02142-503-M

2142-503-M = RL-2043

THE UNIVERSITY OF MICHIGAN

COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING
Radiation Laboratory

ANALYSES AND DIGITAL PROGRAMS FOR COMPUTING THE MONOPULSE POINTING ERRORS ASSOCIATED WITH THREE-DIMENSIONAL CONICAL AND OGIVAL RADOMES WITH OR WITHOUT A SURROUNDING (WEAK) PLASMA

By

T.B.A. Senior and V.V. Liepa

February 1970

Prepared for

Harry Diamond Laboratories
Microwave Branch, 250
Attn: Dr. Irvin Pollin, AMXDO-RDB
Washington, DC 20438

Ann Arbor, Michigan, 48108
201 East Catherine Street

TABLE OF CONTENTS

I. INTRODUCTION

In a preceding memorandum (No. 02142-502-M) entitled "The Monopulse Pointing Error Associated With a Two-Dimensional Conical or Ogival Radome With or Without a Surrounding (Weak) Plasma", hereinafter referred to as S-L, we described the motivation and background of the present study, and discussed the procedures which are appropriate to the determination of the pointing error in the two-dimensional case. In addition the analytical steps were described, and a summary and print-out of the computer program were presented.

It was felt desirable to study the two-dimensional case in some detail before attempting the more general and complex problem posed by a three-dimensional geometry. This was undoubtedly a wise decision, and the knowledge gained from a consideration of a problem whose geometry is sufficiently simple to permit a direct interpretation of the results has enabled us to arrive at a three-dimensional problem which is more efficient than might otherwise have been the case. There is, moreover a certain degree of simplicity between the analyses in the two cases, and a considerable similarity in the concepts.

The present memorandum is devoted entirely to the three-dimensional problem in which a plane wave of arbitrary polarization is incident at an arbitrary direction on a rotationally-symmetric radome whose surfaces are described by either linear or quadratic equations of stated form, corresponding to conical and ogival radomes respectively. Wherever appropriate, reference is made to the previous memorandum, S-L, for details which are common to the analyses for the two- and three-dimensional problems. There are, however, numerous features which are unique to the latter case, and these are described in Sections II through V.

A discussion of the numerical procedures, including the choice of tolerances, ray sampling areas, etc, is given in Section VI, and the Appendix contains a listing of the computer program. It should be noted that there are actually two separate programs, one pertaining to the conical radome and the other to the ogival one. This separation was found convenient because of the need for numerical iteration to locate the ray intercepts when the radome is ogival.

In the two-dimensional case it was possible to run the program for a wide variety of incidence angles and for a number of radome/plasma/polarization combinations , and to examine the nature of the resulting pointing errors. Because of the much greater length (and hence, running time) of the three-dimenional program, such detailed checking has not been possible in the present case. Indeed, the funds that were available have allowed only those tests that were required to de-bug the program and to give confidence that it will work in any given case, together with a few runs of the complete program for single incidence and polarization angles. The conditions under which these particular runs were carried out, and the results that were obtained, are described in Section VI.

It is a pleasure to acknowledge the assistance of Dr. K-H Hsu with the programming.

II RAY TRACING PRESCRIPTIONS

The radomes considered here are assumed to be of single layer construction and formed from a material which can be treated as a homogeneous, isotropic and lossless dielectric whose permeability is the same as that of free space. The electromagnetic properties can therefore be represented by a real refractive index $\tilde{n}$, a typical value of which is 2.5 appropriate to fiberglass in the GHz range of frequencies.

Two particular radome configurations are treated, namely, ogival and conical. In either instance it is assumed that outer and inner surfaces are both ogival or both conical, and formed by the revolution of a straight line about some axis, or by the revolution of an arc of a circle about a chord. Although the choice does permit the radome thickness to be non-uniform, it will be appreciated that the type of thickness variation that is encompassed is distinctly limited.

The radome surface themselves are described in terms of a rectangular Cartesian coordinate system (x, y, z) whose z axis is the axis of the radome. For computational purposes it is convenient to choose the origin of coordinates at a small but non-zero distance ℓ to the left of the radome tip (see Fig. 1). In the ogival case, the outer surface can then be defined as

$$\rho_{outer} = \sqrt{A^2+B^2-(z-C)^2} - A \qquad (1)$$

valid for $z \geq C-B$, where $\rho = \sqrt{x^2+y^2}$; A, B and C are positive real numerical constants in terms of which the maximum diameter of the radome is

$$2\left(\sqrt{A^2+B^2} - A\right)$$

occurring at $x = C$, the (longitudinal) radius of curvature is

$$R = \sqrt{A^2 + B^2} \quad ,$$

and ℓ = C-B. The overall length of the radome is 2B, but in practice interest is confined to that portion of the interior extending to at most the position of the maximum diameter, i.e. to $z \leq C$. The inner radome surface is defined in a similar manner.

For the conical radome the definitions of the radome surfaces are more straightforward, and for the outer surface we have

$$\rho = a(z-\ell) \tag{2}$$

valid for $z \geq \ell$, with a similar definition for the inner surface.

The electromagnetic field incident on the radome is assumed to be a uniform plane wave of arbitrary polarization propagating in an arbitrary direction. It is convenient to write the electric vector as

$$\underline{E} = (\ell_1, m_1, n_1) e^{ik(\ell x + my + nz)} \quad , \tag{3}$$

where (ℓ, m, n) and (ℓ_1, m_1, n_1) are two sets of direction cosines* measured with respect to (x, y, z), and such that

$$(\ell, m, n) \cdot (\ell_1, m_1, n_1) = \ell\ell_1 + mm_1 + nn_1 = 0 \tag{4}$$

to make $\underline{E}$ perpendicular to the direction of propagation, as required. We note that the magnetic vector corresponding to $\underline{E}$ is

$$\underline{H} = Y_0 (\ell_2, m_2, n_2) e^{ik(\ell x + my + nz)} \tag{5}$$

where Y_0 is the intrinsic admittance of free space. Since

$$(\ell_2, m_2, n_2) = (\ell, m, n) \wedge (\ell_1, m_1, n_1) \ , \tag{5}$$

*Of course, $\ell^2 + m^2 + n^2 = 1 = \ell_1^2 + m_1^2 + n_1^2$.

a knowledge of 5 of the 6 direction cosines $\ell, m, n, \ell_1, m_1, n_1$ serves to specify the incident field in its entirety. In practice, however, we shall regard all 6 as input parameters chosen in accordance with Eq. (4).

We now turn to a consideration of ray tracing. A general discussion of the philosophy and limitations of ray tracing as applied to this type of problem was given in S-L, and will not be enlarged upon here. Instead, we shall confine ourselves merely to a description of the 'mechanics' of carrying through this procedure for a three-dimensional geometry of the sort shown in Fig. 1.

The incident field (3) was taken to have zero phase wavefront passing through the origin of the coordinate system. This is the wavefront (or phase plane) which will be sampled. Choose $x^{(0)}$, $y^{(0)}$ such that the point $x = x^{(0)}$, $y=y^{(0)}$, $z=z^{(0)}$ lies in this plane. Then

$$z^{(0)} = -\frac{1}{n}\left(\ell x^{(0)} + m y^{(0)}\right), \tag{7}$$

The ray through this point is perpendicular to the phase front and has equations

$$x=x^{(0)}+\frac{\ell}{n}\left(z-z^{(0)}\right), \quad y=y^{(0)}+\frac{m}{n}\left(z-z^{(0)}\right) . \tag{8}$$

This will intercept (say) the lower surface of the outer radome. The z coordinate, $z=z^{(1)}$ of the point of interception is given by the equation

$$\rho = \left[\left\{x^{(0)}+\frac{\ell}{n}\left(z-z^{(0)}\right)\right\}^2 + \left\{y^{(0)}+\frac{m}{n}\left(z-z^{(0)}\right)\right\}^2\right]^{1/2} = \rho(z) \tag{9}$$

where $\rho=\rho(z)$ is the equation of the radome surface. For a conical radome (see Eq. (2)), the determination of $z^{(1)}$ from (9) requires only the solution of a quadratic equation, but for an ogival radome (see Eq. (1)) it is necessary to resort to iteration. Having found $z^{(1)}$, the remaining coordinates $x^{(1)}$ and $y^{(1)}$ of the intercept point follow from (8). The distance

$$d_{o1} = \left\{ (x^{(1)}-x^{(0)})^2+(y^{(1)}-y^{(0)})^2+(z^{(1)}-z^{(0)})^2 \right\}^{1/2} \tag{10}$$

is then computed and stored.

Compute the direction cosines

$$(\ell_n, m_n, n_n) = \frac{1}{\sqrt{1+(\partial\rho/\partial z)^2}} \left(\frac{x}{\rho}, \frac{y}{\rho}, -\frac{\partial\rho}{\partial z}\right) \tag{11}$$

of the outward normal to the radome at the point of intercept. This vector and the ray direction establish the plane of incidence at the point, and because the manner of transmission and reflection at the interface depends on the inclination of the electric vector to the plane, we now perform the first of an almost endless series of rotations of coordinates.

Choose (x_1, y_1, z_1) such that

$$(x_1, y_1, z_1) = \begin{pmatrix} \ell & m & n \\ \ell_1 & m_1 & n_1 \\ \ell_2 & m_2 & n_2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} . \tag{12}$$

The incident field then takes the form

$$\underline{E} = \hat{y}_1 e^{ikx_1}, \quad \underline{H} = Y_0 \hat{z}_1 e^{ikx_1}, \tag{13}$$

and the ray that strikes the radome is traveling in the positive x_1 direction. Since $\hat{x}_1 = \ell\hat{x}+m\hat{y}+n\hat{z}$ and the normal direction $\ell_n\hat{x}+m_n\hat{y}+n_n\hat{z}$ specify the plane of incidence, a rotation of these new coordinates about the x_1 axis can produce a coordinate system x_2, y_2, z_2 with $x_2=x_1$ and $\hat{z}_2$ normal to this plane. But this normal has direction cosines

$$\pm(\ell_n, m_n, n_n) \wedge (\ell, m, n) = \pm(m_n n - n_n m,\ n_n \ell - \ell_n n,\ \ell_n m - m_n \ell),$$

and hence we can take

$$(x_2, y_2, z_2) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & A & B \\ 0 & -B & A \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \tag{14}$$

with

$$A = \gamma(\ell_n \ell_1 + m_n m_1 + n_n n_1) \left\{ (\ell_n \ell_1 + m_n m_1 + n_n n_1)^2 + (\ell_n \ell_2 + m_n m_2 + n_n n_2)^2 \right\}^{-1/2}$$

$$B = \gamma(\ell_n \ell_2 + m_n m_2 + n_n n_2) \left\{ (\ell_n \ell_1 + m_n m_1 + n_n n_1)^2 + (\ell_n \ell_2 + m_n m_2 + n_n n_2)^2 \right\}^{-1/2} \tag{15}$$

and $\gamma = \pm 1$. Although either sign for γ would be acceptable, it is necessary to choose one and to know the implications of that choice in order that we may later deduce angles correctly from a knowledge of their sines or cosines. To remove this arbitrariness, we impose the requirement that

$$\hat{y}_2 \cdot (\ell_n, m_n, n_n) \geq 0 \ , \tag{16}$$

i.e. that $\hat{y}_2$ makes an acute angle with the outward normal. It is a trivial matter to show that (16) implies

$$\gamma = 1 \ . \tag{17}$$

These new coordinates are related to the original coordinates x, y, z via the matrix equation

$$\underline{x}_2 = \underline{\underline{M}}_2 \underline{\underline{M}}_1 \underline{x} \tag{18}$$

where

$$\underline{\underline{M}}_1 = \begin{pmatrix} \ell & m & n \\ \ell_1 & m_1 & n_1 \\ \ell_2 & m_2 & n_2 \end{pmatrix} , \qquad \underline{\underline{M}}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & A & B \\ 0 & -B & A \end{pmatrix}$$

and A and B are given by Eqs. (15) and (17), and the incident electric vector now takes the form

$$\underline{E} = (A\hat{y}_2 - B\hat{z}_2)e^{ikx_2} \quad . \tag{19}$$

Recalling that $\hat{y}_2$ lies in the plane of incidence, A and -B therefore represent the electric field components in and perpendicular to the plane of incidence, and we can invoke the Fresnel reflection and transmission formulae to determine the strengths of the reflected and transmitted fields.

Consider the elementary problem illustrated in Fig. 2. Medium 1 is free space and medium 2 is a pure dielectric with refractive index $\tilde{n}$ relative to free space. A plane wave is incident at an angle α to the normal to the interface, and from Snell's law, the direction which the transmitted wave makes with the normal is then β, where

$$\sin\beta = \frac{\sin\alpha}{\tilde{n}} \quad . \tag{20}$$

If the incident field is polarized with its electric vector perpendicular to the plane of incidence, the reflection and transmission coefficients are respectively

$$R_{12}^{\perp} = -\frac{1-\Gamma'}{1+\Gamma'} \quad , \quad T_{12}^{\perp} = \frac{2\Gamma'}{1+\Gamma'} \tag{21}$$

(see Section II of S-L) with

$$\Gamma' = \frac{\cos\alpha}{\tilde{n}\cos\beta} \quad ; \tag{22}$$

whereas for the electric vector in the plane of incidence

$$R_{12}^{\parallel} = \frac{1-\Gamma}{1+\Gamma} \;, \quad T_{12}^{\parallel} = \frac{2}{1+\Gamma} \;, \tag{23}$$

with

$$\Gamma = \frac{\tilde{n}\cos\alpha}{\cos\beta} = \tilde{n}^2\,\Gamma' \quad . \tag{24}$$

The above coefficients are, of course, amplitude factors only, relating to the electric field, and in using these formulae, particularly those for the parallel case, it is essential that the orientation of the electric vectors in the reflected and transmitted waves be taken as shown in Fig. 2. We also note in passing that if the plane wave were to have originated in medium 2 and to have struck the interface with medium 1 (free space) at an angle β to the normal, the reflection and transmission coefficients corresponding to the above would be

$$R_{21}^{\perp} = \frac{1-\Gamma'}{1+\Gamma'}, \quad T_{21}^{\perp} = \frac{2}{1+\Gamma'}, \tag{25}$$

$$R_{21}^{\parallel} = -\frac{1-\Gamma}{1+\Gamma}, \quad T_{21}^{\parallel} = \frac{2\Gamma}{1+\Gamma}, \tag{26}$$

where Γ' and Γ are as given in (22) and (24) respectively.

Reverting now to the ray tracing procedure through the radome, we observe that the angle α appearing in the above formulae is that which the ray direction makes with the inward normal at the point of intercept, and hence

$$\begin{aligned} \cos\alpha &= -\hat{x}_2 \cdot (\ell_n \hat{x} + m_n \hat{y} + n_n \hat{z}) \\ &= -(\ell \ell_n + m m_n + n n_n) \end{aligned} \tag{27}$$

with

$$\begin{aligned} \sin\alpha &= \hat{y}_2 \cdot (\ell_n \hat{x} + m_n \hat{y} + n_n \hat{z}) \\ &= \frac{1}{\gamma} . \end{aligned} \tag{28}$$

Equations (27) and (28) specify α uniquely, and using (20), the angle β can then be found.

The reflected and transmitted rays both lie in the plane of incidence, and though a ray which is reflected at its initial impact with the radome is of no real concern to us, it is convenient to describe the tracing procedure for such a ray as well, since the analysis is almost directly applicable to reflection and transmission at any surface of the radome.

To describe the reflected ray, we introduce new coordinates (x_3, y_3, z_3) obtained from (x_2, y_2, z_2) by rotation about the z_2 axis and such that $\hat{x}_3$ is in the direction of the reflected ray (see Fig. 3a). Clearly

$$\underline{x}_3 = \underline{\underline{M}}_3 \underline{x}_2 \tag{29}$$

with

$$\underline{\underline{M}}_3 = \begin{pmatrix} -\cos 2\alpha & \sin 2\alpha & 0 \\ -\sin 2\alpha & -\cos 2\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{30}$$

implying

$$\underline{x}_3 = \underline{\underline{M}}_3 \underline{\underline{M}}_2 \underline{\underline{M}}_1 \underline{x} \ , \tag{31}$$

and hence from Eqs. (19), (21) and (23), the electric vector associated with the reflected ray is

$$\underline{E} = (AR_{12}^{\parallel} \hat{y}_3 - BR_{12}^{\perp} \hat{z}_3)\, e^{ikx_3} \quad . \tag{32}$$

The final step is to rotate the coordinates yet again so as to cast this result into a form as compact as that possessed by the electric vector of the incident ray in the (x_1, y_1, z_1) coordinates . This can be achieved by a rotation about the x_3 axis to produce coordinates (x_4, y_4, z_4) :

$$\underline{x}_4 = \underline{\underline{M}}_4 \underline{x}_3 \tag{33}$$

where

$$\underline{\underline{M}}_4 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{AR_{12}^{\parallel}}{\gamma_4} & -\frac{BR_{12}^{\perp}}{\gamma_4} \\ 0 & \frac{BR_{12}^{\perp}}{\gamma_4} & \frac{AR_{12}^{\parallel}}{\gamma_4} \end{pmatrix} \tag{34}$$

with

$$\gamma_4 = \left\{ (AR_{12}^{\parallel})^2 + (BR_{12}^{\perp})^2 \right\}^{1/2} \quad (\gamma_4 \geq 0) \ , \tag{35}$$

implying

$$\underline{x}_4 = \underline{\underline{M}}_4 \, \underline{\underline{M}}_3 \, \underline{\underline{M}}_2 \, \underline{\underline{M}}_1 \, \underline{x} \;, \tag{36}$$

in which case the electric vector becomes

$$\underline{E} = \gamma_4 \, \hat{y}_4 \, e^{ikx_4} \;. \tag{37}$$

The procedure for the transmitted ray is very similar. By rotating the coordinate system about the z_2 axis, a new set of coordinates $(x_3', \; y_3', \; z_3')$ can be established having $\hat{x}_3'$ in the direction of the transmitted ray (see Fig. 3b). We have

$$\underline{x}'_3 = \underline{\underline{M}}'_3 \, \underline{x}_2 \tag{38}$$

with

$$\underline{\underline{M}}'_3 = \begin{pmatrix} \cos(\alpha-\beta) & -\sin(\alpha-\beta) & 0 \\ \sin(\alpha-\beta) & \cos(\alpha-\beta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{39}$$

implying

$$\underline{x}'_4 = \underline{\underline{M}}'_3 \, \underline{\underline{M}}_2 \, \underline{\underline{M}}_1 \underline{x} \;. \tag{40}$$

The electric vector associated with the transmitted ray is then

$$\underline{E} = (AT_{12}^{\parallel} \, \hat{y}_3' - BT_{12}^{\perp} \, \hat{z}_3')e^{iNkx_3'} \tag{41}$$

and if we now rotate the coordinates again to produce (x_4', y_4', z_4') where

$$\underline{x}'_4 = \underline{\underline{M}}'_4 \, \underline{x}'_3 \tag{42}$$

with

$$\underline{\underline{M}}_4' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \dfrac{AT_{12}^{\parallel}}{\gamma_4'} & -\dfrac{BT_{12}^{\perp}}{\gamma_4'} \\ 0 & \dfrac{BT_{12}^{\perp}}{\gamma_4'} & \dfrac{AT_{12}^{\parallel}}{\gamma_4'} \end{pmatrix} \tag{43}$$

and

$$\gamma_4' = \left\{ (AT_{12}^{\parallel})^2 + (BT_{12}^{\perp})^2 \right\}^{1/2} \quad , \ (\gamma_4' \geq 0) , \tag{44}$$

implying

$$\underline{x}_4' = \underline{\underline{M}}_4' \underline{\underline{M}}_3' \underline{\underline{M}}_2' \underline{\underline{M}}_1' \underline{x} , \tag{45}$$

the electric vector in the transmitted ray becomes

$$\underline{E} = \gamma_4' \hat{y}_4' e^{i\tilde{n}kx_4'} \quad . \tag{46}$$

This restores the ray to the form it had prior to its intercept with the radome.

Although the above discussion has been phrased in terms of the initial impact of a ray with the outer surface of the radome, the procedure is that which must be applied at any interface. To illustrate this fact, consider the ray (45) transmitted through the outer surface. Under almost all circumstances, this ray will sooner or later strike the inner surface, and knowing the ray direction, $\hat{x}_4'$, in terms of the coordinates (x, y, z), together with a point on the ray (namely, $x=x^{(1)}$, $y=y^{(1)}$, $z=z^{(1)}$), the point of interception $x=x^{(2)}$, $y=y^{(2)}$, $z=z^{(2)}$ with the inner surface can be found in the same way as before. The distance

$$d_{12} = \left\{ (x^{(2)}-x^{(1)})^2 + (y^{(2)}-y^{(1)})^2 + (z^{(2)}-z^{(1)})^2 \right\}^{1/2} \tag{47}$$

is then computed, and $n\, d_{12}$ added to d_{01} to give the accumulated optical distance from the reference phase plane. From now on the analysis carries through in the manner already described apart from such trivial changes as are caused by the fact that the ray is incident in the denser medium. Specifically, in Eqs. (30) and (39) α must be replaced by β, and vice versa, and the reflection and transmission coefficients given by (21) and (23) must be replaced by those in (25) and (26). These changes apart, the same analysis is applicable.

The above procedure enables us to trace a ray from the outside of the radome through the layer and into the interior space, keeping track of the direction of the ray and of the electric vector, as well as of the amplitude and phase (or, in effect, the optical distance) associated with the ray. At the inner surface of the radome a reflected ray is, of course, produced, and this is directed back to the outer surface where transmission and reflection again occur. Only the reflected wave is of interest, and we remark that the analysis appropriate to this is identical to that corresponding to reflection at the inner radome surface providing that the normal direction now used is the inward-directed one. In the analysis, therefore, the direction cosines ℓ_n, m_n, n_n of the outward normal must be reversed in sign. This reflected ray can now be traced back to the inner radome surface; and so on. And if, at some stage, a ray crosses the interior of the radome to strike the upper surface, or strikes the upper surface initially rather than the lower, the <u>nature</u> of the analysis is in no way affected.

By repeated applications of the above procedures we can trace the progress of any one ray from its starting point on the (reference) phase plane, and can follow all of those rays which are spawned by reflection at the radome surfaces. There are an infinity of such rays, but inasmuch as the amplitude of any one is reduced at each reflection, it is in practice sufficient to establish an amplitude level (or tolerance) and to ignore any ray once its amplitude has fallen below this level. We can likewise ignore any ray that passes through the monopulse plane in the region outside the monopulse plate, but even so we may still be faced with a relatively large number of significant ray contributions all stemming from the single ray incident on the radome. These rays provide our first (partial) sampling of the field distribution over the monopulse plate. To complete the determination of this field we must now sample the incident phase front at other points, and continue this process until we achieve an adequate specification of the field to which the monopulse is subject. This requires us to sample the incident phase front at a sufficiently dense number of points over a region which at least includes the origin of all rays capable of yielding any contribution to the monopulse excitation exceeding the tolerance level. It is convenient to carry out this sampling at a rectangular lattice of points whose separation (or stepping distance) Δ will be taken equal to that which was found desirable in the two-dimensional problem (see S-L, pp. 42-43), and the sampling procedure is described in the next section.

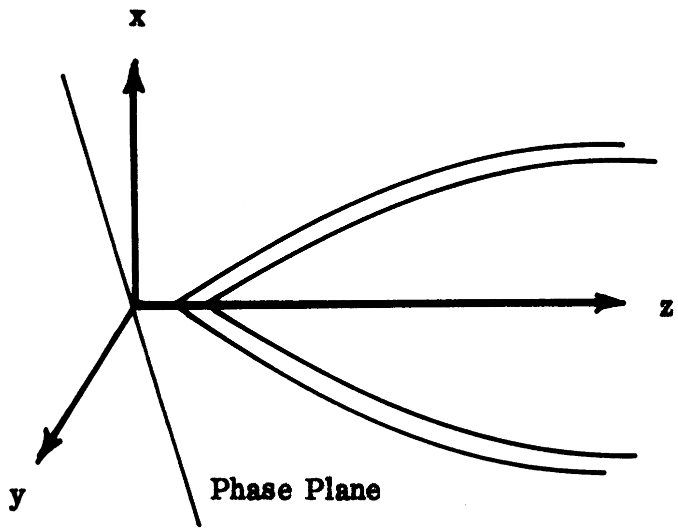

Fig. 1: Coordinate System.

