO 320
      IF(DELD.LT.DIM/120) GO TO 330
280   VX=VX+DT(KT)*(AX+4.0*AXH+AAX)/6.0
      VY=VY+DT(KT)*(AY+4.0*AYH+AAY)/6.0
      DT(KT+1)=DT(KT)
      GO TO 440
300   V1SQ=VX**2+VY**2
      DVSQ=A1SQ*DT(KT)**2
      IF(V1SQ.GT.DVSQ*1.D4) GO TO 280
      IK=1
      GO TO 410
```

```
320     DT(KT)=DT(KT)/1.1
        GO TO 100
330     DT(KT)=DT(KT)*1.1
        GO TO 100
400     IF(IK.EQ.-1) TYPE 420
        GO TO 440
410     TYPE 430
420     FORMAT(' ELECTRON HAS ESCAPED POTENTIAL REGION
     1  INCLUDED IN X-Y COORDINATE GRID')
430     FORMAT(' PARTICLE VELOCITY AND ACCELERATION BOTH
     1  EQUAL ZERO')
440     RETURN
        END
        SUBROUTINE VPLOT
        IMPLICIT REAL *8 (A-H,O-Z)
        COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
        DIMENSION X(250),Y(250)
        CALL INITT
        CALL SELINI
        CALL DWINDO(SNGL(XX(1)),SNGL(XX(82)),SNGL(YY(1)),SNGL(YY(22)))
        II=INT(SNGL(780*DY/DX))
        CALL TWINDO(0,780,0,II)
        CALL MOVEA(SNGL(XX(2)),SNGL(YY(2)))
        CALL DRAWA(SNGL(XX(82)),SNGL(YY(2)))
        PHIMIN=1.0D38
        PHIMAX=-1.0D38
        DO 10 I=2,82
        DO 5 J=2,22
        IF(PHI(I,J).GT.PHIMIN) GO TO 2
        PHIMIN=PHI(I,J)
 2      IF(PHI(I,J).LT.PHIMAX) GO TO 5
        PHIMAX=PHI(I,J)
 5      CONTINUE
10      CONTINUE
        K=0
12      K=K+1
        V=PHIMIN+K*0.2
        L=0
        DO 20 J=2,22
        DO 15 I=3,82
        IF((PHI(I-1,J)-V)*(PHI(I,J)-V).GE.0.0) GO TO 15
        L=L+1
        X(L)=XX(I-1)+(XX(I)-XX(I-1))*(V-PHI(I-1,J))/(PHI(I,J)-
     1  PHI(I-1,J))
        Y(L)=YY(J)
15      CONTINUE
20      CONTINUE
        DO 30 I=2,82
        DO 25 J=3,22
        IF((PHI(I,J-1)-V)*(PHI(I,J)-V).GE.0.0) GO TO 25
        L=L+1
        X(L)=XX(I)
        Y(L)=YY(J-1)+(YY(J)-YY(J-1))*(V-PHI(I,J-1))/(PHI(I,J)-
```

```
     1  PHI(I,J-1))
25      CONTINUE
30      CONTINUE
        LMAX=L
        XMIN=XX(82)
        DO 35 L=1,LMAX
        IF(X(L).GT.XMIN) GO TO 35
        XMIN=X(L)
        LMIN=L
35      CONTINUE
        XS=X(1)
        YS=Y(1)
        X(1)=X(LMIN)
        Y(1)=Y(LMIN)
        X(LMIN)=XS
        Y(LMIN)=YS
        CALL MOVEA(SNGL(X(1)),SNGL(Y(1)))
        LL=1
40      LL=LL+1
        DSQ=1.0D38
        DO 45 L=LL,LMAX
        DSQL=(X(L)-X(LL-1))**2+(Y(L)-Y(LL-1))**2
        IF(DSQL.GT.DSQ) GO TO 45
        DSQ=DSQL
        LDMIN=L
45      CONTINUE
        XS=X(LL)
        YS=Y(LL)
        X(LL)=X(LDMIN)
        Y(LL)=Y(LDMIN)
        X(LDMIN)=XS
        Y(LDMIN)=YS
        CALL DRAWA(SNGL(X(LL)),SNGL(Y(LL)))
        IF(LL.EQ.LMAX) GO TO 50
        GO TO 40
50      IF(V+0.2.LE.PHIMAX) GO TO 12
        CALL FINITT(800,500)
        RETURN
        END
```

