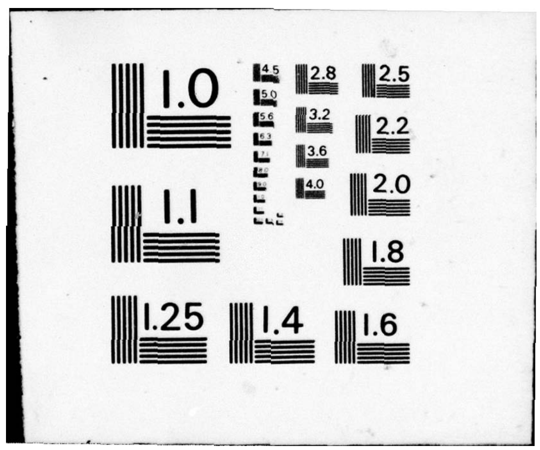
AD-A031 298 MB ASSOCIATES SAN RAMON CALIF F/G 9/5
ANTENNA MODELING PROGRAM SUPPLEMENTARY COMPUTER PROGRAM MANUAL --ETC(U)
APR 75 N00014-71-C-0187
UNCLASSIFIED MB-R-75/37 NL

1 OF 1
ADA031298

END
DATE
FILMED
11 - 76


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

1 B.S.

6

ANTENNA MODELING PROGRAM SUPPLEMENTARY COMPUTER PROGRAM MANUAL (AMPJ).

15 N00014-71-C-0187

14 MB-R-75/37

12 58p.

11 April 1975

PREPARED BY
MBA MBAssociates
Bollinger Canyon Road, San Ramon, California 94583

222 400
LB

FOREWORD

This manual is a supplement to the Engineering, User's and Systems manuals prepared for the Antenna Modeling Program (AMP), and describes the operation, theory and coding of the changes made to AMP for more accurate treatment of multiple wire junctions and reduction of the time for interaction calculations on large structures.

The AMP code as modified (AMPJ) has been delivered to the Naval Ship Engineering Center and U.S. Army Strategic Communications Command and was developed under Office of Naval Research Contract N00014-71-C-0187.

TABLE OF CONTENTS

1.0 INTRODUCTION

Wire antennas and their supporting structures frequently include junctions of several wires connected together at a point. Such multiple junctions are especially common when wire grids are used to model solid surfaces as is often done in mathematical modeling of antennas. The antenna modeling computer program AMP[1, 2, 3] for wire antennas and the extended program AMP2[4] for modeling wires and surfaces both allow for modeling multiple wire junctions, and have demonstrated good results for many such structures. In some cases, however, when segments of greatly different lengths have been joined at multiple junctions the method used in these programs has yielded inaccurate results. Hence a more careful treatment of the current interpolation at multiple junctions was developed and is included as an option in program AMPJ. The details of this method are given in Section 3.0 of this manual and the data cards to request the optional treatment at specific junctions are described in Section 2.0. Although the optional technique is numerically more stable for the general case of unequal segment lengths at a junction, it was not included in the extended code AMP2 for the following reasons: 1) the technique used in AMP2 has demonstrated good results for equal segment lengths and has a good record; 2) the new technique as implemented for testing requires an extra unknown at each multiple wire junction where used, and there are many junctions in a wire grid; 3) the new technique requires more development and testing which was outside the scope of this contract.

Program AMPJ also includes an approximate matrix filling method that may be used for interactions between segments separated by more than a specified distance to reduce matrix fill time. Hence the only feature of AMP2 that is not available in AMPJ is modeling of surfaces via the magnetic field integral equation. AMPJ includes all features of program AMP and in addition the optional junction interpolation and approximate matrix fill method.

Section 2.0 of this manual contains instructions for use of the features of AMPJ not in program AMP, and supplements the AMP Users Manual[1]. Section 3.0 gives the equations for the junction interpolation and approximate matrix fill methods, and supplements the AMP Engineering Manual[2]. Finally, section 4.0 details the coding of the routines that differ from those in program AMP, supplementing the AMP Systems Manual[3]. Listings of the changed routines are included.

2.0 PROGRAM OPERATION

The basic information needed to use program AMPJ is contained in the AMP Users Manual[1]. This section contains supplementary instructions and information for using the optional junction interpolation and approximate matrix fill methods. If the new options are not required, AMPJ may be used exactly as AMP. The one exception is that AMPJ uses a time saving approximation in filling the interaction matrix for interaction distances greater than one wavelength. For results identical to those of AMP in all digits printed, this approximation range should be increased to greater than the maximum structure dimension in wavelengths by use of a KH data card.

The standard interpolation method involves the extrapolation of a segment current by the average distance of the centers of other segments connected to the junction, requiring that the extrapolated current for the segment equal the negative of the sum of the currents at the centers of the other segments. Because of the average, this technique can lead to problems when the segment lengths at the junction are greatly unequal. The optional junction interpolation method satisfies Kirchhoff's Current Law directly at the junction and also forces the derivatives of the current with respect to distance at the ends of each of the segments at the junction to be equal. The condition on current derivatives is based on continuity of potential for equal segment radii.

Since an additional unknown is required in the matrix equation for each junction at which the optional junction interpolation method is used, this method should be used only when required for accuracy. The standard interpolation method has been found to work well as long as the connected segments at multisegment junctions have nearly equal lengths. Results in Section 3.0 show an error of about 15 percent in the input admittance of an antenna when the segment lengths at a junction near the source differ by a factor of two. The accuracy of the computed radiated field is less sensitive to the segment lengths than is current. In general, however, segment lengths should be kept within a factor of two when the standard interpolation method is used.

With the optional junction interpolation method limited testing has shown that good accuracy may be obtained with segment lengths at a junction differing by factors of 8 to 10. With the AMPJ program an antenna model may conveniently be run both with standard and optional interpolation methods to test the need for the optional method. When complex structures are being modeled, it is quite advantageous to be able to use different length segments at junctions with confidence. As a result it may be possible to reduce the number of segments so that the matrix size and the running time are decreased in spite of the additional unknowns for multiple junctions.

The program execution time is the same as for program AMP except for differences due to use of the optional junction interpolation method and approximate matrix fill. The central processor time approximately follows the formula

$$T = Ak\,(1 - 0.7\,R_w)\,N^2/M + B\,(N + N_j)^3/M^2$$
$$+ CN_e N^2/M + DkN_f N$$

where N = number of segments in model

M = number of degrees of symmetry

N_e = number of different excitations

N_f = number of far field calculation points

N_j = number of junctions at which optional interpolation method is used

R_w = the fraction of all segment pairs for which the separation is greater than R_o where R_o is the limit set on the KH card for change over to the approximate matrix fill method.

k = 1 if structure is in free space

2 if structure is over ground

and A, B, C, D are proportionality constants. The first term in this equation represents the time for matrix filling; the second term, matrix factoring; the third term, solution for the current and the fourth term, calculation of the far fields. Each term represents only the dominant component neglecting terms of lower order in N. The proportionality factors depend on the computer system on which the program is run. To give an idea of the importance of the terms, the factors in seconds for a CDC 6600 computer with the program compiled under the Run compiler and the matrix fitting in core are roughly

$$A = 2 \times 10^{-3}, \; B = 5 \times 10^{-6}, \; C = 2 \times 10^{-5}, \; D = 3 \times 10^{-4}$$

If an antenna is analyzed for only a single excitation and the far field is computed at a few angles the execution time will consist almost entirely of the time to fill and factor the interaction matrix. If a number of excitations are requested, especially for out of core solutions, the time to solve the factored matrix equation for the current distribution can become significant, and if a large number of far field calculations are requested their computation time must be considered.

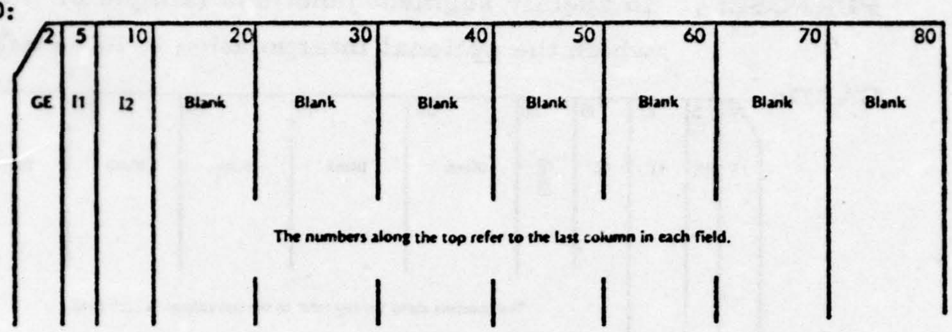
2.1 NEW INPUT CARDS

The input to program AMPJ is identical to that for program AMP except for the addition of data cards to specify junctions at which the optional interpolation of the current is to be applied, and the separation distance at which the matrix filling changes over to an approximate form. Hence the user should refer to the AMP Users Manual[1] for the basic input data structure. If only the basic data cards are used the interpolation at all junctions will be the same as that in program AMP and approximate matrix fill will be used for segments separated by more than one wavelength.

To specify matched derivative interpolation at one of more junctions a one must be punched in column 10 of the GE card at the end of the geometry data. This card may be followed by one or more cards with the mnemonic JX to specify the junctions at which matched derivative interpolation is desired and finally a card with mnemnic JE to indicate the end of junction specifications. The form of these data cards is shown below.

END GEOMETRY INPUT (GE)

CARD:



The function of this card is the same as in program AMP except for the addition of the integer I2. If I2 is equal to 1 the program reads junction specification cards following the GE card. If I2 is blank standard data cards, described in the AMP User's Manual are expected after the GE card.

JUNCTION SPECIFICATION (JX)

PURPOSE: to specify segment junctions (simple or multiple) at which the optional interpolation is to be used.

CARD:

2	5	10	15	20	30	40	50	60	70	80
JX	IX	IY	IZ	Blank	Blank	Blank	Blank	Blank	Blank	Blank

The numbers along the top refer to the last column in each field.

PARAMETERS:

INTEGERS

IX - tag number of one segment at the junction. Blank or zero for IX implies that the segment will be identified by the absolute segment number in the next location (IY).

IY - equal to m, specifies the m^{th} segment of the set of segments with tag numbers equal to IX. If IX is zero or blank, IY is the absolute segment number.

IZ - specifies the end of the segment determined by IX and IY. IZ equal to 2 specifies end 2 of the segment and 1 or blank specifies end 1 (reference direction for current is toward end 2).

NOTES:

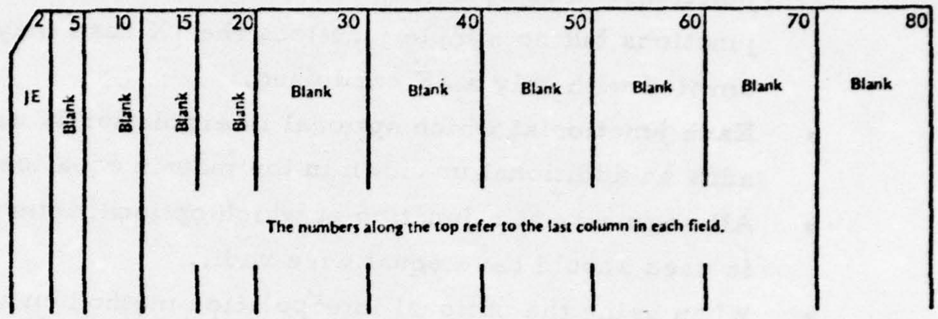
- Optional interpolation is used for the junction at the specified end of the specified segment. The junction may be either simple or multiple. A JX card is required for only one segment end at a junction to cause use of optional interpolation for all segments at the junction.