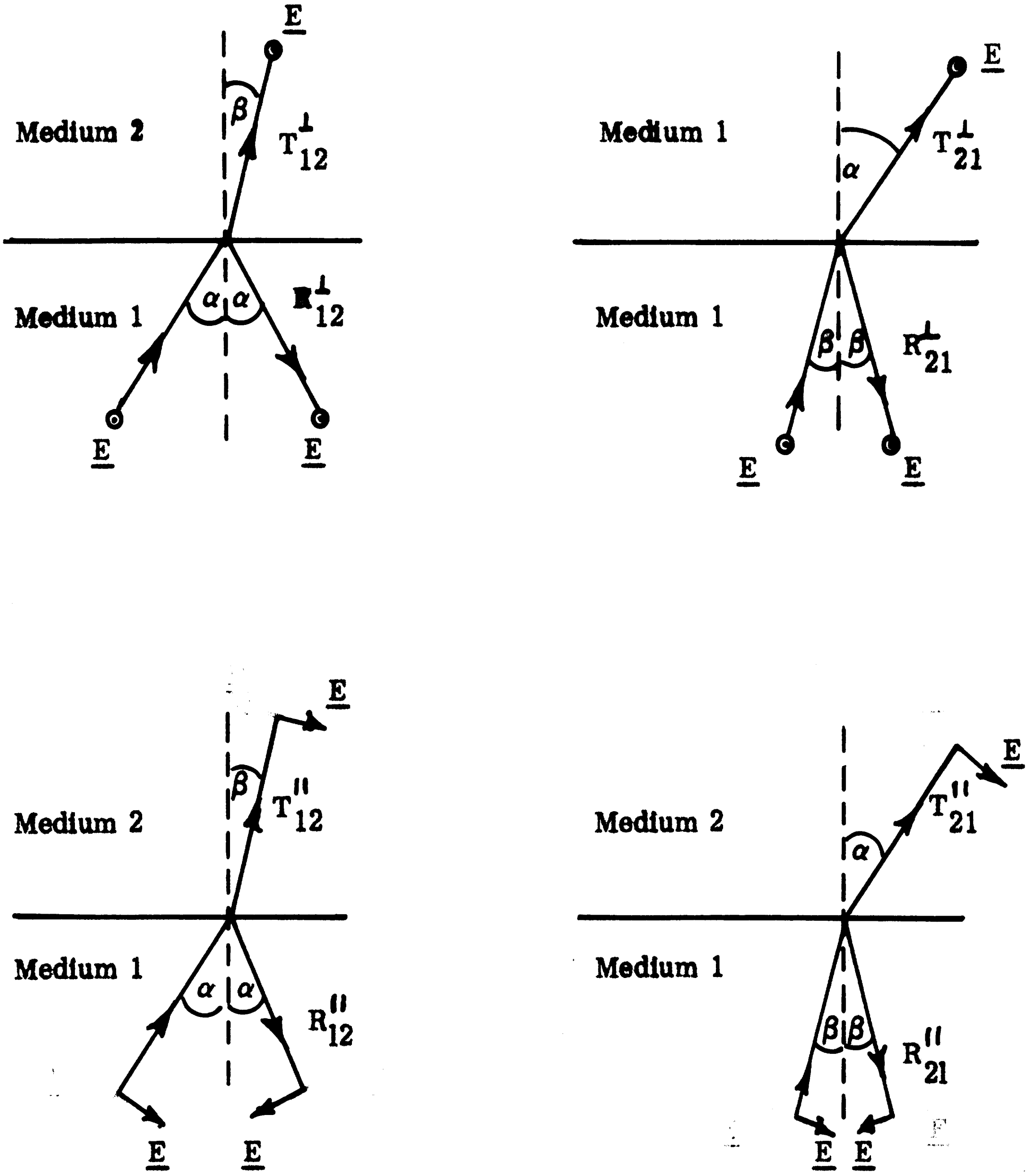

Fig. 2: Geometry for Reflection and Transmission at an Interface.

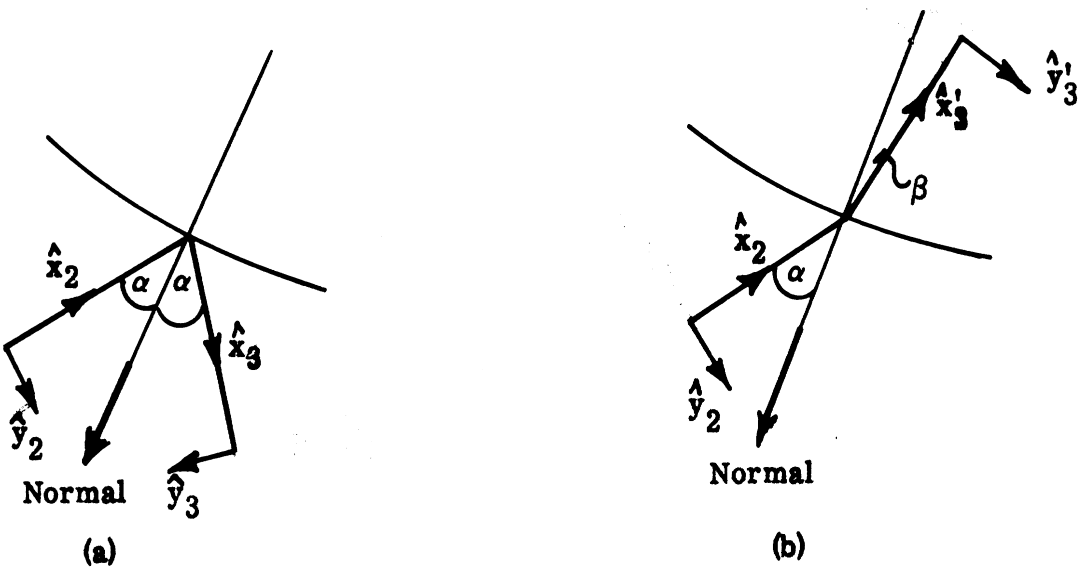

Fig. 3: Coordinate Transformations for Reflection and Transmission.

III INCIDENT FIELD SAMPLING

The monopulse plate pivots about a point on the (z) axis of symmetry of the radome (see Figs. 1 and 4). Let z_m be the coordinate of the pivot point, and let z_n be the corresponding coordinate of the radome tip, i.e. the intercept of the outer radome surface with with z axis. The monpulse plate itself is often somewhat irregular in shape, but for the sake of the following analysis it is convenient to speak in terms of a disc of radius a centered at the point $z=z_m$. It is natural to choose 2a to equal the maximum diameter (or dimension) of the monpulse plate, but under most circumstances it would appear adequate to choose a somewhat smaller value such that the disc just encompasses the monopulse slots. A reduction in the value of a is, of course, reflected in a decrease in the number of sampling points and, hence, in the running time of the program.

The equation of the incident phase front is

$$\ell x + my + nz = 0 . \tag{48}$$

We are required to compute the field distribution over a disc of radius a lying in the center of the monopulse plane when the latter is aligned parallel to the above phase front. The projection of this disc on to the wavefront is therefore itself a disc of radius a . Since any line perpendicular to the wavefront and passing through the point (x', y', z') has equations

$$\frac{x-x'}{\ell} = \frac{y-y'}{m} = \frac{z-z'}{n} , \tag{49}$$

the projection of the center of the disc has coordinates (x_o, y_o, z_o) where

$$x_o = -\ell n z_m, \quad y_o = -m n z_m, \quad z_o = (1-n^2) z_m , \tag{50}$$

and the projection of the cone tip is likewise (x_t, y_t, z_t) where

$$x_t = -\ell n z_n, \quad y_t = -m n z_n , \quad z_t = (1-n^2) z_n . \tag{51}$$

The equation of the line passing through (x_o, y_o, z_o) and (x_t, y_t, z_t) is therefore

$$\frac{x-x_o}{x-x_t} = \frac{y-y_o}{y-y_t} = \frac{z-z_o}{z-z_t} ,$$

and inserting the preceding expressions for x_o, x_t, etc, (51) reduces to

$$x = -\frac{\ell n}{1-n^2} z , \quad y = -\frac{m\,n}{1-n^2} z \quad . \tag{52}$$

This is the reference line for our sampling procedure. It is clearly just the projection of the z axis on to the incident wavefront, and therefore passes through the origin of coordinates as well as through the points (x_o, y_o, z_o) and (x_t, y_t, z_t).

It is now a trivial matter to locate the projection of the monopulse 'disc' on to the phase front. Since the projection is also a disc of radius a, moving down the reference line a distance a from the point (x_o, y_o, z_o) shows the point A (see Fig. 5) to have coordinates

$$\begin{aligned} x &= -\ell n \left(z_m + \frac{a}{\sqrt{1-n^2}}\right) , \\ y &= -m\,n\left(z_m + \frac{a}{\sqrt{1-n^2}}\right) , \\ z &= (1-n^2)\left(z_m + \frac{a}{\sqrt{1-n^2}}\right) . \end{aligned} \tag{53}$$

Likewise, B has coordinates

$$\begin{aligned} x &= -\ell n \left(z_m - \frac{a}{\sqrt{1-n^2}}\right) , \\ y &= -m\,n \left(z_m - \frac{a}{\sqrt{1-n^2}}\right) , \\ z &= (1-n^2)\left(z_m - \frac{a}{\sqrt{1-n^2}}\right) . \end{aligned} \tag{54}$$

But any line parallel to AB has equations (see (52))

$$x = -\frac{\ell n}{1-n^2} z + C_1 , \quad y = -\frac{m n}{1-n^2} z + C_2 \tag{55}$$

and lies in the phase front if

$$\ell C_1 + mC_2 = 0 . \tag{56}$$

By considering a sphere of radius a centered on the point (x_o, y_o, z_o) and requiring that the above line have only one intersection with the sphere, i.e. be tangent to it, we can specify C_1 appropriate to the points C and D and, hence, determine the z coordinates of these points. It then follows that the coordinates of C are

$$\begin{aligned} x &= -\ell n z_m + \frac{ma}{\sqrt{1-n^2}} , \\ y &= -m n z_m - \frac{\ell a}{\sqrt{1-n^2}} , \\ z &= (1-n^2)\,(z_m + \frac{a}{\sqrt{1-n^2}}\,) , \end{aligned} \tag{57}$$

whilst those of D are

$$\begin{aligned} x &= -\ell n z_m - \frac{m a}{\sqrt{1-n^2}} , \\ y &= -m n z_m + \frac{\ell a}{\sqrt{1-n^2}} , \\ z &= (1-n^2)(z_m + \frac{a}{\sqrt{1-n^2}}) . \end{aligned} \tag{58}$$

The reference or starting point for the sampling procedure that has been developed is the point A in Fig. 5, the coordinates of which are

$$
\begin{aligned}
x &= -\ell n \left(z_m + \frac{a - N\Delta}{\sqrt{1-n^2}} \right) + \frac{m M \Delta}{\sqrt{1-n^2}} \; , \\
y &= -m n \left(z_m + \frac{a - N\Delta}{\sqrt{1-n^2}} \right) - \frac{\ell M \Delta}{\sqrt{1-n^2}} \; , \qquad (59) \\
z &= (1-n^2) \left(z_m + \frac{a - N\Delta}{\sqrt{1-n^2}} \right)
\end{aligned}
$$

with $M = N = 0$. Sampling therefore starts at A and proceeds by stepping up the line AB a distance Δ each time. The coordinates of the successive points are given by Eqs. (59) with $N = 1, 2, 3 \ldots$ and $M = 0$, and sampling continues beyond the point B (for which* $N = \left[\frac{2a}{\Delta} \right]$) until the corresponding rays no longer intercept the radome. For angles of incidence outside the "backward cone", the last contributing sampling point will be just beyond T, corresponding to $N = \left[\frac{1}{\Delta} \left\{ a + (z_m - z_n) \sqrt{1-n^2} \right\} \right]$. Having completed this traverse, the next sampling point is on the line AF a distance Δ towards E. This is the starting point for a second traverse parallel to the first, and the coordinates of the successive points are now given by (59) with $N = 1, 2, 3, \ldots$ and $M = 1$. The third traverse is to the right of AB by a distance Δ (so that $M = -1$), and so on, running alternatively to the left ($M = 1, 2, 3, \ldots$) and to the right ($M = -1, -2, -3, \ldots$) of the central line AT. In each case sampling along a given traverse ceases when no further intercepts with the radome are obtained. The final two traverses are carried out with $M = \pm \left[\frac{a}{\Delta} \right]$, and proceed beyond the points C or D respectively (for which $N = \left[\frac{a}{\Delta} \right]$) until terminated in the same manner as before.

The region of the wavefront over which the sampling actually extends in any practical situation depends, amongst other things, on the incidence angles. It is, of course, obvious that if a ray emanating from a given point does not intercept

---

* $[x]$ denotes the integer which is just larger than (or equal to) x .

the radome surface, it cannot possibly produce any contribution to the field distribution over the monopulse plate, but if a ray does intercept, it does not follow automatically that it will excite the monopulse. Only by following through the laborious and time consuming process described in Section II can we discover whether it was necessary to have considered that sampling point in the first place. Nevertheless, any ray which intercepts the radome is a potential (and, in general, actual) source of monopulse excitation, and since the time spent in searching for, and computing, initial intercepts with the outer radome surface is a very small fraction of the total computation time, it is prudent (if not necessary) to carry out the wavefront sampling over a region which is large enough to encompass all rays which could conceivably intercept the radome.

Although the above procedure, in which the computer itself decides when sampling of the wavefront along a particular traverse shall cease, is aesthetically a satisfying one, the actual numerical program has been written on the basis of a pre-specified sampling area. The area selected is rectangular in shape and extends from EF up to a parallel line through B or through the tip projection T, whichever is the farthest. In consequence, M runs from 0 to $\pm\left[\frac{a}{\Delta}\right]$, and N from 0 to max. $\left(\left[\frac{2a}{\Delta}\right], \left[\frac{1}{\Delta}\left\{a + (z_m - z_n)\sqrt{1-n^2}\right\}\right]\right)$.

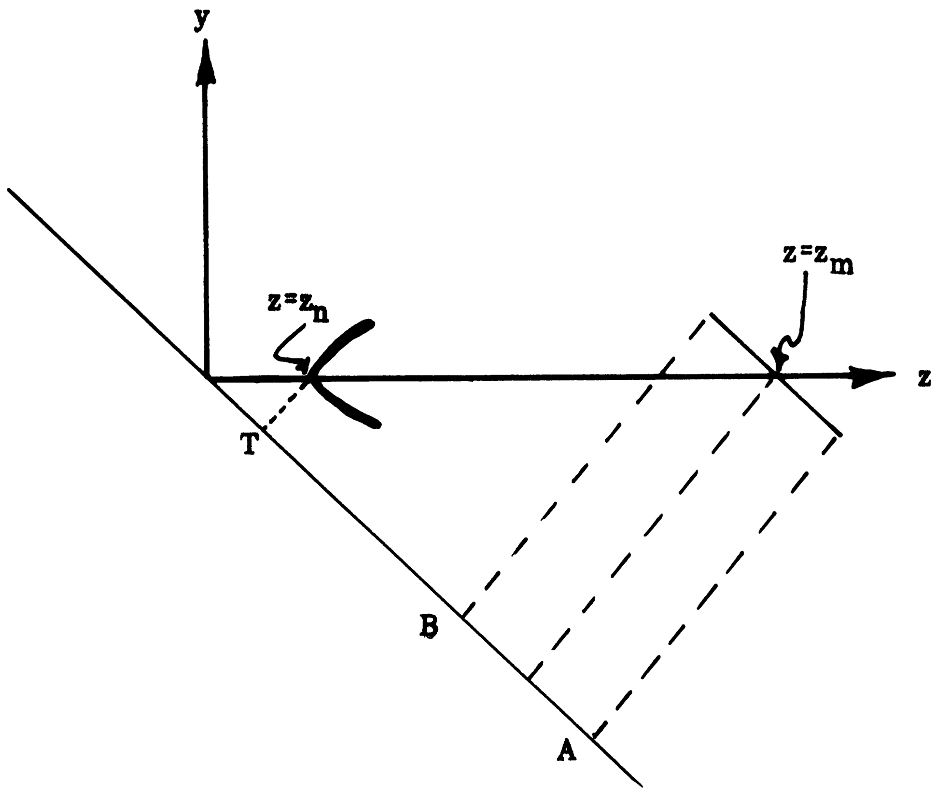

Fig. 4: Projection of the Equivalent Monopulse Disc.

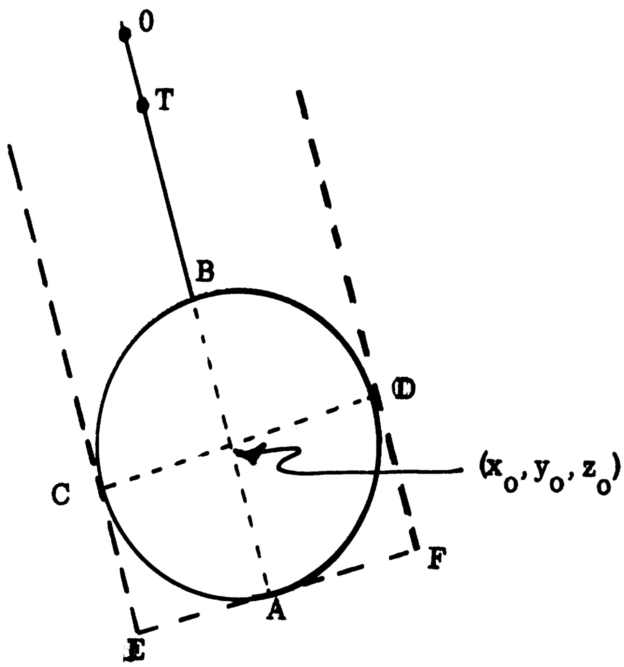

Fig. 5: Geometry for Sampling the Wavefront.

IV MONOPULSE PLATE AND POINTING ERROR

The rays which emanate from the incident wave front are all parallel and of constant amplitude (unity), and if the wave front is sampled in the manner just described, these rays provide for us a uniform sampling of the uniform field that would exist in the absence of the radome. When each ray strikes the radome, however, reflection and refraction occur, generating additional rays, and this takes place whenever the original ray or any of its subsidiaries intercepts a radome surface. It is therefore not surprising to find that the field within the radome is markedly non-uniform, and our knowledge of it is obtained from the information carried by those rays which penetrate into the interior. This information consists of the amplitude, optical distance of travel, direction of propagation and polarization of each ray, i.e.

$$A_s,\ d_s,\ (\ell, m, n)_s,\ (\ell_1, m_1, n_1)_s \ ,$$

respectively, at some position along the path of the s'th ray.

If the radome were not present, each ray within the radome would be only a continuation of one emanating from the phase plane, and its direction cosines $(\ell, m, n)_s$ would be the same as those of the normal to the wave front (ℓ, m, n), but because of reflection and refraction at the radome walls, the actual values of $(\ell, m, n)_s$ can be quite different from those of the incident ray. The amplitude A_s must be real, positive and less than unity. In contrast to the situation in the two-dimensional case where the amplitude could be positive or negative, any 180° change of phase (equivalent to a reversal in direction of the electric vector) is now incorporated into the polarization direction cosines $(\ell_1, m_1, n_1)_s$. The optical distance d_s is measured in inches, and can be converted to an electrical phase by multiplying by $k = 2\pi/\lambda$, where λ is the free space wavelength (also measured in inches).

The ensemble of all these rays constitutes a non-uniform sampling over any given plane of the non-uniform field existing within the radome, and it is this field which excites the monopulse. The monopulse plate configuration used in this program is that described* by Dr. I. Pollin and shown in Fig. 6. The overall dimensions of the pate are 9 x 9 inches, but since the plate has its corners chopped off, the maximum cross dimension is about 10.5 inches. The plate contains a total of 32 horizontal slots symmetrically located about x and y axes through its center. Each slot is 1.060 inches long and 0.140 inches wide, corresponding to 0.442 λ and 0.0583 λ, respectively, at 5 GHz. Apparently the slots have been chosen to be of resonant size at this frequency of operation.

The electric vector in the incident field is specified by the direction cosines (ℓ_1, m_1, n_1), and since these can be chosen at will providing $\underline{E}$ remains in the plane perpendicular to the propagation vector, there is no loss of generality in assuming that any motion of the monopulse plate is such as to keep the slots with their long dimension horizontal (appropriate to a field vertically polarized with respect to a horizontal plane, x = 0). Any displacement of the monopulse plate can then be treated as a rotation about a line through its center parallel to the x axis, followed by a rotation (or tilting) about a horizontal line. Such displacement is, of course, brought about by the voltages induced in the slots by the field incident upon them, and for the reasons discussed in S-L, these voltages are computed for a monopulse plate aligned parallel to the wavefront of the plane wave incident on the radome. The above mentioned angles of rotation are now specified by the direction cosines (ℓ, m, n), and with the notation shown in Fig. 7 , we have

$$\ell = \sin\alpha_x\,, \quad m = -\cos\alpha_x \sin\alpha_y\,, \qquad n = \cos\alpha_x \cos\alpha_y\,. \tag{60}$$

*Private communications, dated 13 November 1968 and 26 May 1969

Similarly, if the electric vector lies in the x'' y'' plane and makes an angle Ω with the x'' axis, the direction cosines of this vector with respect to the original coordinate system (x, y, z) are

$$(\cos\alpha_x \cos\Omega, \sin\alpha_x \sin\alpha_y \cos\Omega + \cos\alpha_y \sin\Omega, -\sin\alpha_x \cos\alpha_y \cos\Omega + \sin\alpha_y \sin\Omega) . \tag{61}$$

In the computer program the description of the monopulse plate is fed* in as a statement of the center point locations of each slot, and of their length and width. The symmetrical layout of the slots is employed to reduce the number of parameters to be specified, and only the information concerning the slots in the first quadrant, Q = 1, is explicitly introduced. For the other three quadrants (mirror images of the first), the slot locations are deduced by appropriate changes in the signs of the coordinates. The convention used in the program for designating the quadrants Q = 1, ... 4, and the slots S(1, 1), ..., S(4, 8) is shown in Fig. 6. The first number in the slot designation specifies the quadrant the slot is in, whilst the second specifies the slot number based on the first quadrant as reference. Note that the slot numbers in the second through fourth quadrants are those of the mirror-image slots in the first quadrant.

The voltages induced in the slots are produced by the field incident on the monopulse plate, and it is to be expected that this field will be of a highly non-uniform nature. The response function for a receiving antenna, e.g. a slot, is generally stated for the case of an incident plane wave, and it might now appear that to use such information in the present case would require us to match the sampled data provided by the ensemble of ray contributions to a set of plane waves impinging on the slots at the appropriate

---

*Although the present computer program has the monopulse plate data inserted as part of the main program, it would not be a difficult task to modify it to have the plate specifications read in as input data.

angles of incidence. Fortunately, this process is not necessary, and we can avoid the matching by assuming each ray to excite a slot directly. This is in line with the procedure developed in the two-dimensional case. Sections IV and VI of S-L contain a detailed discussion of the effects of amplitude taper, polar diagram, slot weighting, etc., and lead us to characterize the response of a slot by the following parameters.

(a) Slot size: The dimensions chosen for the slot are not the physical dimensions shown in Fig. 7. The reasons for this are two-fold. Firstly, examinations of the surface currents excited on a conducting sheet by a narrow resonant slot shows* that a strong field exists beyond the immediate confines of the slot, and the actual area 'excited' may be much larger than the area of the slot. Without a detailed knowledge of the actual slot characteristics, it is not possible to estimate the effective dimensions of a slot, but it was found in the two-dimensional study that the number and locations of slots in the monopulse plate affect only the smaller 'wiggles' in the pointing error curve, whereas the dominant features of the curve are almost exclusively determined by the radome and plasma (if present).

The second reason for choosing increased slot dimensions is to keep to a minimum the number of sampling rays necessary for an accurate pointing error computation. Experience has shown that of order ten (or more) primary rays are required to 'hit' each slot. By 'primary' rays we here mean those rays which reach the monopulse plate after a single passage through the radome layer, and without reflection at any surface. In the two-dimensional case it was found that for a spacing $\Delta = 0.2$ inches of the successive points at which the incident wave front was sampled, the number of rays striking a $\lambda/2$ slot was adequate for the pointing error determination, but it is most unlikely that this same sampling

---

*D.K. Reynolds, "Surface Current and Charge Measurements on Flat Metal Sheets," Harvard University Cruft Laboratory Tech. Report No. TR-53 (p. 68), 1 August 1948.

distance would produce more than 2 or 3 primary rays which strike a 1.060 by 0.140 inch slot. To decrease the sampling distance would be most undesirable from the point of view of the computation time, and thus the only alternative is to increase the area over which a ray striking the monopulse plate can contribute. It is fortunate that such an increase in the slot dimensions is a physically-justifiable procedure.