```
      cROUTINE QPOLES.F4
        IMPLICIT REAL *8(A-H,O-Z)
        COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
        COMMON/DATA/XO(50),YO(50),VXO(50),VYO(50),XT(50),YT(50)
        COMMON/SAVE/X(110),Y(110),DT(110),KI(50)
        COMMON/COEFF/VI(5,6),AL(5)
        DIMENSION XN(6)
        EXTERNAL TEKHAN,PLTHAN,SCPHAN
        CALL INITTC
        TYPE 1
    1   FORMAT(' ENTER X AND Y DIMENSIONS OF POTENTIAL REGION',$)
        ACCEPT 2,DX,DY
    2   FORMAT(2D)
        TYPE 19
   19   FORMAT(' ENTER 1 TO INPUT DATA FROM FILE TRAJ.DAT, ENTER 0
       1  TO INPUT DATA MANUALLY',$)
        ACCEPT 4, NT
        IF(NT.EQ.1) GO TO 140
        TYPE 3
    3   FORMAT(' ENTER NUMBER OF ELECTRON TRAJECTORIES',$)
        ACCEPT 4,KK
    4   FORMAT(I)
        DO 8 K=1,KK
        TYPE 5
    5   FORMAT(' ENTER INITIAL X COORDINATE AND VELOCITY',$)
        ACCEPT 2,XO(K),VXO(K)
        TYPE 6
    6   FORMAT(' ENTER INITIAL Y COORDINATE AND VELOCITY',$)
        ACCEPT 2,YO(K),VYO(K)
        TYPE 7
    7   FORMAT(' ENTER FINAL X AND Y COORDINATES',$)
        ACCEPT 2,XT(K),YT(K)
        type 670,k,xO(k),xt(k)
  670   format(i2,2d12.6)
    8   CONTINUE
    9   EQ=1.60210D-19
        EM=9.1091D-31
        QM=EQ/EM
        AO=2*VXO(KK)*VYO(KK)/(QM*(XT(KK)-XO(KK)))
        TYPE 15,AO
   15   FORMAT(' AO=',D)
        XX(1)=-DX/80
        XX(2)=0.0
        DO 20 I=3,82
        XX(I)=XX(I-1)+DX/80
   20   CONTINUE
        YY(1)=-DY/20
        YY(2)=0.0
        DO 30 J=3,32
        YY(J)=YY(J-1)+DY/20
   30   CONTINUE
   32   DO 50 I=1,82
        DO 40 J=1,32
```

