MaN 48

Distribution Category UC-32

LAWRENCE LIVERMORE LABORATORY

University of California/Livermore, California/94550

UCRL-51821

FORTRAN SUBROUTINES FOR THE NUMERICAL EVALUATION OF SOMMERFELD INTEGRALS UNTER ANDEREM

Darrel L. Lager
R. Jeff Lytle

MS.date: May 21, 1975

Foreword

This work was sponsored by the Defense Advanced Research Projects Agency. It was performed under the direction of James Goodwyn and monitored by Donald Barrick.

Contents

FORTRAN SUBROUTINES FOR THE NUMERICAL EVALUATION OF THE SOMMERFELD INTEGRALS UNTER ANDEREM

Abstract

A description is given of the subroutine package added to the computer program WF-LLL2A to extend its capabilities to include solving for the currents on thin wire structures when the structures interact strongly with a lossy ground. This includes not only the case of a structure located above the ground, but also buried beneath it and with portions above and below the ground. The routines solve for the mutual impedance between two segments of the structure by the use of approximate formulas due to Norton (when appropriate) or, when necessary, by the evaluation of Sommerfeld integrals through numerical contour integration.

Introduction

This report describes a subroutine package added to the computer program, WF-LLL2A (described in Appendix C), to extend its capabilities for determining the currents on thin wire structures interacting strongly with a lossy ground. The program divides the structure into N straight-wire segments and calculates the mutual impedance between each of the segments to determine the entries in the impedance matrix, Z. It then computes the currents on the segments by solving the system of equations, E = ZI, for the currents I, where E is the field incident on the structure.

The subroutine package calculates the mutual impedance between two segments by determining the tangential field at the observer segment due to a unit current on the source segment. The package uses two approaches to calculate the tangential fields, one is the numerical evaluation of Sommerfeld integrals,[1,2] listed in Appendix A, via the "optimum" contours of integration determined by Lytle and Lager,[3] and the other uses Norton's approximate formulas,[4] listed in Appendix B. Incidentally, the title of this paper specifically mentions the Sommerfeld evaluations since they required most of the development effort; however, this package obviously does more than simply evaluate integrals, hence the term unter anderem, German for "among other things." In the interest of reducing computer time, the section using Norton's formulas is automatically chosen whenever the parameters of the problem are within the range where his formulas are known to provide sufficient accuracy. Presently, Norton's formulas are used whenever the source and observer are both above ground and the separation between them is greater than a wavelength.

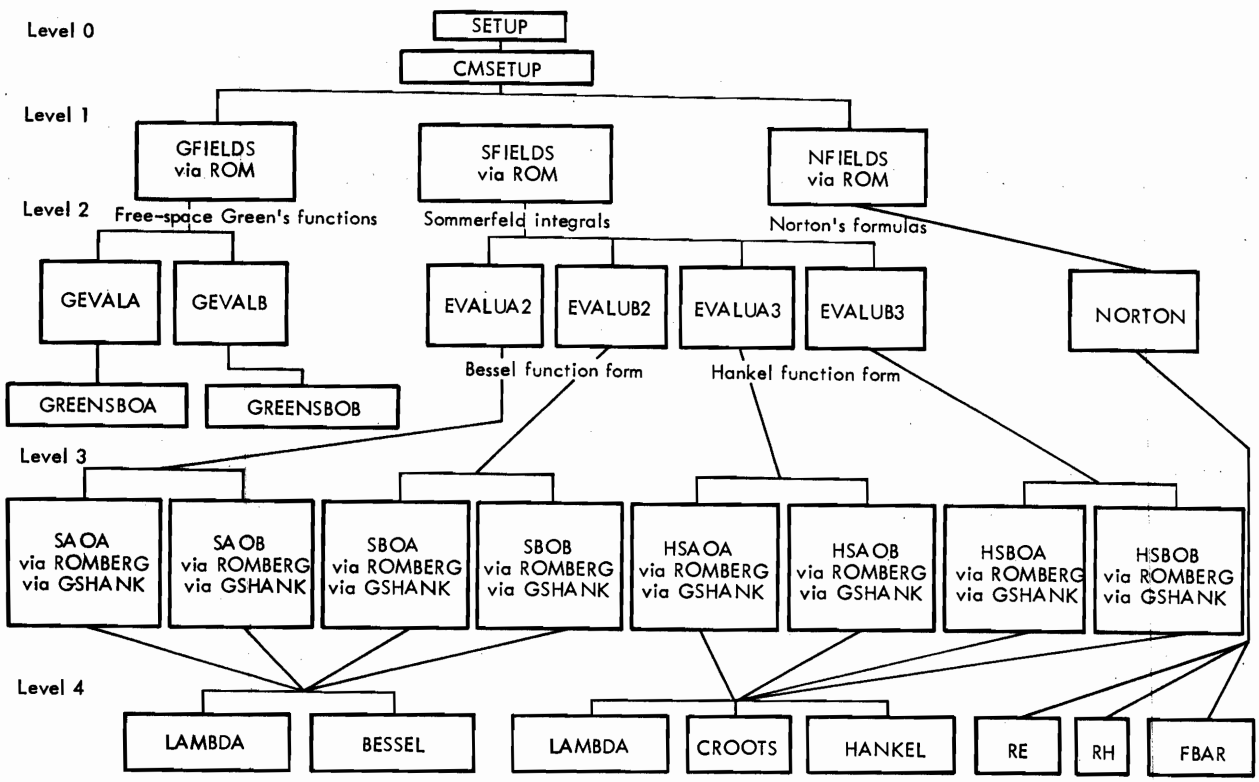

Fig. 1. Flow chart for subroutine package for calculating ground interactions.

Logic Flow

Figure 1 is a flow chart showing the path of control through the routines. The logic is a "tree" structure with control beginning at the top and proceeding to the lower levels via "branches" until the bottom level is reached. Control then returns in the reverse direction along the same path. This method appears to be much easier to debug, verify, and modify than other possible arrangements; it is also intended to make the program easier to understand.

The level 0 routine, SETUP, initializes variables in the various common blocks to the values of the ground parameters and the frequency. In the level 0 routine, CMSETUP, the mutual impedance between two segments is determined by calculating the tangential field at the center of the observer segment due to a unit current on the axis of the source segment according to the geometry indicated in Fig. 2. This is found by integrating the fields due to a Hertzian dipole as a function of its position along the axis of the source segment. The routines at level 1 (GFIELDS, SFIELDS, and NFIELDS) find the E field tangential to the observer segment as a function of the parameter T, which specifies the position of the Hertzian dipole along the axis of the source segment.

The calculations of the tangential field using Norton's formulas is done by subroutine NORTON. The calculation of the tangential field using the Sommerfeld integrals is done by the routines SFIELDS and GFIELDS. The Sommerfeld integrals have been broken

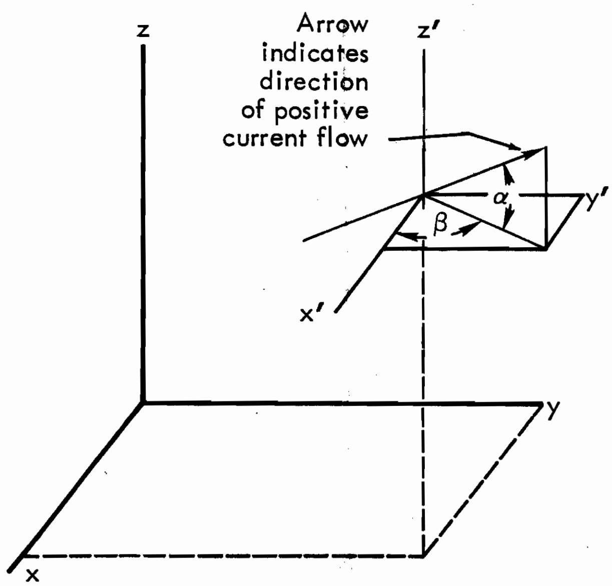

Fig. 2. Notation for geometry describing location and orientation of a segment in space P (α,β,z),

into two terms, one for the freespace Green's functions (by GFIELDS) and the other a Sommerfeld "correction" to the free-space fields (by SFIELDS). The subroutine ROM, called by CMSETUP, performs an adaptive Romberg integration, using GFIELDS, SFIELDS, or NFIELDS as an integrand, to find the mutual impedance between the two segments. The results from the integration of SFIELDS and GFIELDS are added to get the correct results for the Sommerfeld integrals. The subroutine, CMSETUP, then stores the result in the impedance matrix, CM, after performing the interpolation appropriate to the sine, cosine, and constant basis functions used.

The level 2 routines return the values of the ρ,ϕ, and z components of the E-fields at an observation point due to vertical and horizontal Hertzian dipoles located at a source point. When using Norton's equations (Appendix B) to make the calculation, the subroutine NORTON is used. When using the Sommerfeld integrals (Appendix A), the EVALU routines are used — EVALUA2 and EVALUA3 for an above-surface source using the Bessel and Hankel function formulations, respectively — EVALUB2 and EVALUB3 for a below-surface source using the Bessel and Hankel formulations, respectively. When using the equations for the-free space Green's functions, the subroutines GEVALUA and GEVALUB are used for above- and below-surface sources respectively.

The level 3 routines are the various integrands for the Sommerfeld integrals. The routines ROMBERG and GSHANK perform numerical complex contour integration of level 3 routines to obtain the values of the U and V integrals (and their ρ and z partial derivatives). The equations for the U and V integrals are shown in Appendix A. The notation used is that of Baños.[5] The level 3 routines are named according to the functional form used and the positions of the source and observer relative to the interface. For instance, the routine HSAOB returns the U's, V's, and their derivatives for the Hankel function formulation for the source above and the observer below the interface.

The level 4 routines calculate the elementary functions. They evaluate the Bessel functions (BESSEL), Hankel functions (HANKEL), the value of the variable of integration on the complex contour (LAMBDA), and the appropriate complex square root (CROOTS) for the Sommerfeld integrals. For Norton's formulas they evaluate the value of the reflection coefficients (RE and RH) and the Sommerfeld attenuation function (FBAR).

In the section *Subroutines and Functions* a more detailed description of each of the subroutines is given (in alphabetical order), listing the equations used, the input and output calling arguments, the input and output COMMON blocks, and calls to other subroutines. An input COMMON block is defined as one where the variables in it are only used as data and are not modified by the subroutines. For example, the subroutine EVALUA2 lists EVALCOM as an input common block since the variables in EVALCOM are not modified by EVALUA2, but are only used to obtain the values of the observer height., and other "constants." An output COMMON block is defined as one whose variables are modified by the subroutine. EVALUA2 lists CONTOUR as an output common block since the variables in it — ITYPE, A, and B — are modified by EVALUA2. The reason for differentiating between input and output common is to simplify the task of determining the flow of data through the program. The section *Common Blocks* gives a description of the major common blocks, listing and defining each of the variables in them.

Numerical Results

BOUNDARY CONDITIONS

One of the test problems done to verify the correctness of subroutines in the package is shown in Fig. 3 where the fields due to vertical and horizontal Hertzian sources were calculated for observers slightly above and below the interface. The

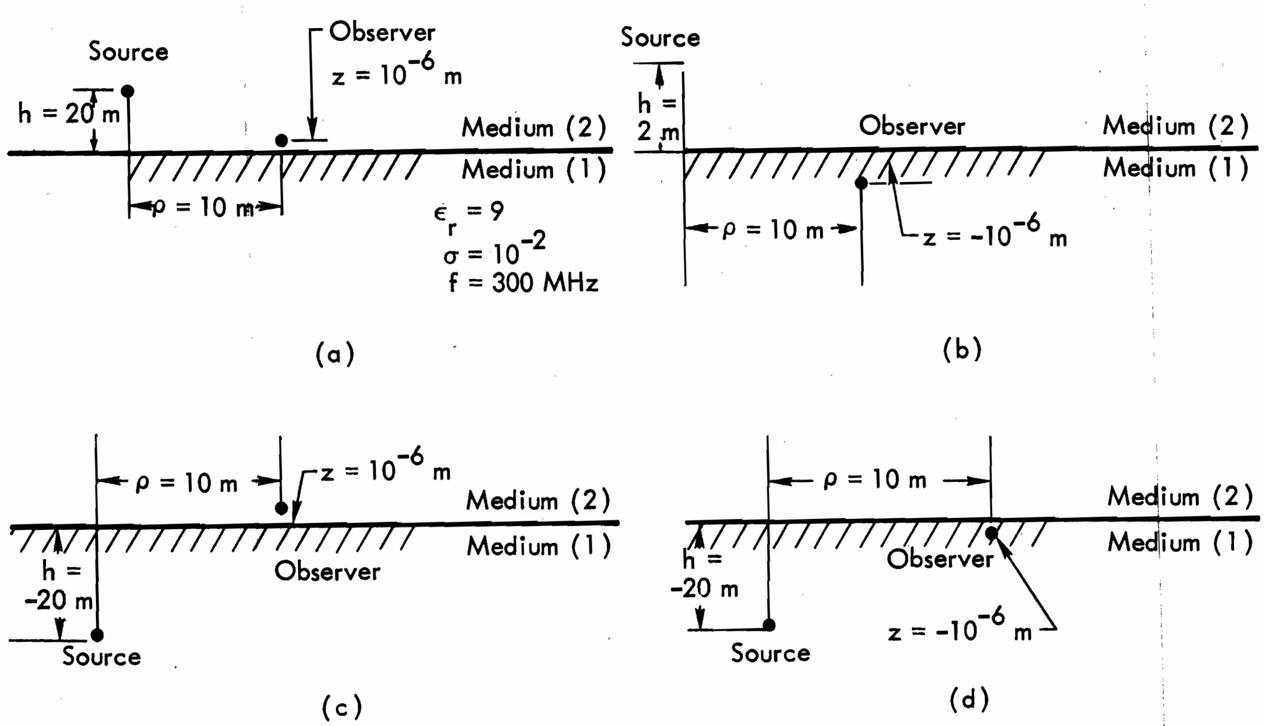

Fig. 3. Test cases to verify the correctness of the evaluation of the Sommerfeld integrals by determining if the boundary conditions at the interface are satisfied.

observed fields satisfied the boundary conditions at the interface (i.e., continuity of tangential $\mathscr{E}$ and normal flux density, $D = \varepsilon\mathscr{E}$) stated below:

$E\rho(2) = E\rho(1)$, tangential $\mathscr{E}$ continuous,

$E\phi(2) = E\phi(1)$, tangential $\mathscr{E}$ continuous, and

$Ez(2) = \varepsilon/\varepsilon_0\ E_z(1)$, normal D continuous,

where $E_x(i)$ is the x component of the $\mathscr{E}$ field, due to either a vertical or horizontal source, for the observer located in medium (i). ε is the complex $\varepsilon = \varepsilon_0\varepsilon_r + i\sigma/w$.

The values for the fields on either side of the interface were calculated for the cases depicted in Fig. 3 and are listed in Table 1. The real and imaginary parts,

Table 1. Fields at the observer for the cases of Fig. 3.

F= 2.000E+00
ER= 9.000E+00
SIG= 1.000E-02
RHO= 1.000E+01

H= 2.000E+01

2= 1.000E-06

A3ERV	-1.18693E-02R	1.68570E-03I	1.19884E-02M	1.71917E+02P
A3EZV	-1.40863E-01R	1.77549E-01I	2.26640E-01M	1.28328E+02P
A3ERH	-2.09793E-02P	4.64563E-03I	2.13875E-02M	1.67514E+02P
A3EZH	2.26535E-02R	-1.49365E-01I	1.51075E-01M	-8.13609E+01P
A3EPH	2.21803E-02R	-4.30386E-03I	2.25940E-02M	-1.09812E+01P

Z=-1.000E-06

A3ERV	-1.18660E-02R	1.68500E-03I	1.19850E-02M	1.71918E-02P
A3EZV	2.54676E-03R	2.73408E-03I	3.73647E-03M	4.70316E+01P
A3ERH	-2.09793E-02R	4.64562E-03I	2.13875E-02M	1.67514E-02P
A3EZH	-2.38222E-03R	-7.36578E-04I	2.49350E-03M	-1.62818E-02P
A3EPH	2.21803E-02R	-4.30386E-03I	2.25940E-02M	-1.09812E+01P

H=-2.000E+01

Z= 1.000E-06

B3ERV	5.08796E-05R	4.08100E-05I	6.52242E-05M	3.87320E+01P
B3EZV	-1.32329E-05R	4.32717E-05I	4.52498E-05M	1.07004E+02P
B3ERH	-7.36010E-05R	-4.78335E-05I	8.77790E-05M	-1.46980E+02P
B3EZH	-1.77493E-04R	-1.39608E-05I	1.78041E-04M	-1.75503E+02P
B3EPH	1.34234E-04R	5.83473E-05I	1.46366E-04M	2.34931E+01P

z=-1.000E-06

B3ERV	5.08873E-05R	4.08081E-05I	6.52290E-05M	3.87272E+01P
B3EZV	6.72620E-07R	3.70021E-07I	7.44877E-07M	2.54440E+01P
B3ERH	-7.36010E-05R	-4.78335E-05I	8.77790E-05M	-1.46980E+02P
B3EZH	6.63037E-07R	-2.86272E-06I	2.93850E-06M	-7.69596E+01P
B3EPH	1.34234E-04R	5.83473E-05I	1.46366E-04M	2.34931E+01P

magnitude, and phase of each of the components are given according to the notation:

$$\left\{\begin{matrix} A3 \\ B2 \end{matrix}\right\} E \left\{\begin{matrix} R \\ P \\ Z \end{matrix}\right\} \left\{\begin{matrix} V \\ H \end{matrix}\right\}$$

where A3 means routine EVALUA3 calculated the fields using the Hankel function form of the Sommerfeld integrals for an above-surface source; and

B3 means routine EVALUB3 calculated the fields using the Hankel function form of the Sommerfeld integrals for a below-surface source.

E indicates the $\mathscr{E}$ field,

R means the ρ components,

P means the ϕ component,

Z means the z component,

V means a vertical source, and

H means a horizontal source.

Thus the name, A3EPH, indicates the ϕ component of the $\mathscr{E}$ field due to a horizontal source was calculated for an above surface source using routine EVALUA3.

A comparison of the fields in Table 1 shows that they do, indeed, satisfy the boundary conditions for the cases depicted in Fig. 3 where $\varepsilon/\varepsilon_0 \cong 9 + i\ 60$.

The entries in Table 1 were determined by a "driver" routine named BOUNDARY which called the level 2 subroutines in the package. This was done to simplify the debugging by verifying the correctness of the subroutine package before combining it with the WF-LLL2A code.

INPUT RESISTANCE OF DIPOLES

The Sommerfeld/Norton subroutine package was made part of the computer program, WF-LLL2A, for determining the currents flowing on thin wire structures. A brief description of the theoretical background, capabilities, and timing for the code is given in APPENDIX C.

A test problem done to verify the correctness of the code after introducing the subroutine package involved the calculation of the input resistance of vertical and horizontal half-wave dipoles as functions of the ground conductivity, height, and the ground treatment used. Figures 4 and 5 show the exact agreement between the WF-LLL2A code and the results of Miller, et al.[6,7,8] who also used specular reflection coefficient and Sommerfeld ground treatment. For convenience, the values of Figs. 4 and 5, calculated by the WF-LLL2A code, are listed in Tables 2 and 3.

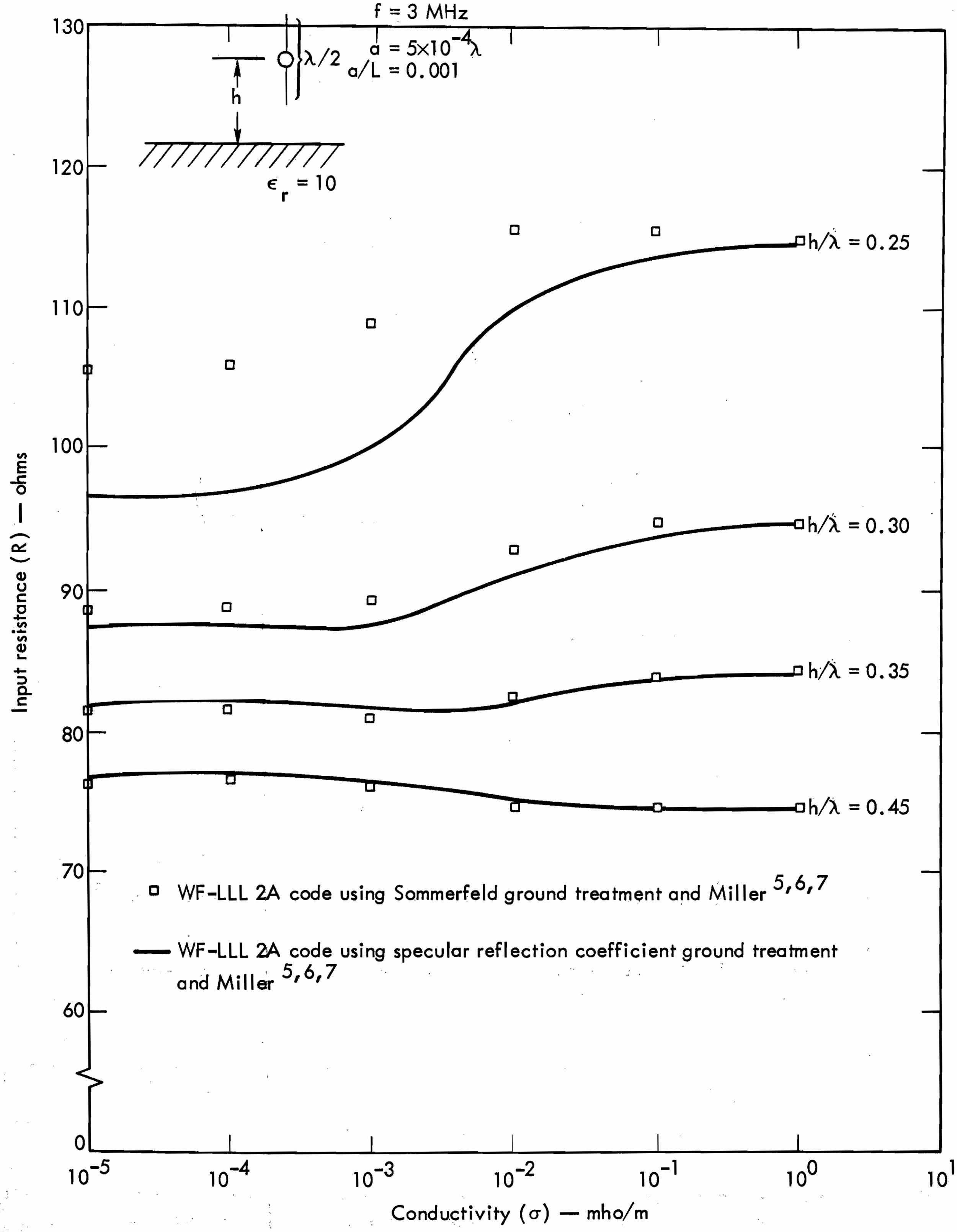

Fig. 4. Input resistance vs ground conductivity and height for five-segment, vertical, half-wave dipole. Results by the WF-LLL2A code and Miller[8] are the same.

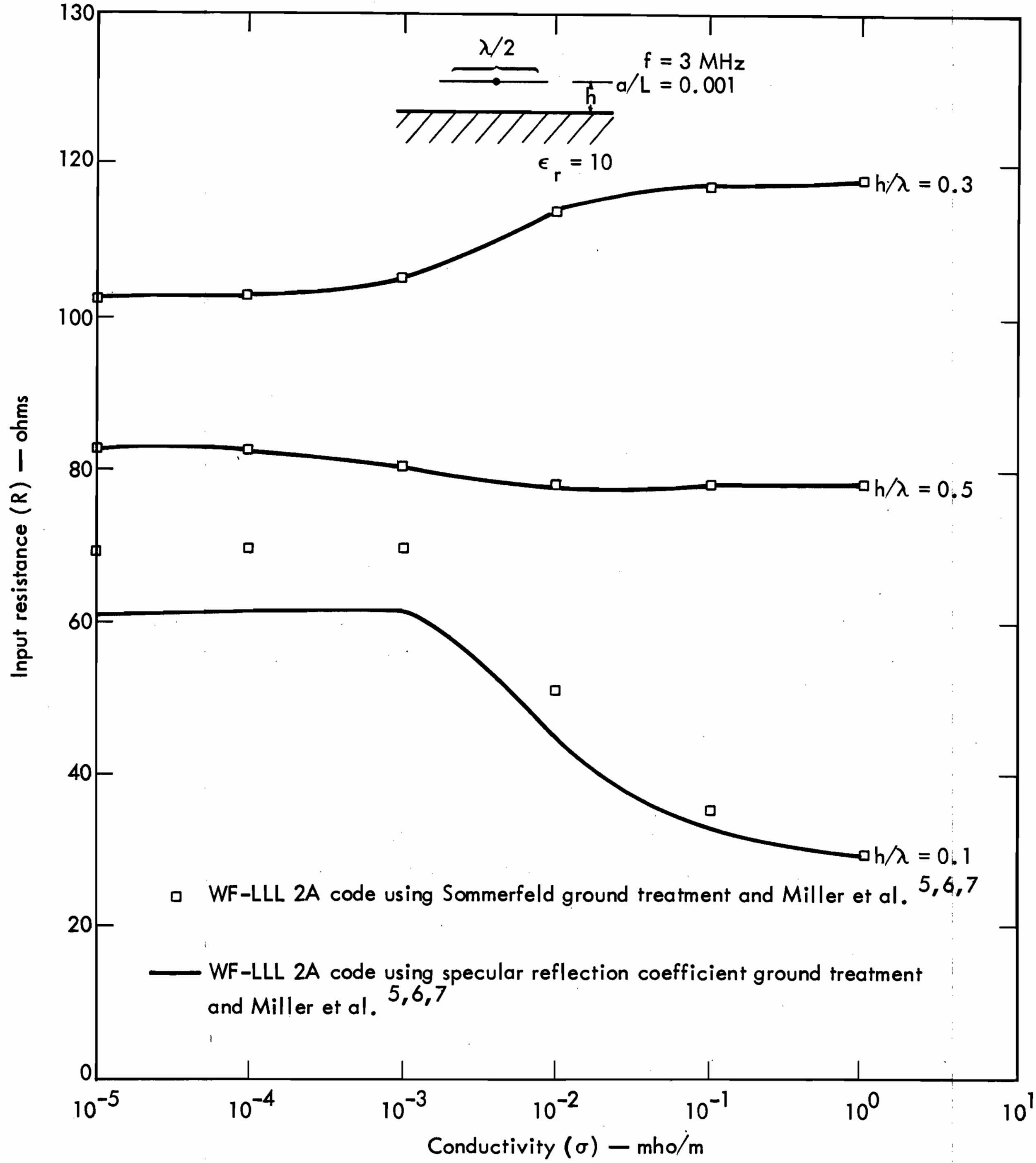

Fig. 5. Input resistance vs conductivity and height for five-segment, horizontal, half-wave dipole. Results by the WF-LLL2A code and Miller[8] are the same.

Table 2. Input resistance of five-segment, vertical, half-wave dipole vs height and conductivity.

Conductivity	Input resistance, Ω Height, m			
(σ)	25	30	35	45
10^{-5}	105.7	88.78	81.21	76.14
10^{-4}	105.8	88.75	81.14	76.10
10^{-3}	108.8	89.02	80.06	75.61
10^{-2}	115.4	92.91	82.23	74.63
10^{-1}	115.2	94.54	83.63	74.67
10^{-0}	114.9	94.99	84.10	74.74

Table 3. Input resistance of five-segment, horizontal, half-wave dipole vs height and conductivity.

Conductivity	Input resistance, Ω Height, m		
(σ)	10	30	50
10^{-5}	69.45	102.2	83.40
10^{-4}	69.67	102.4	82.85
10^{-3}	69.75	104.6	80.88
10^{-2}	50.73	114.0	72.82
10^{-1}	35.18	117.6	78.38
10^{-0}	29.81	118.5	78.76

Future Extensions

Future improvements are directed to reducing the computer time to calculate the ground interactions by following two approaches. The first is to decrease the time for the evaluation of the Sommerfeld integrals. Currently, the program spends about 40% of the time evaluating Bessel and Hankel functions. This was decreased from about 90% by writing faster versions of the Hankel and Bessel routines, but further improvement is still possible. Another planned improvement is the implementation of the vertical contour of integration for the Hankel function form of the Sommerfeld integral discussed by Lytle and Lager.[3]

The second approach is to use approximate expressions wherever they provide sufficient accuracy. Planned improvements include formulas for a buried antenna by Baños[5] and a series of empirical formulas valid for various ranges of separation and values of ground parameters. It will be possible to use the "full-bore" Sommerfeld integrals to determine the relative accuracy of the various approximate expressions and to determine the ranges in value of the parameters involved for which sufficient accuracy is attainable.

Common Blocks

This section gives a description of the major common blocks used by the subroutine package. These blocks appear in most of the routines. A brief description of each of the blocks follows, stating the general purpose of the common block and the specific use for each of the variables in it.

COMMON EVALCOM

EVALCOM is the common block used by most every routine to get values of the locations of source and observer in cylindrical coordinate system and the various ranges shown in Fig. 2. In addition, the values of the propagation constants in the two media, k_1 and k_2, are specified. The variables are all initialized by the routine SETUP.

- HI - Height of source, h.
- ZI - Height of observer, z.
- ZMH - z - h.
- ZPH - z + h.
- R - Separation between source and observer in cylindrical coordinate system, ρ.
- R1 - Range from source to observer, $R_1 = \sqrt{\rho^2 + (z + h)^2}$.
- R2 - Range from image of source to observer, $R_2 = \sqrt{\rho^2 + (z - h)^2}$.
- CJ - $\sqrt{-1}$.
- CK1 - Propagation constant in lower medium, k_1.
- CK1SQ - k_1^2.
- CK2 - Propagation constant in upper medium, k_2.
- CK2SQ - k_2^2.
- COEE - Constant multiplier for Sommerfeld integrals = $j\omega\mu_0/4\pi$.
- COEH - Constant multiplier for Sommerfeld integrals = $-1/(4\pi)$.

COMMON RCOM

RCOM contains the variables to be incremented by integration routine ROM for determining the speed of convergence for a given integral.

- NEVALS - incremented by the number of evaluations of the integrand necessary to achieve convergence.
- TROUBLE - incremented whenever convergence cannot be achieved.

COMMON ROMCOM

ROMCOM contains the variables to be incremented by integration routine ROMBERG for determining the speed of convergence for a given integral.

NEVALS - incremented by the number of evaluations for the integrand necessary to achieve convergence.

TROUBLE - incremented whenever convergence cannot be achieved.

COMMON SFCOM

SFCOM is the common block used to communicate to the routines SFIELDS, NFIELDS, and GFIELDS, the locations of the centers of the source and observer segments (in the x,y,z coordinate system) and their orientation, wire radius, etc.

XS, YS, ZS } x,y,z coordinates of center of source segment

AS, BS } α,β angles of orientation of source segment (see Fig. 2)

SS length of source segment

WR wire radius of source segment

XO, YO, ZO } x,y,z coordinates of center of observer segment

AO, BO } α,β angles of orientation of observer segment (see Fig. 2)

IMUTUAL = 1 when mutual impedance between two different segments is desired.

= 0 when self-impedance of a segment is desired.

Subroutines and Functions

A brief description of each subroutine and function is given, along with a description of each of the arguments and a list of the pertinent equations. The routines appear in alphabetical order on the following pages so it will be necessary to refer to the block diagram (Fig. 1) in order to determine the logic flow.

A casual observation will show there are many routines which appear to be very nearly duplicates. A trade-off was made between the amount of code generated and simplification of the logic flow. The tree-type structure of Fig. 1 would be impossible to recognize without the duplication of code. This has several advantages, one of which is that if it is known that a particular problem will never use the Norton formulas, for instance, then the routines NFIELDS, NORTON, RE, RH, and FBAR could be left out. The other advantage is that it is much easier to do some fine tuning of such things as step sizes, contours of integration used, etc. for the situation of below-surface source, for example.

SUBROUTINE BESSEL (Z, JO, JOP)

BESSEL returns the value of the zeroth-order cylindrical Bessel function of the first kind, J_0, and its first derivative, J_0', for complex argument, z, via ascending series and asymptotic expansions.[9]

INPUT ARGS:

Z = z, the argument of the Bessel function (complex).

OUTPUT ARGS:

$JO = J_0\ (z)$, (COMPLEX),

$JOP = J_0'\ (z) = \frac{dJ_0\ (z)}{dZ}$, (COMPLEX).

INPUT COMMON: None.

OUTPUT COMMON: None.

ROUTINES CALLED:

CEXP - Complex exponential.

CSQRT - Complex square root.

SQRT

SUBROUTINE CMSETUP (ZRATIP, ZRATIM, KSYMP, IPGND, IVERTRC)

CMSETUP performs the calculations necessary to fill the impedance matrix, CM for the computer program WF-LLL2A, which performs a method-of-moments solution for the currents on a thin wire structure using sine, cosine, and constant basis functions by solving the matrix equation, E = ZI, for I; where Z is the impedance matrix, E is the incident field on each segment of the structure, and I is the current on each segment.

The entries in the impedance matrix, CM(I,J), are determined by calculating the mutual impedance between the segments of the structure by finding the E field tangential to the Ith segment (called the source). The arguments KSYMP, IPGND, and IVERTRC control the method used to obtain the value of the tangential field. A value of KSYMP = 1 means that the structure is located in free-space so the fields are determined by integrating the free space Green's functions. A value of KSYMP = 2 means the structure is located over a lossy ground and that the E field is to be determined by adding the free space fie to an image field modified by the specular reflection coefficient (IVERTRC = 0), or the vertical incidence reflection coefficient (IVERTRC = 1). For the case of a perfect ground IPGND = 1) the reflection coefficients are set to unity. A value of KSYMP = 3 means the structure is in the presence of a lossy half-space and the Sommerfeld/Norton subroutine package, the subject of this report, is to be used.

CMSETUP is the level 0 subroutine in Fig. 1 which calls the level 1 routines in the package, GFIELDS, SFIELDS, and NFIELDS after filling the entries in common block SFCOM with the coordinates of the source and observer segments. The E field obtained by using the Sommerfeld integrals is found by adding the contributions of GFIELDS (the free-space Green's functions) and SFIELDS (the Sommerfeld "ground correction"). When the separation between the source and observer is greater than one wave length, NFIELDS is called to obtain the E field using Norton's formulas.

To reduce the computer time used to fill the CM matrix, CMSETUP exploits the Toeplitz symmetry present in the structure. CMSETUP assumes that the user has numbered the segments in the structure so that the segments which are not Toeplitz-symmetric come before the segments which are. Figure 6 gives a Beverage antenna as an example in which the driven and loaded segments (which are not Toeplitz-symmetric) are numbered 1 and 2, respectively, and the segments on the horizontal wire (which have full Toeplitz symmetry) are numbered 3 through 10. To obtain the number of non-Toeplitz segments, CMSETUP reads a data card of the form:

NONTOEP N,

where the word NONTOEP begins in column 1 and the number N is 2 for the example.

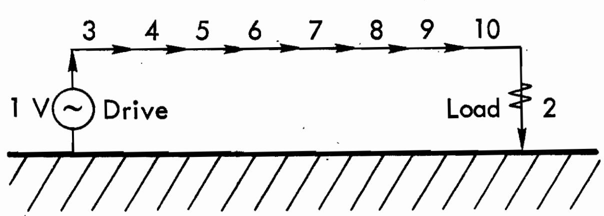

Fig. 6. Thin-wire model for Beverage antenna showing sequence of numbering segments so that Toeplitz symmetry may be exploited.

A significant decrease in computer time is obtained by exploiting Toeplitz symmetry, since it is only necessary to calculate the entries in the first row of the submatrix containing the Toeplitz antenna element. The other entries of the submatrix are then obtained according to the relation:

$$A_{ij} = A_{1,\,|i-j|+1} \qquad i = 2,\ldots M;\ j = 1,\ \ldots M,$$

where A is the Toeplitz submatrix and M is the number of Toeplitz segments (8 for the example).

INPUT ARGS:

ZRATIP - The ratio of the impedance of the upper medium to the lower.

ZRATIM - The ratio of the impedance of the lower medium to the upper.

KSYMP - Controls the ground treatment used.

= 1 for free space.

= 2 for specular or vertical incidence reflection coefficient.

= 3 for Sommerfeld/Norton.

IPGND - Specifies presence of perfect ground for KSYMP = 2.

= 1 for perfect ground (infinite conductivity)

= 0 otherwise.

IVERTRC- Specifies type of reflection coefficient for KSYMP = 2.

= 1 for vertical incidence reflection coefficient.

= 0 for specular reflection coefficient.

OUTPUT ARGS: None (See OUTPUT COMMON).

INPUT COMMON

GEOM - To get the position and orientation of the segments of the structure.

ANGL - To get sines and cosines of the angles of orientation of the segments of the structure.

JUNK - To get the connections between the various segments so that interpolation can be performed.

RCOM - To get the number of evaluations of the functions integrated by routine ROM.

ROMCOM - To get the number of evaluations of the functions integrated by routine ROMBERG.

REFL - Contains variables used for calculating reflection coefficients.

FREQ - To get the frequency.

JOBCOM - To get the current job number.

OUTPUT COMMON

333 - This block contains the CM matrix filled by CMSETUP.

SFCOM - To pass the locations and orientations of the source and observer segments to the routines GFIELDS, SFIELDS, and NFIELDS.

FUNCTION CROOTS (Z, XK, XL)

CROOTS returns the complex square root of Z, where Z is assumed to be $(\lambda-k)$, for the value of lamba, XL, relative to the vertical branch cut beginning at point k, XK. The choice of root implied by the vertical branch cut is shown in Fig. 7b, the Z plane, where the root chosen is always to the right of the heavy line. The numbers in the octants of the Z plane indicate the quadrant in the Z plane (Fig. 7a) in which Z must lie to produce a root in a given octant. Note that the root chosen by the complex square root routine, CSQRT, with the real part positive, will be the correct root for Z in quadrants

1, 2, and 4, and that it will be the negative of the desired root for Z in quadrant 2. CROOTS calls CSQRT to get the real part positive root and negates the results for the case of the Z in quadrant 2.

INPUT ARGS:

Z - number for which correct complex square root is to be found using a vertical branch cut (COMPLEX).

XK - the point in the complex lambda plane where the vertical branch cut begins (XK is either k, on k2, the propagation constants), (COMPLEX).

XL - lambda, (COMPLEX).

OUTPUT ARGS:

CROOTS = $\sqrt{z}$ according to vertical branch cut shown in Fig. 7.

INPUT COMMON: None.

OUTPUT COMMON: None.

ROUTINES CALLED:

CSQRT — to get complex square root with real part positive.

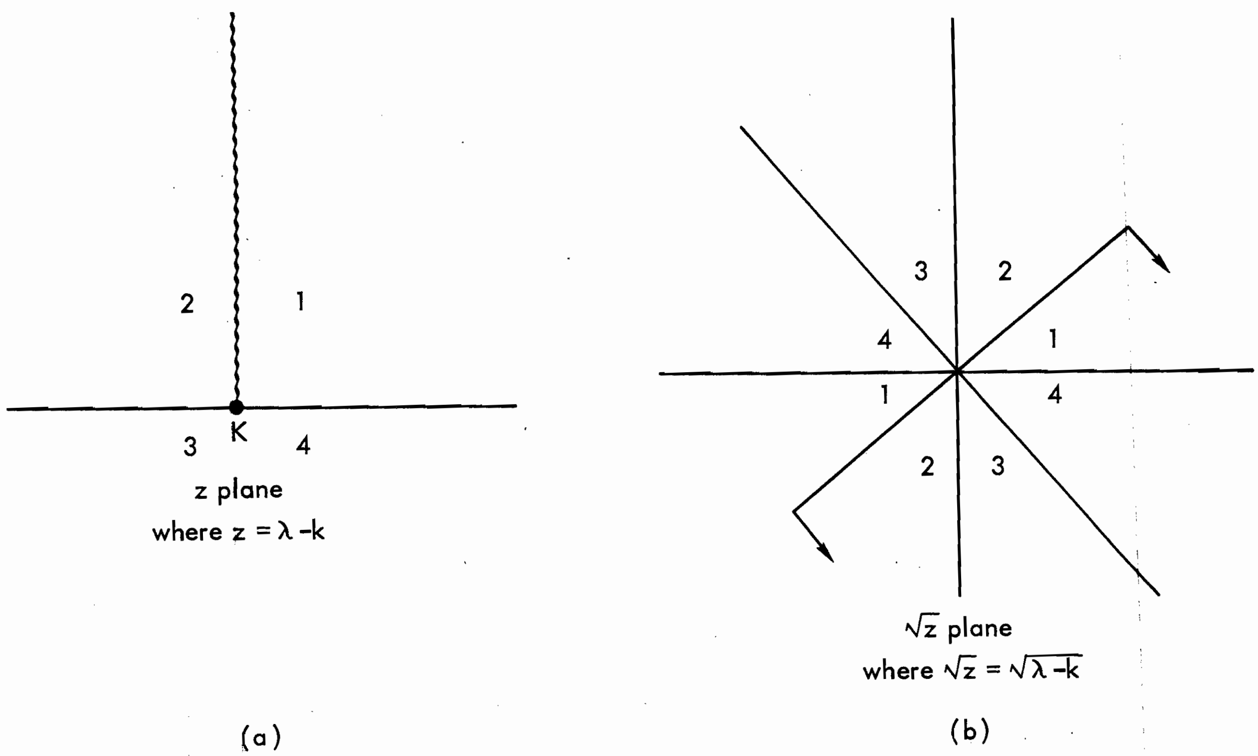

Fig. 7. Root chosen by CROOTS for vertical branch cut.

SUBROUTINE EVALUA2 (ERV, EZV, ERH, EZH, EPH)

EVALUA2 returns the ρ, z, and ϕ fields due to vertical and horizontal dipoles using the Bessel function form of the U and V Sommerfeld integrals from Appendix A for a source located above ground and using contour of integration No. 2 shown in Fig. 8. The integration of the path (1) to (2) is performed by routine ROMBERG and the path from (2) to infinity is performed by routine GSHANK.

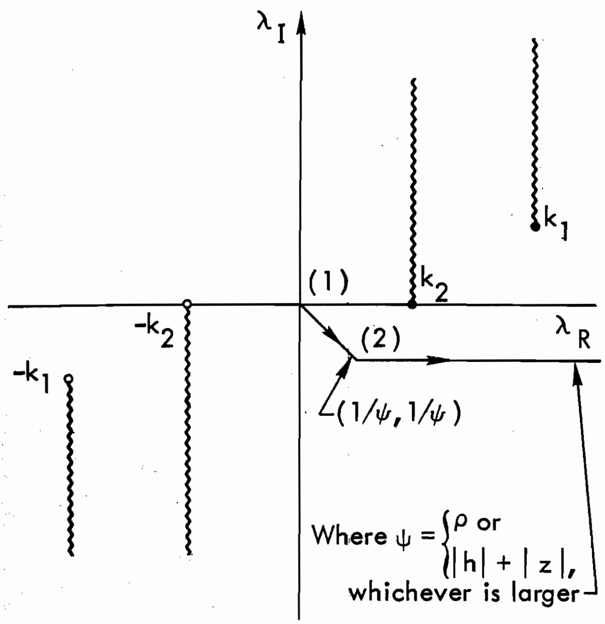

Fig. 8. Contour of integration used by routines EVALUA2 and EVALUB2 for Bessel function form of the U and V integrals.

INPUT ARGS: None.

OUTPUT ARGS: ERV - ρ field due to vertical source (COMPLEX).

EZV - z field due to vertical source (COMPLEX).

ERH - ρ field due to horizontal source (COMPLEX).

EZH - z field due to horizontal source (COMPLEX).

EPH - ϕ field due to horizontal source (COMPLEX).

INPUT COMMON: EVALCOM - To get ρ, z, h and propagation constants, etc.

ROMCOM - To communicate with ROMBERG.

OUTPUT COMMON: CONTOUR - To communicate with routine LAMBDA.

ITYPE - Set to 1 for linear path (INTEGER).

A - Set to lower limit of integration (COMPLEX).

B - Set to upper limit of integration (COMPLEX).

SUBROUTINES CALLED:

ROMBERG - To get contribution of path (1) to (2) in Fig. 8.

GSHANK - To get contribution of path (2) to infinity in Fig. 8.

SQRT

SUBROUTINE EVALUA3 (ERV, EZV, ERH, EZH, EPH)

EVALUA3 returns the ρ, z, and ϕ fields for vertical and horizontal dipoles usin the Hankel function form of the U and V Sommerfeld integrals from Appendix A for a source located below ground and contour of integration No. 3 shown in Fig. 9. The inte gration of paths (1) to (2) to (3) to (4) are performed by routine ROMBERG, and the paths from (4) to (5) to infinity and minus infinity to (1) are performed by routine GSHANK.

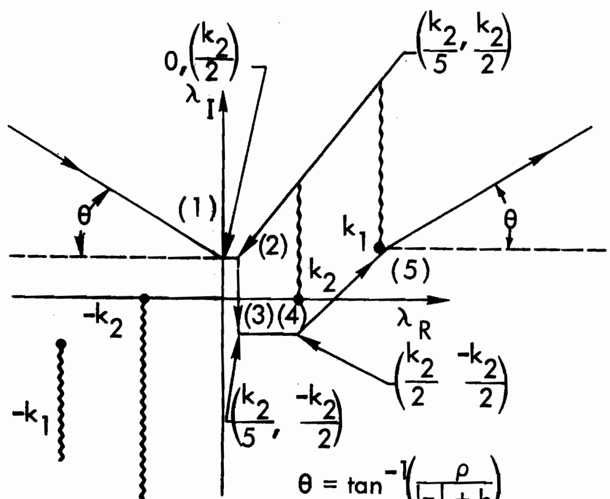

Fig. 9. Contour of integration used by routines EVALUA3 and EVALUB3 for the Hankel function form of the U and V integrals.

INPUT ARGS: None

OUTPUT ARGS:
- ERV - ρ field due to vertical source (COMPLEX).
- EZV - z field due to vertical source (COMPLEX).
- ERH - ρ field due to horizontal source (COMPLEX).
- EZH - z field due to horizontal source (COMPLEX).
- EPH - ϕ field due to horizontal source (COMPLEX).

INPUT COMMON:
- EVALCOM - To get ρ, z, h, propagation constants, etc.
- ROMCOM - To communicate with ROMBERG.

OUTPUT COMMON: CONTOUR- To communicate with routine LAMBDA.
- ITYPE - Set to 1 for linear path (INTEGER).
- A - Lower limit of integration (COMPLEX).
- B - Upper limit of integration (COMPLEX).

SUBROUTINES CALLED:
- ROMBERG - To get contribution of paths (1) to (2) to (3) to (4) in Fig. 5.
- GSHANK - To get contributions of semi-infinite paths in Fig. 9.
- SQRT

SUBROUTINE EVALUB2 (ERV, EZV, ERH, EZH, EPH)

EVALUB2 returns the ρ, z, and ϕ fields for vertical and horizontal dipoles using the Bessel function form of the U and V Sommerfeld integrals from Appendix A for a source located below ground and contour of integration No. 2 in Fig. 8. The integration of path (1) and (2) is performed by routine ROMBERG and the path form (2) to infinity is performed by routine GSHANK.

INPUT ARGS: None.

OUTPUT ARGS:
- ERV - ρ field due to vertical source (COMPLEX).
- EZV - z field due to vertical source (COMPLEX).
- ERH - ρ field due to horizontal source (COMPLEX).
- EZH - z field due to horizontal source (COMPLEX).
- EPH - ψ field due to horizontal source (COMPLEX).

INPUT COMMON:
- EVALCOM - To get ρ, z, h, ϕ propagation constants, etc.
- ROMCOM - To communicate with ROMBERG.

OUTPUT COMMON: CONTOUR - To communicate with routine LAMBDA.
- ITYPE - Set to 1 for linear path (INTEGER).
- A - Set to lower limit of integration (COMPLEX).
- B - Set to upper limit of integration (COMPLEX).

SUBROUTINES CALLED:
- ROMBERG - To get contribution of path (1) to (2) in Fig. 8.
- GSHANK - To get contribution of semi-infinite path in Fig. 8.
- SQRT

SUBROUTINE EVALUB3 (ERV, EZV, ERH, EZH, EPH)

EVALUB3 returns ρ, z, and ϕ fields for vertical and horizontal dipoles using the Hankel function function form of the U and V Sommerfeld integrals from Appendix A for a source located below ground and contour of integration No. 3 shown in Fig. 9. The integration of paths (1) to (2) to (3) to (4) are performed by routine ROMBERG and the paths from (4) to (5) to infinity and minus infinity to (1) are performed by routine GSHANK.

INPUT ARGS: None.

OUTPUT ARGS:
- ERV - ρ field due to vertical source (COMPLEX).
- EZV - z field due to vertical source (COMPLEX).
- ERH - ρ field due to horizontal source (COMPLEX).
- EZH - z field due to horizontal source (COMPLEX).
- EPH - ϕ field due to horizontal source (COMPLEX).

INPUT COMMON: EVALCOM - To get ρ, z, h, propagation constants, etc.

ROMCOM - To communicate with routine ROMBERG.

OUTPUT COMMON: CONTOUR - To communicate with routine LAMBDA.

ITYPE - Set to 1 for linear path (INTEGER).

A - Set to lower limit of integration (COMPLEX).

B - Set to upper limit of integration (COMPLEX).

ROUTINES CALLED:

ROMBERG - To get contribution of paths (1) to (2) to (3) to (4) in Fig. 5.

GSHANK - To get contribution of semi-infinite paths in Fig. 9.

SQRT

FUNCTION FBAR (P)

FBAR returns the value of Sommerfeld's attenuation function for a given Sommerfe numerical distance, P. The expansions used are taken from R. W. P. King.[10]

INPUT ARGS: P - Sommerfeld's numerical distance, either electrical, Pe, or Magnetic Pm.

OUTPUT ARGS: FBAR - which is $\overline{F}(P) = (1 - i\sqrt{\pi P})$ EXP(-P) ERFC($i\sqrt{P}$), where ERFC is the complementary error function.

INPUT COMMON: None.

OUTPUT COMMON: None.

ROUTINES CALLED: CEXP

SQRT

SUBROUTINE FNPLOT (A, B, FCN, K, NAMBCD)

FNPLOT is called by ROMBERG or ROM whenever they fail to converge for a given integral. It calculates the value of the Kth integrand of the function, FCN, for 25 points between the limits A and B. FNPLOT then plots the real part, imaginary part, magnitude, and phase of the values, so that the user may determine why the integral failed to converge. A title, NAMBCD, is printed on each plot for identification.

INPUT ARGS:

A - Lower limit.

B - Upper limit.

FCN - Name of function to be plotted.

K - Integrand number in FCN to be plotted.

NAMBCD - Title to be printed on plot.

OUTPUT ARGS: None.

INPUT COMMON: None.

OUTPUT COMMON: None.

ROUTINE CALLED:

- FCN - To get value of function.
- CARTMM - To determine maximum and minimum of values to be plotted.
- FRAME - Advance plotter one frame.
- MAPX - Establish mapping factors and draw grid lines.
- TRACE - Plot a series of points connected by straight lines.
- SETCH - Position beam of CRT plotter.
- CRTBCD - Point title on plot.
- PLOTEA - Empty plot buffers.

SUBROUTINE GEVALA (ERV, EZV, ERH, EZH, EPH)

GEVALA returns the ρ, z, and ϕ fields due to vertical and horizontal Hertzian dipoles for the free-space Green's functions when the source is located above the surface. For the observer above the surface, the fields are calculated according to the equations shown in Appendix A. For the observer below the surface, the fields are set to zero.

INPUT ARGS: None.

OUTPUT ARGS:
- ERV - ρ field due to vertical source.
- EZV - z field due to vertical source.
- ERH - ρ field due to horizontal source.
- EZH - z field due to horizontal source.
- EPH - ϕ field due to horizontal source,

INPUT COMMON: EVALCOM - To get locations of source and observer in cylindrical coordinate system.

OUTPUT COMMON: None.

ROUTINES CALLED:

- GREENSAOA - To get free-space Green's functions and their derivatives for both source and observer above ground (G_{22}).

SUBROUTINE GEVALB (ERV, EZV, ERH, EZH, EPH)

GEVALB returns the ρ, z, ϕ fields due to vertical and horizontal Hertzian dipoles for the free-space Green's functions when the source is located below the surface. For the observer below the surface, the fields are calculated according to the equations shown in Appendix A. For the observer above the surface, the fields are set to zero.

INPUT ARGS: None.