(b) Amplitude taper : The concept of an amplitude taper has been taken over directly from the two-dimensional study where it was found that its use removes the discontinuity in induced voltage that would otherwise occur when any one ray hits just inside, rather than just outside, a slot. Here again the concept is not only a physically reasonable one, but also enables us to decrease the sampling rate.

For a slot whose mid point is located at the origin of the x''', y''' coordinates (see right hand lower corner of Fig. 6), the following taper factor is assumed:

$$f^a = \cos\left(\frac{\pi y'''}{w}\right)\cos\left(\frac{\pi x'''}{\ell}\right) \;,\quad |y'''| < \frac{w}{2}\,,\ |x'''| < \frac{\ell}{2}$$
$$= 0 \qquad\qquad \text{otherwise} \tag{62}$$

where w and ℓ are the effective width and effective length, respectively, of the slot.

(c) Pattern factor : This takes into account the direction of arrival of a ray with respect to the normal to the slot. In general, the pattern factor is a rather complicated function involving the polarization as well as the direction of arrival, but it is convenient to treat these two effects separately.

In the two-dimensional analysis, a $\sin x/x$ pattern factor was employed appropriate to a uniformly excited aperture, but if the aperture dimensions are no more than $\lambda/2$ and the incidence is not too far from normal to the slot, a $\sin x/x$ factor is indistinguishable from a $\cos x$ one. We shall therefore take the pattern factor to be

$$f^p = \cos \theta_i \tag{63}$$

where θ_i is the angle between the direction of the incoming ray and the normal to the monopulse plate. We note that this can be written as

$$f^p = (\ell, m, n) \cdot (\ell', m', n') \ , \tag{64}$$

where (ℓ', m', n') are the direction cosines of the ray in the (x, y, z) coordinates, and also note that this choice of factor is not at all unreasonable. In the horizontal plane (see Fig. 6) a slot (being the dual of a dipole) will have a cosine pattern, and though the pattern would be a constant in a vertical plane were the conducting plate infinite in size, the fact that the actual plate dimensions are finite will produce some reduction in the pattern in this direction also.

(d) Polarization Sensitivity: It is obvious that the slots will be polarization sensitive. The induced signal will be a maximum when the incoming ray has its electric vector in the vertical direction, and for other polarizations the variation in signal strength will be proportional to

$$f^{pol} = \cos \phi \tag{65}$$

where ϕ is the angle which the electric vector makes with the plane containing the direction of propagation and the y'' axis (see Fig. 7) . Hence,

$$\cos \Omega = (\ell_1', m_1', n_1') \cdot \left\{ (\ell', m', n') \wedge y'' \right\}$$

where (ℓ_1', m_1', n_1') and (ℓ', m', n') are the direction cosines of the electric and propagation vectors, respectively, and since, from Fig. 7,

$$\hat{y}'' = \sqrt{m^2+n^2} \left\{ (m n' + n n')\hat{x} + \ell'(m\hat{y} + n\hat{z}) \right\} ,$$

we have

$$f^{pol} = \frac{1}{\sqrt{m^2+n^2}} \left\{ \ell_1'(mm' + nn') + \ell'(mm_1' + nn_1') \right\} . \tag{66}$$

The signal (in amplitude and phase) which a single ray induces in, say, the slot S(1, 3) is now obtained by multiplying the amplitude of that ray by all of the above factors, and by converting the optical distance of travel into an electrical distance. We thus have

$$V_s(1,3) = A_s f_s^a f_s^p f_s^{pol} e^{ikd_s} \tag{67}$$

and the total signal induced in this slot is

$$V(1,3) = \sum_s V_s(1,3) , \tag{68}$$

where the summation extends over all rays that strike the slot. The signals induced in the remaining 31 slots are computed in a similar manner, and are designated V(1, 1), . . . , V(4, 8) .

Were the slots excited by a uniform plane wave, i.e. were the radome and plasma absent, the signals induced in all 32 slots would be the same apart from such small discrepancies as are associated with the finite sampling rate. With the radome and, perhaps, plasma present, however, the field that strikes the monopulse plate is no longer uniform or planar, and the slot signals will now be different. The pointing error ascribed to the monopulse system will then depend on the way in which the various slot signals are combined and utilized.

It is obvious that with a total of 32 slots there are numerous ways in which the slots can be interconnected, with each ray equivalent to some form of addition (or subtraction) with varying (amplitude) weightings and phase shifts associated with the outputs from the individual slots. Although some information about the amplitude weightings has been supplied to us, an uncertainty about the electrical connections and the implied phase shifts has made it desirable to print out the computed values for the signals (see , for example, (68)) induced in each slot. For any given design of the electrical system, it is then possible to find the appropriate pointing error with only a trivial computation.

For illustrative purposes, however, a very elementary form of monopulse system has been considered in which the monopulse plate is treated as four antennas, each occupying a quadrant. The signals obtained from the quadrants are then

$$V_1 = \sum_{n=1}^{8} V(1,n) ,$$

$$\vdots$$

$$V_4 = \sum_{n=1}^{8} V(4, n) ,$$

and these can be combined to give

$$\begin{aligned} V_{12} &= V_1+V_2 = |V_{12}|\, e^{i\psi_{12}} \\ V_{23} &= V_2+V_3 = |V_{23}|\, e^{i\psi_{23}} \\ V_{34} &= V_3+V_4 = |V_{34}|\, e^{i\psi_{34}} \\ V_{41} &= V_4+V_1 = |V_{41}|\, e^{i\psi_{41}} \end{aligned} \tag{69}$$

from which a pointing error can be deduced in the same manner as in the two-dimensional program (see Section IV of S-L). The 'vertical' pointing error is then

$$\epsilon_V = \sin^{-1} \left\{ \frac{1}{4\pi \bar{\bar{x}}_V} (\psi_{34}-\psi_{12}) \right\} \tag{70}$$

and the 'horizontal' error is

$$\epsilon_H = \sin^{-1} \left\{ \frac{1}{4\pi \bar{\bar{x}}_H} (\psi_{41}-\psi_{34}) \right\} \tag{71}$$

where ϵ_V and ϵ_H are measured in radians. The quantities $\bar{x}_V$ and $\bar{x}_H$ are the mean, normalized (with respect to wavelength) slot displacements from the y'' and x'' axes respectively, and for the geometry shown in Fig. 6 with $\lambda = 2.4$, we have

$$\bar{x}_V = 0.8698, \qquad \bar{x}_H = 0.8810 . \tag{72}$$

Concerning the signs of ϵ_V and ϵ_H, we observe that if $\epsilon_V > 0$, then $\psi_{34} > \psi_{12}$. This in turn implies that the dominant rays have arrived in directions 'above' the plane $x''=0$, and must travel further to reach the lower slots compared with the upper, leading to an upward tilting of the monopulse plate in its effort to find a constant-phase plane. Hence, in the notation of Fig. 7, a positive value of ϵ_V represents a decrease in α_x, whereas a positive value of ϵ_H represents an increase in α_y . In other words, as viewed from the incident wave front, the monopulse points too high or too low according as $\epsilon_V > 0$ or < 0, respectively, and points to the right or left according as $\epsilon_H > 0$ or < 0, respectively.

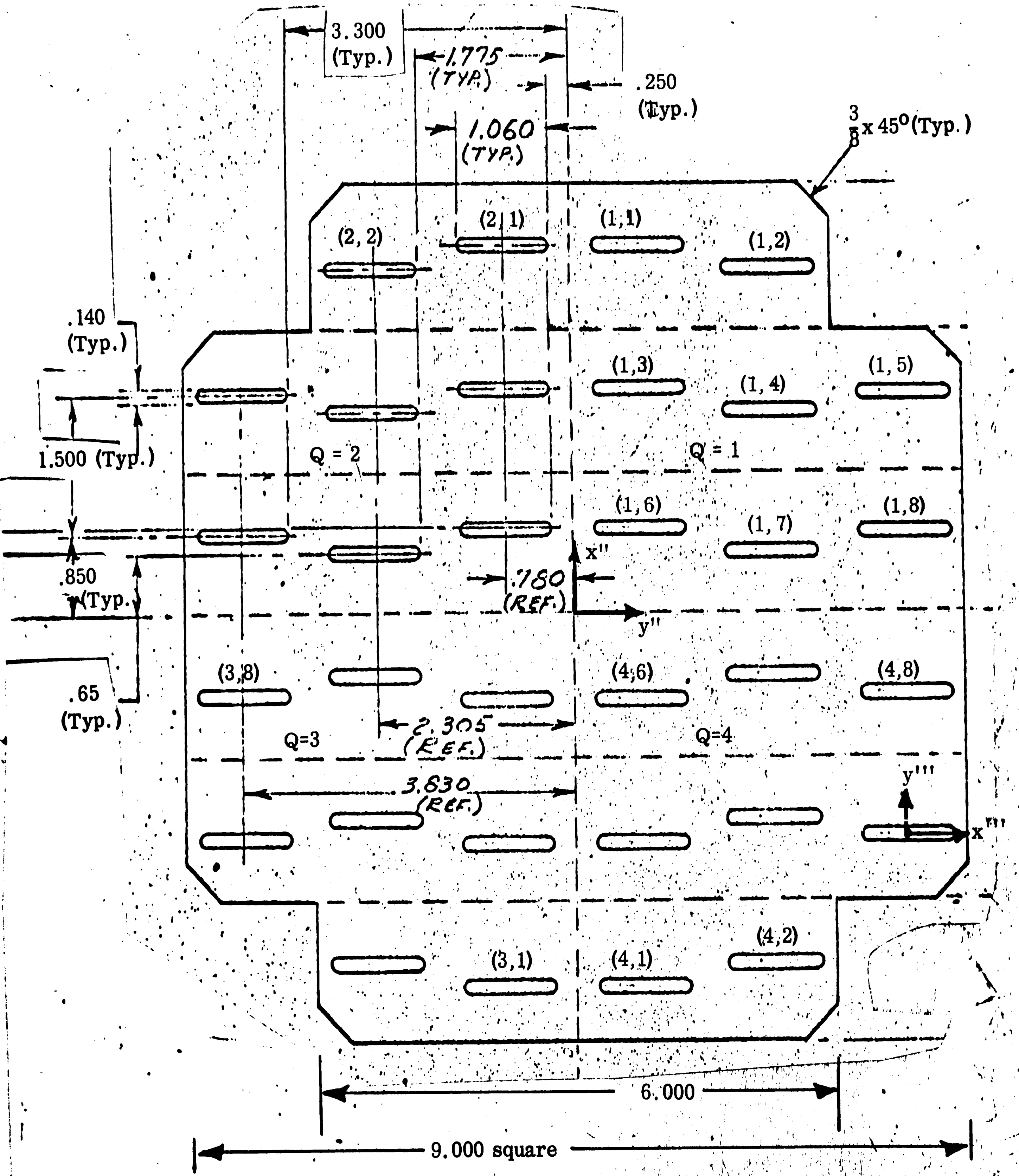

Fig. 6: Mechanical Specification for the Monopulse Plate.

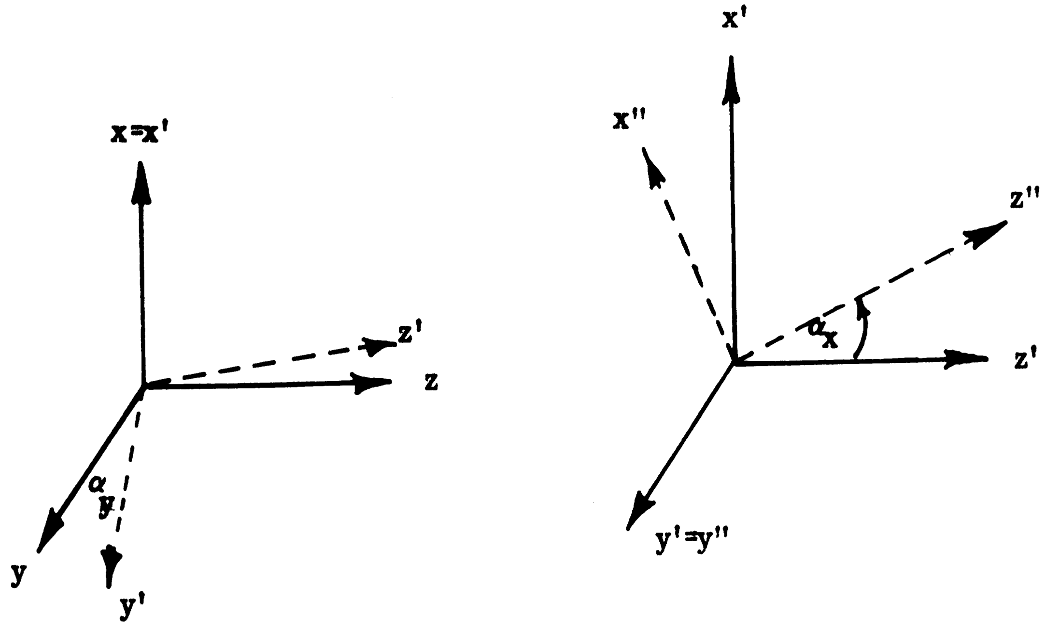

Fig. 7: Monopulse Plate Displacements.

V PLASMA MODIFICATION

Under some conditions of operation, the radome may be surrounded by a weak plasma, and even though the plasma can be treated as a lossless dielectric for purposes of analysis, it will still produce some effect on the field that penetrates into the interior of the radome. This will, in turn, lead to a change in the pointing error of the monopulse system, and the determination of this change is the main objective of the present study.

A detailed examination of the effect of the plasma on the rays passing through it was given in Section V of S-L. It was there noted that because the maximum departure of the equivalent refractive index from its free space value, unity, is less than 0.02 for the particular electron density profiles supplied to us, it is possible to avoid the time-consuming process of numerical integration that would be required to follow precisely the progress of each ray through the layer. The alternative procedure is to simulate the plasma by a form of jump discontinuity in which each ray is displaced a certain distance Δ_1 along the tangent to the radome in the plane of incidence, and the optical distance along the ray is increased by $\delta \ell$. In terms of the angle α between the incident ray direction and the (inward) normal

$$\Delta_1 \simeq 1.61 \times 10^{-12}\, t\, I \tan\alpha \sec^2\alpha \ , \tag{73}$$

$$\delta \ell \simeq - \Delta_1 \cos 2\alpha \operatorname{cosec}\alpha \ , \tag{74}$$

where t is the thickness of the layer and I is a normalized integrated electron density. For the plasma characteristics that have been specified (see Section V of S-L), both t and I are functions of the coordinate $z^{(1)}$ of the radome intercept, but are independent of $x^{(1)}$ and $y^{(1)}$.

In view of the detailed discussion given in S-L, we shall here confine ourselves to a description of the way in which we can implement the above procedure in the three-dimensional case.

Since the essence of the procedure is to rely as much as possible on the computational techniques developed for the bare radome, the analysis follows rather closely that developed in Section II of the present Memorandum. Starting with a point $x^{(0)}$, $y^{(0)}$, $z^{(0)}$ on the incident phase front, we first locate the point at which the ray intercepts the radome. Let this point have coordinates $x^{(1)}, y^{(1)}, z^{(1)}$. We now perform two notations of coordinates to produce the system x_2, y_2, z_2 (see Fig. 3), and if, for convenience, we take the origin of the new coordinates at the intercept point, we have (see Eq. 18)

$$\begin{aligned} x_2 &= \ell(x-x^{(1)}) + m(y-y^{(1)}) + n(z-z^{(1)}) , \\ y_2 &= (A\ell_1+B\ell_2)(x-x^{(1)}) + (Am_1+Bm_2)(y-y^{(1)}) + (An_1+Bn_2)(z-z^{(1)}) , \\ z_2 &= (-B\ell_1+A\ell_2)(x-x^{(1)}) + (-Bm_1+Am_2)(y-y^{(1)}) + (-Bn_1+An_2)(z-z^{(1)}) , \end{aligned} \qquad (75)$$

where A and B are given by Eqs. (15) and (17). Conversely,

$$\begin{aligned} x-x^{(1)} &= \ell x_2 + (A\ell_1+B\ell_2)y_2 + (-B\ell_1+A\ell_2)z_2 , \\ y-y^{(1)} &= mx_2 + (Am_1+Bm_2)y_2 + (-Bm_1+Am_2)z_2 , \\ z-z^{(1)} &= nx_2 + (An_1+Bn_2)y_2 + (-Bn_1+An_2)z_2 . \end{aligned} \qquad (76)$$

In the x_2y_2 plane, the geometry is as shown in Fig. 8. By virtue of the definitions of x_2, y_2, z_2 in terms of x, y, z, the normal lies in this plane, and the angle ψ is such that (see Eq. 16)

$$\begin{aligned} \cos\psi &= \hat{y}_2 \cdot (\ell_n\hat{x} + m_n\hat{y} + n_n\hat{z}) \\ &= A(\ell_n\ell_1 + m_n m_1 + n_n n_1) + B(\ell_n\ell_2 + m_n m_2 + n_n n_2) \\ &= \Gamma \end{aligned} \qquad (77)$$

where

$$\Gamma = \left\{ (\ell_n\ell_1 + m_n m_1 + n_n n_1)^2 + (\ell_n\ell_2 + m_n m_2 + n_n n_2)^2 \right\}^{1/2} . \qquad (78)$$

Hence

$$\tan\psi = \frac{\sqrt{1-\Gamma^2}}{\Gamma} ,$$

and the line representing the intersection of the tangent plane with the plane $z_2=0$ now has equation

$$y_2 = \frac{\sqrt{1-\Gamma^2}}{\Gamma} \quad x_2 \ . \qquad (79)$$

If we step a distance Δ_1 along this line, the displacement being in the direction of increasing x_2 (so as to have a positive component in the direction of the incident ray), we arrive at the point

$$x_2 = \Delta_1\Gamma \ , \qquad y_2 = \Delta_1\sqrt{1-\Gamma^2} \qquad (80)$$

and here we now draw a line parallel to the original normal to the radome. The equation of this line is clearly

$$y_2 - \Delta_1\sqrt{1-\Gamma^2} = -\cot\psi\,(x_2 - \Delta_1\Gamma)$$

i.e.

$$y_2 = -\frac{\Gamma}{\sqrt{1-\Gamma^2}}\left(x_2 - \frac{\Delta_1}{\Gamma}\right) , \qquad (81)$$

and the point at which it intercepts the radome is our new, displaced intercept where penetration into the radome is presumed to take place. The corresponding ray has the direction that it originally had (of course), but its optical distance of travel is increased by an amount $\delta\ell$ (see Eq. 74) over and above its value d_{01} (see Eq. 10) appropriate to the undisplaced intercept. It will be appreciated that because of the curvature of the radome surface, the normal to the radome at the new intercept point will not, in general, lie in the x_2y_2 plane. It is therefore necessary to establish a further set of coordinates tailored to the plane of intercept at the new point, and to proceed as we did in transforming from the (x_1, y_1, z_1) system to the (x_2, y_2, z_2) system with the bare radome.

With the above formulation of the method, the determination of the point at which the line (81) intercepts the radome (always, of course, in the plane $z_2=0$) requires us to express the equation $\rho = \rho(z)$ of the outer surface of the radome in terms of the (x_2, y_2, z_2) coordinates. This is not necessarily a convenient thing to do, primarily because all of the coordinate transformations demanded by the bare radome analysis are merely rotations of base vectors, whereas we are here concerned with an actual coordinate system which is displaced in origin as well as being rotated relative to the original (x, y, z) system. The simplest way to avoid this difficulty is to express Eq. (81) in terms of (x, y, z). From Eq. (75) and bearing in mind that the line represented by (81) lies in the plane $z_2=0$, its equations now take the form

$$\left(A\ell_1+B\ell_2+\frac{\ell\Gamma}{\sqrt{1-\Gamma^2}}\right)(x-x^{(1)}) + \left(Am_1+bm_2+\frac{m\Gamma}{\sqrt{1-\Gamma^2}}\right)(y-y^{(1)}) + \left(An_1+Bn_2+\frac{n\Gamma}{\sqrt{1-\Gamma^2}}\right)(z-z^{(1)}) = \frac{\Delta_1}{\sqrt{1-\Gamma^2}} , \tag{82}$$

$$(-B\ell_1+A\ell_2)(x-x^{(1)})+(-Bm_1+Am_2)(y-y^{(1)})+(-Bn_1+An_2)(z-z^{(1)}) = 0 .$$

These can be reduced to

$$\left\{n\sqrt{1-\Gamma^2}-\Gamma(n_1A+n_2B)\right\}(x-x^{(1)})-\left\{\ell\sqrt{1-\Gamma^2}-\Gamma(\ell_1A+\ell_2B)\right\}(z-z^{(1)})= \Delta_1(-Bm_1+Am_2) ,$$

$$\left\{n\sqrt{1-\Gamma^2}-\Gamma(n_1A+n_2B)\right\}(y-y^{(1)})-\left\{m\sqrt{1-\Gamma^2}-\Gamma(m_1A+m_2B)\right\}(z-z^{(1)})=-\Delta_1(-B\ell_1+A\ell_2) , \tag{83}$$

but the form is still not a convenient one: it is by no means apparent that the line <u>does</u> have direction cosines ℓ_n, m_n, n_n, which it must have since it is parallel to the radome normal at the point $x^{(1)}, y^{(1)}, z^{(1)}$.

The final step in the reduction process is one of those on which hours can (and were) 'wasted'. We first note that because (ℓ, m, n), (ℓ_1, m_1, n_1) and (ℓ_2, m_2, n_2) are the direction cosines of three mutually orthogonal vectors,

$$\ell^2+\ell_1^2+\ell_2^2=m^2+m_1^2+m_2^2=n^2+n_1^2+n_2^2 = 1 \tag{84}$$

and

$$\ell m+\ell_1 m_1+\ell_2 m_2=\ell n+\ell_1 n_1+\ell_2 n_2 = mn+m_1 n_1+m_2 n_2 = 0 .$$

Hence, from Eq. (15),

$$\begin{aligned} n\sqrt{1-\Gamma^2}-\Gamma(n_1 A+n_2 B) &= n\sqrt{1-\Gamma^2}-\left\{n_n(n_1^2+n_2^2)+\ell_n(\ell_1 n_1+\ell_2 n_2)+m_n(m_1 n_1+m_2 n_2)\right\} \\ &= -n_n+n\left\{\sqrt{1-\Gamma^2}+(\ell_n \ell+m_n m+n_n n)\right\} , \end{aligned}$$

and by a further application of the identities (84), it can be shown that

$$(\ell_n \ell+m_n m+n_n n)^2+(\ell_n \ell_1+m_n m_1+n_n n_1)^2+(\ell_n \ell_2+m_n m_2+n_n n_2)^2 = 1 ,$$

i.e.

$$\ell_n \ell+m_n m+n_n n = -\sqrt{1-\Gamma^2} .$$

Similarly for the other factors in the Eqs. (83), leading to the following equations specifying the line:

$$\begin{aligned} x &= x^{(1)}+\frac{1}{n_n}\left\{\ell_n(z-z^{(1)})+\Delta_1 (Bm_1-Am_2)\right\} \\ y &= y^{(1)}+\frac{1}{n_n}\left\{m_n(z-z^{(1)})+\Delta_1 (B\ell_1-A\ell_2)\right\} . \end{aligned} \tag{85}$$

In this form, the equations are quite similar to those describing the original ray emanating from the point $x^{(0)}$, $y^{(0)}$, $z^{(0)}$ on the phase front. The point of interception of that ray with the radome specifies $z^{(1)}$ and hence, from (8), $x^{(1)}$ and $y^{(1)}$. The z coordinate, $z = z^{(2)}$, of the displaced intercept is given by

$$\left[x^{(1)}+\frac{1}{n_n}\left\{\ell_n(z-z^{(1)})+\Delta_1(Bm_1-Am_2)\right\}\right]^2+\left[y^{(1)}+\frac{1}{n_n}\left\{m_n(z-z^{(1)})+\Delta_1(B\ell_1-A\ell_2)\right\}\right]^2$$

$$=\left\{\rho(z)\right\}^2 , \qquad (86)$$

where Δ_1 is given its value appropriate to the original location $z=z^{(1)}$, and where $\rho = \rho(z)$ is the equation of the outer radome surface; and having determined $z^{(2)}$, the $x^{(2)}$ and $y^{(2)}$ follow from the Eqs. (85). The ray is now presumed to enter the radome at the displaced point $x^{(2)}, y^{(2)}, z^{(2)}$ rather than at the point $x^{(1)}, y^{(1)}, z^{(1)}$ that it would have done in the absence of the plasma, and the phase (or optical distance) that is associated with it is $d_{o1} + \delta\ell$ (see Eqs. (10) and (74)) instead of d_{o1} . Ray tracing then proceeds as it did in the case of the bare radome.