```
      PHI(I,J)=AO*YY(J)
40    CONTINUE
50    CONTINUE
      IT=0
      ITT=0
      DD=1.0D30
69    D=DD
      DD=0.0
      RD=0.0
      DO 90 K=1,KK
      KT=0
      YMAX=0.0
      DT(1)=DY/(100*VYO(K))
      TYPE 80,XO(K),YO(K)
70    KT=KT+1
      CALL DVOGEL(K,KT,KIK)
      IF(Y(KT+1).LT.YMAX) GO TO 75
      YMAX=Y(KT+1)
      XN(K)=X(KT+1)
75    IF(Y(KT+1).GT.YO(K)) GO TO 70
      TYPE 80, X(KT+1),Y(KT+1)
      IF(K.EQ.KK) GO TO 90
80    FORMAT(' X=',D12.6,' Y=',D12.6)
      DD=DD+DABS(XT(K)-X(KT+1))
      RD=RD+(X(KT+1)-XO(K))/(XT(K)-XO(K))
85    FORMAT(' IT=',I3,' DD=',D12.6,' RD=',D12.6)
      YL=1.4*DY
      DO 87 L=1,KK-1
      XL=XN(L)
      VII=0.0
      DVII=0.0
      VI(K,L)=0.0
      DO 86 I=1,KT
      DVI=DVII
      DVII=((Y(I)+YL)**3-3*(Y(I)+YL)*(X(I)-XL)**2)/((X(I)-XL)**2
     1 +(Y(I)+YL)**2)**3+((Y(I)-YL)**3-3*(Y(I)-YL)*(X(I)
     2 -XL)**2)/((X(I)-XL)**2+(Y(I)-YL)**2)**3
      VVI=VII
      VII=VVI+0.5*(DVII+DVI)*DT(I)
      VI(K,L)=VI(K,L)+0.5*(VII+VVI)*DT(I)
86    CONTINUE
      VI(K,L)=VI(K,L)*QM
      TYPE 88,K,L,VI(K,L)
87    CONTINUE
88    FORMAT(' K=',I3,' L=',I3,' VI(K,L)=',D12.6)
      VI(K,KK)=X(KT+1)-XT(K)
      TYPE 89,K,VI(K,KK)
89    FORMAT(' K=',I3,' DISCREPANCY=',D12.6)
90    CONTINUE
      TYPE 85,IT,DD,RD
      IF(D.LT.DD) ITT=ITT+1
      IF(ITT.GT.3) GO TO 125
      IF(DD.LT..01*DX) GO TO 126
```

```
      ERR=DABS((X(KT+1)-XT(KK))/XT(KK))
      IF(ERR.LT.0.03) GO TO 905
      SF=(X(KT+1)-X0(KK))/(XT(KK)-X0(KK))
      TYPE 901,SF
901   FORMAT(' SCALE FACTOR=',D12.6)
      DO 903 I=2,82
      DO 902 J=2,32
      PHI(I,J)=SF*PHI(I,J)
902   CONTINUE
903   CONTINUE
      GO TO 69
905   CALL GAUSS(KK-1)
      AA= .0
      DO 906 K=1,KK-1
      IF(DABS(AL(K)).LT.AA) GO TO 906
      AA=DABS(AL(K))
906   CONTINUE
      DO 908 K=1,KK-1
      IF(AA.LT.0.1) GO TO 908
      AL(K)=0.1*AL(K)/AA
908   CONTINUE
      IT=IT+1
      YL=1.4*DY
      DO 95 L=1,KK-1
      XL=XN(L)
      TYPE 91,AL(L),XL,YL
 91   FORMAT(' AL=',D12.6,' XL=',D12.6,' YL=',D12.6)
      DO 93 I=2,82
      DO 92 J=2,32
      PHI(I,J)=PHI(I,J)+AL(L)*(XX(I)-XL)*((YY(J)+YL)/((XX(I)-XL)**2
     1 +(YY(J)+YL)**2)**2+(YY(J)-YL)/((XX(I)-XL)**2
     2 +(YY(J)-YL)**2)**2)
 92   CONTINUE
 93   CONTINUE
 95   CONTINUE
      GO TO 69
124   FORMAT(' FOUR DEPARTURES FROM MONOTONE DECREASING
     1 SUM OF ABSOLUTE DISCREPANCIES HAVE OCCURED')
125   TYPE 124
126   DO 128 I=10,70
      TYPE 129,PHI(I,3),PHI(I,5),PHI(I,7),PHI(I,9),PHI(I,11),
     1 PHI(I,13),PHI(I,15),PHI(I,17),PHI(I,19),PHI(I,21)
128   CONTINUE
129   FORMAT(10D12.6)
130   CALL VPLOT
      GO TO 160
140   OPEN(UNIT=1,FILE='TRAJ.DAT')
      READ (1,4) KK
      DO 150 K=1,KK
      READ(1,2) X0(K),VX0(K)
      READ(1,2) Y0(K),VY0(K)
      READ(1,2) XT(K),YT(K)
      type 155,x0(k),xt(k)
```