- If IX and IY are blank, the card will cause optional interpolation at all multiple junctions. Such a JX card must occur before any other JX cards specifying simple junctions. For optional interpolation at all multiple junctions but no simple junctions the JX card may be omitted with only a JE card used.
- Each junction at which optional interpolation is used adds an additional unknown in the matrix equation.
- All segments at a junction at which optional interpolation is used should have equal wire radii.
- When using the optional interpolation method on a structure for which symmetry is used in the solution, any junctions to be specified in the first symmetric section must be specified first followed by the same junctions in the second section and continuing through all sections in the order that they were produced by reflection or rotation. In addition, the junctions must be specified in the same order in each section. If only a JE card (no JX card) is used to specify all multiple segment junctions these rules will automatically be satisfied. Use of a JX card with IX and IY blank to specify all multiple junctions followed by other JX cards for simple junctions is not allowed with symmetry since the junctions will not be specified in the proper order.

END JUNCTION SPECIFICATION (JE)

PURPOSE: to mark the end of the JX cards or specify all multiple junctions.

CARD:



PARAMETERS: None

If a 1 is punched in column 10 of the GE card then a JE card is required to return the program to reading the standard input cards. If one or more JX cards are used the JE card is placed at the end of the JX cards and its only function is to mark the end of these cards. If no JX cards are used the JE card alone will cause the use of optional interpolation at all multiple junctions (three or more wires joined).

INTERACTION APPROXIMATION RANGE (KH)

PURPOSE: to set the minimum segment separation distance for use of a time saving approximation in filling the matrix.

CARD:

2	5	10	15	20	30	40	50	60	70	80
KH	Blank	Blank	Blank	Blank	RKH	Blank	Blank	Blank	Blank	Blank

The numbers along the top refer to the last column in each field.

PARAMETERS:

DECIMAL NUMBERS

RKH - The approximation is used for interaction between two segments separated by more than RKH wavelengths.

NOTES:

- For segments separated by more than RKH wavelengths the interaction field is computed from an impulse approximation to the segment current. The field of a current element located at the segment center is used. For separations less than RKH a current interpolation function is integrated over the segment length as in the basic AMP program.
- The KH card can be placed anywhere in the data cards following the geometry and junction cards (with FR, GN, EX, etc.) and affects all calculations requested following its occurrence. The value of RKH may be changed within a data set by use of a new KH card.
- If no KH card is used RKH has a default value of 1 wavelength. Hence to exactly duplicate a run with the AMP program a KH card should be used with RKH greater than the maximum structure dimension.

- The minimum value of RKH which can be used to obtain results within a few percent of the no approximation case seems to depend to some extent on the structure size, type, segmentation, and excitation. Values of .25 wavelengths or less have been found acceptable for symmetrically excited structures and electrically small wire grids; on the other hand, values up to .5 wavelengths have been required for very asymmetrically fed structures. No exact guidelines have been developed for RKH; therefore, it is best to experiment on any given problem type if a minimum value is desired. RKH should never be less than the length of the longest segment, however.
- The matrix fill time using the RUN compiler on a CDC 6600 computer is approximately $T_f = (2. - 1.4R_w)(10^{-3}) N^2$ seconds where R_w is the number of segment pairs for which the separation is greater than RKH, divided by the total number of segment pairs (N^2). Thus the fill time is decreased by about $70R_w$ percent.

2.2 PROGRAM OUTPUT

The program output is essentially unchanged from the basic deck. Segments connected to junctions at which optional interpolation has been specified are indicated in the block of segmentation data by connection numbers (I+ ot I-) less than -90,000. Also, the value of RKH is printed following the printing of frequency and wavelength.

3.0 FORMULATION OF THE INTERPOLATION METHOD AND APPROXIMATE MATRIX FILLING

3.1 INTERPOLATION METHOD

For the current interpolation method used in program AMP the current on segment j is approximated as

$$I_j(s) = A_j + B_j \sin k(s - s_j) + C_j \cos k(s - s_j) \qquad (1)$$

where k = free space wave number $(2\pi/\lambda)$

s_j = s at the center of segment j.

A_j, B_j and C_j are constants to be determined so that equation (1) yields the best possible approximation to the true current on the segment. Of the 3N constants to be determined for a structure having N segments, 2N are eliminated by enforcing conditions on the local behavior of the current. These conditions are used to eliminate the constant and C_j in terms of the current at the center of each segment, $I_j = I_j(s_j)$. The N unknowns, I_j, are then computed by solution of the matrix equation derived from the electric field integral equation.

In the basic program the condition used to eliminate two of the three unknowns for a segment is that equation (1), when extrapolated forward over the distance to the center of the next segment, must match the current of the center of that segment, and when extrapolated back must match the current at the center of the previous segment. At a multiple junction, where an end of segment j is connected to two or more segments, equation (1) is extrapolated a distance equal to the average of the distances from the center of segment j to the centers of all other segments connected to the junction and required to equal the algebraic sum of the currents at the centers of the other segments, relative to the reference direction of segment j. These conditions are based on the continuity of current at the junction (Kirchhoff's Current Law). Applying these conditions to equation (1) yields the three equations

$$A_j \qquad\qquad + C_j = I_j \tag{2}$$

$$A_j - B_j \sin k\,\delta_j^- + C_j \cos k\,\delta_j^- = K_j^- \tag{3}$$

$$A_j + B_j \sin k\,\delta_j^+ + C_j \cos k\,\delta_j^+ = K_j^+ \tag{4}$$

where

Δ_j = half the length of segment j

$$\delta_j^- = \Delta_j + \frac{1}{n_-} \sum_{\ell} \Delta_\ell$$

$$\delta_j^+ = \Delta_j + \frac{1}{n_+} \sum_{k} \Delta_k$$

$$K_j^- = \sum_{\ell} (\pm I_\ell)$$

$$K_j^+ = \sum_{k} (\pm I_k)$$

The summation index ℓ takes on the values of the numbers of all segments connected to the first or - end of segment j, of which there is a total of n_-, and k takes on the values of the numbers of all segments connected to the second or + end of which there is a total of n_+. The plus sign in the summation of currents is used when the reference directions for segment j and segment ℓ or k are parallel and the minus sign when reference directions are opposed. Solving equations (2), (3) and (4) for A_j, B_j and C_j yields

$$A_j = \frac{1}{\Delta}\left[K_j^- \sin k\,\delta_j^+ - I_j \sin k\,(\delta_j^- + \delta_j^+) + K_j^+ \sin k\,\delta_j^-\right] \tag{5}$$

$$B_j = \frac{1}{\Delta}\Big[K_j^- (\cos k\,\delta_j^+ - 1) + I_j (\cos k\,\delta_j^- - \cos k\,\delta_j^+)$$
$$+ K_j^+ (1 - \cos k\,\delta_j^-)\Big] \tag{6}$$

$$C_j = \frac{-1}{\Delta}\left[K_j^- \sin k\,\delta_j^+ - I_j (\sin k\,\delta_j^- + \sin k\,\delta_j^+) + K_j^+ \sin k\,\delta_j^-\right] \tag{7}$$

where $\quad \Delta = \sin k\,\delta_j^- + \sin k\,\delta_j^+ - \sin k\,(\delta_j^- + \delta_j^+)$ (8)

To solve for the I_j the terms are regrouped as

$$I_j\,(s) = K_j^- X_j\,(s) + I_j\,Y_j\,(s) + K_j^+ Z_j\,(s) \tag{9}$$

$$X_j\,(s) = \frac{1}{\Delta}\left[\sin k\,\delta_j^+ + (\cos k\,\delta_j^+ - 1)\sin k\,(s - s_j) - \sin k\,\delta_j^+ \cos k\,(s - s_j)\right] \tag{10}$$

$$Y_j\,(s) = \frac{1}{\Delta}\left[-\sin k\,(\delta_j^- + \delta_j^+) + (\cos k\,\delta_j^- - \cos k\,\delta_j^+)\sin k\,(s - s_j) + (\sin k\,\delta_j^- + \sin k\,\delta_j^+)\cos k\,(s - s_j)\right] \tag{11}$$

$$Z_j\,(s) = \frac{1}{\Delta}\left[\sin k\,\delta_j^- + (1 - \cos k\,\delta_j^-)\sin k\,(s - s_j) - \sin k\,\delta_j^- \cos k\,(s - s_j)\right] \tag{12}$$

The electric fields for filling the interaction matrix are obtained in the form of equation (9) by replacing the constant, $\sin k\,(s - s_j)$ and $\cos k\,(s - s_j)$ terms in equations (10) through (12) by the fields at the observation point due to these current distributions. The coefficient of each I_j then represents a contribution to the matrix element in row i and column j where i is the segment at which the field is evaluated.

The optional interpolation method enforces the following two conditions at a junction.

1. The sum of currents leaving the junction is zero.
2. The derivatives with respect to distance at the ends of all segments at the junction are set equal.

These conditions are applied at the junction rather than by extrapolating to the segment centers, thus eliminating the discontinuities at the junctions. The second condition is based on the continuity of potential as stated in reference 5. For equal wire radii, continuity of potential implies that the charge densities on the wire ends at a junction are equal, which through the continuity of current law,

$$\frac{\partial}{\partial s}\,I\,(s) = -\,jwq\,(s) \tag{13}$$

implies that the current derivatives are equal.

To apply these two conditions at a junction an additional unknown is introduced representing the derivative of the current on the end of each wire at the junction. For the optional interpolation on the positive end of segment j the conditions used to determine A_j, B_j and C_j in equation (1) are

$$I_j\,(s_j - \delta_j^-) = K_j^- \tag{14}$$

$$I_j\,(s_j) = I_j \tag{15}$$

$$I_j'\,(s_j + \Delta_j) = \alpha_j^+ \tag{16}$$

where α_j^+ = the current derivative at the positive end of segment j

The solutions for A_j, B_j and C_j are

$$A_j = \frac{1}{D}\left[K_j^- \cos k\,\Delta_j - I_j \cos k\,(\Delta_j + \delta_j^-) + \alpha_j^+ \sin k\,\delta_j^-\right] \tag{17}$$

$$B_j = \frac{1}{D}\left[(I_j - K_j^-)\sin k\,\Delta_j + \alpha_j^+ (1 - \cos k\,\delta_j^-)\right] \tag{18}$$

$$C_j = I_j - A_j = \frac{1}{D}\left[(I_j - K_j^-)\cos k\,\Delta_j - \alpha_j^+ \sin k\,\delta_j^-\right] \tag{19}$$

$$D = \cos k\,\Delta_j - \cos k\,(\Delta_j + \delta_j^-) \tag{20}$$

For the optional interpolation on the negative end of segment j the conditions on I(s) are

$$I_j'\,(s_j - \Delta_j) = \alpha_j^- \tag{21}$$

$$I_j\,(s_j) = I_j \tag{22}$$

$$I_j\,(s_j + \delta_j^+) = K_j^+ \tag{23}$$

and the solutions for A_j, B_j, and C_j are

$$A_j = \frac{1}{D}\left[\alpha_j^- \sin k\,\delta_j^+ + I_j \cos k\,(\Delta_j + \delta_j^+) - K_j^+ \cos k\,\Delta_j\right] \tag{24}$$

$$B_j = \frac{1}{D}\left[\alpha_j^- (\cos k\,\delta_j^+ - 1) + (I_j - K_j^+)\sin k\,\Delta_j\right] \tag{25}$$

$$C_j = I_j - A_j = \frac{1}{D}\left[(I_j - K_j^+)\cos k\,\Delta_j + \alpha_j^- \sin k\,\delta_j^+\right] \quad (26)$$

$$D = \cos k\,(\Delta_j + \delta_j^+) - \cos k\,\Delta_j \quad (27)$$

For the optional interpolation on both ends of segment j the conditions on I(s) are

$$I_j'(s_j - \Delta_j) = \alpha_j^- \quad (28)$$

$$I_j(s_j) = I_j \quad (29)$$

$$I_j'(s_j + \Delta_j) = \alpha_j^+ \quad (30)$$

and the solutions for A_j, B_j and C_j are

$$A_j = I_j - \frac{\alpha_j^- - \alpha_j^+}{2\sin k\,\Delta_j} \quad (31)$$

$$B_j = \frac{\alpha_j^- + \alpha_j^+}{2\cos k\,\Delta_j} \quad (32)$$

$$C_j = I_j - A_j = \frac{\alpha_j^- - \alpha_j^+}{2\sin k\,\Delta_j} \quad (33)$$