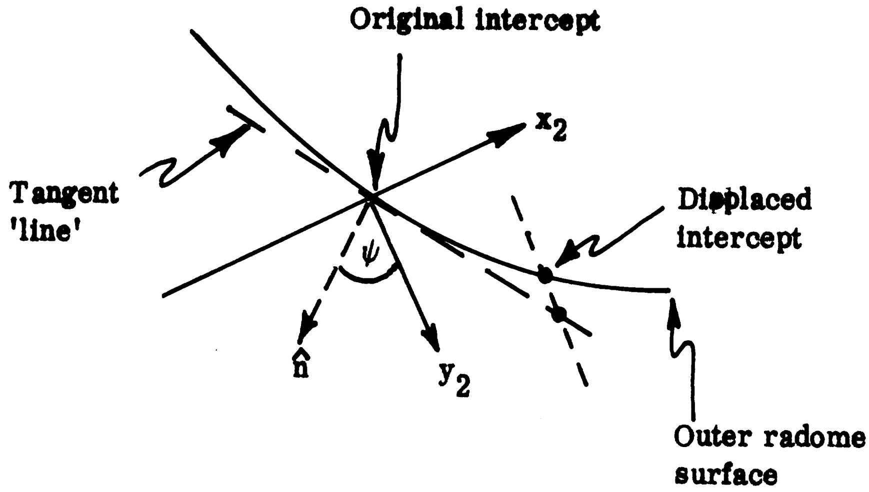

Fig. 8: Geometry for Displacement of the Intercept.

VI COMPUTER PROGRAM AND NUMERICAL DISCUSSION

An examination of the basic analyses presented in Sections II through V suggest that the computer program for ray tracing through a three-dimensional radome will be a fairly long and complicated one, and this is indeed the ease. The program consists of a main program, four subprograms and 17 subroutines. The FORTRAN listing of the complete program* is presented in the Appendix to this Memorandum.

In essence, the MAIN program controls the input and output data , initiates a sampling ray at the incident phase front, takes into account the plasma (if present), and guides a ray into the interior of the radome. If the ray hits the monopulse plate, the results are recorded or neglected depending on whether the ray intercepts a slot or not. Another possibility for a ray inside the radome is that it hit the upper surface**. In such a case, the SECOND program (a subprogram) takes over and follows the rays through reflection and refraction at the upper surface. The THIRD program picks up

---

*As noted in the introduction, the programs for the conical and ogival radomes are separate. Although the general format is the same for each, the following descriptions in this section pertain to the conical radome program.

**We assume here that the incident illumination is from below the axis of the radome.

the downward rays , some of which will either hit the monopulse plate or the lower surface. The rays that hit the monopulse plate are 'home' and are recorded, whilst the others are picked up by the FOURTH program and, if the situation calls for it, the FIFTH (and final) program that is similar to the THIRD. Throughout this ray tracing process, the rays resulting from multiple bounces within the dielectric layer are picked up by a subprogram appropriate to rays in that particular region. It is to be expected that in most cases the computations will be performed by the MAIN, SECOND and THIRD programs and only for angles of incidence greater than (about) 50° will the FOURTH and FIFTH programs come into play.

Subroutines are used to perform the mathematical, vector, and other operations required in the ray tracing procedure. In total there are 17 subroutines and these carry out the following operations:

1. SUBROUTINE REPEAT - traces a ray inside the dielectric layer; the transmitted portion of the ray is picked up by one of the subprograms.
2. SUBROUTINE PINT - determines the point of intersection of a ray with the outer (and inner) radome surfaces.
3. SUBROUTINE PNORM - computes the direction cosines for the normal to the surface at the point where a ray intersects.
4. SUBROUTINE ANGLES - computes the angles α and β as described in Fig. 3b.
5. SUBROUTINE NEWCOR - sets up a new coordinate system at the point of intersection of a ray with a surface.
6. SUBROUTINE TR12 - computes the transmission and reflection coefficients for a ray going from air into the dielectric.
7. SUBROUTINE TR 21 - computes the transmission and reflection coefficients for the ray going from the dielectric into air.
8. SUBROUTINE CONAB - computes the quantities A and B as defined in Eqs. (15) and (17).

9. FUNCTION DIST - computes the distance between two points.
10. SUBROUTINE TRANSM - determines the direction cosines for the transmitted ray.
11. SUBROUTINE REFLCT - determines the direction cosines for the reflected ray.
12. SUBROUTINE PLASMA - adjusts the path and optical distance traveled by a ray when it goes through a plasma layer.
13. SUBROUTINE RECORD - computes the intersection point of a ray with the monopulse plane.
14. SUBROUTINE SLOTS - determines which slot, if any, is intercepted by a ray.
15. SUBROUTINE APSLOT - computes the signal induced in the slot by a ray.
16. SUBROUTINE VECTOP - computes a vector cross product.
17. SUBROUTINE VECTDT - computes a vector dot product.

In the theoretical analysis as well as the computer program it is convenient to specify the direction of propagation and the polarization of the incident wave in terms of the direction cosines (ℓ, m, n) and (ℓ_1, m_1, n_1) respectively. Since it is not always easy to visualize the particular wave which is implied by given values of these quantities, we have also employed the same angles α_x and α_y that were used (see Section IV) to describe the rotation of the monopulse plate to describe the direction of propagation. Such a choice is natural inasmuch as we have specified that the initial alignment of the monopulse plate shall be perpendicular to the direction of incidence outside the radome and, hence, parallel to the external wavefront. In addition, the polarization of the incoming wave can be described in terms of the angle Ω referred to in Section IV.

To illustrate the application of these angles, consider the following situation. For a plane wave with propagation vector in the positive z direction (axial incidence), the phase front lies in the xy plane. The polarisation angle Ω is then the angle between the E vector and the x axis, and α_x and α_y are zero. We now imagine any arbitrary alignment of the wave front to be obtained by an initial rotation about the x axis through an angle α_y, followed by a rotation or tilting of the new plane through an angle α_x away from the vertical (see Fig. 7), and if we do this, the pertinent polarization angle Ω is the angle between the E vector and the x'' axis (see p. 27). Thus, for example, a vertically-polarized wave incident at an angle of 20^o below the z axis has

$$\alpha_x = 20^o, \ \alpha_y = 0 \ , \ \Omega = \ 0,$$

whereas for a horizontally-polarized wave incident in the horizontal plane but from a direction 15^o to the left as viewed from the monopulse plate,

$$\alpha_x = 0 \ , \ \alpha_y = 15^o, \ \Omega = 90^o \ .$$

The expressions for the direction cosines of the propagation and electric vectors in terms of α_x, α_y and Ω are given in Eqs. (60) and (61) respectively, and these have been incorporated into the MAIN program to enable the angles α_x (ANGLE X), α_y (ANGLE Y) and Ω (OMEGA) to be employed as input data. The other input parameters for the program are the stepping or sampling distance Δ (STEP OF DATA), radome refractive index $\tilde{n}$ (ANEX), presence (or otherwise) of the plasma (KPLASM = 1 or 0), equivalent radius a of the monopulse plate (CONSTANT A), and the number of increments (M(MAX), N(MAX)) over which the phase front sampling extends. It may be noted that the tolerance level, i.e. the level at which any ray amplitude is regarded as so small as to be negligible, is not an input parameter, and has been set at 0.01 in accordance with out findings in the two-dimensional study.

Concerning the choice of values for CONSTANT A, M(MAX) and N(MAX), we observe that together with (ℓ, m, n) these parameters define a set of coordinate points on the incident phase front from which the sampling rays originate. The procedure for specifying the sampling area was discussed in Section III, and in essence the area that must be covered is the projected area of the monopulse plate extended upwards to include the projection of the frontal portion of the radome. For computational purposes, however, it has been found convenient to specify a rectangular area which includes the above region. The rays that emanate from sampling points that do not produce intercepts with the radome are soon abandoned, and their initial inclusion in the computational sequence does not add significantly to the overall computation time. Indeed, the execution time required to find whether a ray does intercept the radome is only a small fraction (less than one percent) of that entailed in tracking an intercepting ray through all of its reflections and refractions to the monopulse plate.

The coordinates of a general point within the rectangular sampling area of the incident phase front are given by the Eqs. (59), and sampling is carried out at all such points having

$$M = -M_{max}, \; \ldots \; 0, \; M_{max} \, ,$$

$$N = 0, \; \ldots, \; N_{max} \, ,$$

with

$$M_{max} = \left[\frac{a}{\Delta}\right], \; N_{max} = \max\left(\left[\frac{2a}{\Delta}\right], \; \left[\frac{1}{\Delta}\left\{a + (z_a - z_n)\sqrt{1-n^2}\right\}\right]\right) . \tag{87}$$

In order to test the program and thereby obtain some assurance that it is operating correctly, a very small number of test runs of the conical radome program have been made for a specific angle of incidence of a plane wave on the conical radome described* by Dr. I. Pollin. The outer and inner radome

---

*Private communication, 13 November 1968.

surfaces were defined by Eq. (2) with

$$a = \frac{1}{6}, \quad \ell = 1 \qquad \text{(outer)},$$

$$a = \frac{1}{6}, \quad \ell = 4.191 \qquad \text{(inner)} .$$

The quantity z_n ($\equiv \ell$) above is therefore 1, and z_m = 43. All dimensions are, of course, in inches. The frequency employed was 4.918 GHz for which λ = 2.4, though we may note that other runs have been made for $\lambda = \pi$ (f = 3.757 GHz). For a fiberglass radome, the refractive index $\tilde{n}$ has been assumed to be 2.5 .

To test the computer program we chose the case

$\alpha_x = 15^o$, $\alpha_y = 5^o$, $\Omega = 10^o$
$a = 5.25$,
no plasma

and took $\Delta = 0.4$. This value of Δ is somewhat larger than is desirable for an accurate estimate of the pointing error. In the two-dimensional program we found that a stepping distance $\Delta = 0.2$ was necessary to obtain a pointing error that was not unduly sensitive to the particular choice of initial sampling point. For lack of any other information it would seem prudent to choose $\Delta = 0.2$ in the three-dimensional program as well, but since the objectives of the present test were to check the computer program, and to provide specific outputs against which to compare any subsequent runs of this program elsewhere, the larger value of Δ was entirely adequate. We may note that the choice $\Delta = 0.4$ reduces the number of sampling rays by a factor of 4, and decreases the running time (and cost) by the same amount.

To a person looking in the direction of the negative z axis, the above incident field is approaching the radome from a direction 15^o below the horizontal and 5^o to the left. The selection of a value for the polarization angle Ω not equal to zero was made only to produce a case of complete generality and to avoid any suggestion of symmetry. With the above input data we have, from Eq. (87),

$$M_{max} = 13, \quad N_{max} = 37 ,$$

implying $M_{max}(1+2N_{max}) = 999$ sampling rays of which about 500 will intercept the radome.

The computer printed out

a) the induced signal in each of the 32 slots,

b) the combined signal for each of the four quadrants,

c) the signals obtained by adding those in each of the two upper quadrants, and in each of the two lower quadrants,

d) the signals obtained by adding those in each of the two right-hand quadrants and in each of the two left-hand quadrants,

e) the pointing errors

$$\epsilon_V = -6.5 \quad \text{and } \epsilon_H = 9.1 \quad \text{milliradians}$$

for the vertical and horizontal planes respectively.

These pointing errors are about twice as large as were obtained with a similar case for the two-dimensional radome, but this is not unreasonable since the rays now undergo many more reflections and refractions, leading to perturbations of the wave front in a second plane. The signs of the pointing errors are also reasonable. According to definition, the pointing error is the angle between the direction of arrival of the signal outside the radome and the direction in which the monopulse plate points, i.e. the normal to the monopulse plate. If $\epsilon_V > 0$ ($\epsilon_V < 0$) the monopulse points too high (too low), and for $\epsilon_H > 0$ ($\epsilon_H < 0$) the plate is pointing too far to the right (or left). With $\alpha_x = 15^o$, the rays which strike the monopulse have come through the lower surfaces of the radome and the dominant contributions are provided by those rays which then proceed directly to the monopulse plate; in addition, however, there are contributions from rays reflected off the upper portion of the inner radome surface, and because of these it might be thought that the radome would tend to point above the true direction (i.e. give $\epsilon_V > 0$). This is not the case. The action of the monopulse plate is to rotate in such a way as to take up a position in a (mean) constant-phase plane. The rays which are reflected off the upper radome surface must travel a greater distance than the direct rays, and for the chosen angle of incidence, these rays are incident primarily on the upper slots. Accordingly, the phase of the signal induced in the upper slots should exceed that of the lower slot signals, and this was confirmed by an examination of the actual slot signals. To achieve a constant-phase plane, the monopulse must now point below the direction of the main (direct) rays, corresponding to a value of ϵ_V less than zero.

As a second test, the previous case was now run with the fiberglass radome replaced with a dimensionally-equivalent styrofoam one for which the appropriate refractive index is $\tilde{n} = 1.015$. Since $\tilde{n}$ is so little different from unity, the radome should have a correspondingly small effect on the field passing through it, and the test was run with the expectation that any small pointing errors computed would be due primarily to the low rate at which the wavefront is sampled. It was found that

$$\epsilon_V = -0.72 \qquad \text{and} \qquad \epsilon_H = -0.077 \text{ milliradians.}$$

Compared with 2 milliradians, ϵ_H is certainly so small as to be entirely negligible, but ϵ_V is almost 40 percent of the maximum pointing error that is tolerable, and is therefore too large to be ignored. Part of this is undoubtedly a real effect indicative of the actual field perturbation produced by the radome, but our belief is that a significant fraction of it is due to an insufficient sampling rate. This could be verified in the same manner as we did in the two-dimensional program by systematically displacing the initial sampling point a distance $\Delta / 10$ in each of two directions, and thereby generating a pointing error curve which is an oscillatory function of the displacement. Δ must then be chosen so that the total variation of this curve is within an acceptable range - say, less than 0.2 milliradians, and it was on this basis that we were led to choose $\Delta = 0.2$ in the two-dimensional program.

The third test run that was made was for the same situation as in the first run, i.e. for a fiberglass radome with $\alpha_x = 15^0$, $\alpha_y = 5^0$ and $\Omega = 10^0$, but with the plasma present. The pointing errors that were computed are

$$\epsilon_V = -7.1 \qquad \text{and} \qquad \epsilon_H = 8.5 \quad \text{milliradians.}$$

The change in pointing error produced by the plasma is quite small, amounting to only 0.6 milliradians in each direction.

VII CONCLUSIONS

In this Memorandum we have been concerned with the determination of the pointing errors associated with a general type of monopulse system mounted within a (three-dimensional) conical or ogival radome that may have a weak plasma surrounding it. In concept and formulation the method that we have developed is analogous to that previously discussed (Memorandum 02142-502-M, and referred to as 'S-L') in relation to a 'two-dimensional' radome, and it is suggested that the reader acquaint himself with this more simple case before attempting to understand the more complex analysis and procedures required by the three-dimensional geometry. We have omitted from the present Memorandum such details as are common to the two cases , and have included here only those features as are unique to the three-dimensional geometry. Nevertheless, it has been our intention to include full details of the complicated analytical techniques which the geometry entails, and print-outs of the resulting computer programs can be found in the Appendix.

We believe that these programs constitute a unique and versatile tool for assessing the influence of the radome (and, if present, plasma) on the behavior of a monopulse system, and because of the considerable experience gained from the study of the two-dimensional problem, the programs themselves are rather efficient. In spite of this, the running times of the program are not inconsiderable, and the contract funds available have permitted only such test runs as were essential to the checking out of the program. As an indication of these running times, we may note that each of the test runs 1 and 3 discussed in Section VI took between 3 and 4 minutes on an IBM 360/67 computer, and were the sampling distance reduced to no more than the 0.2 value required for an accurate pointing error determination, these times would increase by at least a factor 4. The times will be greater still at wide angles, and yet greater for the ogival radome where the computation of the ray intercepts is more troublesome. To compute the pointing errors for a variety of incidence angles (in two directions) and polarizations would therefore require no small amount of computer time.

In conclusion, it should be emphasized that because the entire approach has been based on ray theory, the results obtained are only approximate. It is our belief, based primarily on the detailed studies of the two-dimensional geometry, that the values for the pointing errors obtained with these programs will be accurate to within 1 milliradian, but prudence would dictate that before complete reliance is placed on these data, some attempt be made to verify these conclusions experimentally. We await such confirmation with considerable optimism in the outcome.

APPENDIX

COMPUTER PROGRAM PRINT-OUTS

```
      DIMENSION L(3),M(3),N(3),DN(3),
     1 XL1(3),XM1(3),XN1(3),
     2 XL2(3),XM2(3),XN2(3),
     3 XL3(3),XM3(3),XN3(3),
     4 XL4(3),XM4(3),XN4(3),
     5 XL5(3),XM5(3),XN5(3),
     6 XL6(3),XM6(3),XN6(3)
      DIMENSION AMPVAL(4,8)
      COMPLEX VALUE(4,8),SVALUE
      COMPLEX V1,V2,V3,V4,V12,V23,V34,V41
      REAL L,M,N
      REAL KCON
      COMMON VALUE/R/KCON,MAXN/RR/ANEX
      COMMON /RRR/BGL,CGL,L
      DATA PI/3.1415927/
      DATA DELTA/1.0/,DELT/0.689/
      REAL SLOTX(4,8)/3.85,3.85,-3.85,-3.85,3.65,3.65,-3.65,-3.65,
     1                2.35,2.35,-2.35,-2.35,2.15,2.15,-2.15,-2.15,
     2                2.35,2.35,-2.35,-2.35,0.85,0.85,-0.85,-0.85,
     3                0.65,0.65,-0.65,-0.65,0.85,0.85,-0.85,-0.85/
      REAL SLOTY(4,8)/0.78,-0.78,-0.78,0.78,2.305,-2.305,-2.305,2.305,
     1                0.78,-0.78,-0.78,0.78,2.305,-2.305,-2.305,2.305,
     2                3.83,-3.83,-3.83,3.83,0.78,-0.78,-0.78,0.78,
     3                2.305,-2.305,-2.305,2.305,3.83,-3.83,-3.83,3.83/
C     SAMPLE INPUT DATA
C       &INPUT ANGLEX=15.,ANGLEY=5.,OMEGA=10.,
C       DSTEP=0.4,DA=5.25,MMAX=13,NMAX=37,KPLSMA=0,ANEX=2.5 &END
      NAMELIST /INPUT/ANGLEX,ANGLEY,OMEGA,DSTEP,DA,MMAX,NMAX,KPLSMA,ANEX
    1 READ(1,INPUT)
      WRITE(6,3003)ANGLEX,ANGLEY,OMEGA,DSTEP,DA,MMAX,NMAX
 3003 FORMAT(1H1,'ANGLEX=',F6.2/'ANGLEY=',F6.2/'OMEGA=',F6.2/
     1 'STEP OF DATA=',F6.2/ 'CONSTANT A=',F6.2/
     2 'M(MAX)=',I5/ 'N(MAX)=',I5/)
      IF(KPLSMA .EQ. 0)GO TO 15
      WRITE(6,3004)
 3004 FORMAT('WITH PLASMA')
      GO TO 16
   15 WRITE(6,3005)
 3005 FORMAT('NO PLASMA')
   16 CONTINUE
      V1=(0.,0.)
      V2=(0.,0.)
      V3=(0.,0.)
      V4=(0.,0.)
      BGL=ANGLEX*PI/180.
      CGL=ANGLEY*PI/180.
      ROMEGA=OMEGA*PI/180.
      L(1)=SIN(BGL)
      L(2)=-COS(BGL)*SIN(CGL)
      L(3)=COS(BGL)*COS(CGL)
      M(1)=COS(BGL)*COS(ROMEGA)
      M(2)=SIN(BGL)*COS(ROMEGA)*SIN(CGL)+SIN(ROMEGA)*COS(CGL)
      M(3)=-SIN(BGL)*COS(ROMEGA)*COS(CGL)+SIN(ROMEGA)*SIN(CGL)
      WRITE(6,3001)L(1),L(2),L(3)
 3001 FORMAT(3X,2HL=F10.3,3X,2HM=F10.3,3X,2HN=F10.3)
      WRITE(6,3002)M(1),M(2),M(3)
 3002 FORMAT(2X,3HL1=F10.3,2X,3HM1=F10.3,2X,3HN1=F10.3//)
      CALL VECTCP(L,M,N)
      KCON=2.*PI/2.46
      ANC  =9.47*PI/180.
```

```
      AA=TAN(AACON)
      DELTB=DELTA+DELT/SIN(AACON)
      INDEX=1
      DLDZ=INDEX*AA
      DO 111 I=1,4
      DO 111 J=1,8
  111 VALUE(I,J)=(0.,0.)
      C1=L(1)*L(1)/L(3)/L(3)
      C2=-(86.+C1*AA)
      C3=C1*AA*DELTB+43.*43.
      C4=C2*C2-4.*C3
      IF(C4 .LT. 0.) GO TO 20
      C5=SQRT(C4)
      ZZ1=(-C2+C5)/2.
      ZZ2=(-C2-C5)/2.
      IF(ZZ1 .GT. ZZ2)GO TO 21
      ZEND1=ZZ1
      ZEND2=ZZ2
      GO TO 22
   21 ZEND1=ZZ2
      ZEND2=ZZ1
      GO TO 22
   20 WRITE(6,5000)
 5000 FORMAT('NO INTERSECTION FOR ZEND')
      GO TO 99
   22 CONTINUE
      SON2=SQRT(1.-L(3)*L(3))
      MM=2*MMAX+1
      DO 555 IIM=1,MM
      IM=IIM-MMAX-1
      XC1=IM*DSTEP/SON2
      DO 555 IIN=1,MMAX
      IN=IIN-1
      XC2=43.0+(DA-IN*DSTEP)/SON2
      XO=-L(1)*L(3)*XC2-XC1*L(2)
      YO=-L(2)*L(3)*XC2+XC1*L(1)
      ZO=-(L(1)*XO+L(2)*YO)/L(3)
      IF(ABS(L(1)) .LE. 0.001)GO TO 3
      ZINT=ZO-XO*L(3)/L(1)
      GO TO 2
    3 IF(ABS(L(2)) .LE. 0.001)GO TO 82
      ZINT=ZO-YO*L(3)/L(2)
    2 IF(ZINT .GT. DELTA .AND. ZINT .LT. DELTB)GO TO 99
      IF(ZINT .GT. DELTA)GO TO 80
      ISIGN=1
      GO TO 81
   82 IF(L(3) .EQ. 1.)GO TO 83
      GO TO 99
   83 IF(XO .EQ. 0. .AND. YO .EQ. 0.)GO TO 99
      IF(XO .GT. 0.)GO TO 87
      GO TO 80
   87 ISIGN=1
      GO TO 81
   80 ISIGN=0
   81 CONTINUE
      IF(ISIGN .EQ. 0)GO TO 60
      ZEND=ZEND2
      GO TO 61
   60 ZEND=ZEND1
   61 CONTINUE
```

```
C       FIND INTERSECTION POINT
        CALL PINT(DELTA,AA,X0,Y0,Z0,L,X1,Y1,Z1,KIC,ISIGN)
        IF(KIC .GT. 1)GO TO 99
        IF(Z1 .GE. ZEND2)GO TO 99
        IF(Z1 .GE. 68.)GO TO 99
C       COMPUTE DISTANCE
        PATH=DIST(X0,Y0,Z0,1.,X1,Y1,Z1)
C       FIND DERIVATIVE OF RO
C       FIND NORMAL DIRECTION AT INTERSECT POINT
        CALL PNORM(X1,Y1,Z1,DLDZ,DN)
C       FIND CONSTANTS A $ B AND GAMA
        CALL COMAB(M,N,DN,GA,GB,GAMA)
C       FIND ANGLES ALPHA AND BETA
        CALL ANGLES(L,DN,GAMA,ALPHA,BETA,0,KICK)
        IF(KICK .GT. 1)GO TO 99
        IF(KPLSMA .EQ. 1)GO TO 62
        GO TO 63
     62 CALL PLASMA(X1,Y1,Z1,L,M,N,DN,GA,GB,GAMA,
       1 ALPHA,BETA,DLDZ,DELTA,AA,PATH,IO)
        IF (IO .EQ. 0)GO TO 63
        GO TO 99
C       RAY WILL PASS FROM MEDIUM 1 TO 2
C       COMPUTE T12 AND R12
     63 CALL TR12(ALPHA,BETA,R12P,T12P,R12V,T12V)
        GAMAB=SQRT((GA*T12P)**2+(GB*T12V)**2)
        AMP=GAMAB
        CALL NEWCOR(L,M,N,GA,GB,XL1,XM1,XN1)
C       COMPUTE DIRECTION COSINES OF TRANSMITTED RAY
        CALL TRANSM(GA,GB,ALPHA,BETA,T12P,T12V,
       1 XL1,XM1,XN1,GAMAB,XL2,XM2,XN2)
C       RAY WILL PASS THE INNER SURFACE
C       COMPUTE INNER INTERSECT POINT
        CALL PINT(DELTB,AA,X1,Y1,Z1,XL2,X2,Y2,Z2,KIC,0)
        IF(KIC .GT. 1)GO TO 99
        PATH=PATH+DIST(X1,Y1,Z1,ANEX,X2,Y2,Z2)
        PAT=PATH
        DD=PATH
        CALL PNORM(X2,Y2,Z2,DLDZ,DN)
        CALL COMAB(XM2,XN2,DN,GA,GB,GAMA)
        CALL ANGLES(XL2,DN,GAMA,BETA,ALPHA,1,KICK)
        IF(KICK .GT. 1)GO TO 99
C       COMPUTE T21 AND R21
        CALL TR21(ALPHA,BETA,R21P,T21P,R21V,T21V)
        GAMAB=SQRT((GA*T21P)**2+(GB*T21V)**2)
        GAMABB=SQRT((GA*R21P)**2+(GB*R21V)**2)
        AMPT=AMP*GAMAB
        AMPR=AMP*GAMABB
        TAMP=AMPT
        CALL NEWCOR(XL2,XM2,XN2,GA,GB,XL3,XM3,XN3)
C       COMPUTE DIRECTION COSINES OF TRANSMITTED RAY
        CALL TRANSM(GA,GB,BETA,ALPHA,T21P,T21V,XL3,XM3,XN3,
       1 GAMAB,XL4,XM4,XN4)
        JJ=1
        CALL RECORD(L,XL4,X2,Y2,Z2,X3,Y3,Z3,DDT,K)
        IF(K .EQ. 1)GO TO 30
        IF(Z3 .LT. ZFID)GO TO 31
        GO TO 32
     30 DDT=DDT+DD
        CALL APSLOT(AMPT,Y3,Y3,XL4,XM4,
       1 SLOTX,SLOTY,III,JJJ,AMPSL)
        IF(JJJ .EQ. 9)GO TO 51
        AZCD=AZCD*DDT
        CSPD=COS(AZCD)
```

```
      SNKD=SIN(AKCON)
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   51 CONTINUE
      GO TO 32
   31 CALL SECOND(XL4,XM4,XN4,X2,Y2,Z2,AA,INDEX,
     1 DELTA,DELTB,PAT,TAMP,L,ISIGN,ZEND1,ZEND2)
   32 CALL REFLCT(GA,GB,BETA,R21P,R21V,XL3,XM3,XN3,GAMABB,
     1 XL5,XM5,XN5)
   40 JJ=JJ+1
      CALL REPEAT(INDEX,DELTA,DELTB,AA,X2,Y2,Z2,XL5,XM5,XN5,PATH,
     1 AMPT,AMPR,XL4,XM4,XN4,X3,Y3,Z3,DD,K,L,3)
      RAMP=AMPR
      TAMP=AMPT
      PAT=PATH
      IF(K .EQ. 1)GO TO 41
      IF(K .EQ. 0)GO TO 99
      IF(Z3 .LT. ZEIN)GO TO 44
      GO TO 42
   44 CALL SECOND(XL4,XM4,XN4,X2,Y2,Z2,AA,INDEX, DELTA,DELTB,PAT,TAMP,L,
     1 ISIGN,ZEND1,ZEND2)
      GO TO 42
   41 DDT=DD+PATH
      CALL APSLOT(AMPT,X3,Y3,XL4,XM4,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
      IF(JJJ .EQ. 9)GO TO 91
      AKCON=KCON*DDT
      SNKD=SIN(AKCON)
      CSKD=COS(AKCON)
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   91 CONTINUE
   42 IF(ABS(TAMP) .LE. 0.01 )GO TO 99
      IF(JJ .LT. 20)GO TO 40
   99 CONTINUE
  555 CONTINUE
      WRITE(6,9001)
 9001 FORMAT(1H1,2X,4HSLOT,3X,11X,4HREAL,6X,9HIMAGINARY,
     1 6X,9HAMPLITUDE/)
      DO 432 I=1,4
      DO 432 J=1,8
  432 AMPVAL(I,J)=CABS(VALUE(I,J))
      DO 431 I=1,4
      DO 431 J=1,8
      WRITE(6,9000)I,J,VALUE(I,J),AMPVAL(I,J)
 9000 FORMAT(2I5,3E15.5)
  431 CONTINUE
      DO 433 I=1,8
      V1=V1+VALUE(1,I)
      V2=V2+VALUE(2,I)
      V3=V3+VALUE(3,I)
      V4=V4+VALUE(4,I)
  433 CONTINUE
      V1A=CABS(V1)
      V2A=CABS(V2)
      V3A=CABS(V3)
      V4A=CABS(V4)
      PHI1=ATAN2(AIMAG(V1),REAL(V1))
      PHI2=ATAN2(AIMAG(V2),REAL(V2))
```

```
      PHI3=ATAN2(AIMAG(V3),REAL(V3))
      PHI4=ATAN2(AIMAG(V4),REAL(V4))
      V12=V1+V2
      V23=V2+V3
      V34=V3+V4
      V41=V4+V1
      V12A=CABS(V12)
      V23A=CABS(V23)
      V34A=CABS(V34)
      V41A=CABS(V41)
      PHI12=ATAN2(AIMAG(V12),REAL(V12))
      PHI23=ATAN2(AIMAG(V23),REAL(V23))
      PHI34=ATAN2(AIMAG(V34),REAL(V34))
      PHI41=ATAN2(AIMAG(V41),REAL(V41))
      WRITE(6,9004)
 9004 FORMAT(1X//10X,'(V1,V2,V3,V4)')
      WRITE(6,9006)
 9006 FORMAT(11X,4HREAL,6X,9HIMAGINARY,5X,10HABS. VALUE,10X,5HPHASE)
      WRITE(6,9002) V1,V1A,PHI1,
     1                   V2,V2A,PHI2,
     2                   V3,V3A,PHI3,
     3                   V4,V4A,PHI4,
      WRITE(6,9005)
 9005 FORMAT(1X//10X,'(V12,V23,V34,V41)')
      WRITE(6,9002)V12,V12A,PHI12,
     1    V23,V23A,PHI23,
     2    V34,V34A,PHI34,
     3    V41,V41A,PHI41
      VVV=1./4./PI
      VV1=VVV/0.8698*(PHI34-PHI12)
      VV2=VVV/0.8810*(PHI41-PHI23)
      ERRORV=ARSIN(VV1)
      ERRORH=ARSIN(VV2)
      WRITE(6,9003)ERRORV,ERRORH
 9003 FORMAT('***','ERRORV=',E12.5,5X,'ERRORH=',E12.5)
 9002 FORMAT(4E15.5)
      GO TO 1
      END
```