```
155     format(' x0=',d12.6 ,'    xt=',d12.6)
150     CONTINUE
        GO TO 9
160     STOP
        END
        SUBROUTINE GAUSS(N)
        IMPLICIT REAL *8(A-H,O-Z)
        COMMON/COEFF/VI(5,6),AL(5)
        DO 20 K=1,N-1
        B=VI(K,K)
        DO 5 J=K,N+1
        VI(K,J)=VI(K,J)/B
  5     CONTINUE
        DO 15 I=K+1,N
        B=VI(I,K)
        DO 10 J=K,N+1
        VI(I,J)=VI(I,J)-B*VI(K,J)
 10     CONTINUE
 15     CONTINUE
 20     CONTINUE
        AL(N)=VI(N,N+1)/VI(N,N)
        K=N
 25     K=K-1
        AL(K)=VI(K,N+1)
        DO 30 J=K+1,N
        AL(K)=AL(K)-VI(K,J)*AL(J)
 30     CONTINUE
        IF(K.GT.1) GO TO 25
        RETURN
        END
        SUBROUTINE NEWTON(X,Y,EX,EY,PSI,IK)
        IMPLICIT REAL *8(A-H,O-Z)
        COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
        DIMENSION FM(4,4,4),FMN(4,4,4),XM(4),XM1(4),YN(4),YN1(4)
  5     FORMAT(2D12.6)
        IF(X.GT.XX(82)) GO TO 220
        IF(X.LT.XX(1)) GO TO 220
        IF(Y.GT.YY(32)) GO TO 225
        IF(Y.LT.YY(1)) GO TO 220
        HX=DX/80.0
        HY=DY/20.0
        I=0
 10     I=I+1
        IF(X.GT.XX(I)) GO TO 10
        IF(I.LE.3) GO TO 12
        IB=I-2
        GO TO 14
12      IB=2
14      IF(I.EQ.82) GO TO 16
        IE=I+1
        GO TO 18
16      IE=I
18      J=0
```

```
20      J=J+1
        IF(Y.GT.YY(J)) GO TO 20
        IF(J.LE.3) GO TO 25
        JB=J-2
        GO TO 30
25      JB=2
30      IF(J.EQ.32) GO TO 35
        JE=J+1
        GO TO 40
35      JE=J
40      MM=IE-IB+1
        XM(1)=1.0
        XM1(1)=0.0
        DO 70 M=2,MM
        IM=IB+M-2
        XM(M)=XM(M-1)*(X-XX(IM))
        XM1(M)=0.0
        DO 60 I=2,M
        X1=1.0
        DO 50 K=2,M
        IF(K.EQ.I) GO TO 50
        X1=X1*(X-XX(IB+K-2))
50      CONTINUE
        XM1(M)=XM1(M)+X1
60      CONTINUE
70      CONTINUE
        NN=JE-JB+1
        YN(1)=1.0
        YN1(1)=0.0
        DO 100 N=2,NN
        JN=JB+N-2
        YN(N)=YN(N-1)*(Y-YY(JN))
        YN1(N)=0.0
        DO 90 J=2,N
        Y1=1.0
        DO 80 L=2,N
        IF(L.EQ.J) GO TO 80
        Y1=Y1*(Y-YY(JB+L-2))
80      CONTINUE
        YN1(N)=YN1(N)+Y1
90      CONTINUE
100     CONTINUE
        DO 110 M=1,MM
        DO 105 N=1,NN
        FM(1,M,N)=PHI(IB+M-1,JB+N-1)
105     CONTINUE
110     CONTINUE
        DO 140 L=2,NN
        DO 130 N=L,NN
        DO 120 M=1,MM
        FM(L,M,N)=(FM(L-1,M,N-1)-FM(L-1,M,N))/(HY*(1-L))
120     CONTINUE
130     CONTINUE
```