OUTPUT ARGS: ERV - ρ field due to vertical source.
EZV - z field due to vertical source.
ERH - ρ field due to horizontal source.
EZH - z field due to horizontal source.
EPH - ϕ field due to horizontal source.

INPUT COMMON: EVALCOM - To get locations of source and observer in cylindrical coordinate system.

OUTPUT COMMON: None.

ROUTINES CALLED:

GREENSBOB - To get free-space Green's functions and their derivatives for both source and observer below ground (G_{11}).

SUBROUTINE GFIELDS (T, ETANG)

GFIELDS uses the free-space Green's functions (shown in Appendix A) to calculate the E field, EGANG, tangential to the segment containing the observation point due to a Hertzian dipole located at a point on the axis of the source segment specified by solving for x, y, and z in parametric equations of parameter T.

GFIELDS is a function to be integrated by the subroutine ROM to find the E field tangential to the surface of the observation segment due to a current flowing along the axis of the source segment, since the fields due to a line current are equivalent to the integral of the fields due to a Hertzian dipole as a function of its position along the line. Hence the parameter T becomes the variable of integration specifying the location of the dipole along the line. As T varies from -1 to +1, the position moves from the negative end of the source segment to the positive end.

The solution by the method of moments used by the program WF-LLL2A, uses constant sine, and cosine basis functions requiring GFIELDS to return three values of ETANG — the first for the constant term, the second for the sine term, and the third for the cosine term.

The observation point is always located away from the axis of the observation segment a distance equal to the wire radius, along a line that is described by the intersection of two planes, one perpendicular to the segment, passing through its center, and the other parallel to the x-y plane.

INPUG ARGS: T - Variable of integration; as T is varied from -1 to +1 the position of the dipole moves from the negative end of the source segment to the positive end (REAL).

OUTPUT ARGS: ETANG - E field tangential to observer segment for each of the three basis functions (COMPLEX (3)).

INPUT COMMON: SFCOM - To get the position and orientation of the source and observer segments.

EVALCOM - To get the propagation constants in the upper and lower media.

OUTPUT COMMON:EVALCOM - To pass the locations of the source and observer in the cylindrical coordinate system to the routines GEVALA and GEVALB.

SUBROUTINES CALLED:

GEVALA - To get fields due to vertical and horizontal Hertzian dipoles for a source above ground.

GEVALB - To get fields due to vertical and horizontal Hertzian dipoles for a source below ground.

CSINCOS - Complex sine and cosine.

ATAN2 - Arc tangent.

COS

SIN

SUBROUTINE GREENSAOA
(G22, G22R, G22RZ, G22R2, G22Z2, G21, G21R, G21R2, G21Z2)

GREENSAOA returns the values of free-space Green's functions and their associated derivatives for source and observer above the surface (G_{22}) according to the equations in Appendix A.

INPUT ARGS: None.

OUTPUT ARGS: G22 = Green's function for source and observer above surface.

$$G22R = \frac{\partial G_{22}}{\partial \rho}.$$

$$G22RZ = \frac{\partial^2 G_{22}}{\partial \rho \partial z}.$$

$$G22R2 = \frac{\partial^2 G_{22}}{\partial \rho^2}$$

$$G22Z2 = \frac{\partial^2 G_{22}}{\partial z^2}.$$

G21 = free-space Green's function due to "image" of source.

$$G21R = \frac{\partial G_{22}}{\partial \rho}.$$

$$G21R = \frac{\partial^2 G_{21}}{\partial\rho\partial z} .$$

$$G21R2 = \frac{\partial^2 G_{21}}{\partial \rho^2} .$$

$$G21Z2 = \frac{\partial^2 G_{21}}{\partial z^2} .$$

$$G21Z2 = \frac{\partial^2 G_{21}}{\partial z^2}$$

INPUT COMMON: EVALCOM - To get positions of source and observer in cylindrical coordinate system.

OUTPUT COMMON: None.

ROUTINES CALLED: CEXP - Complex exponential.

SUBROUTINE GREENSBOB

(G11, G11R, G11R, G11RZ, G11R2, G11Z2, G12, G12R, G12RZ, G12R2, G12Z2)

GREENSBOB returns the values of free-space Green's functions and their associated derivatives for source and observer below surface according to the equations shown in Appendix A.

INPUT ARGS: None.

OUTPUT ARGS: G11 = Green's function for source and observer below surface.

$$G11R = \frac{\partial G_{11}}{\partial\rho} .$$

$$G11RZ = \frac{{}^2G_{11}}{\partial\rho\partial z} .$$

$$G11R2 = \frac{\partial^2 G_{11}}{\partial \rho^2} .$$

$$G11Z2 = \frac{\partial^2 G_{11}}{\partial z^2} .$$

G12 = free-space Green's function due to "image" of source.

$$G12R = \frac{\partial G_{11}}{\partial\rho} .$$

$$G12RZ = \frac{\partial^2 G_{12}}{\partial\rho\partial z}.$$

$$G12R2 = \frac{\partial^2 G_{12}}{\partial \rho^2}.$$

$$G12Z2 = \frac{\partial^2 G_{12}}{\partial z^2}.$$

INPUT COMMON: EVALCOM - To get positions of source and observer in cylindrical coordinate system.

OUTPUT COMMON: None.

ROUTINES CALLED: CEXP - Complex exponential.

SUBROUTINE GSHANK

(FCN, STARTER, DELTA, SUM, NANS, SEEDER, IBREAK, BREAK, DELB)

Gshank performs Shank's algorithm[11] to calculate a semi-infinite integral. The integral is expressed as an infinite series consisting of a sum of integrals over equal-spaced intervals as shown:

$$I = \int_A^\infty f(\lambda)\, d\lambda = I_0 + \sum_{n=1}^{\infty} I_i,$$

where $I_0 = \int_A^{A'} f(\lambda)\, d\lambda.$

(I_0 is found before calling GSHANK; its value is passed via the argument SEEDER.)

$$I_i = \int_{A'+(i-1)\Delta}^{A'+i\Delta} f(\lambda)\, d\lambda,$$

where Δ = the interval width (DELTA),

and A' = the lower limit of integration (STARTER).

The implementation of Shank's algorithm in GSHANK calculates S(4), S(6), S(8), etc. until the algorithm converges, where S(N) is found by using the first N + 1 terms of the series to fill the first column of the matrix Q, according to:

$$Q_{0,1} = I_0 = \text{SEED}$$

and

$$Q_{i,1} = Q_{i-1,1} + I_i \qquad \text{for } i = 1, \ldots, N.$$

The remaining columns of the Q matrix are then determined by:

$$Q_{ij} = \frac{(Q_{i+1,j-1})(Q_{i-1,j-1}) - (Q_{i,j-1})^2}{(Q_{i+1,j-1}) + (Q_{i-1,j-1}) - 2(Q_{i,j-1})}$$

for $i = j, \ldots, N - j$,

while $j = 2, \ldots, \frac{N}{2}$.

Since N is an even number, the $\frac{N}{2}$th column of the Q matrix will have only two entries. They are compared to determine if the algorithm has converged. If the percentage difference between the two is less than the convergence criterion (CONCRIT), then the average of the two is taken as the value of the semi-infinite integral, I. If the percentage difference exceeds the convergence criterion, then two more terms are added to the series (N = N + 2), the new entries in the columns of the Q matrix are determined, and the two entries in the new $\frac{N}{2}$th column are compared for convergence. If the percentage difference is too large, N is again incremented and the process repeated. If the process has not converged by S(30), the routine prints a suitable comment and accepts the average of the two values in the last column as the answer.

The function, FCN, actually contains NANS integrands which are integrated in parallel by ROMBERG but are manipulated by Shank's algorithm serially. This is done by storing the results from a call to ROMBERG in a table, A1. Then, when Shank is being applied to the Kth integrand, the appropriate value is taken from A1. If there is no entry in A1, then a call is made to ROMBERG.

GSHANK is used to obtain the values of the semi-infinite integrals for the complex contours in Figs. 8 and 9. The integration of the contour in Fig. 8 is straightforward; the integral of path (1) and (2) is calculated outside GSHANK and passed to it as SEEDER. Then Shank's algorithm is used to find the contribution from path (2) to infinity and return the value of the total integral (from path (1) to (2) to infinity) in SUM for the NANS integrands in FCN.

The integration of the contour in Fig. 9 is more complicated. The contribution of the paths (1) to (4) are found outside GSHANK and passed as SEEDER. Shank's algorithm is then applied to a straight line beginning at (4) and passing through point (5) to infinity If the algorithm converges before point (5) is reached, the value of the total integral is returned in SUM. However, if the point (5) is reached before converging, the contour of integration makes a change in slope and proceeds to infinity at angle, θ. This has been implemented in GSHANK by passing the point (5) as the argument BREAK, and testing the upper limit of I_i to determine when it equals or exceeds BREAK. If it exceeds BREAK, the limit is made equal to BREAK and the integral determined by ROMBERG. This means that ROMBERG has been used to directly determine the integral from point (4) to (5). GSHANK performs the integration on the contour from point (5) to infinity by defining a new I_0 as the old I_0 plus the integral from STARTER to BREAK, defining a new Δ equal to DELB, and starting Shank's algorithm at the beginning. When convergence has occurred, the answer for the total integral from point (4) to (5) to infinity is returned in the table, SUM, for all of the integrands in FCN.

INPUT ARGS:	FCN	Name of function to be integrated (EXTERNAL SUBROUTINE).
	STARTER	- The lower limit of integration A' (COMPLEX).
	DELTA	- The interval width for I_i to be used before the point BREAK is reached (COMPLEX).
	NANS	- The number of integrands in function FCN (Integer, 10 maximum).
	SEEDER	- I , contribution to integral done outside GSHANK (COMPLEX, 10 maximum).
	IBREAK	- 0 means no change in slope of contour of integration; 1 means a change in slope occurs at point BREAK (INTEGER).
	BREAK	- Point in contour of integration where slope changes (COMPLEX).
	DELB	- New interval width for I_i used after point BREAK has been reached (COMPLEX).
OUTPUT ARGS:	SUM	- value of integral to infinity (including contribution by SEEDER), (COMPLEX, 10 maximum).
INPUT COMMON:	ROMCON	- issued to determine the number of evaluations of the integrand
OUTPUT COMMON:	CONTOUR	- used to communicate with routine LAMBDA.
	ITYPE	- 1 for straight line path (INTEGER)
	A	- lower limit of integration (COMPLEX)
	B	- upper limit of integration (COMPLEX)

SUBROUTINES CALLED:

ROMBERG - to get integrals, I_i, of FCN.

QSOLVE - solves for entry, Q(I,J) in Shank's matrix.

SUBROUTINE HANKEL (Z, HO, HOP)

HANKEL returns the value of the zeroth-order Hankel function (of the first kind) and its first derivative ($H_0(z)$, $H_0'(z)$) for complex argument Z, (where Z must be nonzero), via ascending series and asymptotic expansions.[9]

INPUT ARGS: z - Argument of Hankel function (COMPLEX).

OUTPUT ARGS: HO - $H_0(z)$.

HOP - $H_0'(z) = \frac{dH_0(z)}{dZ}$.

INPUT COMMON: None.

OUTPUT COMMON: None.

ROUTINES CALLED: SQRT
- ATAN2 - Arc tangent.
- CSQRT - Complex square root.
- CEXP - Complex exponential.
- EXIT - System routine called when job is finished. Used when HANKEL is called with Z = 0.

SUBROUTINE HSAOA (T, ANS)

HSAOA is the function to be integrated by ROMBERG for a source above ground and an observer above ground using the Hankel function form of the U_{22} and V_{22} integrals shown in Appendix A. This routine is used when the variable of integration, lambda, follows the complex contour shown in Fig. 9 specified by routine EVALUA3. Since ROMBERG can only perform integration between real limits, a change of variable has been made from lambda, which is complex, to the real variable, T.

INPUT ARGS: T - Variable of integration As T is varied from 0 to 1, lambda move from one end of the path in complex plane to the other (REAL).

OUTPUT ARGS:

$$\text{ANS (1)} = \frac{\partial^2 V_{22}}{\partial \rho^2} \text{ (COMPLEX).}$$

$$\text{ANS (2)} = \frac{{}^2 V_{22}}{\partial z^2} .$$

$$\text{ANS (3)} = \frac{\partial^2 V_{22}}{\partial \rho \partial z} .$$

$$\text{ANS (4)} = \frac{\partial^2 V_{22}}{\partial \rho} .$$

$$\text{ANS (5)} = V_{22}.$$

$$\text{ANS (6)} = U_{22}.$$

The subscript 22 indicates both source and observer located above ground (in medium 2).

INPUT COMMON: EVALCOM - To get the positions of the source and observer in the cylindrical coordinate system and the propagation constants for the upper and lower media.

OUTPUT COMMON: None

SUBROUTINES CALLED:

LAMBDA - To find the value of lambda in the complex plane as a function of parameter T.

CROOTS - To return the correct complex square root for the vertical branch cuts used in this formulation.

HANKEL - To get the value of the Hankel function.

SUBROUTINE HSAOB (T, ANS)

HSAOB is the function to be integrated by ROMBERG for a source above ground and an observer below ground using the Hankel function form of the U_{21} and V_{21} integrals shown in Appendix A. This routine is used when the variable of integration, lambda, follows the complex contour shown in Fig. 9 specified by routine EVALUA3. Since ROMBERG can only perform integration between real limits, a change of variable has been made from lambda, which is complex, to the real variable, T.

INPUT ARGS: T - Variable of integration. As T varies from 0 to 1, lambda moves from one end of the path in complex plane to the other (REAL).

OUTPUT ARGS:

$$ANS\ (1) = \frac{\partial^2 V_{21}}{\partial \rho^2} .$$

$$ANS\ (2) = \frac{\partial^2 V_{21}}{\partial \rho^2} .$$

$$ANS\ (3) = \frac{\partial^2 V_{21}}{\partial \rho \partial z} .$$

$$ANS\ (4) = \frac{\partial V_{21}}{\partial \rho} .$$

$$ANS\ (5) = V_{21} .$$

$$ANS\ (6) = U_{21} .$$

$$ANS\ (7) = \frac{\partial^2 V_{21}}{\partial \rho \partial h} .$$

The subscript 21 indicates a source above ground and an observer below.

INPUT COMMON: EVALCOM - To get the positions of the source and observer in the cylindrical coordinate system and the propagation constan for the upper and lower media.

OUTPUT COMMON: None.

ROUTINES CALLED:

LAMBDA - To find the value of lambda in the complex plane as a function of the parameter T.

CROOTS - To return correct complex square root for the vertical branch cuts used in this formulation.

HANKEL - To get the value of the HANKEL function.

SUBROUTINE HSBOA (T, ANS)

HSBOA is the same as HSAOB except that the formulation is for a source below ground and an observer above ground, using the Hankel function form of the U_{12} and V_{12} integrals shown in Appendix A.

INPUT ARGS: T - Variable of integration. As T varies from 0 to 1, lambda moves from one end of the path in complex plane to the other (REAL).

OUTPUT ARGS:

$$\text{ANS (1)} = \frac{\partial^2 V_{12}}{\partial \rho^2} \text{ (COMPLEX)}.$$

$$\text{ANS (2)} = \frac{\partial^2 V_{12}}{\partial z^2} .$$

$$\text{ANS (3)} = \frac{\partial^2 V_{12}}{\partial \rho \partial z} :$$

$$\text{ANS (4)} = \frac{\partial V_{12}}{\partial \rho} .$$

$$\text{ANS (5)} = V_{12} .$$

$$\text{ANS (6)} = U_{12} .$$

$$\text{ANS (7)} = \frac{{}^2 V_{12}}{\partial \rho \partial h} .$$

The subscript 12 indicates a source below ground and an observer above ground.

INPUT COMMON: EVALCOM - To get the positions of the source and observer in the cylindrical coordinate system and the propagation constants for the upper and lower media.

OUTPUT COMMON: None.

ROUTINES CALLED:

LAMBDA - To find the value of lambda in the complex plane as a function of the parameter T.

CROOTS - To return correct complex square root for the vertical branch cuts used in this formation.

HANKEL - To get the value of the Hankel function.

SUBROUTINE HSBOB (T, ANS)

HSBOB is the same as HSAOA except that the formulation is for both the source and observer below ground, using the Hankel function form of the U_{11} and V_{11} integrals shown in Appendix A.

INPUT ARGS: T - variable of integration. As T varies from 0 to 1, lambda moves from one end of path in the complex plane to the other (REAL).

OUTPUT ARGS:

$$ANS\ (1) = \frac{\partial^2 V_{11}}{\partial \rho^2} .$$

$$ANS\ (2) = \frac{\partial^2 V_{11}}{\partial z^2} .$$

$$ANS\ (3) = \frac{\partial^2 V_{11}}{\partial \rho \partial z} .$$

$$ANS\ (4) = \frac{\partial V_{11}}{\partial \rho} .$$

$$ANS\ (5) = V_{11} .$$

$$ANS\ (6) = U_{11} .$$

The subscript 11 indicates both source and observer in the ground (medium 1).

INPUT COMMON: EVALCOM - To get the positions of the source and observer in the cylindrical coordinate system and the propagation constants for the upper and lower media.

OUTPUT COMMON: None.

ROUTINES CALLED:

LAMBDA - To find the value of lambda in the complex plane as a function of the parameter T.

CROOTS - To return the correct complex square root for the vertical branch cuts used in this formulation.

HANKEL - To get the value of the Hankel function.

SUBROUTINE INFINITY

(FCN, START, DELT, ANS, N, SEED, IBREAK, BREAK, DELB)

INFINITY calculates the values of a semi-infinite integral, I, expressing it as an infinite series consisting of a sum of integrals over equal-spaced intervals.

$$I = \int_{A}^{\infty} f(\lambda)\, d\lambda = I_0 + \sum_{i=1}^{\infty} I_i ,$$

where

$$I_0 = \int_{A}^{A'} f(\lambda)\, d\lambda ,$$

$$I_i = \int_{A+(i-1)\Delta}^{A'+i\Delta} f(\lambda)\, d\lambda ,$$

Δ = interval width (DELT, delb), and

A' = lower limit of integration (START).

I_0 is calculated before calling INFINITY, its value is passed via the argument SEED. Th value of each interval, I_i, is determined by subroutine ROMBERG. The infinite series is considered to have converged when the fifth consecutive interval, I_i, has contributed less than 10^{-3} of the sum of all the previous intervals (including I_0). If convergence has not been achieved when $A' + i\Delta$ equals or exceeds BREAK, a new interval width, Δ, is chosen equal to DELB, similar to the routine GSHANK.

INFINITY is to be substituted for GSHANK whenever a case arises where GSHANK appears to be falsely convering.

INPUT ARGS: FCN - Function to be integrated (EXTERNAL SUBROUTINE).

START - Initial lower limit of integration (COMPLEX).

DELT - Interval width for I_i used before the point BREAK is reached (COMPLEX).

N - Number of integrands in FCN to be done in parallel (INTEGER, 10 maximum).

SEED - Contribution to integral done outside INFINITY (COMPLEX, 10 maximum).

IBREAK - 0; there is no break (bend) in contour of integration (INTEGER).

- 1; there is a break (bend) in contour at point BREAK.

BREAK - Point at which new increment between intervals is to be used (COMPLEX).

DELB - New increment between intervals to be used after reaching BREAK (COMPLEX).

OUTPUTS ARGS: ANS - Value of integral to infinity including contribution due to SEED (COMPLEX).

INPUT COMMON: ROMCOM - Used to determine the number of evaluations of the integrand.

OUTPUT COMMON:

CONTOUR - To communicate with routine LAMBDA.

ITYPE - Set to 1 for straight-line path (INTEGER).

A - Set to lower limit of integration (COMPLEX).

B - Set to upper limit of integration (COMPLEX).

ROUTINES CALLED: ROMBERG - To do integration for interval I_i of function FCN.

SUBROUTINE LAMBDA (T, XL, DXL)

LAMBDA returns the value of λ, XL, and dλ/dt, DXL, as functions of the parameter, T. This routine is necessary since ROMBERG can only perform integration between real limits, while the contours of integration used for the evaluation of the Sommerfeld integrals require integration between complex limits. The integral over the path connecting the points, A and B, in the complex plane can be expressed as a function of the real parameter, T, according to:

$$I = \int_A^B f(\lambda)\, d\lambda = \int_0^1 f(\lambda(t))\, \frac{d\lambda}{dt}\, dt,$$

where

A and B are the complex limits of integration,
λ is the complex variable of integration, and
t is the real variable of integration.

For a contour consisting of a straight line between A and B,

$$\lambda(t) = A + t\ (B - A),$$

$$\frac{d\lambda(t)}{dt} = B - A.$$

For a contour consisting of a circular arc centered at A and starting at B,

$$\lambda(t) = A + |B - A|\varepsilon^{j[2\pi t + \angle(B - A)]},$$

$$\frac{d\lambda(t)}{dt} = j2\pi\ |B - A|\varepsilon^{j[2\pi t + \angle(B - A)]}.$$

INPUT ARGS: T - REAL variable of integration. As T varies from 0 to 1, λ va
from complex lower limit, A, to complex upper limit, B.

OUTPUT ARGS: XL - λ.

DXL - dλ/dt.

INPUT COMMON: CONTOUR

ITYPE - 1 for a straight line between A and B.
- 2 for a circle centered at A, starting at B (INTEGER).

A - Lower limit (COMPLEX).

B - Upper limit (COMPLEX).

OUTPUT COMMON: None.

ROUTINES CALLED:

SQRT

CANG - Phase angle of complex number.

CEXP - Complex exponential.

SUBROUTINE NFIELDS (T, ETANG)

NFIELDS uses Norton's equations (shown in Appendix B) to calculate the E fiel ETANG, tangential to the segment containing the observation point, due to a Hertzian dipole located at a point on the axis of the source segment specified by solving for x y, and z in parametric equations of parameter T.

NFIELDS is a function to be integrated by the subroutine ROM to find the E fie tangential to the surface of the observation segment due to a current flowing along th axis of the source segment, since the fields due to a line current are equivalent to t integral of the fields due to a Hertzian dipole as a function of its position along th line. Thus the parameter T becomes the variable of integration specifying the locatio of the dipole along the line. As T varies from -1 to +1, the position moves from the negative end of the source segment to the positive end.

The solution by the method of moments used by the program WF-LLL2A, uses constant, sine, and cosine basis functions requiring NFIELDS to return three values of ETANG — the first for the constant term, the second for the sine term, and the third for the cosine term.

The observation point is always located away from the axis of the observation segment a distance equal to the wire radius, along a line that is described by the intersection of two planes, one perpendicular to the segment passing through its center, and the other parallel to the x-y plane.

INPUT ARGS: T - Variable of integration. As T is varied from -1 to +1, the position of the dipole moves from the negative end of the source segment to the positive end (REAL).

OUTPUT ARGS: ETANG - E field tangential to observer segment for each of the three basis functions (COMPLEX (3)).

INPUT COMMON: SFCOM - To get the position and orientation of the source and observer segments.

EVALCOM - To get the propagation constants in the upper and lower media.

OUTPUT COMMON: EVALCOM - To pass the locations of the source and observer in the cylindrical coordinate system to the routine NORTON.

SUBROUTINES CALLED:

NORTON - To get the fields due to vertical and horizontal Hertzian dipoles using Norton's equations.

CSINCOS - Complex sine and cosine.

ATAN2 - Arc tangent.

COS

SIN

SUBROUTINE NORTON (ERV, EZV, ERH, EZH, EPH)

NORTON returns the ρ, z and ϕ fields for vertical and horizontal dipoles using Norton's equations for source and observer above ground shown in Appendix B.

INPUT ARGS: None.

OUTPUT ARGS: ERV - ρ field due to vertical source (COMPLEX).

EZH - z field due to vertical source (COMPLEX).

ERH - ρ field due to horizontal source (COMPLEX).

EZH - z field due to horizontal source (COMPLEX).

EPH - ϕ field due to horizontal source (COMPLEX).

INPUT COMMON:

NORTSETUP - To get frequency and ground parameters

EVALCOM - To get position of source and observer in cylindrical coordinate system.

OUTPUT COMMON:

CXNN to pass XNN - the index of refraction squared (for use by routines RE and RH).

SUBROUTINES CALLED:

RE - E-field reflection coefficient.

RH - H-field reflection coefficient.

FBAR - Sommerfield's attenuation function.

SQRT

CEXP - Complex exponent.

CSQRT - Complex square root.

ASIN - Arc sine.

FUNCTION QSOLVE (Q,I,J)

QSOLVE solves for the Q_{ij} entry in Shank's matrix, according to the formula:

$$Q_{ij} = \text{QSOLVE} = \frac{(Q_{i+1,\ j-1})(Q_{i-1,\ j-1}) - (Q_{i,\ j-1})^2}{(Q_{i+1,\ j-1}) + (Q_{i-1,\ j-1}) - 2(Q_{i,\ j-1})} .$$

INPUT ARGS: Q - Shank's matrix.

I - Row index.

J - Column index.

OUTPUT ARGS: QSOLVE = Q_{ij}

INPUT COMMON: None.

OUTPUT COMMON: None.

FUNCTION RE(T)

RE returns the reflection coefficient for a plane wave incident at angle theta, T, with the E field perpendicular to the plane of incidence according to:

$$RE = \frac{\cos\theta - \sqrt{N^2 - \sin^2\theta}}{\cos\theta + \sqrt{N^2 - \sin^2\theta}} ,$$

where

N is the index of refraction in lower medium, and
θ is the angle of incidence.

INPUTS ARGS: T - Angle of incidence, theta (REAL).

OUTPUT ARGS: RE - Reflection coefficient (COMPLEX).

INPUT COMMON: CXNN - To get XNN index of refraction squared (COMPLEX).

OUTPUT COMMON: None.

SUBROUTINES CALLED:

SIN

COS

CSQRT - Complex square root.

FUNCTION RH(T)

RH returns the reflection coefficient for a plane wave incident at angle theta, T, with the H field perpendicular to the plane of incidence according to:

$$RH = \frac{N^2\cos\theta - \sqrt{N^2 - \sin^2\theta}}{N^2\cos\theta + \sqrt{N^2 - \sin^2\theta}} ,$$

where

N is the index of refraction in lower media, and
θ is angle of incidence.

INPUT ARGS: T - Angle of incidence, theta (REAL).

OUTPUT ARGS: RH - Reflection coefficient (COMPLEX).

INPUT COMMON: CXNN - To get XNN the index of refraction squared (COMPLEX).

OUTPUT COMMON: None.

SUBROUTINES CALLED:

SIN

COS

CSQRT - Complex square root.

SUBROUTINE ROM (A, B, FCN, N, SUM, RX)

ROM returns the integral of function, FCN, between real limits, A and B, according to the adaptive ROMBERG scheme of Miller.[12] There are N integrands to be integrated in parallel, specified by subroutine FCN. The results are returned in SUM. RX is the criterion for determining convergence and is usually set to 10^{-4},

INPUT ARGS: A - Lower limit (REAL).
B - Upper limit (REAL).
FCN - Function to be integrated (EXTERNAL SUBROUTINE).
N - Number of integrands of FCN (INTEGER, 10 maximum).
RX - Convergence criterion (REAL).

OUTPUT ARGS: SUM - Result (COMPLEX (10)).

INPUT COMMON: None.

OUTPUT COMMON: RCOM - To indicate number of evaluations of FCN needed.

ROUTINES CALLED:

FCN - Function to be integrated. FCN must have arguments (T, ANS), where T is the variable of integration (set by ROM), and ANS becomes the complex values of the integrand, FCN, at T. ANS can have a maximum dimension of 10.

TESTC - To test for convergence (COMPLEX FUNCTION).

FNPLOT - To plot integrand when convergence could not be achieved.

SUBROUTINE ROMBERG (A, B, FCN, N, SUM, NX)

ROMBERG returns the integrand of function, FCN, between real limits, A and B, according to the adaptive Romberg scheme of Miller.[12] There are N integrands to be done in parallel, specified by subroutine FCN. The results are returned in SUM. In order to prevent false convergence for oscillatory integrands, the argument NX is used to specify the largest interval size that ROMBERG is to consider, expressed as a fraction of the interval between A and B (i.e., Δ_{max} = (A - B)/NX). The value of NX is chosen so that the largest interval contains only one or two oscillations of the integrand. For smoothly varying functions, a value of NX equal to 2 will give convergence with the fewest number of evaluations of FCN.

INPUT ARGS: A - Lower limit (REAL).
B - Upper limit (REAL).
FCN - Function to be integrated (EXTERNAL SUBROUTINE).
N - Number of integrands in FCN (INTEGER, 10 maximum).
NX - Controls interval size (INTEGER).

OUTPUT ARGS: SUM - Result (COMPLEX, 10 maximum).

INPUT COMMON: None.

OUTPUT COMMON: ROMCOM - To indicate number of evaluations of FCN needed.

ROUTINES CALLED:

FCN - To get value of integrands (EXTERNAL SUBROUTINE). FCN must have arguments (T, ANS), where T is the variable of integration (set by ROMBERG), and ANS becomes the complex value of the integrand, FCN at T. ANS can have a maximum dimension of 10.

TESTC - To test for convergence (COMPLEX FUNCTION).

FNPLOT - To plot integrand when convergence could not be achieved.

SUBROUTINE SAOA (T, ANS)

SAOA is the function to be integrated by ROMBERG for a source above ground and an observer above ground, using the Bessel function form of the U_{22} and V_{22} integrals shown in Appendix A. This routine is used when the variable of integration, lambda, follows the complex contour shown in Fig. 8, specified by routine EVALUA2. Since ROMBERG can only perform integration between real limits, a change of variable has been made from lambda, which is complex, to the real variable, T.

INPUT ARGS: T - The variable of integration. As T varies from 0 to 1, lambda moves from one end of the path in the complex plane to the other (REAL).

OUTPUT ARGS: ANS (1) = $\dfrac{\partial^2 V_{22}}{\partial \rho^2}$ (COMPLEX).

ANS (2) = $\dfrac{\partial^2 V_{22}}{\partial z^2}$.

ANS (3) = $\dfrac{\partial^2 V_{22}}{\partial \rho \partial z}$.

ANS (4) = $\dfrac{\partial V_{22}}{\partial \rho}$.

ANS (5) = V_{22} .

ANS (6) = U_{22} .

The subscript 22 indicates that both the source and observer are located above ground (in medium 2).

INPUT COMMON: EVALCOM - To get the positions of the source and observer in the cylindrical coordinate system and the propagation constants for the upper and lower media.

OUTPUT COMMON: None.

SUBROUTINES CALLED:

LAMBDA - Find the value of lambda in the complex plane as a function of the parameter T.

BESSEL - To get the value of the Bessel function.

SUBROUTINE SAOB (T, ANS)

SAOB is the function to be integrated by ROMBERG for a source above ground and an observer below ground, using the Bessel function form of the U_{21} and V_{21} integrals show in Appendix A. This routine is used when the variable of integration, lambda, follows the complex contour shown in Fig. 8 specified by routine EVALUA3. Since ROMBERG can only perform integration between real limits, a change of variable has been made from lambda, which is complex, to the real variable T.

INPUT ARGS: T - The variable of integration. As T varies from 0 to 1, lambda moves from one end of the path in the complex plane to the other (REAL).

OUTPUT ARGS: ANS (1) = $\frac{\partial^2 V_{21}}{\partial \rho^2}$ (COMPLEX).

$$\text{ANS (2)} = \frac{\partial^2 V_{21}}{\partial z^2} .$$

$$\text{ANS (3)} = \frac{\partial^2 V_{21}}{\partial \rho \partial z} .$$

$$\text{ANS (4)} = \frac{\partial V_{21}}{\partial \rho} .$$

$$\text{ANS (5)} = V_{21} .$$

$$\text{ANS (6)} = U_{21} .$$

$$\text{ANS (7)} = \frac{\partial^2 V_{21}}{\partial \rho \partial h} .$$

The subscript 21 indicates a source above ground and the observer below.

INPUT COMMON: EVALCOM - To get the positions of the source and observer in the cylindrical coordinate system and the propagation constants for the upper and lower media.

OUTPUT COMMON: None.

ROUTINES CALLED:

LAMBDA - To find the value of lambda in the complex plane as a function of the parameter T.

BESSEL - To find the value of the Bessel function.

SUBROUTINE SBOA (T, ANS)

SBOA is the same as SAOB, except that the formulation is for a source below ground and an observer above ground, using the Bessel function form of the U_{12} and V_{12} integrals shown in Appendix A.

INPUT ARGS: T - The variable of integration. As T varies from 0 to 1, lambda moves from one end of the path in the complex plane to the other (REAL).

OUTPUT ARGS: ANS (1) = $\frac{\partial^2 V_{12}}{\partial \rho^2}$ (COMPLEX).

ANS (2) = $\frac{\partial^2 V_{12}}{\partial z^2}$.

ANS (3) = $\frac{\partial^2 V_{12}}{\partial \rho \partial z}$.

ANS (4) = $\frac{\partial V_{12}}{\partial \rho}$.

ANS (5) = V_{12} .

ANS (6) = U_{12} .

ANS (7) = $\frac{\partial^2 V_{12}}{\partial \rho \partial h}$.

The subscript 12 indicates a source below ground and an observer above ground.

INPUT COMMON: EVALCOM - To get the positions of the source and observer in the cylindrical coordinate system and the propagation constants for the upper and lower media.

OUTPUT COMMON: None.

ROUTINES CALLED:

LAMBDA - To find the value of lambda in the complex plane as a function of the parameter T.

BESSEL - To find the value of the Bessel function.

SUBROUTINE SBOB (T, ANS)

SBOB is the same as SAOA, except that the formulation is for both the source and observer below ground, using the Bessel function form of the U_{11} and V_{11} integrals shown in Appendix A.

INPUT ARGS: T - The variable of integration. As T varies from 0 to 1, lambda moves from one end of the path in the complex plane to the other (REAL).

OUTPUT ARGS: ANS (1) = $\frac{\partial^2 V_{11}}{\partial \rho^2}$ (COMPLEX).

ANS (2) = $\frac{\partial^2 V_{11}}{\partial z^2}$.

ANS (3) = $\frac{\partial^2 V_{11}}{\partial \rho \partial z}$.

ANS (4) = $\frac{\partial V_{11}}{\partial \rho}$.

ANS (5) = V_{11} .

ANS (6) = U_{11} .

The subscript 11 indicates both source and observer in the ground (medium 1).

INOUT COMMON: EVALCOM - To get the positions of the source and observer in the cylindrical coordinate system and the propagation constants for the upper and lower media.

OUTPUT COMMON: None.

ROUTINES CALLED:

LAMBDA - To find the values of lambda in the complex plane as a function of the parameter T.

BESSEL - To find the value of the Bessel function.

SUBROUTINE SETUP (FRQ, EPSR1, SIG1, EPSR2, SIG2)

SETUP initializes the variables in common blocks EVALCOM and NORTSETUP. SETUP must be called before any of the level 1 routines of Fig. 1 every time the frequency or media parameters change.

INPUT ARGS: FRQ - Frequency in Hertz (REAL).

EPSR1 - ε_r for lower medium (REAL).

SIG1 - σ for lower medium (REAL).

EPSR2 - ε_r for upper medium (REAL).

SIG2 - σ for upper medium (REAL).

OUTPUT ARGS: None.

INPUT COMMON: None.

OUTPUT COMMON: EVALCOM - Used by many routines as input common.

CJ = $j = \sqrt{-1}$, (COMPLEX).

CK1 = propagation constant in lower medium, k_1, (COMPLEX).

CK1SQ = $k_1^2 = \omega^2\mu_0\varepsilon_0\varepsilon_{r_1} + j\omega\mu_0\sigma_2$, (COMPLEX).

CK2 = propagation constant in upper medium, k_2, (COMPLEX).

CK2SQ = $k_2^2 = \omega^2\mu_0\varepsilon_0\varepsilon_{r_2} + j\omega\mu_0\sigma_2$, (COMPLEX).

COEE = constant multiplier of the Sommerfeld integrals $= \frac{j\omega\mu_0}{4\pi}$, (COMPLEX).

COEH = constant multiplier of the Sommerfeld integrals $= \frac{-1}{4\pi}$ (COMPLEX).

NORTSETUP - Used by NORTON as input common.

F = frequency in Hertz, (REAL).

ER = ε_r in lower medium (upper medium assumed to be free space), (REAL).

SIG = σ in lower medium (upper medium assumed to be free space), (REAL).

ROUTINE CALLED:

CSQRT - Complex square root.

SUBROUTINE SFIELDS (T, ETANG)

SFIELDS uses the results of the U and V Sommerfeld integrals, shown in Appendix to calculate the E field, ETANG, tangential to the segment containing the observation point due to a Hertzian dipole located at a point on the axis of the source segment specified by solving for x, y, and z in parametric equations of parameter T.

SFIELDS is a function to be integrated by the subroutine ROM to find the E field tangential to the surface of the observation segment due to a current flowing alon the axis of the source segment since the fields due to a line current are equivalent to the integral of the field due to a Hertzian dipole as a function of its position along the line. Thus the parameter T becomes the variable of integration specifying the location of the dipole along the line. As T varies from -1 to +1, the position moves from the negative end of the source segment to the positive end.

The solution by the method of moments used by the program WF-LLL2A uses constant, sine, and cosine basis functions, requiring SFIELDS to return three values of ETANG, the first for the constant term, the second for the sine term, and the third for the cosine term.

The observation point is always located away from the axis of the observation segment a distance equal to the wire radius, along a line that is described by the intersection of two planes, one perpendicular to the segment passing through its center, and the other parallel to the x-y plane.

INPUT ARGS: T - Variable of integration. As T is varied from -1 to +1, the position of the dipole moves from the negative end of the source to the positive end (REAL).

OUTPUT ARGS: ETANG - E field tangential to observer segment for each of the three basis functions (COMPLEX (3)).

INPUT COMMON: SFCOM - To get the position and orientation of the source segments.

EVALCOM - To get the propagation constants in the upper and lower media.

OUTPUT COMMON: EVALCOM - To pass the locations of the source and observer in the cylindrical coordinate system to the EVALU routines.

SUBROUTINES CALLED:

EVALUA2 - Get the fields for a source above ground, using Bessel form of the U and V integrals.

EVALUB2 - Get the fields for a source below ground, using Bessel form of the U and V integrals.

EVALUA3 - Get the fields for source above ground, using Hankel form of the U and V integrals.

EVALUB3 - Get fields for source below ground, using Hankel form of the U and V integrals.

CSINCOS - Complex sine and cosine.

ATAN2 - Arc tangent.

COS

SIN

FUNCTION TESTC (F1, F2)

TESTC is the function used by ROMBERG and ROM to determine if an integral has converged.

INPUT ARGS: F1, F2 } Two successive approximations to the integral for which the percentage difference is to be determined (COMPLEX).

OUTPUT ARGS: TESTC (COMPLEX),

where

Re [TESTC] = (|Re[F1) - Re[F2]|)/(Re[F2])

Im [TESTC] = (|Im[F1] - Im[F2])/(Im[F2]).

INPUT COMMON: None.

OUTPUT COMMON: None.

ROUTINES CALLED: None.

ACKNOWLEDGMENTS

The authors wish to thank their colleqgues at Lawrence Livermore Laboratory for a number of helpful discussions and particularly to thank Edmund K. Miller for his timely observations and Frederick J. Deadrick for his cogent comments.

Appendix A — U and V Sommerfeld Integrals

MATHEMATICAL FORMULAE

The mathematical formulae describing the fields due to either a vertical electric or a horizontal electric dipole located either above or below the ground-air interface are given below. These formulae were obtained from Baños,[5] and are expressed in his notation (with the exception that the cylindrical radius is herein expressed as ρ, rather than r) as shown in Fig. 10. The air is assumed to be medium 2, the ground medium 1. The current moment of the source is denoted as p, where $p = I\ell$ ampere metre. The operating frequency is $\omega = 2\pi f$ rad/s. The propagation constants in the ground and in the air are, respecti

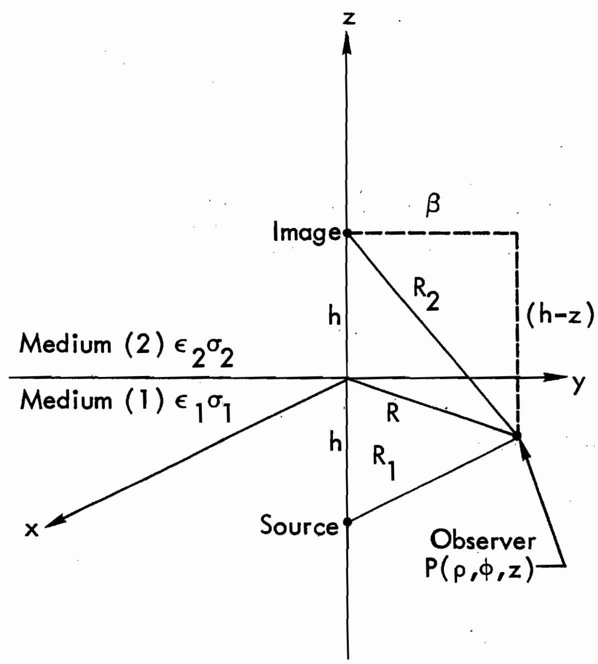

Fig. 10. Figure defines notation used by Baños in formulating Sommerfeld integrals. The source is located on z axis; the observer is at point P(ρ,ϕ,z). The case for a source located in the lower medium is depicted.

$$k_1^2 = \omega^2\mu_0\varepsilon_0\varepsilon_r = i\omega\mu_0\sigma,$$

and

$$k_2^2 = \omega^2\mu_0\varepsilon_0,$$

where μ is the free-space permeability, ε_C is the free-space permittivity, ε_r is the relative dielectric constant of the ground and σ is the conductivity of the ground. The usual cylindrical coordinate system notation (ρ′,ϕ,z) is used. $E_{1\rho}^H$ designates t radial electric field intensity in region (ground) due to a horizontal electric curr source. Note that the location of the source is not explicitly designated using this notation. The source height is designated as h, and h is assumed to be positive, without regard to whether the source is buried or elevated. The receiver height is designated as z, where z is negative if the receiver is buried, and z is positive if the receiver is elevated. The radial separation between source and receiver is ρ. The source is assumed to lie in the x-z plane, and the receiver lies in a plane oriented at an angle of ϕ relative to the x-z plane. With this notation, the electromagnetic fields due to elevated and buried electric current sources can be expressed in terms of what are designated as U and V integrals (defined below).

The names of the subroutines which calculate the individual terms of each expression are indicated below in the corresponding term.

VERTICAL ELECTRIC DIPOLE IN MEDIUM 1 (GROUND)

Source Below, Observer Below Interface

$$E^V_{1\rho} = \underbrace{\frac{i\omega\rho\mu_0}{4\pi k_1^2}\frac{\partial^2}{\partial\rho\partial z}\{(G_{11} - G_{12})\}}_{\text{GEVALB}} + \underbrace{\frac{i\omega\rho\mu_0}{4\pi k_1^2}\left\{\frac{\partial^2}{\partial\rho\partial z}\quad k_2^2 V_{11}\right\}}_{\text{EVALUAB2, EVALUAB3}}.$$

$$E^V_{1z} = \underbrace{\frac{i\omega\rho\mu_0}{4\pi k_1^2}\left\{\left(\frac{\partial^2}{\partial z^2}\right) + k_1^2 \;(G_{11} - G_{12})\right\}}_{\text{GEVALB}} + \underbrace{\frac{i\omega\rho\mu_0}{4\pi k_1^2}\left\{\left(\frac{\partial^2}{\partial z^2} + k_1^2\right) \; k_2^2 V_{11}\right\}}_{\text{EVALUB2, EVALUB3}}.$$

Source Below, Observer Above Interface

$$E^V_{2\rho} = \underbrace{\frac{i\omega\rho\mu_0}{4\pi}\left\{\frac{\partial^2 V_{12}}{\partial\rho\partial z}\right\}}_{\text{EVALUB2, EVALUB3}}$$

$$E^V_{2z} = \underbrace{\frac{i\omega\rho\mu_0}{4\pi}\left\{\left(\frac{\partial^2}{\partial z^2} \quad + \quad k_2^2 \; V_{12}\right\}\right.}_{\text{EVALUB2, EVALUB3}}$$

VERTICAL ELECTRIC DIPOLE IN MEDIUM 2 (AIR)

Source Above, Observer Above Interface

$$E^V_{2\rho} = \underbrace{\frac{i\omega\rho\mu_0}{4\pi k_2^2}\left\{\frac{\partial^2}{\partial\rho\partial z}\quad (G_{22} - G_{21})\right\}}_{\text{GEVELA}} + \underbrace{\frac{i\omega\rho\mu_0}{4\pi k_2^2}\left\{\frac{\partial^2}{\partial\rho\partial z}\quad k_1^2 V_{22}\right\}}_{\text{EVALUA2, EVALUA3}}.$$

$$E^V_{2z} = \underbrace{\frac{i\omega\rho\mu_0}{4\pi k_2^2}\left\{\frac{\partial^2}{\partial z^2} + k_2^2\right)(G_{22} - G_{21})\Big\}}_{\text{GEVELA}} + \underbrace{\frac{i\omega\rho\mu_0}{4\pi k_2^2}\left\{\left(\frac{\partial^2}{\partial z^2} + k_2^2\right) k_1^2 V_{22}\right\}}_{\text{EVALUA2, EVALUA3}}.$$

Source Above, Observer Below Interface

$$\underbrace{E_{1\rho} = \frac{i\omega p\mu_0}{4\pi}\left\{\frac{\partial^2 V_{21}}{\partial\rho\partial z}\right\}}_{\text{EVALUA2, EVALUA3}}.$$

$$\underbrace{E_{1z} = \frac{i\omega p\mu_0}{4\pi}\left\{\left(\frac{\partial^2}{\partial z^2} + k_1^2\right)V_{21}\right\}}_{\text{EVALUA2, EVALUA3}}.$$

HORIZONTAL ELECTRIC DIPOLE IN MEDIUM 1 (GROUND)

Source Below, Observer Below Interface

$$E^H_{1\rho} = \frac{i\omega p\mu_0}{4\pi k_1^2}\cos\phi\left\{\frac{\partial^2}{\partial\rho^2}\left[G_{11} - G_{12} + k_1^2 V_{11}\right] + k_1^2[G_{11} - G_{12} + U_{11}]\right\}$$

$$= \underbrace{\cos\phi}_{\text{GFIELDS}}\underbrace{\left[\frac{i\omega p\mu_0}{4\pi k_1^2}\left\{\left(\frac{\partial^2}{\partial\rho^2} + k_1^2\right)(G_{11} - G_{12})\right\}\right]}_{\text{GEVALB}} + \underbrace{\cos\phi}_{\text{SFIELDS}}\underbrace{\left[\frac{i\omega p\mu_0}{4\pi k_1^2}\left\{\frac{\partial^2}{\partial\rho^2}\left(k_1^2 V_{11}\right) + k_1^2 U_{11}\right\}\right]}_{\text{EVALUB2, EVALUB3}}.$$

$$E^H_{1\phi} = \frac{-i\omega p\mu_0}{4\pi k_1^2}\sin\phi\left\{\frac{1}{\rho}\frac{\partial}{\partial\rho}\left[G_{11} - G_{12} + k_1^2 V_{11}\right] + k_1^2[G_{11} - G_{12} + U_{11}]\right\}$$

$$= \underbrace{\sin\phi}_{\text{GFIELDS}}\underbrace{\left[\frac{-i\omega p\mu_0}{4\pi k_1^2}\left\{\frac{1}{\rho}\left(\frac{\partial}{\partial\rho} + k_1^2\right)(G_{11} + G_{12})\right\}\right]}_{\text{GEVALB}} + \underbrace{\cos\phi}_{\text{SFIELDS}}\underbrace{\left[\frac{-i\omega p\mu_0}{4\pi k_1^2}\left\{\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(k_1^2 V_{11}\right) + k_1^2 U\right.\right.}_{\text{EVALUB2, EVALUB3}}$$

$$E^H_{1z} = \frac{i\omega p\mu_0}{4\pi k_1^2}\cos\phi\left\{\frac{\partial^2}{\partial z\partial\rho}\left[G_{11} + G_{12} - k_2^2 V_{11}\right]\right\}$$

$$= \underbrace{\cos\phi}_{\text{GFIELDS}}\underbrace{\left[\frac{i\omega p\mu_0}{4\pi k_1^2}\left\{\frac{\partial^2}{\partial z\partial\rho}(G_{11} + G_{12})\right\}\right]}_{\text{GEVALB}} + \underbrace{\cos\phi}_{\text{SFIELDS}}\underbrace{\left[\frac{i\omega p\mu_0}{4\pi k_1^2}\left\{\frac{\partial^2}{\partial z\partial\rho}\left(-k_2^2 V_{11}\right)\right\}\right]}_{\text{EVALUA2, EVALUB3}}.$$

Source Below, Observer Above Interface

$$E_{2\rho}^{H} = \underbrace{\cos\phi}_{\text{SFIELDS}} \underbrace{\left[\frac{+i\omega p\mu_0}{4\pi}\left\{\frac{\partial^2 V_{12}}{\partial\rho^2} + U_{12}\right\}\right]}_{\text{EVALUB2, EVALUB3}}$$

$$E_{2\phi}^{H} = \underbrace{\sin\phi}_{\text{SFIELDS}} \underbrace{\left[\frac{-i\omega p\mu_0}{4\pi}\left\{\frac{1}{\rho}\frac{\partial V_{12}}{\partial\rho} + U_{12}\right\}\right]}_{\text{EVALUB2, EVALUB3}}.$$

$$E_{2z}^{H} = \underbrace{\cos\phi}_{\text{SFIELDS}} \underbrace{\left[\frac{i\omega p\mu_0}{4\pi}\left\{\frac{\partial^2 V_{12}}{\partial h\partial\rho}\right\}\right]}_{\text{EVALUB2, EVALUB3}}.$$

HORIZONTAL ELECTRIC DIPOLE IN MEDIUM 2 (AIR)

Source Above, Observer Below Interface

$$E_{2\rho}^{H} = \underbrace{\cos\phi}_{\text{GFIELDS}} \underbrace{\left[\frac{i\omega p\mu_0}{4\pi k_2^2}\left\{\left(\frac{\partial^2}{\partial\rho^2} + k_2^2\right)(G_{22} - G_{21})\right\}\right]}_{\text{GEVALA}} + \underbrace{\cos\phi}_{\text{SFIELDS}} \underbrace{\left[\frac{-i\omega p\mu_0}{4\pi k_2^2}\left\{\frac{\partial^2}{\partial\rho^2}(k_2^2 V_{22}) + k_2^2 U_{22}\right\}\right]}_{\text{EVALUB2, EVALUB3}}$$

$$E_{2\phi}^{H} = \underbrace{\sin\phi}_{\text{GFIELDS}} \underbrace{\left[\frac{-i\omega p\mu_0}{4\pi k_2^2}\left\{\left(\frac{1}{\rho}\frac{\partial}{\partial\rho} + k_2^2\right)(G_{22} - G_{21})\right\}\right]}_{\text{GEVALA}} + \underbrace{\cos\phi}_{\text{SFIELDS}} \underbrace{\left[\frac{-i\omega p\mu_0}{4\pi k_2^2}\left\{\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(k_2^2 V_{22}\right) + k_2^2 U_{22}\right\}\right]}_{\text{EVALUB2, EVALUB3}},$$

$$E_{2z}^{H} = \underbrace{\cos\phi}_{\text{GFIELDS}} \underbrace{\left[\frac{i\omega p\mu_0}{4\pi k_2^2}\left\{\frac{\partial^2}{\partial z\,\partial p}(G_{22} + G_{21})\right\}\right]}_{\text{GEVALA}} + \underbrace{\cos\phi}_{\text{SFIELDS}} \underbrace{\left[\frac{i\omega p\mu_0}{4\pi k_2^2}\left\{-k_1^2 V_{22}\right\}\right]}_{\text{EVALUA2, EVALUA3}}.$$

$$E^H_{1\rho} = \underbrace{\cos}_{\text{SFIELDS}} \underbrace{\left[\frac{i\omega p\mu_0}{4\pi}\left\{\frac{\partial^2 V_{21}}{\partial\rho^2} + U_{21}\right\}\right]}_{\text{EVALUA2, EVALUA3}} .$$

$$E^H_{1\phi} = \underbrace{\sin\phi}_{\text{SFIELDS}} \underbrace{\left[\frac{-i\omega p\mu_0}{4\pi}\left\{\frac{1}{\rho}\frac{\partial V_{21}}{\partial\rho} + U_{21}\right\}\right]}_{\text{EVALUA2, EVALUA3}} .$$

$$E^H_{1z} = \underbrace{\cos\phi}_{\text{SFIELDS}} \underbrace{\left[\frac{-i\omega p\mu_0}{4\pi}\left\{\frac{\partial^2 V_{21}}{\partial h\partial\rho}\right\}\right]}_{\text{EVALUA2, EVALUA3}} .$$