284 LINES PRINTE

```
      SUBROUTINE SECOND(XL,XM,XN,X2,Y2,Z2,AA,INDEX,
     1 DELTA,DELTB,PATH,BAMP,L,ISIGN,ZEND1,ZEND2,
     2SLOTX,SLOTY)
      DIMENSION SLOTX(4,8),SLOTY(4,8)
      DIMENSION DN(3),XL(3),XM(3),XN(3),
     1XL1(3),XM1(3),XN1(3),
     2XL2(3),XM2(3),XN2(3),
     3XL3(3),XM3(3),XN3(3),
     4XL4(3),XM4(3),XN4(3)
     5,XL7(3),XM7(3),XN7(3)
      DIMENSION L(3)
      COMPLEX VALUE(4,8),SVALUE
      REAL L
      REAL KCON
      COMMON VALUE/R/KCON,MAXN/RR/ANEX
      JJ=1
      IF(ISIGN .EQ. 0)GO TO 60
      ZFIN=ZEND1
      GO TO 61
   60 ZFIN=ZEND2
   61 CONTINUE
      CALL PINT(DELTB,AA,X2,Y2,Z2,XL,X5,Y5,Z5,KIC,1)
      IF(KIC .GT. 1)GO TO 17
      PATH=PATH+DIST(X2,Y2,Z2,1.,X5,Y5,Z5)
      DLDZ=INDEX*AA
      CALL PNORM(X5,Y5,Z5,DLDZ,DN)
      DO 10 I=1,3
   10 DN(I)=-DN(I)
      CALL CONAB(XM,XN,DN,GA,GB,GAMA)
      CALL NEWCOR(XL,XM,XN,GA,GB,XL1,XM1,XN1)
      CALL ANGLES(XL,DN,GAMA,ALPHA,BETA,0,KICK)
      IF(KICK .GT. 1)GO TO 17
      CALL IR12(ALPHA,BETA,R12P,T12P,R12V,T12V)
      GAMABB=SQRT((GA*R12P)**2+(GB*R12V)**2)
      GAMAB=SQRT((GA*T12P)**2+(GB*T12V)**2)
      AMPT=BAMP*GAMAB
      AMPR=BAMP*GAMABB
      CALL REFLCT(GA,GB,ALPHA,R12P,R12V,XL1,XM1,XN1,
     1 GAMABB,XL2,XM2,XN2)
      CALL RECORD(L,XL2,X5,Y5,Z5,X6,Y6,Z6,DD,K)
      IF(K .EQ. 1)GO TO 11
      IF(Z6 .LT. ZFIN)GO TO 30
      GO TO 12
   30 CALL THIRD(XL2,XM2,XN2,X5,Y5,Z5,AA,INDEX,DELTA,DELTB,
     1 PATH,AMPR,L,ISIGN,ZEND1,ZEND2,SLOTX,SLOTY)
      GO TO 12
   11 DD=DD+PATH
      CALL APSLOT(AMPT,X6,Y6,XL2,XM2,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
      IF(JJJ .EQ. 9)GO TO 51
      AKCON=KCON*DD
      CSKD=COS(AKCON)
      SNKD=SIN(AKCON)
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   51 CONTINUE
   12 CALL TRANSM(GA,GB,ALPHA,BETA,T12P,T12V,XL1,XM1,XN1,
     1 GAMAB,XL3,XM3,XN3)
   20 JJ=JJ+1
      CALL REPEAT(INDEX,DELTA,DELTB,AA,X5,Y5,Z5,XL3,XM3,XN3,PATH,
```

```
      1 [illegible],KK,L,III
      IF(KK .EQ. 1)GO TO 13
      IF(KK .EQ. 0)GO TO 17
      IF(ZZ .LT. ZFIN)GO TO 31
      GO TO 14
   31 CALL THIRD(XL7,XM7,XN7,X5,Y5,Z5,AA,INDEX,DELTA,DELTB,
     1 PATH,AMPT,L,ISIGN,ZEND1,ZEND2,SLOTX,SLOTY)
      GO TO 14
   13 DD=DD+PATH
      IF(ABS(AMPT) .LT. 0.01 )GO TO 17
      CALL APSLOT(AMPT,X7,Y7,XL7,XM7,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
      IF(JJJ .EQ. 9)GO TO 91
      AKCON=KCON*DD
      CSKD=COS(AKCON)
      SNKD=SIN(AKCON)
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   91 CONTINUE
   14 IF(ABS(AMPT) .LE. 0.01 )GO TO 17
      IF(JJ .GE. 20)GO TO 17
      GO TO 20
   17 RETURN
      END
      SUBROUTINE THIRD(XL,XM,XN,X2,Y2,Z2,AA,INDEX,
     1 DELTA,DELTB,PATH,BAMP,L,ISIGN,ZEND1,ZEND2,
     2SLOTX,SLOTY)
      DIMENSION SLOTX(4,8),SLOTY(4,8)
      DIMENSION DN(3),XL(3),XM(3),XN(3),
     1XL1(3),XM1(3),XN1(3),
     2XL2(3),XM2(3),XN2(3),
     3XL3(3),XM3(3),XN3(3),
     4XL4(3),XM4(3),XN4(3)
     5,XL7(3),XM7(3),XN7(3)
      DIMENSION L(3)
      COMPLEX VALUE(4,8),SVALUE
      REAL L
      REAL KCON
      COMMON VALUE/R/KCON,MAXN/RR/ANEX
      JJ=1
      IF(ISIGN .EQ. 0)GO TO 40
      ZFIN=ZEND2
      GO TO 41
   40 ZFIN=ZEND1
   41 CONTINUE
      CALL PIOT(DELTB,AA,X2,Y2,Z2,XL,X5,Y5,Z5,KIC,1)
      IF(KIC .GT. 1)GO TO 17
      PATH=PATH+DIST(X2,Y2,Z2,1.,X5,Y5,Z5)
      DLDZ=INDEX*AA
      CALL NORM(X5,Y5,Z5,DLDZ,DN)
      DO 10 I=1,3
   10 DN(I)=-DN(I)
      CALL COMAR(XM,XN,DN,GA,GB,GAMA)
      CALL NEWCOR(XL,XM,XN,GA,GB,XL1,XM1,XN1)
      CALL ANGLES(XL,DN,GAMA,ALPHA,BETA,0,KICK)
      IF(KICK .GT. 1)GO TO 17
      CALL TR12(ALPHA,BETA,R12P,T12P,R12V,T12V)
      GAMARR=SQRT((GA*R12P)**2+(GB*R12V)**2)
      GAMAR=SQRT((GA*T12P)**2+(GB*T12V)**2)
      AMPT=BAMP*GAMAR
      AMPR=BAMP*GAMARR
      CALL REFLCT(GA,GB,ALPHA,R12P,R12V,XL1,XM1,XN1,
```

```
     1 GAMMAB,XL2,XM2,XN2)
       CALL RECORD(L,XL2,X5,Y5,Z5,X6,Y6,Z6,DD,K)
       IF(K .EQ. 1)GO TO 11
       IF(Z6 .LT. ZFIN)GO TO 30
       GO TO 12
    30 CALL FOURTH(XL2,XM2,XN2,X5,Y5,Z5,AA,INDEX,DELTA,DELTB,PATH,
     1 AMPR,L,ISIGN,ZEND1,ZEND2,SLOTX,SLOTY)
       GO TO 12
    11 DD=DD+PATH
       CALL APSLOT(AMPT,X6,Y6,XL2,XM2,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
       IF(JJJ .EQ. 9)GO TO 91
       AKCON=KCON*DD
       CSKD=COS(AKCON)
       SNKD=SIN(AKCON)
       AMPRL=AMPSL*CSKD
       AMPIM=AMPSL*SNKD
       SVALUE=CMPLX(AMPRL,AMPIM)
       VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
    91 CONTINUE
    12 CALL TRANSM(GA,GB,ALPHA,BETA,T12P,T12V,XL1,XM1,XN1,
     1 GAMMAB,XL3,XM3,XN3)
    20 JJ=JJ+1
       CALL REPEAT(INDEX,DELTA,DELTB,AA,X5,Y5,Z5,XL3,XM3,XN3,PATH,
     1 AMPT,AMPR,XL7,XM7,XN7,X7,Y7,Z7,DD,KK,L,JJ)
       IF(KK .EQ. 1)GO TO 13
       IF(KK .EQ. 0)GO TO 17
       IF(Z7 .LT. ZFIN)GO TO 31
       GO TO 14
    31 CALL FOURTH(XL7,XM7,XN7,X5,Y5,Z5,AA,INDEX,DELTA,DELTB,PATH,
     1 AMPT,L,ISIGN,ZEND1,ZEND2,SLOTX,SLOTY)
       GO TO 14
    13 DD=DD+PATH
       IF(ABS(AMPT) .LT. 0.01 )GO TO 17
       CALL APSLOT(AMPT,X7,Y7,XL7,XM7,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
       IF(JJJ .EQ. 9)GO TO 51
       AKCON=KCON*DD
       CSKD=COS(AKCON)
       SNKD=SIN(AKCON)
       AMPRL=AMPSL*CSKD
       AMPIM=AMPSL*SNKD
       SVALUE=CMPLX(AMPRL,AMPIM)
       VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
    51 CONTINUE
    14 IF(ABS(AMPT) .LE. 0.01 )GO TO 17
       IF(JJ .GE. 20)GO TO 17
       GO TO 20
    17 RETURN
       END
       SUBROUTINE FOURTH(XL,XM,XN,X2,Y2,Z2,AA,INDEX,
     1 DELTA,DELTB,PATH,BAMP,L,ISIGN,ZEND1,ZEND2,
     2SLOTX,SLOTY)
       DIMENSION SLOTX(4,8),SLOTY(4,8)
       DIMENSION DD(3),XL(3),XM(3),XN(3),
     1XL1(3),XM1(3),XN1(3),
     2XL2(3),XM2(3),XN2(3),
     3XL3(3),XM3(3),XN3(3),
     4XL4(3),XM4(3),XN4(3)
     5,XL7(3),XM7(3),XN7(3)
```

```
      DIMENSION L(3)
      COMPLEX VALUE(4,8),SVALUE
      REAL L
      REAL KCON
      COMMON VALUE/R/KCON,MAXN/RR/ANEX
      JJ=1
      IF(ISIGN .EQ. 0)GO TO 60
      ZFIM=ZEND1
      GO TO 61
   60 ZFIM=ZEND2
   61 CONTINUE
      CALL PINT(DELTB,AA,X2,Y2,Z2,XL,X5,Y5,Z5,KIC,1)
      IF(KIC .GT. 1)GO TO 17
      PATH=PATH+DIST(X2,Y2,Z2,1.,X5,Y5,Z5)
      DLDZ=INDEX*AA
      CALL PNORM(X5,Y5,Z5,DLDZ,DN)
      DO 10 I=1,3
   10 DN(I)=-DN(I)
      CALL CONAB(XM,XN,DN,GA,GB,GAMA)
      CALL NEWCOR(XL,XM,XN,GA,GB,XL1,XM1,XN1)
      CALL ANGLES(XL,DN,GAMA,ALPHA,BETA,0,KICK)
      IF(KICK .GT. 1)GO TO 17
      CALL TR12(ALPHA,BETA,R12P,T12P,R12V,T12V)
      GAMABB=SQRT((GA*R12P)**2+(GB*R12V)**2)
      GAMAB=SQRT((GA*T12P)**2+(GB*T12V)**2)
      AMPT=BAMP*GAMAB
      AMPR=BAMP*GAMABB
      CALL REFLCT(GA,GB,ALPHA,R12P,R12V,XL1,XM1,XN1,
     1 GAMABB,XL2,XM2,XN2)
      CALL RECORD(L,XL2,X5,Y5,Z5,X6,Y6,Z6,DD,K)
      IF(K .EQ. 1)GO TO 11
      IF(Z6 .LT. ZFIM)GO TO 30
      GO TO 12
   30 CALL FIFTH(XL2,XM2,XN2,X5,Y5,Z5,AA,INDEX,DELTA,DELTB,
     1 PATH,AMPR,L,ISIGN,ZEND1,ZEND2,SLOTX,SLOTY)
      GO TO 12
   11 DD=DD+PATH
      CALL APSLOT(AMPT,X6,Y6,XL2,XM2,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
      IF(JJJ .EQ. 9)GO TO 51
      AKCON=KCON*DD
      CSKD=COS(AKCON)
      SNKD=SIN(AKCON)
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   51 CONTINUE
   12 CALL TRANSM(GA,GB,ALPHA,BETA,T12P,T12V,XL1,XM1,XN1,
     1 GAMAB,XL3,XM3,XN3)
   20 JJ=JJ+1
      CALL REPEAT(INDEX,DELTA,DELTB,AA,X5,Y5,Z5,XL3,XM3,XN3,PATH,
     1 AMPT,AMPR,XL7,XM7,XN7,X7,Y7,Z7,DD,KK,L,JJ)
      IF(KK .EQ. 1)GO TO 13
      IF(KK .EQ. 0)GO TO 17
      IF(Z7 .LT. ZFIM)GO TO 31
      GO TO 14
   31 CALL FIFTH(XL7,XM7,XN7,X5,Y5,Z5,AA,INDEX,DELTA,DELTB,
     1 PATH,AMPT,L,ISIGN,ZEND1,ZEND2,SLOTX,SLOTY)
      GO TO 14
```

```
   13 DD=DD+PATH
      IF(ABS(AMPT) .LT. 0.01 )GO TO 17
      CALL APSLOT(AMPT,X7,Y7,XL7,XM7,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
      IF(JJJ .EQ. 9)GO TO 91
      AKCOM=KCON*DD
      CSKD=COS(AKCOM)
      SNKD=SIN(AKCOM)
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   91 CONTINUE
   14 IF(ABS(AMPT) .LE. 0.01 )GO TO 17
      IF(JJ .GE. 20)GO TO 17
      GO TO 20
   17 RETURN
      END
      SUBROUTINE FIFTH(XL,XM,XN,X2,Y2,Z2,AA,INDEX,
     1 DELTA,DELTB,PATH,BAMP,L,ISIGN,ZEND1,ZEND2,
     2SLOTX,SLOTY)
      DIMENSION SLOTX(4,8),SLOTY(4,8)
      DIMENSION DN(3),XL(3),XM(3),XN(3),
     1XL1(3),XM1(3),XN1(3),
     2XL2(3),XM2(3),XN2(3),
     3XL3(3),XM3(3),XN3(3),
     4XL4(3),XM4(3),XN4(3)
     5,XL7(3),XM7(3),XN7(3)
      DIMENSION L(3)
      COMPLEX VALUE(4,8),SVALUE
      REAL L
      REAL KCON
      COMMON VALUE/R/KCON,MAXN/RR/ANEX
      JJ=1
      CALL PINT(DELTB,AA,X2,Y2,Z2,XL,X5,Y5,Z5,KIC,1)
      IF(KIC .GT. 1)GO TO 17
      PATH=PATH+DIST(X2,Y2,Z2,1.,X5,Y5,Z5)
      DLDZ=INDEX*AA
      CALL PNORM(X5,Y5,Z5,DLDZ,DN)
      DO 10 I=1,3
   10 DN(I)=-DN(I)
      CALL CONAB(XM,XN,DN,GA,GB,GAMA)
      CALL NEWCOR(XL,XM,XN,GA,GB,XL1,XM1,XN1)
      CALL ANGLES(XL,DN,GAMA,ALPHA,BETA,0,KICK)
      IF(KICK .GT. 1)GO TO 17
      CALL TR12(ALPHA,BETA,R12P,T12P,R12V,T12V)
      GAMABB=SQRT((GA*R12P)**2+(GB*R12V)**2)
      GAMAB=SQRT((GA*T12P)**2+(GB*T12V)**2)
      AMPT=BAMP*GAMAB
      AMPR=BAMP*GAMABB
      IF(AMPR .LE. 0.01 )GO TO 17
      CALL REFLCT(GA,GB,ALPHA,R12P,R12V,XL1,XM1,XN1,
     1 GAMABB,XL2,XM2,XN2)
      CALL RECORD(L,XL2,X5,Y5,Z5,X6,Y6,Z6,DD,K)
      IF(K .EQ. 1)GO TO 11
      GO TO 12
   11 DD=DD+PATH
      CALL APSLOT(AMPT,X6,Y6,XL2,XM2,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
      IF(JJJ .EQ. 9)GO TO 91
      AKCOM=KCON*DD
      CSKD=COS(AKCOM)
      SNKD=SIN(AKCOM)
```

```
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   91 CONTINUE
   12 CALL TRANSM(GA,GB,ALPHA,BETA,T12P,T12V,XL1,XM1,XN1,
     1 GAMAB,XL3,XM3,XN3)
   20 JJ=JJ+1
      CALL REPEAT(INDEX,DELTA,DELTB,AA,X5,Y5,Z5,XL3,XM3,XN3,PATH,
     1 AMPT,AMPR,XL7,XM7,XN7,X7,Y7,Z7,DD,KK,L,JJ)
      IF(KK .EQ. 1)GO TO 13
      IF(KK .EQ. 0)GO TO 17
      GO TO 14
   13 DD=DD+PATH
      IF(ABS(AMPT) .LT. 0.01 )GO TO 17
      CALL APSLOT(AMPT,X7,Y7,XL7,XM7,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
      IF(JJJ .EQ. 9)GO TO 51
      AKCON=KCON*DD
      CSKD=COS(AKCON)
      SNKD=SIN(AKCON)
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   51 CONTINUE
   14 IF(ABS(AMPT) .LE. 0.01 )GO TO 17
      IF(JJ .GE. 10)GO TO 17
      GO TO 20
   17 RETURN
      END
```