```
140     CONTINUE
        DO 160 M=1,MM
        DO 150 N=1,NN
        FMN(1,M,N)=FM(N,M,N)
150     CONTINUE
160     CONTINUE
        DO 190 K=2,MM
        DO 180 M=K,MM
        DO 170 N=1,NN
        FMN(K,M,N)=(FMN(K-1,M-1,N)-FMN(K-1,M,N))/(HX*(1-K))
170     CONTINUE
180     CONTINUE
190     CONTINUE
        EX=0.0
        EY=0.0
        PSI=0.0
        DO 210 N=1,NN
        DO 200 M=1,MM
        EX=EX-XM1(M)*YN(N)*FMN(M,M,N)
        EY=EY-YN1(N)*XM(M)*FMN(M,M,N)
        PSI=PSI+XM(M)*YN(N)*FMN(M,M,N)
200     CONTINUE
210     CONTINUE
        GO TO 230
220     IK=-1
        GO TO 230
225     IK=2
230     RETURN
        END
        SUBROUTINE DVOGEL(K,KT,KIK)
        IMPLICIT REAL *8(A-H,O-Z)
        COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
        COMMON/DATA/X0(50),Y0(50),VX0(50),VY0(50),XT(50),YT(50)
        COMMON/SAVE/X(110),Y(110),DT(110),KI(50)
        DIM=DSQRT(DX**2+DY**2)
        QM=1.76D11
        IK=0
        IF(KT.GT.1) GO TO 200
        VX=VX0(K)
        VY=VY0(K)
        X(1)=X0(K)
        Y(1)=Y0(K)
100     CALL NEWTON(X(KT),Y(KT),EX,EY,PSI,IK)
101     FORMAT(' EX=',D12.4,'EY=',D12.4)
        IF(IK.NE.0) GO TO 400
        AX=QM*EX
        AY=QM*EY
        XH=X(KT)-(0.5*VX-0.125*AX*DT(KT))*DT(KT)
        YH=Y(KT)-(0.5*VY-0.125*AY*DT(KT))*DT(KT)
        CALL NEWTON(XH,YH,EX,EY,PSI,IK)
        IF(IK.NE.0) GO TO 400
        AXH=QM*EX
        AYH=QM*EY
```

```
200     XH=X(KT)+(0.5*VX+(4.0*AX-AXH)*DT(KT)/24.0)*DT(KT)
        YH=Y(KT)+(0.5*VY+(4.0*AY-AYH)*DT(KT)/24.0)*DT(KT)
        CALL NEWTON(XH,YH,EX,EY,PSI,IK)
        IF(IK.NE.0) GO TO 400
        AXH=QM*EX
        AYH=QM*EY
        X(KT+1)=X(KT)+(VX+(AX+2.0*AXH)*DT(KT)/6.0)*DT(KT)
        Y(KT+1)=Y(KT)+(VY+(AY+2.0*AYH)*DT(KT)/6.0)*DT(KT)
260     FORMAT(2I)
        AAX=AX
        AAY=AY
        CALL NEWTON(X(KT+1),Y(KT+1),EX,EY,PSI,IK)
        IF(IK.NE.0) GO TO 400
        AX=QM*EX
        AY=QM*EY
        A1SQ=AAX**2+AAY**2
        IF(A1SQ.LT.1.D-6) GO TO 300
        DELD=DSQRT((X(KT+1)-X(KT))**2+(Y(KT+1)-Y(KT))**2)
275     FORMAT(' DELD=',D12.6,' KT=',I3)
        IF(DELD.GT.DIM/40) GO TO 320
        IF(DELD.LT.DIM/120) GO TO 330
280     VX=VX+DT(KT)*(AX+4.0*AXH+AAX)/6.0
        VY=VY+DT(KT)*(AY+4.0*AYH+AAY)/6.0
        DT(KT+1)=DT(KT)
        GO TO 440
300     V1SQ=VX**2+VY**2
        DVSQ=A1SQ*DT(KT)**2
        IF(V1SQ.GT.DVSQ*1.D4) GO TO 280
        IK=1
        GO TO 410
320     DT(KT)=DT(KT)/1.1
        GO TO 100
330     DT(KT)=DT(KT)*1.1
        GO TO 100
400     IF(IK.EQ.-1) TYPE 420
        IF(IK.EQ.2) GO TO 402
        GO TO 440
402     KI(K)=1
        KIK=1
        TYPE 425
        GO TO 440
410     TYPE 430
420     FORMAT(' ELECTRON HAS ESCAPED COORDINATE GRID')
425     FORMAT(' ELECTRON HAS ESCAPED POTENTIAL REGION')
430     FORMAT(' PARTICLE VELOCITY AND ACCELERATION BOTH
     1  EQUAL ZERO')
440     RETURN
        END
        SUBROUTINE VPLOT
        IMPLICIT REAL *8 (A-H,O-Z)
        COMMON/COORD/PHI(82,32),XX(82),YY(32),DX,DY
        DIMENSION X(250),Y(250)
        CALL INITT
```