These equations give the constants for the current interpolation function of equation (1) once the current values I_j have been found. They could be used as a starting point to obtain the matrix elements. In the program, however, the matrix elements are obtained by starting with equations in the form of equation (9). Taking the derivative of equation (9) with respect to s and applying the appropriate conditions at the segment ends leads to the following expressions for the matrix element contributions for the row corresponding to the field observation point on segment i:

1. Optional interpolation on + end, standard on - end

$$\text{column } \ell: \int_{-\Delta_j}^{\Delta_j} G_i(s)\, X_j(s)\, ds - \frac{Z}{X} \int_{-\Delta_j}^{\Delta_j} G_i(s)\, Z_j(s)\, ds \tag{34}$$

$$\text{column } j: \int_{-\Delta_j}^{\Delta_j} G_i(s)\, Y_j(s)\, ds - \frac{Y}{X} \int_{-\Delta_j}^{\Delta_j} G_i(s)\, Z_j(s)\, ds \tag{35}$$

$$\text{column } \propto: -\frac{1}{X} \int_{-\Delta_j}^{\Delta_j} G_i(s)\, Z_j(s)\, ds \tag{36}$$

2. Optional interpolation on - end, standard on + end

$$\text{column } k: \int_{-\Delta_j}^{\Delta_j} G_i(s)\, Z_j(s)\, ds - \frac{Z}{X} \int_{-\Delta_j}^{\Delta_j} G_i(s)\, X_j(s)\, ds \tag{37}$$

$$\text{column } j: \int_{-\Delta_j}^{\Delta_j} G_i(s)\, Y_j(s)\, ds - \frac{Y}{X} \int_{-\Delta_j}^{\Delta_j} G_i(s)\, X_j(s)\, ds \tag{38}$$

$$\text{column } \propto: \frac{1}{X} \int_{-\Delta_j}^{\Delta_j} G_i(s)\, X_j(s)\, ds \tag{39}$$

3. Optional interpolation on both - and + ends

$$\text{column } j: \int_{-\Delta_j}^{\Delta_j} G_i(s)\, Y_j(s)\, ds - \int_{-\Delta_j}^{\Delta_j} G_i(s)\, X_j(s)\, ds + \int_{-\Delta_j}^{\Delta_j} G_i(s)\, Z_j(s)\, ds \tag{40}$$

$$\text{column } \propto_-: \frac{1}{X^2 - Z^2} \left(X \int_{-\Delta_j}^{\Delta_j} G_i(s)\, X_j(s)\, ds - Z \int_{-\Delta_j}^{\Delta_j} G_i(s)\, Z_j(s)\, ds \right) \tag{41}$$

$$\text{column } \propto_+: \frac{1}{X^2 - Z^2} \left(X \int_{-\Delta_j}^{\Delta_j} G_i(s)\, X_j(s)\, ds - Z \int_{-\Delta_j}^{\Delta_j} G_i(s)\, Z_j(s)\, ds \right) \tag{42}$$

For each case G_i (s) is the component of the free space dyadic Green's function for the electric field tangent to segment i at the center of that segment due to a current at s on segment j. X_j (s), Y_j (s), Z_j (s) are as defined in equations (10) through (12) with the exception that $\delta_j^- = \Delta_j$ when optional interpolation is used on the - end of segment j and $\delta_j^+ = \Delta_j$ when the optional interpolation is used on the + end, and $s_j = 0$. Also

$$X = \frac{k}{\Delta} \left[\cos k (\Delta_j + \delta_j) - \cos k \Delta_j\right] \tag{43}$$

$$Y = \frac{k}{\Delta} \left[1 - \cos k (\Delta_j + \delta_j)\right] \tag{44}$$

$$Z = \frac{k}{\Delta} \left[\cos k \Delta_j - 1\right] \tag{45}$$

where Δ is defined in equation (8) and $\delta_j = \delta_j^+$ if optional interpolation is on the - end only, $\delta_j = \delta_j^-$ if optional interpolation is on the + end only, and $\delta_j = \Delta_j$ if optional interpolation is on both ends. The column indicies ℓ and k take the values of the numbers of all segments connected to the - and + ends, respectively, when the standard interpolation end of a segment is a multiple junction.

The columns designated by α represent the unknown current derivative common to all segment ends at the junction. An additional equation for this unknown is obtained from the derivative of equation (9) evaluated at the segment end at which the current derivative is α. K_j^+ or K_j^- is replaced by the current at the segment end. By forming the sum of these equations for each segment at the junction the currents at the segment ends are eliminated by the condition that their sum be zero, leading to the equation

$$
\sum_{j=1}^{M}
\left\{
\begin{array}{l}
① \; I_{\ell_j} Z/X + I_j Y/X + \alpha/X \\
\text{or} \\
② \; - I_{k_j} Z/X - I_j Y/X + \alpha/X \\
\text{or} \\
③ \; - I_j - \alpha X/(Z^2 - X^2) - \alpha^- Z/(Z^2 - X^2) \\
\text{or} \\
④ \; I_j - \alpha X/(Z^2 - X^2) - \alpha^+ Z/(Z^2 - X^2)
\end{array}
\right\} = 0 \qquad (46)
$$

where j = 1, ---M represents the numbers of all segments connected to the junction and the expressions apply as follows:

① for junction α on the + end of segment j, standard interpolation on the - end.

② for junction α on the - end of segment j, standard interpolation on the + end.

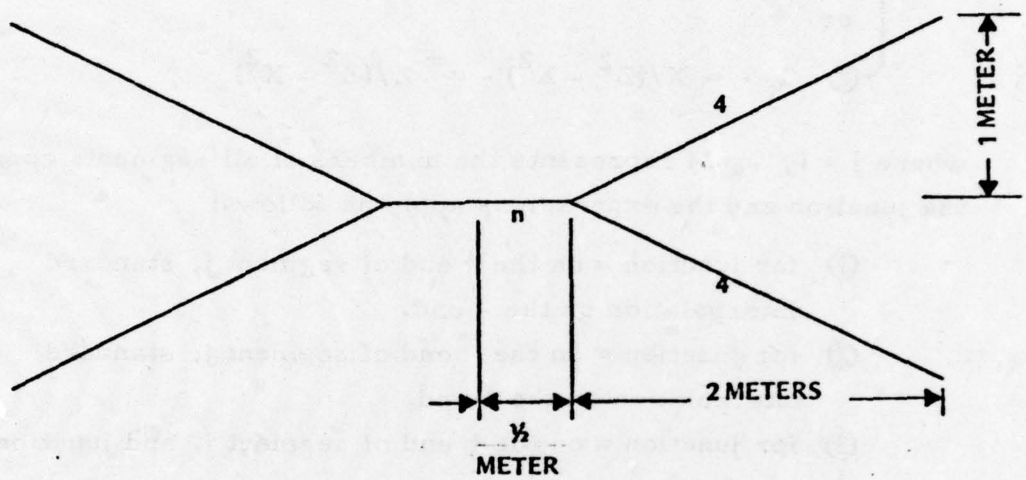
③ for junction α on the + end of segment j, and junction α^- on the - end.

④ for junction α on the - end of segment j, and junction α^+ on the + end.

The above equations are used in the program AMPJ. It is possible to eliminate the current derivative unknowns (α) rather than include them in the set of equations solved numerically but this is difficult if the new interpolation method is used on both ends of a segment. Also, the code could be generalized for unequal wire radii at junctions but this has not been done.

The stability of the two interpolation methods for modeling multiple junctions is shown in Tables 1 through 3 for an antenna composed of a linear element, fed at the center, with vee shaped loads on each end. The input impedance computed with standard interpolation at all segment junctions is shown in Table 1 for varying segment lengths on the entire center section. For this model there are 4 segments on each of the arms and n segments on half of the center element. Though no attempt

NUMERICAL TEST ANTENNA



WHEN n = 1, JUNCTION SEGMENTS ARE APPROXIMATELY EQUAL IN LENGTH

2 SEGMENT SOURCE IS USED

TABLE OF INPUT IMPEDANCE/2.

n ↓ \ FREQ. (MHz) →	270	280	290
1	33.5 + j79.5	36.7 + j95.4	40.0 + j111.
2	35.8 + j93.	39.5 + j110.	43.4 + j126.8
4	33.2 + j119.6	36.8 + j135.	40.6 + 150.4
6	28.2 + j152.5	31.1 + j165.	34.2 + j177.5
8	21.7 + j190.2	23.8 + j199.5	26.0 + j208.6
10	13.9 + j233.	15.1 + j237.9	16.3 + j243.2

TABLE 1
INPUT IMPEDANCE OF TEST ANTENNA USING STANDARD JUNCTION INTERPOLATION SCHEME FOR VARIOUS SEGMENT LENGTHS AT JUNCTION

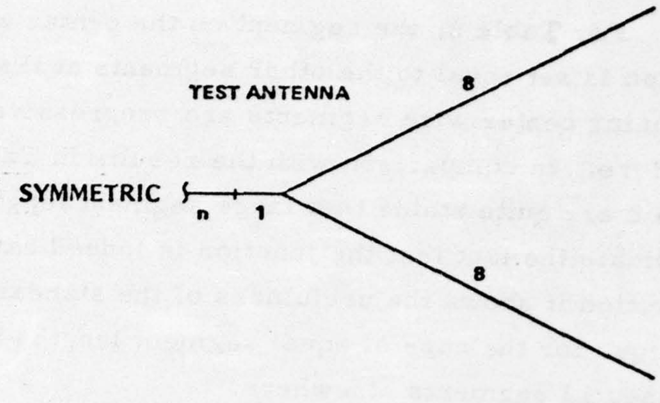
has been made to keep the source width constant, the input impedance of the structure should not vary as rapidly as indicated by these results.