GREEN'S FUNCTION DEFINITIONS

$$G_{11} = \frac{e^{ik_1R}}{R_1} .$$

$$G_{12} = \frac{e^{ik_1R_2}}{R_2} .$$

$$G_{21} = \frac{e^{ik_2R_1}}{R_1} .$$

$$G_{22} = \frac{e^{ik_2R_2}}{R_2} .$$

$$R_1 = \sqrt{\rho^2 + (h + z)^2} .$$

$$R_2 = \sqrt{\rho^2 + (h - z)^2} .$$

$$U_{ij} = U(a,b,\rho).$$

$$V_{ij} = V(a,b,\rho).$$

$$U_{11} = \underbrace{U(h - z,0,\rho)}_{\text{SBOB, HSBOB}} .$$

$$U_{12} = \underbrace{U(h,z,\rho)}_{\text{SBOA, HSBOA}} .$$

$$U_{22} = \underbrace{U(0,h + z,\rho)}_{\text{SAOA, HSAOA}} .$$

$$U_{21} = \underbrace{U(-z,h,\rho)}_{\text{SAOB, HSAOB}} .$$

$$V_{11} = \underbrace{V(h - z,0,\rho)}_{\text{SBOB, HSBOB}} .$$

$$V_{12} = \underbrace{V(h,z,\rho)}_{\text{SBOA, HSBOA}} .$$

$$V_{22} = \underbrace{V(0,h + z,\rho)}_{\text{SAOA, HSAOA}} .$$

$$V_{21} = \underbrace{V(-z,h,\rho)}_{\text{SAOB, HSAOB}} .$$

$$U(a,b,\rho) = \int_{\infty}^{\infty} \frac{e^{-\gamma_1 a-\gamma_2 b}}{\gamma_1 + \gamma_2} H_0^1(\lambda\rho)\lambda d\lambda .$$

$$= \int_0^{\infty} \frac{2e^{-\gamma_0 a-\gamma_2 b} J_0(\lambda\rho)\lambda d\lambda}{\gamma_1 + \gamma_2} .$$

$$V(a,b,\rho) = \int_{-\infty}^{\infty} \frac{e^{-\gamma_1 a-\gamma_2 b}}{k_1^2\gamma_2 + k_2^2\gamma_1} H_0^1(\lambda\rho)\lambda d\lambda$$

$$= \int_0^{\infty} \frac{2e^{-\gamma_1 a-\gamma_2 b} J_0(\lambda\rho)\lambda d\lambda}{k_1^2\gamma_2 + k_2^2\gamma_1} .$$

Appendix B — Nortons Formulas

$$E_Z^V \cong -\frac{j\omega\mu_0 (I\ell)_v}{4\pi} \cdot \Bigg[\underbrace{\sin^2\theta_D \frac{\exp(-j\beta_0 R_D)}{R_D}}_{\text{Direct}} + \underbrace{R^H(\theta_R) \sin^2\theta_R \frac{\exp(-j\beta_0 R_R)}{R_R}}_{\text{Reflected}}$$

$$+ \underbrace{\left(1 - R^H(\theta_R)\right) F_e \sin^2\theta_R \frac{\exp(-j\beta_0 R_R)}{R_R}}_{\text{First-order surface wave}}$$

$$+ \underbrace{\frac{\exp\left(-j\beta_0 R_D\right)}{R_D} \left(\frac{1}{j\beta_0 R_D} + \frac{1}{(j\beta_0 R_D)^2}\right) \left(1 - 3\cos^2\theta_D\right)}_{\text{Near field of direct}}$$

$$+ \underbrace{R_N^H \frac{\exp\left(-j\beta_0 R_R\right)}{R_R} \left(\frac{1}{j\beta_0 R_R} + \frac{1}{(j\beta_0 R_R)^2}\right)\left(1 - 3\cos^2\theta_R\right)}_{\text{Near field of reflected}}$$

$$+ \underbrace{2 \frac{\sqrt{N^2 - \sin^2\theta_R}}{N^2} \cos\theta_R \frac{\exp(-j\beta_0 R_R)}{j\beta_0 R_R^2}}_{\text{Near-field correction}} \Bigg]$$

$$E_\rho^V \cong \frac{j\omega\mu_0 (I\ell)_v}{4\pi} \cdot \Bigg[\underbrace{\cos\theta_D \sin\theta_D \frac{\exp(-j\beta_0 R_D)}{R_D}}_{\text{Direct}}$$

$$+ \underbrace{R^H(\theta_R) \cos\theta_R \sin\theta_R \frac{\exp(-j\beta_0 R_R)}{R_R}}_{\text{Reflected}}$$

$$- \underbrace{\left(1 - R^H(\theta_R)\right) F_e \sin\theta_R \frac{\sqrt{N^2 - \sin^2\theta_R}}{N^2} \frac{\exp(-j\beta_0 R_R)}{R_R}}_{\text{First-order surface wave}}$$

$$+ \underbrace{3 \cos\theta_D \sin\theta_D \left(\frac{1}{j\beta_0 R_D} + \frac{1}{(j\beta_0 R_D)^2} \right) \frac{\exp(-j\beta_0 R_D)}{R_D}}_{\text{Near field of direct}}$$

$$+ \underbrace{R_N^H \, 3 \cos\theta_R \sin\theta_R \left(\frac{1}{j\beta_0 R_R} + \frac{1}{(j\beta_0 R_R)^2} \right) \frac{\exp(-j\beta_0 R_R)}{R_R}}_{\text{Near field of reflected}}$$

$$- \underbrace{\left(1 - R^H(\theta_R)\right) \cos\theta_R \sin\theta_R \frac{\exp(-j\beta_0 R_R)}{j\beta_0 R_2^2}}_{\text{Near-field correction.}}$$

$$- \underbrace{\left(1 - R^H(\theta_R)\right) \sin\theta_R \frac{\sqrt{N^2 - \sin^2\theta_R}}{N^2} \frac{\exp(-j\beta_0 R_R)}{j2\beta_0 R_R^2}}_{\text{Near-field correction.}} \Bigg]$$

$$E_Z^H = \frac{j\omega\mu_0 (I\ell)_H}{4\pi} \cos(\phi - \beta) \Bigg[\underbrace{\cos\theta_D \sin\theta_D \frac{\exp(-j\beta_0 R_D)}{R_D}}_{\text{Direct}}$$

$$- \underbrace{R^H(\theta_R) \cos\theta_R \sin\theta_R \frac{\exp(-j\beta_0 R_R)}{R_R}}_{\text{Reflected}}$$

$$+ \underbrace{\left(1 - R^H(\theta_R)\right) F_e \sin\theta_R \frac{\sqrt{N^2 - \sin^2\theta_R}}{N^2} \frac{\exp(-j\beta_0 R_R)}{R_R}}_{\text{First-order surface wave}}$$

$$+ \underbrace{3 \cos\theta_D \sin\theta_D \left(\frac{1}{j\beta_0 R_D} + \frac{1}{(j\beta_0 R_D)^2} \right) \frac{\exp(-j\beta_0 R_D)}{R_D}}_{\text{Direct induction and static}}$$

$$- \underbrace{R_N^H \, 3 \cos\theta_R \sin\theta_R \left(\frac{1}{j\beta_0 R_R} + \frac{1}{(j\beta_0 R_R)^2} \right) \frac{\exp(-j\beta_0 R_R)}{R_R}}_{\text{Reflected induction and static}}$$

$$+ \underbrace{\left(1 - R^H(\theta_R)\right) \cos\theta_R \sin\theta_R \frac{\exp(-j\beta_0 R_R)}{j\beta_0 R_R^2}}_{\text{Higher-order induction}}$$

$$+ \underbrace{\left(1 - R^H(\theta_R)\right) \sin\theta_R \frac{\sqrt{N^2 - \sin^2\theta_R}}{N^2} \frac{\exp(-j\beta_0 R_R)}{j2\beta_0 R_R^2}}_{\text{Higher-order induction}} \Bigg] .$$

$$E_\phi^H \cong \frac{j\omega\mu_0 (I\ell)_H}{4\pi} \sin(\phi - \beta) \Bigg[\underbrace{\frac{\exp(-j\beta_0 R_D)}{R_D}}_{\text{Direct}} + \underbrace{R^E(\theta_R) \frac{\exp(-j\beta_0 R_R)}{R_R}}_{\text{Reflected}}$$

$$+ \underbrace{\left(1 - R^E(\theta_R)\right) F_m \frac{\exp(-j\beta_0 R_R)}{R_R}}_{\text{First-order surface wave}}$$

$$+ \underbrace{\left(1 + \frac{1}{j\beta_0 R_D}\right) \frac{\exp(-j\beta_0 R_D)}{j\beta_0 R_D^2}}_{\text{Near field of direct}}$$

$$+ \underbrace{R_N^E \left(1 + \frac{1}{j\beta_0 R_R}\right) \frac{\exp(-j\beta_0 R_R)}{j\beta_0 R_R^2}}_{\text{Near field of reflected}}$$

$$- \underbrace{\left(1 - R^H(\theta_R)\right) \frac{F_e}{N^2} \left(-1 - \frac{\cos^2\theta_R}{2} + \frac{N^2 - \sin^2\theta_R}{2N^4} - \frac{1}{2j\beta_0 R_R}\right) \frac{\exp(-j\beta_0 R_R)}{j\beta_0 R_R^2}}_{\text{Higher-order surface wave}}$$

$$+ \underbrace{\frac{1}{N^2} \left(\left(1 + R^H(\theta_R)\right) + \frac{3}{2j\beta_0 R_R} + \frac{R^H(\theta_R)}{2j\beta_0 R_R}\right) \frac{\exp(-j\beta_0 R_R)}{j\beta_0 R_R^2}}_{\text{Near-field correction}} .$$

$$E_\rho^H \cong -\frac{j\omega\mu_0 (I\ell)_H}{4\pi} \cos(\phi - \beta) \Bigg[\underbrace{\cos^2\theta_D \frac{\exp(-j\beta_0 R_D)}{R_D}}_{\text{Direct}}$$

$$\underbrace{- R^H(\theta_R) \cos^2\theta_R \frac{\exp(-j\beta_0 R_R)}{R_R}}_{\text{Reflected}}$$

$$\underbrace{- \left(1 - R^H(\theta_R)\right) F_e \left(\frac{N^2 - \sin^2\theta_R}{N^4}\right) \frac{\exp(-j\beta_0 R_R)}{R_R}}_{\text{First-order surface wave}}$$

$$\underbrace{+ \left(\frac{1}{j\beta_0 R_D} + \frac{1}{(j\beta_0 R_D)^2}\right)\left(1 - 3\sin^2\theta_D\right) \frac{\exp(-j\beta_0 R_D)}{R_D}}_{\text{Direct induction and static}}$$

$$\underbrace{- R_N^H \left(\frac{1}{j\beta_0 R_R} + \frac{1}{(j\beta_0 R_R)^2}\right)\left(1 - 3\sin^2\theta_R\right) \frac{\exp(-j\beta_0 R_R)}{R_R}}_{\text{Reflected induction and static}}$$

$$\text{Higher-order terms} \begin{cases} + \dfrac{1}{N^2} \left(\dfrac{1}{j\beta_0 R_R} + \dfrac{1}{(j\beta_0 R_R)^2}\right) \left(\left(1 + R^H(\theta_R)\right) + \left(1 + R^H(\theta_R)\, F_e\right)\left(1 - \sin^2\theta_R\right)\right) \dfrac{\exp(-j\beta_0 R_R)}{R_R} \\ + \dfrac{1}{N^2} \sin^2\theta_R \left(1 - R^H(\theta_R)\right)\left(1 + \dfrac{1}{j\beta_0 R_R}\right) \Bigg\{ F_e \left(\dfrac{N^2 - \sin^2\theta_R}{N^4} - \cos^2\theta_R \right. \\ \left. + \dfrac{1}{j\beta_0 R_R}\right) - \dfrac{1}{j\beta_0 R_R} \Bigg\} \dfrac{\exp(-j\beta_0 R_R)}{R_R} \Bigg]. \end{cases}$$

Where the defining formulae for the symbols above are:

$$N^2 = \epsilon_r - j\frac{\sigma}{\omega\epsilon_0}, \quad \beta_0 = \omega\sqrt{\mu_0\epsilon_0} = \frac{2\pi}{\lambda_o}$$

$$R^H(\theta) = \frac{N^2 \cos\theta - \sqrt{N^2 - \sin^2\theta}}{N^2 \cos\theta + \sqrt{N^2 - \sin^2\theta}}$$

$$R^E(\theta) = \frac{\cos\theta - \sqrt{N^2 - \sin^2\theta}}{\cos\theta + \sqrt{N^2 - \sin^2\theta}}$$

$$F_e = F(P_e), \; F_m = F(P_m)$$

$$F(P) = 1 - j\sqrt{\pi P}\exp(-P)\,\mathrm{erfc}\,(j\sqrt{P})$$

NOTE: $F(P) \cong -\frac{1}{2P}$ for $P >> 1$

$$P_e = \frac{-j\beta_0 R_R}{2\sin^2\theta_R} \cdot \left[\cos\theta_R + \frac{\sqrt{N^2 - \sin^2\theta_R}}{N^2}\right]^2$$

$$P_m = \frac{-j\beta_0 R_R}{2\sin^2\theta_R} \cdot \left[\cos\theta_R + \sqrt{N^2 - \sin^2\theta_2}\right]^2$$

The ground is assumed to have a conductivity σ and a relative dielectric constant ε_r everywhere, i.e., it is a homogeneous ground.

Appendix C — WF-LLL2A Code

WF-LLL2A (Wire configuration, Frequency domain, developed at Lawrence Livermore Laboratory, code number 2, version A).

PURPOSE

WF-LLL2A solves the problem of electromagnetic radiation from wire structures in free space or in the presence of a lossy half-space, including structures penetrating the interface. The solution is effected in the frequency domain via the application of the thin-wire electric-field integral equation. Antenna structures may be composed of many interconnected wires of differing radii, which may also be impedance loaded. Electric space and surface wave fields may be evaluated.

LANGUAGE

LLLTRAN, single precision, is the FORTRAN language implemented by LLL's Computations Department. It is essentially FORTRAN IV with a few added features, such as alphanumeric statement labels and drop-through IF statements.

AUTHORS

The code was originally developed at MB Associates, San Ramon, California. After its arrival at LLL, F. J. Deadrick and E. K. Miller made extensive improvements, including development of the capability for allowing complex wave numbers in both the upper and lower half-space. D. L. Lager and R. J. Lytle added the Sommerfeld and Norton ground treatments.

ACKNOWLEDGMENT

The authors gratefully acknowledge ARPA's funding of the effort to develop the Sommerfeld/Norton portion of the code.

AVAILABILITY

A source deck containing approximately 4000 cards, plus a listing, is available from LLL.

The theory for the Sommerfeld and Norton ground treatments is contained in the LLL report, "Numerical Evaluation of Sommerfeld Integrals," by R. J. Lytle and D. L. Lager.[2]

A user's manual for the program is currently being published. Until it is available, the manual for the WAMP code (WF-MBA/LLL1) may be used to obtain the theoretical discussion of the thin-wire electric-field integral equation used and the method-of-moments solution used. Moreover, most of the routines in WF-LLL2A differ only slightly from those in WAMP.

Debugging and modeling advice is available from the LLL contacts listed below.

DESCRIPTION

Program WF-LLL2A uses a moment method to obtain the numerical solution. A subsectional collocation method using point matching and a three-term (constant, sine, and cosine) current expansion function is used for the thin-wire electric-field integral equation:

$$\bar{E}^I(\bar{r})\cdot\hat{t}(\bar{r}) = \frac{i\omega\mu_0}{4\pi}\int_{C(\bar{r})} i(\bar{r})\left\{\hat{t}(\bar{r})\cdot\hat{t}(\bar{r}') + \frac{1}{k^2}[t(\bar{r})\cdot\nabla)][t(\bar{r}')\cdot\nabla)]\right\} g\, ds',$$

where

$$g = e^{-ikR}/R,$$

$$k = \omega\sqrt{\mu_0\varepsilon_0},$$

and

$$R = |\bar{r} - \bar{r}' + \bar{a}(\bar{r}')| ,$$

with $\hat{t}(\bar{r})$ the tangent vector to the wire at observation point $\bar{r}$, $\bar{E}^I$ the incident field, $\bar{a}(\bar{r}')$ the wire radius at $\bar{r}'$ in the direction $\hat{t}(\bar{r}') \times (\bar{r} - \bar{r}')$, $i(r')$ the wire current at r' (assumed uniform around the wires), and the (suppressed) time variation $e^{i\omega t}$.

The above equation applies only to wire structures located in free space. Location of a structure near the interface between two electrically dissimilar media, however, leads to reflected fields which can modify the free-space current distributions.

The present implementation of the code uses four methods for the ground treatment. Two represent a geometric optics approach, using either the normal incidence or specular plane-wave reflection coefficients to account for the reflected fields. The perfect image fields are modified according to the electrical parameters of the lower half-space. A third, more rigorous, approach is the use of Norton's formulas when the distances and ground parameters involved are within the proper range. The fourth and most rigorous approach involves the use of two representations for Sommerfeld integrals. The program automatically chooses either the Hankel function form or the Bessel function form, whichever is the most efficient. The integrals are evaluated by performing numerical complex contour integration using an adaptive Romberg integration scheme. Shank's algorithm is combined with Romberg integration to obtain optimum convergence for the integrals along a semi-infinite contour.

The numerical solution of a wire structure involves four basic operations. The first defines the contour $C(\bar{r})$ over which to evaluate the thin-wire integral equation. This is done by decomposing the structure into many interconnected straight-wire segments of a finite radius. (A loop, for example, would be modeled as a n-sided polygon.) The program allows for multiple junctions of wires and the loading of segments with an arbitrary impedance. Several input formats are provided to make the task of specifying the locations and interconnections between the segments more user-oriented.

Once the geometry of the structure has been defined in terms usable by the program, the structure's impedance matrix is computed by calculating the tangential electric field at each segment observation point, i, due to a unit current flowing on source segment j. For the geometric optics ground treatment the tangential fields are found by summing two terms, the direct or free-space contribution, and the ground-reflection contribution, found by computing the perfect image fields modified by the reflection coefficient at the specular point. For the Sommerfeld ground treatment the tangential fields are also found by summing two terms, the direct contribution found by integrating the free-space Green's functions, and a "ground-correction" contribution found by evaluating the Sommerfeld integrals. For the Norton ground treatment the fields are found by summing many terms including the direct term, the reflected term, the first-order surface-wave term, the direct and reflected near-field terms, and a near-field correction.

Once the impedance matrix has been evaluated, it is then factored into an upper and lower triangular matrix, and solved via a Gauss-Jordan elimination algorithm. The source vector used in the solution of the system of equations represents the tangential electric field at each segment. Thus for an antenna problem, a single segment may be driven, whereas for scattering problems, each segment will have a source field dependent on the incident electric fields. The matrix factorization and solution yields the current distribution on the structure. From the solved currents, one may then compute an input impedance for an antenna and the far-field radiation pattern (including the surface-wave fields). The program takes advantage of any Toeplitz symmetry specified for a structure to reduce significantly the fill time for the impedance matrix.

LIMITATIONS

At least six current samples/wavelength should be used in setting up a numerical model, and the segment-length-to-wire-radius ratio should be greater than 5 for the thin-wire approximations to be valid.

The two geometric optics ground treatment and Norton's formulas are only valid for structures located in the upper half-space. The coding for the Sommerfeld integrals is valid for structures located above, below, or penetrating the interface between the two half-spaces. At the present time the coding has been thoroughly checked out only for cases where the structure is above the interface. For example, good agreement has been achieved with other calculations and/or experimental measurements for long horizontal wires near the interface (e.g., Beverage antennas), and vertical and horizontal half-wave dipoles.

STORAGE

To improve interaction in a time-sharing environment, the code uses a dynamic storage-allocation scheme where the size of the largest array, called the CM matrix, is determined after reading the data cards. The program then tells the operating system exactly how much memory is necessary to solve the given problem. For the current implementation on the CDC 7600, the code varies in size from about 61,000 words to 241,000 words as the number of segments is varied from 1 to 300. Of this, about 16,000 words are for the Sommerfeld/Norton routines.

The storage necessary for the CM matrix is $2N^2$, where N is the number of segments in the structure. The factor of 2 is due to the array being type COMPLEX. There are also about 25 arrays of fixed size which are dimensioned at the maximum number of segments allowed (currently 300). It is possible to reduce the memory requirements by making these arrays smaller. There are about 5000 words used for I/O buffers; these could also easily be reduced in size. For a particular problem it is also possible to reduce the memory requirement by eliminating routines which would not be called. For example, the subroutine SURF, which is only used for plotting surface-wave radiation pattern patterns, could easily be eliminated when those plots are not desired. Another example would be the elimination of the Sommerfeld routines EVALUB2 and EVALUB3 when performing calculations for above-surface structures, since these routines are called only for below-surface structures.

TIMING

The computer time required depends strongly on the number of segments in the structure, the ground treatment used, and the symmetry inherent in the structure. If a structure contains an element horizontal to the ground (a Beverage antenna, for example), it is possible to make use of the Toeplitz symmetry in that element to reduce the matrix fill time. For a Beverage antenna or a horizontal dipole, which are highly Toeplitz symmetric, the fill time is proportional to N, rather than the N^2 for other structures. The ground treatment affects the matrix fill time since the calculation of the Sommerfeld integrals is much slower than either Norton's formulas or the geometric optics treatments. The code automatically uses Norton's formulas when the separation between the source and observer is greater than one wavelength, making the computer time for a large structure (say a 10-wavelength Beverage) only slightly longer than the time for a one-wavelength structure.

An estimate of the CDC 7600 time necessary to find the current distribution on a vertical dipole is given by:

$$T = \alpha N^2 + \beta N^3 \text{ s},$$

where

N is the number of segments,

β is the coefficient for factoring and solving the system of equations $\beta \cong 2.6 \times 10^{-6}$,

and

α is the coefficient for filling the impedance matrix; $\alpha = 0.0022$ for the geometric optics ground treatment, or $\alpha \simeq 0.34$ for the Sommerfeld ground treatment.

Since there is no Toeplitz symmetry for a vertical antenna, the matrix fill time varies as N^2. The time necessary for a $\lambda/2$ vertical dipole with N = 5 is about 8 s, using the Sommerfeld ground treatment.

An estimate of the time necessary for a horizontal dipole less than one wavelength long is given by:

$$T = \alpha N + \beta N^3 \text{ s},$$

where

$\alpha = 0.0016$ for the geometric optics ground treatment,

or $\alpha = 7.6$ for the Sommerfeld ground treatment.

Since a horizontal dipole has full Toeplitz symmetry, the matrix fill times varies as N. The time necessary for a $\lambda/2$ horizontal dipole with N = 5 is about 4 s.

An estimate of the time necessary for a horizontal dipole longer than one wavelength, using the Sommerfeld/Norton ground treatment, is:

$$T = \alpha_S M + \alpha_N (N - M) + \beta N^3 \text{ s},$$

where

M is the number of segments in one wavelength,

α_S is the matrix fill coefficient for the Sommerfeld ground trea (used when the separation between source and observer is less 1 wavelength) $\cong$ 0.76, and

α_N is the matrix fill coefficient for the Norton ground treatment when the separation is greater than 1 wavelength) $\cong$ 0.002.

The time necessary for 10-wavelength horizontal dipole with 101 segments is abou

REPRESENTATIVE GEOMETRY

A representative structure, Fig. 11, is a horizontal Beverage antenna 10 lengths long, only 1/15 wavelength above a lossy ground, and terminated with a 30 load

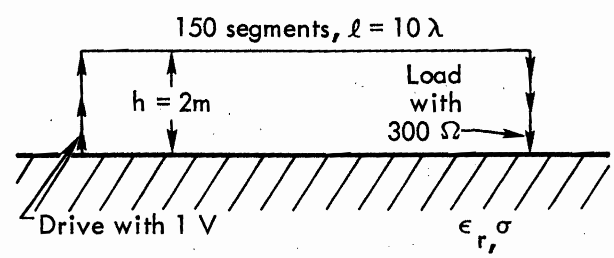

Fig. 11. Representative structure analyzed by WF-LLL2A code.