340 LINES PRINTE

```
      SUBROUTINE REPEAT(INDEX,DELTA,DELTB,AA,X2,Y2,Z2
     1 ,XL5,XM5,XN5,PATH,AMPT,AMPR,XL4,XM4,XN4,
     2 X3,Y3,Z3,DD,K,L,JJ)
      DIMENSION DN(3),L(3),
     1XL2(3),XM2(3),XN2(3),
     2XL3(3),XM3(3),XN3(3),
     3XL4(3),XM4(3),XN4(3),
     4XL5(3),XM5(3),XN5(3),
     5XL6(3),XM6(3),XN6(3)
      REAL L
      COMMON /RR/ANEX
      CALL PINT(DELTA,AA,X2,Y2,Z2,XL5,X1,Y1,Z1,KIC,1)
      IF(KIC .GT. 1)GO TO 10
      PATH=PATH+DIST(X2,Y2,Z2,ANEX,X1,Y1,Z1)
      DLDZ=INDEX*AA
      CALL PNORM(X1,Y1,Z1,DLDZ,DN)
      DO 48 I=1,3
   48 DN(I)=-DN(I)
      CALL COMAB(XM5,XN5,DN,GA,GB,GAMA)
      CALL NEWCOR(XL5,XM5,XN5,GA,GB,XL6,XM6,XN6)
      CALL ANGLES(XL5,DN,GAMA,BETA,ALPHA,1,KICK)
      IF(KICK .GT. 1)GO TO 10
      CALL TR21(ALPHA,BETA,R21P,T21P,R21V,T21V)
      GAMABB=SQRT((GA*R21P)**2+(GB*R21V)**2)
      IF(JJ .LE. 2)GO TO 11
      AMP=AMPR*GAMABB
      GO TO 12
   11 AMP=AMPT*GAMABB
   12 CONTINUE
      CALL REFLCT(GA,GB,BETA,R21P,R21V,XL6,XM6,XN6,
     1 GAMABB,XL2,XM2,XN2)
      CALL PINT(DELTB,AA,X1,Y1,Z1,XL2,X2,Y2,Z2,KIC,0)
      IF(KIC .GT. 1)GO TO 10
      PATH=PATH+DIST(X1,Y1,Z1,ANEX,X2,Y2,Z2)
      CALL PNORM(X2,Y2,Z2,DLDZ,DN)
      CALL COMAB(XM2,XN2,DN,GA,GB,GAMA)
      CALL ANGLES(XL2,DN,GAMA,BETA,ALPHA,1,KICK)
      IF(KICK .GT. 1)GO TO 10
      CALL TR21(ALPHA,BETA,R21P,T21P,R21V,T21V)
      GAMA=SQRT((GA*T21P)**2+(GB*T21V)**2)
      GAMAB=SQRT((GA*R21P)**2+(GB*R21V)**2)
      AMPT=AMP*GAMA
      AMPR=AMP*GAMAB
      CALL NEWCOR(XL2,XM2,XN2,GA,GB,XL3,XM3,XN3)
      CALL TRANSM(GA,GB,BETA,ALPHA,T21P,T21V,XL3,XM3,XN3,
     1 GAMA,XL4,XM4,XN4)
      CALL REFLCT(GA,GB,BETA,R21P,R21V,XL3,XM3,XN3,
     1 GAMAB,XL5,XM5,XN5)
      CALL RECORD(L,XL4,X2,Y2,Z2,X3,Y3,Z3,DD,K)
      RETURN
   10 K=0
      RETURN
      END
      SUBROUTINE PINT(DEL,A,X0,Y0,Z0,DL,X1,Y1,Z1,KIC,JJJ)
      DIMENSION DL(3)
      XX=DL(1)*DL(1)+DL(2)*DL(2)
      X2=XX/DL(3)/DL(3)
      C1=A*A-X2
      X3=X0*DL(1)+Y0*DL(2)
      X4=X3/DL(3)
      X5=X0*X0+Y0*Y0
      C2=-2.*(DEL*A*A+X4-X2*Z0)
      C3=A*A*DEL*DEL-X5+2.*X4*Z0-X2*Z0*Z0
```

```
      X6=C2*C2-4.*C1*C3
      IF(X6 .GE. 0.)GO TO 2
      KIC=2
      GO TO 5
    2 C5=SQRT(X6)
      ZZ1=(-C2-C5)/2./C1
      ZZ2=(-C2+C5)/2./C1
      IF(ZZ1 .LE. DEL .OR. ZZ2 .LE. DEL)GO TO 8
      IF(ZZ1 .GT. ZZ2) GO TO 4
      IF(JJJ .EQ. 1)GO TO 6
      ZZ=ZZ1
      GO TO 3
    6 ZZ=ZZ2
      GO TO 3
    4 IF(JJJ .EQ. 1)GO TO 7
      ZZ=ZZ2
      GO TO 3
    7 ZZ=ZZ1
    3 X1=X0+(ZZ-Z0)*DL(1)/DL(3)
      Y1=Y0+(ZZ-Z0)*DL(2)/DL(3)
      Z1=ZZ
      KIC=0
      GO TO 5
    8 KIC=2
    5 RETURN
      END
      FUNCTION DIST(X0,Y0,Z0,BNEX,X1,Y1,Z1)
      DIST=(X1-X0)**2+(Y1-Y0)**2+(Z1-Z0)**2
      DIST=SQRT(DIST)*BNEX
      RETURN
      END
      SUBROUTINE PNORM(X1,Y1,Z1,DLDZ,DN)
      DIMENSION DN(3)
      P1=SQRT(1+DLDZ*DLDZ)
      P2=SQRT(X1*X1+Y1*Y1)
      DN(1)=X1/P1/P2
      DN(2)=Y1/P1/P2
      DN(3)=-DLDZ/P1
      RETURN
      END
      SUBROUTINE COMAB(M,N,DN,A,B,GAMA)
      DIMENSION M(3),N(3),DN(3)
      REAL M,N
      G1=VECTOT(DN,M)
      G2=VECTOT(DN,N)
      G3=SQRT(G1*G1+G2*G2)
      GAMA=-1./G3
      A=-GAMA*G1
      B=-GAMA*G2
      RETURN
      END
      SUBROUTINE ANGLES(L,DN,GAMA,ALPHA,BETA,II,KICK)
      DIMENSION L(3),DN(3)
      COMMON /RR/ANEX
      REAL L
      PI=3.1415927
      ARG1=-VECTOT(L,DN)
      ARG2=-1./GAMA
      IF(ABS(ARG2) .GT. 1)GO TO 10
      GO TO 11
   10 KICK=2
      GO TO 23
   11 IF(II .EQ. 1)GO TO 2
```

```
      ARG3=ARG2/ANEX
      GO TO 3
    2 ARG3=ARG2*ANEX
      IF(ABS(ARG3) .GE. 1.)GO TO 4
      GO TO 3
    4 KICK=2
      GO TO 23
    3 CONTINUE
      IF(ARG1 .GE. 0. .AND. ARG2 .GE. 0.) GO TO 21
      KICK=2
      GO TO 23
   21 ALPHA=ARSIN(ARG2)
      BETA=ARSIN(ARG3)
      KICK=0
   23 RETURN
      END
      SUBROUTINE TR12(ALPHA,BETA,RP,TP,RV,TV)
      COMMON /RR/ANEX
      ALBT=ALPHA-BETA
      GX=ANEX*COS(ALPHA)/COS(BETA)
      RP=(1.-GX)/(1.+GX)
      TP=2./(1+GX)
      GY=ANEX*COS(BETA)/COS(ALPHA)
      RV=(1.-GY)/(1.+GY)
      TV=2./(1.+GY)
      RETURN
      END
      SUBROUTINE TR21(ALPHA,BETA,RP,TP,RV,TV)
      COMMON /RR/ANEX
      ALBT=ALPHA-BETA
      GX=ANEX*COS(ALPHA)/COS(BETA)
      GY=ANEX*COS(BETA)/COS(ALPHA)
      RP=-(1.-GX)/(1.+GX)
      TP=2.*GX/(1.+GX)
      RV=-(1.-GY)/(1.+GY)
      TV=2.*GY/(1.+GY)
      RETURN
      END
      SUBROUTINE NEWCOR(L,M,N,A,B,XL,XM,XN)
      DIMENSION L(3),M(3),N(3),XL(3),XM(3),XN(3)
      REAL L,M,N
      DO 30 I=1,3
      XL(I)=L(I)
      XM(I)=A*M(I)+B*N(I)
   30 XN(I)=-B*M(I)+A*N(I)
      RETURN
      END
      SUBROUTINE TRANSM(A,B,ALPHA,BETA,TP,TV,L,M,N,GAMA,XL,XM,XN)
      DIMENSION L(3),M(3),N(3),XL(3),XM(3),XN(3)
      DIMENSION YL(3),YM(3),YN(3)
      REAL L,M,N
      ALBT=ALPHA-BETA
      DO 32 I=1,3
      YL(I)=COS(ALBT)*L(I)-SIN(ALBT)*M(I)
      YM(I)=SIN(ALBT)*L(I)+COS(ALBT)*M(I)
   32 YN(I)=N(I)
      DO 33 I=1,3
      XL(I)=YL(I)
   33 XM(I)=(A*TP*YM(I)-B*TV*YN(I))/GAMA
      CALL VECTCP(XL,XM,XN)
```

```
      RETURN
      END
      SUBROUTINE REFLCT(A,B,ALPHA,RP,RV,L,M,N,GAMA,XL,XM,XN)
      DIMENSION L(3),M(3),N(3),XL(3),XM(3),XN(3)
      DIMENSION YL(3),YM(3),YN(3)
      REAL L,M,N
      ALPH2=ALPHA*2.
      DO 32 I=1,3
      YL(I)=-COS(ALPH2)*L(I)+SIN(ALPH2)*M(I)
      YM(I)=-SIN(ALPH2)*L(I)-COS(ALPH2)*M(I)
   32 YN(I)=N(I)
      DO 33 I=1,3
      YL(I)=YL(I)
   33 XM(I)=(-A*RP*YM(I)-B*RV*YN(I))/GAMA
      CALL VECTCP(XL,XM,XN)
      RETURN
      END
      SUBROUTINE RECORD(L,XL,X2,Y2,Z2,X,Y,Z,DD,K)
      DIMENSION L(3),XL(3)
      REAL L
      C1=L(1)*XL(1)/XL(3)+L(2)*XL(2)/XL(3)
      C2=C1*Z2
      C3=-L(1)*X2-L(2)*Y2+L(3)*43.
      C4=C2+C3
      C5=C1+L(3)
      Z=C4/C5
      X=X2+XL(1)/XL(3)*(Z-Z2)
      Y=Y2+XL(2)/XL(3)*(Z-Z2)
      CC=X*X+Y*Y+(Z-43.)**2
      DD=DIST(X2,Y2,Z2,1.0,X,Y,Z)
      IF(CC .LE. 30.25)GO TO 2
      K=3
      GO TO 3
    2 K=1
    3 RETURN
      END
      SUBROUTINE VECTCP(L,M,N)
      DIMENSION L(3),M(3),N(3)
      REAL L,M,N
      N(1)=L(2)*M(3)-L(3)*M(2)
      N(2)=L(3)*M(1)-L(1)*M(3)
      N(3)=L(1)*M(2)-L(2)*M(1)
      RETURN
      END
      FUNCTION VECTDT(L,M)
      DIMENSION L(3),M(3),N(3)
      REAL L,M,N
      DO 1 I=1,3
    1 N(I)=L(I)*M(I)
      VECTDT=N(1)+N(2)+N(3)
      RETURN
      END
      SUBROUTINE SLOTS(XX,YY,BGL,III,JJJ,CGL)
      IF(XX .GT. 0.) GO TO 2
      IF(YY .GT. 0.)GO TO 3
      III=3
      GO TO 5
    3 III=4
      GO TO 5
    2 IF(YY .GT. 0.)GO TO 4
```

```
      III=2
      GO TO 5
    4 III=1
    5 CONTINUE
      CCC=COS(BGL)
      DDD=COS(CGL)
      X3=4.5*CCC
      X2=3.0*CCC
      X1=1.5*CCC
      Y1=1.5*DDD
      Y2=3.0*DDD
      Y3=4.5*DDD
      X=ABS(XX)
      Y=ABS(YY)
      IF(X .LT. X3 .AND. Y .LT. Y3)GO TO 10
      GO TO 30
   10 IF(X .LT. X2 .AND. Y .LT. Y2)GO TO 11
      IF(X .GE. X2 .AND. Y .GE. Y2)GO TO 30
      IF(X .GE. X2 .AND. Y .LE. Y2)GO TO 12
      IF(X .GT. X1)GO TO 13
      JJJ=8
      GO TO 20
   13 JJJ=5
      GO TO 20
   12 IF(Y .GT. Y1)GO TO 14
      JJJ=1
      GO TO 20
   14 JJJ=2
      GO TO 20
   11 IF(X .LE. X1 .AND. Y .LE. Y1)GO TO 15
      IF(X .GE. X1 .AND. Y .GE. Y1)GO TO 16
      IF(X .GE. X1)GO TO 17
      JJJ=7
      GO TO 20
   17 JJJ=3
      GO TO 20
   16 JJJ=4
      GO TO 20
   15 JJJ=6
   20 RETURN
   30 JJJ=9
      RETURN
      END
      SUBROUTINE APSLOT(AMP,X,Y,XL,XM,SLOTX,SLOTY,III,JJJ,AMPSL)
      DIMENSION SLOTX(4,8),SLOTY(4,8),XL(3),XM(3)
      DIMENSION L(3)
      COMMON /RRR/BGL,CGL,L
      REAL L
      PI=3.1415927
      H=1.06
      W=1.060
      HH=H*COS(BGL)
      WW=W*COS(CGL)
      HO2=HH/2.
      WO2=WW/2.
      CALL SLOTS(X,Y,BGL,III,JJJ,CGL)
      IF(JJJ .EQ. 9)GO TO 3
      CX=SLOTX(III,JJJ)*COS(BGL)
      CY=SLOTY(III,JJJ)*COS(CGL)
      PX=ABS(X-CX)
      PY=ABS(Y-CY)
      IF(PX .LE. HO2 .AND. PY .LE. WO2)GO TO 2
      III=0
```

```
      GO TO 3
    2 AMPSL=AMP*COS(PY*PI/WW)*COS(PX*PI/HH)
      AC1=VECTDT(XL,L)
      B1=L(2)*L(2)+L(3)*L(3)
      B2=SQRT(B1)
      B3=L(2)*XL(2)+L(3)*XL(3)
      B4=L(2)*XM(2)+L(3)*XM(3)
      AC2=(XM(1)*B3+XL(1)*B4)/B2
      AMPSL=AMPSL*AC1*AC2
    3 RETURN
      END
      SUBROUTINE PLASMA(X1,Y1,Z1,L,M,N,DN,GA,GB,GAMA,
     1 ALPHA,BETA,DLDZ,DELTA,AA,PATH,IO)
      DIMENSION L(3),M(3),N(3),DN(3)
      REAL L,M,N
      DATA CON/1.61E-12/
         ZZ1=Z1
      GA=-GA
      GB=-GB
      IF(Z1 .LT. 0. .OR. Z1 .GE. 48.)GO TO 99
      IF(Z1 .LT. 10.)GO TO 1
      IF(Z1 .LT. 15.)GO TO 2
      IF(Z1 .LT. 48.)GO TO 3
      GO TO 99
    1 TEE=0.01227*Z1**(0.5)+0.0012
      IF(Z1 .GE. 5.)GO TO 11
      CONI=2.039*10.E09
      GO TO 5
   11 CONI=2.223*10.E08
      GO TO 5
    2 TEE=0.012*Z1-0.08
      CONI=1.147*10.E08
      GO TO 5
    3 TEE=0.0224*Z1**(0.8)-0.0958
      IF(Z1 .LT. 20.)GO TO 31
      CONI=6.962*10.E08
      GO TO 5
   31 CONI=1.147*10.E08
    5 CONTINUE
      DEL1=CON*TEE*CONI*TAN(ALPHA)/COS(ALPHA)**2
      DEL=-DEL1*COS(2.*ALPHA)/SIN(ALPHA)
      PATH=PATH+DEL
      P1=DEL1*(GB*M(2)-GA*N(2))/DN(3)
      P2=DEL1*(GB*M(1)-GA*N(1))/DN(3)
      XB=X1-DN(1)/DN(3)*Z1-P1
      YB=Y1-DN(2)/DN(3)*Z1+P2
      C3=(AA*DELTA)**2-(XB*XB+YB*YB)
      C2=2.*DELTA*AA*AA+2.*(XB*DN(1)+YB*DN(2))/DN(3)
      C1=AA*AA-(DN(1)*DN(1)+DN(2)*DN(2))/DN(3)/DN(3)
      C4=C2*C2-4.*C1*C3
      IF(C4 .LT. 0.)GO TO 99
      C5=SQRT(C4)
      CZ1=(C2+C5)/2./C1
      CZ2=(C2-C5)/2./C1
      IF(CZ1 .GT. CZ2)GO TO 10
      Z1=CZ1
      GO TO 15
   10 Z1=CZ2
   15 IF(Z1 .LT. ZZ1)GO TO 99
      X1=XB+DN(1)*Z1/DN(3)
      Y1=YB+DN(2)*Z1/DN(3)
      CALL PNORM(X1,Y1,Z1,DLDZ,DN)
      CALL COMAB(M,N,DN,GA,GB,GAMA)
```

```
      CALL ANGLES(L,DN,GAMA,ALPHA,BETA,O,KICK)
      IF(KICK .GT. 1)GO TO 99
      IO=0
      GO TO 999
   99 IO=1
  999 RETURN
      END
```