For Table 2, the segment on the center wire connected to the junction is set equal to the other segments at the junction while the remaining center wire segments are progressively decreased in length as before. In comparison with the results in Table 1, the results in Table 2 are quite stable to a large segment length factor. This serves to validate the fact that the junction is indeed causing the problem, and in addition it shows the usefulness of the standard junction interpolation technique for the case of equal segment lengths at a multiple junction and unequal segments elsewhere.

Table 3 contains the results obtained using the new interpolation technique at the two multiple junctions and the standard interpolation at all junctions of two segments. The segments on the center section are progressively decreased in length including the segment connected to the junction. These results show much greater stability than the results given in Table 1 where the segment lengths at the junction are varied in a similar manner. On the other hand, the results for the new technique are somewhat less stable than for the case of equal segments at the junction (Table 2), but when complicated structures are being modeled, it is quite advantageous to be able to use different length segments at junctions with confidence.

3.2 APPROXIMATE MATRIX ELEMENTS

When wire segments in a structure are distant from an observation point with respect to wavelength, simple expressions can be used to obtain accurate values for the fields. This fact can be used to substantially reduce the time required in calculating the corresponding interaction matrix elements. The following expressions are used in the AMPJ code when segment-observation point separation permits:



SEGMENTS AT THE MULTIPLE WIRE JUNCTION
ARE APPROXIMATELY EQUAL IN LENGTH

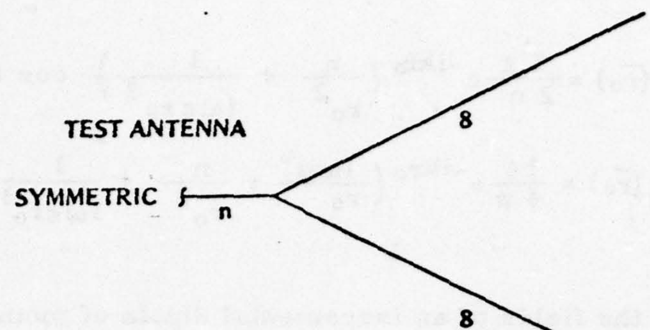
2 SEGMENT SOURCE

TABLE OF INPUT IMPEDANCE/2.

FREQ. (MHz) ⟶

n ↓	270	280	290
1	34.4 + j80.3	37.9 + j96.7	41.7 + j112.9
2	35.1 + j80.8	38.8 + j97.5	42.9 + j114.1
4	35.4 + j81.	39.4 + j98.	43.7 + j115.
6	35.6 + j82.2	39.5 + j98.6	43.9 + j115.7
8	35.6 + j83.8	39.6 + j98.6	43.9 + j112.4
10	35.9 + j99.	39.6 + j97.8	44.1 + j121.9

TABLE 2
INPUT IMPEDANCE OF TEST ANTENNA USING STANDARD JUNCTION
INTERPOLATION SCHEME FOR AN EQUAL SEGMENT LENGTH JUNCTION



WHEN n = 2, JUNCTION SEGMENTS LENGTHS ARE APPROXIMATELY EQUAL.

2 SEGMENT SOURCE

CURRENTS AND DERIVATIVES MATCHED AT JUNCTION

TABLE OF INPUT IMPEDANCE /2.

n	f = 270 MHz
2	34.5 + j78.9
4	36.1 + j77.6
6	37.0 + j76.9
8	37.7 + j76.1
10	38.1 + j75.4

TABLE 3
INPUT IMPEDANCE OF TEST ANTENNA USING THE NEW JUNCTION TECHNIQUE FOR UNEQUAL JUNCTION SEGMENT LENGTHS

$$E_r(\overline{r}_o) = \frac{I\,\ell}{2\,\pi} e^{-ikr_o} \left(\frac{\eta}{r_o^2} + \frac{1}{i\omega\varepsilon r_o^3} \right) \cos\theta \tag{47}$$

$$E_\theta(\overline{r}_o) = \frac{I\ell}{4\,\pi} e^{-ikr_o} \left(\frac{i\omega u}{r_o} + \frac{\eta}{r_o^2} + \frac{1}{i\omega\varepsilon r_o^3} \right) \sin\theta \tag{48}$$

These are the fields of an incremental dipole of moment $I\ell$ located at the origin of a standard spherical coordinate system and oriented in the z direction[6]. At sufficient distances equations (47) and (48) are used for the field of a segment where ℓ is set equal to the segment length and I is set equal to the center point current. Thus, these expressions are the same as would be obtained using a pulse function current expansion and one step integration.

This approximation has been found to yield good results for separation distances as small as .25 to .2 wavelengths. Table 4 shows the accuracy obtained for a particular structure, a 2λ dipole, for various segmentations and for various separation distances for which the expressions in equations (47) and (48) were used. The KH parameter in the table specifies the distance at which change over to the approximate field expressions occurs. The column on the left hand side of the table shows the number of segments away from the field segment which are integrated over. For this example it can be seen that the impedance accuracy remains within a few percent for a KH down to .21 wavelengths. It should be pointed out, however, that due to the quantized nature of the problem a KH parameter slightly less than .2 wavelengths will cause an abrupt change to integration over one fewer segments. For the case of $.2\lambda$ segment lengths, this means integration for the self term only and the results are poor. This problem can be avoided by keeping the KH parameter larger than the longest segment. It should also be pointed out that the minimum value for KH seems to depend to some extent on the structure size, type, segmentation, and excitation. Values of KH up to $.5\lambda$ have been necessary to obtain only a few percent error for some

NUMBER OF SEGMENTS INCLUDED	STRUCTURE SEGMENT LENGTHS		
	0.2	0.1	0.05
0	KH PARAMETER .01	.01	.01
	% ERROR REAL, IMAG. 47.2, 53.	77.3, 135.	97.4, 170.4
1	.21	.11	.06
	2.2, .62	12.4, 12.4	21.4, 190
2	.41	.21	.11
	.068, .015	1.3, 2.4	.12, 31.5
3	.61	.31	.16
	.35, .015	.09, .09	.23, 9.5
4	.81	.41	.21
	.19, .33	.028, .30	.13, 3.
6	1.21	.61	.31
	.06, .003	.035, .12	.01, .23
8	1.65	.81	.41
	.02, .022	.09, .19	.037, .041

Table 4

PER CENT ERROR OF THE INPUT IMPEDANCE OF A 2λ DIPOLE USING PARTIAL INTEGRATION AS COMPARED TO COMPLETE INTEGRATION structures with very asymmetric feeds. No exact guidelines have been established; therefore, it is probably best to experiment with any given class of problems if a minimum value of KH is sought. The default value for the KH parameter in the AMPJ code is one wavelength.

4.0 DESCRIPTION OF COMPUTER CODE CHANGES

The following subroutines have been changed to implement the optional junction interpolation and approximate matrix filling:

CABC
CMSET
DATAGN
FACTR
INTG
JMELS
LFACTR
MAIN
SOLVES
TRIO

In addition the variable JMAX has been added to common block/DATA/ throughout the program. JMAX is the total number of junctions at which the new interpolation method is used.

The matrix filled by subroutine CMSET consists of, first, segment field equations in the order of segment numbers and then equations for the current derivatives from equation (46). For a structure with symmetry the field equations for the first section are followed by the current derivative equations for that section and the equations continue in that order through all symmetric sections. Thus, for a structure with two symmetric sections the matrix equation has the form

$$\left[\begin{array}{cc|cc} A_1 & B_1 & A_2 & B_2 \\ \hdashline C_1 & D_1 & C_2 & D_2 \\ \hline A_2 & B_2 & A_1 & B_1 \\ \hdashline C_2 & D_2 & C_1 & D_1 \end{array}\right] \left[\begin{array}{c} I_1 \\ \hdashline \alpha_1 \\ \hline I_2 \\ \hdashline \alpha_2 \end{array}\right] = \left[\begin{array}{c} E_1 \\ \hdashline 0 \\ \hline E_2 \\ \hdashline 0 \end{array}\right]$$

where I_1 and α_1 represent column vectors of the unknown currents and current derivatives, respectively, on the first section. Only the upper half of the matrix, representing equations for the first section, is stored and is stored in transposed form as in program AMP. Subroutine ETMNS which fills the right hand side vector has not been modified and hence fills the applied field values in consecutive locations. Subroutine SOLVES, however, has been modified to insert zeros for the current derivative equations and reposition the applied field values for structures with symmetry before solving the matrix equation.

Some of the common block lengths have been changed from those in program AMP. The maximum number of segments is 800 in AMPJ although this is also the upper limit for the sum of the number of segments and the number of junctions at which the optional interpolation is used. Also, the area in core for storage of the interaction matrix has been increased to 10000 complex numbers. This allows structures with up to 100 unknowns (segments plus junctions with new interpolation method) to run in core. In this form the program requires approximately 240000_8 words of storage to load on a CDC 6000 series computer when compiled with the FTN compiler.

The following are brief descriptions of the changes to the modified subroutines. Lists of these routines are included at the end of this section. Since the routines have not been sequence numbered the changes can be located by the gaps in the old sequence numbers. References to statement labels refers to the labels in the left hand column of the list.

CABC

This routine computes the constants A_j, B_j and C_j for equation (1) for either the old or new interpolation method. The statements down to label 15 +1 set CLO, CLL and CLY as follows:

$$CLO = K_j^- \text{ or } \alpha^-$$

$$CLL = I_j$$

$$CLY = K_j^+ \text{ or } \alpha^+$$

A connection number less than -90000 for a segment end indicates the new interpolation method is used at that end requiring α^- or α^+. The statements from label 16 through 21 compute $AX = A_j$, $BX = B_j$ and $CX = C_j$. Statements labeled 16 + 2 through 16 + 5 evaluate equations (5) and (6); 17 + 1 through 17 + 4 evaluate equations (24) and (25); 18 through 18 + 3 evaluate equations (17) and (18); and 19 through 19 + 2 evaluate equations (31) and (32). For each case statement 20 evaluates $C_j = I_j - A_j$. Finally from statement 20 + 1 through 21 the real and imaginary parts of the constants are stored in arrays.

CMSET
Sequence number references

CM19. 1+2	:	JSEQ(J) = matrix row for segment J
CM19. 1+2	:	JJEQ(J) = matrix row for current derivative at junction J.
CM122. 1	:	Branch to section for approximate matrix fill.
CM127+1	:	

$$CE1 = \int_{-\Delta_j}^{\Delta_j} G_i(s)\; X_j(s)\, ds$$

$$CE2 = \int_{-\Delta_j}^{\Delta_j} G_i(s)\; Y_j(s)\, ds$$

$$CE3 = \int_{-\Delta_j}^{\Delta_j} G_i(s)\; Z_j(s)\, ds$$

CM127 + 11 to CM 127 + 13	:	expressions (37), (38) and (39) are evaluated
label 24 + 1 to label 24 + 3	:	expressions (34), (35) and (36) are evaluated
label 26 to label 26 + 3	:	equations (40), (41) and (42) are evaluated. In the above three cases the contributions to α columns are entered into the matrix. Other contributions are entered in the following code.

label 27 to label 28	:	fill matrix elements for - end of segment (currents in K_j^-)
label 29	:	fill matrix element for I_j
label 29 + 1 to label 30	:	fill matrix elements for + end of segment (currents in K_j^+)
CM 144.2 to CM145	:	Approximate matrix fill section. Equations (47) and (48) are evaluated.
label 18 + 1 to label 66	:	Equation (46) is evaluated for each segment. JCAS corresponds to the cases 1, 2, 3 and 4 in the equation.