LLL CONTACTS

D. L. Lager, F. J. Deadrick, R. J. Lytle, and E. K. Miller.

Appendix D – Listing of WF-LLL2A Code

```
      PROGRAM WFLLL2A(TAPE2,HSP,TAPE6,TAPE8,TAPE9)
C
C
C
C PROGRAM WFLLL2A
C   W--THIN WIRE STRUCTURES
C   F--FREQUENCY DOMAIN
C   LLL--LAWRENCE LIVERMORE LAB WAS MAJOR CONTRIBUTOR
C   2--CODE NO. 2
C   A--VERSION A
C
C
C
C
C A BRIEF DESCRIPTION OF THE CODE AND ITS CAPABILITIES IS GIVEN IN
C   THE COMPUTER CODE NEWSLETTER, VOL. 2 NO. 1, APRIL 1, 1975.
C
C
C
      CODE ANALYSIS
      CALL NEWNAME
      CALL INLINE(              )
      COMMON/NEWCOM/NNEW,NOLD
      LCM (333)
      PARAMETER (NS = 300)
       COMMON/GEOM/N,NP,X(NS),Y(NS),Z(NS),SI(NS),BI(NS),ALP(NS),
     1  BET(NS),ICON1(NS),ICON2(NS),COLAM
       INTEGER P
       COMPLEX  ADMIT,ZPED,RRV,RRH,ZRSIN,RRD,ERX,ERY,ERZ,EPX,EPY,
     1 EPSILONP,EPSILONM,ZRATIP,ZRATIM,WKP,WKM,ECONST
       COMPLEX CM,FJ,EINC,EXA,CIX,CIY,CIZ,ERC,CSQRT
      COMMON /FROSTCOM/-INTEGER-ALPHA,BETA(15)
      COMMON /GOBCOM/ IFIL(42)
      COMMON /333/ CM(1)
      COMMON /FREQ/ FREQ,WKP,WKM,ECONST
      COMMON /ANGL/ CAB(NS),SAB(NS),SALP(NS)
       COMMON/JUNK/NCOX,JOX(25),NCIX,JIX(25),NCOZ,JOZ(25),NCIZ,JIZ(25)
      COMMON /MEDIA/ EPSRP,SIGP,EPSRM,SIGM
      DIMENSION EINC(NS),P(NS),IBET(2)
      COMMON/COMCOM/COM(8)
      DIMENSION CURR(NS),CURI(NS)
       DIMENSION THETR(20),PHYR(20),ETAR(20),DTHR(20),DPHR(20),NTHR(20),
     1 NPHR(20)
      COMMON /ABC/ AIR(NS),AII(NS),BIR(NS),BII(NS),CIR(NS),CII(NS)
       DIMENSION ISEG(2,151),ENCR(2,150),ENCI(2,150)
      DIMENSION CME(1)
      EQUIVALENCE (CM,CME)
 3     FORMAT(4I5)
 3333  FORMAT(37H   X(I)       Y(I)       Z(I)       S(I))

      COMMON/JOBCOM/JOBNO $$$ USED IN CMSETUP (FOR INTERRUPT)
      JOBNO=0

      MTIME=0
      NDIM = NS
       NRPAGE=45
      FJ=CMPLX(0.,1.)
       ZZ=376.72727
       PI=3.141592654
       TP=2.*PI
       TA=.01745329252
       TD=57.29577951
       CONST=ZZ/(2.*TP)
      C = 2.99793E+8
```

```
      IOFILE=2RTA
C
C        FIND SIZE OF LARGE CORE FIELD LENGTH.
C
      IFLL=IFIL(16)
C
 1000 CONTINUE
      JOBNO=JOBNO+1
      READ(2,458) (COM(I),I=1,8)
      IF(EOF,2)1600,1700
 458   FORMAT(8A10)
 1700 READ(2,460) NTYPE,NPRINT,NRUN,ILOAD,ISEL,MODE,IDISK,IPGND,IGSCRN
     *  IBO,ISEGEX,NORMPWR,ISURF,IVERTRC
 460  FORMAT(16I5)
C      IF(IBO .EQ. 1) CALL ASSIGN(3,15,2RBO)
C      IF(IBO .EQ. 1) CALL KEEPBO(1)
      READ(2,9010)GHZ,GR,NFS,KSYMP,EPSRP,SIGP,EPSRM,SIGM
C      WRITE(3,9010) GHZ,GR,NFS,KSYMP,EPSRP,SIGP,EPSRM,SIGM
      CALL SETCH(0,100.,1,0,0,0,0)
C      WRITE(100,9010) GHZ,GR,NFS,KSYMP,EPSRP,SIGP,EPSRM,SIGM
 9010  FORMAT(2F10.5,2I5,4F10.5)
C
C  COLAM IS MADE NEGATIVE TO INDICATE TO OTHER PROGRAMS THAT DIMENSIONS
C  ARE GIVEN IN METERS FOR THE WFLLL2A PROGRAM
C
      COLAM = -0.299793 / GHZ
      FREQ = GHZ*1.E9
 5553 WRITE(3,8332)
      WRITE(100,8332)
 8332  FORMAT(///,29H1****************************)
      WRITE(3,8333) JOBNO
      WRITE(100,8333) JOBNO
 8333 FORMAT(/,25H PROGRAM WFLLL2A RUN NUMBERI4,/)
      WRITE(3,8334)
      WRITE(100,8334)

 8334  FORMAT(29H ****************************)
      WRITE(3,457) (COM(I),I=1,8)
      WRITE(100,457) (COM(I),I=1,8)
 457   FORMAT(///,1X,8A10/)
      WRITE(3,460)NTYPE,NPRINT,NRUN,ILOAD,ISEL,MODE,IDISK,IPGND,IGSCRN
     * ,IBO,ISEGEX,NORMPWR,ISURF,IVERTRC
      WRITE(100,460)NTYPE,NPRINT,NRUN,ILOAD,ISEL,MODE,IDISK,IPGND,IGSCRN
     * ,IBO,ISEGEX,NORMPWR,ISURF,IVERTRC
 459   FORMAT(7I5,F10.5)
      WRITE(3,9011)  GHZ,GR,NFS,COLAM
      WRITE(100,9011)  GHZ,GR,NFS,COLAM
 9011 FORMAT(/1X9HFREQUENCY12X1H=E13.5/1X22HFREQUENCY INCREMENT  =E13.5/
     11X22HNO. FREQUENCY STEPS   =I4/1X22HWAVELENGTH   (METERS) =E13.5//)
      WRITE(3,9012) SIGP,EPSRP,SIGM,EPSRM
      WRITE(100,9012) SIGP,EPSRP,SIGM,EPSRM
 9012 FORMAT(  MEDIA PARAMETERS ,/,   UPPER HALFSPACE--SIGMA =   ,E12.4,
     1      EPSILON =   ,E12.4,//,    LOWER HALFSPACE--SIGMA =   ,E12.4,
     2      EPSILON =   ,E12.4,//,   THE INTERFACE IS LOCATED AT Z = 0 )
 FWOT1 FORMAT(/// **A PERFECT GROUND WAS USED** ///)
       IF(IPGND.EQ.1) WOT 3,FWOT1
       IF(IPGND.EQ.1) WOT 100,FWOT1
 FWOT2 FORMAT(/// **USED VERTICAL INCIDENCE REF. COEF. ONLY** ///)
       IF(IVERTRC) WOT 3,FWOT2
       IF(IVERTRC) WOT 100,FWOT2
       GO TO (21,22,26,24) NTYPE
 21     CALL DATAGN1
        GO TO 20
 22     CALL DATAGN2
```

```
        GO TO 20
   26   CALL DATAGN6
        GO TO 20
   24   CALL DATATIP
        GO TO 20
   20   WRITE(3,9006) N,NP
        WRITE(100,9006) N,NP
C
C          NOW THAT THE SIZE OF  CM  ARRAY IS KNOWN, ISSUE A FROSTCALL TO
C          CREATE A NEW DISKFILE OF THE PROPER LENGTH.
C
        NFLL=IFLL+N*N*2
        BETA=NNEW
        BETA(2)=NFLL+230B
        CALL FROST(0101B,1,BETA,ERR1)
C
C          ISSUE A FROSTCALL TO ADJUST THE LARGE-CORE MEMORY SIZE.
C
        BETA=NFLL
        CALL FROST(1300B,0,BETA,ERR2)
        NDIM=N
        CALL ANTPLOT
        CALL PLOTEA
        CALL SETCH(0,100.,1,0,0,0,0)
 9006 FORMAT(//1X22HNUMBER OF SEGMENTS   =I4/1X22HNO. SEG. IN A SECTOR =
     1I4/)
 3456 FORMAT(50H1STRUCTURE GEOMETRY   (DIMENSIONS IN METERS)      //2X
     163HCOORDINATES OF SEG.CENTER      SEG.      WIRE    ORIENTATION AN
     222HGLES    CONNECTION DATA/6X34HX          Y          Z       LENGTH
     347H    RADIUS    ALPHA     BETA        I-    I     I+)
         IP=NRPAGE
         SLEN=0.
         DO  40  I=1,N
         IF(NPRINT+1)801,800,800
  800    AP=ALP(I)*TD
         BT=BET(I)*TD
         IP=IP+1
         IF(IP.LE.NRPAGE)GO TO 66
        WRITE(3,3456)
        WRITE(100,3456)
         IP=1
   66 WRITE(3,6)X(I),Y(I),Z(I),SI(I),BI(I),AP,BT,ICON1(I),I,ICON2(I)
        WRITE(100,6)X(I),Y(I),Z(I),SI(I),BI(I),AP,BT,ICON1(I),I,ICON2(I)
  6      FORMAT(1X,7F10.5,3I5)
  801    ALPI=ALP(I)
         BETI=BET(I)
         CALP=COSF(ALPI)
         SALP(I)=SINF(ALPI)
         CAB(I)=CALP*COSF(BETI)
         SAB(I)=CALP*SINF(BETI)
         SLEN=SLEN+SI(I)
         IF(SI(I).GT.0) GO TO 40
        WRITE(3,9001) I
        WRITE(100,9001) I
 9001 FORMAT(30H NEGATIVE SEGMENT LENGTH.    I=I5)
        CALL EXIT
   40    CONTINUE
        WRITE(3,9008) SLEN
        WRITE(100,9008) SLEN
 9008 FORMAT(/23H TOTAL WIRE LENGTH     =E18.11)
         IF(MODE.EQ.2)GO TO 70001
         DO 730 I=1,2
         ISEG(I,151)=0
         DO 730 K=1,150
```

```
      ISEG(I,K)=0
      ENCR(I,K)=0.0
730   ENCI(I,K)=0.0
      IM=0
      WRITE(3,76047)
      WRITE(100,76047)
76047 FORMAT(/28HANTENNA SOURCE DISTRIBUTIONS/23H CASE SEG.          VOLTA
     12HGE/29H NO.  NO.     MAG.      PHASE)
  734 READ(2,76046)I,IS,ECM,ECA,NFLD
C    INPUT  FOR  ANT. USE
C    I - CASE NUMBER  ( LESS THAN 2)
C    ECM - VOLTAGE MAG.
C   ECA - VLOTAGE PHASE
      WRITE(3,76046)I,IS,ECM,ECA,NFLD
      WRITE(100,76046)I,IS,ECM,ECA,NFLD
76046 FORMAT(2I5,2F10.5,I5)
      IF(1.LE.I.AND.I.LE.2) GO TO 731
      IERR=1    $   GO TO 13
731   IF(1.LE.IS.AND.IS.LE.N) GO TO 732
      IERR=2   $  GO TO 13
732   IF(I.GT.IM)IM=I
      K=ISEG(I,151)+1
      IF(K.LE.150) GO TO 733
      IERR=3   $  GO TO 13
733   ISEG(I,151)=K
      ISEG(I,K)=IS
      ECA=ECA*TA
      ENCR(I,K)=ECM*COSF(ECA)
      ENCI(I,K)=ECM*SINF(ECA)
      IF(NFLD.NE.0) GO TO 734
      KM=IM
70001 K=0
745   K=K+1
      KMR=K
 7905 IF(ILOAD.EQ.0) GO TO 8811
 6506 FORMAT(I5,2E10.0,3I5)
 6507 FORMAT(  CASE= ,I3,9H SEGMENTS,I4,5H THRU,I4,12H LOADED WITH,E10.3,
     120H OHMS RESISTANCE AND,E10.3,19H HENRIES INDUCTANCE)
      PARAMETER (MAXLOADS=20)
      DIMENSION LOADS(3,MAXLOADS),ZLOADS(2,MAXLOADS)
      DO DLO I=1,MAXLOADS
      RIT 2,6506,ICASENO,ZR,ZI,I1,I2,MORE
      IF(I2.EQ.0) I2=I1
      WOT 3,6507,ICASENO,I1,I2,ZR,ZI
      WOT 100,6507,ICASENO,I1,I2,ZR,ZI
      IF(I2.GE.I1) GO TO GOK1
 FER1 FORMAT( **ERROR** I1.GT.I2 FOR SEGMENT LOADING )
      WOT 59,FER1
      WOT 3,FER1
      WOT 100,FER1
      CALL EXIT
 GOK1 CONTINUE
      LOADS(1,I)=ICASENO
      LOADS(2,I)=I1
      LOADS(3,I)=I2
      ZLOADS(1,I)=ZR
      ZLOADS(2,I)=ZI
      IF(MORE.EQ.0) GO TO GOUT1
  DLO CONTINUE
 FER2 FORMAT( **ERROR**TOO MANY LOADS SPECIFIED )
      WOT 59,FER2
      WOT 3,FER2
      WOT 100,FER2
      CALL EXIT
```

```
GOUTI CONTINUE
      NLOADS=1
C *****BEGIN FREQUENCY DO LOOP.
8811    DO 773 MKS=1, NFS
        FR=(GHZ+GR)/GHZ
        IF(MKS.EQ.1)   FR = 1.
        GHZ=GHZ*FR
      FREQ = FREQ*FR
COMMENT----SET UP INITIAL PARAMTERS TO SOMMERFELD ROUTINE
       IF(KSYMP.EQ.3) CALL SETUP(FREQ,EPSRM,SIGM,EPSRP,SIGP)
COMMENT----ABOVE IS ONLY VALID FOR SORCE AND OBSERVER ABOVE GROUND
C
C  EPSRP AND SIGP ARE THE MEDIA PARAMETERS OF THE UPPER (+)  HALFSPACE
C  EPSRM AND SIGM ARE THE MEDIA PARAMETERS OF THE LOWER (-)  HALFSPACE
C
      EPSILONP = EPSRP - FJ*SIGP/(TP*FREQ*8.854E-12)
      EPSILONM = EPSRM - FJ*SIGM/(TP*FREQ*8.854E-12)
C
C  WKP AND WKM ARE THE COMPLEX WAVE NUMBERS OF THE UPPER AND LOWER MEDIA
C
      WKP = TP*FREQ/C * CSQRT(EPSILONP)
      WKM = TP*FREQ/C * CSQRT(EPSILONM)
      IF(AIMAG(WKP) .GT. 0.) WKP = -WKP
      IF(AIMAG(WKM) .GT. 0.) WKM = -WKM
C
C  COMPUTE A ZRATI FOR THE UPPER AND LOWER MEDIA
C
      ZRATIP = CSQRT(EPSILONP/EPSILONM)
      ZRATIM = 1./ZRATIP
      ZRATIPR = REAL(ZRATIP)
      ZRATIPI = AIMAG(ZRATIP)
      ZRATIMR = REAL(ZRATIM)
      ZRATIMI = AIMAG(ZRATIM)
      WKPR = REAL(WKP)
      WKPI = AIMAG(WKP)
      WKMR = REAL(WKM)
      WKMI = AIMAG(WKM)
      WRITE(3,2) WKPR,WKPI, ZRATIPR,ZRATIPI,WKMR,WKMI,ZRATIMR,ZRATIMI
      WRITE(100,2) WKPR,WKPI, ZRATIPR,ZRATIPI,WKMR,WKMI,ZRATIMR
     1  ,ZRATIMI
 2    FORMAT(  K+ =   E12.4,5X,E12.4,   ZRATI+ =  ,E12.4,5X,E12.4,/
     1          K- =   E12.4,5X,E12.4,   ZRATI- =  ,E12.4,5X,E12.4 /)
      ECONST = TP*FREQ*0.99930012O2E-7*FJ
C********************************************************************




 1009 CONTINUE
      CALL OOTIM
      CALL CMSETUP(ZRATIP,ZRATIM,KSYMP,IPGND,IVERTRC)
      CALL OOTIM(ITIME)


C********************************************************************
       IF(ILOAD.EQ.0) ICASENO=MAXLOADS+1 $$$ TO SHUT OFF DO DLOAD
        IF(ILOAD.EQ.0) GO TO 741
COMMENT----MODIFY CM MATRIX BY LOADS
       DO DLOAD ICASENO=1,MAXLOADS $$$ ALLOWS UP TO MAXLOADS CASES
       IF(ICASENO.GT.1) GO TO GL1
COMMENT----ON FIRST PASS SAVE CM MATRIX ON DISK
       CALL FFOST(24078,0,BETA,ERR1) $$$ GET SUFFIX
 ERR1  ISUFFIX=(BETA.SHL.6).INT.778
       ICMMX=(8RCMMATRIX.SHL.6).UN.ISUFFIX
       CALL ASSIGN(44,ICMMX)
```

```
        CALL DEVICE( DESTROY ,ICMMX)
        CALL DEVICE( CREATE ,ICMMX,N*N*2+10008)
        BUFFER OUT (44,1) (CM,CME(N*N*2))
 WAITO  IF(UNIT,44) WAITO,OKOUT,,,
  FOUT  FORMAT( **ERROR** BUFFERING OUT CM MATRIX )
        WOT 59,FOUT
        WOT 3,FOUT
        WOT 100,FOUT
        CALL EXIT(1)
 OKOUT  CONTINUE
        IFLAG=1
        GO TO GL2
   GL1  CONTINUE
        IF(IFLAG.EQ.1) GO TO GL2 $$$ DONT NEED TO RE-READ
COMMENT----READ CM MATRIX FROM DISK
        REWIND (44)
        BUFFER IN (44,1) (CM,CME(N*N*2))
 WAITI  IF(UNIT,44) WAITI,OKIN,,,
   FIN  FORMAT( **ERROR** BUFFERING IN CM MATRIX )
        WOT 59,FIN
        WOT 3,FIN
        WOT 100,FIN
        CALL EXIT(1)
  OKIN  CONTINUE
        IFLAG=1
   GL2  CONTINUE
        DO DL1 INUM=1,NLOADS
        IF(LOADS(1,INUM).NE.ICASENO) GO TO GL3
  FNEW  FORMAT(////  MODIFY CM MATRIX BY SEGMENT LOADS )
        IF(IFLAG) WOT 3,FNEW
        IF(IFLAG) WOT 100,FNEW
        IFLAG=0
        WOT 3,6507,ICASENO,LOADS(2,INUM),LOADS(3,INUM),ZLOADS(1,INUM),
       .  ZLOADS(2,INUM)
        WOT 100,6507,ICASENO,LOADS(2,INUM),LOADS(3,INUM),ZLOADS(1,INUM),
       .  ZLOADS(2,INUM)
        DO DL3 I=LOADS(2,INUM),LOADS(3,INUM)
        COMPLEX ZL
        ZL=CMPLX(ZLOADS(1,INUM)/SI(I),ZLOADS(2,INUM)*TP*FREQ/SI(I))
   DL3  CM(I+N*(I-1))=CM(I+N*(I-1))-ZL
   GL3  CONTINUE
   DL1  CONTINUE
        IF(IFLAG.EQ.1) GO TO GLOAD $$$ DID NOT FIND THIS CASE NO.
 741     IF(NPRINT-1) 743,743,742
 742     DO 744 I=1,NP
        DIMENSION TMP(8)
        WOT 3,F546,I
        WOT 100,F546,I
        DO D791 J=1,N,2
        JJ=4
         COMPLEX TEMPCX
        TEMPCX=CM(I+N*(J-1))
        TMP(1)=REAL(TEMPCX)
        TMP(2)=AIMAG(TEMPCX)
        TMP(3)=CABS(TEMPCX)
        TMP(4)=TD*CANG(TEMPCX)
        IF(J+1 .GT. N) GO TO G791
        JJ=8
        TEMPCX=CM(I+N*J)
        TMP(5)=REAL(TEMPCX)
        TMP(6)=AIMAG(TEMPCX)
        TMP(7)=CABS(TEMPCX)
        TMP(8)=TD*CANG(TEMPCX)
  G791  CONTINUE
```

```
      WOT 3,546,(TMP(IJK),IJK=1,JJ)
      WOT 100,546,(TMP(IJK),IJK=1,JJ)
 0791 CONTINUE
  744 CONTINUE
 F546 FORMAT(/2X, I= ,I3)
  546 FORMAT(2X,2(E11.3, R ,E11.3, I ,E11.3, M ,F9.3, P ,5X))
C *****SOLUTION OF THE MATRIX EQUATION
743    CONTINUE
C**********************************************************************



      CALL OOTIM
       CALL FACTOR(N,P,NDIM)
      CALL OOTIM(JTIME)


C**********************************************************************
      MTIME=ITIME
      TIME=ITIME
      T=TIME*1.E-6
      WRITE(3,6505) T
      WRITE(100,6505) T
 6505 FORMAT(/31H MATRIX INVERSION TIME IN SEC. F10.3/)
      KA=1
       IF(MODE.EQ.2)GO TO 747
       DO 70004 I=1,N
70004 EINC(I)=CMPLX(0.,0.)
       ISEGL=ISEG(KA,151)
       DO 70005 I=1,ISEGL
       IS=ISEG(KA,I)
70005  EINC(IS)=-CMPLX(ENCR(KA,I),ENCI(KA,I))/SI(IS)
       IF(NPRINT.LT.0)GO TO 747
      WRITE(3,6409)
      WRITE(100,6409)
 6408 FORMAT(/3X,I5,5X,E11.3,3X,E11.3)
 6409 FORMAT (44H SEGMENT EXCITATION  (VOLTS/METER          ) / 41H SEG N
     1UMBER    REAL PART    IMAGINARY PART      )
      IF(ISEGEX .NE. 1) GO TO 6410
      CALL ASSIGN(9,0,8RGUYEXCIT,0)
      BUFFER IN (9,1)(EINC,EINC(NS))
 6411 IF(UNIT,9) 6411,6410, ,
      WRITE(3,6412)
      WRITE(100,6412)
 6412 FORMAT(  ERROR ON SEG EXCIT BUFFER IN OP--RUN TERMINATED.   )
      CALL EXIT
 6410 CONTINUE
      DO 6407 IP=1,N
      X1=REAL(EINC(IP))
      X2=AIMAG(EINC(IP))
      IF(X1 .NE. 0. .OR. X2 .NE. 0.) WRITE(3,6408) IP,X1,X2
      IF(X1 .NE. 0. .OR. X2 .NE. 0.) WRITE(100,6408) IP,X1,X2
 6407 COINTINUE
747    CONTINUE
C**********************************************************************



      CALL OOTIM
       CALL SOLVE(N,P,EINC,NDIM)
      CALL OOTIM(KTIME)


C**********************************************************************
```

```
749     DO 160 I=1,N
        CURR(I)=REAL(EINC(I))
160     CURI(I)=AIMAG(EINC(I))
        IF(NPRINT.LT.0)GO TO 430
        NHALF=(N+1)/2
        IP=NRPAGE
        DO 161 I=1,N
        J=I+NHALF
        IP=IP+1
      CMAG = SQRTF(CURR(I)*CURR(I) + CURI(I)*CURI(I))
      PH = TD*ATAN2(CURI(I),CURR(I))
       IF(J.GT.N)GO TO 162
      CMAGP = SQRTF(CURR(J)*CURR(J) + CURI(J)*CURI(J))
      PHP = TD*ATAN2(CURI(J),CURR(J))
       IF(IP.LE.NRPAGE)GO TO G161
      WRITE(3,9002)
      WRITE(100,9002)
       IP=1
 G161 WRITE(3,9003)I,CURR(I),CURI(I),CMAG,PH,J,CURR(J),CURI(J),CMAGP,PHP
  161 WRITE(100,9003)I,CURR(I),CURI(I),CMAG,PH,J,CURR(J),CURI(J),CMAGP,PHP
  162 WRITE(3,9003)I,CURR(I),CURI(I),CMAG,PH
      WRITE(100,9003)I,CURR(I),CURI(I),CMAG,PH
 9002 FORMAT(5H1SEG.4X9HCURRENT-,48X4HSEG.4X9HCURRENT -/1X3HNO.5X4HREAL,
     163H         IMAGINARY   MAGNITUDE        PHASE                NO.    RE
     243HAL         IMAGINARY   MAGNITUDE        PHASE/)
 9003 FORMAT(1X,I4,E13.4,E12.4,E16.8,F9.3,9X,I4,E13.4,E12.4,E16.8,F9.3)
430     IF(MODE.EQ.2)GO TO 2011
        IF(ISEGL.GT.1)GO TO 2011
        ADMIT=CURR(IS)+FJ*CURI(IS)
        ZPED=1./ADMIT
      WOT 3, (///)
      WOT 100, (///)
FADZP FORMAT(A10, =   ,E11.3, R ,E11.3, I ,E11.3, M ,F9.3,
     .     P     AT ,E12.5,  GHZ )
      WOT 3,FADZP, ADMITTANCE ,REAL(ADMIT),AIMAG(ADMIT),
     .    CABS(ADMIT),TD*CANG(ADMIT),GHZ
      WOT 3,FADZP, IMPEDANCE ,REAL(ZPED),AIMAG(ZPED),
     .    CABS(ZPED),TD*CANG(ZPED),GHZ
      WOT 100,FADZP, ADMITTANCE ,REAL(ADMIT),AIMAG(ADMIT),
     .    CABS(ADMIT),TD*CANG(ADMIT),GHZ
      WOT 100,FADZP, IMPEDANCE ,REAL(ZPED),AIMAG(ZPED),
     .    CABS(ZPED),TD*CANG(ZPED),GHZ
2011    CONTINUE
      IF(NPRINT.GT.0)WOT 3,9004
      IF(NPRINT.GT.0)WOT 100,9004
 9004 FORMAT(4H1   16X2HAR10X2HAI11X2HBR10X2HBI11X2HCR10X2HCI)
        DO 402 I=1,N
        CALL TRIO(I,JCO1,JCO2,DIL,DIK)
        S=SI(I)
        CL=TP*DIL/(-COLAM)
        CK=TP*DIK/(-COLAM)
      SINL=SINF(CL)
      COSL=COSF(CL)
      SINK=SINF(CK)
      COSK=COSF(CK)
      SILK=SINF(CL+CK)
       CELLO=SINL+SINK-SILK
        IF(JCO1) 403,404,405
403     CRLO=0.0
        CILO=0.0
        IF(NCIX.LT.1) GO TO 4065
        DO 406 K=1,NCIX
        JIXK=JIX(K)
        CRLO=CRLO+CURR(JIXK)
```

```
 406    CILO=CILO+CURI(JIXK)
 4065   CONTINUE
        IF(NCOX.LT.1) GO TO 4105
        DO 410 K=1,NCOX
        JOXK=JOX(K)
        CRLO=CRLO-CURR(JOXK)
 410    CILO=CILO-CURI(JOXK)
 4105   CONTINUE
        GO TO 411
 404    CRLO=0.0
        CILO=0.0
        GO TO 411
 405    CRLO=CURR(JCO1)
        CILO=CURI(JCO1)
 411    CRLL=CURR(I)
        CILL=CURI(I)
        IF(JCO2) 412,413,414
 412    CRLY=0.0
        CILY=0.0
        IF(NCOZ.LT.1) GO TO 4155
        DO 415 K=1,NCOZ
        JOZK=JOZ(K)
        CRLY=CRLY+CURR(JOZK)
 415    CILY=CILY+CURI(JOZK)
 4155   CONTINUE
        IF(NCIZ.LT.1) GO TO 4165
        DO 416 K=1,NCIZ
        JIZK=JIZ(K)
        CRLY=CRLY-CURR(JIZK)
 416    CILY=CILY-CURI(JIZK)
 4165   CONTINUE
        GO TO 417
 413    CRLY=0.0
        CILY=0.0
        GO TO 417
 414    CRLY=CURR(JCO2)
        CILY=CURI(JCO2)
 417    AIR(I)=(CRLO*SINK-CRLL*SILK+CRLY*SINL)/CELLO
        AII(I)=(CILO*SINK-CILL*SILK+CILY*SINL)/CELLO
        BIR(I)=(CRLO*(COSK-1.)+CRLL*(COSL-COSK)+CRLY*(1.-COSL))/CELLO
        BII(I)=(CILO*(COSK-1.0)+CILL*(COSL-COSK)+CILY*(1.0-COSL))/CELLO
        CIR(I)=-(CRLO*SINK-CRLL*(SINL+SINK)+CRLY*SINL)/CELLO
        CII(I)=-(CILO*SINK-CILL*(SINL+SINK)+CILY*SINL)/CELLO
        IF(NPRINT) 402,402,401
   401 WRITE(3,98)I,AIR(I),AII(I),BIR(I),BII(I),CIR(I),CII(I)
       WRITE(100,98)I,AIR(I),AII(I),BIR(I),BII(I),CIR(I),CII(I)
 98     FORMAT(1X,I4,3(1X,2E12.4))
 402    CONTINUE
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C   IF NORMPWR = 1,  THEN NORMALIZE THE ANTENNA CURRENTS TO A REFERENCE
C   POWER OF 1 WATT FOR THE NEAR AND FAR FIELD CALCULATIONS
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       IF(NORMPWR .EQ. 0) GO TO 418
       PWRSUM=0.
       DO NORM I=1,ISEGL
       IEXCIT=ISEG(KA,I)
       PWR=0.5*REAL((ENCR(KA,I)+FJ*ENCI(KA,I))*
      .  (CURR(IEXCIT)-FJ*CURI(IEXCIT)))
       PWRSUM=PWRSUM+PWR
  NORM CONTINUE
       WATT1=SQRT(1./PWRSUM)
       WRITE(3,419) WATT1
```

```
      WRITE(100,419) WATTI
      WOT 3, (///)
      WOT 100, (///)
 419  FORMAT( 1 * * * * * * * * * * * * * * * * * * * * * * //     VOLTAGE TO D
     1RIVE ANTENNA AT 1-WATT =  ,F12.5,//     ALL FIELDS ARE NORMALIZED
     2 TO THIS POWER // * * * * * * * * * * * * * * * * * * * * * * )
      DO 418 I=1,N
C
C  NORMALIZE THE ANTENNA CURRENTS TO AN EQUIAVLENT 1 WATT DRIVE FOR
C  FIELD CALCULATIONS.
C
      AIR(I) = AIR(I)*WATTI
      AII(I) = AII(I)*WATTI
      BIR(I) = BIR(I)*WATTI
      BII(I) = BII(I)*WATTI
      CIR(I) = CIR(I)*WATTI
      CII(I) = CII(I)*WATTI
 418  CONTINUE
       IP=NRPAGE
C
C
C  IF IDISK = 1, THEN WRITE THE A B AND C CURRENTS OUT ON DISK PLUS
C   OTHER REQUIRED DATA FOR DOING NEAR FIELD CALCULATIONS OFFLINE
C
      IF(IDISK .EQ. 0) GO TO 7906
      CALL ASSIGN(8,0,IOFILE,10000)
C
C  FIRST BUFFER OUT THE GEOM COMMON BLOCK
C
      BUFFER OUT(8,1)(N,COLAM)
 B1   IF(UNIT,8) B1,,,ERPAR
C
C  NEXT DO THE ANGL COMMON BLOCK
C
      BUFFER OUT (8,1)(CAB,SALP(NS))
 B2   IF(UNIT,8) B2,,,ERPAR
C
C
C   BUFFER OUT THE A,B,AND C CURRENTS
C
      BUFFER OUT (8,1)(AIR,CII(NS))
 B3   IF(UNIT,8) B3,,,ERPAR
C
C  BUFFER OUT THE MEDIA PARAMETERS
C
      BUFFER OUT (8,1)(EPSRP,SIGM)
 B4   IF(UNIT,8) B4,WEOF,WEOF,ERPAR
ERPAR WRITE(3,ERMSG)
      WRITE(100,ERMSG)
ERMSG FORMAT( PARITY ERROR GENERATED--DONT USE DISK FILE )
 WEOF CALL WRTEOF(8)
      CALL DEVICE( CLOSER ,IOFILE)
      WOT 3,FMTS,IOFILE
      WOT 100,FMTS,IOFILE
 FMTS  FORMAT(//, SCRATCH FILE FOR THIS RUN IS  ,A10,//)
      IOFILE=IOFILE+1
      IF((IOFILE .INT. 77B) .EQ. 73B)
       IOFILE=IOFILE+46B
 7906  CONTINUE
      IF(ISURF .EQ. 1) CALL SURF
      IF(ISURF.EQ.1) CALL FRAME
      IF(ISURF.EQ.1) CALL SETCH(0,100.,1,0,0,0,0)
```

```
      WOT 3,899,ITIME*1.E-6,JTIME*1.E-6,KTIME*1.E-6
      WOT 100,899,ITIME*1.E-6,JTIME*1.E-6,KTIME*1.E-6
  899 FORMAT(// MATRIX FILL TIME= ,F12.4,   SECONDS ,/,
     .     FACTORING TIME= ,F12.4,   SECONDS ,/,
     .       SOLUTION TIME= ,F12.4,   SECONDS //)

GLOAD CONTINUE
DLOAD CONTINUE
773    CONTINUE
       GO TO 1000
 13    WRITE(3,9055) IERR
       WRITE(100,9055) IERR
      WOT 59,9055,IERR
 9055 FORMAT(17H STOP   ERROR NO.15)
1600   CONTINUE
      IF(ILOAD.NE.0) CALL DEVICE( DESTROY ,ICMMX)

      CALL FROST(24108,02008,BETA,IXT)
      TIMCPU=BETA(1)/60.E6 $$$ CONVERT TO MINUTES
      TIMIO=BETA(2)/60.E6 $$$ CONVERT TO MINUTES
      WOT 3,900,TIMCPU,TIMIO,TIMCPU+TIMIO
      WOT 100,900,TIMCPU,TIMIO,TIMCPU+TIMIO
  900 FORMAT(/// CPU TIME USED= ,F12.5,  MIN. ,/,
     .    I/O TIME USED= ,F12.5,  MIN. ,/,
     .   8X, TOTAL= ,F12.5,  MIN. )

      CALL EXIT
       END
```

```
      CODE ANALYSIS
      SUBROUTINE ANTPLOT
       COMMON/GEOM/N,NP,X(300),Y(300),Z(300),SI(300),BI(300),ALP(300),
     1 BET(300),ICON1(300),ICON2(300),COLAM
      LABEL=0
C
C  FIND THE MAXIMUM VALUES OF X, Y, AND Z
C
      XMAX=AMAXAF(X,1,500)
      YMAX=AMAXAF(Y,1,500)
      ZMAX=AMAXAF(Z,1,500)
      PLTMAX=XMAX
      IF(YMAX .GT. XMAX) PLTMAX=YMAX
      IF(ZMAX .GT. PLTMAX) PLTMAX=ZMAX
      CALL FRAME
      CALL DDERS(1)
      CALL MAP  (-PLTMAX,PLTMAX,-PLTMAX,PLTMAX,.14,.85,.19,.90)
      CALL SETCRT(0.,0.,1,0)
C
C  PLOT PLAN VIEW X-Y PLANE
C
      DO 10 I=1,N
      IF(ALP(I) .GT. 1.5) GO TO 10
      DX = SI(I)*COSF(ALP(I))*COSF(BET(I))/2.
      DY = SI(I)*COSF(ALP(I))*SINF(BET(I))/2.
      X1 = X(I) - DX
      X2 = X(I) + DX
      Y1 = Y(I) - DY
      Y2 = Y(I) + DY
      CALL PLOTV(Y1,X1,Y2,X2)
   10 CONTINUE
C
C  LABEL THE PLOT
C
      CALL SETCH (45.,5.,1,0,2,0)
      CALL CRTBCD(10H PLAN VIEW       )
C
C  LABEL THE SEGMENTS
C
      IF(LABEL .EQ. 0) GO TO 11
      DO 11 I=1,N
      IF(ALP(I) .GT. 1.5) GO TO 11
      CALL SETLCH(X(I),Y(I),1,0,0)
      WOT 100 ,12, I
   12 FORMAT(1X,I3)
   11 CONTINUE
      CALL FRAME
C
C  PLOT THE Y-Z  PLANE
C
      CALL MAP (-PLTMAX,PLTMAX,-PLTMAX,PLTMAX,.14,.85,.19,.90)
      CALL SETCRT(0.,0.,1,0)
      CALL LINE(-PLTMAX*1.1,0.,PLTMAX*1.1,0.)
      DO 20,I=1,N
      IF(BET(I) .GT. 1.60 .AND. BET(I) .LT. 4.65) GO TO 20
      DY = SI(I)*COSF(ALP(I))*SINF(BET(I))/2.
      DZ = SI(I)*SINF(ALP(I))/2.
      Y1 = Y(I) - DY
      Y2 = Y(I) + DY
      Z1 = Z(I) - DZ
      Z2 = Z(I) + DZ
      CALL PLOTV(Y1,Z1,Y2,Z2)
   20 CONTINUE
C
```

```
C  LABEL THE PLOT
C
      CALL SETCH(45.,5.,1,0,2,0)
      CALL CRTBCD (11H FRONT VIEW   )
C
C  LABEL SEGMENTS
C
      IF(LABEL .EQ. 0) GO TO 21
      DO 21 I=1,N
      IF(BET(I) .GT. 1.60 .AND. BET(I) .LT. 4.65) GO TO 21
      CALL SETLCH(Y(I),Z(I),1,0,0)
      WOT 100,12,I
 21   CONTINUE
      CALL FRAME
C
C  PLOT THE X-Z PLANE
C
      CALL MAP (-PLTMAX,PLTMAX,-PLTMAX,PLTMAX,.14,.85,.19,.90)
      CALL SETCRT (0.,0.,1,0)
      CALL LINE(-PLTMAX*1.1,0.,PLTMAX*1.1,0.)
      DO 40 I=1,N
      IF(BET(I) .GT. 3.15 .OR. BET(I) .LT. 0.) GO TO 40
      DX=SI(I)*COSF(ALP(I))*COSF(BET(I))/2.
      DZ=SI(I)*SINF(ALP(I))/2.
      X1=X(I)-DX
      X2=X(I)+DX
      Z1=Z(I)-DZ
      Z2=Z(I)+DZ
      CALL PLOTV(X1,Z1,X2,Z2)
 40   CONTINUE
C
C  LABEL THE PLOT AND THE SEGMENTS
C
      CALL SETCH(45.,5.,1,0,2,0)
      CALL CRTBCD(10H SIDE VIEW)
      IF(LABEL .EQ. 0) GO TO 41
      DO 41 I=1,N
      IF(BET(I) .GT. 3.15 .OR. BET(I) .LT. 0) GO TO 41
      CALL SETLCH(X(I),Z(I),1,0,0)
      WOT 100,12,I
 41   CONTINUE
      CALL FRAME
C
C  PLOT AN ISOMETRIC VIEW OF THE ANTENNA
C
      THETA = 0.785
      ST = SINF(THETA)
      CT = COSF(THETA)
      SCALE=PLTMAX*1.71
      CALL MAP (-PLTMAX,SCALE,-PLTMAX,SCALE,.14,.85,.19,.90)
      CALL SETCRT(0.,0.,1,0)
      DO 30 I=1,N
      DX = SI(I)*COSF(ALP(I))*COSF(BET(I))/2.
      DY = SI(I)*COSF(ALP(I))*SINF(BET(I))/2.
      DZ = SI(I)*SINF(ALP(I))/2.
      X1 = X(I) - DX
      Y1 = Y(I) - DY
      Z1 = Z(I) - DZ
      X2 = X(I) + DX
      Y2 = Y(I) + DY
      Z2 = Z(I) + DZ
C
C  CONVERT TO PERSPECTIVE COORDINATES
C
```

```
      XIPLOT = YI - XI*CT
      YIPLOT = ZI - XI*ST
      X2PLOT = Y2 - X2*CT
      Y2PLOT = Z2 - X2*ST
      CALL PLOTV(XIPLOT,YIPLOT,X2PLOT,Y2PLOT)
30    CONTINUE
      CALL SETCH(45.,5.,1,0,2,0)
      CALL CRTBCD(15H ISOMETRIC VIEW )
      RETURN
      END
```

```
      CODE ANALYSIS
      FUNCTION CANG(A)
      DATA (PI=3.14159265359)
      COMPLEX A
      X=REAL(A)
      Y=AIMAG(A)
      IF (X.NE.0) GO TO 10
      ANG=0.
      IF (Y .GT. 0) ANG = PI/2.
      IF (Y .LT. 0) ANG=-PI/2.
      CANG=ANG
      RETURN
10    CONTINUE
      ANG = ATAN(ABSF(Y)/ABSF(X))
      IF (X .GT. 0) GO TO 20
      IF (Y .GE. 0) ANG = PI-ANG
      IF (Y .LT. 0) ANG = ANG-PI
      CANG = ANG
      RETURN
20    CONTINUE
      IF (Y .LT. 0) ANG = -ANG
      CANG = ANG
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE CSINCOS(Z,SINZ,COSZ)
      X=REAL(Z)
      Y=AIMAG(Z)
      SX=SINF(X)
      CX=COSF(X)
      IF(ABSF(Y) .LT. 1.E-6) GO TO 1
      EY=EXPF(Y)
      GO TO 2
1     EY=1.+Y
2     SY=1./EY
      CY=(EY+SY)*.5
      SY=(EY-SY)*.5
      SINZ=CMPLX(SX*CY,CX*SY)
      COSZ=CMPLX(CY*CX,-SX*SY)
      RETURN
      END
```

```
      CODE ANALYSIS
       SUBROUTINE DATAGN1
      PARAMETER (MS = 300)
       COMMON/GEOM/N,NP,X(MS),Y(MS),Z(MS),SI(MS),BI(MS),ALP(MS),
     1  BET(MS),ICON1(MS),ICON2(MS),COLAM
       COMMON/CSEG/ISEG(10),NDIP
       TA=0.017453292
      READ(2,900)HI,ROTAT,ELEV,NDIP
      WRITE(3,900)HI,ROTAT,ELEV,NDIP
 900   FORMAT(3F10.5,I5)
       RLAM=COLAM*8.467E-2
       HI=HI*RLAM
       ROTAT=ROTAT*TA
       ELEV=ELEV*TA
       IE2=0
       DO 10 I=1,NDIP
      WRITE(3,901)X1,X2,Z2,SORCL,WRAD,NS
 901   FORMAT(5F10.5,I5)
       X1=X1*RLAM
       X2=X2*RLAM
       Z2=Z2*RLAM
       SORCL=SORCL*RLAM
       WRAD=WRAD*RLAM
       SORCH=SORCL*.5
       CALL LINE2(X1,0.,-SORCH,IE2+1,SL,ALF,BUT,X1,0.,SORCH,IC,1,1,WRAD)
       CALL LINE2(X1,0.,SORCH,IC+1,EL,ALF,BUT,X2,0.,Z2,IE1,NS,1,WRAD)
       CALL LINE2(X1,0.,-SORCH,IE1+1,EL,ALF,BUT,X2,0.,-Z2,IE2,NS,1,WRAD)
       ISEG(I)=IC
       ICON1(IC)=-1
       ICON1(IE1+1)=-1
       ICON2(IE1)=0
       ICON2(IE2)=0
      EL=2.*EL+SL
      WRITE(3,902) I,EL
  902 FORMAT(11H DIPOLE NO.I5I1H     LENGTH=F10.5)
 10    CONTINUE
      N=IE2 $ NP=IE2
       CALL MOVE(0.,0.,HI,ROTAT,ELEV,0.)
       RETURN
       END
```

```
CODE ANALYSIS
 SUBROUTINE DATAGN2
PARAMETER (MS = 300)
 COMMON/GEOM/N,NP,X(MS),Y(MS),Z(MS),SI(MS),BI(MS),ALP(MS),
1 BET(MS),ICON1(MS),ICON2(MS),COLIU
 COMMON/CSEG/ISEG(10),NDIP
 TA=0.017453292
READ(2,900) HI,ROTAT,ELEV,PIVHT,NDIP
WRITE(2,900)HI,ROTAT,ELEV,PIVHT,NDIP
900  FORMAT(4F10.5,I5)
 RLAM=0.0254
 PIVHT=PIVHT*RLAM
 HI=HI*RLAM
 ROTAT=ROTAT*TA
 ELEV=ELEV*TA
 IE2=0
 DO 10 I=1,NDIP
READ(2,901) X1,X2,Z2,SORCL,WRAD,NS
WRITE(3,901)X1,X2,Z2,SORCL,WRAD,NS
901  FORMAT(5F10.5,I5)
 X1=X1*RLAM
 X2=X2*RLAM
 Z2=Z2*RLAM
 SORCL=SORCL*RLAM
 WRAD=WRAD*RLAM
 SORCH=SORCL*.5
 CALL LINE2(X1,0.,-SORCH,IE2+1,SL,ALF,BUT,X1,0.,SORCH,IC,1,1,WRAD)
 CALL LINE2(X1,0.,SORCH,IC+1,EL,ALF,BUT,X2,0.,Z2,IE1,NS,1,WRAD)
 CALL LINE2(X1,0.,-SORCH,IE1+1,EL,ALF,BUT,X2,0.,-Z2,IE2,NS,1,WRAD)
 ISEG(I)=IC
 ICON1(IC)=-1
 ICON1(IE1+1)=-1
 ICON2(IE1)=0
 ICON2(IE2)=0
 EL=2.*EL+SL
WRITE(3,902) I,EL
 902 FORMAT(11H DIPOLE NO.I511H    LENGTH=F10.5)
10   CONTINUE
N=IE2 $ NP=IE2
 CALL MOVE(0.,-PIVHT,0.,0.,0.,0.)
 CALL MOVE(0.,0.,HI,-ROTAT,0.,0.)
READ(2,902) TX1,TZ1,TX2,TZ2,TX3,TZ3,TWRD,NS1,NS2
WRITE(3,902)TX1,TZ1,TX2,TZ2,TX3,TZ3,TWRD,NS1,NS2
903  FORMAT(7F10.5,2I5)
 TX1=TX1*RLAM
 TZ1=TZ1*RLAM
 TX2=TX2*RLAM
 TZ2=TZ2*RLAM
 TX3=TX3*RLAM
 TZ3=TZ3*RLAM
 TWRD=TWRD*RLAM
 IE1=N+1
 CALL LINE2(TX1,0.,TZ1,IE1,EL,ALF,BUT,TX2,0.,TZ2,IE2,NS1,1,TWRD)
 CALL LINE2(TX2,0.,TZ2,IE2+1,EL,ALF,BUT,TX3,0.,TZ3,IE2,NS2,1,TWRD)
 ICON1(IE1)=0
 ICON2(IE2)=0
N = IE2
NP = IE2
 CALL MOVE(0.,0.,0.,0.,-ELEV,0.)
 RETURN
 END
```

```
      CODE ANALYSIS
       SUBROUTINE DATAGN6
      PARAMETER (NS = 300)
       COMMON/GEOM/N,NP,X(NS),Y(NS),Z(NS),SI(NS),BI(NS),ALP(NS),
     1  BET(NS),ICON1(NS),ICON2(NS),COLAM
      WRITE(3,901)
  901 FORMAT(///39H COLLOCATION PROGRAM FOR LINEAR DIPOLES/)
       NP=1
       PI=3.141592654
       I1=1
       N=0
C***** COLAM IS THE DIPOLE LENGTH IN METERS
    1 READ(2,9001) EL,ALF,BUT,XC,YC,ZC,SOEL,M,ICONT
 9001  FORMAT(7F10.5,2I5)
       A=SOEL*EL
       N=N+M
       AL=ALF*0.01745329252
       BT=BUT*0.01745329252
      CA=COSF(AL)
       ELO2=EL*.5
      X1=XC-ELO2*CA*COSF(BT)
      Y1=YC-ELO2*CA*SINF(BT)
      Z1=ZC-ELO2*SINF(AL)
       CALL LINE1(X1,Y1,Z1,I1,EL    ,AL,BT,X2,Y2,Z2,I2,M,1,A)
       ICON1(I1)=0
       ICON2(I2)=0
       I1=I2+1
       IF(ICONT.NE.0) GO TO 1
       NP=N
       RETURN
       END
```

```
      CODE ANALYSIS
      SUBROUTINE DATATIP
      PARAMETER (NS = 300)
       COMMON/GEOM/N,NP,X(NS),Y(NS),Z(NS),SI(NS),BI(NS),ALP(NS),
     1  BET(NS),ICON1(NS),ICON2(NS),COLAM
      DATA (IBIAS = 1RA)
      FACTOR = 0.3048
      ISEG = 1
C
C  READ IN 2 DATA CARDS FOR EACH MAJOR LINE. LINE LABEL WILL DETERMINE
C  WHICH OPTION WILL BE USED FOR DATA GENERATION.
C
    1 READ (2,1000) LINE,X1,Y1,Z1,NCON1,X2,Y2,Z2,NCON2
 1000 FORMAT (1R1,3F10.5,I5,3F10.5,I5)
      READ (2,1001) NSEGS,TAU,WT,TENS,WIREDIA
 1001 FORMAT(I5,4F10.5)
C
C  CONVERT DIMENSIONS IN FEET TO METERS
C
      X1 = X1 * FACTOR
      Y1 = Y1 * FACTOR
      Z1 = Z1 * FACTOR
      X2 = X2 * FACTOR
      Y2 = Y2 * FACTOR
      Z2 = Z2 * FACTOR
      WIREDIA = WIREDIA * FACTOR
      WIRELTH = WIRELTH * FACTOR
      IOPT = LINE - IBIAS + 1
      GO TO (10,20,30,40,50,60,70,80,90)  IOPT
C
C  LINE  A--NO SYMMETRIC ELEMENTS
C
   10 CALL LINE3 (ISEG,NSEGS,TAU,WT,TENS,WIREDIA,X1,Y1,Z1,X2,Y2,Z2)
      NSEGSA = NSEGS
      ICON1(ISEG) = ISEG
      ISEG = ISEG + NSEGS
      ICON2(ISEG - 1) = NCON2
      GO TO 1
C
C  LINE B
C
   20 CALL LINE3 (ISEG,NSEGS,TAU,WT,TENS,WIREDIA,X1,Y1,Z1,X2,Y2,Z2)
      NSEGSB = NSEGS
      CALL XFRM (ISEG,NSEGS,NLAST)
      CALL ENDS (ISEG,NSEGS,NCON1,NCON2,0,3)
      ISEG = NLAST
      GO TO 1
C
C  LINE C
C
   30 CALL LINE3 (ISEG,NSEGS,TAU,WT,TENS,WIREDIA,X1,Y1,Z1,X2,Y2,Z2)
      NSEGSC = NSEGS
      CALL XFRM (ISEG,NSEGS,NLAST)
      CALL ENDS (ISEG,NSEGS,NCON1,NCON2,0,1)
      ISEG = NLAST
      GO TO 1
C
C  LINE D
C
   40 CALL LINE3 (ISEG,NSEGS,TAU,WT,TENS,WIREDIA,X1,Y1,Z1,X2,Y2,Z2)
      NSEGSD = NSEGS
      CALL XFRM (ISEG,NSEGS,NLAST)
      CALL ENDS (ISEG,NSEGS,NCON1,NCON2,0,1)
C
```

```
C  LINE I IS MIRROR OF LINE D
C
      CALL FLOP(ISEG,NSEGS)
      ISEG = NLAST
      CALL XFRM (ISEG,NSEGS,NLAST)
      NCON2 = NCON2 - 4
      CALL ENDS (ISEG,NSEGS,NCON1,NCON2,1,1)
      ISEG = NLAST
      GO TO 1
C
C  LINE  E
C
 50   CALL LINE3 (ISEG,NSEGS,TAU,WT,TENS,WIREDIA,X1,Y1,Z1,X2,Y2,Z2)
      NSEGSD = NSEGS
      CALL XFRM (ISEG,NSEGS,NLAST)
      CALL ENDS (ISEG,NSEGS,NCON1,NCON2,0,1)
C
C  LINE  J IS MIRROR OF E
C
      CALL FLOP (ISEG,NSEGS)
      ISEG = NLAST
      CALL XFRM (ISEG,NSEGS,NLAST)
      NCON2 = NCON2 - 4
      CALL ENDS (ISEG,NSEGS,NCON1,NCON2,1,1)
      ISEG = NLAST
      GO TO 1
C
C  LINE  F
C
 60   CALL LINE3 (ISEG,NSEGS,TAU,WT,TENS,WIREDIA,X1,Y1,Z1,X2,Y2,Z2)
      NSEGSF = NSEGS
      CALL XFRM (ISEG,NSEGS,NLAST)
      CALL ENDS (ISEG,NSEGS,NCON1,NCON2,0,1)
C
C  LINE  H IS MIRROR OF LINE F
C
      CALL FLOP(ISEG,NSEGS)
      ISEG = NLAST
      CALL XFRM (ISEG,NSEGS,NLAST)
      NCON1 = NCON1 - 4
      CALL ENDS (ISEG,NSEGS,NCON1,NCON2,0,1)
      ISEG = NLAST
      GO TO 1
C
C  LINE G  TOWER
C
 70   CALL LINE3 (ISEG,NSEGS,TAU,WT,TENS,WIREDIA,X1,Y1,Z1,X2,Y2,Z2)
      NSEGSG = NSEGS
      CALL XFRM (ISEG,NSEGS,NLAST)
      CALL ENDS (ISEG,NSEGS,NCON1,NCON2,0,2)
      ISEG = NLAST
      GO TO 1
C
C   AN ALPHA CHARACTER OF I WILL ALLOW ONE TO GENERATE A SINGLE LINE
C   AT THE SPECIFIED COORDINATES AND LABEL THE ENDS
C
 90   CALL LINE3 (ISEG,NSEGS,TAU,WT,TENS,WIREDIA,X1,Y1,Z1,X2,Y2,Z2)
      ICON1(ISEG) = NCON1
      ISEG = ISEG + NSEGS
      ICON2(ISEG-1) = NCON2
      GO TO 1
C
C  AN ALPHA CHARACTER OF H WILL TERMINATE THE INPUT DATA
C
```

```
 80   CONTINUE
      N = NP = ISEG - 1
      RETURN
      END

      CODE ANALYSIS
      SUBROUTINE EFLD(B,S,RH,ZP,IJ,EZS,ERS,EZC,ERC,EZK,ERK,MEDIA)
      COMPLEX EZS,ERS,EZC,ERC,EZK,ERK
      COMPLEX WK,WKI
      COMPLEX ER1,ER2,TR1,TR2,TX1,TX2,EINT,ZD1,ZD2,CST,SST
      COMMON/TMI/ZPX,RB2,IJX,-COMPLEX-FJWK
      COMMON/FREQ/FREQ,-COMPLEX-WKP,WKM,ECONST

      WK=WKP
      IF(MEDIA.EQ.-1) WK=WKM
      WKI=1./WK

COMMENT----FILL COMMON BLOCK TMI FOR USE BY GF WHEN CALLED BY INTX
      IJX=IJ
      ZPX=ZP
      FJWK=CMPLX(AIMAG(WK),-REAL(WK)) $$$ -FJ*WK

      RB2=RH*RH+B*B
      RB=SQRTI(RB2)
      SH=.5*S
      R1=SQRTI(RB2+(ZP+SH)**2)
      ER1=CEXP(CMPLX(R1*AIMAG(WK),-R1*REAL(WK)))
      R2=SQRTI(RB2+(ZP-SH)**2)
      ER2=CEXP(CMPLX(R2*AIMAG(WK),-R2*REAL(WK)))

      ZD1=CMPLX(ZP*REAL(WK),ZP*AIMAG(WK))+CMPLX(SH*REAL(WK),SH*AIMAG(WK))
      ZD2=CMPLX(ZP*REAL(WK),ZP*AIMAG(WK))-CMPLX(SH*REAL(WK),SH*AIMAG(WK))
      CALL CSINCOS(CMPLX(SH*REAL(WK),SH*AIMAG(WK)),SST,CST)

      ER1=ER1*CMPLX(REAL(WKI*ECONST)/R1,AIMAG(WKI*ECONST)/R1)
      ER2=ER2*CMPLX(REAL(WKI*ECONST)/R2,AIMAG(WKI*ECONST)/R2)
      TR1=CMPLX(-AIMAG(WKI)/R1,REAL(WKI)/R1)+
     .   CMPLX(REAL(WKI)/R1,AIMAG(WKI)/R1)**2
      TR2=CMPLX(-AIMAG(WKI)/R2,REAL(WKI)/R2)+
     .   CMPLX(REAL(WKI)/R2,AIMAG(WKI)/R2)**2
      TX1=TR1*ZD1
      TX2=TR2*ZD2

      EZS=(CST-TX2*SST)*ER2-(CST+TX1*SST)*ER1
      EZC=-(SST+TX2*CST)*ER2-(SST-TX1*CST)*ER1
      ERS=-((ZD2*CST+SST-TX2*ZD2*SST)*ER2-(ZD1*CST-SST+TX1*ZD1*SST)*ER1)*
     .   CMPLX(REAL(WKI)*(RH/(RB*RB)),AIMAG(WKI)*(RH/(RB*RB)))
      ERC=((ZD2*SST-CST+TX2*ZD2*CST)*ER2+(ZD1*SST+CST-TX1*ZD1*CST)*ER1)*
     .   CMPLX(REAL(WKI)*(RH/(RB*RB)),AIMAG(WKI)*(RH/(RB*RB)))
      EZK=-TX2*ER2+TX1*ER1
      ERK=CMPLX(-RH*REAL(WK),-RH*AIMAG(WK))*(TR2*ER2-TR1*ER1)
      CALL INTX(-SH,+SH,B,IJ,EINT)
      EZK=EZK-ECONST*EINT
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE ENDS (ISEG,NSEGS,N1,N2,MIRROR,NTYPE)
      PARAMETER (NS = 300)
       COMMON/GEOM/N,NP,X(NS),Y(NS),Z(NS),SI(NS),BI(NS),ALP(NS),
     1  BET(NS),ICON1(NS),ICON2(NS),COLAM
C
C  MIRROR = 1 FOR MIRROR IMAGE AND NUMBERING, OTHERWISE MAKE MIRROR = 0
C  NTYPE = 1 FOR NORMAL LABELING, = 2 FOR GND PLANE, = 3 FOR FREE END
C
      DO 10 J = 1,4
      IFIRST = ISEG + NSEGS*(J-1)
      ILAST = IFIRST + NSEGS - 1
      IF (MIRROR .EQ. 1) 20,30
 20   IF (J .EQ. 1) 21,22
 21   ICON1(IFIRST) = N1 - 3
      GO TO 40
 22   ICON1(IFIRST) = N1 - J + 2
      GO TO 40
 30   GO TO (31,32,33) NTYPE
 31   ICON1(IFIRST) = N1 - J + 1
      GO TO 40
 32   ICON1(IFIRST) = IFIRST
      GO TO 40
 33   ICON1(IFIRST) = N1
 40   ICON2(ILAST) = N2 - J + 1
  10  CONTINUE
      RETURN
      END
```

```
      CODE ANALYSIS
      COMPLEX F,FBAR
      FUNCTION F(W,R,ER,S)
      PARAMETER (PI=3.14159265358979323846264З,MU=12.56636E-7,E0=8.854E-12)
      B0=W*SQRT(MU*E0)
      COMPLEX XNN
      XNN=CMPLX(ER,-S/(W*E0))
      F=FBAR(CMPLX(0,-B0*R*.5)*((XNN-1.)/(XNN*XNN)))
      RETURN
      END
```

```
      CODE ANALYSIS
      OPTIMIZE
       SUBROUTINE FACTOR(N,P,NDIM)
C
C SUBROUTINE TO FACTOR A MATRIX INTO A UNIT LOWER TRIANGULAR MATRIX AND AN
C UPPER TRIANGULAR MATRIX USING THE GAUSS-DOOLITTLE ALGORITHM PRESENTED ON
C PAGES 411-416 OF A. RALSTON--A FIRST COURSE IN NUMERICAL ANALYSIS.  COMMENTS
C BELOW REFER TO COMMENTS IN RALSTONS TEXT.
C
      LCM (333)
       COMPLEX CM,D,DETER
       INTEGER R,P,RM1,RP1,PJ,PR
       DIMENSION P(NDIM)
      COMMON /333/ CM(1)
      PARAMETER (NS = 300)
      COMMON /SCRATM/ D(NS)
       IFLG=0
       DO 60 R=1,N
C
C STEP 1
C
       DO 10 K=1,N
       D(K)=CM(K+N*(R-1))
   10  CONTINUE
C
C STEPS 2 AND 3
C
       RM1=R-1
       IF(RM1.LT.1) GO TO 31
       DO 30 J=1,RM1
       PJ=P(J)
       CM(J+N*(R-1))=D(PJ)
       D(PJ)=D(J)
       JP1=J+1
      COMPLEX XCM
      NJ1=N*(J-1)
      XCM=CM(J+N*(R-1))
       DO 20 I=JP1,N
      D(I)=D(I)-CM(I+NJ1)*XCM
   20  CONTINUE
   30  CONTINUE
   31  CONTINUE
C
C STEP 4
C
       DMAX=D(R)*CONJG(D(R))
       P(R)=R
       RP1=R+1
       IF(RP1.GT.N) GO TO 41
       DO 40 I=RP1,N
       ELMAG=D(I)*CONJG(D(I))
       IF(ELMAG.LT.DMAX) GO TO 40
       DMAX=ELMAG
       P(R)=I
   40  CONTINUE
   41  CONTINUE
       IF(DMAX.LT.1.E-10) IFLG=1
       PR=P(R)
       CM(R+N*(R-1))=D(PR)
       D(PR)=D(R)
C
C STEP 5
C
       IF(RP1.GT.N) GO TO 51
```

```
       DO 50 I=RP1,N
       CM(I+N*(R-1))=D(I)/CM(R+N*(R-1))
  50   CONTINUE
  51   CONTINUE
 100   FORMAT(1H ,4(E16.8,E16.8))
       IF(IFLG.EQ.0) GO TO 60
      WRITE(3,102) R,DMAX
  102 FORMAT( 7H PIVOT(I32H)=E16.8)
       IFLG=0
  60   CONTINUE
C      WRITE(3,101) (P(R),R=1,N)
 101   FORMAT(1H ,24I5)
      DETER=CMPLX(1.,0.)
      RETURN
      WRITE(3,105)
 105  FORMAT(10H   R        ,5H DMAG    )
       DO 70 R=1,N
       DMAG=CABS(DETER)
      WRITE(3,106) R,DMAG
 106  FORMAT(1H ,I5,E10.2)
       IF(DMAG.GT.1.0E300.OR.DMAG.LT.1.0E-270) GO TO 80
       DETER=DETER*CM(R+N*(R-1))
  70   CONTINUE
      WRITE(3,103) DETER
  103 FORMAT(14H0DETERMINANT=(E16.8,1H,E16.81H))
       DMAG=CABS(DETER)
       IF(DMAG.EQ.0.) CALL EXIT
       RETURN
 80   WRITE(3,104) DMAG,R
      CONTINUE
  104 FORMAT(23H0DETERMINANT MAGNITUDE=E16.88H   AT R=I3)
       RETURN
       END
```

```
      CODE ANALYSIS
      SUBROUTINE FIELDS(ETV,ETH,EPH,R,P,T,XS,YS,ZS,A,PS,C,F,
     *  EPSRI,SIGI)
      PARAMETER (PI=3.14159265,MU=12.56636E-7,E0=8.854E-12)
      COMPLEX RH,RE,FBAR
      COMPLEX COMFAC,FERH,FE,FM,EXPP,EXPM,ETV,ETH,EPH,FPRIME
      COMPLEX PE,PM
      COMPLEX C
      COMMON /CXNN/ -COMPLEX-XNN
      W=2.*PI*F
      TP=2.*PI
      STHET=SIN(T)
      CTHET=COS(T)
      SPHI=SIN(P)
      CPHI=COS(P)
      XNN = CMPLX(EPSRI,-SIGI/(W*E0))
      B0=W*SQRT(MU*E0)
      EXPP=CEXP(CMPLX(0.,B0*ZS*CTHET))
      EXPM=CEXP(CMPLX(0.,-B0*ZS*CTHET))

      COMFAC=CMPLX(0,(W*MU/(4.*PI*R)))*C*
     .  CEXP(CMPLX(0,-B0*R+B0*(XS*STHET*CPHI+
     .    YS*STHET*SPHI)))
      IF(T .GE. PI/2.) GO TO OVERLAND
      IF (ZS .LT. 0) GO TO BELOW

COMMENT----SOURCE ABOVE SURFACE----COMPUTE SPACE WAVE
      ETV=-COMFAC*SIN(A)*STHET*(EXPP+RH(T)*EXPM)
      ETH=-COMFAC*COS(A)*COS(P-PS)*(CTHET*(EXPP-RH(T)*EXPM))
      EPH=-COMFAC*COS(A)*SIN(P-PS)*(EXPP+RE(T)*EXPM)


      RETURN

 OVERLAND  CONTINUE   $$$ TOTAL PATH IS OVER LAND
      PM = CMPLX(0,-B0*R/2.)*(XNN-1.)
      PE = PM/(XNN*XNN)
      FERH = (1.-RH(T))*FBAR(PE)
      FM = FBAR(PM)
      ETV = -COMFAC*SIN(A)*FERH*EXPM
      ETH = -COMFAC*COS(A)*COS(P-PS)*FERH*EXPM*CSQRT(XNN-1.)/XNN
      IF(ZS .GE. 0.) RETURN
COMMENT--FOR BURIED SOURCES THE SURFACE WAVE IS CALCULATED AS AN
C          ELEVATED SOURCE AND THEN THE FIELDS ARE MODIFIED BY THE
C          FOLLOWING FACTOR.
      FPRIME=CEXP(CMPLX(0.,B0)*ZS*CSQRT(XNN-1.))
      ETH=ETH*FPRIME
      ETV=ETV*FPRIME/XNN
      RETURN

COMMENT--SOURCE BELOW SURFACE--CALCULATE SPACE WAVE
  BELOW CONTINUE
      COMFAC=COMFAC*CEXP(CMPLX(0,B0)*ZS*CSQRT(XNN-STHET**2))
      ETV=COMFAC*SIN(A)*STHET*(1.+RH(T))/XNN
      ETH=-COMFAC*COS(A)*COS(P-PS)*((1.+RH(T))/XNN)*CSQRT(XNN-STHET**2)
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE FLOP(IFIRST,ISEGS)
      PARAMETER (NS = 300)
       COMMON/GEOM/N,NP,X(NS),Y(NS),Z(NS),SI(NS),BI(NS),ALP(NS),
     1  BET(NS),ICON1(NS),ICON2(NS),COLAM
      PIO2 = 3.141592654 / 2.
      IBIAS = 4* ISEGS
      IX = IFIRST + IBIAS
      IEND = IFIRST + ISEGS - 1
      DO 100 I = IFIRST,IEND
      X(IX) = Y(I)
      Y(IX) = X(I)
      Z(IX) = Z(I)
      SI(IX) = SI(I)
      BI(IX) = BI(I)
      ALP(IX) = ALP(I)
      BET(IX) = PIO2 - BET(I)
      ICON1(IX) = ICON1(I) + IBIAS
      ICON2(IX) = ICON2(I) + IBIAS
      IX = IX + 1
  100 CONTINUE
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE GF(ZK,CO,SI)
      COMPLEX E
      COMMON/TMI/ZPK,RKB2,IJ,-COMPLEX-FJWK
      IF(IJ),GSELF   $$$ IJ=0 MEANS DO SELF TERM
      RK=SQRTI(RKB2+(ZK-ZPK)**2)
      E=CEXP(CMPLX(RK*REAL(FJWK),RK*AIMAG(FJWK)))
      CO=REAL(CMPLX((1./RK)*REAL(E),(1./RK)*AIMAG(E)))
      SI=AIMAG(CMPLX((1./RK)*REAL(E),(1./RK)*AIMAG(E)))
      RETURN
GSELF CONTINUE
      RK=SQRTI(RKB2+(ZK-ZPK)**2)
      E=CEXP(CMPLX(RK*REAL(FJWK),RK*AIMAG(FJWK)))
      CO=REAL(CMPLX((1./RK)*(REAL(E)-1.),(1./RK)*AIMAG(E)))
      SI=AIMAG(CMPLX((1./RK)*(REAL(E)-1.),(1./RK)*AIMAG(E)))
      RETURN
      END
```

```
      SUBROUTINE GN(EZ,ER)
C
C     SUBROUTINE GN MODIFIES THE PERFECT IMIAGE FIELDS BY THE
C     APPROPRIATE REFLECTION COEFFICIENTS EVALUATED AT THE SPECULAR
C     POINTS.
C
      COMPLEX EZ,ER,ERX,ERY,ERZ,EPX,EPY,REFS,REFPS
      COMMON /REFL/ RHOX,RHOY,RHOZ,CABJ,SABJ,SALPR,PX,PY,REFS,REFPS
      ERX=RHOX*ER+CABJ*EZ
      ERY=RHOY*ER+SABJ*EZ
      ERZ=RHOZ*ER+SALPR*EZ
      EPY=PX*ERX+PY*ERY
      EPX=PX*EPY
      EPY=PY*EPY
      ERX=REFS*ERX+REFPS*EPX
      ERY=REFS*ERY+REFPS*EPY
      ERZ=REFS*ERZ
      EZ=ERX*CABJ+ERY*SABJ+ERZ*SALPR
      ER=ERX*RHOX+ERY*RHOY+ERZ*RHOZ
      RETURN
      END
```

```
       CODE ANALYSIS
       SUBROUTINE INLINE(ID)


COMMENT----THE ONLY INPUT LINES ALLOWED ARE-
COMMENT----          TP2FILE BOX NNN
COMMENT----          TP2FILE TP3FILE
COMMENT----          TP2FILE TP3FILE DD80FILE
COMMENT----          TP2FILE TP3FILE BOX NNN
COMMENT----THE SYMBOL . MAY BE USED IF THERE IS NO TP2 FILE
COMMENT----IF DD80FILE IS CHARACTERS DD80 THE FILE WILL BE GIVEN AWAY
COMMENT----FOR THE LAST THREE SETS THE DD80 FILE IS KEPT ON DISK

       ALL INTEGER
       DIMENSION NSYMBOLS(10), ID(1)
       COMMON /GOBCOM/ GCOM(41)

COMMENT----WHO IS THE CONTROLLER

       DIMENSION CONTROLLER(2)
       CALL FROST (24258, 0, CONTROLLER,TTY)
       IF (CONTROLLER .EQ. 6R*ORDER) GO TO G44
  TTY  CONTINUE
COMMENT----ERROR RETURN MEANS TTY IS CONTROLLER


       CALL MESSAGE (16,1,-10, NSYMBOLS, NOMES)
       DO D1, I=1,5
       IF (NSYMBOLS(I) .EQ. 400204B ) LF=I
  D1   CONTINUE

       IF (LF .LT. 3 .OR. LF .GT. 5) ERROR,
       IF(NSYMBOLS(1).EQ.1R.) ,G33
       NFLAG=1
       GO TO NO2
   G33 CONTINUE
       CALL ASSIGN (2, NSYMBOLS(1))
       CALL DEVICE ( OPEN , NSYMBOLS(1), LNTH, IOERR)
       IF (IOERR .EQ. 0 .OR. IOERR .EQ. 2) GO TO OK
       WOT 59, FIOER, NSYMBOLS(1)
FIOER FORMAT (33H*INLINE*  FAILED TO OPEN TP2 FILE-  A10)
       CALL EXIT

  OK   CONTINUE
   NO2 CONTINUE
       IF(LF.EQ.3) GO TO G10
       IF(LF.EQ.4 .AND. NSYMBOLS(2).EQ.3RBOX) GO TO GCASE1
COMMENT---AT THIS POINT WE HAVE--TP2FILE TP3FILE DD80FILE
       IF(NSYMBOLS(3).EQ.4RDD80) GO TO GDD80 $$$ DONT KEEP ON DISK
COMMENT----THE IRX AND IRY IN NEXT LINES IS NOT A BUG
COMMENT----DESTROY EXISTING DD80 FILE, IF ANY
       CALL DEVICE( DESTROY ,(NSYMBOLS(3).SHL.6).UN.IRX)
       CALL DEVICE( DESTROY ,(NSYMBOLS(3).SHL.6).UN.IRY)
       CALL KEEP80((NSYMBOLS(3).SHL.6).UN IRY)
GDD80 CONTINUE
       GO TO G10
GCASE1 CONTINUE
       GCOM(36) = ((NSYMBOLS(LF-1).INT.7777777B) .SHL. 18) .UN. 3HBOX
       GCOM(37) = ID(1)   $ GCOM(38) = ID(2)
       WOT 59, FMT, (GCOM(I), I=36,38)
  FMT  FORMAT (18HOUTPUT WILL GO TO  3A10)
  G10  CONTINUE
       IF(LF.EQ.4 .AND. NSYMBOLS(2).EQ.3RBOX) GO TO G55
       L=2
```

```
      CALL DEVICE( CREATE ,NSYMBOLS(2),L)
      CALL DEVICE( DESTROY ,NSYMBOLS(2))
      CALL ASSIGN (3, 15, NSYMBOLS(2))
      IF(LF.EQ.3) CALL KEEP80(1)
 G55  CONTINUE
 G44  CONTINUE
      COMMON /NEWCOM/ NNEW,NOLD
      CALL CLOCK(TIME,DAY)
      WOT 3,FMT1,NOLD,GCOM(31),GCOM(32)
      CALL CRTID(ID,1)
      CALL SETCH(0,100.,1,0,0,0,0)
      WOT 100,FMT1,NOLD,GCOM(31),GCOM(32)
 FMT1 FORMAT(///,10X,17HCONTROLLEE NAMED:,A10,8X,11HWAS LOADED:,2A10)
      WOT 3,FMT2,NNEW,TIME,DAY
      WOT 100,FMT2,NNEW,TIME,DAY
 FMT2 FORMAT(10X,20HNAME WAS CHANGED TO:,A10,5X,9HTODAY IS:,2A10///)
      IF(NFLAG.EQ.1) RETURN
      WOT 3,FMT3,NSYMBOLS(1)
      WOT 100,FMT3,NSYMBOLS(1)
 FMT3 FORMAT(//,10X,38HTHE FOLLOWING IS INPUT DATA FROM FILE:,A10//)
COMMENT----WRITE INPUT DATA TO TAPE3 FILE AND DD80 FILE
      DIMENSION ICOMM(8)
 REED RIT 2, (8A10) ,(ICOMM(I),I=1,8)
      IF(EOF,2) G100,
      WOT 3, (1X,8A10) ,(ICOMM(I),I=1,8)
      WOT 100, (1X,8A10) ,(ICOMM(I),I=1,8)
      GO TO REED
 G100 CONTINUE
      REWIND (2)
      WOT 3, (//////)
      WOT 100, (//////)
      RETURN

 NOMES CONTINUE
ERROR CONTINUE
      WOT 59, FERR
 FERR FORMAT (25H*INLINE* ERROR---RESTART  )
      CALL EXIT
      END
```

```
      CODE ANALYSIS
       SUBROUTINE INTG(B,S,RH,ZP,Q1,QP2,ETR,ETI,DIL,DIK,IJ,IP,MEDIA,ETS,ETC,ETK)
      COMPLEX WKP,WKM,ECONST
      COMPLEX ET,ETS,ETC,ETK,CL,CK,SINL,COSL,SINK,COSK,SILK,CONS
      COMMON /FREQ/ FREQ,WKP,WKM,ECONST
       DIMENSION ETR(3),ETI(3),ET(3)
      COMPLEX EZS,ERS,EZC,ERC,EZK,ERK
       CALL EFLD(B,S,RH,ZP,IJ,EZS,ERS,EZC,ERC,EZK,ERK,MEDIA)
       IF(IP.NE.2)GO TO 4
      CALL GN(EZS,ERS)
      CALL GN(EZC,ERC)
      CALL GN(EZK,ERK)
    4 ETS=EZS*Q1+ERS*QP2
      ETC=EZC*Q1+ERC*QP2
      ETK=EZK*Q1+ERK*QP2
      IF(MEDIA .EQ. +1) CL=WKP*DIL
      IF(MEDIA .EQ. -1) CL=WKM*DIL
      IF(MEDIA .EQ. +1) CK=WKP*DIK
      IF(MEDIA .EQ. -1) CK=WKM*DIK
      CALL CSINCOS(CL,SINL,COSL)
      CALL CSINCOS(CK,SINK,COSK)
      SILK = SINL*COSK + SINK*COSL
      CONS = 1./(SINL + SINK - SILK)
      ET(1) = (SINK*ETK + (COSK-CMPLX(1.,0.))*ETS - SINK*ETC)*CONS
      ET(2) = (-SILK*ETK + (COSL-COSK)*ETS + (SINL+SINK)*ETC)*CONS
      ET(3) = (SINL*ETK + (CMPLX(1.,0.)-COSL)*ETS - SINL*ETC)*CONS
      ETR(1) = REAL(ET(1))
      ETI(1) = AIMAG(ET(1))
      ETR(2) = REAL(ET(2))
      ETI(2) = AIMAG(ET(2))
      ETR(3) = REAL(ET(3))
      ETI(3) = AIMAG(ET(3))
       RETURN
       END
```

```
      CODE ANALYSIS
      SUBROUTINE INTX(EL1,EL2,B,IJ,SG)
      COMPLEX SG
      STRUCTURE(SG,SGR/SGI)
      DATA(NX=1),(NM=65536),(NTS=4),(RX=1.E-4)
      Z=EL1
      ZE=EL2
      IF(IJ.EQ.0) ZE=0.
      S=ZE-Z
      EP=10*NM
      EP=S/EP
      ZEND=ZE-EP
      SGR=0.0
      SGI=0.0
      NS=NX
      NT=0
      CALL GF(Z,G1R,G1I)
1     DZ=S/NS
      DZOT=DZ*0.5
      ZP=Z+DZ
      IF(ZP-ZE)4,4,5
5     DZ=ZE-Z
      IF(ABSF(DZ)-EP)100,100,4
4     ZP=Z+DZOT
      CALL GF(ZP,G3R,G3I)
      ZP=Z+DZ
      CALL GF(ZP,G5R,G5I)
23    T00R=(G1R+G5R)*DZOT
      T00I=(G1I+G5I)*DZOT
      T01R=(T00R+DZ*G3R)*0.5
      T01I=(T00I+DZ*G3I)*0.5
      T10R=(4.0*T01R-T00R)/3.0
      T10I=(4.0*T01I-T00I)/3.0
      TE1R=TEST(T01R,T10R)
      TE1I=TEST(T01I,T10I)
      IF(TE1I-RX)49,49,20
49    IF(TE1R-RX)50,50,20
20    ZP=Z+DZ*0.25
      CALL GF(ZP,G2R,G2I)
      ZP=Z+DZ*0.75
      CALL GF(ZP,G4R,G4I)
      T02R=(T01R+DZOT*(G2R+G4R))*0.5
      T02I=(T01I+DZOT*(G2I+G4I))*0.5
      T11R=(4.0*T02R-T01R)/3.0
      T11I=(4.0*T02I-T01I)/3.0
      T20R=(16.0*T11R-T10R)/15.0
      T20I=(16.0*T11I-T10I)/15.0
      TE2R=TEST(T11R,T20R)
      TE2I=TEST(T11I,T20I)
      IF(TE2I-RX)48,48,21
48    IF(TE2R-RX)51,51,21
50    SGR=SGR+T10R
      SGI=SGI+T10I
60    NT=NT+2
      GO TO 52
51    SGR=SGR+T20R
      SGI=SGI+T20I
58    NT=NT+1
52    Z=Z+DZ
      IF(Z-ZEND)53,100,100
53    G1R=G5R
      G1I=G5I
      IF(NT-NTS)1,54,54
54    IF(NS-NX)1,1,55
```

```
55    NS=NS/2
      NT=1
      GO TO 1
21    NT=0
      IF(NS-NM)22,44,44
  44  WRITE(3,93) Z
   93 FORMAT(24H STEP SIZE LIMITED AT Z=F10.5)
      GO TO 51
22    NS=NS*2
      DZ=S/NS
      DZOT=DZ*0.5
      G5R=G3R
      G5I=G3I
      G3R=G2R
      G3I=G2I
      GO TO 23
100   CONTINUE
      IF(IJ)56,57,56
 57   SGR=2.*(SGR+LOGF((SQRTF(B*B+S*S)+S)/B))
      SGI=2.*SGI
56    CONTINUE
      RETURN
      END
```

```
      CODE ANALYSIS
       SUBROUTINE JMELS(ETR,ETI,NCP,JP,NCM,JM,I)
      LCM (333)
      PARAMETER (NS = 300)
       INTEGER P
       COMPLEX CM,FJ,EINC
      COMMON /333/ CM(1)
      COMMON /GEOM/ N
       DIMENSION JP(25),JM(25)
      FJ=CMPLX(0.,1.)
       IF(NCP.LT.1) GO TO 2
       DO 1 J=1,NCP
       JPJ=JP(J)
1      CM(I+N*(JPJ-1)   )=CM(I+N*(JPJ-1)   )+ETR+FJ*ETI
2      CONTINUE
       IF(NCM.LT.1) GO TO 4
       DO 3 J=1,NCM
       JMJ=JM(J)
3      CM(I+N*(JMJ-1)   )=CM(I+N*(JMJ-1)   )-ETR-FJ*ETI
4      CONTINUE
       RETURN
       END
```

```
      CODE ANALYSIS
       SUBROUTINE JUNC(J,JNO,NC1,NSEG1,NC2,NSEG2,D)
      PARAMETER (NS = 300)
       COMMON/GEOM/N,NP,X(NS),Y(NS),Z(NS),SI(NS),BI(NS),ALP(NS),
     1  BET(NS),ICON1(NS),ICON2(NS),COLAM
       DIMENSION NSEG1(25),NSEG2(25)
       NC1=0
       NC2=0
       SNC=0.0
       DO 10 I=1,N
       IF(ICON1(I)-JNO)2,1,2
 1     IF(I.EQ.J) GO TO 2
       NC1=NC1+1
       IF(NC1.GT.25) GO TO 99
       NSEG1(NC1)=I
       SNC=SNC+SI(I)
 2     IF(ICON2(I)-JNO)10,3,10
 3     IF(I.EQ.J) GO TO 10
       NC2=NC2+1
       IF(NC2.GT.25) GO TO 99
       NSEG2(NC2)=I
       SNC=SNC+SI(I)
 10    CONTINUE
       FC=NC1+NC2
       D=(SI(J)+SNC/FC)/2.0
       RETURN
  99  WRITE(3,91) JNO
    91 FORMAT(41H ERROR - TOO MANY CONNECTIONS TO JUNCTION(4))
      CALL EXIT
       END
```

```
      CODE ANALYSIS
       SUBROUTINE LINE1(X1,Y1,Z1,I1,EL,ALF,BUT,X2,Y2,Z2,I2,NSEG,NB,A)
      PARAMETER (NS = 300)
       COMMON/GEOM/N,NP,X(NS),Y(NS),Z(NS),SI(NS),BI(NS),ALP(NS),
     1  BET(NS),ICON1(NS),ICON2(NS),COLAM
      CA=COSF(ALF)
      SA=SINF(ALF)
      CAB=CA*COSF(BUT)
      SAB=CA*SINF(BUT)
       X2=X1+EL*CAB
       Y2=Y1+EL*SAB
       Z2=Z1+EL*SA
       GO TO 1
       ENTRY LINE2
       XINC=X2-X1
       YINC=Y2-Y1
       ZINC=Z2-Z1
      ELXY=SQRTF(XINC*XINC+YINC*YINC)
      EL=SQRTF(XINC*XINC+YINC*YINC+ZINC*ZINC)
       SA=ZINC/EL
       CA=ELXY/EL
      ALF=ASINF(SA)
       IF(ELXY.LT.1.E-20)GO TO 3
       SB=YINC/ELXY
       CB=XINC/ELXY
       BUT=ATAN2(SB,CB)
       GO TO 2
 3     SB=0.
       CB=1.
       BUT=0.
 2     CAB=CA*CB
       SAB=CA*SB
 1     I2=I1+(NSEG-1)*NB
       SI(I1)=EL/NSEG
       SO2=SI(I1)/2.0
       X(I1)=X1+SO2*CAB
       Y(I1)=Y1+SO2*SAB
       Z(I1)=Z1+SO2*SA
       ALP(I1)=ALF
       BET(I1)=BUT
       BI(I1)=A
       IS=I1+NB
       ICON1(I1)=I1-NB
       ICON2(I1)=IS
       XINC=SI(I1)*CAB
       YINC=SI(I1)*SAB
       ZINC=SI(I1)*SA
       IF(IS.GT.I2) GO TO 11
       DO 10 I=IS,I2,NB
        IL=I-NB
       X(I)=X(IL)+XINC
       Y(I)=Y(IL)+YINC
       Z(I)=Z(IL)+ZINC
       SI(I)=SI(I1)
       BI(I)=A
       ALP(I)=ALF
       BET(I)=BUT
        ICON1(I)=IL
 10     ICON2(I)=I+NB
 11    CONTINUE
       RETURN
       END
```

```
      SUBROUTINE LINE3 (I,NS,TAU,WT,TENS,WRAD,X1,Y1,Z1,X2,Y2,Z2)
      PARAMETER (MS = 300)
       COMMON/GEOM/N,NP,X(MS),Y(MS),Z(MS),SI(MS),BI(MS),ALP(MS),
     1  BET(MS),ICON1(MS),ICON2(MS),COLAM
C
C  THIS SUBROUTINE IS USED TO CALCULATE THE GEOMETRIC COORDINATES OF
C  EACH MAJOR ANTENNA ARM.  THE DATA GENERATED BY THIS SUBROUTINE IS THE
C  X,Y,Z COORDINATE OF THE CENTER OF A SEGMENT PLUS THE ALPHA AND BETA
C  ORIENTATION ANGLES OF EACH POINT.  INTERCONNECTION DATA IS ALSO GENERATED FOR
C  EACH OF THE SEGMENTS.  BY SPECIFYING THE PROPER PARAMETERS IN THE SUB
C  CALL ONE HAS THE CHOICE OF USING A TAPERED SEGMENT LENGTH WITH  A
C  CATENARY FORM OR A LINEAR FORM.
C
      IF(TENS .LE. 1.) TENS = 1E100   $$$ IF NO TENSION,DONT USE CATENARY
      XINC = X2 - X1
      YINC = Y2 - Y1
      ZINC = Z2 - Z1
      RHO = SQRTF(XINC**2 + YINC**2 + ZINC**2)
      RHOXY = SQRTF(XINC**2 + YINC**2)
      BETA = ATAN2(YINC,XINC)
      EXPSUM = 0.
      NEXP = -1
C
C  CALCULATE SEGMENT LENGTH SLO.  IF TAPERED SEG IS USED SPECIFY TAU
C
      DO 10 LS = 1,NS
   10 EXPSUM = EXPSUM + (1.+TAU)**(LS-1)
      SLO = RHO / EXPSUM
C
C  CALC AN APPROX VALUE FOR THE ALPHA ANGLE USING ST LINE SEG.  ALFA
C  WILL BE USED TO DETERMINE THE INCREMENTAL X AND Y STEP
C
      ALFA = ATAN2(ZINC,RHOXY)
      CA = COSF(ALFA)
      SA = SINF(ALFA)
      CAB = CA*COSF(BETA)
      SAB = CA*SINF(BETA)
C
C  SET UP SEGMENT PARAMETERS.  IF WIRE WEIGHT IS SPECIFIED. CATENARY WILL
C  BE CALCULATED.  FOR VERTICAL ELEMENTS NO CATENARY WILL BE USED.
C
      NEND = NS + I - 1
      XX1 = X1
      YY1 = Y1
      ZZ1 = Z1
      CAT  = WT*COLAM/(2.*TENS*0.3048)
      DO 100 M=I,NEND
      NEXP = NEXP + 1
      SL  = SLO*(1.+TAU)**NEXP  $$$ A TAPERED SEGMENT FOR NONZERO TAU
      SLX = SL*CAB              $$$ SEGMENT PROJECTIONS ALONG THE
      SLY = SL*SAB              $$$      X AND Y AXIS
      XX2 = XX1 + SLX
      YY2 = YY1 + SLY
      XPRIME = SQRTF((XX2-X1)**2 + (YY2-Y1)**2)    $$$ HORIZ DIST ON WIRE
      IF(ABSF(ALFA) .LE. 1.5) GO TO 111
      ZZ2 = ZZ1 + SL*SA
      GO TO 110                                    $$$   WITHIN 4 DEG OF VERT
  111 ZZ2 = Z2 - (RHOXY-XPRIME)*(CAT*XPRIME + ZINC/RHOXY)
  110 XY = SQRTF(SLX**2 + SLY**2)
      ALPHA = ATAN2((ZZ2 - ZZ1),XY)  $$$  CALC THE CORRECT ALPHA ANGLE
      X(M) = (XX1 + XX2)/2.
      Y(M) = (YY1 + YY2)/2.
      Z(M) = (ZZ1 + ZZ2)/2.
      SI(M) = SQRTF(XY**2 + (ZZ2-ZZ1)**2)
```

```
      BI(M) = WRAD
      ALP(M) = ALPHA
      BET(M) = BETA
      ICON1(M) = M - 1
      ICON2(M) = M + 1
      XX1 = XX2
      YY1 = YY2
      ZZ1 = ZZ2
  100 CONTINUE
      RETURN
      END

     CODE ANALYSIS
     COMPLEX MIXPATH,F,CSQRT,CEXP
     FUNCTION MIXPATH(W,EN,SN,EF,SF,PHI,RANGE,DLAND)
     PARAMETER (PI=3.141592653589793238462643,MU=12.56636E-7,E0=8.854E-12)

     MIXPATH=1.
     IF(PHI.GE.PI*.5 .AND. PHI.LE.PI*1.5) RETURN $$$ PATH ALL LAND
     PH=PHI
     IF(PHI.GT.PI*1.5) PH=2.*PI-PHI
     D=DLAND/COS(PH)
     IF(D/RANGE.GE.1) RETURN $$$ PATH IS STILL ALL LAND
     MIXPATH=(F(W,RANGE,EF,SF)*F(W,RANGE-D,EF,SF)*F(W,D,EN,SN))/
             (F(W,RANGE,EN,SN)*F(W,RANGE-D,EN,SN)*F(W,D,EF,SF))
     MIXPATH=CSQRT(MIXPATH)
C     DIMENSION -COMPLEX- FC(8) $ EQUIVALENCE (FC,FCC) $ DIMENSION FCC(16)
C     FC(1)=F(W,RANGE,EF,SF) $ FC(2)=F(W,RANGE-D,EF,SF)
C     FC(3)=F(W,D,EN,SN) $ FC(4)=F(W,RANGE,EN,SN)
C     FC(5)=F(W,RANGE-D,EN,SN) $ FC(6)=F(W,D,EF,SF)
C     FC(7)=(FC(1)*FC(2)*FC(3))/(FC(4)*FC(5)*FC(6))
C     FC(7)=CSQRT(FC(7))
C     FC(8)=CABS(MIXPATH)
C     WOT 3, (6E12.4) ,FCC,PHI,RANGE,DLAND
     RETURN
     END
```

```
      SUBROUTINE MOVE(XS,YS,ZS,ROX,ROY,ROZ)
      PARAMETER (NS = 300)
      COMMON/GEOM/N,NP,X(NS),Y(NS),Z(NS),SI(NS),BI(NS),ALP(NS),
     1 BET(NS),ICON1(NS),ICON2(NS),COLAM
      SPS=SINF(ROX)
      CPS=COSF(ROX)
      STH=SINF(ROY)
      CTH=COSF(ROY)
      SPH=SINF(ROZ)
      CPH=COSF(ROZ)
      XX=CPH*CTH
      XY= CPH*STH*SPS-SPH*CPS
      XZ= CPH*STH*CPS+SPH*SPS
      YX=SPH*CTH
      YY= SPH*STH*SPS+CPH*CPS
      YZ= SPH*STH*CPS-CPH*SPS
      ZX=-STH
      ZY=CTH*SPS
      ZZ=CTH*CPS
      DO 10 I=1,N
      XI=X(I)
      YI=Y(I)
      ZI=Z(I)
      X(I)=XI*XX+YI*XY+ZI*XZ+XS
      Y(I)=XI*YX+YI*YY+ZI*YZ+YS
      Z(I)=XI*ZX+YI*ZY+ZI*ZZ+ZS
      PXP=COSF(ALP(I))
      PX=PXP*COSF(BET(I))
      PY=PXP*SINF(BET(I))
      PZ=SINF(ALP(I))
      PXP=PX*XX+PY*XY+PZ*XZ
      PYP=PX*YX+PY*YY+PZ*YZ
      PZP=PX*ZX+PY*ZY+PZ*ZZ
      ALP(I)=ASINF(PZP)
      BET(I)=ATAN2(PYP,PXP)
   10 CONTINUE
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE NEWNAME
      ALL INTEGER
      COMMON /NEWCOM/ NNEW,NOLD
      DIMENSION BETA(3)
COMMENT----GET PRESENT NAME AND SUFFIX
      CALL FROST (24078,0,BETA,ERR1)
      SUFFIX=(BETA.SHL.6).INT.77B
      IF(SUFFIX.EQ.0) SUFFIX=1RF
      NOLO=BETA(2)
      DO D1 I=1,10
      IF(((NOLD.SHL.(I*6)).INT.77B).NE.0) GO TO G1
   D1 CONTINUE
      I=11
   G1 IF(I.LT.3) I=3
      NNEW=((NOLD.SHL.((I-1)*6).INT..COMP.7777B).UN.(2R* .UN.SUFFIX)).SHL.((11-I
     .)*6)
COMMENT----CREATE DROP FILE
      BETA(1)=NNEW
      BETA(2)=0
      CALL FROST(0101B,0001B,BETA,ERR2)
COMMENT----SET UP POINTER TO QUIT
      COMMON/GOBCOM/IGOB(42)
      EXTERNAL QUIT
      IGOB(4)=(.LOC.QUIT)+1
      RETURN
 ERR1 WOT 59, (37H**NEWNAME**ERROR GETTING PRESENT NAME)
      CALL EXIT(1)
 ERR2 WOT 59, (35H**NEWNAME**ERROR CREATING DROP FILE)
      CALL EXIT(1)
      END
```

```
      CODE ANALYSIS
      SUBROUTINE PLOTIT (XAXIS,YAXIS,NPTS,XTITLE,YTITLE,PTITLE)


COMMENT----LIST ARGS AND DEFINE

C     MED      0=NEITHER  1=TP3  2=DD80  3=BOTH
C     MXN            0=NONE  1=XONLY  2=YONLY  3=BOTH   MAXIMA AND MINIMA
C     PLT            0=SOLID LINE 1=DOT  LINE 2=PTS W/CHR 3=PTS W/CHR DOT
C                        4=PTS W/CHR SOLID
C      MAP           MAPPING USED IN PLOTS SEE MANUAL (9=MOST COMMON)
C     CHR            CHARACTER PLOTTED OVER POINTS (0=MOST COMMON)
C     DOT            NUMBER USED TO CREATE DOTTED LINE (3 IS BEST)
C     XRA, YRA       STARTING LOCS OF ARRAYS TO BE PLOTTED
C     IXRA, IYRA     INCREMENT BETWEEN VALUES IN ARRAYS
C     NPLT           NUMBER OF POINTS TO BE PLOTTED
C     XMIN,XMAX,YMIN,YMAX   MAX AND MIN VALUES OF ARRAYS
C      PTA,PTB       LOC OF PLOT TITLES (2 ALLOWED PER PLOT)
C     XTA, XTB       LOC OF X-AXIS TITLES (2)
C      YTA, YTB      LOC OF Y-AXIS TITLES (2)
C     FMT            FORMAT LISTS TO BE WRITTEN OUT
C                        FMT(1,N) = .LOC. OF FORMAT IN LIST N
C                        FMT(2,N)= .LOC. OF NO. OF VARIABLES IN LIST N
C                        FMT(3 TO 10, N)= .LOC. OF VARIABLES IN LIST N
C     NFMT           ORDER IN WHICH FORMAT LISTS ARE TO BE PROCESSED
C     NFMTS          HOW MANY LISTS TO BE PROCESSED


      COMMON/PLTCOM/-INTEGER-MED,MXN,PLT,FRM,MAP,CHR,DOT,
     .    IXRA, IYRA, XRA, YRA,
     * NPLT,PTA,XTA,YTA,PTB,XTB,YTB,FMT(10,5), NFMT(5),NFMTS,
     *      -REAL- XMIN,XMAX,YMIN,YMAX
      EQUIVALENCE (P1,FMT1), (P2,FMT2),(P3,FMT3),(P4,FMT4),(P5,FMT5)
      POINTER(PTA,PT1),(PTB,PT2),(XTA,XT1),(XTB,XT2),(YTA,YT1),(YTB,YT2)
      POINTER (XRA, XARRAY), (YRA, YARRAY)  $$C ALLOWS LOC OF ARRAYS TO BE PASSE
      POINTER (XRA,XLRRAY), (YRA, YLRRAY)
COMMENT----POINTER ALLOWS LOC OF ARRAYS TO BE PASSED
      ABSOLUTE ISLCM(168)
      DIMENSION XARRAY(1), YARRAY(1)
      LCM XLRRAY, YLRRAY
      DIMENSION XLRRAY(1), YLRRAY(1)


      XRA=.LOC.XAXIS $ YRA=.LOC.YAXIS $ NPLT=NPTS $XTA=.LOC.XTITLE
      YTA=.LOC.YTITLE $ PTA=.LOC.PTITLE
      ENTRY PLOT
COMMENT----FIND APPROPRIATE MAX AND MIN (MXN)

      IF (MED .EQ. 0) GO TO GRET
      IF (MXN .EQ. 0) GO TO GSKP
      IF (MXN .EQ. 2) GO TO GDOY
      IF ( XRA .GE. ISLCM) GO TO GXLCM
      XMIN=XMINN=XMAX=XARRAY(1)
      DO DXX, IA=1+IXRA, NPLT*IXRA, IXRA
      IF (XMAX .LT. XARRAY(IA)) XMAX=XARRAY(IA)
      IF (XMIN .GT. XARRAY(IA))XMIN=XARRAY(IA)
  DXX IF((XMINN.GT.XARRAY(IA) .AND. XARRAY(IA).GT.0) .OR. XMINN.LE.0)
     .  XMINN=XARRAY(IA)
      GO TO G88
GXLCM CONTINUE
      XMIN=XMINN=XMAX=XLRRAY(1)
      DO DXL,IA=1+IXRA, NPLT*IXRA, IXRA
      IF (XMAX .LT. XLRRAY(IA)) XMAX=XLRRAY(IA)
      IF (XMIN .GT. XLRRAY(IA))XMIN=XLRRAY(IA)
```

```
  DXL IF((XMINN.GT.XLRRAY(IA) .AND. XLRRAY(IA).GT.0) .OR. XMINN.LE.0)
     .   XMINN=XLRRAY(IA)
 G88 CONTINUE
      IF (MXN .EQ. 1) GO TO G89
 GDOY CONTINUE
      IF ( YRA .GE. ISLCM) GO TO GYLCM
      YMAX=YMINN=YMIN=YARRAY(1)
      DO DYY, IB=1+IYRA, NPLT*IYRA, IYRA
      IF (YMAX .LT. YARRAY(IB)) YMAX=YARRAY(IB)
      IF (YMIN .GT. YARRAY(IB)) YMIN=YARRAY(IB)
  DYY IF((YMINN.GT.YARRAY(IB) .AND. YARRAY(IB).GT.0) .OR. YMINN.LE.0)
     .   YMINN=YARRAY(IB)
      GO TO G89
GYLCM CONTINUE
      YMAX=YMINN=YMIN=YLRRAY(1)
      DO DYL,IB=1+IYRA, NPLT*IYRA, IYRA
      IF (YMAX .LT. YLRRAY(IB)) YMAX=YLRRAY(IB)
      IF (YMIN .GT. YLRRAY(IB)) YMIN=YLRRAY(IB)
  DYL IF((YMINN.GT.YLRRAY(IB) .AND. YLRRAY(IB).GT.0) .OR. YMINN.LE.0)
     .   YMINN=YLRRAY(IB)
 G89  CONTINUE

COMMENT----ATTEMPT TO MAKE MIN .NE. 0 FOR LOG MAPS
COMMENT----ADJUST MAX AND MIN SO PLOT DOES NOT RUN ON EDGE
      IF(MXN.EQ.2) GO TO GON1
      IF(MAP.EQ.2 .OR. MAP.EQ.4 .OR. MAP.EQ.6 .OR. MAP.EQ.8 .OR.
     .    MAP.EQ.10 .OR. MAP.EQ.12)  .GON1L
      XMIN=XMINN
      IF(XMAX.LE.0 .OR. XMIN.LE.0) GO TO GON1
      IF(XMAX.GT.1.) XMAX=XMAX**1.05
      IF(XMAX.EQ.1.) XMAX=1.05
      IF(XMAX.LT.1.) XMAX=XMAX**.95
      IF(XMIN.GT.1.) XMIN=XMIN**.95
      IF(XMIN.EQ.1.) XMIN=.95
      IF(XMIN.LT.1.) XMIN=XMIN**1.05
      GO TO GON1
  GON1L CONTINUE
      DX=.05*(XMAX-XMIN)
      IF(DX.EQ.0) DX=.05*ABSF(XMAX)
      IF(DX.EQ.0) DX=1.
      XMAX=XMAX+DX
      XMIN=XMIN-DX
  GON1 CONTINUE
      IF(MXN.EQ.1) GO TO GSKP
      IF(MAP.EQ.2 .OR. MAP.EQ.3 .OR. MAP.EQ.6 .OR. MAP.EQ.7 .OR.
     .    MAP.EQ.10 .OR. MAP.EQ.11)  .GON2L
      YMIN=YMINN
      IF(YMAX.LE.0 .OR. YMIN.LE.0) GO TO GON2
      IF(YMAX.GT.1.) YMAX=YMAX**1.05
      IF(YMAX.EQ.1.) YMAX=1.05
      IF(YMAX.LT.1.) YMAX=YMAX**.95

      IF(YMIN.GT.1.) YMIN=YMIN**.95
      IF(YMIN.EQ.1.) YMIN=.95
      IF(YMIN.LT.1.) YMIN=YMIN**1.05
      GO TO GON2
  GON2L CONTINUE
      DY=.05*(YMAX-YMIN)
      IF(DY.EQ.0) DY=.05*ABSF(YMAX)
      IF(DY.EQ.0) DY=1.
      YMAX=YMAX+DY
      YMIN=YMIN-DY
  GON2 CONTINUE
  GSKP CONTINUE
```

```
COMMENT----PLOT ON APPROPRIATE OUTPUT MEDIUM (MED)
       IF (MED .EQ. 2) GO TO GDD80
C      CALL PLTTP3 (XARRAY, YARRAY, NPLT, XMAX, XMIN, YMAX, YMIN,
C     *    PT1, XT1, YT1, PT2, XT2, YT2)
       IF (MED .EQ. 1) GO TO GTP3   $$C DO TP3 ONLY
COMMINT----WHEN PLOTTING ON DD80 USE RIGHT PLOTTING MODE (PLT) AND MAP (MAP)
 GDD80 CONTINUE
       IF(FRM),GNOF
       CALL FRAME
       INTEGER FRMNO
       FRMNO=FRMNO+1
       CALL SETCH(90.,64.,1,0,0,0)
       WOT 100, (11HPLOT NO. = I5) ,FRMNO
       INTEGER TIME,ALL,DAY,YEAR
       CALL CLOCK(TIME,IMD)
       YEAR=IMD.INT.7777B
       MT=((IMD.SHL.18).INT.77B)-20B  $$$ ASCII TO OCTAL
       MU=((IMD.SHL.24).INT.77B)-20B   $$$ ASCII TO OCTAL
       DIMENSION MONTH(12)
       DATA (MONTH=3RJAN,3RFEB,3RMAR,3RAPR,3RMAY,3RJUN,3RJUL,3RAUG,
      .   3RSEP,3ROCT,3RNOV,3RDEC)
       DAY=(IMD.SHL.42).INT.7777B
       MACH=(IMD.SHL.6) .INT. 77B
       ALL=(((DAY.SHL.24).UN.MONTH(10*MT+MU)).SHL.18).UN.YEAR
       WOT 100, (7HTIME = A10,R2,A10) ,TIME,MACH,ALL
       IF(MAP .LE. 12)
      .  CALL MAPX (MAP, XMIN, XMAX, YMIN, YMAX, .11328, .999, .25, .90)
       IF(MAP .GT. 12) CALL MAPP (XMAX,.11328,.90,.25)
  GNOF CONTINUE
       CALL DDERS(1)
       IF (PLT .LT. 2) GO TO GLIN    $$$ DO LINE PLOTS ONLY
       CALL SETPCH (1,0,1,0,20)
       CALL POINTC (CHR, XARRAY, YARRAY, NPLT, IXRA, IYRA)
       IF (PLT .EQ. 2) GO TO GPTS  $$$ DO PTS W/CHR ONLY
 GLIN CONTINUE
       IF (PLT .EQ. 1  .OR.  PLT .EQ. 3)
      *    CALL TRACEP (XARRAY, YARRAY, NPLT, DOT, IXRA, IYRA)
       IF (PLT .EQ. 0  .OR.  PLT .EQ. 4)
      *    CALL TRACE (XARRAY, YARRAY, NPLT, IXRA, IYRA)
 GPTS CONTINUE
COMMENT----NOW TO WRITE THE TITLES
       IF(FRM) ,GRET
       INTEGER OLDXTA,OLDYTA,OLDPTA,OLDXTB,OLDYTB,OLDPTB
       IF(.NOT.(XT1.XOR.  . )) XTA=OLDXTA
       IF(.NOT.(YT1.XOR.  . )) YTA=OLDYTA
       IF(.NOT.(PT1.XOR.  . )) PTA=OLDPTA
       IF(.NOT.(XT2.XOR.  . )) XTB=OLDXTB
       IF(.NOT.(YT2.XOR.  . )) YTB=OLDYTB
       IF(.NOT.(PT2.XOR.  . )) PTB=OLDPTB
       OLDXTA=XTA
       OLDYTA=YTA
       OLDPTA=PTA
       OLDXTB=XTB
       OLDYTB=YTB
       OLDPTB=PTB
```

```
      CALL SETCH (10.,8.,1,0,1,0)  $ CALL CRTBCD (XT1)
      CALL SETCH (10.,7.,1,0,1,0)   $ CALL CRTBCD(XT2)
      CALL SETCH (1.,20.,1,0,1,1)   $ CALL CRTBCD (YT1)
      CALL SETCH (2.,20.,1,0,1,1)   $ CALL CRTBCD (YT2)
      CALL SETCH (10.,32.,1,0,2,0)    $ CALL CRTBCD (PT1)
      CALL SETCH (10.,30.5,1,0,2,0)    $ CALL CRTBCD (PT2)


COMMENT----POSITION BEAM SO WOT 100 STATEMENTS MAY BE DONE
      CALL SETCH (1., 5., 1,0,0,0)


COMMENT----WRITE FORMAT LISTS OUT

GTP3  CONTINUE
      DO DWOT, N=3, 100, 97
      IF (MED .EQ. 2) N=100    $$C JUST DO TAPE 100
COMMENT----SET UP ABSOLUTE ADDRESSING
      ABSOLUTE A(1)  $ DIMENSION -INTEGER- A(1)
      EQUIVALENCE (L,NFMT)
      DIMENSION L(1)
      DO OFMT, K=1,NFMTS
      M=FMT(1,L(K))
 OFMT WOT N, A(M)            , ( A(FMT(IZ,L(K))),IZ=3,2+A(FMT(2,L(K))) )
      IF (MED .EQ. 1) N=100  $$C JUST DO TP3 (CAUSE LOOP TO TERMINATE)
 DWOT CONTINUE
GRET CONTINUE
      RETURN
      END
```