379 LINES PRINT

```
      DIMENSION L(3),M(3),N(3),DN(3),
     1 XL1(3),XM1(3),XN1(3),
     2 XL2(3),XM2(3),XN2(3),
     3 XL3(3),XM3(3),XN3(3),
     4 XL4(3),XM4(3),XN4(3),
     5 XL5(3),XM5(3),XN5(3),
     6 XL6(3),XM6(3),XN6(3)
      DIMENSION AMPVAL(4,8)
      COMPLEX VALUE(4,8),SVALUE
      COMPLEX V1,V2,V3,V4,V12,V23,V34,V41
      REAL L,M,N
      REAL KCON
      COMMON VALUE/R/KCON,MAXN/RR/ANEX
      COMMON /RRR/BGL,CGL,L
      COMMON /RRRR/AA,CC,BOUT,BIN
      DATA PI/3.1415927/
      DATA DELTA/1.0/,DELT/0.689/
      REAL SLOTX(4,8)/3.85,3.85,-3.85,-3.85,3.65,3.65,-3.65,-3.65,
     1                2.35,2.35,-2.35,-2.35,2.15,2.15,-2.15,-2.15,
     2                2.35,2.35,-2.35,-2.35,0.85,0.85,-0.85,-0.85,
     3                0.65,0.65,-0.65,-0.65,0.85,0.85,-0.85,-0.85/
      REAL SLOTY(4,8)/0.78,-0.78,-0.78,0.78,2.305,-2.305,-2.305,2.305,
     1                0.78,-0.78,-0.78,0.78,2.305,-2.305,-2.305,2.305,
     2                3.83,-3.83,-3.83,3.83,0.78,-0.78,-0.78,0.78,
     3                2.305,-2.305,-2.305,2.305,3.83,-3.83,-3.83,3.83/
C
C     SAMPLE INPUT DATA
C     &INPUT ANGLEX=15.,ANGLEY=5.,OMEGA=10.,
C     DSTEP=0.4,DA=5.25,MMAX=13,NMAX=37,KPLSMA=0,ANEX=2.5,
C     AA=140.,CC=49.,BOUT=48.,BIN=45.832 &END
C
      NAMELIST /INPUT/ANGLEX,ANGLEY,OMEGA,
     1                DSTEP,DA,MMAX,NMAX,KPLSMA,ANEX,
     2                AA,CC,BOUT,BIN
    1 READ(1,INPUT)
 3003 FORMAT(1H1,'ANGLEX=',F6.2/'ANGLEY=',F6.2/'OMEGA=',F6.2/
     1 'STEP OF DATA=',F6.2/ 'CONSTANT A=',F6.2/
     2 'M(MAX)=',I5/ 'N(MAX)=',I5/)
      WRITE(6,3003)ANGLEX,ANGLEY,OMEGA,DSTEP,DA,MMAX,NMAX
      WRITE(6,3333)AA,CC,BOUT,BIN
 3333 FORMAT('A = ',F10.2/'C = ',F10.2/
     1 'B(OUTER SURFACE)=',F10.2/
     2 'B(INNER SURFACE)=',F10.2)
      IF(KPLSMA .EQ. 0)GO TO 15
      WRITE(6,3004)
 3004 FORMAT('WITH PLASMA')
      GO TO 16
   15 WRITE(6,3005)
 3005 FORMAT('NO PLASMA')
   16 CONTINUE
      V1=(0.,0.)
      V2=(0.,0.)
      V3=(0.,0.)
      V4=(0.,0.)
      BGL=ANGLEX*PI/180.
      CGL=ANGLEY*PI/180.
      ROMEGA=OMEGA*PI/180.
      L(1)=SIN(BGL)
      L(2)=-COS(BGL)*SIN(CGL)
      L(3)=COS(BGL)*COS(CGL)
```

```
      M(1)=COS(BGL)*COS(ROMEGA)
      M(2)=SIN(BGL)*COS(ROMEGA)*SIN(CGL)+SIN(ROMEGA)*COS(CGL)
      M(3)=-SIN(BGL)*COS(ROMEGA)*COS(CGL)+SIN(ROMEGA)*SIN(CGL)
      WRITE(6,3001)L(1),L(2),L(3)
 3001 FORMAT(3X,2HL=F10.3,3X,2HM=F10.3,3X,2HN=F10.3)
      WRITE(6,3002)M(1),M(2),M(3)
 3002 FORMAT(2X,3HL1=F10.3,2X,3HM1=F10.3,2X,3HN1=F10.3//)
      CALL VECTCP(L,M,N)
      KCON=2.*PI/2.46
      INDEX=1
      DO 111 I=1,4
      DO 111 J=1,8
  111 VALUE(I,J)=(0.,0.)
      DELTA=CC-BOUT
      DELTB=CC-BIN
      RADIUS=SQRT(AA*AA+BOUT*BOUT)-AA
      ZEND1=CC-RADIUS*SIN(BGL)
      ZEND2=CC+RADIUS*SIN(BGL)
      SQN2=SQRT(1.-L(3)*L(3))
      MM=2*MMAX+1
      DO 555 IIM=1,MM
      IM=IIM-MMAX-1
      XC1=IM*DSTEP/SQN2
      DO 555 IIN=1,NMAX
      IN=IIN-1
      XC2=43.0+(DA-IN*DSTEP)/SQN2
      XO=-L(1)*L(3)*XC2-XC1*L(2)
      YO=-L(2)*L(3)*XC2+XC1*L(1)
      ZO=-(L(1)*XO+L(2)*YO)/L(3)
      IF(ABS(L(1)) .LE. 0.001)GO TO 3
      ZINT=ZO-XO*L(3)/L(1)
      GO TO 2
    3 IF(ABS(L(2)) .LE. 0.001)GO TO 82
      ZINT=ZO-YO*L(3)/L(2)
    2 IF(ZINT .GT. DELTA .AND. ZINT .LT. DELTB)GO TO 99
      IF(ZINT .GT. DELTA)GO TO 80
      ISIGN=1
      GO TO 81
   82 IF(L(3) .EQ. 1.)GO TO 83
      GO TO 99
   83 IF(XO .EQ. 0. .AND. YO .EQ. 0.)GO TO 99
      IF(XO .GT. 0.)GO TO 87
      GO TO 80
   87 ISIGN=1
      GO TO 81
   80 ISIGN=0
   81 CONTINUE
      IF(ISIGN .EQ. 0)GO TO 60
      ZFIN=ZEND2
      GO TO 61
   60 ZFIN=ZEND1
   61 CONTINUE
C       FIND INTERSECTION POINT
      CALL PINT(XO,YO,ZO,L,X1,Y1,Z1,KIC,ISIGN,1,DLDZ)
      IF(KIC .GT. 1)GO TO 99
      IF(Z1 .GE. ZEND2)GO TO 99
      IF(Z1 .GE. 48.)GO TO 99
C       COMPUTE DISTANCE
      PATH=DIST(XO,YO,ZO,1.,X1,Y1,Z1)
C       FIND NORMAL DIRECTION AT INTERSECT POINT
```

```
      CALL PNORM(X1,Y1,Z1,DLDZ,DN)
C       FIND CONSTANTS A $ B AND GAMA
      CALL CONAB(M,N,DN,GA,GB,GAMA)
C       FIND ANGLES ALPHA AND BETA
      CALL ANGLES(L,DN,GAMA,ALPHA,BETA,0,KICK)
      IF(KICK .GT. 1)GO TO 99
      IF(KPLSMA .EQ. 1)GO TO 62
      GO TO 63
   62 CALL PLASMA(X1,Y1,Z1,L,M,N,DN,GA,GB,GAMA,
     1 ALPHA,BETA,PATH,DLDZ,IQ)
      IF (IQ .EQ. 0)GO TO 63
      GO TO 99
C       RAY WILL PASS FROM MEDIUM 1 TO 2
C       COMPUTE T12 AND R12
   63 CALL TR12(ALPHA,BETA,R12P,T12P,R12V,T12V)
      GAMAB=SQRT((GA*T12P)**2+(GB*T12V)**2)
      AMP=GAMAB
      CALL NEWCOR(L,M,N,GA,GB,XL1,XM1,XN1)
C       COMPUTE DIRECTION COSINES OF TRANSMITTED RAY
      CALL TRANSM(GA,GB,ALPHA,BETA,T12P,T12V,
     1 XL1,XM1,XN1,GAMAB,XL2,XM2,XN2)
C       RAY WILL PASS THE INNER SURFACE
C       COMPUTE INNER INTERSECT POINT
      CALL PINT(X1,Y1,Z1,XL2,X2,Y2,Z2,KIC,0,0,DLDZ)
      IF(KIC .GT. 1)GO TO 99
      PATH=PATH+DIST(X1,Y1,Z1,ANEX,X2,Y2,Z2)
      PAT=PATH
      DD=PATH
      CALL PNORM(X2,Y2,Z2,DLDZ,DN)
      CALL CONAB(XM2,XN2,DN,GA,GB,GAMA)
      CALL ANGLES(XL2,DN,GAMA,BETA,ALPHA,1,KICK)
      IF(KICK .GT. 1)GO TO 99
C       COMPUTE T21 AND R21
      CALL TR21(ALPHA,BETA,R21P,T21P,R21V,T21V)
      GAMAB=SQRT((GA*T21P)**2+(GB*T21V)**2)
      GAMABB=SQRT((GA*R21P)**2+(GB*R21V)**2)
      AMPT=AMP*GAMAB
      AMPR=AMP*GAMABB
      TAMP=AMPT
      CALL NEWCOR(XL2,XM2,XN2,GA,GB,XL3,XM3,XN3)
C       COMPUTE DIRECTION COSINES OF TRANSMITTED RAY
      CALL TRANSM(GA,GB,BETA,ALPHA,T21P,T21V,XL3,XM3,XN3,
     1 GAMAB,XL4,XM4,XN4)
      JJ=1
      CALL RECORD(L,XL4,X2,Y2,Z2,X3,Y3,Z3,DDT,K)
      IF(K .EQ. 1)GO TO 30
      IF(Z3 .LT. ZFIN)GO TO 31
      GO TO 32
   30 DDT=DDT+DD
      CALL APSLOT(AMPT,X3,Y3,XL4,XM4,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
      IF(JJJ .EQ. 9)GO TO 51
      AKCON=KCON*DDT
      CSKD=COS(AKCON)
      SNKD=SIN(AKCON)
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   51 CONTINUE
      GO TO 32
   31 CALL SECOND(XL4,XM4,XN4,X2,Y2,Z2,    INDEX,
     1 DELTA,DELTB,PAT,TAMP,L,ISIGN,ZEND1,ZEND2)
```

```
   32 CALL REFLCT(GA,GB,BETA,R21P,R21V,XL3,XM3,XN3,GAMABB,
     1 XL5,XM5,XN5)
   40 JJ=JJ+1
      CALL REPEAT(INDEX,DELTA,DELTB,    X2,Y2,Z2,XL5,XM5,XN5,PATH,
     1 AMPT,AMPR,XL4,XM4,XN4,X3,Y3,Z3,DD,K,L,3)
      RAMP=AMPR
      TAMP=AMPT
      PAT=PATH
      IF(K .EQ. 1)GO TO 41
      IF(K .EQ. 0)GO TO 99
      IF(Z3 .LT. ZFIN)GO TO 44
      GO TO 42
   44 CALL SECOND(XL4,XM4,XN4,X2,Y2,Z2,    INDEX, DELTA,DELTB,PAT,TAMP,L,
     1 ISIGN,ZEND1,ZEND2)
      GO TO 42
   41 DDT=DD+PATH
      CALL APSLOT(AMPT,X3,Y3,XL4,XM4,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
      IF(JJJ .EQ. 9)GO TO 91
      AKCON=KCON*DDT
      SNKD=SIN(AKCON)
      CSKD=COS(AKCON)
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   91 CONTINUE
   42 IF(ABS(TAMP) .LE. 0.01 )GO TO 99
      IF(JJ .LT. 20)GO TO 40
   99 CONTINUE
  555 CONTINUE
      WRITE(6,9001)
 9001 FORMAT(1H1,2X,4HSLOT,3X,11X,4HREAL,6X,9HIMAGINARY,
     1 6X,9HAMPLITUDE/)
      DO 432 I=1,4
      DO 432 J=1,8
  432 AMPVAL(I,J)=CABS(VALUE(I,J))
      DO 431 I=1,4
      DO 431 J=1,8
      WRITE(6,9000)I,J,VALUE(I,J),AMPVAL(I,J)
 9000 FORMAT(2I5,3E15.5)
  431 CONTINUE
      DO 433 I=1,8
      V1=V1+VALUE(1,I)
      V2=V2+VALUE(2,I)
      V3=V3+VALUE(3,I)
      V4=V4+VALUE(4,I)
  433 CONTINUE
      V1A=CABS(V1)
      V2A=CABS(V2)
      V3A=CABS(V3)
      V4A=CABS(V4)
      PHI1=ATAN2(AIMAG(V1),REAL(V1))
      PHI2=ATAN2(AIMAG(V2),REAL(V2))
      PHI3=ATAN2(AIMAG(V3),REAL(V3))
      PHI4=ATAN2(AIMAG(V4),REAL(V4))
      V12=V1+V2
      V23=V2+V3
      V34=V3+V4
      V41=V4+V1
      V12A=CABS(V12)
      V23A=CABS(V23)
      V34A=CABS(V34)
```

```
      V41A=CABS(V41)
      PHI12=ATAN2(AIMAG(V12),REAL(V12))
      PHI23=ATAN2(AIMAG(V23),REAL(V23))
      PHI34=ATAN2(AIMAG(V34),REAL(V34))
      PHI41=ATAN2(AIMAG(V41),REAL(V41))
      WRITE(6,9004)
 9004 FORMAT(1X//10X,'(V1,V2,V3,V4)')
      WRITE(6,9006)
 9006 FORMAT(11X,4HREAL,6X,9HIMAGINARY,5X,10HABS. VALUE,10X,5HPHASE)
      WRITE(6,9002) V1,V1A,PHI1,
     1              V2,V2A,PHI2,
     2              V3,V3A,PHI3,
     3 V4,V4A,PHI4
      WRITE(6,9005)
 9005 FORMAT(1X//10X,'(V12,V23,V34,V41)')
      WRITE(6,9002)V12,V12A,PHI12,
     1   V23,V23A,PHI23,
     2   V34,V34A,PHI34,
     3   V41,V41A,PHI41
      VVV=1./4./PI
      VV1=VVV/0.8698*(PHI34-PHI12)
      VV2=VVV/0.8810*(PHI41-PHI23)
      ERRORV=ARSIN(VV1)
      ERRORH=ARSIN(VV2)
      WRITE(6,9003)ERRORV,ERRORH
 9003 FORMAT('***','ERRORV=',E12.5,5X,'ERRORH=',E12.5)
 9002 FORMAT(4E15.5)
      GO TO 1
      END
```

```
      SUBROUTINE SECOND(XL,XM,XN,X2,Y2,Z2,INDEX,
     1 DELTA,DELTB,PATH,BAMP,L,ISIGN,ZEND1,ZEND2,
     2SLOTX,SLOTY)
      DIMENSION SLOTX(4,8),SLOTY(4,8)
      DIMENSION DN(3),XL(3),XM(3),XN(3),
     1XL1(3),XM1(3),XN1(3),
     2XL2(3),XM2(3),XN2(3),
     3XL3(3),XM3(3),XN3(3),
     4XL4(3),XM4(3),XN4(3)
     5,XL7(3),XM7(3),XN7(3)
      DIMENSION L(3)
      COMPLEX VALUE(4,8),SVALUE
      REAL L
      REAL KCON
      COMMON VALUE/R/KCON,MAXN/RR/ANEX
      COMMON /RRRR/AA,CC,BOUT,BIN
      JJ=1
      IF(ISIGN .EQ. 0)GO TO 60
      ZFIN=ZEND1
      GO TO 61
   60 ZFIN=ZEND2
   61 CONTINUE
      CALL PINT(X2,Y2,Z2,XL,X5,Y5,Z5,KIC,1,0,DLDZ)
      IF(KIC .GT. 1)GO TO 17
      PATH=PATH+DIST(X2,Y2,Z2,1.,X5,Y5,Z5)
      CALL PNORM(X5,Y5,Z5,DLDZ,DN)
      DO 10 I=1,3
   10 DN(I)=-DN(I)
      CALL CONAB(XM,XN,DN,GA,GB,GAMA)
      CALL NEWCOR(XL,XM,XN,GA,GB,XL1,XM1,XN1)
      CALL ANGLES(XL,DN,GAMA,ALPHA,BETA,0,KICK)
      IF(KICK .GT. 1)GO TO 17
      CALL TR12(ALPHA,BETA,R12P,T12P,R12V,T12V)
      GAMABB=SQRT((GA*R12P)**2+(GB*R12V)**2)
      GAMAB=SQRT((GA*T12P)**2+(GB*T12V)**2)
      AMPT=BAMP*GAMAB
      AMPR=BAMP*GAMABB
      CALL REFLCT(GA,GB,ALPHA,R12P,R12V,XL1,XM1,XN1,
     1 GAMABB,XL2,XM2,XN2)
      CALL RECORD(L,XL2,X5,Y5,Z5,X6,Y6,Z6,DD,K)
      IF(K .EQ. 1)GO TO 11
      IF(Z6 .LT. ZFIN)GO TO 30
      GO TO 12
   30 CALL THIRD(XL2,XM2,XN2,X5,Y5,Z5,   INDEX,DELTA,DELTB,
     1 PATH,AMPR,L,ISIGN,ZEND1,ZEND2,SLOTX,SLOTY)
      GO TO 12
   11 DD=DD+PATH
      CALL APSLOT(AMPT,X6,Y6,XL2,XM2,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
      IF(JJJ .EQ. 9)GO TO 51
      AKCON=KCON*DD
      CSKD=COS(AKCON)
      SNKD=SIN(AKCON)
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   51 CONTINUE
   12 CALL TRANSM(GA,GB,ALPHA,BETA,T12P,T12V,XL1,XM1,XN1,
     1 GAMAB,XL3,XM3,XN3)
   20 JJ=JJ+1
      CALL REPEAT(INDEX,DELTA,DELTB,    X5,Y5,Z5,XL3,XM3,XN3,PATH,
     1 AMPT,AMPR,XL7,XM7,XN7,X7,Y7,Z7,DD,KK,L,JJ)
```

```
      IF(KK .EQ. 1)GO TO 13
      IF(KK .EQ. 0)GO TO 17
      IF(Z7 .LT. ZFIN)GO TO 31
      GO TO 14
   31 CALL THIRD(XL7,XM7,XN7,X5,Y5,Z5,    INDEX,DELTA,DELTB,
     1 PATH,AMPT,L,ISIGN,ZEND1,ZEND2,SLOTX,SLOTY)
      GO TO 14
   13 DD=DD+PATH
      IF(ABS(AMPT) .LT. 0.01 )GO TO 17
      CALL APSLOT(AMPT,X7,Y7,XL7,XM7,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
      IF(JJJ .EQ. 9)GO TO 91
      AKCON=KCON*DD
      CSKD=COS(AKCON)
      SNKD=SIN(AKCON)
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   91 CONTINUE
   14 IF(ABS(AMPT) .LE. 0.01 )GO TO 17
      IF(JJ .GE. 20)GO TO 17
      GO TO 20
   17 RETURN
      END
      SUBROUTINE THIRD(XL,XM,XN,X2,Y2,Z2,INDEX,
     1 DELTA,DELTB,PATH,BAMP,L,ISIGN,ZEND1,ZEND2,
     2SLOTX,SLOTY)
      DIMENSION SLOTX(4,8),SLOTY(4,8)
      DIMENSION DN(3),XL(3),XM(3),XN(3),
     1XL1(3),XM1(3),XN1(3),
     2XL2(3),XM2(3),XN2(3),
     3XL3(3),XM3(3),XN3(3),
     4XL4(3),XM4(3),XN4(3)
     5,XL7(3),XM7(3),XN7(3)
      DIMENSION L(3)
      COMPLEX VALUE(4,8),SVALUE
      REAL L
      REAL KCON
      COMMON VALUE/R/KCON,MAXN/RR/ANEX
      COMMON /RRRR/AA,CC,BOUT,BIN
      JJ=1
      IF(ISIGN .EQ. 0)GO TO 40
      ZFIN=ZEND2
      GO TO 41
   40 ZFIN=ZEND1
   41 CONTINUE
      CALL PINT(X2,Y2,Z2,XL,X5,Y5,Z5,KIC,1,0,DLDZ)
      IF(KIC .GT. 1)GO TO 17
      PATH=PATH+DIST(X2,Y2,Z2,1.,X5,Y5,Z5)
      CALL PNORM(X5,Y5,Z5,DLDZ,DN)
      DO 10 I=1,3
   10 DN(I)=-DN(I)
      CALL CONAB(XM,XN,DN,GA,GB,GAMA)
      CALL NEWCOR(XL,XM,XN,GA,GB,XL1,XM1,XN1)
      CALL ANGLES(XL,DN,GAMA,ALPHA,BETA,0,KICK)
      IF(KICK .GT. 1)GO TO 17
      CALL TR12(ALPHA,BETA,R12P,T12P,R12V,T12V)
      GAMABB=SQRT((GA*R12P)**2+(GB*R12V)**2)
      GAMAB=SQRT((GA*T12P)**2+(GB*T12V)**2)
      AMPT=BAMP*GAMAB
      AMPR=BAMP*GAMABB
      CALL REFLCT(GA,GB,ALPHA,R12P,R12V,XL1,XM1,XN1,
```

```
  1 GAMABB,XL2,XM2,XN2)
   CALL RECORD(L,XL2,X5,Y5,Z5,X6,Y6,Z6,DD,K)
   IF(K .EQ. 1)GO TO 11
   IF(Z6 .LT. ZFIN)GO TO 30
   GO TO 12
30 CALL FOURTH(XL2,XM2,XN2,X5,Y5,Z5,    INDEX,DELTA,DELTB,PATH,
  1 AMPR,L,ISIGN,ZEND1,ZEND2,SLOTX,SLOTY)
   GO TO 12
11 DD=DD+PATH
   CALL APSLOT(AMPT,X6,Y6,XL2,XM2,
  1 SLOTX,SLOTY,III,JJJ,AMPSL)
   IF(JJJ .EQ. 9)GO TO 91
   AKCON=KCON*DD
   CSKD=COS(AKCON)
   SNKD=SIN(AKCON)
   AMPRL=AMPSL*CSKD
   AMPIM=AMPSL*SNKD
   SVALUE=CMPLX(AMPRL,AMPIM)
   VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
91 CONTINUE
12 CALL TRANSM(GA,GB,ALPHA,BETA,T12P,T12V,XL1,XM1,XN1,
  1 GAMAB,XL3,XM3,XN3)
20 JJ=JJ+1
   CALL REPEAT(INDEX,DELTA,DELTB,    X5,Y5,Z5,XL3,XM3,XN3,PATH,
  1 AMPT,AMPR,XL7,XM7,XN7,X7,Y7,Z7,DD,KK,L,JJ)
   IF(KK .EQ. 1)GO TO 13
   IF(KK .EQ. 0)GO TO 17
   IF(Z7 .LT. ZFIN)GO TO 31
   GO TO 14
31 CALL FOURTH(XL7,XM7,XN7,X5,Y5,Z5,    INDEX,DELTA,DELTB,PATH,
  1 AMPT,L,ISIGN,ZEND1,ZEND2,SLOTX,SLOTY)
   GO TO 14
13 DD=DD+PATH
   IF(ABS(AMPT) .LT. 0.01 )GO TO 17
   CALL APSLOT(AMPT,X7,Y7,XL7,XM7,
  1 SLOTX,SLOTY,III,JJJ,AMPSL)
   IF(JJJ .EQ. 9)GO TO 51
   AKCON=KCON*DD
   CSKD=COS(AKCON)
   SNKD=SIN(AKCON)
   AMPRL=AMPSL*CSKD
   AMPIM=AMPSL*SNKD
   SVALUE=CMPLX(AMPRL,AMPIM)
   VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
51 CONTINUE
14 IF(ABS(AMPT) .LE. 0.01 )GO TO 17
   IF(JJ .GE. 20)GO TO 17
   GO TO 20
17 RETURN
   END
```

```
      SUBROUTINE FOURTH(XL,XM,XN,X2,Y2,Z2,    INDEX,
     1 DELTA,DELTB,PATH,BAMP,L,ISIGN,ZEND1,ZEND2,
     2SLOTX,SLOTY)
      DIMENSION SLOTX(4,8),SLOTY(4,8)
      DIMENSION DN(3),XL(3),XM(3),XN(3),
     1XL1(3),XM1(3),XN1(3),
     2XL2(3),XM2(3),XN2(3),
     3XL3(3),XM3(3),XN3(3),
     4XL4(3),XM4(3),XN4(3)
     5,XL7(3),XM7(3),XN7(3)
      DIMENSION L(3)
      COMPLEX VALUE(4,8),SVALUE
      REAL L
      REAL KCON
      COMMON VALUE/R/KCON,MAXN/RR/ANEX
      COMMON /RRRR/AA,CC,BOUT,BIN
      JJ=1
      IF(ISIGN .EQ. 0)GO TO 60
      ZFIN=ZEND1
      GO TO 61
   60 ZFIN=ZEND2
   61 CONTINUE
      CALL PINT(X2,Y2,Z2,XL,X5,Y5,Z5,KIC,1,0,DLDZ)
      IF(KIC .GT. 1)GO TO 17
      PATH=PATH+DIST(X2,Y2,Z2,1.,X5,Y5,Z5)
      CALL PNORM(X5,Y5,Z5,DLDZ,DN)
      DO 10 I=1,3
   10 DN(I)=-DN(I)
      CALL CONAB(XM,XN,DN,GA,GB,GAMA)
      CALL NEWCOR(XL,XM,XN,GA,GB,XL1,XM1,XN1)
      CALL ANGLES(XL,DN,GAMA,ALPHA,BETA,0,KICK)
      IF(KICK .GT. 1)GO TO 17
      CALL TR12(ALPHA,BETA,R12P,T12P,R12V,T12V)
      GAMABB=SQRT((GA*R12P)**2+(GB*R12V)**2)
      GAMAB=SQRT((GA*T12P)**2+(GB*T12V)**2)
      AMPT=BAMP*GAMAB
      AMPR=BAMP*GAMABB
      CALL REFLCT(GA,GB,ALPHA,R12P,R12V,XL1,XM1,XN1,
     1 GAMABB,XL2,XM2,XN2)
      CALL RECORD(L,XL2,X5,Y5,Z5,X6,Y6,Z6,DD,K)
      IF(K .EQ. 1)GO TO 11
      IF(Z6 .LT. ZFIN)GO TO 30
      GO TO 12
   30 CALL FIFTH(XL2,XM2,XN2,X5,Y5,Z5,    INDEX,DELTA,DELTB,
     1 PATH,AMPR,L,ISIGN,ZEND1,ZEND2,SLOTX,SLOTY)
      GO TO 12
   11 DD=DD+PATH
      CALL APSLOT(AMPT,X6,Y6,XL2,XM2,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
      IF(JJJ .EQ. 9)GO TO 51
      AKCON=KCON*DD
      CSKD=COS(AKCON)
      SNKD=SIN(AKCON)
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   51 CONTINUE