DATAGN

The coding from statement label 9 + 2 through the end of this routine sets the connection numbers for segment ends at which the new interpolation will be used. The JX or JE data cards are read at statement 111. Statements 21 + 1 through 207 set the connection number for a segment end specified by a JX card and for all other segments connected to that segment end. Statements 208 through 210 search for all multiple junctions and reset the connection numbers for the new interpolation. The latter section is entered when a blank JX card occurs first or when there is no JX card but only a JE card.

The variable JMAX is used to count the number of junctions at which the new interpolation is specified. The connection numbers for all segments connected to junction number JMAX are set to -(90000 + JMAX). On exit from the routine, JMAX is left as the final number of junctions with the new interpolation and passed through common/DATA/ to other routines.

FACTR

Minor modifications have been made to FACTR following sequence number FA31 and at FA59 to use temporary variables to avoid unnecessary evaluations of subscript references.

INTG

Statements added following IG42 evaluate X, Y and Z of equations (43), (44) and (45) and store them in XM, YM, and ZM respectively.

JMELS

The calculations of the matrix row indicies JPJ and JMJ have been changed. While JP(J) and JM(J) represent segment numbers JPJ and JMJ are the locations in the matrix corresponding to these segments, taking into account the additional matrix rows for current derivative unknowns.

LFACTR

Minor modifications have been made to LFACTR at LF58 and LF92 to use temporary variables to avoid unnecessary evaluations of subscript references.

MAIN

Sequence number references		
MA64 to MA65	:	JPMAX = number of junctions with new interpolation in one symmetric section
		NEQ = total number of unknowns
		NPEQ = number of unknowns for one symmetric section.
MA139 + 1	:	RKH = default value for separation distance at which matrix filling changes over to approximate form.
MA188 to MA189	:	Set new value for RKH

SOLVES

Statements between sequence number SS11 and SS13 insert zeros in the B vector for the right hand side of the matrix equation in locations corresponding to the current derivative equations. For structures having symmetry the applied field values are relocated, using Y as scratch storage, to make room for the zeros within the vector.

Statements between sequence numbers SS67 and SS68 rearrange the solution vector for the case of a symmetric structure which uses the new interpolation on some junctions. The solution vector at SS67 consists of the currents for the first symmetric section followed by the current derivatives for junctions on the first section, then currents for the next section, and continues in this order through all sections. These statements put the currents in consecutive locations so that I_j is in location j, with the current derivatives in consecutive locations following the last current value.

TRIO

The statements following TR12 and TR18 have been added so that DIL or DIK are set to Δ_j for new interpolation on the - or + segment end respectively.