CODE ANALYSIS

```
      SUBROUTINE SOLVE(N,P,B,NDIM)
C
C SUBROUTINE TO SOLVE THE MATRIX EQUATION LU*X=B WHERE L IS A UNIT LOWER
C TRIANGULAR MATRIX AND U IS AN UPPER TRIANGULAR MATRIX BOTH OF WHICH ARE STORED
C IN A.  THE RHS VECTOR B IS INPUT AND THE SOLUTION IS RETURNED THROUGH VECTOR B
C
      LCM (333)
      PARAMETER (NS = 300)
      COMPLEX CM,B,Y,SUM
      INTEGER P,PI
      COMMON /333/ CM(1)
      DIMENSION P(NDIM),B(NDIM)
      COMMON /SCRATM/ Y(NS)
C
C FORWARD SUBSTITUTION
C
      DO 20 I=1,N
      PI=P(I)
      Y(I)=B(PI)
      B(PI)=B(I)
      IP1=I+1
      IF(IP1.GT.N) GO TO 11
      DO 10 J=IP1,N
      B(J)=B(J)-CM(J+N*(I-1))*Y(I)
   10 CONTINUE
   11 CONTINUE
   20 CONTINUE
C
C BACKWARD SUBSTITUTION
C
      DO 40 K=1,N
      I=N-K+1
      SUM=CMPLX(0.,0.)
      IP1=I+1
      IF(IP1.GT.N) GO TO 31
      DO 30 J=IP1,N
      SUM=SUM+CM(I+N*(J-1))*B(J)
   30 CONTINUE
   31 CONTINUE
      B(I)=(Y(I)-SUM)/CM(I+N*(I-1))
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SURF
      CODE ANALYSIS
      COMMON/COMCOM/COM(8)

      DIMENSION-INTEGER-TITLE(10)
      DATA(TITLE(10)=778)
      COMMON /PLTCOM/ -INTEGER-MED,MXN,PLT,FRM,MAP,CHR,DOT,
     .  IXRA,IYRA,XRA,YRA,NPLT,PTA,XTA,YTA,PTB,XTB,YTB,
     .  FMT(10,5),NFMT(5),NFMTS,-REAL- XMIN,XMAX,YMIN,YMAX
      DATA (I77=778),(PTA=I77,I77,I77,TITLE,I77,I77)
      DATA (MED=2,3,0,1,11,1H0,3,1,1)
 F100 FORMAT( MAXIMUM FIELD STRENGTH =     E15.5)
 F200 FORMAT( POWER IN SPACE WAVE(THETA COMPONENT, PHI COMP., TOTAL)= ,3E13.5)
 F250 FORMAT ( ERN=  E12.4,     SIGN=  E12.4,     ERF=  E12.4,     SIGF=  E12.4 )
 F300 FORMAT(8A10)
 F400 FORMAT( DLAND= ,E12.4,     RANGE= ,E12.4)
      DATA(FMT=F100,.LOC.1,XXMAX)
      DATA(FMT(11)=F200,.LOC.3,POWERT,POWERP,POWER)
      DATA(FMT(21)=F250,.LOC.4,ERN,SIGN,ERF,SIGF)
      DATA(FMT(31)=F300,.LOC.8,COM,COM(2),COM(3),COM(4),COM(5),COM(6),COM(7),COM
     .  (8))
      DATA(FMT(41)=F400,.LOC.2,DLAND,RANGE)
      DATA(NFMT=4,3,5,2,1,5)

      COMPLEX MIXPATH
      COMPLEX ETV,ETH,EPH,SEPH,SETV,SETH,CURMOM
      DATA (DLAND=200.)
      COMPLEX CSQRT,CEXP
      PARAMETER (PI=3.14159265358979323846264 3)
      PARAMETER (DTR=PI/180.,RTO=180./PI)
C     DATA (RANGE=1.E5),(NPHI=72),(PHIMAX=2.*PI),(PHIMIN=0)
C     DATA (NTHET=1),(THETMAX=PI/6.),(THETMIN=PI/6.)
      DIMENSION ANG(360),VERT(360),HORZ(360)
      IFIRST=0

 GMORE MORE=0

      NAMELIST /LIST/ INFILE,NPHI,PHIMAX,PHIMIN,NTHET,THETMAX,THETMIN,
     .  VMIN,VMAX,HMIN,HMAX,ERN,SIGN,ERF,SIGF,RANGE,MORE,TITLE,OLDVERSION
     .  ,DLAND,NPHIP,NTHETP

      INPUT DATA LIST 2,3

COMMENT----READ COMMENT FILE BY FRED.
      DIMENSION ICOMM(8)
      IF(IFIRST .EQ. 1) GO TO G100
      IFIRST=1
C       CALL ANTPLOT
C       CALL FRAME
C       INFLE=INFILE
C       CALL SETCH(0,100.,1,0,1,0,0)
C       CALL ASSIGN(6,INFLE)
C       CALL DEVICE( OPEN ,INFLE,LNTH,IERR)
C       IF(IERR.EQ.0 .OR. IERR.EQ.2) GO TO REED
C       WOT 59,FBAD,INFLE,IERR
C       WOT 100,FBAD,INFLE,IERR
C   FBAD FORMAT( COULD NOT OPEN COMMENT FILE  A10,  S-FIELD IS  I3)
C       GO TO G100
C   REED RIT 6, (8A10) ,(ICOMM(I),I=1,8)
C       IF(EOF,6) G100,
C       WOT 100, (1X,8A10) ,(ICOMM(I),I=1,8)
C       GO TO REED
  G100 CONTINUE
C       CALL DEVICE( CLOSER ,INFLE)
```

```
      FRM=1
      INFLE=INFILE
      PARAMETER (NS=300)
      COMMON/ABC/    AIR(NS),AII(NS),BIR(NS),BII(NS),CIR(NS),CII(NS)
      COMMON /GEOM/ N,NP,X(NS),Y(NS),Z(NS),SI(NS),BI(NS),ALP(NS),
     * BET(NS),ICON1(NS),ICON2(NS),COLAM
      COMMON /ANGL/ CAB(NS),SAB(NS),SALP(NS)
      COMMON /MEDIA/ EPSRP,SIGP,EPSRM,SIGM

      FREQ=-2.997963E+8/COLAM
      ERN=EPSRM $ SIGN=SIGM

COMMENT----DO INTEGRAL TO FIND POWER IN SPACE WAVE

      IF(NPHIP.LE.0 .OR. NTHETP.LE.0) GO TO GNPWR
      SUMT=0 $ SUMP=0 $ DTHET=.5*PI/NTHETP
      DO DP1 IP1=1,NTHETP
      THET=(IP1-.5)*.5*PI/NTHETP
      STHET=SIN(THET)
      NNPHI=NPHIP*STHET
      IF(NNPHI.LT.1) NNPHI=1
      DPHI=2.*PI/NNPHI
      DO DP1 IP2=1,NNPHI
      PHI=(IP2-1)*2.*PI/NNPHI
      ETV=ETH=EPH=0.
      SETV=SETH=SEPH=0.
      DO DP3 K=1,N $$$ NO. OF SEGS IN ANTENNA
      CURMOM=CMPLX(AIR(K)+CIR(K),AII(K)+CII(K))*SI(K)
      CALL FIELDS(ETV,ETH,EPH,RANGE,PHI,THET,
     .  X(K),Y(K),Z(K),
     .   ALP(K),BET(K),CURMOM,FREQ,EPSRM,SIGM)
      SETV=SETV+ETV $ SETH=SETH+ETH  $ SEPH=SEPH+EPH
 DP3 CONTINUE
      ESQT=(SETV+SETH)*CONJG(SETV+SETH)
      ESQP=SEPH*CONJG(SEPH)
      ESQT=ESQT*DPHI*STHET*DTHET
      ESQP=ESQP*DPHI*STHET*DTHET
      SUMT=SUMT+ESQT
  DP1 SUMP=SUMP+ESQP
      POWERT=SUMT*.5*RANGE**2/377.
      POWERP=SUMP*.5*RANGE**2/377.
      POWER=POWERT+POWERP
      WOT 3, (//20HPOWERT,POWERP,POWER= ,3E15.5) ,POWERT,POWERP,POWER
  GNPWR CONTINUE

      NANG=NPHI
      IF(NTHET.GT.1) NANG=NTHET
      DO D2 I=1,NANG
      PHI=DTR*PHIMIN $ THET=DTR*THETMIN
      IF(NPHI.GT.1 .AND. NANG.GT.1)
     .   PHI=DTR*(PHIMIN+((PHIMAX-PHIMIN)*(I-1))/(NANG-1))
      IF(NTHET.GT.1 .AND. NANG.GT.1)
     .   THET=DTR*(THETMIN+((THETMAX-THETMIN)*(I-1))/(NANG-1))
      ANG(I)=PHI
      IF(NTHET .GT. 1) ANG(I)=PI/2. - THET
      IF(THET.LT.0) PHI=PHI+PI
      IF(THET.LT.0) THET=-THET
      ETV=ETH=EPH=0.
      SETV=SETH=SEPH=0.
      DO D3 K=1,N $$$ NO. OF SEGS IN ANTENNA
      CURMOM=CMPLX(AIR(K)+CIR(K),AII(K)+CII(K))*SI(K)
      CALL FIELDS(ETV,ETH,EPH,RANGE,PHI,THET,
     .  X(K),Y(K),Z(K),
```

```
          ALP(K),BET(K),CURMOM,FREQ,EPSRM,SIGM)
       SETV=SETV+ETV $ SETH=SETH+ETH  $ SEPH=SEPH+EPH
 D3 CONTINUE
       VERT(I)=CABS(SETV+SETH)
       HORZ(I)=CABS(SEPH)
       IF(THET .GE. 3.1415/2.) VERT(I)=VERT(I)*CABS(MIXPATH(2.*PI*FREQ,
     .  EPSRM,SIGM,ERF,SIGF,PHI,RANGE,DLAND))
  D2 CONTINUE
       XMIN=VMIN   $   XMAX=VMAX
       MXN=2
       IF(VMAX.EQ.0 .AND. VMIN.EQ.0) MXN=3
       MAP=13
       XXMAX=AMAXAF(VERT,1,NANG,1,IDUMMY)
        CALL PLOTIT(VERT,ANG,NANG,778,778, THETA COMPONENT )
       WOT 3,FMTV,XXMAX
 FMTV  FORMAT(  *THETA*  MAX FLD STRENGTH OF THETA COMPONENT =  ,
     .  E15.5)
       XXMAX=AMAXAF(HORZ,1,NANG,1,IDUMMY)
       WOT 3,FMTH,XXMAX
 FMTH  FORMAT(  *PHI*    MAX FLD STRENGTH OF PHI COMPONENT =    ,
     .  E15.5)
       IF(NTHET.LE.1) GO TO GOBYY
       XMIN=HMIN  $  XMAX=HMAX
       MXN=2
       IF(HMIN .EQ. 0 .AND. HMAX .EQ. 0) MXN=3
       CALL PLOTIT(HORZ,ANG,NANG,778,778, PHI COMPONENT )
       FRM=1
  GOBYY CONTINUE
COMMENT----PLOT RUNNING AVERAGE OF EFIELD SQUARED
       IF(NTHET.GT.1) GO TO GOBY
       DIMENSION RINT(360),DANG(360)
       RINT(1)=0
       DANG(1)=ANG(1)*180./PI
       DO DM1 IM1=2,NANG
       IF(ANG(IM1-1).GT.PI*.5) GO TO GM1
       DANG(IM1)=180.*ANG(IM1)/PI
       ESQUAREAVG=.5*(VERT(IM1-1)**2+VERT(IM1)**2)
  DM1 RINT(IM1)=RINT(IM1-1)+ESQUAREAVG*(ANG(IM1)-ANG(IM1-1))
  GM1 NNANG=IM1-1
       RINT(1)=VERT(1)**2
       DO DM2 IM2=2,NNANG
  DM2 RINT(IM2)=RINT(IM2)/ANG(IM2)
       MAP=9 $ MXN=3
       CALL PLOTIT(DANG,RINT,NNANG, PHI , AVERAGE , RUNNING AVERAGE--E SQUARED )
  GOBY CONTINUE
  GFRED CONTINUE
  D1 CONTINUE
       IF(MORE) GO TO GMORE
       DIMENSION CURMAG(NS),CURPHAZ(NS),SEGNUM(NS)
       DO D4 I=1,N
       SEGNUM(I)=I
       CURMAG(I)=SQRT((AIR(I)+CIR(I))**2 + (AII(I)+CII(I))**2)
       CURPHAZ(I)=RTD*ATAN2((AII(I)+CII(I)),(AIR(I)+CIR(I)))
 D4    CONTINUE
       MXN=3   $    MAP=9
       TITLE=778
       CALL PLOTIT(SEGNUM,CURMAG,N, SEGMENT NUMBER , CURRENT ,
     .    MAGNITUDE )
       CALL PLOTIT(SEGNUM,CURPHAZ,N, SEGMENT NUMBER , CURRENT ,
     .    PHASE )
       RETURN
       END
```

```
      CODE ANALYSIS
       FUNCTION TEST(F1,F2)
      IF(ABSF(F2)-1.0E-40)2,2,1
    1 TEST=ABSF((F1-F2)/F2)
       RETURN
    2  TEST=0.
       RETURN
       END
```

```
        SUBROUTINE TRIO(J,JCO1,JCO2,DIL,DIK)
       PARAMETER (NS = 300)
        COMMON/GEOM/N,NP,X(NS),Y(NS),Z(NS),SI(NS),BI(NS),ALP(NS),
      1  BET(NS),ICON1(NS),ICON2(NS),COLAM
        COMMON/JUNK/NCOX,JOX(25),NCIX,JIX(25),NCOZ,JOZ(25),NCIZ,JIZ(25)
        S=SI(J)
        JCO1=ICON1(J)
        JCO2=ICON2(J)
        IF(JCO1)1,2,3
    1   CALL JUNC(J,JCO1,NCOX,JOX,NCIX,JIX,DIL)
        GO TO 4
    2   DIL=S/2.0
        GO TO 4
    3   DIL=(SI(JCO1)+S)/2.0
    4   IF(JCO2) 5,6,7
    5   CALL JUNC(J,JCO2,NCOZ,JOZ,NCIZ,JIZ,DIK)
        GO TO 8
    6   DIK=S/2.0
        GO TO 8
    7   DIK=(SI(JCO2)+S)/2.0
    8   CONTINUE
        RETURN
        END
```

```
      SUBROUTINE XFRM (ISEG,NSEG,LASTSEG)
      PARAMETER (NS = 300)
       COMMON/GEOM/N,NP,X(NS),Y(NS),Z(NS),SI(NS),BI(NS),ALP(NS),
     1  BET(NS),ICON1(NS),ICON2(NS),COLAM
C
C  THIS SUBROUTINE TAKES ADVANTAGE OF SYMMETRY TO ASSIGN VALUES TO
C  THE /DATA/ COMMON BLOCK --- DATA FROM THE FIRST OCTANT IS CALCULATED
C  FOR THE OTHER THREE OCTANTS
C
      PI = 3.141592654
      PIO2 = PI/2.
      NSEG2 = ISEG + NSEG - 1
      INDEX = NSEG2
      DO 100 J=2,4
      PIM1  = PIO2 * (J-1)
      SINN  = SINF(PIO2*J)
      COSN  = COSF(PIO2*J)
      SINM1 = SINF(PIM1)
      COSM1 = COSF(PIM1)
      NEND = (NSEG2 - ISEG + 1)*(J - 1)
      DO 100 M = ISEG,NSEG2
      INDEX = INDEX + 1
      X(INDEX) = X(M)* SINN  +  Y(M) * COSN
      Y(INDEX) = X(M)* SINM1 +  Y(M) * COSM1
      Z(INDEX) = Z(M)
      SI(INDEX) = SI(M)
      BI(INDEX) = BI(M)
      ALP(INDEX) = ALP(M)
      BET(INDEX) = BET(M) + PIM1
      ICON1(INDEX) = ICON1(M) + NEND
      ICON2(INDEX) = ICON2(M) + NEND
  100 CONTINUE
      LASTSEG = INDEX + 1
      RETURN
      END
```