   12 CALL TRANSM(GA,GB,ALPHA,BETA,T12P,T12V,XL1,XM1,XN1,
     1 GAMAB,XL3,XM3,XN3)
   20 JJ=JJ+1
      CALL REPEAT(INDEX,DELTA,DELTB,    X5,Y5,Z5,XL3,XM3,XN3,PATH,
     1 AMPT,AMPR,XL7,XM7,XN7,X7,Y7,Z7,DD,KK,L,JJ)
```

```
      IF(KK .EQ. 1)GO TO 13
      IF(KK .EQ. 0)GO TO 17
      IF(Z7 .LT. ZFIN)GO TO 31
      GO TO 14
   31 CALL FIFTH(XL7,XM7,XN7,X5,Y5,Z5,    INDEX,DELTA,DELTB,
     1 PATH,AMPT,L,ISIGN,ZEND1,ZEND2,SLOTX,SLOTY)
      GO TO 14
   13 DD=DD+PATH
      IF(ABS(AMPT) .LT. 0.01 )GO TO 17
      CALL APSLOT(AMPT,X7,Y7,XL7,XM7,
     1 SLOTX,SLOTY,III,JJJ,AMPSL)
      IF(JJJ .EQ. 9)GO TO 91
      AKCON=KCON*DD
      CSKD=COS(AKCON)
      SNKD=SIN(AKCON)
      AMPRL=AMPSL*CSKD
      AMPIM=AMPSL*SNKD
      SVALUE=CMPLX(AMPRL,AMPIM)
      VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
   91 CONTINUE
   14 IF(ABS(AMPT) .LE. 0.01 )GO TO 17
      IF(JJ .GE. 20)GO TO 17
      GO TO 20
   17 RETURN
      END
      SUBROUTINE FIFTH(XL,XM,XN,X2,Y2,Z2,    INDEX,
     1 DELTA,DELTB,PATH,BAMP,L,ISIGN,ZEND1,ZEND2,
     2SLOTX,SLOTY)
      DIMENSION SLOTX(4,8),SLOTY(4,8)
      DIMENSION DN(3),XL(3),XM(3),XN(3),
     1XL1(3),XM1(3),XN1(3),
     2XL2(3),XM2(3),XN2(3),
     3XL3(3),XM3(3),XN3(3),
     4XL4(3),XM4(3),XN4(3)
     5,XL7(3),XM7(3),XN7(3)
      DIMENSION L(3)
      COMPLEX VALUE(4,8),SVALUE
      REAL L
      REAL KCON
      COMMON VALUE/R/KCON,MAXN/RR/ANEX
      COMMON /RRRR/AA,CC,BOUT,BIN
      JJ=1
      CALL PINT(X2,Y2,Z2,XL,X5,Y5,Z5,KIC,1,0,DLDZ)
      IF(KIC .GT. 1)GO TO 17
      PATH=PATH+DIST(X2,Y2,Z2,1.,X5,Y5,Z5)
      CALL PNORM(X5,Y5,Z5,DLDZ,DN)
      DO 10 I=1,3
   10 DN(I)=-DN(I)
      CALL CONAB(XM,XN,DN,GA,GB,GAMA)
      CALL NEWCOR(XL,XM,XN,GA,GB,XL1,XM1,XN1)
      CALL ANGLES(XL,DN,GAMA,ALPHA,BETA,0,KICK)
      IF(KICK .GT. 1)GO TO 17
      CALL TR12(ALPHA,BETA,R12P,T12P,R12V,T12V)
      GAMABB=SQRT((GA*R12P)**2+(GB*R12V)**2)
      GAMAB=SQRT((GA*T12P)**2+(GB*T12V)**2)
      AMPT=BAMP*GAMAB
      AMPR=BAMP*GAMABB
      IF(AMPR .LE. 0.01 )GO TO 17
      CALL REFLCT(GA,GB,ALPHA,R12P,R12V,XL1,XM1,XN1,
     1 GAMABB,XL2,XM2,XN2)
      CALL RECORD(L,XL2,X5,Y5,Z5,X6,Y6,Z6,DD,K)
      IF(K .EQ. 1)GO TO 11
      GO TO 12
```

```
11 DD=DD+PATH
   CALL APSLOT(AMPT,X6,Y6,XL2,XM2,
  1 SLOTX,SLOTY,III,JJJ,AMPSL)
   IF(JJJ .EQ. 9)GO TO 91
   AKCON=KCON*DD
   CSKD=COS(AKCON)
   SNKD=SIN(AKCON)
   AMPRL=AMPSL*CSKD
   AMPIM=AMPSL*SNKD
   SVALUE=CMPLX(AMPRL,AMPIM)
   VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
91 CONTINUE
12 CALL TRANSM(GA,GB,ALPHA,BETA,T12P,T12V,XL1,XM1,XN1,
  1 GAMAB,XL3,XM3,XN3)
20 JJ=JJ+1
   CALL REPEAT(INDEX,DELTA,DELTB,   X5,Y5,Z5,XL3,XM3,XN3,PATH,
  1 AMPT,AMPR,XL7,XM7,XN7,X7,Y7,Z7,DD,KK,L,JJ)
   IF(KK .EQ. 1)GO TO 13
   IF(KK .EQ. 0)GO TO 17
   GO TO 14
13 DD=DD+PATH
   IF(ABS(AMPT) .LT. 0.01 )GO TO 17
   CALL APSLOT(AMPT,X7,Y7,XL7,XM7,
  1 SLOTX,SLOTY,III,JJJ,AMPSL)
   IF(JJJ .EQ. 9)GO TO 51
   AKCON=KCON*DD
   CSKD=COS(AKCON)
   SNKD=SIN(AKCON)
   AMPRL=AMPSL*CSKD
   AMPIM=AMPSL*SNKD
   SVALUE=CMPLX(AMPRL,AMPIM)
   VALUE(III,JJJ)=VALUE(III,JJJ)+SVALUE
51 CONTINUE
14 IF(ABS(AMPT) .LE. 0.01 )GO TO 17
   IF(JJ .GE. 10)GO TO 17
   GO TO 20
17 RETURN
   END
```

```
      SUBROUTINE REPEAT(INDEX,DELTA,DELTB,    X2,Y2,Z2
     1 ,XL5,XM5,XN5,PATH,AMPT,AMPR,XL4,XM4,XN4,
     2 X3,Y3,Z3,DD,K,L,JJ)
      DIMENSION DN(3),L(3),
     1XL2(3),XM2(3),XN2(3),
     2XL3(3),XM3(3),XN3(3),
     3XL4(3),XM4(3),XN4(3),
     4XL5(3),XM5(3),XN5(3),
     5XL6(3),XM6(3),XN6(3)
      REAL L
      COMMON /RR/ANEX
      COMMON /RRRR/AA,CC,BOUT,BIN
      CALL PINT(X2,Y2,Z2,XL5,X1,Y1,Z1,KIC,1,1,DLDZ)
      IF(KIC .GT. 1)GO TO 10
      PATH=PATH+DIST(X2,Y2,Z2,ANEX,X1,Y1,Z1)
      CALL PNORM(X1,Y1,Z1,DLDZ,DN)
      DO 48 I=1,3
   48 DN(I)=-DN(I)
      CALL CONAB(XM5,XN5,DN,GA,GB,GAMA)
      CALL NEWCOR(XL5,XM5,XN5,GA,GB,XL6,XM6,XN6)
      CALL ANGLES(XL5,DN,GAMA,BETA,ALPHA,1,KICK)
      IF(KICK .GT. 1)GO TO 10
      CALL TR21(ALPHA,BETA,R21P,T21P,R21V,T21V)
      GAMABB=SQRT((GA*R21P)**2+(GB*R21V)**2)
      IF(JJ .LE. 2)GO TO 11
      AMP=AMPR*GAMABB
      GO TO 12
   11 AMP=AMPT*GAMABB
   12 CONTINUE
      CALL REFLCT(GA,GB,BETA,R21P,R21V,XL6,XM6,XN6,
     1 GAMABB,XL2,XM2,XN2)
      CALL PINT(X1,Y1,Z1,XL2,X2,Y2,Z2,KIC,0,0,DLDZ)
      IF(KIC .GT. 1)GO TO 10
      PATH=PATH+DIST(X1,Y1,Z1,ANEX,X2,Y2,Z2)
      CALL PNORM(X2,Y2,Z2,DLDZ,DN)
      CALL CONAB(XM2,XN2,DN,GA,GB,GAMA)
      CALL ANGLES(XL2,DN,GAMA,BETA,ALPHA,1,KICK)
      IF(KICK .GT. 1)GO TO 10
      CALL TR21(ALPHA,BETA,R21P,T21P,R21V,T21V)
      GAMA=SQRT((GA*T21P)**2+(GB*T21V)**2)
      GAMAB=SQRT((GA*R21P)**2+(GB*R21V)**2)
      AMPT=AMP*GAMA
      AMPR=AMP*GAMAB
      CALL NEWCOR(XL2,XM2,XN2,GA,GB,XL3,XM3,XN3)
      CALL TRANSM(GA,GB,BETA,ALPHA,T21P,T21V,XL3,XM3,XN3,
     1 GAMA,XL4,XM4,XN4)
      CALL REFLCT(GA,GB,BETA,R21P,R21V,XL3,XM3,XN3,
     1 GAMAB,XL5,XM5,XN5)
      CALL RECORD(L,XL4,X2,Y2,Z2,X3,Y3,Z3,DD,K)
      RETURN
   10 K=0
      RETURN
      END
      SUBROUTINE PNORM(X1,Y1,Z1,DLDZ,DN)
      DIMENSION DN(3)
      P1=SQRT(1+DLDZ*DLDZ)
      P2=SQRT(X1*X1+Y1*Y1)
      DN(1)=X1/P1/P2
      DN(2)=Y1/P1/P2
      DN(3)=-DLDZ/P1
      RETURN
      END
      SUBROUTINE CONAB(M,N,DN,A,B,GAMA)
```

```
      DIMENSION M(3),N(3),DN(3)
      REAL M,N
      G1=VECTDT(DN,M)
      G2=VECTDT(DN,N)
      G3=SQRT(G1*G1+G2*G2)
      GAMA=-1./G3
      A=-GAMA*G1
      B=-GAMA*G2
      RETURN
      END
      SUBROUTINE ANGLES(L,DN,GAMA,ALPHA,BETA,II,KICK)
      DIMENSION L(3),DN(3)
      COMMON /RR/ANEX
      REAL L
      PI=3.1415927
      ARG1=-VECTDT(L,DN)
      ARG2=-1./GAMA
      IF(ABS(ARG2) .GT. 1)GO TO 10
      GO TO 11
   10 KICK=2
      GO TO 23
   11 IF(II .EQ. 1)GO TO 2
      ARG3=ARG2/ANEX
      GO TO 3
    2 ARG3=ARG2*ANEX
      IF(ABS(ARG3) .GE. 1.)GO TO 4
      GO TO 3
    4 KICK=2
      GO TO 23
    3 CONTINUE
      IF(ARG1 .GE. 0. .AND. ARG2 .GE. 0.) GO TO 21
      KICK=2
      GO TO 23
   21 ALPHA=ARSIN(ARG2)
      BETA=ARSIN(ARG3)
      KICK=0
   23 RETURN
      END
      SUBROUTINE TR12(ALPHA,BETA,RP,TP,RV,TV)
      COMMON /RR/ANEX
      ALBT=ALPHA-BETA
      GX=ANEX*COS(ALPHA)/COS(BETA)
      RP=(1.-GX)/(1.+GX)
      TP=2./(1+GX)
      GY=ANEX*COS(BETA)/COS(ALPHA)
      RV=(1.-GY)/(1.+GY)
      TV=2./(1.+GY)
      RETURN
      END
      SUBROUTINE TR21(ALPHA,BETA,RP,TP,RV,TV)
      COMMON /RR/ANEX
      ALBT=ALPHA-BETA
      GX=ANEX*COS(ALPHA)/COS(BETA)
      GY=ANEX*COS(BETA)/COS(ALPHA)
      RP=-(1.-GX)/(1.+GX)
      TP=2.*GX/(1.+GX)
      RV=-(1.-GY)/(1.+GY)
      TV=2.*GY/(1.+GY)
      RETURN
      END
      SUBROUTINE NEWCOR(L,M,N,A,B,XL,XM,XN)
      DIMENSION L(3),M(3),N(3),XL(3),XM(3),XN(3)
      REAL L,M,N
```

```
      DO 30 I=1,3
      XL(I)=L(I)
      XM(I)=A*M(I)+B*N(I)
   30 XN(I)=-B*M(I)+A*N(I)
      RETURN
      END
      SUBROUTINE TRANSM(A,B,ALPHA,BETA,TP,TV,L,M,N,GAMA,XL,XM,XN)
      DIMENSION L(3),M(3),N(3),XL(3),XM(3),XN(3)
      DIMENSION YL(3),YM(3),YN(3)
      REAL L,M,N
      ALBT=ALPHA-BETA
      DO 32 I=1,3
      YL(I)=COS(ALBT)*L(I)-SIN(ALBT)*M(I)
      YM(I)=SIN(ALBT)*L(I)+COS(ALBT)*M(I)
   32 YN(I)=N(I)
      DO 33 I=1,3
      XL(I)=YL(I)
   33 XM(I)=(A*TP*YM(I)-B*TV*YN(I))/GAMA
      CALL VECTCP(XL,XM,XN)
      RETURN
      END
      SUBROUTINE REFLCT(A,B,ALPHA,RP,RV,L,M,N,GAMA,XL,XM,XN)
      DIMENSION L(3),M(3),N(3),XL(3),XM(3),XN(3)
      DIMENSION YL(3),YM(3),YN(3)
      REAL L,M,N
      ALPH2=ALPHA*2.
      DO 32 I=1,3
      YL(I)=-COS(ALPH2)*L(I)+SIN(ALPH2)*M(I)
      YM(I)=-SIN(ALPH2)*L(I)-COS(ALPH2)*M(I)
   32 YN(I)=N(I)
      DO 33 I=1,3
      XL(I)=YL(I)
   33 XM(I)=(-A*RP*YM(I)-B*RV*YN(I))/GAMA
      CALL VECTCP(XL,XM,XN)
      RETURN
      END
      SUBROUTINE RECORD(L,XL,X2,Y2,Z2,X,Y,Z,DD,K)
      DIMENSION L(3),XL(3)
      REAL L
      C1=L(1)*XL(1)/XL(3)+L(2)*XL(2)/XL(3)
      C2=C1*Z2
      C3=-L(1)*X2-L(2)*Y2+L(3)*43.
      C4=C2+C3
      C5=C1+L(3)
      Z=C4/C5
      X=X2+XL(1)/XL(3)*(Z-Z2)
      Y=Y2+XL(2)/XL(3)*(Z-Z2)
      CC=X*X+Y*Y+(Z-43.)**2
      DD=DIST(X2,Y2,Z2,1.0,X,Y,Z)
      IF(CC .LE. 30.25)GO TO 2
      K=3
      GO TO 3
    2 K=1
    3 RETURN
      END
      SUBROUTINE VECTCP(L,M,N)
      DIMENSION L(3),M(3),N(3)
      REAL L,M,N
      N(1)=L(2)*M(3)-L(3)*M(2)
      N(2)=L(3)*M(1)-L(1)*M(3)
```

```
      N(3)=L(1)*M(2)-L(2)*M(1)
      RETURN
      END
      FUNCTION VECTDT(L,M)
      DIMENSION L(3),M(3),N(3)
      REAL L,M,N
      DO 1 I=1,3
    1 N(I)=L(I)*M(I)
      VECTDT=N(1)+N(2)+N(3)
      RETURN
      END
      SUBROUTINE SLOTS(XX,YY,BGL,III,JJJ,CGL)
      IF(XX .GT. 0.) GO TO 2
      IF(YY .GT. 0.)GO TO 3
      III=3
      GO TO 5
    3 III=4
      GO TO 5
    2 IF(YY .GT. 0.)GO TO 4
      III=2
      GO TO 5
    4 III=1
    5 CONTINUE
      CCC=COS(BGL)
      DDD=COS(CGL)
      X3=4.5*CCC
      X2=3.0*CCC
      X1=1.5*CCC
      Y1=1.5*DDD
      Y2=3.0*DDD
      Y3=4.5*DDD
      X=ABS(XX)
      Y=ABS(YY)
      IF(X .LT. X3 .AND. Y .LT. Y3)GO TO 10
      GO TO 30
   10 IF(X .LT. X2 .AND. Y .LT. Y2)GO TO 11
      IF(X .GE. X2 .AND. Y .GE. Y2)GO TO 30
      IF(X .GE. X2 .AND. Y .LE. Y2)GO TO 12
      IF(X .GT. X1)GO TO 13
      JJJ=8
      GO TO 20
   13 JJJ=5
      GO TO 20
   12 IF(Y .GT. Y1)GO TO 14
      JJJ=1
      GO TO 20
   14 JJJ=2
      GO TO 20
   11 IF(X .LE. X1 .AND. Y .LE. Y1)GO TO 15
      IF(X .GE. X1 .AND. Y .GE. Y1)GO TO 16
      IF(X .GE. X1)GO TO 17
      JJJ=7
      GO TO 20
   17 JJJ=3
      GO TO 20
   16 JJJ=4
      GO TO 20
   15 JJJ=6
   20 RETURN
   30 JJJ=9
```

```
      RETURN
      END
      SUBROUTINE APSLOT(AMP,X,Y,XL,XM,SLOTX,SLOTY,III,JJJ,AMPSL)
      DIMENSION SLOTX(4,8),SLOTY(4,8),XL(3),XM(3)
      DIMENSION L(3)
      COMMON /RRR/BGL,CGL,L
      REAL L
      PI=3.1415927
      H=1.06
      W=1.060
      HH=H*COS(BGL)
      WW=W*COS(CGL)
      HO2=HH/2.
      WO2=WW/2.
      CALL SLOTS(X,Y,BGL,III,JJJ,CGL)
      IF(JJJ .EQ. 9)GO TO 3
      CX=SLOTX(III,JJJ)*COS(BGL)
      CY=SLOTY(III,JJJ)*COS(CGL)
      PX=ABS(X-CX)
      PY=ABS(Y-CY)
      IF(PX .LE. HO2 .AND. PY .LE. WO2)GO TO 2
      JJJ=9
      GO TO 3
    2 AMPSL=AMP*COS(PY*PI/WW)*COS(PX*PI/HH)
      AC1=VECTDT(XL,L)
      B1=L(2)*L(2)+L(3)*L(3)
      B2=SQRT(B1)
      B3=L(2)*XL(2)+L(3)*XL(3)
      B4=L(2)*XM(2)+L(3)*XM(3)
      AC2=(XM(1)*B3+XL(1)*B4)/B2
      AMPSL=AMPSL*AC1*AC2
    3 RETURN
      END
```

```
      SUBROUTINE PINT(X0,Y0,Z0,DL,X1,Y1,Z1,KIC,JJJ,III,DLDZ)
      DIMENSION DL(3),D(5)
      DIMENSION CCZ(4),TZ(4)
      COMMON /RRRR/AA,CC,BOUT,BIN
      CCC=Z0-CC
      IF(III .EQ. 1)GO TO 10
      BB=BIN
      GO TO 11
   10 BB=BOUT
   11 AABB=AA*AA+BB*BB
      C1=0.5/AA/DL(3)/DL(3)
      C4=(DL(1)*X0+DL(2)*Y0)/DL(3)
      C2=(CCC+C4)/AA
      C3=0.5*(X0*X0+Y0*Y0-2.*AA*AA-BB*BB+CCC*CCC)/AA
      D(5)=C1*C1
      D(4)=2.*C1*C2
      D(3)=2.*C1*C3+C2*C2+1.
      D(2)=2.*C2*C3+2.*CCC
      D(1)=C3*C3-AABB+CCC*CCC
      CALL ZPOINT(Z0,D,CCZ,KJ,KIC)
      IF(KIC .GT. 0)GO TO 99
      IF(KJ .EQ. 0)GO TO 99
      KK=0
      DO 60 I=1,KJ
      ZZ=CCZ(I)
      QQ=AABB-(ZZ-CC)**2
      IF(QQ .LT. 0.)GO TO 60
      QQ=SQRT(QQ)-AA
      XX=X0+(ZZ-Z0)*DL(1)/DL(3)
      YY=Y0+(ZZ-Z0)*DL(2)/DL(3)
      XY=SQRT(XX*XX+YY*YY)
      DEL=QQ-XY
      IF(ABS(DEL) .LT. 0.01)GO TO 50
      GO TO 60
   50 KK=KK+1
      TZ(KK)=CCZ(I)
   60 CONTINUE
      IF(KK .EQ. 2)GO TO 70
      GO TO 99
   70 ZL=AMAX1(TZ(1),TZ(2))
      ZS=AMIN1(TZ(1),TZ(2))
      IF(JJJ .EQ. 1)GO TO 80
      Z1=ZS
      GO TO 81
   80 Z1=ZL
   81 X1=X0+(Z1-Z0)*DL(1)/DL(3)
      Y1=Y0+(Z1-Z0)*DL(2)/DL(3)
      KIC=0
      XX=SQRT(AABB-(Z1-CC)**2)
      YY=-(Z1-CC)
      DLDZ=YY/XX
      RETURN
   99 KIC=2
      RETURN
      END
      SUBROUTINE PLASMA(X1,Y1,Z1,L,M,N,DN,GA,GB,GAMA,
     1 ALPHA,BETA,PATH,DLDZ,IQ)
      DIMENSION L(3),M(3),N(3),DN(3),D(5)
      DIMENSION CCZ(4),TZ(4)
      REAL L,M,N
```

```
      COMMON /RRRR/AA,CC,BOUT,BIN
      DATA CON/1.61E-12/
      IF(Z1 .LT. 0. .OR. Z1 .GE. 48.)GO TO 99
      IF(Z1 .LT. 10.)GO TO 1
      IF(Z1 .LT. 15.)GO TO 2
      IF(Z1 .LT. 48.)GO TO 3
      GO TO 99
    1 TEF=0.01227*Z1**(0.5)+0.0012
      IF(Z1 .GE. 5.)GO TO 11
      CONI=2.039*10.E09
      GO TO 5
   11 CONI=2.223*10.E08
      GO TO 5
    2 TEF=0.012*Z1-0.08
      CONI=1.147*10.E08
      GO TO 5
    3 TEF=0.0224*Z1**(0.8)-0.0958
      IF(Z1 .LT. 20.)GO TO 31
      CONI=6.962*10.E08
      GO TO 5
   31 CONI=1.147*10.E08
    5 CONTINUE
      DEL1=CON*TEF*CONI*TAN(ALPHA)/COS(ALPHA)**2
      DEL=-DEL1*COS(2.*ALPHA)/SIN(ALPHA)
      PATH=PATH+DEL
      P1=DEL1*(GB*M(2)-GA*N(2))/DN(3)
      P2=DEL1*(GB*M(1)-GA*N(1))/DN(3)
      XB=X1+P1
      YB=Y1-P2
      BB=BOUT
      CCC=CC-Z1
      AABB=AA*AA+BB*BB
      ABC=-AABB+CCC*CCC
      C1=0.5/DN(3)/DN(3)/AA
      C2=(DN(1)*XB+DN(2)*YB)/DN(3)-CCC
      C2=C2/AA
      C3=XB*XB+YB*YB-AA*AA+ABC
      C3=C3/2./AA
      D(5)=C1*C1
      D(4)=2.*C1*C2
      D(3)=2.*C1*C3+C2*C2+1.
      D(2)=2.*C2*C3-2.*CCC
      D(1)=C3*C3+ABC
      CALL ZPOINT(Z1,D,CCZ,KJ,KIC)
      IF(KIC .GT. 0)GO TO 99
      IF(KJ .EQ. 0)GO TO 99
      KK=0
      DO 60 I=1,KJ
      ZZ=CCZ(I)
      OO=AABB-(ZZ-CC)**2
      IF(OO .LT. 0.)GO TO 60
      OO=SQRT(OO)-AA
      XX=XB+DN(1)/DN(3)*(ZZ-Z1)
      YY=YB+DN(2)/DN(3)*(ZZ-Z1)
      XY=SQRT(XX*XX+YY*YY)
      DEL=OO-XY
      IF(ABS(DEL) .LT. 0.01)GO TO 50
      GO TO 60
   50 KK=KK+1
      TZ(KK)=CCZ(I)
```

```
60 CONTINUE
   IF(KK .EQ. 2)GO TO 70
   GO TO 99
70 ZL=AMAX1(TZ(1),TZ(2))
   ZS=AMIN1(TZ(1),TZ(2))
   IF(ZS .GT. Z1)GO TO 90
   IF(ZL .GT. Z1)GO TO 91
   GO TO 99
91 Z8=ZL
   GO TO 81
90 Z8=ZS
81 X1=XB+DN(1)/DN(3)*(Z8-Z1)
   Y1=YB+DN(2)/DN(3)*(Z8-Z1)
   Z1=Z8
   XX=SQRT(AABB-(Z1-CC)**2)
   YY=-(Z1-CC)
   DLDZ=YY/XX
   CALL PNORM(X1,Y1,Z1,DLDZ,DN)
   CALL CONAB(M,N,DN,GA,GB,GAMA)
   CALL ANGLES(L,DN,GAMA,ALPHA,BETA,0,KICK)
   IF(KICK .GT. 1)GO TO 99
   IQ=0
   RETURN
99 IQ=2
   RETURN
   END
   FUNCTION DIST(X0,Y0,Z0,BNEX,X1,Y1,Z1)
   DIST=(X1-X0)**2+(Y1-Y0)**2+(Z1-Z0)**2
   DIST=SQRT(DIST)*BNEX
   RETURN
   END
   SUBROUTINE ZPOINT(Z0,D,CCZ,KJ,KIC)
   COMMON /RRRR/AA,CC,BOUT,BIN
   DIMENSION D(5),COF(5),ROOTR(4),ROOTI(4),
  1 CZ(4),TZ(4),CCZ(4)
   DD=CC-BIN
   DIMENSION TTZ(4)
   CALL POLRT(D,COF,4,ROOTR,ROOTI,IER)
   IF(IER .EQ. 0)GO TO 12
   GO TO 99
12 CONTINUE
   DO 19 I=1,4
19 CZ(I)=0.
   K=0
   IF(ABS(ROOTI(1)).LT. 0.001)GO TO 20
   GO TO 21
20 K=K+1
   CZ(K)=ROOTR(1)
21 IF(ABS(ROOTI(2)).LT. 0.001)GO TO 22
   GO TO 23
22 K=K+1
   CZ(K)=ROOTR(2)
23 IF(ABS(ROOTI(3)).LT. 0.001)GO TO 24
   GO TO 25
24 K=K+1
   CZ(K)=ROOTR(3)
25 IF(ABS(ROOTI(4)).LT. 0.001)GO TO 26
   GO TO 27
26 K=K+1
   CZ(K)=ROOTR(4)
```

```
27 KK=K
   IF(KK .EQ. 0)GO TO 99
   DO 28 I=1,KK
28 TZ(I)=Z0+CZ(I)
   IF(KK .EQ. 2)GO TO 30
   IF(KK .EQ. 4)GO TO 40
   GO TO 99
30 KJ=0
   IF(TZ(1) .GT. DD)GO TO 31
   GO TO 32
31 KJ=KJ+1
   CCZ(KJ)=TZ(1)
32 IF(TZ(2) .GT. DD)GO TO 33
   GO TO 34
33 KJ=KJ+1
   CCZ(KJ)=TZ(2)
34 IF(KJ .EQ. 0)GO TO 99
   GO TO 48
40 KJ=0
   IF(TZ(1) .GT. DD)GO TO 41
   GO TO 42
41 KJ=KJ+1
   CCZ(KJ)=TZ(1)
42 IF(TZ(2) .GT. DD)GO TO 43
   GO TO 44
43 KJ=KJ+1
   CCZ(KJ)=TZ(2)
44 IF(TZ(3) .GT. DD)GO TO 45
   GO TO 46
45 KJ=KJ+1
   CCZ(KJ)=TZ(3)
46 IF(TZ(4) .GT. DD)GO TO 47
   GO TO 48
47 KJ=KJ+1
   CCZ(KJ)=TZ(4)
48 CONTINUE
   KIC=0
   RETURN
99 KIC=2
   RETURN
   END

                                         221 LINES PRINTED
```

SUBROUTINE POLRT

This subroutine computes real and complex roots of a real polynomial and is available in IBM Scientific Subroutine Package (SSP). Ref. System/360 Scientific Subroutine Package, Programmer's Manual, Form H20-0205-n.