Lists of the changed routines follow.

```
      PROGRAM AMPJ(INPUT,OUTPUT,TAPE11,TAPE12,TAPE13,TAPE14,TAPE15,TAPE1
     16,TAPE17)
      COMMON /DATA/  N,NP,X(800),Y(800),Z(800),SI(800),BI(800),ALP(800),
     1 BET(800),ICON1(800),ICON2(800),ITAG(800),WLAM,IPSYM,JMAX
      COMMON CM(10000)
      COMMON /MATPAR/  ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      COMMON /SAVE/  IX(800),IP(800)
      COMMON /RESTRT/   IC1,IC2,IC3,NRES,NPRES,IFLOW
      COMMON /ANGL/  SALP(800)
      COMMON /JUNK/  NCOK,JOK(25),NCIK,JIK(25),NCIZ,JOZ(25),NCIZ,JIZ(25)
      COMMON /CRNT/  AIR(800),AII(800),BIR(800),BII(800),CIR(800),CII(80
     10),CUR(800)
      COMMON /GND/  ZRATI,ZRATI2,CL,CH,SCRWL,SCRWR,NRAD,KSYMP,IFAR,IPER
     1F
      COMMON /SCRATM/  GAIN(1600)
      COMMON /ZLOAD/  ZARRAY(800)
      DIMENSION CAB(1), SAB(1)
      DIMENSION ISEG1(30), ISEG2(30), Y11R(30), Y11I(30), Y12R(30), Y12I
     1(30), Y22R(30), Y22I(30), NTYP(30), [illegible]
      DIMENSION LDTYP(15), LDTAG(15), LDTAGF(15), LDTAGT(15), ZLR(15), Z
     1LI(15), ZLC(15), IGTP(4), IGAX(4), IGNTP(10)
      DIMENSION COM(13,5), ITAP(7), ATST(17), ISEGN(5), PNET(6), HPOL(3)
      DIMENSION FNORM(200)
      EQUIVALENCE (CAB(1),ALP(1)), (SAB(1),BET(1))
      INTEGER   AIN,ATST,PNET,HPOL,HBLK,HCIR,HCLIF
      COMPLEX   CM,FJ,VSANT,ETH,EPH,ZRATI,CUR,CURI,ZARRAY,ERD,ZRATI2
      COMPLEX   EX,EY,EZ,ZPED
      DATA   (ITAP(I), I =1,7)/11,12,13,14,15,16,17/
      DATA   ATST/2HCE,2HFR,2HLD,2HGN,2HEX,2HNT,2HXQ,2HNE,2HGD,2HRP,2HCM
     1, 2HNX,2HEN,2HTL,2HPT,2HKS,2HKH/
      DATA  ISEGN/5H2,3,9,6H2,3,10,5H2,3,3,[illegible]/
      DATA   HPOL/6HLINEAR,5HRIGHT,4HLEFT/,HBLK,HCIR/1H ,6HCIRCLE/
      DATA   PNET/6H      ,2H  ,6HSTRAIG,[illegible]CROSSE,[illegible]/
      DATA   IGTP/6H    - ,6HPOWER ,6H- DIRE,6HCTIVE /
      DATA   IGAX/6H MAJOR,6H MINOR,6H VERT.,6H HOR. /
      DATA   IGNTP/6H MAJOR,6H AXIS ,6H MINOR,6H AXIS ,6H   VER,6HTICAL
     1, 6H HORIZ,6HONTAL ,6H      ,6HTOTAL /
      DATA   PI,TA,TD,FJ/3.14159265,1.745329252E-02,57.29578,(0.,1.)/
      DATA   LOADMX,NSMAX,NETMX/15,10,30/,NORMF/200/
      ISTART=0
      IRESRV=10000
 1    K=0
 2    K=K+1
      IF (K.GT.5) K=5
      READ 125,  AIN,(COM(I,K),I=1,13)
      IF (AIN.NE.ATST(16)) GO TO 3
      ISTART=1
      GO TO 1
 3    IF (K.GT.1) GO TO 4
      PRINT 126
      PRINT 127
      PRINT 128
 4    PRINT 129,  (COM(I,K),I=1,13)
      IF (AIN.EQ.ATST(11)) GO TO 2
      IF (AIN.EQ.ATST(1)) GO TO 5
      PRINT 130
      STOP
 5    CONTINUE
      MPCNT=0
C
C     SETTING UP GEOMETRIC DATA IN SUB. DATAGN AND PRINTING
C
      CALL   DATAGN
      NOP=N/NP
      JPMAX=JMAX/NOP
      IF(JPMAX*NOP.EQ.JMAX)GO TO 212
      PRINT 213
      STOP
 212  NEQ=N+JMAX
      NPEQ=NP+JPMAX
      IFLOW=1
      NPES=N
      NPRES=NP
      PRINT 131
      PRINT 132
      DO 6 I=1,N
      ALPI=ALP(I)*TD
      BETI=BET(I)*TD
      PRINT 133,  I,X(I),Y(I),Z(I),SI(I),ALPI,BETI,BI(I),ICON1(I),I,
     1 ICON2(I),ITAG(I)
      CALP=COS(ALP(I))
      SALP(I)=SIN(ALP(I))
      CAB(I)=CALP*COS(BET(I))
      SAB(I)=CALP*SIN(BET(I))
      IF (SI(I).GT.1.E-20.AND.BI(I).GT.1.E-20) GO TO 6
      PRINT 134
      STOP
 6    CONTINUE
C
C     DECIDING WHAT TYPE OF MATRIX SOLUTION WILL BE USED IE. IN-CORE
C     OR OUT-OF-CORE
C
      NROW=NEQ
      NROWS=NPEQ
      CALL FBLOCK (NBLOKS,NPBLK,NLAST,IRESRV,NEQ,NPEQ,INT)
```

```
      IF(NEQ.NE.NPEQ)GO TO 8
      IF (INT.EQ.1) GO TO 7
      NCOL=NPEQ
      ICASE=1
      GO TO 11
7     NCOL=2*NPBLK
      ICASE=3
      GO TO 11
8     IF (INT.EQ.1) GO TO 9
      NCOL=NPEQ
      NCOLS=NPEQ
      ICASE=2
      GO TO 11
9     CALL FPLOCK (NBLSYM,NPSYM,NLSYM,IRESRV,NPEQ,NPEQ,INT)
      NCOL=2*NPBLK
      IF (INT.EQ.1) GO TO 10
      NCOLS=NPEQ
      ICASE=4
      GO TO 11
10    NCOLS=2*NPSYM
      ICASE=5
11    CONTINUE
      NBLOKX=NBLOKS
      NPBLKX=NPBLK
      NLASTX=NLAST
      NROWX=NROW
      NCOLX=NCOL
      PRINT 135
C
C     FILE PREPARATION FOR OUT-OF-CORE MATRIX SOLUTION. FILES REWOUND
C     AND ENDFILE WRITTEN.     AND TESTING FOR RESTART
C
      IF (ISTART.EQ.0) GO TO 12
      CALL   UNCAT
      GO TO 14
12    IF (ICASE.LT.3) GO TO 14
      DO 13 I=1,7
      NUNIT=ITAP(I)
      REWIND   NUNIT
      END FILE   NUNIT
      REWIND   NUNIT
13    CONTINUE
14    CONTINUE
C     DEFAULT VALUES FOR INPUT PARAMETERS AND FLAGS
      IGO=1
      FMHZS=300.
      FMHZ=300.
      NFRQ=1
      PKH=1.
      NLOAD=0
      KSYMP=1
      IXTYP=0
      NET=0
      NRADL=0
      NEAR=-1
      IPTFLG=-2
      IFAR=-1
      ZPATI=CMPLX(1.,0.)
      IPERF=0
      IPED=0
C
C     MAIN INPUT SECTION - STANDARD READ STATEMENT.- JUMPS TO APPRO-
C     PRIATE SECTION FOR SPECIFIC PARAMETER SET UP .
C
15    READ 136,  AIN,ITMP1,ITMP2,ITMP3,ITMP4,TMP1,TMP2,TMP3,TMP4,TMP5,TM
     1P6
      MPCNT=MPCNT+1
      PRINT 137,  MPCNT,AIN,ITMP1,ITMP2,ITMP3,ITMP4,TMP1,TMP2,TMP3,TMP4,
     1 TMP5,TMP6
      IF (AIN.EQ.ATST(2)) GO TO 17
      IF (AIN.EQ.ATST(3)) GO TO 18
      IF (AIN.EQ.ATST(4)) GO TO 22
      IF (AIN.EQ.ATST(5)) GO TO 25
      IF (AIN.EQ.ATST(5)) GO TO 29
      IF (AIN.EQ.ATST(14)) GO TO 29
      IF (AIN.EQ.ATST(15)) GO TO 32
      IF (AIN.EQ.ATST(7)) GO TO 38
      IF (AIN.EQ.ATST(8)) GO TO 33
      IF (AIN.EQ.ATST(9)) GO TO 35
      IF (AIN.EQ.ATST(10)) GO TO 37
      IF(AIN.EQ.ATST(17))GO TO 201
      IF (AIN.EQ.ATST(12)) GO TO 1
      IF (AIN.NE.ATST(13)) GO TO 16
      IF (ITMP1.NE.0) CALL   CATLOG
      STOP
16    PRINT 138
      STOP
C
C     FREQUENCY PARAMETERS
C
17    IFRQ=ITMP1
      NFRQ=ITMP2
      IF (NFRQ.EQ.0) NFRQ=1
      FMHZ=TMP1
      DELFRQ=TMP2
      IF (IPED.EQ.1) ZPNORM=0.
      IGO=1
      IFLOW=2
      GO TO 15
```

```
C
C      MATRIX INTEGRATION LIMIT
C
201    RKH=TMP1
       IGO=1
       IFLOW=2
       GO TO 15
C
C      LOADING PARAMETERS
C
18     IF (IFLOW.EQ.3) GO TO 19
       NLOAD=0
       IFLOW=3
       IF (IGO.GT.2) IGO=2
       IF (ITMP1.EQ.(-1)) GO TO 15
19     NLOAD=NLOAD+1
       IF (NLOAD.LE.LOADMX) GO TO 20
       PRINT 139
       STOP
20     LDTYP(NLOAD)=ITMP1
       LDTAG(NLOAD)=ITMP2
       IF (ITMP4.EQ.0) ITMP4=ITMP3
       LDTAGF(NLOAD)=ITMP3
       LDTAGT(NLOAD)=ITMP4
       IF (ITMP4.GE.ITMP3) GO TO 21
       PRINT 140,  NLOAD,ITMP3,ITMP4
       PRINT 144,  ISECN(3)
       STOP
21     ZLR(NLOAD)=TMP1
       ZLI(NLOAD)=TMP2
       ZLC(NLOAD)=TMP3
       GO TO 15
C
C      GROUND PARAMETERS UNDER THE ANTENNA
C
22     IFLOW=4
       IF (IGO.GT.2) IGO=2
       IF (ITMP1.NE.(-1)) GO TO 23
       KSYMP=1
       NRADL=0
       GO TO 15
23     IPERF=ITMP1
       NRADL=ITMP2
       KSYMP=2
       EPSR=TMP1
       SIG=TMP2
       IF (NRADL.EQ.0) GO TO 24
       SCRWLT=TMP3
       SCRWRT=TMP4
       GO TO 15
24     EPSR2=TMP3
       SIG2=TMP4
       CLT=TMP5
       CHT=TMP6
       GO TO 15
C
C      EXCITATION PARAMETERS
C
25     IF (IFLOW.EQ.5) GO TO 26
       IPTFLG=-2
       NSANT=0
       IPED=0
       IFLOW=5
       IF (IGO.GT.3) IGO=3
26     MASYM=ITMP4/10
       IF (ITMP1.GT.0) GO TO 28
       IXTYP=ITMP1
       NTSOL=0
       NSANT=NSANT+1
       IF (NSANT.LE.NSMAX) GO TO 27
       PRINT 141
       STOP
27     ISANT(NSANT)=ISEGNO(ITMP2,ITMP3)
       IPED=ITMP4-MASYM*10
       VSANT(NSANT)=CMPLX(TMP1,TMP2)
       IF (CABS(VSANT(NSANT)).LT.1.E-20) VSANT(NSANT)=(1.,0.)
       ZPNORM=TMP3
       IF (IPED.EQ.1.AND.ZPNORM.GT.0) IPED=2
       GO TO 15
28     IF (IXTYP.EQ.0) NTSOL=0
       IXTYP=ITMP1
       NTHI=ITMP2
       NPHI=ITMP3
       XPR1=TMP1
       XPR2=TMP2
       XPR3=TMP3
       XPR4=TMP4
       XPR5=TMP5
       XPR6=TMP6
       NSANT=0
       THETIS=XPR1
       PHISS=XPR2
       GO TO 15
C
C      NETWORK PARAMETERS
C
29     IF (IFLOW.EQ.6) GO TO 30
       NET=0
       NTSOL=0
```

```
      IFLOW=6
      IF (IGO.GT.3) IGO=3
      IF (ITMP2.EQ.(-1)) GO TO 15
30    NET=NET+1
      IF (NET.LE.NETMX) GO TO 31
      PRINT 142
      STOP
31    NTYP(NET)=2
      IF (AIN.EQ.ATST(6)) NTYP(NET)=1
      ISEG1(NET)=ISEGNO(ITMP1,ITMP2)
      ISEG2(NET)=ISEGNO(ITMP3,ITMP4)
      Y11R(NET)=TMP1
      Y11I(NET)=TMP2
      Y12R(NET)=TMP3