Appendix E – Sommerfeld/Norton Subroutines Package

```
      CODE ANALYSIS
       OPTIMIZE
      SUBROUTINE BESSEL(Z,AJ0,BOP)
COMMENT----VERSION OF 26 FEB 74...OPTIMIZED FOR 7600....D LAGER

COMMENT----ASYMPTOTIC EXPANSION IS INVALID FOR Z A NEGATIVE REAL NO.
COMMENT----THE SYMMETRY RELATION J0(-X)=-J0(X) CAN BE USED INSTEAD.
COMMENT----HOWEVER THIS WAS NOT DONE HERE SINCE WE DON'T FEEL
COMMENT----THIS SITUATION WILL OCCUR.......

      PARAMETER (PI = 3.1415926535897932)
      COMPLEX BOP
       COMPLEX Z,Z2,Z3,Z4,Z5,P0Z,P1Z,Q0Z,Q1Z,AJ0,AJ1,ZEX,UM0,UM1,
     1CHI0,CHI1,CX0,CX1,SX0,SX1,S
       COMPLEX ZP,ZQ,ZR,ZS
      DIMENSION -REAL-K11(30),K12M(30)
      DIMENSION M(1,101)
      DIMENSION FAC((0,30))
      IF(IPASSED).GFIRST
   G1 CONTINUE
      COMPLEX ZK
      IF(REAL(Z)**2+AIMAG(Z)**2).RET $$$ IF(CABS(Z).EQ.0) RETURN
      IF((REAL(Z)**2+AIMAG(Z)**2)-100.).,LARGE $$$ IF(CABS(Z).GT.10.) GO TO LARG
      IZ=1.+(REAL(Z)**2+AIMAG(Z)**2)
      MIZ=M(IZ)
      AB=1. $$$ ACCUMULATES 1.**K
      AJ0=1. $$$ ZEROTH TERM OF SERIES
      AJ1=1. $$$ ZEROTH TERM OF SERIES
      ZK=1. $$$ ACCUMULATES (Z**(2*K))/(K FACTORIAL)
      ZEX=CMPLX(.25*REAL(Z*Z),.25*AIMAG(Z*Z))
      DO D11 K=1,MIZ
      ZK=ZK*CMPLX(REAL(ZEX)*K12M(K),AIMAG(ZEX)*K12M(K))
      AJ0=AJ0+CMPLX(REAL(ZK),AIMAG(ZK))
  D11 AJ1=AJ1+CMPLX(REAL(ZK)*K11(K),AIMAG(ZK)*K11(K))
      BOP=CMPLX(-.5*REAL(Z*AJ1),-.5*AIMAG(Z*AJ1))
      RETURN
  RET AJ0=1.
      BOP=0.
      RETURN
LARGE CONTINUE
      Z2=Z*Z
       Z3 = Z2*Z
       Z4 = Z3*Z
       Z5 = Z4*Z
       P0Z = 1. - P10/Z2 + P20/Z4
       Q0Z = -Q10/Z + Q20/Z3
       P1Z = 1. + P11/Z2 - P21/Z4
       Q1Z = Q11/Z - Q21/Z3
       CHI0 = Z - PI/4.
       CHI1 = CHI0 - PI/2.
       CHI0 = CHI0*S
        CHI1 = CHI1*S
      ZP=CEXP(CHI0)
      ZQ=CEXP(-CHI0)
       CX0 = (ZP + ZQ)/2.
       SX0 =(ZP - ZQ)/(2.*S)
      ZR=CEXP(CHI1)
      ZS=CEXP(-CHI1)
       CX1 = (ZR + ZS)/2.
       SX1 = (ZR - ZS)/(2.*S)
       AJ0 = P0Z*CX0 - Q0Z*SX0
       AJ1 = P1Z*CX1 - Q1Z*SX1
      ZR=CSQRT(Z)
       ZS = QWW/ZR
```

```
        AJ0 = AJ0*ZS
        AJ1 = AJ1*ZS
      B0P=-AJ1
      RETURN
 GFIRST CONTINUE
      IPASSED=1

COMMENT----INITIALIZE PARAMETERS FOR BESSEL FUNCTION

      S=CMPLX(0.,1.)
        QWW = SQRTF(2./PI)
        FAC(0)=1.
        DO 8 K = 1,25
        X = K
      KI1(K)=1./(K+1)
COMMENT----THE - SIGN BECOMES (-1)**N FOR POWER SERIES EXPANSION
      KI2M(K)=-(1./K)**2
   8    FAC(K) = FAC(K-1)*X
COMMENT----M ARRAY DETERMINES HO MANY TERMS TO USE IN POWER SERIES EXPANSION
      DO D44 I=1,101
      DO D43 N=1,24
      IF((I*.25)**N .LT. 1.E-5*FAC(N)**2) GO TO D44
   D43 CONTINUE
   D44 M(I)=N
C       WOT 59, (20I3) ,(M(I),I=1,101)
        P10 = 9./128.
        P20 = 9.*25.*49./(64.*64.*24.)
        Q10 = 1./8.
        Q20 = 9.*25./(48.*64.)
        P11 = 15./128.
        P21 = 15.*21.*45./(64.*64.*24.)
        Q11 = 3./8.
        Q21 = 15.*21./(48.*64.)

       GO TO G1
       END
```

```
COMMENT----11 JUL 74 ...ADDED INTERRUPT FEATURE

COMMENT----27 JUN 74....MODIFIED TO HANDLE ARBITRARY NUMBER OF NON-TOEPLITZ
COMMENT----   SEGMENTS.  THEY MUST BE THE FIRST FEW SEGMENTS OF THE STRUCTURE

      CODE ANALYSIS
      SUBROUTINE CMSETUP(ZRATIP,ZRATIM,KSYMP,IPGND,IVERTRC)
      LCM (333)
      PARAMETER (NS = 300)
       COMMON/GEOM/N,NP,X(NS),Y(NS),Z(NS),SI(NS),BI(NS),ALP(NS),
     1  BET(NS),ICON1(NS),ICON2(NS),COLAM
       INTEGER P
       COMPLEX ZRATI,REFS,REFPS,ZRSIN,ZRATIP,ZRATIM
       COMPLEX FJ,CM,EINC,CSQRT
      COMMON /333/ CM(1)
      COMMON /ANGL/ CAB(NS),SAB(NS),SALP(NS)
       COMMON/JUNK/NCOX,JOX(25),NCIX,JIX(25),NCOZ,JOZ(25),NCIZ,JIZ(25)
       COMMON/REFL/RHOX,RHOY,RHOZ,CABJ,SABJ,SALPR,PX,PY,REFS,REFPS
       DIMENSION ETR(3),ETI(3)
      COMMON /RCOM/ -INTEGER- NEVALSR,TROUBLER
      COMMON /ROMCOM/ -INTEGER- NEVALS,TROUBLE




COMMENT----READ IN NUMBER OF NON-TOEPLITZ SEGMENTS.  ASSUMMED TO BE
COMMENT----   THE FIRST FEW SEGMENTS OF THE STRUCTURE

      RIT 2, (A7,2I10) ,ITOEP,NONTOEP,ITIMING
      IF(ITOEP.XOR. NONTOEP ) ,GOK
FTOEP FORMAT( **CMSETUP** ,1X,A7,1X, IS NOT WORD NONTOEP.  FIX IT. )
      WOT 59,FTOEP,ITOEP
      WOT 3,FTOEP,ITOEP
      WOT 100,FTOEP,ITOEP
      CALL EXIT

  GOK CONTINUE
FTOEP2 FORMAT( **CMSETUP**NONTOEP= I10,10X, ITIMING= ,I10)
      WOT 59,FTOEP2,NONTOEP,ITIMING
      WOT 3,FTOEP2,NONTOEP,ITIMING
      WOT 100,FTOEP2,NONTOEP,ITIMING



      IF(ITIMING) CALL BEGINMAP



COMMENT----ENABLE CTRL-E I INTERRUPT FEATURE
COMMENT----   WHENEVER CTRL-E I IS TYPED AT TTY THE PROGRAM WILL
COMMENT---   PRINT OUT THE VALUES OF THE LOOP COUNTERS AND TIME
COMMENT----   USED SO FAR.

      LOCIN=.LOC.INTERRUPT
      CALL FROST(33B,01B,0,0,LOCIN,0,ERRINT1)
ERRINT1 CONTINUE $$$ IGNORE ERRORS




      NEVALS=0
      NEVALSR=0
      FJ=CMPLX(0.,1.)
      PI = 3.14159265
```

```
          SIGN=-1.
          DO 1 I=1,N
          DO 1 J=1,N
    1 CM(I+N*(J-1))=CMPLX(0.,0.)
C
C   J INDEX IS THE SOURCE LOOP
C
         LIMJL=NONTOEP+1 $ LIMJU=NONTOEP+1 $$$ SET LIMITS ON DO 3000
         LIMIL=NONTOEP+1 $ LIMIU=NP $$$ SET LIMITS ON DO 3000

         DO D10 I10=1,2

          DO 3000 J=LIMJL,LIMJU

          CALL TRIO(J,JCO1,JCO2,DIL,DIK)
          S=SI(J)
          B=BI(J)
          XJ=X(J)
          YJ=Y(J)
          ZJ=Z(J)
         MEDSOR = +1
         IF(ZJ .LT. 0.) MEDSOR = -1
         MEDIA = MEDSOR
         IF(MEDSOR .GT. 0) ZRATI=ZRATIP
         IF(MEDSOR .LT. 0) ZRATI=ZRATIM
          CABJ=CAB(J)
          SABJ=SAB(J)
          SALPJ=SALP(J)

C
C   I INDEX IS THE OBSERVATION LOOP
C
          DO 3000 I=LIMIL,LIMIU

COMMENT----FOR I10=2 CALCULATE ONLY NON-TOEPLITZ ROWS AND COLUMNS
         IF(I10.EQ.2 .AND. I.GT.NONTOEP .AND. J.GT.NONTOEP) GO TO 3000

          XIJ=X(I)-XJ
          YIJ=Y(I)-YJ
         MEDOBS = +1
         IF(Z(I) .LT. 0.) MEDOBS = -1
          IJ=I-J
          CABI=CAB(I)
          SABI=SAB(I)
          SALPI=SALP(I)
          RFL=-1.
         IF (MEDSOR*MEDOBS .LT. 0 .AND. KSYMP.NE.1) GO TO 740
         IF(KSYMP.NE.3) GO TO GRCOEF $$$ KSYMP=3 MEANS DO SOMMERFELD CALC.
C         IF(IJ.EQ.0) GO TO GRCOEF $$$ USE REF COEF FOR SELF TERM
C         WOT 59, (2I10) ,I,J
C         WOT 3, (2I10) ,I,J
         COMMON /SFCOM/ XS,YS,ZS,AS,BS,SS,WR,XO,YO,ZO,AO,BO,IMUTUAL
         XS=XJ
         YS=YJ
         ZS=ZJ
         AS=ALP(J)
         BS=BET(J)
         SS=SI(J)
         WR=B
          IMUTUAL=IJ
         XO=X(I)
         YO=Y(I)
         ZO=Z(I)
         AO=ALP(I)
```

```
      BO=BET(I)
      FACTOR=1.
      ALIM=-1.
      CTEST=1.E-2
      IF(IMUTUAL.NE.0) GO TO GMUTUAL
COMMENT----FOR SELF TERM DO TWICE INTEGRAL FROM 0 TO L/2
C      FACTOR=2.
C      ALIM=0
      CTEST=1.E-4  $$$ CONVERGENCE CRITERION
GMUTUAL CONTINUE

      COMMON /FREQ/ FREQ,-COMPLEX-WKP,WKM,ECONST
      WVLENGTH=3.E8/FREQ
      SEPARATION=SQRT((XS-XO)**2 + (YS-YO)**2 + (ZS-ZO)**2)
      IF(SEPARATION.GT.WVLENGTH) GO TO GNORT  $$$ USE NORTONS FORMULAS

C      WOT 59, (2I10) ,I,J

      EXTERNAL SFIELDS,GFIELDS
      DIMENSION -COMPLEX-ETANG(3),ET(3)
      COMPLEX ETK,ETS,ETC,CL,CK,SINL,COSL,SINK,COSK,SILK,CONS

      CALL ROM(ALIM,1.,GFIELDS,3,ETANG,CTEST)
      ETK=ETANG(1)*FACTOR
      ETS=ETANG(2)*FACTOR
      ETC=ETANG(3)*FACTOR

      CALL ROM(ALIM,1.,SFIELDS,3,ETANG,CTEST)
      ETK=ETK+ETANG(1)*FACTOR
      ETS=ETS+ETANG(2)*FACTOR
      ETC=ETC+ETANG(3)*FACTOR

      GO TO GINTERP

GNORT CONTINUE
      EXTERNAL NFIELDS

      CALL ROM(ALIM,1.,NFIELDS,3,ETANG,CTEST)
      ETK=ETANG(1)*FACTOR
      ETS=ETANG(2)*FACTOR
      ETC=ETANG(3)*FACTOR

GINTERP CONTINUE

C      WOT 3, (2I10,4E15.5) ,I,J,REAL(ETK),AIMAG(ETK),CABS(ETK),(180./PI)
C         *CANG(ETK)

      DIMENSION -COMPLEX-ES(NS),EC(NS),EK(NS)
      IF(I10.EQ.2) GO TO G43
      ES(I)=ETS
      EC(I)=ETC
      EK(I)=ETK
  G43 CONTINUE
COMMENT----DO INTERPOLATION
      IF(MEDIA .EQ. +1) CL=WKP*DIL
      IF(MEDIA .EQ. -1) CL=WKM*DIL
      IF(MEDIA .EQ. +1) CK=WKP*DIK
```

```
      IF(MEDIA .EQ. -1) CK=WKM*DIK
      CALL CSINCOS(CL,SINL,COSL)
      CALL CSINCOS(CK,SINK,COSK)
      SILK = SINL*COSK + SINK*COSL
      CONS = 1./(SINL + SINK - SILK)
      ET(1) = (SINK*ETK + (COSK-CMPLX(1.,0.))*ETS - SINK*ETC)*CONS
      ET(2) = (-SILK*ETK + (COSL-COSK)*ETS + (SINL+SINK)*ETC)*CONS
      ET(3) = (SINL*ETK + (CMPLX(1.,0.)-COSL)*ETS - SINL*ETC)*CONS
      CM(I+N*(J-1))=CM(I+N*(J-1))+ET(2)
      IF(JCO1.LT.0) CALL JMELS(REAL(ET(1)),AIMAG(ET(1)),NCIX,JIX,NCOX,JOX,I)
      IF(JCO1.GT.0) CM(I+N*(JCO1-1))=CM(I+N*(JCO1-1))+ET(1)
      IF(JCO2.LT.0) CALL JMELS(REAL(ET(3)),AIMAG(ET(3)),NCOZ,JOZ,NCIZ,JIZ,I)
      IF(JCO2.GT.0) CM(I+N*(JCO2-1))=CM(I+N*(JCO2-1))+ET(3)
      GO TO 610
GRCOEF CONTINUE $$$ USE REFLECTION COEFF APPROACH
      KLOOP=KSYMP
      IF(KSYMP.GT.2) KLOOP=2
       DO 300 IP=1,KLOOP
       RFL=-RFL
       ZIJ=Z(I)-RFL*ZJ
       Q1=CABI*CABJ+SABI*SABJ+SALPI*SALPJ*RFL
       Q2=XIJ*CABI+YIJ*SABI+ZIJ*SALPI
       ZP=XIJ*CABJ+YIJ*SABJ+ZIJ*SALPJ*RFL
       RS=XIJ*XIJ+YIJ*YIJ+ZIJ*ZIJ
       RH2=RS-ZP*ZP
       IF(RH2.LT.1.E-20) GO TO 601
      RH=SQRTF(RH2)
       QP2=(Q2-ZP*Q1)/RH
       GO TO 602
 601   QP2=0.
       RH=0.
 602  CONTINUE
      IF(IP.NE.2) GO TO 700
       SALPR=SALPJ*RFL
       RHOX=XIJ-CABJ*ZP
       RHOY=YIJ-SABJ*ZP
       RHOZ=ZIJ-SALPJ*ZP*RFL
      RMAG=SQRTF(RHOX*RHOX+RHOY*RHOY+RHOZ*RHOZ)
       IF(RMAG.GT.1.E-6)GO TO 3
      RHOX=0. $ RHOY=0. $ RHOZ=0.
       GO TO 5
 3     RHOX=RHOX/RMAG
       RHOY=RHOY/RMAG
       RHOZ=RHOZ/RMAG
    5 RMAG=SQRTF(YIJ*YIJ+XIJ*XIJ)
       IF(RMAG.GT.1.E-6)GO TO 6
      PX=0. $ PY=0.
       CTH=1.
      ZRSIN=CMPLX(1.,0.)
       GO TO 7
 6     PX=YIJ/RMAG
       PY=-XIJ/RMAG
      CTH=ABS(ZIJ/SQRT(RS))
      IF(IVERTRC) CTH=1. $$$ USE VERTICAL REF. COEF. ONLY--RATHER THAN SPECULAR
       ZRSIN=CSQRT(1.-ZRATI*ZRATI*(1.-CTH*CTH))
 7     REFS=(CTH-ZRATI*ZRSIN)/(CTH+ZRATI*ZRSIN)
       REFPS=-(ZRATI*CTH-ZRSIN)/(ZRATI*CTH+ZRSIN)
       REFPS=REFPS-REFS
      IF(IPGND .NE. 1) GO TO 700
      ZRSIN=CMPLX(1.,0.)
      REFS=CMPLX(1.,0.)
      REFPS=CMPLX(0.,0.)
 700  CONTINUE
       CALL INTG(B,S,RH,ZP,Q1,QP2,ETR,ETI,DIL,DIK,IJ,IP,MEDIA,ETS,ETC,ETK)
```

```
      IF(II0.EQ.2) GO TO G44
      IF(IP.EQ.1) ES(I)=ETS
      IF(IP.EQ.1) EC(I)=ETC
      IF(IP.EQ.1) EK(I)=ETK
      IF(IP.EQ.2) ES(I)=ES(I)+ETS*SIGN
      IF(IP.EQ.2) EC(I)=EC(I)+ETC*SIGN
      IF(IP.EQ.2) EK(I)=EK(I)+ETK*SIGN
  G44 CONTINUE
       IJ=1
       IF(IP.NE.2) GO TO 4
       DO 2 IC=1,3
       ETR(IC)=SIGN*ETR(IC)
 2     ETI(IC)=SIGN*ETI(IC)
4      IF(JCO1)614,611,615
614    CALL JMELS(ETR(1),ETI(1),NCIX,JIX,NCOX,JOX,I)
       GO TO 611
615    CM(I+N*(JCO1-1))=CM(I+N*(JCO1-1))+ETR(1)+FJ*ETI(1)
611    CM(I+N*(J-1))=CM(I+N*(J-1))+ETR(2)+FJ*ETI(2)
       IF(JCO2) 619,300,620
619    CALL JMELS(ETR(3),ETI(3),NCOZ,JOZ,NCIZ,JIZ,I)
       GO TO 300
620    CM(I+N*(JCO2-1))=CM(I+N*(JCO2-1))+ETR(3)+FJ*ETI(3)
300    CONTINUE
C      WOT 3, (4I2,2F5.2,6E15.5) I,J,JCO1,JCO2,DIL,DIK,REAL(ETK),AIMAG(ETK),
C        REAL(ETC),AIMAG(ETC),REAL(ETS),AIMAG(ETS)
  G10 CONTINUE
 3000 CONTINUE

C      WOT 3, (I10,3E15.5) ,(I,REAL(EK(I)),AIMAG(EK(I)),CABS(EK(I)),I=1,N
C     1)

      IF(II0.EQ.2) GO TO D10
       DO DIA I=1,N
       DO DIA J=1,N
  DIA CM(I+N*(J-1))=CMPLX(0.,0.)
      LIMJL=1 $ LIMJU=N $$$ RESET LIMITS ON DO 3000
      LIMIL=1 $ LIMIU=N $$$ RESET LIMITS ON DO 3000
  D10 CONTINUE



COMMENT----FILL CM MATRIX USING TOEPLITZ SYMMETRY



      DO LOOP J=NONTOEP+1,N
      CALL TRIO(J,JCO1,JCO2,DIL,DIK)
      DO LOOP I=NONTOEP+1,N
      SGN=1.
      IF(I-J.LT.0) SGN=-1. $$$ SIN TERM CHANGES SIGN FOR THIS CASE
      INDX=1+IABS(I-J)+NONTOEP $$$ TOEPLITZ INDEX
      ETK=EK(INDX)
      ETS=ES(INDX)*SGN
      ETC=EC(INDX)
COMMENT----DO INTERPOLATION
      IF(MEDIA .EQ. +1) CL=WKP*DIL
      IF(MEDIA .EQ. -1) CL=WKM*DIL
      IF(MEDIA .EQ. +1) CK=WKP*DIK
      IF(MEDIA .EQ. -1) CK=WKM*DIK
      CALL CSINCOS(CL,SINL,COSL)
      CALL CSINCOS(CK,SINK,COSK)
      SILK = SINL*COSK + SINK*COSL
      CONS = 1./(SINL + SINK - SILK)
      ET(1) = (SINK*ETK + (COSK-CMPLX(1.,0.))*ETS - SINK*ETC)*CONS
```

```
      ET(2) = (-SILK*ETK + (COSL-COSK)*ETS + (SINL+SINK)*ETC)*CONS
      ET(3) = (SINL*ETK + (CMPLX(1.,0.)-COSL)*ETS - SINL*ETC)*CONS
      CM(I+N*(J-1))=CM(I+N*(J-1))+ET(2)
      IF(JCO1.LT.0) CALL JMELS(REAL(ET(1)),AIMAG(ET(1)),NC1X,J1X,NCOX,JOX,I)
      IF(JCO1.GT.0) CM(I+N*(JCO1-1))=CM(I+N*(JCO1-1))+ET(1)
      IF(JCO2.LT.0) CALL JMELS(REAL(ET(3)),AIMAG(ET(3)),NCOZ,JOZ,NC1Z,J1Z,I)
      IF(JCO2.GT.0) CM(I+N*(JCO2-1))=CM(I+N*(JCO2-1))+ET(3)
 LOOP CONTINUE
      IF(ITIMING) CALL ENDMAP(3)
      IF(KSYMP.EQ.3) WOT 59, (8HNEVALS= I10,1X,I10) ,NEVALS,NEVALSR
      IF(KSYMP.EQ.3) WOT 3, (//8HNEVALS= I10,1X,I10//) ,NEVALS,NEVALSR
      RETURN
C
C  THIS IS A PATCH UNTIL TRANSMISSION CALCULATIONS ARE READY
C
 740  WRITE(3,FERR)
 FERR FORMAT(  A MIXED MEDIA CALCULATION ERROR WAS GENERATED-STOPPED )
      CALL EXIT
COMMENT----WHEN CTRL-E I IS RECEIVED CONTROL IS TRANSFERRED HERE
INTERRUPT CONTINUE
      CALL TICHEK(DUMMY1,DUMMY2,ICPU,I10)
FRMAT FORMAT(     JOBNO= ,I3,     I10= ,I1,     J= ,I3,     I= ,I3,
            TIME= F6.2,  (MIN) )
      COMMON/JOBCOM/JOBNO
      WOT 59,FRMAT,JOBNO,I10,J,I,(ICPU+I10)/60.E6
COMMENT----RESET INTERRUPT FLAG
      LOCIN=.LOC.INTERRUPT
      CALL FROST(33B,01B,0,0,LOCIN,0,ERRINT2)
ERRINT2 CONTINUE $$$ IGNORE ERRORS
COMMENT----RETURN FROM INTERRUPT
      ABSOLUTE IWORD0(0)
      IWORD0=20000000B
      CALL QBEXCH
       END
```

```
      CODE ANALYSIS
      COMPLEX CROOTS,Z,XK,XL
      FUNCTION CROOTS(Z,XK,XL)
COMMENT----RETURNS CORRECT SQUARE ROOT OF Z RELATIVE TO BRANCH POINT
COMMENT----XK AND XL (LAMBDA)
      CROOTS=CSQRT(Z)
      IF(REAL(XL).LT.REAL(XK) .AND. AIMAG(XL).GE.AIMAG(XK)) CROOTS=-CROOTS
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE EVALUA2(ERV,EZV,ERH,EZH,EPH) $$$ SOURCE ABOVE
      PARAMETER (PI=3.1415926535897932)
      COMMON /EVALCOM/ HI,ZI,ZMH,ZPH,R,RI,R2,-COMPLEX-CJ,
        CKI,CKISQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX C1,C2,C3
      COMPLEX ERV,EZV,ERH,EZH,EPH
      COMPLEX COEE1,COEE2,COEH1,COEH2
      COMPLEX CE1R,CE1Z,CE1RH,CE1ZH,CE1PH
      COMPLEX CE2R,CE2Z,CE2RH,CE2ZH,CE2PH

      COEE1=COEE
      COEE2=COEE/CK2SQ
      COEH1=COEH*CK1SQ
      COEH2=COEH

      IF(ICALLED.EQ.0) WOT 3, (     EVALUA2 CALLED   )
      ICALLED=1

      IF(ZI.LT.0) GO TO 150


C        IT IS TO BE NOTED THAT THE SINE OF PHI AND COSINE OF PHI TERMS
C        THAT ARE ASSOCIATED WITH THE HORIZONTAL ELECRIC DIPOLE TERMS HAVE
C        BEEN OMITTED FROM THE EQUATIONS FOR THE FIELDS DUE TO A
C        HORIZONTALLY ORIENTED SOURCE. ONLY THE REAL AND THE IMAGINARY
C        (OR THE MAGNITUDE AND THE PHASE) TERMS FOR THE FIELDS HAVE BEEN
C        CALCULATED. IT IS ASSUMED THAT THE PERSON USING THIS PROGRAM
C        KNOWS WHICH FUNCTION OF PHI GOES WITH WHICH COMPONENT OF THE
C        VARIOUS ELECTROMAGNETIC FIELDS.
C

C        THIS IS THE GENERAL EVALUATION

C        THIS SECTION IS USED FOR THE EVALUATION OF THE U AND V BASIC
C        INTEGRALS. THIS SECTION IS FOR Z LESS THAN ZERO, THUS THE U AND
C        V INTEGRAL FORMS ARE FOR A SUBSCRIPT OF 22.

      DIMENSION -COMPLEX-ANS(10),SUM(10)
      EXTERNAL SAOA,SAOB
      COMMON/CONTOUR/ ITYPE,-COMPLEX-A,B
      COMMON/ROMCOM/-INTEGER-NEVALS,TROUBLE
      NEVALS=0
      DO DZEROA I=1,6
DZEROA SUM(I)=0

      DIMENSION PATH(8)
      PATH(1)=0
      PATH(2)=0
      PATH(3)=1./MAXIF(R,HI+ABS(ZI))
      PATH(4)=-PATH(3)

      DO RLOOP1 I=1,1,2
      ITYPE=1 $ A=CMPLX(PATH(I),PATH(I+1)) $ B=CMPLX(PATH(I+2),PATH(I+3))
      CALL ROMBERG(0,1.,SAOA,6,ANS,10) $$$ SOURCE ABOVE OBSERVER ABOVE
C       IF(TROUBLE.NE.0) WOT 59, (8HTROUBLE=,I10) ,TROUBLE
      IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C       WOT 59, (7HNEVALS=,I10) ,NEVALS
C       WOT  3, (7HNEVALS=,I10) ,NEVALS
C       WOT 3, (//9HPATH NO. I2) ,I
C       WOT 3, (3E15.5) ,(REAL(ANS(J)),AIMAG(ANS(J)),CABS(ANS(J)),J=1,7)
      DO DR1 J=1,6
  DR1 SUM(J)=SUM(J)+ANS(J)
RLOOP1 CONTINUE
```

```
C      NEVALS=0
      DEL=.2*PI*(1./MAXIF(R,HI+ABS(ZI)))
      COMPLEX DELTA
      DELTA=DEL
      CALL GSHANK(SAOA,CMPLX(PATH(3),PATH(4)),DELTA,ANS,6,SUM,
     .  0,CMPLX(0.,0.),CMPLX(0.,0.))
      DO DSH1 I=1,6
 DSH1 SUM(I)=ANS(I)
      IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C      WOT  3, (7HNEVALS=,I10) ,NEVALS
C      WOT 3, (3E15.5) ,(REAL(SUM(I)),AIMAG(SUM(I)),CABS(SUM(I)),I=1,6)

C
C      THIS SECTION COMBINES THE REAL AND THE IMAGINARY PARTS OF THE SIX
C      BASIC INTEGRALS IN ORDER TO FORM THE FIELD INTENSITIES.

      C1=SUM(3)
      C2=SUM(5)*CK2SQ+SUM(2)
      C3=SUM(4)

C    THIS SECTION EVALUATES THE FIELDS FOR A VERTICAL ELECTRIC DIPOLE.

      CE2R=COEE2*CK1SQ*2.*C1
      CE2Z=COEE2*CK1SQ*2.*C2
      CH2A=COEH2*CK1SQ*2.*C3


C    THIS SECTION EVALUATES THE FIELDS FOR A HORIZONTAL ELECTRIC DIPOLE

      C1=SUM(1)+SUM(6)
      IF(R.EQ.0) C2=SUM(4)+SUM(6)
      IF(R.NE.0) C2=SUM(4)/R+SUM(6)
      C3=SUM(3)

      CE2RH=COEE2*CK2SQ*2.*C1
      CE2PH=-COEE2*CK2SQ*2.*C2
      CE2ZH=-COEE2*2.*CK1SQ*C3

      ERV=CE2R
      EZV=CE2Z
      ERH=CE2RH
      EZH=CE2ZH
      EPH=CE2PH

      RETURN

C      THE FOLLOWING SECTION IS USED IF Z IS .LT. 0.

  150 CONTINUE
C
C      THIS SECTION PERFORMS THE ROMBERG INTEGRATIONS FOR THE U AND
C      V INTEGRALS WITH A 21 SUBSCRIPT.

C      THE PROCEDURE FOLLOWED IS THE SAME AS DESCRIBED ABOVE FOR Z
C      LESS THAN ZERO.

C      NEVALS=0
      DO DZEROB I=1,7
DZEROB SUM(I)=0

      PATH(1)=0
      PATH(2)=0
      PATH(3)=1./MAXIF(R,HI+ABS(ZI))
```

```
      PATH(4)=-PATH(3)

      DO RLOOP2 I=1,1,2
      ITYPE=1 $ A=CMPLX(PATH(I),PATH(I+1)) $ B=CMPLX(PATH(I+2),PATH(I+3))
      CALL ROMBERG(0,1.,SAOB,7,ANS,10) $$$ SOURCE ABOVE OBSERVER BELOW
C      IF(TROUBLE.NE.0) WOT 59, (8HTROUBLE=,I10) ,TROUBLE
      IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C      WOT 59, (7HNEVALS=,I10) ,NEVALS
C      WOT  3, (7HNEVALS=,I10) ,NEVALS
C      WOT 3, (//9HPATH NO. I2) ,I
C      WOT 3, (3E15.5) ,(REAL(ANS(J)),AIMAG(ANS(J)),CABS(ANS(J)),J=1,7)
      DO DR2 J=1,7
  DR2 SUM(J)=SUM(J)+ANS(J)
RLOOP2 CONTINUE

C      NEVALS=0
      DEL=.2*PI*(1./MAXIF(R,HI+ABS(ZI)))
      DELTA=DEL
      CALL GSHANK(SAOB,CMPLX(PATH(3),PATH(4)),DELTA,ANS,7,SUM,
     .  0,CMPLX(0.,0.),CMPLX(0.,0.))
      DO DSH2 I=1,7
 DSH2 SUM(I)=ANS(I)
      IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C      WOT  3, (7HNEVALS=,I10) ,NEVALS
C      WOT 3, (3E15.5) ,(REAL(SUM(I)),AIMAG(SUM(I)),CABS(SUM(I)),I=1,7)

      C1=SUM(3)
      C2=SUM(5)*CK1SQ+SUM(2)
      C3=SUM(4)

C    THIS SECTION EVALUATES THE FIELDS FOR A VERTICAL ELECTRIC DIPOLE.

       CEIR = COEE1*2.*C1
       CEIZ = COEE1*2.*C2
       CHIA = COEH1*2.*C3

C    THIS SECTION EVALUATES THE FIELDS FOR A HORIZONTAL ELECTRIC DIPOLE

C       THE SINGULARITY DIFFICULTY AT RHO = 0 (MENTIONED ABOVE) OCCURS
C       HERE ALSO, SIMILAR REMARKS AS ABOVE APPLY.

      C1=SUM(1)+SUM(6)
      IF(R.EQ.0) C2=SUM(4)+SUM(6)
      IF(R.NE.0) C2=SUM(4)/R+SUM(6)
      C3=SUM(7) $$$ MAKE SURE THIS IS RIGHT
       CEIRH = COEE1*2.*C1
       CEIPH = -COEE1*2.*C2
       CEIZH = -COEE1*2.*C3
      ERV=CEIR
      EZV=CEIZ
      ERH=CEIRH
      EZH=CEIZH
      EPH=CEIPH
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE EVALUA3(ERV,EZV,ERH,EZH,EPH) $$$ SOURCE ABOVE
      PARAMETER (PI=3.1415926535897932)
      COMMON /EVALCOM/ HI,ZI,ZMH,ZPH,R,RI,R2,-COMPLEX-CJ,
     .  CKI,CKISQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX C1,C2,C3
      COMPLEX ERV,EZV,ERH,EZH,EPH
      COMPLEX COEE1,COEE2,COEH1,COEH2
      COMPLEX CE1R,CE1Z,CE1RH,CE1ZH,CE1PH
      COMPLEX CE2R,CE2Z,CE2RH,CE2ZH,CE2PH

      COEE1=COEE
      COEE2=COEE/CK2SQ
      COEH1=COEH*CK1SQ
      COEH2=COEH

      IF(ICALLED.EQ.0) WOT 3, (     EVALUA3 CALLED  )
      ICALLED=1

      IF(ZI.LT.0) GO TO 150


C       IT IS TO BE NOTED THAT THE SINE OF PHI AND COSINE OF PHI TERMS
C       THAT ARE ASSOCIATED WITH THE HORIZONTAL ELECRIC DIPOLE TERMS HAVE
C       BEEN OMITTED FROM THE EQUATIONS FOR THE FIELDS DUE TO A
C       HORIZONTALLY ORIENTED SOURCE. ONLY THE REAL AND THE IMAGINARY
C       (OR THE MAGNITUDE AND THE PHASE) TERMS FOR THE FIELDS HAVE BEEN
C       CALCULATED. IT IS ASSUMED THAT THE PERSON USING THIS PROGRAM
C       KNOWS WHICH FUNCTION OF PHI GOES WITH WHICH COMPONENT OF THE
C       VARIOUS ELECTROMAGNETIC FIELDS.
C

C       THIS IS THE GENERAL EVALUATION

C       THIS SECTION IS USED FOR THE EVALUATION OF THE U AND V BASIC
C       INTEGRALS. THIS SECTION IS FOR Z LESS THAN ZERO, THUS THE U AND
C       V INTEGRAL FORMS ARE FOR A SUBSCRIPT OF 22.

      DIMENSION -COMPLEX-ANS(10),SUM(10)
      EXTERNAL HSAOA,HSAOB
      COMMON/CONTOUR/ ITYPE,-COMPLEX-A,B
      COMMON/ROMCOM/-INTEGER-NEVALS,TROUBLE
C      NEVALS=0
      DO DZEROA I=1,6
DZEROA SUM(I)=0
      DIMENSION PATH(8)
      PATH(1)=.2*REAL(CK2)
      PATH(2)=-.5*REAL(CK2)
      PATH(3)=REAL(CK2)
      PATH(4)=PATH(2)

      PATH(1)=0
      PATH(2)=.5*REAL(CK2)
      PATH(3)=.2*REAL(CK2)
      PATH(4)=PATH(2)
      PATH(5)=PATH(3)
      PATH(6)=-.5*REAL(CK2)
      PATH(7)=REAL(CK2)
      PATH(8)=PATH(6)

      DO RLOOP1 I=1,5,2
      ITYPE=1 $ A=CMPLX(PATH(I),PATH(I+1)) $ B=CMPLX(PATH(I+2),PATH(I+3))
      CALL ROMBERG(0,1.,HSAOA,6,ANS,10) $$$ SOURCE ABOVE OBSERVER ABOVE
C      WOT 3, (  --PATH--      ,I5,I10,4E15.5) ,I,NEVALS,
```

```
C       .  REAL(A),AIMAG(A),REAL(B),AIMAG(B)
C        IF(TROUBLE.NE.0) WOT 59, (8HTROUBLE=,I10) ,TROUBLE
        IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C        WOT 59, (7HNEVALS=,I10) ,NEVALS
C        WOT  3, (7HNEVALS=,I10) ,NEVALS
C        WOT 3, (//9HPATH NO. I2) ,I
C        WOT 3, (3E15.5) ,(REAL(ANS(J)),AIMAG(ANS(J)),CABS(ANS(J)),J=1,7)
        DO DRI J=1,6
  DRI  SUM(J)=SUM(J)+ANS(J)
RLOOPI CONTINUE
C        NEVALS=0
        BLIM=PATH(3)
        NCON=0
        DEL=.2*PI*(1./MAXIF(R,HI+ABS(ZI)))
        DEL=.1*REAL(CK2)
        DEL=.2*PI/R
        DEL=MINIF(.2*PI/R,2.*REAL(CK2))
        DEL=.2*PI/R    $$$ SPECIAL CASE FOR ZI=HI=0
        COMPLEX DELTA
        DELTA=CMPLX(0.,DEL)   $$$ PATH PARALLEL TO IMAGINARY AXIS
        DO DSWINI I=1,6
DSWINI SUM(I)=-SUM(I)   $$$ SWINDLE SINCE DOING PATH BACKWARD
COMMENT----SLOPING PATH PASSING THRU (0.+.5*CK2)
        DELTA=CMPLX(-.2*DEL,DEL)
        DELTA=CMPLX(0.,DEL)   $$$ PATH PARALLEL TO IMAGINARY AXIS
        SLOPE=R/(HI+ABS(ZI))
        DEL=.2*PI/MAXIF(R,HI+ABS(ZI))
        DELTA=CMPLX(-DEL,SLOPE*DEL)
        IF(ABS(SLOPE).GE.1.) DELTA=CMPLX(-DEL/SLOPE,DEL)
        CALL GSHANK(HSAOA,CMPLX(PATH(1),PATH(2)),DELTA,ANS,6,SUM,
       . 0,CMPLX(0.,0.),CMPLX(0.,0.))
        DO DSHI I=1,6
  DSHI SUM(I)=-ANS(I)   $$$ SWINDLE SINCE DID PATH BACKWARD
COMMENT----SLOPING PATH PASSING THRU CKI
COMMENT----THE .99 FACTOR IS TO AVOID PASSING DIRECTLY THRU CKI
C        DEL=.2*PI/R    $$$ SPECIAL CASE FOR ZI=HI=0
        DELDEL=(1.01*REAL(CKI)-PATH(7))/(.99*AIMAG(CKI)-PATH(8))
        DELTA=CMPLX(DEL*DELDEL,DEL)
        IF(DELDEL.GT.1.) DELTA=CMPLX(DEL,DEL/DELDEL)
        COMPLEX BREAK,DELTA2
        DELTA2=CMPLX(DEL,SLOPE*DEL)
        IF(SLOPE.GT.1) DELTA2=CMPLX(DEL/SLOPE,DEL)
        BREAK=CMPLX(1.01*REAL(CKI),.99*AIMAG(CKI))
        CALL GSHANK(HSAOA,CMPLX(PATH(7),PATH(8)),DELTA,ANS,6,SUM,
          1,BREAK,DELTA2)
        DO DSWIN2 I=1,6
DSWIN2 SUM(I)=ANS(I)
        IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C        WOT 59, (7HNEVALS=,I10) ,NEVALS
C        WOT  3, (7HNEVALS=,I10) ,NEVALS
C        WOT 3, (3E15.5) ,(REAL(SUM(I)),AIMAG(SUM(I)),CABS(SUM(I)),I=1,6)

C
C        THIS SECTION COMBINES THE REAL AND THE IMAGINARY PARTS OF THE SIX
C        BASIC INTEGRALS IN ORDER TO FORM THE FIELD INTENSITIES.

        C1=SUM(3)
        C2=SUM(5)*CK2SQ+SUM(2)
        C3=SUM(4)

C     THIS SECTION EVALUATES THE FIELDS FOR A VERTICAL ELECTRIC DIPOLE.

        CE2R=COEE2*CKISQ*2.*C1
        CE2Z=COEE2*CKISQ*2.*C2
```

```
      CH2A=COEH2*CK1SQ*2.*C3


C     THIS SECTION EVALUATES THE FIELDS FOR A HORIZONTAL ELECTRIC DIPOLE

      C1=SUM(1)+SUM(6)
      IF(R.EQ.0) C2=SUM(4)+SUM(6)
      IF(R.NE.0) C2=SUM(4)/R+SUM(6)
      C3=SUM(3)

      CE2RH=COEE2*CK2SQ*2.*C1
      CE2PH=-COEE2*CK2SQ*2.*C2
      CE2ZH=-COEE2*2.*CK1SQ*C3
      ERV=CE2R
      EZV=CE2Z
      ERH=CE2RH
      EZH=CE2ZH
      EPH=CE2PH

      RETURN

C       THE FOLLOWING SECTION IS USED IF Z IS .LT. 0.

 150  CONTINUE
C
C       THIS SECTION PERFORMS THE ROMBERG INTEGRATIONS FOR THE U AND
C       V INTEGRALS WITH A 21 SUBSCRIPT.

C       THE PROCEDURE FOLLOWED IS THE SAME AS DESCRIBED ABOVE FOR Z
C       LESS THAN ZERO.

C       NEVALS=0
      DO DZEROB I=1,7
DZEROB SUM(I)=0

      PATH(1)=0
      PATH(2)=.5*REAL(CK2)
      PATH(3)=.2*REAL(CK2)
      PATH(4)=PATH(2)
      PATH(5)=PATH(3)
      PATH(6)=-.5*REAL(CK2)
      PATH(7)=REAL(CK2)
      PATH(8)=PATH(6)

      DO RLOOP2 I=1,5,2
      ITYPE=1 $ A=CMPLX(PATH(I),PATH(I+1)) $ B=CMPLX(PATH(I+2),PATH(I+3))
      CALL ROMBERG(0,1.,HSAOB,7,ANS,10) $$$ SOURCE ABOVE OBSERVER BELOW
C      IF(TROUBLE.NE.0) WOT 59, (8HTROUBLE=,I10) ,TROUBLE
      IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C      WOT 59, (7HNEVALS=,I10) ,NEVALS
C      WOT  3, (7HNEVALS=,I10) ,NEVALS
C      WOT 3, (//9HPATH NO. I2) ,I
C      WOT 3, (3E15.5) ,(REAL(ANS(J)),AIMAG(ANS(J)),CABS(ANS(J)),J=1,7)
      DO DR2 J=1,7
  DR2 SUM(J)=SUM(J)+ANS(J)
RLOOP2 CONTINUE


C       NEVALS=0
      DO DSWIN18 I=1,7
DSWIN18 SUM(I)=-SUM(I) $$$ SINCE DOING PATH BACKWARD

      SLOPE=R/(HI+ABS(ZI))
      DEL=.2*PI*(1./MAXIF(R,HI+ABS(ZI)))
```

```
      DELTA=CMPLX(-DEL,SLOPE*DEL)
      IF(ABS(SLOPE).GE.1.) DELTA=CMPLX(-DEL/SLOPE,DEL)
      CALL GSHANK(HSA08,CMPLX(PATH(1),PATH(2)),DELTA,ANS,7,SUM,
     0,CMPLX(0.,0.),CMPLX(0.,0.))
      DO DSH2 I=1,7
 DSH2 SUM(I)=-ANS(I) $$$ SINCE DDID PATH BACKWARD
COMMENT----SLOPING PATH PASSING THRU CK1
COMMENT----THE .99 FACTOR IS TO AVOID PASSING DIRECTLY THRU CK1
      DELDEL=(1.01*REAL(CK1)-PATH(7))/(.99*AIMAG(CK1)-PATH(8))
      DELTA=CMPLX(DEL*DELDEL,DEL)
      IF(DELDEL.GT.1.) DELTA=CMPLX(DEL,DEL/DELDEL)
      DELTA2=CMPLX(DEL,SLOPE*DEL)
      IF(SLOPE.GT.1) DELTA2=CMPLX(DEL/SLOPE,DEL)
      BREAK=CMPLX(1.01*REAL(CK1),.99*AIMAG(CK1))
      CALL GSHANK(HSA08,CMPLX(PATH(7),PATH(8)),DELTA,ANS,7,SUM,
     1,BREAK,DELTA2)
      DO DSWIN28 I=1,7
DSWIN28 SUM(I)=ANS(I)

      IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C      WOT  3, (7HNEVALS=,I10) ,NEVALS
C      WOT 3, (3E15.5) ,(REAL(SUM(I)),AIMAG(SUM(I)),CABS(SUM(I)),I=1,7)

      C1=SUM(3)
      C2=SUM(5)*CK1SQ+SUM(2)
      C3=SUM(4)

C    THIS SECTION EVALUATES THE FIELDS FOR A VERTICAL ELECTRIC DIPOLE.

       CE1R = COEE1*2.*C1
       CE1Z = COEE1*2.*C2
       CH1A = COEH1*2.*C3

C    THIS SECTION EVALUATES THE FIELDS FOR A HORIZONTAL ELECTRIC DIPOLE

C      THE SINGULARITY DIFFICULTY AT RHO = 0 (MENTIONED ABOVE) OCCURS
C      HERE ALSO, SIMILAR REMARKS AS ABOVE APPLY.

      C1=SUM(1)+SUM(6)
      IF(R.EQ.0) C2=SUM(4)+SUM(6)
      IF(R.NE.0) C2=SUM(4)/R+SUM(6)
      C3=SUM(7) $$$ MAKE SURE THIS IS RIGHT
       CE1RH = COEE1*2.*C1
       CE1PH = -COEE1*2.*C2
       CE1ZH = -COEE1*2.*C3
      ERV=CE1R
      EZV=CE1Z
      ERH=CE1RH
      EZH=CE1ZH
      EPH=CE1PH
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE EVALUB2(ERV,EZV,ERH,EZH,EPH) $$$ SOURCE BELOW
      PARAMETER (PI=3.1415926535897932)
      COMMON /EVALCOM/ HI,ZI,ZMH,ZPH,R,RI,R2,-COMPLEX-CJ,
     . CKI,CKISQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX C1,`2,C3
      COMPLEX ERV,EZV,ERH,EZH,EPH
      COMPLEX COEEI,COEE2,COEHI,COEH2
      COMPLEX CEIR,CEIZ,CEIRH,CEIZH,CEIPH
      COMPLEX CE2R,CE2Z,CE2RH,CE2ZH,CE2PH

      COEEI=COEE/CKISQ
      COEE2=COEE
      COEHI=COEH
      COEH2=COEH*CK2SQ

      IF(ICALLED.EQ.0) WOT 3, (     EVALUB2 CALLED  )
      ICALLED=1

      IF(ZI.GE.0) GO TO 150


C       IT IS TO BE NOTED THAT THE SINE OF PHI AND COSINE OF PHI TERMS
C       THAT ARE ASSOCIATED WITH THE HORIZONTAL ELECRIC DIPOLE TERMS HAVE
C       BEEN OMITTED FROM THE EQUATIONS FOR THE FIELDS DUE TO A
C       HORIZONTALLY ORIENTED SOURCE. ONLY THE REAL AND THE IMAGINARY
C       (OR THE MAGNITUDE AND THE PHASE) TERMS FOR THE FIELDS HAVE BEEN
C       CALCULATED. IT IS ASSUMED THAT THE PERSON USING THIS PROGRAM
C       KNOWS WHICH FUNCTION OF PHI GOES WITH WHICH COMPONENT OF THE
C       VARIOUS ELECTROMAGNETIC FIELDS.
C

C       THIS IS THE GENERAL EVALUATION

C       THIS SECTION IS USED FOR THE EVALUATION OF THE U AND V BASIC
C       INTEGRALS. THIS SECTION IS FOR Z LESS THAN ZERO. THUS THE U AND
C       V INTEGRAL FORMS ARE FOR A SUBSCRIPT OF 11.

      DIMENSION -COMPLEX-ANS(10),SUM(10)
      EXTERNAL SBOA,SBOB
      COMMON/CONTOUR/ ITYPE,-COMPLEX-A,B
      COMMON/ROMCOM/-INTEGER-NEVALS,TROUBLE
C      NEVALS=0
      DO DZEROA I=1,6
DZEROA SUM(I)=0

      DIMENSION PATH(8)
      PATH(1)=0
      PATH(2)=0
      PATH(3)=1./MAXIF(R,HI+ABS(ZI))
      PATH(4)=-PATH(3)

      DO RLOOPI I=1,1,2
      ITYPE=1 $ A=CMPLX(PATH(I),PATH(I+1)) $ B=CMPLX(PATH(I+2),PATH(I+3))
      CALL ROMBERG(0,1.,SBOB,6,ANS,10) $$$ SOURCE BELOW OBSERVER BELOW
      IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C      WOT  3, (7HNEVALS=,I10) ,NEVALS
C      WOT 3, (//9HPATH NO. I2) ,I
C      WOT 3, (3E15.5) ,(REAL(ANS(J)),AIMAG(ANS(J)),CABS(ANS(J)),J=1,7)
      DO DRI J=1,6
  DRI SUM(J)=SUM(J)+ANS(J)
RLOOPI CONTINUE

C      NEVALS=0
```

```
      DEL=.2*PI*(1./MAX1F(R,H1+ABS(Z1)))
      COMPLEX DELTA
      DELTA=DEL
      CALL GSHANK(SBOB,CMPLX(PATH(3),PATH(4)),DELTA,ANS,6,SUM,
     . 0,CMPLX(0.,0.),CMPLX(0.,0.))
      DO DSH1 I=1,6
 DSH1 SUM(I)=ANS(I)
      IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C      WOT  3, (7HNEVALS=,I10) ,NEVALS
C      WOT 3, (3E15.5) ,(REAL(SUM(I)),AIMAG(SUM(I)),CABS(SUM(I)),I=1,6)

C

C      THIS SECTION COMBINES THE REAL AND THE IMAGINARY PARTS OF THE SIX
C      BASIC INTEGRALS IN ORDER TO FORM THE FIELD INTENSITIES.

      C1=SUM(3)
      C2=SUM(5)*CK1SQ+SUM(2)
      C3=SUM(4)

C    THIS SECTION EVALUATES THE FIELDS FOR A VERTICAL ELECTRIC DIPOLE.

      CE1R=COEE1*CK2SQ*2.*C1
      CE1Z=COEE1*CK2SQ*2.*C2
      CH1A=COEH1*CK2SQ*2.*C3


C    THIS SECTION EVALUATES THE FIELDS FOR A HORIZONTAL ELECTRIC DIPOLE

      C1=SUM(1)+SUM(6)
      IF(R.EQ.0) C2=SUM(4)+SUM(6)
      IF(R.NE.0) C2=SUM(4)/R+SUM(6)
      C3=SUM(3)
      CE1RH=COEE1*CK1SQ*2.*C1
      CE1PH=-COEE1*CK1SQ*2.*C2
      CE1ZH=-COEE1*2.*CK2SQ*C3

      ERV=CE1R
      EZV=CE1Z
      ERH=CE1RH
      EZH=CE1ZH
      EPH=CE1PH

      RETURN

C      THE FOLLOWING SECTION IS USED IF Z IS .GT. 0.

  150 CONTINUE
C
C      THIS SECTION PERFORMS THE ROMBERG INTEGRATIONS FOR THE U AND
C      V INTEGRALS WITH A 12 SUBSCRIPT.

C      THE PROCEDURE FOLLOWED IS THE SAME AS DESCRIBED ABOVE FOR Z
C      LESS THAN ZERO.

C      NEVALS=0
      DO DZEROB I=1,7
 DZEROB SUM(I)=0

      PATH(1)=0
      PATH(2)=0
      PATH(3)=1./MAX1F(R,H1+ABS(Z1))
      PATH(4)=-PATH(3)

      DO RLOOP2 I=1,1,2
```

```
      ITYPE=1 $ A=CMPLX(PATH(I),PATH(I+1)) $ B=CMPLX(PATH(I+2),PATH(I+3))
      CALL ROMBERG(0,1.,SBOA,7,ANS,10) $$$ SOURCE BELOW OBSERVER ABOVE
C      IF(TROUBLE.NE.0) WOT 59, (8HTROUBLE=,I10) ,TROUBLE
      IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C      WOT 59, (7HNEVALS=,I10) ,NEVALS
C      WOT  3, (7HNEVALS=,I10) ,NEVALS
C      WOT 3, (//9HPATH NO. I2) ,I
C      WOT 3, (3E15.5) ,(REAL(ANS(J)),AIMAG(ANS(J)),CABS(ANS(J)),J=1,7)
      DO DR2 J=1,7
  DR2 SUM(J)=SUM(J)+ANS(J)
RLOOP2 CONTINUE


C      NEVALS=0
      DEL=.2*PI*(1./MAXIF(R,H1+ABS(Z1)))
      DELTA=DEL
      CALL GSHANK(SBOA,CMPLX(PATH(3),PATH(4)),DELTA,ANS,7,SUM,
        0,CMPLX(0.,0.),CMPLX(0.,0.))
      DO DSH2 I=1,7
 DSH2 SUM(I)=ANS(I)
      IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C      WOT  3, (7HNEVALS=,I10) ,NEVALS
C      WOT 3, (3E15.5) ,(REAL(SUM(I)),AIMAG(SUM(I)),CABS(SUM(I)),I=1,7)

      C1=SUM(3)
      C2=SUM(5)*CK2SQ+SUM(2)
      C3=SUM(4)

C    THIS SECTION EVALUATES THE FIELDS FOR A VERTICAL ELECTRIC DIPOLE.

       CE2R = COEE2*2.*C1
       CE2Z = COEE2*2.*C2
       CH2A = COEH2*2.*C3

C    THIS SECTION EVALUATES THE FIELDS FOR A HORIZONTAL ELECTRIC DIPOLE

C      THE SINGULARITY DIFFICULTY AT RHO = 0 (MENTIONED ABOVE) OCCURS
C      HERE ALSO. SIMILAR REMARKS AS ABOVE APPLY.

      C1=SUM(1)+SUM(6)
      IF(R.EQ.0) C2=SUM(4)+SUM(6)
      IF(R.NE.0) C2=SUM(4)/R+SUM(6)
      C3=SUM(7) $$$ MAKE SURE THIS IS RIGHT
       CE2RH = COEE2*2.*C1
       CE2PH = -COEE2*2.*C2
       CE2ZH = -COEE2*2.*C3
      ERV=CE2R
      EZV=CE2Z
      ERH=CE2RH
      EZH=CE2ZH
      EPH=CE2PH
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE EVALUB3(ERV,EZV,ERH,EZH,EPH) $$$ SOURCE BELOW
      PARAMETER (PI=3.14159265358979321)
      COMMON /EVALCOM/ HI,ZI,ZMH,ZPH,R,RI,R2,-COMPLEX-CJ,
     .  CKI,CKISQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX C1,C2,C3
      COMPLEX ERV,EZV,ERH,EZH,EPH
      COMPLEX COEE1,COEE2,COEH1,COEH2
      COMPLEX CE1R,CE1Z,CE1RH,CE1ZH,CE1PH
      COMPLEX CE2R,CE2Z,CE2RH,CE2ZH,CE2PH

      COEE1=COEE/CK1SQ
      COEE2=COEE
      COEH1=COEH
      COEH2=COEH*CK2SQ

      IF(ICALLED.EQ.0) WOT 3, (     EVALUB3 CALLED   )
      ICALLED=1

      IF(ZI.GE.0) GO TO 150


C       IT IS TO BE NOTED THAT THE SINE OF PHI AND COSINE OF PHI TERMS
C       THAT ARE ASSOCIATED WITH THE HORIZONTAL ELECRIC DIPOLE TERMS HAVE
C       BEEN OMITTED FROM THE EQUATIONS FOR THE FIELDS DUE TO A
C       HORIZONTALLY ORIENTED SOURCE. ONLY THE REAL AND THE IMAGINARY
C       (OR THE MAGNITUDE AND THE PHASE) TERMS FOR THE FIELDS HAVE BEEN
C       CALCULATED. IT IS ASSUMED THAT THE PERSON USING THIS PROGRAM
C       KNOWS WHICH FUNCTION OF PHI GOES WITH WHICH COMPONENT OF THE
C       VARIOUS ELECTROMAGNETIC FIELDS.
C

C       THIS IS THE GENERAL EVALUATION

C       THIS SECTION IS USED FOR THE EVALUATION OF THE U AND V BASIC
C       INTEGRALS. THIS SECTION IS FOR Z LESS THAN ZERO. THUS THE U AND
C       V INTEGRAL FORMS ARE FOR A SUBSCRIPT OF 11.

       DIMENSION -COMPLEX-ANS(10),SUM(10)
       EXTERNAL HSBOA,HSBOB
       COMMON/CONTOUR/ ITYPE,-COMPLEX-A,B
       COMMON/ROMCOM/-INTEGER-NEVALS,TROUBLE
C       NEVALS=0
       DO DZEROA I=1,6
DZEROA SUM(I)=0
       DIMENSION PATH(8)

       PATH(1)=0
       PATH(2)=.5*REAL(CK2)
       PATH(3)=.2*REAL(CK2)
       PATH(4)=PATH(2)
       PATH(5)=PATH(3)
       PATH(6)=-.5*REAL(CK2)
       PATH(7)=REAL(CK2)
       PATH(8)=PATH(6)

       DO RLOOP1 I=1,5,2
       ITYPE=1 $ A=CMPLX(PATH(I),PATH(I+1)) $ B=CMPLX(PATH(I+2),PATH(I+3))
       CALL ROMBERG(0,1.,HSBOB,6,ANS,10) $$$ SOURCE BELOW OBSERVER BELOW
       IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I 0) ,TROUBLE
C       WOT  3, (7HNEVALS=,I10) ,NEVALS
C       WOT 3, (//5HPATH NO. I2) , I
C       WOT 3, (3E15.5) ,(REAL(ANS(J)),AIMAG(ANS(J)),CABS(ANS(J)),J=1,7)
       DO CR1 J=1,6
```

```
   DR1 SUM(J)=SUM(J)+ANS(J)
RLOOP1 CONTINUE


C       NEVALS=0
      DO DSWIN1 I=1,6
DSWIN1 SUM(I)=-SUM(I) $$$ SINCE DOING PATH BACKWARD

      SLOPE=R/(H1+ABS(Z1))
      DEL=.2*PI*(1./MAX1F(R,H1+ABS(Z1)))
      COMPLEX DELTA
      DELTA=CMPLX(-DEL,SLOPE*DEL)
      IF(ABS(SLOPE).GE.1.) DELTA=CMPLX(-DEL/SLOPE,DEL)
      CALL GSHANK(HS808,CMPLX(PATH(1),PATH(2)),DELTA,ANS,6,SUM,
     .  0,CMPLX(0.,0.),CMPLX(0.,0.))
      DO DSH1 I=1,6
  DSH1 SUM(I)=-ANS(I)  $$$ SINCE DID PATH BACKWARD

COMMENT----SLOPING PATH PASSING THRU CK1
COMMENT----THE .99 FACTOR IS TO AVOID PASSING DIRECTLY THRU CK1
      DELDEL=(1.01*REAL(CK1)-PATH(7))/(.99*AIMAG(CK1)-PATH(8))
      DELTA=CMPLX(DEL*DELDEL,DEL)
      IF(DELDEL.GT.1.) DELTA=CMPLX(DEL,DEL/DELDEL)
      COMPLEX BREAK,DELTA2
      DELTA2=CMPLX(DEL,SLOPE*DEL)
      IF(SLOPE.GT.1) DELTA2=CMPLX(DEL/SLOPE,DEL)
      BREAK=CMPLX(1.01*REAL(CK1),.99*AIMAG(CK1))
      CALL GSHANK(HS808,CMPLX(PATH(7),PATH(8)),DELTA,ANS,6,SUM,
     .  1,BREAK,DELTA2)
      DO DSWIN2 I=1,6
DSWIN2 SUM(I)=ANS(I)

      IF(TROUBLE.NE.0) WOT   3, (8HTROUBLE=,I10) ,TROUBLE
C      WOT   3, (7HNEVALS=,I10) ,NEVALS
C      WOT 3, (3E15.5) ,(REAL(SUM(I)),AIMAG(SUM(I)),CABS(SUM(I)),I=1,6)

C
C      THIS SECTION COMBINES THE REAL AND THE IMAGINARY PARTS OF THE SIX
C      BASIC INTEGRALS IN ORDER TO FORM THE FIELD INTENSITIES.

      C1=SUM(3)
      C2=SUM(5)*CK1SQ+SUM(2)
      C3=SUM(4)

C    THIS SECTION EVALUATES THE FIELDS FOR A VERTICAL ELECTRIC DIPOLE.

      CE1R=COEE1*CK2SQ*2.*C1
      CE1Z=COEE1*CK2SQ*2.*C2
      CH1A=COEH1*CK2SQ*2.*C3


C    THIS SECTION EVALUATES THE FIELDS FOR A HORIZONTAL ELECTRIC DIPOLE

      C1=SUM(1)+SUM(6)
      IF(R.EQ.0) C2=SUM(4)+SUM(6)
      IF(R.NE.0) C2=SUM(4)/R+SUM(6)
      C3=SUM(3)
      CE1RH=COEE1*CK1SQ*2.*C1
      CE1PH=-COEE1*CK1SQ*2.*C2
      CE1ZH=-COEE1*2.*CK2SQ*C3

      ERV=CE1R
      EZV=CE1Z
      ERH=CE1RH
```

```
      EZH=CEIZH
      EPH=CEIPH

      RETURN

C       THE FOLLOWING SECTION IS USED IF Z IS .GT. 0.

  150 CONTINUE
C
C       THIS SECTION PERFORMS THE ROMBERG INTEGRATIONS FOR THE U AND
C       V INTEGRALS WITH A 12 SUBSCRIPT.

C       THE PROCEDURE FOLLOWED IS THE SAME AS DESCRIBED ABOVE FOR Z
C       LESS THAN ZERO.

C       NEVALS=0
      DO DZEROB I=1,7
DZEROB SUM(I)=0

      PATH(1)=0
      PATH(2)=.5*REAL(CK2)
      PATH(3)=.2*REAL(CK2)
      PATH(4)=PATH(2)
      PATH(5)=PATH(3)
      PATH(6)=-.5*REAL(CK2)
      PATH(7)=REAL(CK2)
      PATH(8)=PATH(6)

      DO RLOOP2 I=1,5,2
      ITYPE=1 $ A=CMPLX(PATH(I),PATH(I+1)) $ B=CMPLX(PATH(I+2),PATH(I+3))
      CALL ROMBERG(0.,1.,HSBOA,7,ANS,I0) $$$ SOURCE BELOW OBSERVER ABOVE
      IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C       WOT  3, (7HNEVALS=,I10) ,NEVALS
C       WOT 3, (//9HPATH NO. I2) ,I
C       WOT 3, (3E15.5) ,(REAL(ANS(J)),AIMAG(ANS(J)),CABS(ANS(J)),J=1,7)
      DO DR2 J=1,7
  DR2 SUM(J)=SUM(J)+ANS(J)
RLOOP2 CONTINUE


C       NEVALS=0
      DO DSWIN1B I=1,7
DSWIN1B SUM(I)=-SUM(I) $$$ SWINDLE SINCE DOING PATH BACKWARD

      SLOPE=R/(HI+ABS(ZI))
      DEL=.2*PI*(1./MAX1F(R,HI+ABS(ZI)))
      DELTA=CMPLX(-DEL,SLOPE*DEL)
      IF(ABS(SLOPE).GE.1.) DELTA=CMPLX(-DEL/SLOPE,DEL)
      CALL GSHANK(HSBOA,CMPLX(PATH(1),PATH(2)),DELTA,ANS,7,SUM,
        0,CMPLX(0.,0.),CMPLX(0.,0.))

      DO DSH2 I=1,7
  DSH2 SUM(I)=-ANS(I)   $$$ SINCE DID PATH BACKWARD

COMMENT----SLOPING PATH PASSING THRU CK1
COMMENT----THE .99 FACTOR IS TO AVOID PASSING DIRECTLY THRU CK1
      DELDEL=(1.01*REAL(CK1)-PATH(7))/(.99*AIMAG(CK1)-PATH(8))
      DELTA=CMPLX(DEL*DELDEL,DEL)
      IF(DELDEL.GT.1.) DELTA=CMPLX(DEL,DEL/DELDEL)
      DELTA2=CMPLX(DEL,SLOPE*DEL)
      IF(SLOPE.GT.1) DELTA2=CMPLX(DEL/SLOPE,DEL)
      BREAK=CMPLX(1.01*REAL(CK1),.99*AIMAG(CK1))
      CALL GSHANK(HSBOA,CMPLX(PATH(7),PATH(8)),DELTA,ANS,7,SUM,
        1,BREAK,DELTA2)
```

```
      DO DSWIN28 I=1,7
DSWIN28 SUM(I)=ANS(I)

      IF(TROUBLE.NE.0) WOT  3, (8HTROUBLE=,I10) ,TROUBLE
C      WOT  3, (7HNEVALS=,I10) ,NEVALS
C      WOT 3, (3E15.5) ,(REAL(SUM(I)),AIMAG(SUM(I)),CABS(SUM(I)),I=1,7)

      C1=SUM(3)
      C2=SUM(5)*CK2SQ+SUM(2)
      C3=SUM(4)

C    THIS SECTION EVALUATES THE FIELDS FOR A VERTICAL ELECTRIC DIPOLE.

      CE2R = COEE2*2.*C1
      CE2Z = COEE2*2.*C2
      CH2A = COEH2*2.*C3

C    THIS SECTION EVALUATES THE FIELDS FOR A HORIZONTAL ELECTRIC DIPOLE

C      THE SINGULARITY DIFFICULTY AT RHO = 0 (MENTIONED ABOVE) OCCURS
C      HERE ALSO. SIMILAR REMARKS AS ABOVE APPLY.

     C1=SUM(1)+SUM(6)
     IF(R.EQ.0) C2=SUM(4)+SUM(6)
     IF(R.NE.0) C2=SUM(4)/R+SUM(6)
     C3=SUM(7) $$$ MAKE SURE THIS IS RIGHT
      CE2RH = COEE2*2.*C1
      CE2PH = -COEE2*2.*C2
      CE2ZH = -COEE2*2.*C3
     ERV=CE2R
     EZV=CE2Z
     ERH=CE2RH
     EZH=CE2ZH
     EPH=CE2PH
     RETURN
     END
```

```
      CODE ANALYSIS
      COMPLEX FBAR
      FUNCTION FBAR(P)
       COMPLEX S,AUX,AUX1,AUX2,Q,G
       COMPLEX P,F
      COMPLEX CSQRT,CEXP
       STRUCTURE (G,G1/G2),(Q,Q1/Q2)
       STRUCTURE (S,S1/S2)
       DATA (S1=0.),(S2=1.)
       Q = P
       PM = SQRTF(Q1*Q1 + Q2*Q2)
       PP = 57.29578*ATAN2(Q1,Q2)
       IF (PM - 10.) 10,11,11
   10  CONTINUE
        AUX = 3.1415926536*Q
       AUX1=CSQRT(AUX)
       AUX2=CEXP(-Q)
       G = 1. - S*AUX1*AUX2 - 2.*Q
       AUX = 2.*Q
       A10 = -1.
       DO 12 N = 2,50
       AN = 2*N - 1
       A10 = -A10
       AUX = AUX*2.*Q/AN
       G = G + A10*AUX
   12  CONTINUE
       GO TO 13
   11  CONTINUE
       AUX = 1./(2.*Q)
       AUX1 = AUX
       G = -AUX
       DO 14 N = 2,6
       AN = 2*N - 1
       AUX1 = AUX1*AUX*AN
       G = G - AUX1
   14  CONTINUE
   13  CONTINUE
       FM = SQRTF(G1*G1 + G2*G2)
       FP = 57.29578*ATAN2(G1,G2)
       IF (FM - 1.) 40,40,41
   41  CONTINUE
       WRITE OUTPUT TAPE 3,44
   44  FORMAT (17HERROR ERROR ERROR)
   40  CONTINUE
       F = G
      FBAR=F
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE FNPLOT(A,B,FCN,K,NAMBCD)
      PARAMETER (N=25)
      COMPLEX EVAL((0,N)) $ STRUCTURE (EVAL,EVALR/EVALI)
      DIMENSION VAL((0,N))

      DO LOOP I=0,N
      VAL(I)=A+I*(B-A)/N
      DIMENSION -COMPLEX-Z(10)
      CALL FCN(VAL(I),Z)
 LOOP EVAL(I)=Z(K)

      CALL CARTMM(N,YMINR,YMAXR,EVAL,2)
      CALL FRAME
      CALL MAPX(9,A,B,YMINR,YMAXR,.11328,.999,.25,.9)
      CALL TRACE(VAL,EVAL,N+1,1,2)
      CALL SETCH(10.,32.,1,0,2,0)
      CALL CRTBCD(NAMBCD)
      CALL SETCH(1.,20.,1,0,1,1) $ CALL CRTBCD( REAL PART )

      CALL CARTMM(N,YMINI,YMAXI,EVALI,2)
      CALL FRAME
      CALL MAPX(9,A,B,YMINI,YMAXI,.11328,.999,.25,.9)
      CALL TRACE(VAL,EVALI,N+1,1,2)
      CALL SETCH(10.,32.,1,0,2,0)
      CALL CRTBCD(NAMBCD)
      CALL SETCH(1.,20.,1,0,1,1) $ CALL CRTBCD( IMAGINARY PART )

      DO D1 I=0,N
   D1 EVAL(I)=CMPLX(CABS(EVAL(I)),CANG(EVAL(I))*180./3.1415926)

      CALL CARTMM(N,YMINR,YMAXR,EVAL,2)
      CALL FRAME
      CALL MAPX(9,A,B,YMINR,YMAXR,.11328,.999,.25,.9)
      CALL TRACE(VAL,EVAL,N+1,1,2)
      CALL SETCH(10.,32.,1,0,2,0)
      CALL CRTBCD(NAMBCD)
      CALL SETCH(1.,20.,1,0,1,1) $ CALL CRTBCD( MAGNITUDE )

      CALL CARTMM(N,YMINI,YMAXI,EVALI,2)
      CALL FRAME
      CALL MAPX(9,A,B,YMINI,YMAXI,.11328,.999,.25,.9)
      CALL TRACE(VAL,EVALI,N+1,1,2)
      CALL SETCH(10.,32.,1,0,2,0)
      CALL CRTBCD(NAMBCD)
      CALL SETCH(1.,20.,1,0,1,1) $ CALL CRTBCD( PHASE )
      CALL PLOTEA
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE GEVALA(ERV,EZV,ERH,EZH,EPH) $$$ SOURCE ABOVE
      PARAMETER (PI=3.1415926535897932)
      COMMON /EVALCOM/ HI,ZI,ZMH,ZPH,R,RI,R2,-COMPLEX-CJ,
     .  CKI,CKISQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX CI,C2,C3,C4,CE2RFR,CE2ZFR
       TYPE COMPLEX CE2R20,CE2PI0,CE2ZRZ,CH2AFR
      COMPLEX G22,G22R,G22RZ,G22R2,G22Z2,G21,G21R,G21RZ,G21R2,G21Z2
      COMPLEX ERV,EZV,ERH,EZH,EPH
      COMPLEX COEE1,COEE2,COEH1,COEH2
      COMPLEX CE1R,CE1Z,CE1RH,CE1ZH,CE1PH
      COMPLEX CE2R,CE2Z,CE2RH,CE2ZH,CE2PH
      COEE1=COEE
      COEE2=COEE/CK2SQ
      COEH1=COEH*CK1SQ
      COEH2=COEH
      ERV=0
      EZV=0
      ERH=0
      EZH=0
      EPH=0
      IF(ZI.LT.0) RETURN
C       THIS IS FOR THE VERTICAL ELECTRIC DIPOLE.
      CALL GREENSAOA(G22,G22R,G22RZ,G22R2,G22Z2,
     .  G21,G21R,G21RZ,G21R2,G21Z2) $$$ SOURCE ABOVE OBSERVER ABOVE
      CE2RFR = COEE2*(G22RZ - G21RZ)
      CE2ZFR = COEE2*(G22Z2 - G21Z2 + CK2SQ*(G22 - G21))
      CH2AFR = COEH2*(G22R - G21R)
C       THIS IS FOR THE HORIZONTAL ELECTRIC DIPOLE.
       CE2R20 = COEE2*(G22R2 - G21R2+CK2SQ*(G22-G21))
C       SPECIAL CARE IS TAKEN AT THE POINT RHO = 0.
C       THE FUNCTION IS WELL BEHAVED AT THIS POINT, HOWEVER, WHEN
C       USING THE GENERAL FORMULAE IN THE NUMERICAL PROGRAM, ONE WOULD
C     OBTAIN A RESULT OF ZERO OVER ZERO FOR RHO = 0. THE ACTUAL RESULT
C       IS WELL BEHAVED, AT RHO = 0, AND THIS IS TAKEN CARE OF
C       BY TAKING THE PROPER LIMIT AT RHO = 0.
       IF(R.EQ.0) CE2PI0 = -COEE2*(G22*(-1./R2 + CJ*CK2)/R2 - G21*(-1./R1 +
     1 CJ*CK2./R1 + CK2SQ*(G22 - G21))
       IF(R.NE.0) CE2PI0 = -COEE2*((G22R - G21R)/R+CK2SQ*(G22-G21))
       CE2ZRZ = COEE2*(G22RZ + G21RZ)
C       WOT 59, (3E15.5) ,REAL(CE2RFR),REAL(CE2ZFR),REAL(CH2AFR),
C      .  AIMAG(CE2RFR),AIMAG(CE2ZFR),AIMAG(CH2AFR)
C       WOT 59, (/)
C       WOT 59, (3E15.5) ,REAL(CE2R20),REAL(CE2PI0),REAL(CE2ZRZ),
C      .  AIMAG(CE2R20),AIMAG(CE2PI0),AIMAG(CE2ZRZ)
       ERV=CE2RFR
       EZV=CE2ZFR
       ERH=CE2R20
       EZH=CE2ZRZ
       EPH=CE2PI0
       RETURN
       END
```

```
      CODE ANALYSIS
      SUBROUTINE GEVALB(ERV,EZV,ERH,EZH,EPH) $$$ SOURCE BELOW
      PARAMETER (PI=3.1415926535897932)
      COMMON /EVALCOM/ H1,Z1,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
     .  CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX C1,C2,C3,C4,CE1RFR,CE1ZFR
       TYPE COMPLEX CE1R20,CE1P10,CE1ZRZ,CHIAFR
      COMPLEX ERV,EZV,ERH,EZH,EPH
      COMPLEX G11,G11R,G11RZ,G11R2,G11Z2,G12,G12R,G12RZ,G12R2,G12Z2
      COMPLEX COEE1,COEE2,COEH1,COEH2
      COMPLEX CE1R,CE1Z,CE1RH,CE1ZH,CE1PH
      COMPLEX CE2R,CE2Z,CE2RH,CE2ZH,CE2PH

      COEE1=COEE/CK1SQ
      COEE2=COEE
      COEH1=COEH
      COEH2=COEH*CK2SQ

      ERV=0
      EZV=0
      ERH=0
      EZH=0
      EPH=0
      IF(Z1.GT.0) RETURN

C       THIS IS FOR THE VERTICAL ELECTRIC DIPOLE.
      CALL GREENSBCB(G11,G11R,G11RZ,G11R2,G11Z2,
     .  G12,G12R,G12RZ,G12R2,G12Z2) $$$ SOURCE BELOW OBSERVER BELOW
      CE1RFR = COEE1*(G11RZ - G12RZ)
      CE1ZFR = COEE1*(G11Z2 - G12Z2 + CK1SQ*(G11 - G12))
      CHIAFR = COEH1*(G11R - G12R)

C       THIS IS FOR THE HORIZONTAL ELECTRIC DIPOLE.

       CE1R20 = COEE1*(G11R2 - G12R2+CK1SQ*(G11-G12))

C       SPECIAL CARE IS TAKEN AT THE POINT RHO = 0.

C       THE FUNCTION IS WELL BEHAVED AT THIS POINT, HOWEVER, WHEN
C       USING THE GENERAL FORMULAE IN THE NUMERICAL PROGRAM, ONE WOULD
C     OBTAIN A RESULT OF ZERO OVER ZERO FOR RHO = 0. THE ACTUAL RESULT
C       IS WELL BEHAVED, AT RHO = 0, AND THIS IS TAKEN CARE OF
C       BY TAKING THE PROPER LIMIT AT RHO = 0.

      IF(R.EQ.0) CE1P10 = -COEE1*(G11*(-1./R1 + CJ*CK1)/R1 - G12*(-1./R2 +
     1 CJ*CK1)/R2 + CK1SQ*(G11 - G12))
      IF(R.NE.0) CE1P10 = -COEE1*((G11R - G12R)/R+CK1SQ*(G11-G12))
       CE1ZRZ = COEE1*(G11RZ + G12RZ)
C      WOT 59, (3E15.5) ,REAL(CE1RFR),REAL(CE1ZFR),REAL(CHIAFR),
C     .  AIMAG(CE1RFR),AIMAG(CE1ZFR),AIMAG(CHIAFR)
C      WOT 59, (/)
C      WOT 59, (3E15.5) ,REAL(CE1R10),REAL(CE1P10),REAL(CE1ZRZ),
C     .  AIMAG(CE1R10),AIMAG(CE1P10),AIMAG(CE1ZRZ)

      ERV=CE1RFR
      EZV=CE1ZFR
      ERH=CE1R20
      EZH=CE1ZRZ
      EPH=CE1P10
      RETURN
      END
```

```
COMMENT----11 DEC 74----MODIFIED TO ALWAYS PUT THE OBSERVER
COMMENT----  ON THE SURFACE OF THE OBSERVER SEGMENT
       CODE ANALYSIS
       SUBROUTINE GFIELDS(T,ETANG)
       COMMON /SFCOM/ XS,YS,ZS,AS,BS,SS,WR,XO,YO,ZO,AO,BO,IMUTUAL
COMMENT---- -S FOR SOURCE -O FOR OBSERVER
       COMPLEX EX,EY,EZ,PK,SPK,CPK
       COMPLEX ERV,EZV,ERH,EZH,EPH
       PARAMETER (PI=3.1415926535897932)
       COMMON /EVALCOM/ HI,ZI,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
      .  CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
       DIMENSION -COMPLEX- ETANG(3) $$$ 1=K TERM, 2=SIN TERM, 3=COS TERM

COMMENT---XY,YT,ZT SPECIFY POINT ALONG SOURCE SEGMENT AS FCN OF PARAMETER T
COMMENT----WHICH VARIES FROM -1. TO +1.
       SHALF=SS*.5  $$$ HALF OF SEGMENT LENGTH
       XT=XS+T*SHALF*COS(BS)*COS(AS)
       YT=YS+T*SHALF*SIN(BS)*COS(AS)
       ZT=ZS+T*SHALF*SIN(AS)
       GO TO GSELF $$$ ALWAYS OFFSET THE OBSERVER
       IF(IMUTUAL.EQ.0) GO TO GSELF
COMMENT----DO MUTUAL TERM
       R=SQRT((XT-XO)**2+(YT-YO)**2) $$$ FIND RHO OF CYL. COORD. SYSTEM
       PHI=ATAN2(YO-YT,XO-XT) $$$ ANGLE BETWEEN SOURCE AND OBS IN CYL. COORD. SYS
       GO TO G10
GSELF CONTINUE $$$ DO SELF TERM
COMMENT----GO PERPENDICULAR TO ELEMENT IN A PLANE PARALLEL TO X-Y PLANE
COMMENT----FOR THE DISTANCE WR (WIRE RADIUS)
       XOP=XO-WR*SIN(BO)
       YOP=YO+WR*COS(BO)
       R=SQRT((XOP-XT)**2+(YOP-YT)**2)
       PHI=ATAN2(YOP-YT,XOP-XT)
  G10 CONTINUE
       ZI=ZO $$$ Z OF OBSERVER IN CYL. COORD. SYS.
       HI=ABS(ZT) $$$ Z OF SOURCE IN CYL. COORD. SYS.
       ZPH=ZI+HI
       ZMH=ZI-HI
       R1=SQRT(R*R+ZPH*ZPH)
       R2=SQRT(R*R+ZMH*ZMH)
C       WOT 3, (////3H***4E15.5) ,HI,ZI,R,PHI
       IF(ZT.GE.0) CALL GEVALA(ERV,EZV,ERH,EZH,EPH)
       IF(ZT.LT.0) CALL GEVALB(ERV,EZV,ERH,EZH,EPH)
COMMENT----MODIFY CURRENT MOMENTS ACCORDING TO ORIENTATION OF SOURCE
       ERV=ERV*SIN(AS)
       EZV=EZV*[illegible](AS)
       ERH=ERH[illegible])*COS(PHI-BS)
       EZH=EZ[illegible])*COS(PHI-BS)
       EPH=EPH*COS(AS)*SIN(PHI-BS)
COMMENT----FIND COMPONENTS OF FIELD ALONG X,Y, AND Z AXES
       EX=(ERH+ERV)*COS(PHI)-EPH*SIN(PHI)
       EY=(ERH+ERV)*SIN(PHI)+EPH*COS(PHI)
       EZ=EZH+EZV
COMMENT----FIND COMPONENTS OF FIELD TANGENTIAL TO OBSERVER
       PK=CK2 $$$ VALID ONLY FOR SOURCE AND OBSERVER ABOVE GROUND
       IF(ZS.LT.0) PK=CK1  $$$ ASSUMES A SEGMENT NEVER CROSSES INTERFACE
       CALL CSINCOS(PK*T*SHALF,SPK,CPK)
       ETANG(1)=EX*COS(AO)*COS(BO)+EY*COS(AO)*SIN(BO)+EZ*SIN(AO)
       ETANG(1)=ETANG(1)*SS/2 $$$ DUE TO CNG OF VARIABLE TO PARAM. T
       ETANG(1)=CONJG(ETANG(1)) $$$ CONVERT FROM -IWT TO +JWT TIME CONVENTION
       ETANG(2)=ETANG(1)*SPK $$$ SIN INTERPOLATION TERM
C       IF(IMUTUAL.EQ.0) ETANG(2)=0 $$$ NO CONTRIBUTION TO SELF TERM
COMMENT----ABOVE CARD VALID ONLY FOR HORIZ ELEMENTS
       ETANG(3)=ETANG(1)*CPK $$$ COS INTERPOLATION TERM
       IF(DEBUG.NE.0) WOT 3, (5E15.5) ,R,R1,R2,REAL(ETANG(1)),AIMAG(ETANG(1))
```

```
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE GREENSAOA(G22,G22R,G22RZ,G22R2,G22Z2,
     .  G21,G21R,G21RZ,G21R2,G21Z2) $$$ SOURCE ABOVE OBSERVER ABOVE
      PARAMETER (PI=3.1415926535897932)
      COMMON /EVALCOM/ H1,Z1,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
     .  CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX CE1, CE2, CJ1, CJ2
      COMPLEX G22,G22R,G22RZ,G22R2,G22Z2,G21,G21R,G21RZ,G21R2,G21Z2

C       GCAL FORMS THE GREENS FUNCTIONS AND THEIR ASSOCIATED
C       DERIVATIVES WHICH ARE USED TO EVALUATE THE FIELDS.

      CJ1 = CJ*CK2*R2
      CE1=CEXP(CJ1)
      CJ2 = CJ*CK2*R1
      CE2=CEXP(CJ2)
      CALL NOTHING $$$ DUE TO COMPILER BUG
      G22 = CE1/R2
      G21 = CE2/R1
      G22R = (G22/R2)*R*(CJ*CK2 - 1./R2)
      G21R = (G21/R1)*R*(CJ*CK2 - 1./R1)
       RSQ = R*R
       R222 = R2*R2
       R223 = R222*R2
       R224 = R222*R222
       R225 = R224*R2
       R212 = R1*R1
       R213 = R212*R1
       R214 = R212*R212
       R215 = R214*R1
       G22RZ = R*ZMH*CE1*(3./R225-(3.*CJ*CK2)/R224-CK2SQ/R223)
       G21RZ = R*ZPH*CE2*(3./R215-(3.*CJ*CK2)/R214-CK2SQ/R213)
       G22Z2=CE1*((3.*ZMH**2)/R225-(3.*CJ*CK2*ZMH**2)/R224-
     1(ZMH**2*CK2SQ+1.)/R223+(CJ*CK2)/R222)
       G21Z2 = CE2*((3.*ZPH**2)/R215-(3.*CJ*CK2*ZPH**2)/R214-
     1(ZPH**2*CK2SQ+1.)/R213+(CJ*CK2)/R212)
       G22R2 = CE1*((CJ*CK2-1./R2)*(1./R222+CJ*CK2*RSQ/R223
     1   -2.*RSQ/R224) + RSQ/R225)
       G21R2 = CE2*((CJ*CK2-1./R1)*(1./R212+CJ*CK2*RSQ/R213
     1   -2.*RSQ/R214) + RSQ/R215)
       RETURN
       END
```

```
      CODE ANALYSIS
      SUBROUTINE GREENSBOB(G11,G11R,G11RZ,G11R2,G11Z2,
     . G12,G12R,G12RZ,G12R2,G12Z2) $$$ SOURCE BELOW OBSERVER BELOW
      PARAMETER (PI=3.14159265358979 32)
      COMMON /EVALCOM/ H1,Z1,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
     . CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX CE1, CE2, CJ1, CJ2
      COMPLEX G11,G11R,G11RZ,G11R2,G11Z2,G12,G12R,G12RZ,G12R2,G12Z2

C      GOAL FORMS THE GREENS FUNCTIONS AND THEIR ASSOCIATED
C      DERIVATIVES WHICH ARE USED TO EVALUATE THE FIELDS.

      CJ1 = CJ*CK1*R1
      CE1=CEXP(CJ1)
      CJ2 = CJ*CK1*R2
      CE2=CEXP(CJ2)
      CALL NOTHING $$$ DUE TO COMPILER BUG
      G11 = CE1/R1
      G12 = CE2/R2
      G11R = (G11/R1)*R*(CJ*CK1 - 1./R1)
      G12R = (G12/R2)*R*(CJ*CK1 - 1./R2)
       RSQ = R*R
       R112 = R1*R1
       R113 = R112*R1
       R114 = R112*R112
       R115 = R114*R1
       R122 = R2*R2
       R123 = R122*R2
       R124 = R122*R122
       R125 = R124*R2
       G11RZ = R*ZPH*CE1*(3./R115-(3.*CJ*CK1)/R114-CK1SQ/R113)
       G12RZ = R*ZMH*CE2*(3./R125-(3.*CJ*CK1)/R124-CK1SQ/R123)
       G11Z2=CE1*((3.*ZPH**2)/R115-(3.*CJ*CK1*ZPH**2)/R114-
     1(ZPH**2*CK1SQ+1.)/R113+(CJ*CK1)/R112)
       G12Z2 = CE2*((3.*ZMH**2)/R125-(3.*CJ*CK1*ZMH**2)/R124-
     1(ZMH**2*CK1SQ+1.)/R123+(CJ*CK1)/R122)
       G11R2 = CE1*((CJ*CK1-1./R1)*(1./R112+CJ*CK1*RSQ/R113
     1  -2.*RSQ/R114) + RSQ/R115)
       G12R2 = CE2*((CJ*CK1-1./R2)*(1./R122+CJ*CK1*RSQ/R123
     1  -2.*RSQ/R124) + RSQ/R125)
       RETURN
       END
```

```
      CODE ANALYSIS
      SUBROUTINE GSHANK(FCN,STARTER,DELTA,SUM,NANS,SEEDER,IBREAK,BREAK,DELB)
      COMPLEX DEL,START,DELTA,BREAK,DELB,STARTER
      COMPLEX QSOLVE
COMMENT----CONCRIT=CONVERGENCE CRITERION
      DATA(CONCRIT=1.E-3)
COMMENT----NSHANK=MAX NO. OF REGIONS TO BE INTEGRATED
COMMENT----MAXANS=MAX NO. OF INTEGRANDS TO BE DONE IN PARALLEL
COMMENT----NANS=ACTUAL NO. OF INTEGRANDS TO BE DONE IN PARALLEL
COMMENT----NANS MUST ALWAYS BE .LE. MAXANS............
      PARAMETER(NSHANK=30,MAXANS=10)
      DIMENSION-COMPLEX-A1(MAXANS,NSHANK)
      DIMENSION-COMPLEX-SUM(MAXANS),ANS1(MAXANS),ANS2(MAXANS),SEED(MAXANS)
      DIMENSION-COMPLEX-SEEDER(MAXANS)
COMMENT----Q ALSO DIMENSIONED IN QSOLVE
      DIMENSION-COMPLEX-Q((0,NSHANK),NSHANK/2)
COMMENT----SEED IS USED WHEN EXPRESSING THE INTEGRAL IN TWO PARTS
COMMENT----  1)LIMITS FROM 0 TO A  PLUS  2) LIMITS FROM A TO INFINITY.
COMMENT---- SEED IS THE VALUE FOR LIMITS 0 TO A.  IT IS CALCULATED OUTSIDE
COMMENT---- SHANK AND USED BY SHANK AS INPUT.
COMMENT----I MAY LATER USE SEED TO ALLOW SHANK TO CONTINUE IF IT DOESN'T
COMMENT----CONVERGE BY DEFINING A NEW SEED TO BE THE LAST PARTIAL SUM OF
COMMENT----THE OLD SEQUENCE
COMMENT------I=ROW OF SHANK MATRIX
COMMENT------J=COL OF SHANK MATRIX

      START=STARTER
      DEL=DELTA
      DO DSEED I=1,NANS
DSEED SEED(I)=SEEDER(I)

GSTART IBRKHIT=0

     NN=0
      DO D00 K=1,NANS
      Q(0,1)=SEED(K)
      DO D0 N=2,NSHANK,2
      IF(K.NE.1 .AND. N.LE.NN) GO TO NOROM
      COMMON /CONTOUR/ ITYPE-COMPLEX-A,B
      ITYPE=1
      DO DROM NROM=N-1,N
      A=START+(NROM-1)*DEL
      B=START+NROM*DEL

      IF(IBREAK.EQ.0) GO TO GSKIP
      IF(REAL(B).GE.REAL(BREAK) .AND. REAL(A).LT.REAL(BREAK)).GSKIP
      B=BREAK
      IBRKHIT=1
C      WOT 3, (  **GSHANK**BREAK IN PATH REACHED  )
GSKIP CONTINUE

      COMMON/ROMCOM/-INTEGER-NEVALS,TROUBLE
      CALL ROMBERG(C,1.,FCN,NANS,ANS1,2)
C      WOT 3, (  --GSHANK--  ,2I5,I10,4E15.5) ,K,NROM,NEVALS,
C      .  REAL(A),AIMAG(A),REAL(B),AIMAG(B)

      DO D4 L=1,NANS
   D4 A1(L,NROM)=ANS1(L)
      NN=N

      IF(IBRKHIT.EQ.0) GO TO GNOBRK
COMMENT----HIT A BREAK IN THE PATH, RESET THE SEED AND START OVER
      START=BREAK
      DEL=DELB
```

```
      DO D20 L=1,NANS
      DO D20 M=1,NROM
  D20 SEED(L)=SEED(L)+AI(L,M)
C      WOT 3, (   **GSHANK**RESTARTING PATH AT NEW SLOPE   )
      GO TO GSTART

GNOBRK CONTINUE
 DROM CONTINUE
NOROM CONTINUE
      Q(N-1,1)=AI(K,N-1)+Q(N-2,1)
      Q(N,1)=AI(K,N)+Q(N-1,1)

      DO D2 J=2,N/2
      DO D2 I=N-J,N-J+1
      Q(I,J)=QSOLVE(Q,I,J)
C      WOT 3 (2I5,2E15.5) ,I,J,REAL(Q(I,J)),AIMAG(Q(I,J))
   D2 CONTINUE
      X=REAL(Q(N/2,N/2))
      Y=REAL(Q(N/2+1,N/2))
      IF(ABS(X-Y).GT.CONCRIT*ABS(X)) GO TO D0
      X=AIMAG(Q(N/2,N/2))
      Y=AIMAG(Q(N/2+1,N/2))
      IF(ABS(X-Y).LE.CONCRIT*ABS(X)) GO TO CONVERGED
   D0 CONTINUE
      N=NSHANK
 FERR FORMAT( **GSHANK**SHANK DID NOT CONVERGE----FUNCTION NO.= ,I5)
C      WOT 59,FERR,K
      WOT 3,FERR,K
 CONVERGED CONTINUE
      SUM(K)=.5*(Q(N/2,N/2)+Q(N/2+1,N/2))
C      WOT 59, (I10,2E15.5) ,N,REAL(SUM(K)),AIMAG(SUM(K))
C      WOT 3, (I10,2E15.5) ,N,REAL(SUM(K)),AIMAG(SUM(K))
C      DO D10 J=1,N/2
C      WOT 3, (/)
C  D10 WOT 3, (4E15.5) ,(REAL(Q(I,J)),AIMAG(Q(I,J)),I=1,N)
C      WOT 3, (///)
C      WOT 3, (4E15.5) ,REAL(Q(N/2,N/2)),AIMAG(Q(N/2,N/2)),
C     .  REAL(Q(N/2+1,N/2)),AIMAG(Q(N/2+1,N/2))
C      WOT 3, (/)
C      WOT 3, (4E15.5) ,(REAL(Q(I,1)),AIMAG(Q(I,1)),I=0,N)
  D00 CONTINUE
      RETURN
      END
```

```
      CODE ANALYSIS
      OPTIMIZE
      SUBROUTINE HANKEL(Z,H0,H0P)
COMMENT----VERSION OF 24 JUL 74...OPTIMIZED FOR 7600....D LAGER

COMMENT----COMPUTES COMPLEX HANKEL FUNCTION OF THE FIRST KIND
COMMENT----  (ZEROTH ORDER) AND ITS DERIVATIVE, H0 AND H0P,
COMMENT----  FOR COMPLEX ARGUMENT, Z.
COMMENT----THE EQUATIONS USED ARE FROM THE HANDBOOK OF MATHEMATICAL FUNCTIONS
COMMENT----  EDITED BY ABRAMOWITZ AND STEGUN (1965)

COMMENT----MOST OF THE MANIPULATIONS INVOLVING CMPLX(REAL(X),AIMAG(X))
COMMENT----  ARE USED TO BYPASS INEFFICIENT CODE GENERATION BY THE
COMMENT----  CHAT COMPILER

      PARAMETER (PI=3.1415926535897932,GAMMA=.5772156649015328606)

      DIMENSION M(101)
      DIMENSION A1(25),A2(25),A3(25),A4(25)

      COMPLEX CLOG,H0,H0P,J0,J0P,P0Z,P1Z,Q0Z,Q1Z,Y0,Y0P
      COMPLEX Z,Z1,Z12,Z13,Z14,Z15,ZK,ZP,ZR,ZS,ZSQ


      IF(IPASSED),GFIRST
  G1  CONTINUE
      IF(REAL(Z)**2+AIMAG(Z)**2),ERROR $$$ IF(CABS(Z).EQ.0) ERROR
      IF((REAL(Z)**2+AIMAG(Z)**2)-100.),,LARGE $$$ IF(CABS(Z).GT.10.) GO TO LARG


COMMENT----POWER SERIES EXPANSION FOR SMALL ARGUMENT
COMMENT----  (EQS. 9.1.10 AND 9.1.11)

COMMENT----TRUNCATE MAGNITUDE OF Z**2 TO INTEGER AND USE AS INDEX IN
COMMENT----  ARRAY TO DETERMINE LIMITS ON INFINITE SERIES

      IZ=1.+(REAL(Z)**2+AIMAG(Z)**2)
      MIZ=M(IZ)

      J0=1.  $$$ ZEROTH TERM OF SERIES
      J0P=1. $$$ ZEROTH TERM OF SERIES
      Y0=0.  $$$ ZEROTH TERM OF SERIES
      Y0P=0. $$$ ZEROTH TERM OF SERIES
      ZK=1. $$$ ACCUMULATES (Z**(2*K))/(K FACTORIAL)
      ZSQ=Z*Z

      DO D11 K=1,MIZ
      ZK=ZK*CMPLX(A1(K)*REAL(ZSQ),A1(K)*AIMAG(ZSQ))
      J0=J0+CMPLX(REAL(ZK),AIMAG(ZK))
      J0P=J0P+CMPLX(A2(K)*REAL(ZK),A2(K)*AIMAG(ZK))
      Y0=Y0-CMPLX(A3(K)*REAL(ZK),A3(K)*AIMAG(ZK))
  D11 Y0P=Y0P+CMPLX(A4(K)*REAL(ZK),A4(K)*AIMAG(ZK))
      J0P=CMPLX((-.5)*REAL(Z*J0P),(-.5)*AIMAG(Z*J0P))

COMMENT----DO COMPLEX LOG OF Z/2
      CANGLE=ATAN2(AIMAG(Z),REAL(Z))
      IF(CANGLE.LE.-PI*.5) CANGLE=CANGLE+2.*PI
COMMENT----LOGF(CABS(A))=.5*LOGF(REAL(A)**2+AIMAG(A)**2)
COMMENT----  WHERE A=Z/2.
      CLOG=CMPLX(.5*LOGF(.25*(REAL(Z)**2+AIMAG(Z)**2)),CANGLE)

      Y0=J0*CMPLX((2./PI)*REAL(CLOG),(2./PI)*AIMAG(CLOG))
     .  +C2
     .  -CMPLX((1./PI)*REAL(Y0),(1./PI)*AIMAG(Y0))
```

```
       YOP=CMPLX((2./PI)*REAL(1./Z),(2./PI)*AIMAG(1./Z))
      .  +CMPLX(C1*REAL(Z),C1*AIMAG(Z))
      .  +JOP*CMPLX((2./PI)*REAL(CLOG),(2./PI)*AIMAG(CLOG))
      .  +YOP*CMPLX((.5/PI)*REAL(Z),(.5/PI)*AIMAG(Z))

       H0=J0+CMPLX(-AIMAG(Y0),REAL(Y0))
       HOP=JOP+CMPLX(-AIMAG(YOP),REAL(YOP))
       RETURN

ERROR  H0=0
       HOP=0
 FERR  FORMAT(' **HANKEL**NOT VALID FOR Z=0 ')
       WOT 59,FERR
       WOT 3,FERR
       WOT 100,FERR
       CALL EXIT
       RETURN


LARGE CONTINUE


COMMENT----ASYMPTOTIC EXPANSION (EQ. 9.2.7) FOR LARGE ARGUMENT

        ZI=CMPLX(REAL(Z)/(REAL(Z)**2+AIMAG(Z)**2)
       .  ,-AIMAG(Z)/(REAL(Z)**2+AIMAG(Z)**2))   $$$ ZI=1./Z
        ZI2=ZI*ZI
        ZI3=ZI2*ZI
        ZI4=ZI3*ZI
        ZI5=ZI4*ZI

        P0Z=1.-CMPLX(P10*REAL(ZI2),P10*AIMAG(ZI2))
       .  + CMPLX(P20*REAL(ZI4),P20*AIMAG(ZI4))
        P1Z=1.+CMPLX(P11*REAL(ZI2),P11*AIMAG(ZI2))
       .  +CMPLX((-P21)*REAL(ZI4),(-P21)*AIMAG(ZI4))

        Q0Z=CMPLX((-Q10)*REAL(ZI),(-Q10)*AIMAG(ZI))
       .  +CMPLX(Q20*REAL(ZI3),Q20*AIMAG(ZI3))
        Q1Z=CMPLX(Q11*REAL(ZI),Q11*AIMAG(ZI))
       .  +CMPLX((-Q21)*REAL(ZI3),(-Q21)*AIMAG(ZI3))

        ZP=CEXP(CMPLX(-AIMAG(Z),REAL(Z)-(PI*.25)))
        ZR=CMPLX(AIMAG(ZP),-REAL(ZP))   $$$ IS CEXP((Z-PI/4)-PI/2)
        ZS=CMPLX(C3*REAL(CSQRT(ZI)),C3*AIMAG(CSQRT(ZI)))

        H0=ZP*ZS*(P0Z+CMPLX(-AIMAG(Q0Z),REAL(Q0Z)))
        HOP=-ZR*ZS*(P1Z+CMPLX(-AIMAG(Q1Z),REAL(Q1Z)))
        RETURN
  GFIRST CONTINUE
        IPASSED=1

COMMENT----INITIALIZE PARAMETERS FOR HANKEL FUNCTION
COMMENT----THIS BLOCK OF CODE IS ENTERED ON THE FIRST CALL TO HANKEL ONLY



COMMENT----INITIALIZE CONSTANT ARRAYS TO BE USED IN POWER SERIES
COMMENT----   EXPANSIONS (EQS. 9.1.10 AND 9.1.11)

         PSI=-GAMMA
         DO D1 K=1,25
         A1(K)=-(.25/(K*K))
         A2(K)=1./(K+1.)
         PSI=PSI+(1./K)
```

```
      A3(K)=PSI+PSI  $$$ IS 2*PSI(K+1)
   D1 A4(K)=((PSI+PSI)+(1./(K+1.)))/(K+1.)  $$$ IS (PSI(K+1)+PSI(K+2))/(K+1)


COMMENT----DETERMINE HOW MANY TERMS TO USE IN POWER SERIES
COMMENT---- IS DETERMINED WHEN Z**2 / ( .25**K)*(K FACTORIAL)**2) IS .LT. 1.E-7
COMMENT----THE VALUE OF K AT WHICH THIS OCCURS IS SAVED TO BE USED
COMMENT----  AS THE UPPER LIMIT ON THE LOOP WHICH DOES THE INFINITE SERIES


      DO D2 I=1,101
COMMENT----I IS CABS(Z)**2
      TEST=1.
      DO D3 K=1,24
C DONT FORGET TO ADD *A3(K) IN FOLLOWING LINE
      TEST=TEST*(I)*ABS(A1(K))
      IF(TEST*A3(K) .LT. 1.E-7) GO TO D2
   D3 CONTINUE
   D2 M(I)=K

      WOT 3, (2013) ,(M(I),I=1,101)

COMMENT----INITIALIZE CONSTANTS TO BE USED IN POWER SERIES EXPANSION

      C1=(.5/PI)*(-GAMMA+(.5*A3(1))) $$$ C1=(  )*(PSI(1)+PSI(2))
      C2=-(1./PI)*2.*(-GAMMA) $$$ C2=-(  )*2.*(PSI(1))


COMMENT----INITIALIZE CONSTANTS USED IN ASYMPTOTIC EXPANSION
COMMENT----  (EQ. 9.2.7)
COMMENT----  USED FOR CABS(Z) .GT. 10

      C3=SQRT(2./PI)

      P10=9./128.
      P20=9.*25.*49./(64.*64.*24.)
      Q10=1./8.
      Q20=9.*25./(48.*64.)
      P11=15./128.
      P21=15.*21.*45./(64.*64.*24.)
      Q11=3./8.
      Q21=15.*21./(48.*64.)

      GO TO G1
      END
```

```
      CODE ANALYSIS
      SUBROUTINE HSAOA(T,ANS) $$$ SOURCE ABOVE OBSERVER ABOVE
      OPTIMIZE
      DIMENSION -COMPLEX- ANS(10)
      COMPLEX XL,DXL,C1,C2,H0,H0P
      COMPLEX U22,V22,V22R,V22RZ,V22Z2,V22R2
      PARAMETER (PI=3.14159265358979321)
      COMMON /EVALCOM/ H1,Z1,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
     . CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX CGAM1,CGAM2,COM,CINTN
       TYPE COMPLEX COMU
COMMENT----DXL IS DUE TO CHANGE OF VARIABLE OF INTEGRATION
      CALL LAMBDA(T,XL,DXL)
      COMPLEX CROOTS
      CGAM1=CSQRT(XL+CK1)*CROOTS(XL-CK1,CK1,XL)
      CGAM2=CSQRT(XL+CK2)*CROOTS(XL-CK2,CK2,XL)
      CINTN=CK2SQ*CGAM1+CK1SQ*CGAM2
      C2=CEXP(-CGAM2*ZPH)
      COM=DXL*XL*C2/CINTN
      COMU=DXL*XL*C2/(CGAM1+CGAM2)
      CALL HANKEL(XL*R,H0,H0P)
      H0=H0*.5  $$$ FACTOR DUE TO TRANSFORM FROM BESSEL TO HANKEL
      H0P=H0P*.5 $$$ FACTOR DUE TO TRANSFORM FROM BESSEL TO HANKEL
      U22=COMU*H0
      V22=COM*H0
       IF(R.EQ.0) V22R=-COM*XL*XL*.5
       IF(R.NE.0) V22R=COM*XL*H0P
      V22RZ=-COM*XL*CGAM2*H0P
      V22Z2=COM*CGAM2*CGAM2*H0
      V22R2=0
       IF(CABS(XL).EQ.0) GO TO G1
       IF(R.NE.0) V22R2=-COM*XL*XL*(H0P/(XL*R)+H0)
       IF(R.EQ.0) V22R2=-COM*XL*XL*.5
   G1  ANS(1)=V22R2
       ANS(2)=V22Z2
       ANS(3)=V22RZ
       ANS(4)=V22R
       ANS(5)=V22
       ANS(6)=U22
       RETURN
       END
```

```
      CODE ANALYSIS
      SUBROUTINE HSA08(T,ANS) $$$ SOURCE ABOVE OBSERVER BELOW
      OPTIMIZE
      DIMENSION -COMPLEX-ANS(10)
      COMPLEX XL,DXL,C1,C2,H0,HOP,U21,V21,V21R,V21RZ,V21Z2,V21R2,V21RH
      PARAMETER (PI=3.141592653589793Z)
      COMMON /EVALCOM/ H1,Z1,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
     . CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX CGAM1,CGAM2,COM,CINTN
       TYPE COMPLEX COMU
COMMENT----DXL IS DUE TO CHANGE OF VARIABLE OF INTEGRATION
      CALL LAMBDA(T,XL,DXL)
      COMPLEX CROOTS
      CGAM1=CSQRT(XL+CK1)*CROOTS(XL-CK1,CK1,XL)
      CGAM2=CSQRT(XL+CK2)*CROOTS(XL-CK2,CK2,XL)
      CINTN=CK2SQ*CGAM1+CK1SQ*CGAM2
      C2=CEXP(CGAM1*Z1-CGAM2*H1)
      COM=DXL*C2*XL/CINTN
      COMU=DXL*XL*C2/(CGAM1+CGAM2)
      CALL HANKEL(XL*R,H0,HOP)
      H0=H0*.5  $$$ FACTOR DUE TO TRANSFORM FROM BESSEL TO HANKEL
      HOP=HOP*.5 $$$ FACTOR DUE TO TRANSFORM FROM BESSEL TO HANKEL
      U21=COMU*H0
      V21=COM*H0
      IF(R.EQ.0) V21R=-COM*XL*XL*.5
      IF(R.NE.0) V21R=COM*XL*HOP
      V21RZ=COM*XL*CGAM1*HOP
      V21Z2=COM*CGAM1*CGAM1*H0
      V21R2=0
      IF(CABS(XL).EQ.0) GO TO G1
      IF(R.NE.0) V21R2=-COM*XL*XL*(HOP/(XL*R)+H0)
      IF(R.EQ.0) V21R2=-COM*XL*XL*.5
   G1 V21RH=-COM*XL*CGAM2*HOP
      ANS(1)=V21R2
      ANS(2)=V21Z2
      ANS(3)=V21RZ
      ANS(4)=V21R
      ANS(5)=V21
      ANS(6)=U21
      ANS(7)=V21RH
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE HSBOA(T,ANS) $$$ SOURCE BELOW OBSERVER ABOVE
      OPTIMIZE
      DIMENSION -COMPLEX-ANS(10)
      COMPLEX XL,DXL,C1,C2,H0,H0P,U12,V12,V12R,V12RZ,V12Z2,V12R2,V12RH
      PARAMETER (PI=3.14159265358979321)
      COMMON /EVALCOM/ H1,Z1,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
     .  CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX CGAM1,CGAM2,COM,CINTN
       TYPE COMPLEX COMU
COMMENT----DXL IS DUE TO CHANGE OF VARIABLE OF INTEGRATION
      CALL LAMBDA(T,XL,DXL)
      COMPLEX CROOTS
      CGAM1=CSQRT(XL+CK1)*CROOTS(XL-CK1,CK1,XL)
      CGAM2=CSQRT(XL+CK2)*CROOTS(XL-CK2,CK2,XL)
      CINTN=CK2SQ*CGAM1+CK1SQ*CGAM2
      C2=CEXP(-CGAM1*H1-CGAM2*Z1)
      COM=DXL*C2*XL/CINTN
      COMU=DXL*XL*C2/(CGAM1+CGAM2)
      CALL HANKEL(XL*R,H0,H0P)
      H0=H0*.5  $$$ FACTOR DUE TO TRANSFORM FROM BESSEL TO HANKEL
      H0P=H0P*.5 $$$ FACTOR DUE TO TRANSFORM FROM BESSEL TO HANKEL
      U12=COMU*H0
      V12=COM*H0
      IF(R.EQ.0) V12R=-COM*XL*XL*.5
      IF(R.NE.0) V12R=COM*XL*H0P
      V12RZ=-COM*XL*CGAM2*H0P
      V12Z2=COM*CGAM2*CGAM2*H0
      V12R2=0
      IF(CABS(XL).EQ.0) GO TO G1
      IF(R.NE.0) V12R2=-COM*XL*XL*(H0P/(XL*R)+H0)
      IF(R.EQ.0) V12R2=-COM*XL*XL*.5
   G1 V12RH=-COM*XL*CGAM1*H0P
      ANS(1)=V12R2
      ANS(2)=V12Z2
      ANS(3)=V12RZ
      ANS(4)=V12R
      ANS(5)=V12
      ANS(6)=U12
      ANS(7)=V12RH
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE HSBOB(T,ANS) $$$ SOURCE BELOW OBSERVER BELOW
      OPTIMIZE
      DIMENSION -COMPLEX- ANS(10)
      COMPLEX XL,DXL,C1,C2,H0,H0P
      COMPLEX U11,V11,V11R,V11RZ,V11Z2,V11R2
      PARAMETER (PI=3.1415926535897932)
      COMMON /EVALCOM/ H1,Z1,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
        CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX CGAM1,CGAM2,COM,CINTN
       TYPE COMPLEX COMU
COMMENT----DXL IS DUE TO CHANGE OF VARIABLE OF INTEGRATION
      CALL LAMBDA(T,XL,DXL)
      COMPLEX CROOTS
      CGAM1=CSQRT(XL+CK1)*CROOTS(XL-CK1,CK1,XL)
      CGAM2=CSQRT(XL+CK2)*CROOTS(XL-CK2,CK2,XL)
      CINTN=CK2SQ*CGAM1+CK1SQ*CGAM2
      C2=CEXP(+CGAM1*ZMH)
      COM=DXL*XL*C2/CINTN
      COMU=DXL*XL*C2/(CGAM1+CGAM2)
      CALL HANKEL(XL*R,H0,H0P)
      H0=H0*.5  $$$ FACTOR DUE TO TRANSFORM FROM BESSEL TO HANKEL
      H0P=H0P*.5 $$$ FACTOR DUE TO TRANSFORM FROM BESSEL TO HANKEL
      U11=COMU*H0
      V11=COM*H0
      IF(R.EQ.0) V11R=-COM*XL*XL*.5
      IF(R.NE.0) V11R=COM*XL*H0P
      V11RZ=COM*XL*CGAM1*H0P
      V11Z2=COM*CGAM1*CGAM1*H0
      V11R2=0
      IF(CABS(XL).EQ.0) GO TO G1
      IF(R.NE.0) V11R2=-COM*XL*XL*(H0P/(XL*R)+H0)
      IF(R.EQ.0) V11R2=-COM*XL*XL*.5
   G1 ANS(1)=V11R2
      ANS(2)=V11Z2
      ANS(3)=V11RZ
      ANS(4)=V11R
      ANS(5)=V11
      ANS(6)=U11
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE INFINITY(FCN,START,DELT,ANS,N,SEED,IBREAK,BREAK,DELB)
      DIMENSION-COMPLEX-ANS(10),SEED(10),ANSWR(10),SUM(10)
      COMPLEX ALIM,BLIM,START,DELTA,DELT,DELB,BREAK
      COMMON/ROMCOM/-INTEGER-NEVALS,TROUBLE
      COMMON/CONTOUR/ITYPE-COMPLEX-A,B

      DO DA I=1,N
   DA SUM(I)=SEED(I)

      DELTA=DELT
      NCON=0
      BLIM=START
      DO DINF I=1,1000
      ALIM=BLIM
      BLIM=BLIM+DELTA

      IF(IBREAK.EQ.0) GO TO GSKIP
      IF(REAL(BLIM).GE.REAL(BREAK) .AND. REAL(ALIM).LT.REAL(BREAK)),GSKIP
      BLIM=BREAK
      DELTA=DELB
      NCON=0
C      WOT 3, (   **INFINITY**BREAK IN PATH REACHED  )
GSKIP CONTINUE

      ITYPE=1 $ A=ALIM $ B=BLIM
      CALL ROMBERG(0,1.,FCN,N,ANSWR,2)

C      WOT 3, (  --INFINITY--   ,I5,I10,4E15.5) ,I,NEVALS,
C     .   REAL(A),AIMAG(A),REAL(B),AIMAG(B)

      NOCON=0
      DO D1 J=1,N
      SUM(J)=SUM(J)+ANSWR(J)
      IF(REAL(SUM(J)).EQ.0) GO TO G1
      IF(REAL(ANSWR(J))/REAL(SUM(J)).GT.1.E-3) NOCON=1
   G1 IF(AIMAG(SUM(J)).EQ.0) GO TO G2
      IF(AIMAG(ANSWR(J))/AIMAG(SUM(J)).GT.1.E-3) NOCON=1
   G2 CONTINUE
   D1 CONTINUE
      NCON=NCON+1
      IF(NOCON.EQ.1) NCON=0
      IF(NCON.GE.5) GO TO CONVERGED
  DINF CONTINUE
      WOT 3, (   **INFINITY** NO CONVERGE BY INFINITY   )
 CONVERGED CONTINUE
      DO DB I=1,N
   DB ANS(I)=SUM(I)
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE LAMBDA(T,XLAM,DXLAM)
      COMPLEX XLAM,DXLAM,ARGA,ARGB
      PARAMETER (PI=3.141592653589793)
      COMMON/CONTOUR/ITYPE,-COMPLEX-A,B
COMMENT----T IS REAL PARAMETER TO VARY BETWEEN 0. AND 1.
COMMENT----ITYPE=1 MEANS A STRAIGHT LINE BETWEEN POINTS A AND B
COMMENT----ITYPE=2 MEANS A CIRCLE CENTERED AT A STARTING AT B

      IF(ITYPE.EQ.2) GO TO G2
      XLAM=A+(B-A)*T
      DXLAM=B-A
      RETURN
   G2 R=CABS(B-A)
      PHI=CANG(B-A)
      ARGA=CMPLX(0.,PHI+2.*PI*T)
      ARGB=CEXP(ARGA)
      XLAM=A+R*ARGB
      DXLAM=CMPLX(0.,2.*PI)*R*ARGB
      RETURN
      END
```

```
COMMENT----11 DEC 74----MODIFIED TO ALWAYS PUT THE OBSERVER
COMMENT----  ON THE SURFACE OF THE OBSERVER SEGMENT
      CODE ANALYSIS
      SUBROUTINE NFIELDS(T,ETANG)
      COMMON /SFCOM/ XS,YS,ZS,AS,BS,SS,WR,XO,YO,ZO,AO,BO,IMUTUAL
COMMENT---- -S FOR SOURCE -O FOR OBSERVER
      COMPLEX EX,EY,EZ,PK,SPK,CPK
      COMPLEX ERV,EZV,ERH,EZH,EPH
      PARAMETER (PI=3.14159265358979321)
      COMMON /EVALCOM/ HI,ZI,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
        CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      DIMENSION -COMPLEX- ETANG(3) $$$ 1=K TERM, 2=SIN TERM, 3=COS TERM

COMMENT----XY,YT,ZT SPECIFY POINT ALONG SOURCE SEGMENT AS FCN OF PARAMETER T
COMMENT----WHICH VARIES FROM -1. TO +1.
      SHALF=SS*.5  $$$ HALF OF SEGMENT LENGTH
      XT=XS+T*SHALF*COS(BS)*COS(AS)
      YT=YS+T*SHALF*SIN(BS)*COS(AS)
      ZT=ZS+T*SHALF*SIN(AS)
      GO TO GSELF $$$ ALWAYS OFFSET THE OBSERVER
      IF(IMUTUAL.EQ.0) GO TO GSELF
COMMENT----DO MUTUAL TERM
      R=SQRT((XT-XO)**2+(YT-YO)**2) $$$ FIND RHO OF CYL. COORD. SYSTEM
      PHI=ATAN2(YO-YT,XO-XT) $$$ ANGLE BETWEEN SOURCE AND OBS IN CYL. COORD. SYS
      GO TO G10
GSELF CONTINUE $$$ DO SELF TERM
COMMENT----GO PERPENDICULAR TO ELEMENT IN A PLANE PARALLEL TO X-Y PLANE
COMMENT----FOR THE DISTANCE WR (WIRE RADIUS)
      XOP=XO-WR*SIN(BO)
      YOP=YO+WR*COS(BO)
      R=SQRT((XOP-XT)**2+(YOP-YT)**2)
      PHI=ATAN2(YOP-YT,XOP-XT)
  G10 CONTINUE
      ZI=ZO $$$ Z OF OBSERVER IN CYL. COORD. SYS.
      HI=ABS(ZT) $$$ Z OF SOURCE IN CYL. COORD. SYS.
      ZPH=ZI+HI
      ZMH=ZI-HI
      R1=SQRT(R*R+ZPH*ZPH)
      R2=SQRT(R*R+ZMH*ZMH)
C       WOT 3, (////3H***4E15.5) ,HI,ZI,R,PHI

       IF(ZT.LT.0) WOT 3, (   **NFIELDS** SNAFU---Z NEG FOR NORTON--   )
       IF(ZT.LT.0) WOT 59, (   **NFIELDS** SNAFU---Z NEG FOR NORTON--   )

       CALL NORTON(ERV,EZV,ERH,EZH,EPH)

COMMENT----MODIFY CURRENT MOMENTS ACCORDING TO ORIENTATION OF SOURCE
       ERV=ERV*SIN(AS)
       EZV=EZV*SIN(AS)
       ERH=ERH*COS(AS)*COS(PHI-BS)
       EZH=EZH*COS(AS)*COS(PHI-BS)
       EPH=EPH*COS(AS)*SIN(PHI-BS)
```

```
COMMENT----FIND COMPONENTS OF FIELD ALONG X,Y, AND Z AXES
      EX=(ERH+ERV)*COS(PHI)-EPH*SIN(PHI)
      EY=(ERH+ERV)*SIN(PHI)+EPH*COS(PHI)
      EZ=EZH+EZV
COMMENT----FIND COMPONENTS OF FIELD TANGENTIAL TO OBSERVER
      PK=CK2 $$$ VALID ONLY FOR SOURCE AND OBSERVER ABOVE GROUND
      IF(ZS.LT.0) PK=CK1  $$$ ASSUMES A SEGMENT NEVER CROSSES INTERFACE
      CALL CSINCOS(PK*T*SHALF,SPK,CPK)
      ETANG(1)=EX*COS(A0)*COS(B0)+EY*COS(A0)*SIN(B0)+EZ*SIN(A0)
      ETANG(1)=ETANG(1)*SS/2 $$$ DUE TO CNG OF VARIABLE TO PARAM. T
      ETANG(1)=CONJG(ETANG(1)) $$$ CONVERT FROM -IWT TO +JWT TIME CONVENTION
      ETANG(2)=ETANG(1)*SPK $$$ SIN INTERPOLATION TERM
C      IF(IMUTUAL.EQ.0) ETANG(2)=0 $$$ NO CONTRIBUTION TO SELF TERM
COMMENT----ABOVE CARD VALID ONLY FOR HORIZ ELEMENTS
      ETANG(3)=ETANG(1)*CPK $$$ COS INTERPOLATION TERM
      IF(DEBUG.NE.0) WOT 3, (5E15.5) ,R,R1,R2,REAL(ETANG(1)),AIMAG(ETANG(1))
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE NORTON(ERV,EZV,ERH,EZH,EPH)
      PARAMETER (PI=3.1415926535798932,MU0=4.E-7*PI,E0=8.854187B2E-12,
     .  C=2.99792458E8)

      COMPLEX ERV,EZV,ERH,EZH,EPH
      COMPLEX RE,RH,FBAR
      COMPLEX FACTOR,JBRD,JBRR,EJBRD,EJBRR,XN,RNH,RNE,PE,PM
      COMPLEX TERM1,TERM2,TERM3,TERM4,TERM5,TERM6,TERM7
      COMPLEX RETR,RHTR
      COMPLEX TEMP1,TEMP2

      COMMON/CXNN/-COMPLEX-XNN  $$$ USED BY FUNCTION RH


COMMENT----SUBROUTINE SETUP MUST BE CALL BEFORE USING
COMMENT----  NORTON.  SETUP INITIALIZES F,ER, AND SIG

      COMMON/NORTSETUP/F,ER,SIG

COMMENT----VARIABLES IN EVALCOM MUST ALSO BE
COMMENT----  INITIALIZED BEFORE CALLING NORTON.  THEY ARE HI,ZI,R

      COMMON/EVALCOM/HI,ZI,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
     .   CK1,CK1SQ,CK2,CK2SQ,COEE,COEH

      IF(ICALLED.EQ.0) WOT 3, (    NORTON CALLED   )
      ICALLED=1



      H=HI
      Z=ZI
      RHO=R


      W=2.*PI*F
      B=W/C
      RD=SQRT(RHO*RHO+(Z-H)**2)
      RR=SQRT(RHO*RHO+(Z+H)**2)
      STD=RHO/RD $$$ SIN THETA DIRECT
      CTD=(Z-H)/RD $$$ COS THETA DIRECT
      STR=RHO/RR  $$$ SIN THETA REFLECTED
      CTR=(Z+H)/RR  $$$ COS THETA REFLECTED
      TR=ASIN(STR) $$$ THETA REFLECTED

      FACTOR=CMPLX(0.,W*MU0*.25/PI)
      JBRD=CMPLX(0.,B*RD)
      JBRR=CMPLX(0.,B*RR)
      EJBRD=CEXP(-JBRD)/RD
      EJBRR=CEXP(-JBRR)/RR


      XN=CSQRT(CMPLX(ER,-SIG/(W*E0)))
      XNN=XN*XN  $$$ FOR USE IN FUNCTION RH
      RETR=RE(TR)
      RHTR=RH(TR)
      RNH=(XN-1.)/(XN+1.)  $$$ NORMAL REFLECTION COEFF.
      RNH=1.  $$$ LET'S NOT AND SAY WE DID
      RNE=-RNH

      PE=(CMPLX(0.,-B*RR)/(2.*STR*STR))*(CTR+CSQRT(XNN-STR*STR)/XNN)**2
      PM=(CMPLX(0.,-B*RR)/(2.*STR*STR))*(CTR+CSQRT(XNN-STR*STR))**2