```
      CALL SELINI
      CALL DWINDO(SNGL(XX(1)),SNGL(XX(82)),SNGL(YY(1)),SNGL(YY(22)))
      II=INT(SNGL(780*DY/DX))
      CALL TWINDO(0,780,0,II)
      CALL MOVEA(SNGL(XX(2)),SNGL(YY(2)))
      CALL DRAWA(SNGL(XX(82)),SNGL(YY(2)))
      PHIMIN=1.0D38
      PHIMAX=-1.0D38
      DO 10 I=2,82
      DO 5 J=2,22
      IF(PHI(I,J).GT.PHIMIN) GO TO 2
      PHIMIN=PHI(I,J)
 2    IF(PHI(I,J).LT.PHIMAX) GO TO 5
      PHIMAX=PHI(I,J)
 5    CONTINUE
10    CONTINUE
      K=0
12    K=K+1
      V=PHIMIN+K*0.2
      L=0
      DO 20 J=2,22
      DO 15 I=3,82
      IF((PHI(I-1,J)-V)*(PHI(I,J)-V).GE.0.0) GO TO 15
      L=L+1
      X(L)=XX(I-1)+(XX(I)-XX(I-1))*(V-PHI(I-1,J))/(PHI(I,J)-
     1 PHI(I-1,J))
      Y(L)=YY(J)
15    CONTINUE
20    CONTINUE
      DO 30 I=2,82
      DO 25 J=3,22
      IF((PHI(I,J-1)-V)*(PHI(I,J)-V).GE.0.0) GO TO 25
      L=L+1
      X(L)=XX(I)
      Y(L)=YY(J-1)+(YY(J)-YY(J-1))*(V-PHI(I,J-1))/(PHI(I,J)-
     1 PHI(I,J-1))
25    CONTINUE
30    CONTINUE
      LMAX=L
      XMIN=XX(82)
      DO 35 L=1,LMAX
      IF(X(L).GT.XMIN) GO TO 35
      XMIN=X(L)
      LMIN=L
35    CONTINUE
      XS=X(1)
      YS=Y(1)
      X(1)=X(LMIN)
      Y(1)=Y(LMIN)
      X(LMIN)=XS
      Y(LMIN)=YS
      CALL MOVEA(SNGL(X(1)),SNGL(Y(1)))
      LL=1
```

```
40      LL=LL+1
        DSQ=1.0D38
        DO 45 L=LL,LMAX
        DSQL=(X(L)-X(LL-1))**2+(Y(L)-Y(LL-1))**2
        IF(DSQL.GT.DSQ) GO TO 45
        DSQ=DSQL
        LDMIN=L
45      CONTINUE
        XS=X(LL)
        YS=Y(LL)
        X(LL)=X(LDMIN)
        Y(LL)=Y(LDMIN)
        X(LDMIN)=XS
        Y(LDMIN)=YS
        CALL DRAWA(SNGL(X(LL)),SNGL(Y(LL)))
        IF(LL.EQ.LMAX) GO TO 50
        GO TO 40
50      IF(V+0.2.LE.PHIMAX) GO TO 12
        CALL FINITT(800,500)
        RETURN
        END
```