      Y12I(NET)=TMP4
      Y22R(NET)=TMP5
      Y22I(NET)=TMP6
      IF (NTYP(NET).EQ.1.OR.TMP1.GT.0.) GO TO 15
      NTYP(NET)=3
      Y11R(NET)=-TMP1
      GO TO 15
C
C     PRINT CONTROL
C
32    IPTFLG=ITMP1
      IPTAG=ITMP2
      IPTAGF=ITMP3
      IPTAGT=ITMP4
      IF (ITMP4.EQ.0) IPTAGT=IPTAGF
      GO TO 15
C
C     NEAR FIELD CALCULATION PARAMETERS
C
33    IF (.NOT.(IFLOW.EQ.8.AND.NFRQ.NE.1)) GO TO 34
      PRINT 143
      PRINT 144,  ISECN(1)
34    NEAR=ITMP1
      NRX=ITMP2
      NRY=ITMP3
      NRZ=ITMP4
      XNR=TMP1
      YNR=TMP2
      ZNR=TMP3
      DXNR=TMP4
      DYNR=TMP5
      DZNR=TMP6
      IFLOW=8
      IF (NFRQ.NE.1) GO TO 15
      GO TO (42,47,54,72,73), IGO
C
C     GROUND REPRESENTATION
C
35    IF (.NOT.(IFLOW.EQ.9.AND.NFRQ.NE.1)) GO TO 36
      PRINT 144,  ISECN(2)
36    EPSR2=TMP1
      SIG2=TMP2
      CLT=TMP3
      CHT=TMP4
      IFLOW=9
      GO TO 15
C
C     STANDARD OBSERVATION ANGLE PARAMETERS
C
37    IFAR=ITMP1
      NTH=ITMP2
      NPH=ITMP3
      IF (NTH.EQ.0) NTH=1
      IF (NPH.EQ.0) NPH=1
      IPD=ITMP4/10
      IAVP=ITMP4-IPD*10
      INOR=IPD/10
      IPD=IPD-INOR*10
      IAX=INOR/10
      INOR=INOR-IAX*10
      IF (IAX.NE.0) IAX=1
      IF (IPD.NE.0) IPD=1
      IF (NTH.LT.2.OR.NPH.LT.2) IAVP=0
      IF (IFAR.EQ.1) IAVP=0
      THETS=TMP1
      PHIS=TMP2
      DTH=TMP3
      DPH=TMP4
      RFLD=TMP5
      GNOR=TMP6
      IFLOW=10
      GO TO (42,47,54,72,80), IGO
C
C     EXECUTE CARD  -  CALC. INCLUDING RADIATED FIELDS
C
38    IF (IFLOW.EQ.10.AND.ITMP1.EQ.0) GO TO 15
      IF (NFRQ.EQ.1.AND.ITMP1.EQ.0.AND.IFLOW.GT.7) GO TO 15
      IF (ITMP1.NE.0) GO TO 40
      IF (IFLOW.GT.7) GO TO 39
      IFLOW=7
      GO TO 41
39    IFLOW=11
      GO TO 41
40    IFAR=0
      RFLD=0.
```

```
      IPD=0
      IAVP=0
      INOR=0
      IAX=0
      NTH=91
      NPH=1
      THETS=0.
      PHIS=0.
      DTH=1.0
      DPH=0.
      IF (IPTR.EQ.2) PHIS=90.
      IF (IPTR.NE.3) GO TO 41
      NPH=2
      DPH=90.
41    GO TO (42,47,54,72,80), IGO
C
C     END OF THE MAIN INPUT SECTION
C
C     BEGINNING OF THE FREQUENCY DO LOOP
C
42    MHZ=1
43    IF (MHZ.EQ.1) GO TO 45
      IF (IFRQ.EQ.1) GO TO 44
      FMHZ=FMHZ+DELFRQ
      GO TO 45
44    FMHZ=FMHZ*DELFRQ
45    FR=FMHZ/FMHZS
      WLAM=300./FMHZ
      PRINT 145, FMHZ,WLAM
      PRINT 197,PKH
C     FREQUENCY SCALING OF GEOMETRIC PARAMETERS
      FMHZS=FMHZ
      DO 46 I=1,N
      X(I)=X(I)*FR
      Y(I)=Y(I)*FR
      Z(I)=Z(I)*FR
      SI(I)=SI(I)*FR
46    BI(I)=BI(I)*FR
      IGO=2
C     STRUCTURE SEGMENT LOADING
47    PRINT 146
      IF (NLOAD.NE.0) CALL LOAD (LDTYP,LDTAG,LDTAGF,LDTAGT,ZLR,ZLI,ZLC,N
     1LOAD)
      IF (NLOAD.EQ.0) PRINT 147
C     GROUND PARAMETER
      PRINT 148
      IF (KSYMP.EQ.1) GO TO 50
      IF (IPERF.EQ.1) GO TO 49
      ZRATI=CSQRT(1./(EPSR-SIG*WLAM*59.92*FJ))
      IF (NRADL.EQ.0) GO TO 48
      SCRWL=SCRWLT/WLAM
      SCRWR=SCRWRT/WLAM
      PRINT 170, NRADL,SCRWLT,SCRWRT
      PRINT 149
48    PRINT 150, EPSR,SIG
      GO TO 51
49    PRINT 151
      GO TO 51
50    PRINT 152
51    CONTINUE
C
C     STRUCTURE MATRIX SET UP
C
      IF (ISTART.NE.0) GO TO 52
      IC1=0
      IC2=IC1
      IC3=IC1
52    NROW=NROWX
      NCOL=NCOLX
      NBLOKS=NBLOKX
      NPBLK=NPBLKX
      NLAST=NLASTX
      CALL SECOND (TIM1)
      CALL CMSET(NROW,NCOL,CM,NLOAD,PKH)
      CALL SECOND (TIM2)
      TIM=TIM2-TIM1
C
C     MATRIX FACTORIAZTION
C
      CALL FACTRS(NPEQ,NOP,CM,IP,IX,NROW,NCOL,NCOLS,IPSYM)
      ISTART=0
      IF (ICASE.LE.3) GO TO 53
      NROW=NPEQ
      NCOL=NCOLS
53    CALL SECOND (TIM1)
      TIM2=TIM1-TIM2
      PRINT 153, TIM,TIM2
      IGO=3
      NTSOL=0
C
C     EXCITATION SET UP (RIGHT HAND SIDE, -E INC.)
C
54    NTHIC=1
      NPHIC=1
      INC=1
      NPRINT=0
55    IF (IXTYP.EQ.0) GO TO 57
      IF (IPTFLG.LE.0.OR.IXTYP.EQ.4) PRINT 154
      TMP5=TA*XPR5
```

```
      TMP4=TA*XPR4
      IF (IXTYP.LE.3) GO TO 56
      TMP1=XPR1/WLAM
      TMP2=XPR2/WLAM
      TMP3=XPR3/WLAM
      TMP6=XPR6/(WLAM*WLAM)
      PRINT 156.  XPR1.XPR2.XPR3.XPR4.XPR5.XPR6
      GO TO 57
56    TMP1=TA*XPR1
      TMP2=TA*XPR2
      TMP3=TA*XPR3
      TMP6=XPR6
      IF (IPTFLG.LE.0) PRINT 155.  XPR1.XPR2.XPR3.HPOL(IXTYP).XPR6
57    CALL ETMNS (TMP1.TMP2.TMP3.TMP4.TMP5.TMP6.IXTYP.ISANT.VSANT.NSANT.
     1CUR)
C
C     MATRIX SOLVING  (NETWK CALLS SOLVES)
C
      IF (NET.EQ.0.OR.INC.GT.1) GO TO 61
      PRINT 158
      ITMP3=0
      ITMP1=NTYP(1)
      DO 60 I=1.2
      IF (ITMP1.EQ.3) ITMP1=2
      IF (ITMP1.EQ.2) PRINT 159
      IF (ITMP1.EQ.1) PRINT 160
      DO 59 J=1.NET
      ITMP2=NTYP(J)
      IF ((ITMP2/ITMP1).EQ.1) GO TO 58
      ITMP3=ITMP2
      GO TO 59
58    ITMP4=ISEG1(J)
      ITMP5=ISEG2(J)
      IF (ITMP2.GE.2.AND.Y11I(J).LE.0.) Y11I(J)=WLAM*SQRT((X(ITMP5)-X(IT
     1MP4))**2+(Y(ITMP5)-Y(ITMP4))**2+(Z(ITMP5)-Z(ITMP4))**2)
      PRINT 157.  ITAG(ITMP4).ITMP4.ITAG(ITMP5).ITMP5.Y11R(J).Y11I(J).
     1 Y12R(J).Y12I(J).Y22R(J).Y22I(J).PNET(2*ITMP2-1).PNET(2*ITMP2)
59    CONTINUE
      IF (ITMP3.EQ.0) GO TO 61
      ITMP1=ITMP3
60    CONTINUE
61    CONTINUE
      IF (INC.GT.1.AND.IPTFLG.GT.0) NPRINT=1
      CALL NETWK (ISEG1.ISEG2.Y11R.Y11I.Y12R.Y12I.Y22R.Y22I.NET.NTYP.ISA
     1NT.VSANT.NSANT.CM.IP.CUR.NROW.NCOL.IX.PIN.PLOSNT.NPRINT.MASYM.ZPED
     2.NTSOL.JPMAX)
      NTSOL=1
      IF (IPED.EQ.0) GO TO 62
      ITMP1=MHZ+4*(MHZ-1)
      IF (ITMP1.GT.(NORMF-3)) GO TO 62
      FNORM(ITMP1)=REAL(ZPED)
      FNORM(ITMP1+1)=AIMAG(ZPED)
      FNORM(ITMP1+2)=CABS(ZPED)
      FNORM(ITMP1+3)=CANG(ZPED)
      IF (IPED.EQ.2) GO TO 62
      IF (FNORM(ITMP1+2).GT.ZPNORM) ZPNORM=FNORM(ITMP1+2)
62    CONTINUE
C
C     PRINTING STRUCTURE CURRENTS
C
      IF (IPTFLG.EQ.(-1)) GO TO 64
      IF (IPTFLG.GT.0) GO TO 63
      PRINT 161
      PRINT 162
      GO TO 64
63    IF (IPTFLG.EQ.3.OR.INC.GT.1) GO TO 64
      PRINT 163.  XPR3.HPOL(IXTYP).XPR6
64    PLOSS=0.
      ITMP1=0
      JUMP=IPTFLG+1
      DO 70 I=1.N
      CURI=CUR(I)*WLAM
      CMAG=CABS(CURI)
      PH=CANG(CURI)
      IF (NLOAD.EQ.0) GO TO 65
      IF (ABS(REAL(ZARRAY(I))).LT.1.E-20) GO TO 65
      PLOSS=PLOSS+.5*CMAG*CMAG*REAL(ZARRAY(I))*SI(I)
65    IF (JUMP) 69,70,66
66    IF (IPTAG.EQ.0) GO TO 67
      IF (ITAG(I).NE.IPTAG) GO TO 70
67    ITMP1=ITMP1+1
      IF (ITMP1.LT.IPTAGF.OR.ITMP1.GT.IPTAGT) GO TO 70
      IF (IPTFLG.EQ.0) GO TO 69
      IF (IPTFLG.LT.2.OR.INC.GT.NORMF) GO TO 68
      FNORM(INC)=CMAG
      ISAVE=I
68    IF (IPTFLG.NE.3) PRINT 164.  XPR1.XPR2.CMAG.PH.I
      GO TO 70
69    PRINT 165.  I.ITAG(I).X(I).Y(I).Z(I).SI(I).CURI.CMAG.PH
70    CONTINUE
      IF (IXTYP.NE.0) GO TO 71
      TMP1=PIN-PLOSNT-PLOSS
      TMP2=100.*TMP1/PIN
      PRINT 166.  PIN.TMP1.PLOSS.PLOSNT.TMP2
71    CONTINUE
      IGO=4
      IF (IFLOW.NE.7) GO TO 72
      IF (IXTYP.GT.0.AND.IXTYP.LT.4) GO TO 114
      IF (NFRQ.NE.1) GO TO 121
```

```
      PRINT 135
      GO TO 15
C
C     CALCULATION OF A,B,C IN CURRENT EXPANSION
C
72    CALL CABC
      IGF=5
C
C     NEAR FIELD CALCULATION
C
73    IF (NEAR.EQ.(-1)) GO TO 79
      PRINT 167
      ZNRT=ZNR-DZNR
      DO 78 I=1,NRZ
      ZNRT=ZNRT+DZNR
      IF (NEAR.EQ.0) GO TO 74
      CTH=COS(TA*ZNRT)
      STH=SIN(TA*ZNRT)
74    YNRT=YNR-DYNR
      DO 78 J=1,NRY
      YNRT=YNRT+DYNR
      IF (NEAR.EQ.0) GO TO 75
      CPH=COS(TA*YNRT)
      SPH=SIN(TA*YNRT)
75    XNRT=XNR-DXNR
      DO 78 KK=1,NRX
      XNRT=XNRT+DXNR
      IF (NEAR.EQ.0) GO TO 76
      XOB=XNRT*STH*CPH
      YOB=XNRT*STH*SPH
      ZOB=XNRT*CTH
      GO TO 77
76    XOB=XNRT
      YOB=YNRT
      ZOB=ZNRT
77    TMP1=XOB/WLAM
      TMP2=YOB/WLAM
      TMP3=ZOB/WLAM
      CALL NEFLD (TMP1,TMP2,TMP3,EX,EY,EZ)
      TMP1=CABS(EX)
      TMP2=CANG(EX)
      TMP3=CABS(EY)
      TMP4=CANG(EY)
      TMP5=CABS(EZ)
      TMP6=CANG(EZ)
      PRINT 168,  XOB,YOB,ZOB,TMP1,TMP2,TMP3,TMP4,TMP5,TMP6
78    CONTINUE
      IF (NHZ.EQ.NFRQ) NEAR=-1
      IF (NFRQ.NE.1) GO TO 79
      PRINT 135
      GO TO 15
79    CONTINUE
C
C     STANDARD FAR FIELD CALCULATION
C
80    IF (IFAR.EQ.-1) GO TO 114
      IF (IFAR.LT.2) GO TO 82
      PRINT 169
      IF (IFAR.LE.3) GO TO 81