```

```
      TERM1=STD*STD*EJBRD
      TERM2=RH(TR)*STR*STR*EJBRR
      TERM3=(1.-RH(TR))*FBAR(PE)*STR*STR*EJBRR
      TERM4=2.*(CSQRT(XN*XN-STR*STR)/(XN*XN))*CTR*EJBRR/JBRR
      TERM5=EJBRD*((1./JBRD)+(1./JBRD)**2)*(1.-3.*CTD*CTD)
      TERM6=RNH*EJBRR*((1./JBRR)+(1./JBRR)**2)*(1.-3.*CTR*CTR)

C      WOT 3, (  TERM1=  ,4E15.5) ,REAL(TERM1),AIMAG(TERM1),
C     .  CABS(TERM1),(180./PI)*CANG(TERM1)
C      WOT 3, (  TERM2=  ,4E15.5) ,REAL(TERM2),AIMAG(TERM2),
C     .  CABS(TERM2),(180./PI)*CANG(TERM2)
C      WOT 3, (  TERM3=  ,4E15.5) ,REAL(TERM3),AIMAG(TERM3),
C     .  CABS(TERM3),(180./PI)*CANG(TERM3)
C      WOT 3, (  TERM4=  ,4E15.5) ,REAL(TERM4),AIMAG(TERM4),
C     .  CABS(TERM4),(180./PI)*CANG(TERM4)
C      WOT 3, (  TERM5=  ,4E15.5) ,REAL(TERM5),AIMAG(TERM5),
C     .  CABS(TERM5),(180./PI)*CANG(TERM5)
C      WOT 3, (  TERM6=  ,4E15.5) ,REAL(TERM6),AIMAG(TERM6),
C     .  CABS(TERM6),(180./PI)*CANG(TERM6)
C      WOT 3, (  EZV=  ,4E15.5) ,REAL(EZV),AIMAG(EZV),
C     .  CABS(EZV),(180./PI)*CANG(EZV)


      EZV=-FACTOR*(TERM1+TERM2+TERM3+TERM4+TERM5+TERM6)

      TERM1=CTD*CTD*EJBRD
      TERM2=RHTR*CTR*CTR*EJBRR
      TERM3=(1.-RHTR)*FBAR(PE)*((XNN-STR*STR)/(XNN*XNN))*EJBRR
      TERM4=((1./JBRD)+(1./JBRD)**2)*(1.-3.*STD*STD)*EJBRD
      TEMP1=((1./JBRR)+(1./JBRR)**2)*(1.-3.*STR*STR)*EJBRR
      TERM5=RNH*TEMP1
      TERM6=(1./XNN)*TEMP1*((1.+RHTR)+(1.+RHTR)*FBAR(PE))
      TEMP2=(FBAR(PE)*(((XNN-STR*STR)/(XNN*XNN))-CTR*CTR+(1./JBRR)))-(1./JBRR)
      TERM7=(1./XNN)*STR*STR*(1.-RHTR)*(1.+(1./JBRR))*TEMP2*EJBRR

C      WOT 3, (  TERM1=  ,4E15.5) ,REAL(TERM1),AIMAG(TERM1),
C     .  CABS(TERM1),(180./PI)*CANG(TERM1)
C      WOT 3, (  TERM2=  ,4E15.5) ,REAL(TERM2),AIMAG(TERM2),
C     .  CABS(TERM2),(180./PI)*CANG(TERM2)
C      WOT 3, (  TERM3=  ,4E15.5) ,REAL(TERM3),AIMAG(TERM3),
C     .  CABS(TERM3),(180./PI)*CANG(TERM3)
C      WOT 3, (  TERM4=  ,4E15.5) ,REAL(TERM4),AIMAG(TERM4),
C     .  CABS(TERM4),(180./PI)*CANG(TERM4)
C      WOT 3, (  TERM5=  ,4E15.5) ,REAL(TERM5),AIMAG(TERM5),
C     .  CABS(TERM5),(180./PI)*CANG(TERM5)
C      WOT 3, (  TERM6=  ,4E15.5) ,REAL(TERM6),AIMAG(TERM6),
C     .  CABS(TERM6),(180./PI)*CANG(TERM6)
C      WOT 3, (  TERM7=  ,4E15.5) ,REAL(TERM7),AIMAG(TERM7),
C     .  CABS(TERM7),(180./PI)*CANG(TERM7)
C      WOT 3, (  ERH=  ,4E15.5) ,REAL(ERH),AIMAG(ERH),
C     .  CABS(ERH),(180./PI)*CANG(ERH)

      ERH=-FACTOR*(TERM1-TERM2-TERM3+TERM4-TERM5+TERM6+TERM7)


      TERM1=EJBRD
      TERM2=RETR*EJBRR
      TERM3=(1.-RETR)*FBAR(PM)*EJBRR
      TERM4=(1.+(1./JBRD))*EJBRD/JBRD
      TERM5=RNE*(1.+(1./JBRR))*EJBRR/JBRR
      TEMP1=(-1.-.5*CTR*CTR+((XNN-STR*STR)/(2.*XNN*XNN))-.5*(1./JBRR))
      TERM6=(1.-RHTR)*(FBAR(PE)/XNN)*TEMP1*EJBRR/JBRR
      TERM7=(1./XNN)*((1.+RHTR)+1.5*(1./JBRR)+.5*RHTR*(1./JBRR))*EJBRR/JBRR
```

```
C       WOT 3, (  TERM1=   ,4E15.5) ,REAL(TERM1),AIMAG(TERM1),
C     .   CABS(TERM1),(180./PI)*CANG(TERM1)
C       WOT 3, (  TERM2=   ,4E15.5) ,REAL(TERM2),AIMAG(TERM2),
C     .   CABS(TERM2),(180./PI)*CANG(TERM2)
C       WOT 3, (  TERM3=   ,4E15.5) ,REAL(TERM3),AIMAG(TERM3),
C     .   CABS(TERM3),(180./PI)*CANG(TERM3)
C       WOT 3, (  TERM4=   ,4E15.5) ,REAL(TERM4),AIMAG(TERM4),
C     .   CABS(TERM4),(180./PI)*CANG(TERM4)
C       WOT 3, (  TERM5=   ,4E15.5) ,REAL(TERM5),AIMAG(TERM5),
C     .   CABS(TERM5),(180./PI)*CANG(TERM5)
C       WOT 3, (  TERM6=   ,4E15.5) ,REAL(TERM6),AIMAG(TERM6),
C     .   CABS(TERM6),(180./PI)*CANG(TERM6)
C       WOT 3, (  TERM7=   ,4E15.5) ,REAL(TERM7),AIMAG(TERM7),
C     .   CABS(TERM7),(180./PI)*CANG(TERM7)
C       WOT 3, (  EPH=   ,4E15.5) ,REAL(EPH),AIMAG(EPH),
C     .   CABS(EPH),(180./PI)*CANG(EPH)

        EPH=FACTOR*(TERM1+TERM2+TERM3+TERM4+TERM5-TERM6+TERM7)

        TERM1=CTD*STD*EJBRD
        TERM2=RHTR*CTR*STR*EJBRR
        TERM3=(1.-RHTR)*FBAR(PE)*STR*(CSQRT(XNN-STR*STR)/XNN)*EJBRR
        TERM4=(1.-RHTR)*CTR*STR*EJBRR/JBRR
        TERM5=(1.-RHTR)*STR*(CSQRT(XNN-STR*STR)/XNN)*.5*EJBRR/JBRR
        TERM6=3.*CTD*STD*((1./JBRD)+(1./JBRD)**2)*EJBRD
LABEL   CONTINUE $$$ TO BREAK UP BLOCK
        TERM7=RNH*3.*CTR*STR*((1./JBRR)+(1./JBRR)**2)*EJBRR

C       WOT 3, (  TERM1=   ,4E15.5) ,REAL(TERM1),AIMAG(TERM1),
C     .   CABS(TERM1),(180./PI)*CANG(TERM1)
C       WOT 3, (  TERM2=   ,4E15.5) ,REAL(TERM2),AIMAG(TERM2),
C     .   CABS(TERM2),(180./PI)*CANG(TERM2)
C       WOT 3, (  TERM3=   ,4E15.5) ,REAL(TERM3),AIMAG(TERM3),
C     .   CABS(TERM3),(180./PI)*CANG(TERM3)
C       WOT 3, (  TERM4=   ,4E15.5) ,REAL(TERM4),AIMAG(TERM4),
C     .   CABS(TERM4),(180./PI)*CANG(TERM4)
C       WOT 3, (  TERM5=   ,4E15.5) ,REAL(TERM5),AIMAG(TERM5),
C     .   CABS(TERM5),(180./PI)*CANG(TERM5)
C       WOT 3, (  TERM6=   ,4E15.5) ,REAL(TERM6),AIMAG(TERM6),
C     .   CABS(TERM6),(180./PI)*CANG(TERM6)
C       WOT 3, (  TERM7=   ,4E15.5) ,REAL(TERM7),AIMAG(TERM7),
C     .   CABS(TERM7),(180./PI)*CANG(TERM7)
C       WOT 3, (  ERV=   ,4E15.5) ,REAL(ERV),AIMAG(ERV),
C     .   CABS(ERV),(180./PI)*CANG(ERV)
C       WOT 3, (  EZH=   ,4E15.5) ,REAL(EZH),AIMAG(EZH),
C     .   CABS(EZH),(180./PI)*CANG(EZH)

        ERV=FACTOR*(TERM1+TERM2-TERM3-TERM4-TERM5+TERM6+TERM7)

        EZH=FACTOR*(TERM1-TERM2+TERM3+TERM4+TERM5+TERM6-TERM7)

        ERV=CONJG(ERV) $$$ TO GO FROM +JWT TO -IWT
        EZV=CONJG(EZV) $$$ TO GO FROM +JWT TO -IWT
        ERH=CONJG(ERH) $$$ TO GO FROM +JWT TO -IWT
        EZH=CONJG(EZH) $$$ TO GO FROM +JWT TO -IWT
        EPH=CONJG(EPH) $$$ TO GO FROM +JWT TO -IWT

        RETURN
        END
```

```
      CODE ANALYSIS
      SUBROUTINE NOTHING
      RETURN
      END
```

```
      CODE ANALYSIS
      COMPLEX QSOLVE
      FUNCTION QSOLVE(Q,I,J)
COMMENT----Q ALSO DIMENSIONED IN GSHANK
      PARAMETER (NSHANK=30)
      DIMENSION-COMPLEX-Q((0,NSHANK),NSHANK/2)
COMMENT----Q ALSO DIMENSIONED IN GSHANK
      QSOLVE=Q(I+1,J-1)*Q(I-1,J-1)-Q(I,J-1)**2
      QSOLVE=QSOLVE/(Q(I+1,J-1)+Q(I-1,J-1)-2.*Q(I,J-1))
      RETURN
      END
```

```
      CODE ANALYSIS
      COMPLEX RE
      FUNCTION RE(T)
      COMPLEX CSQRT,CEXP
      COMMON /CXNN/ -COMPLEX-XNN
      COMPLEX T1,T2
      T1=COS(T) $ T2=CSQRT(XNN-SIN(T)**2)
      RE=(T1-T2)/(T1+T2)
      RETURN
      END
```

```
      CODE ANALYSIS
      COMPLEX RH
      FUNCTION RH(T)
      COMPLEX CSQRT,CEXP
      COMMON /CXNN/ -COMPLEX-XNN
      COMPLEX T1,T2
      T1=XNN*COS(T) $ T2=CSQRT(XNN-SIN(T)**2)
      RH=(T1-T2)/(T1+T2)
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE ROM(A,B,FCN,N,SUM,RX)
COMMENT----THIS VERSION PARAMETERIZES LINE BETWEEN A AND B AS
COMMENT----FUNCTION OF PARAMETER ZP.   THIS ALLOWS A TO BE .GT. B OR .LT. B
COMMENT----AND TO BE POSITIVE OR NEGATIVE
      COMPLEX TE1,TE2
      DIMENSION -COMPLEX-SUM(10),G1(10),G2(10),G3(10),G4(10),G5(10),
     .   T00(10),T01(10),T10(10),T11(10),T02(10),T20(10)
      COMPLEX TESTC
      COMMON/RCOM/-INTEGER-NEVALS,TROUBLE
      DATA(NX=2),(NM=65536),(NTS=4)
C       DATA (NX=1) $$$ FORCE 5 EVALS ONLY
      DAB=B-A
      EL1=0
      EL2=1.
      Z=EL1
      ZE=EL2
      S=ZE-Z
      EP=S/(10.*NM)
      ZEND=ZE-EP
      DO D1 I=1,N
   D1 SUM(I)=0
      NS=NX
      NT=0
      CALL FCN(A,G1)
      NEVALS=NEVALS+1
    1 DZ=S/NS
      DZOT=DZ*.5
      ZP=Z+DZ
      IF(ZP.LE.ZE) GO TO 4
      DZ=ZE-Z
      IF(ABS(DZ).LE.EP) GO TO 100
    4 DZOT=DZ*.5
      ZP=Z+DZOT
      CALL FCN(A+DAB*ZP,G3)
      ZP=Z+DZ
      CALL FCN(A+DAB*ZP,G5)
      NEVALS=NEVALS+2
   23 NOCON=0
      DO D2 I=1,N
      T00(I)=(G1(I)+G5(I))*DZOT*DAB
      T01(I)=(T00(I)+DZ*DAB*G3(I))*.5
      T10(I)=(4.*T01(I)-T00(I))/3.
      TE1=TESTC(T01(I),T10(I))
      IF(REAL(TE1).GT.RX .OR. AIMAG(TE1).GT.RX) NOCON=1
   D2 CONTINUE
C       NOCON=1 $$$ FORCE 5 EVALS ONLY
      IF(NOCON.EQ.0) GO TO 50
      ZP=Z+DZ*.25
      CALL FCN(A+DAB*ZP,G2)
      ZP=Z+DZ*.75
      CALL FCN(A+DAB*ZP,G4)
      NEVALS=NEVALS+2
      DO D4 I=1,N
      T02(I)=(T01(I)+DZOT*DAB*(G2(I)+G4(I)))*.5
      T11(I)=(4.*T02(I)-T01(I))/3.
   D4 T20(I)=(16.*T11(I)-T10(I))/15.
      NOGO=0
      DO D5 I=1,N
      TE2=TESTC(T11(I),T20(I))
      IF(AIMAG(TE2).GT.RX) NOGO=1RI
      IF(REAL(TE2).GT.RX) NOGO=NOGO+2RR
C       NOGO=0 $$$ FORCE 5 EVALS ONLY
      IF(NOGO.NE.0) GO TO 21
```

```
   D5 CONTINUE
   51 DO D6 I=1,N
   D6 SUM(I)=SUM(I)+T20(I)
      NT=NT+1
   52 Z=Z+DZ
      IF(Z.GE.ZEND) GO TO 100
      DO D9 I=1,N
   D9 G1(I)=G5(I)
      IF(NT.LT.NTS .OR. NS.LE.NX) GO TO 1
      NS=NS/2
      NT=1
      GO TO 1
   50 DO D7 I=1,N
   D7 SUM(I)=SUM(I)+T10(I)
      NT=NT+2
      GO TO 52
   21 NT=0
      IF(NS.LT.NM) GO TO 22
   93 FORMAT( **ROM** STEP SIZE LIMITED AT Z= ,F10.5,   NOGO= ,R2,
     .       FCN NO.= ,I2)
      WOT 3,93,Z,NOGO,I
      WOT 3, (6E15.5) ,REAL(TE2),REAL(G1(I)),REAL(G2(I)),
     .   REAL(G3(I)),REAL(G4(I)),REAL(G5(I)),
     .   AIMAG(TE2),AIMAG(G1(I)),AIMAG(G2(I)),AIMAG(G3(I)),
     .   AIMAG(G4(I)),AIMAG(G5(I))
      XLIM1=A+DAB*(Z-0.*DZ)
      XLIM2=A+DAB*(Z+1.*DZ)
      CALL FNPLOT(XLIM1,XLIM2,FCN,I, **ROM** STEP SIZE LIMITED )
      TROUBLE=TROUBLE+1
      IF(TROUBLE.EQ.1) WOT 59, (  **ROM** STEP SIZE LIMITED  )
      GO TO 51
   22 NS=NS*2
      DZ=S/NS
      DZOT=DZ*.5
      DO D8 I=1,N
      G5(I)=G3(I)
   D8 G3(I)=G2(I)
      GO TO 23
  100 CONTINUE
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE ROMBERG(A,B,FCN,N,SUM,NX)
COMMENT----THIS VERSION PARAMETERIZES LINE BETWEEN A AND B AS
COMMENT----FUNCTION OF PARAMETER ZP.   THIS ALLOWS A TO BE .GT. B OR .LT. B
COMMENT----AND TO BE POSITIVE OR NEGATIVE
      COMPLEX TE1,TE2
      DIMENSION -COMPLEX-SUM(10),G1(10),G2(10),G3(10),G4(10),G5(10),
     .   T00(10),T01(10),T10(10),T11(10),T02(10),T20(10)
      COMPLEX TESTC
      COMMON/ROMCOM/-INTEGER-NEVALS,TROUBLE
      DATA(NM=65536),(NTS=4),(RX=1.E-4)
      DAB=B-A
      EL1=0
      EL2=1.
      Z=EL1
      ZE=EL2
      S=ZE-Z
      EP=S/(10.*NM)
      ZEND=ZE-EP
      DO D1 I=1,N
   D1 SUM(I)=0
      NS=NX
      NT=0
      CALL FCN(A,G1)
      NEVALS=NEVALS+1
    1 DZ=S/NS
      DZOT=DZ*.5
      ZP=Z+DZ
      IF(ZP.LE.ZE) GO TO 4
      DZ=ZE-Z
      IF(ABS(DZ).LE.EP) GO TO 100
    4 DZOT=DZ*.5
      ZP=Z+DZOT
      CALL FCN(A+DAB*ZP,G3)
      ZP=Z+DZ
      CALL FCN(A+DAB*ZP,G5)
      NEVALS=NEVALS+2
   23 NOCON=0
      DO D2 I=1,N
      T00(I)=(G1(I)+G5(I))*DZOT*DAB
      T01(I)=(T00(I)+DZ*DAB*G3(I))*.5
      T10(I)=(4.*T01(I)-T00(I))/3.
      TE1=TESTC(T01(I),T10(I))
      IF(REAL(TE1).GT.RX .OR. AIMAG(TE1).GT.RX) NOCON=1
   D2 CONTINUE
      IF(NOCON.EQ.0) GO TO 50
      ZP=Z+DZ*.25
      CALL FCN(A+DAB*ZP,G2)
      ZP=Z+DZ*.75
      CALL FCN(A+DAB*ZP,G4)
      NEVALS=NEVALS+2
      DO D4 I=1,N
      T02(I)=(T01(I)+DZOT*DAB*(G2(I)+G4(I)))*.5
      T11(I)=(4.*T02(I)-T01(I))/3.
   D4 T20(I)=(16.*T11(I)-T10(I))/15.
      NOGO=0
      DO D5 I=1,N
      TE2=TESTC(T11(I),T20(I))
      IF(AIMAG(TE2).GT.RX) NOGO=1RI
      IF(REAL(TE2).GT.RX) NOGO=NOGO+2RR
      IF(NOGO.NE.0) GO TO 21
   D5 CONTINUE
   51 DO D6 I=1,N
   D6 SUM(I)=SUM(I)+T20(I)
```

```
      NT=NT+1
   52 Z=Z+DZ
      IF(Z.GE.ZEND) GO TO 100
      DO D9 I=1,N
   D9 G1(I)=G5(I)
      IF(NT.LT.NTS .OR. NS.LE.NX) GO TO 1
      NS=NS/2
      NT=1
      GO TO 1
   50 DO D7 I=1,N
   D7 SUM(I)=SUM(I)+T10(I)
      NT=NT+2
      GO TO 52
   21 NT=0
      IF(NS.LT.NM) GO TO 22
   93 FORMAT( **ROMBERG** STEP SIZE LIMITED AT Z= ,F10.5,   NOGO= ,R2,
     .       FCN NO.= ,I2)
      WOT 3,93,Z,NOGO,I
      WOT 3, (6E15.5) ,REAL(TE2),REAL(G1(I)),REAL(G2(I)),
     .   REAL(G3(I)),REAL(G4(I)),REAL(G5(I)),
     .   AIMAG(TE2),AIMAG(G1(I)),AIMAG(G2(I)),AIMAG(G3(I)),
     .   AIMAG(G4(I) ,AIMAG(G5(I))

      COMPLEX XL,DXL
      CALL LAMBDA(Z,XL,DXL)
      WOT 3, (5X,  LAMBDA IS  ,2E15.5) ,REAL(XL),AIMAG(XL)

      XLIM1=A+DAB*(Z)
      XLIM2=A+DAB*(Z+DZ)
      CALL FNPLOT(XLIM1,XLIM2,FCN,I, **ROMBERG** STEP SIZE LIMITED )
      TROUBLE=TROUBLE+1
      IF(TROUBLE.EQ.1) WOT 59, (29H**ROMBERG** STEP SIZE LIMITED)
      GO TO 51
   22 NS=NS*2
      DZ=S/NS
      DZOT=DZ*.5
      DO D8 I=1,N
      G5(I)=G3(I)
   D8 G3(I)=G2(I)
      GO TO 23
  100 CONTINUE
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE SAOA(T,ANS) $$$ SOURCE ABOVE OBSERVER ABOVE
      OPTIMIZE
      DIMENSION -COMPLEX- ANS(10)
      COMPLEX XL,DXL,C1,C2,B0,B0P
      COMPLEX U22,V22,V22R,V22RZ,V22Z2,V22R2
      PARAMETER (PI=3.14159265358979 32)
      COMMON /EVALCOM/ H1,Z1,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
     . CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX CGAM1,CGAM2,COM,CINTN
       TYPE COMPLEX COMU
COMMENT----DXL IS DUE TO CHANGE OF VARIABLE OF INTEGRATION
      CALL LAMBDA(T,XL,DXL)
      CGAM1=CSQRT(XL*XL-CK1SQ)
      CGAM2=CSQRT(XL*XL-CK2SQ)
      IF(REAL(CGAM1).EQ.0) CGAM1=CMPLX(0.,-ABS(AIMAG(CGAM1)))
      IF(REAL(CGAM2).EQ.0) CGAM2=CMPLX(0.,-ABS(AIMAG(CGAM2)))
      CINTN=CK2SQ*CGAM1+CK1SQ*CGAM2
      C2=CEXP(-CGAM2*ZPH)
      COM=DXL*XL*C2/CINTN
      COMU=DXL*XL*C2/(CGAM1+CGAM2)
      CALL BESSEL(XL*R,B0,B0P)
      U22=COMU*B0
      V22=COM*B0
      V22R=0
      IF(R.NE.0) V22R=COM*XL*B0P
      V22RZ=-COM*XL*CGAM2*B0P
      V22Z2=COM*CGAM2*CGAM2*B0
      V22R2=0
      IF(CABS(XL).EQ.0) GO TO G1
      IF(R.NE.0) V22R2=-COM*XL*XL*(B0P/(XL*R)+B0)
      IF(R.EQ.0) V22R2=-COM*XL*XL*.5
  G1  ANS(1)=V22R2
      ANS(2)=V22Z2
      ANS(3)=V22RZ
      ANS(4)=V22R
      ANS(5)=V22
      ANS(6)=U22
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE SA03(T,ANS) $$$ SOURCE ABOVE OBSERVER BELOW
      OPTIMIZE
      DIMENSION -COMPLEX-ANS(10)
      COMPLEX XL,DXL,C1,C2,B0,B0P,U21,V21,V21R,V21RZ,V21Z2,V21R2,V21RH
      PARAMETER (PI=3.1415926535897932)
      COMMON /EVALCOM/ H1,Z1,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
     .  CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX CGAM1,CGAM2,COM,CINTN
       TYPE COMPLEX COMU
COMMENT----DXL IS DUE TO CHANGE OF VARIABLE OF INTEGRATION
      CALL LAMBDA(T,XL,DXL)
      CGAM1=CSQRT(XL*XL-CK1SQ)
      IF(REAL(CGAM1).EQ.0) CGAM1=CMPLX(0.,-ABS(AIMAG(CGAM1)))
      CGAM2=CSQRT(XL*XL-CK2SQ)
      IF(REAL(CGAM2).EQ.0) CGAM2=CMPLX(0.,-ABS(AIMAG(CGAM2)))
      CINTN=CK2SQ*CGAM1+CK1SQ*CGAM2
      C2=CEXP(CGAM1*Z1-CGAM2*H1)
      COM=DXL*C2*XL/CINTN
      COMU=DXL*XL*C2/(CGAM1+CGAM2)
      CALL BESSEL(XL*R,B0,B0P)
      U21=COMU*B0
      V21=COM*B0
      V21R=0
      IF(R.NE.0) V21R=COM*XL*B0P
      V21RZ=COM*XL*CGAM1*B0P
      V21Z2=COM*CGAM1*CGAM1*B0
      V21R2=0
      IF(CABS(XL).EQ.0) GO TO G1
      IF(R.NE.0) V21R2=-COM*XL*XL*(B0P/(XL*R)+B0)
      IF(R.EQ.0) V21R2=-COM*XL*XL*.5
   G1 V21RH=-COM*XL*CGAM2*B0P
      ANS(1)=V21R2
      ANS(2)=V21Z2
      ANS(3)=V21RZ
      ANS(4)=V21R
      ANS(5)=V21
      ANS(6)=U21
      ANS(7)=V21RH
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE SBOA(T,ANS) $$$ SOURCE BELOW OBSERVER ABOVE
      OPTIMIZE
      DIMENSION -COMPLEX-ANS(10)
      COMPLEX XL,DXL,C1,C2,B0,B0P,U12,V12,V12R,V12RZ,V12Z2,V12R2,V12RH
      PARAMETER (PI=3.1415926535897932)
      COMMON /EVALCOM/ H1,Z1,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
     .  CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX CGAM1,CGAM2,COM,CINTN
       TYPE COMPLEX COMU
COMMENT----DXL IS DUE TO CHANGE OF VARIABLE OF INTEGRATION
      CALL LAMBDA(T,XL,DXL)
      CGAM1=CSQRT(XL*XL-CK1SQ)
      IF(REAL(CGAM1).EQ.0) CGAM1=CMPLX(0.,-ABS(AIMAG(CGAM1)))
      CGAM2=CSQRT(XL*XL-CK2SQ)
      IF(REAL(CGAM2).EQ.0) CGAM2=CMPLX(0.,-ABS(AIMAG(CGAM2)))
      CINTN=CK2SQ*CGAM1+CK1SQ*CGAM2
      C2=CEXP(-CGAM1*H1-CGAM2*Z1)
      COM=DXL*C2*XL/CINTN
      COMU=DXL*XL*C2/(CGAM1+CGAM2)
      CALL BESSEL(XL*R,B0,B0P)
      U12=COMU*B0
      V12=COM*B0
      V12R=0
      IF(R.NE.0) V12R=COM*XL*B0P
      V12RZ=-COM*XL*CGAM2*B0P
      V12Z2=COM*CGAM2*CGAM2*B0
      V12R2=0
      IF(CABS(XL).EQ.0) GO TO G1
      IF(R.NE.0) V12R2=-COM*XL*XL*(B0P/(XL*R)+B0)
      IF(R.EQ.0) V12R2=-COM*XL*XL*.5
   G1 V12RH=-COM*XL*CGAM1*B0P
      ANS(1)=V12R2
      ANS(2)=V12Z2
      ANS(3)=V12RZ
      ANS(4)=V12R
      ANS(5)=V12
      ANS(6)=U12
      ANS(7)=V12RH
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE SBOB(T,ANS) $$$ SOURCE BELOW OBSERVER BELOW
      OPTIMIZE
      DIMENSION -COMPLEX- ANS(10)
      COMPLEX XL,DXL,C1,C2,B0,B0P
      COMPLEX U11,V11,V11R,V11RZ,V11Z2,V11R2
      PARAMETER (PI=3.1415926535897932)
      COMMON /EVALCOM/ H1,Z1,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
     .  CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      TYPE COMPLEX CGAM1,CGAM2,COM,CINTN
       TYPE COMPLEX COMU
COMMENT----DXL IS DUE TO CHANGE OF VARIABLE OF INTEGRATION
      CALL LAMBDA(T,XL,DXL)
      CGAM1=CSQRT(XL*XL-CK1SQ)
      CGAM2=CSQRT(XL*XL-CK2SQ)
      IF(REAL(CGAM1).EQ.0) CGAM1=CMPLX(0.,-ABS(AIMAG(CGAM1)))
      IF(REAL(CGAM2).EQ.0) CGAM2=CMPLX(0.,-ABS(AIMAG(CGAM2)))
      CINTN=CK2SQ*CGAM1+CK1SQ*CGAM2
      C2=CEXP(+CGAM1*ZMH)
      COM=DXL*XL*C2/CINTN
      COMU=DXL*XL*C2/(CGAM1+CGAM2)
      CALL BESSEL(XL*R,B0,B0P)
      U11=COMU*B0
      V11=COM*B0
      V11R=0
      IF(R.NE.0) V11R=COM*XL*B0P
      V11RZ=COM*XL*CGAM1*B0P
      V11Z2=COM*CGAM1*CGAM1*B0
      V11R2=0
      IF(CABS(XL).EQ.0) GO TO G1
      IF(R.NE.0) V11R2=-COM*XL*XL*(B0P/(XL*R)+B0)
      IF(R.EQ.0) V11R2=-COM*XL*XL*.5
   G1 ANS(1)=V11R2
      ANS(2)=V11Z2
      ANS(3)=V11RZ
      ANS(4)=V11R
      ANS(5)=V11
      ANS(6)=U11
      RETURN
      END
```

```
      CODE ANALYSIS
      SUBROUTINE SETUP(FRQ,EPSR1,SIG1,EPSR2,SIG2)
      PARAMETER (PI=3.14159265358979323)
      COMMON /EVALCOM/ H1,Z1,ZMH,ZPH,R,R1,R2,-COMPLEX-CJ,
     . CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      PARAMETER (CLIGHT=2.997924562E8)    $$$ NBS JAN 1973
      XMU0 = 4.E-07*PI
      EPS0=1./(XMU0*CLIGHT**2)
      CJ=CMPLX(0.,1.)
      P=1. $$$ P IS CURRENT MOMENT
      W=2.*PI*FRQ
      CK1SQ=CMPLX(W*W*XMU0*EPS0*EPSR1,W*XMU0*SIG1)
      CK2SQ=CMPLX(W*W*XMU0*EPS0*EPSR2,W*XMU0*SIG2)
      CK1=CSQRT(CK1SQ)
      CK2=CSQRT(CK2SQ)

C       THESE ARE THE COEFFICIENTS WHICH MULTIPLY THE VARIOUS COMPONENTS
C       OF THE ELECTROMAGNETIC FIELD.

      COEH=-P/(4.*PI)
      COEE=CJ*W*P*XMU0/(4.*PI)


COMMENT----FOLLOWING INITIALIZES VARIABLES FOR SUBROUTINE NORTON
COMMENT----  NORTON ASSUMES UPPER MEDIUM IS FREE SPACE

      COMMON/NORTSETUP/F,ER,SIG

      F=FRQ
      ER=EPSR1
      SIG=SIG1

      RETURN
      END
```

```
COMMENT----11 DEC 74----MODIFIED TO ALWAYS PUT THE OBSERVER
COMMENT----  ON THE SURFACE OF THE OBSERVER SEGMENT
      CODE ANALYSIS
      SUBROUTINE SFIELDS(T,ETANG)
      COMMON /SFCOM/ XS,YS,ZS,AS,BS,SS,WR,XO,YO,ZO,AO,BO,IMUTUAL
COMMENT---- -S FOR SOURCE -O FOR OBSERVER
      COMPLEX EX,EY,EZ,PK,SPK,CPK
      COMPLEX ERV,EZV,ERH,EZH,EPH
      PARAMETER (PI=3.14159265358979321)
      COMMON /EVALCOM/ HI,ZI,ZMH,ZPH,R,RI,R2,-COMPLEX-CJ,
     .  CK1,CK1SQ,CK2,CK2SQ,COEE,COEH
      DIMENSION -COMPLEX- ETANG(3) $$$ 1=K TERM, 2=SIN TERM, 3=COS TERM
COMMENT----XY,YT,ZT SPECIFY POINT ALONG SOURCE SEGMENT AS FCN OF PARAMETER T
COMMENT----WHICH VARIES FROM -1. TO +1.
      SHALF=SS*.5  $$$ HALF OF SEGMENT LENGTH
      XT=XS+T*SHALF*COS(BS)*COS(AS)
      YT=YS+T*SHALF*SIN(BS)*COS(AS)
      ZT=ZS+T*SHALF*SIN(AS)
      GO TO GSELF $$$ ALWAYS OFFSET THE OBSERVER
      IF(IMUTUAL.EQ.0) GO TO GSELF
COMMENT----DO MUTUAL TERM
      R=SQRT((XT-XO)**2+(YT-YO)**2) $$$ FIND RHO OF CYL. COORD. SYSTEM
      PHI=ATAN2(YO-YT,XO-XT) $$$ ANGLE BETWEEN SOURCE AND OBS IN CYL. COORD. SYS
      GO TO G10
GSELF CONTINUE $$$ DO SELF TERM
COMMENT----GO PERPENDICULAR TO ELEMENT IN A PLANE PARALLEL TO X-Y PLANE
COMMENT----FOR THE DISTANCE WR (WIRE RADIUS)
      XOP=XO-WR*SIN(BO)
      YOP=YO+WR*COS(BO)
      R=SQRT((XOP-XT)**2+(YOP-YT)**2)
      PHI=ATAN2(YOP-YT,XOP-XT)
  G10 CONTINUE
      ZI=ZO $$$ Z OF OBSERVER IN CYL. COORD. SYS.
      HI=ABS(ZT) $$$ Z OF SOURCE IN CYL. COORD. SYS.
      ZPH=ZI+HI
      ZMH=ZI-HI
      RI=SQRT(R*R+ZPH*ZPH)
      R2=SQRT(R*R+ZMH*ZMH)
C      WOT 3, (////3H***4E15.5) ,HI,ZI,R,PHI
      COMMON/FREQ/FREQ,DUMMY(6)
      WVLENGTH=3.E8/FREQ
      IF(ZT.GE.0 .AND. R.LE.WVLENGTH/30.)
     .  CALL EVALUA2(ERV,EZV,ERH,EZH,EPH)
      IF(ZT.GE.0 .AND. R.GT.WVLENGTH/30.)
     .  CALL EVALUA3(ERV,EZV,ERH,EZH,EPH)
C      IF(ZT.LT.0 .AND. R.LE.WVLENGTH/30.)
C     .  CALL EVALUB2(ERV,EZV,ERH,EZH,EPH)
C      IF(ZT.LT.0 .AND. R.GT.WVLENGTH/30.)
C     .  CALL EVALUB3(ERV,EZV,ERH,EZH,EPH)
```

```
COMMENT----MODIFY CURRENT MOMENTS ACCORDING TO ORIENTATION OF SOURCE
      ERV=ERV*SIN(AS)
      EZV=EZV*SIN(AS)
      ERH=ERH*COS(AS)*COS(PHI-BS)
      EZH=EZH*COS(AS)*COS(PHI-BS)
      EPH=EPH*COS(AS)*SIN(PHI-BS)
COMMENT----FIND COMPONENTS OF FIELD ALONG X,Y, AND Z AXES
      EX=(ERH+ERV)*COS(PHI)-EPH*SIN(PHI)
      EY=(ERH+ERV)*SIN(PHI)+EPH*COS(PHI)
      EZ=EZH+EZV
COMMENT----FIND COMPONENTS OF FIELD TANGENTIAL TO OBSERVER
      PK=CK2 $$$ VALID ONLY FOR SOURCE AND OBSERVER ABOVE GROUND
      IF(ZS.LT.0) PK=CK1  $$$ ASSUMES A SEGMENT NEVER CROSSES INTERFACE
      CALL CSINCOS(PK*T*SHALF,SPK,CPK)
      ETANG(1)=EX*COS(AO)*COS(BO)+EY*COS(AO)*SIN(BO)+EZ*SIN(AO)
      ETANG(1)=ETANG(1)*SS/2 $$$ DUE TO CNG OF VARIABLE TO PARAM. T
      ETANG(1)=CONJG(ETANG(1)) $$$ CONVERT FROM -JWT TO +JWT TIME CONVENTION
      ETANG(2)=ETANG(1)*SPK $$$ SIN INTERPOLATION TERM
C      IF(IMUTUAL.EQ.0) ETANG(2)=0 $$$ NO CONTRIBUTION TO SELF TERM
COMMENT----ABOVE CARD VALID ONLY FOR HORIZ ELEMENTS
      ETANG(3)=ETANG(1)*CPK $$$ COS INTERPOLATION TERM
      IF(DEBUG.NE.0) WOT 3, (5E15.5) ,R,R1,R2,REAL(ETANG(1)),AIMAG(ETANG(1))
      RETURN
      END
```

```
      CODE ANALYSIS
      COMPLEX TESTC,F1,F2
      STRUCTURE (F1,F1R/F1I),(F2,F2R/F2I)
      FUNCTION TESTC (F1,F2)
      A=0
      IF(ABS(F2R).GT.1.E-40) A=ABS((F1R-F2R)/F2R)
      B=0
      IF(ABS(F2I).GT.1.E-40) B=ABS((F1I-F2I)/F2I)
      TESTC=CMPLX(A,B)
      RETURN
      END
```

```
TEST PROBLEM--HALF WAVE DIPOLE IN FREE SPACE
4,-1,1,0,0,1,0,0,0,0,0,1,1,0,
.010,0.,1,1,1.,0,1.,0.,
1 -24.6063,0.,8.2,0,24.6063,0,8.2,0,
21,0,0,0,.01,
H
0
1,11,1.,0.,0,
NONTOEP 0,0,
  ERF=9. SIGF=1.E-2 RANGE=1.E5 DLAND=200.
   NPHIP=0 NTHETP=0
  NPHI=1 PHIMIN=0 PHIMAX=0
  NTHET=72 THETMIN=-90. THETMAX=90.
  TITLE=40HELEVATION PATTERN (P-TOTAL=1 WATT) PHI=0 77B
  MORE=1 END
   NPHIP=0 NTHETP=0
  NPHI=1 PHIMIN=90. PHIMAX=90.
  TITLE=42HELEVATION PATTERN (P-TOTAL=1 WATT) PHI=90. 77B
  MORE=1 END
  NPHIP=36 NTHETP=9
  NPHI=144 PHIMIN=0 PHIMAX=360.
  NTHET=1 THETMIN=90. THETMAX=90.
  TITLE=45H SURFACE WAVE (TOTAL RADIATED POWER=1.0 WATT) 77B
  VMIN=0 HMIN=0 VMAX=0 HMAX=0
  END
HALF WAVE DIPOLE 2.5M ABOVE LOSSY GND USING SPECULAR R/C
4,-1,1,0,0,1,0,0,0,0,0,1,1,0,
.010,0.,1,2,1.,0,9.,1.E-2,
1 -24.6063,0.,8.2,0,24.6063,0,8.2,0,
21,0,0,0,.01,
H
0
1,11,1.,0.,0,
NONTOEP 0,0,
  ERF=9. SIGF=1.E-2 RANGE=1.E5 DLAND=200.
   NPHIP=0 NTHETP=0
  NPHI=1 PHIMIN=0 PHIMAX=0
  NTHET=72 THETMIN=-90. THETMAX=90.
  TITLE=40HELEVATION PATTERN (P-TOTAL=1 WATT) PHI=0 77B
  MORE=1 END
   NPHIP=0 NTHETP=0
  NPHI=1 PHIMIN=90. PHIMAX=90.
  TITLE=42HELEVATION PATTERN (P-TOTAL=1 WATT) PHI=90. 77B
  MORE=1 END
  NPHIP=36 NTHETP=9
  NPHI=144 PHIMIN=0 PHIMAX=360.
  NTHET=1 THETMIN=90. THETMAX=90.
  TITLE=45H SURFACE WAVE (TOTAL RADIATED POWER=1.0 WATT) 77B
  VMIN=0 HMIN=0 VMAX=0 HMAX=0
  END
HALF WAVE DIPOLE 2.5M ABOVE LOSSY GND USING SOMMERFELD
4,-1,1,0,0,1,0,0,0,0,0,1,1,0,
.010,0.,1,3,1.,0,9.,1.E-2,
1 -24.6063,0.,8.2,0,24.6063,0,8.2,0,
21,0,0,0,.01,
H
0
1,11,1.,0.,0,
NONTOEP 0,0,
  ERF=9. SIGF=1.E-2 RANGE=1.E5 DLAND=200.
   NPHIP=0 NTHETP=0
  NPHI=1 PHIMIN=0 PHIMAX=0
  NTHET=72 THETMIN=-90. THETMAX=90.
  TITLE=40HELEVATION PATTERN (P-TOTAL=1 WATT) PHI=0 77B
```

```
MORE=1 END
 NPHIP=0 NTHETP=0
NPHI=1 PHIMIN=90. PHIMAX=90.
TITLE=42HELEVATION PATTERN (P-TOTAL=1 WATT) PHI=90. 778
MORE=1 END
NPHIP=36 NTHETP=9
NPHI=144 PHIMIN=0 PHIMAX=360.
NTHET=1 THETMIN=90. THETMAX=90.
TITLE=45H SURFACE WAVE (TOTAL RADIATED POWER=1.0 WATT) 778
VMIN=0 HMIN=0 VMAX=0 HMAX=0
END
```

References

1. A Sommerfled, "Uber die Ausbreitung der Wellen in der drahtlosen Telegraphie," Ann. Phys. 28, 665 (1909).
2. A. Sommerfeld, "Uber die Ausbreitung der Wellen in der drahtlosen Telegraphie," Ann. Phys., Ser. 4 81 (17) 1135 (1926).
3. R. J. Lytle and D. L. Lager, Numerical Evaluation of Sommerfeld Integrals, Lawrence Livermore Laboratory, Rept. UCRL-51628 (1974).
4. K. A. Norton, "The Propagation of Radio Waves over the Surface of the Earth and in the Upper Atmosphere," Proc. IRE 25, (9), 1203 (1937).
5. A. Banos, Dipole Radiation on the Presence of a Conducting Half-Space (Pergamon Press, New York, 1966.
6. E. K. Miller, A. J. Poggio, G. J. Burke, and E. S. Selden, "Analysis of Wire Antennas in the Presence of a Conducting Half-Space, Part I1. The Vertical Antenna in Free space." Can. J. Phys. 50, 879 (1972).
7. E. K. Miller, A. J. Poggio, G. J. Burke, and E. S. Selden, "Analysis of Wire Antennas in the Presence of a Conducting Half-Space, Part II. The Horizontal Antenna in Free Space," Can. J. Phys. 50, 2614 (1972).
8. E. K. Miller, Computer Models for Antennas, Lawrence Livermore Laboratory, Rept. UCRL-51515 (1974).
9. M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, National Bureau of Standards Applied Mathematics Series 55 (1965).
10. R. W. P. King, The Theory of Linear Antennas (Harvard Univ. Press, Cambridge, Mass., 1956), pp. 695-817.
11. D. Shanks,"Non-Linear Transformations of Divergent and Slowly Convergent Sequences," J. Math. Phys. 24, 1 (1955).
12. E. K. Miller, "A Variable Interval Width Quadrature Technique Based on Romberg's Method," J. Comp. Phys. 5, 265 (1970).

CRS:edas