      PRINT 170,  NRADL,SCRWLT,SCRWRT
      IF (IFAR.EQ.4) GO TO 82
81    IF (IFAR.EQ.2.OR.IFAR.EQ.5) HCLIF=HPOL(1)
      IF (IFAR.EQ.3.OR.IFAR.EQ.6) HCLIF=HCIR
      CL=CLT/WLAM
      CH=CHT/WLAM
      ZRATI2=CSQRT(1./(EPSR2-SIG2*WLAM*59.92*FJ))
      PRINT 171,  HCLIF,CLT,CHT,EPSR2,SIG2
82    IF (IFAR.NE.1) GO TO 83
      PRINT 175
      GO TO 85
83    I=2*IPD+1
      J=I+1
      ITMP1=2*IAX+1
      ITMP2=ITMP1+1
      PRINT 172
      IF (RFLD.LT.1.E-20) GO TO 84
      EXRM=WLAM/RFLD
      EXRA=RFLD/WLAM
      EXRA=-360.*(EXRA-AINT(EXRA))
      PRINT 173,  RFLD,EXRM,EXRA
84    PRINT 174,  IGTP(I),IGTP(J),IGAX(ITMP1),IGAX(ITMP2)
85    IF (IXTYP.EQ.0) GO TO 87
      IF (IXTYP.EQ.4) GO TO 86
      PRAD=0.
      GCON=4.*PI/(1.+XPR6*XPR6)
      GCOP=GCON
      GO TO 88
86    PIN=394.51*XPR6*XPR6*WLAM*WLAM
87    GCOP=WLAM*WLAM*2.*PI/(376.73*PIN)
      PRAD=PIN-PLOSS-PLOSNT
      GCON=GCOP
      IF (IPD.NE.0) GCON=GCON*PIN/PRAD
88    I=0
      GMAX=-1.E10
      PINT=0.
      TMP1=DPH*TA
      TMP2=.5*DTH*TA
      PHI=PHIS-DPH
      DO 109 KPH=1,NPH
```

```
      PHI=PHI+DPH
      PHA=PHI*TA
      THET=THETS-DTH
      DO 109 KTH=1,NTH
      THET=THET+DTH
      IF (KSYMP.EQ.2.AND.THET.GT.90.01.AND.IFAR.NE.1) GO TO 109
      THA=THET*TA
      IF (IFAR.EQ.1) GO TO 89
      CALL FFLD (THA,PHA,ETH,EPH)
      GO TO 90
89    CALL GFLD (RFLD/WLAM,PHA,THET/WLAM,ETH,EPH,ERD,ZRATI,KSYMP)
      ERDM=CABS(ERD)
      ERDA=CANG(ERD)
90    ETHM2=REAL(ETH*CONJG(ETH))
      ETHM=SQRT(ETHM2)
      ETHA=CANG(ETH)
      EPHM2=REAL(EPH*CONJG(EPH))
      EPHM=SQRT(EPHM2)
      EPHA=CANG(EPH)
      IF (IFAR.EQ.1) GO TO 108
C     ELLIPTICAL POLARIZATION CALC.
      IF (ETHM2.GT.1.E-20.OR.EPHM2.GT.1.E-20) GO TO 91
      TILTA=0.
      EMAJR2=0.
      EMINR2=0.
      AXRAT=0.
      ISENS=HBLK
      GO TO 96
91    DFAZ=EPHA-ETHA
      IF (EPHA.LT.0.) GO TO 92
      DFAZ2=DFAZ-360.
      GO TO 93
92    DFAZ2=DFAZ+360.
93    IF (ABS(DFAZ).GT.ABS(DFAZ2)) DFAZ=DFAZ2
      CDFAZ=COS(DFAZ*TA)
      TSTOR1=ETHM2-EPHM2
      TSTOR2=2.*EPHM*ETHM*CDFAZ
      TILTA=.5*ATGN2(TSTOR2,TSTOR1)
      STILTA=SIN(TILTA)
      TSTOR1=TSTOR1*STILTA*STILTA
      TSTOR2=TSTOR2*STILTA*COS(TILTA)
      EMAJR2=-TSTOR1+TSTOR2+ETHM2
      EMINR2=TSTOR1-TSTOR2+EPHM2
      IF (EMINR2.LT.0.) EMINR2=0.
      AXRAT=SQRT(EMINR2/EMAJR2)
      TILTA=TILTA*TD
      IF (AXRAT.GT.1.E-5) GO TO 94
      ISENS=HPOL(1)
      GO TO 96
94    IF (DFAZ.GT.0.) GO TO 95
      ISENS=HPOL(2)
      GO TO 96
95    ISENS=HPOL(3)
96    GNMJ=DB10(GCON*EMAJR2)
      GNMN=DB10(GCON*EMINR2)
      GNV=DB10(GCON*ETHM2)
      GNH=DB10(GCON*EPHM2)
      GTOT=DB10(GCON*(ETHM2+EPHM2))
      IF (INOR.LT.1) GO TO 103
      I=I+1
      IF (I.GT.NORMAX) GO TO 103
      GO TO (97,98,99,100,101), INOR
97    TSTOR1=GNMJ
      GO TO 102
98    TSTOR1=GNMN
      GO TO 102
99    TSTOR1=GNV
      GO TO 102
100   TSTOR1=GNH
      GO TO 102
101   TSTOR1=GTOT
102   GAIN(I)=TSTOR1
      IF (TSTOR1.GT.GMAX) GMAX=TSTOR1
103   IF (IAVP.EQ.0) GO TO 104
      TSTOR1=GCOP*(ETHM2+EPHM2)
      TMP3=THA-TMP2
      TMP4=THA+TMP2
      IF (KTH.EQ.1) TMP3=THA
      IF (KTH.EQ.NTH) TMP4=THA
      DA=ABS(TMP1*(COS(TMP3)-COS(TMP4)))
      IF (KPH.EQ.1.OR.KPH.EQ.NPH) DA=.5*DA
      PINT=PINT+TSTOR1*DA
      IF (IAVP.EQ.2) GO TO 109
104   IF (IAX.EQ.1) GO TO 105
      TMP5=GNMJ
      TMP6=GNMN
      GO TO 106
105   TMP5=GNV
      TMP6=GNH
106   IF (RFLD.LT.1.E-20) GO TO 107
      ETHM=ETHM*EXRM
      ETHA=ETHA+EXRA
      EPHM=EPHM*EXRM
      EPHA=EPHA+EXRA
107   PRINT 176, THET,PHI,TMP5,TMP6,GTOT,AXRAT,TILTA,ISENS,ETHM,ETHA,
     1   EPHM,EPHA
      GO TO 109
108   PRINT 177, RFLD,PHI,THET,ETHM,ETHA,EPHM,EPHA,ERDM,ERDA
109   CONTINUE
```

```
      IF (IAVP.EQ.0) GO TO 110
      TMP3=THETS*TA
      TMP4=TMP3+DTH*TA*FLOAT(NTH-1)
      TMP3=ABS(DPH*TA*FLOAT(NPH-1)*(COS(TMP3)-COS(TMP4)))
      PINT=PINT/TMP3
      TMP3=TMP3/PI
      PRINT 178,  PINT,TMP3
110   IF (INOR.EQ.0) GO TO 114
      IF (ABS(GNOR).GT.1.E-20) GMAX=GNOR
      ITMP1=(INOR-1)*2+1
      ITMP2=ITMP1+1
      PRINT 179,  IGNTP(ITMP1),IGNTP(ITMP2),GMAX
      ITMP2=NPH*NTH
      IF (ITMP2.GT.NORMAX) ITMP2=NORMAX
      ITMP1=(ITMP2+2)/3
      ITMP2=ITMP1*3-ITMP2
      ITMP3=ITMP1
      ITMP4=2*ITMP1
      IF (ITMP2.EQ.2) ITMP4=ITMP4-1
      DO 111 I=1,ITMP1
      ITMP3=ITMP3+1
      ITMP4=ITMP4+1
      J=(I-1)/NTH
      TMP1=THETS+FLOAT(I-J*NTH-1)*DTH
      TMP2=PHIS+FLOAT(J)*DPH
      J=(ITMP3-1)/NTH
      TMP3=THETS+FLOAT(ITMP3-J*NTH-1)*DTH
      TMP4=PHIS+FLOAT(J)*DPH
      J=(ITMP4-1)/NTH
      TMP5=THETS+FLOAT(ITMP4-J*NTH-1)*DTH
      TMP6=PHIS+FLOAT(J)*DPH
      TSTOR1=GAIN(I)-GMAX
      IF (I.EQ.ITMP1.AND.ITMP2.NE.0) GO TO 112
      TSTOR2=GAIN(ITMP3)-GMAX
      PINT=GAIN(ITMP4)-GMAX
111   PRINT 180,  TMP1,TMP2,TSTOR1,TMP3,TMP4,TSTOR2,TMP5,TMP6,PINT
      GO TO 114
112   IF (ITMP2.EQ.2) GO TO 113
      TSTOR2=GAIN(ITMP3)-GMAX
      PRINT 180,  TMP1,TMP2,TSTOR1,TMP3,TMP4,TSTOR2
      GO TO 114
113   PRINT 180,  TMP1,TMP2,TSTOR1
114   IF (IXTYP.EQ.0.OR.IXTYP.EQ.4) GO TO 120
      NTHIC=NTHIC+1
      INC=INC+1
      XPR1=XPR1+XPR4
      IF (NTHIC.LE.NTHI) GO TO 55
      NTHIC=1
      XPR1=THETIS
      XPR2=XPR2+XPR5
      NPHIC=NPHIC+1
      IF (NPHIC.LE.NPHI) GO TO 55
      NPHIC=1
      XPR2=PHISS
      IF (IPTFLG.LT.2) GO TO 120
      ITMP1=NTHI*NPHI
      IF (ITMP1.LE.NORMF) GO TO 115
      ITMP1=NORMF
      PRINT 191
115   TMP1=FNORM(1)
      DO 116 J=2,ITMP1
      IF (FNORM(J).GT.TMP1) TMP1=FNORM(J)
116   CONTINUE
      PRINT 192,  TMP1,XPR3,HPOL(IXTYP),XPR6,ISAVE
      DO 119 J=1,NPHI
      ITMP2=NTHI*(J-1)
      DO 117 I=1,NTHI
      ITMP3=I+ITMP2
      IF (ITMP3.GT.ITMP1) GO TO 118
      TMP2=FNORM(ITMP3)/TMP1
      TMP3=DB20(TMP2)
      PRINT 193,  XPR1,XPR2,TMP3,TMP2
      XPR1=XPR1+XPR4
117   CONTINUE
118   XPR1=THETIS
      XPR2=XPR2+XPR5
119   CONTINUE
      XPR2=PHISS
120   IF (MHZ.EQ.NFRQ) IFAR=-1
      IF (NFRQ.NE.1) GO TO 121
      PRINT 135
      GO TO 15
121   MHZ=MHZ+1
      IF (MHZ.LE.NFRQ) GO TO 43
      IF (IPED.EQ.0) GO TO 124
      PRINT 194,  ISANT(NSANT),ZPNORM
      ITMP1=NFRQ
      IF (ITMP1.LE.(NORMF/4)) GO TO 122
      ITMP1=NORMF/4
      PRINT 195
122   IF (IFRQ.EQ.0) TMP1=FMHZ-(NFRQ-1)*DELFRQ
      IF (IFRQ.EQ.1) TMP1=FMHZ/(DELFRQ**(NFRQ-1))
      DO 123 I=1,ITMP1
      ITMP2=I+4*(I-1)
      TMP2=FNORM(ITMP2)/ZPNORM
      TMP3=FNORM(ITMP2+1)/ZPNORM
      TMP4=FNORM(ITMP2+2)/ZPNORM
      TMP5=FNORM(ITMP2+3)
      PRINT 196,  TMP1,FNORM(ITMP2),FNORM(ITMP2+1),FNORM(ITMP2+2),
```

```
     1 FNORM(ITMP2+3),TMP2,TMP3,TMP4,TMP5
      IF (IFRQ.EQ.0) TMP1=TMP1+DFLFRQ
      IF (IFRQ.EQ.1) TMP1=TMP1*DFLFRQ
  123 CONTINUE
      PRINT 135
  124 CONTINUE
      NFRQ=1
      MHZ=1
      GO TO 15
C
C
  125 FORMAT  (A2,13A6)
  126 FORMAT   (1H1)
  127 FORMAT  (///,33X,30H******************************,//,36X,24HANTEN
     1NA MODELING PROGRAM,//,33X,30H******************************)
  128 FORMAT  (////,37X,24H- - - - COMMENTS - - - -,/)
  129 FORMAT  (25X,13A6)
  130 FORMAT  (///,10X,34HINCORRECT LABEL FOR A COMMENT CARD)
  131 FORMAT  (////   33X,33H- - - SEGMENTATION DATA - - - -, //,40X,
     1  21HCOORDINATES IN METERS,//,25X,
     2  50HI+ AND I- INDICATE THE SEGMENTS BEFORE AND AFTER I,//)
  132 FORMAT  (2X,4HSEG.,3X,26HCOORDINATES OF SEG. CENTER,5X,4HSEG.,
     1  5X,18HORIENTATION ANGLES,4X,4HWIRE,4X,15HCONNECTION DATA,3X,
     2    3HTAG,/,2X,3HNO.,7X,1HX,9X,1HY,9X,1HZ,7X,6HLENGTH,5X,5HALPHA,
     3  5X,4HBETA,6X,6HRADIUS,4X,2HI-,3X,1HI,4X,2HI+,4X,3HNO.)
  133 FORMAT  (1X,I5,4F10.5,1X,3F10.5,1X,3I5,2X,I5)
  134 FORMAT  (19H SEGMENT DATA ERROR)
  135 FORMAT  (/////)
  136 FORMAT  (A2,I3,3I5,6E10.3)
  137 FORMAT  (1X, 19H***** DATA CARD NO.,I3,3X,A2,1X,I3,3(1X,I5),
     1  6(1X,E12.5))
  138 FORMAT  (///,10X,45HFAULTY DATA CARD LABEL AFTER GEOMETRY SECTION)
  139 FORMAT  (///,10X,48HNUMBER OF LOADING CARDS EXCEEDS STORAGE ALLOTT
     1ED)
  140 FORMAT  (///,10X,31HDATA FAULT ON LOADING CARD NO.=,I5,5X,11HITAG
     1STEP1=,I5,29H  IS GREATER THAN ITAG STEP2=,I5)
  141 FORMAT  (///,10X,51HNUMBER OF EXCITATION CARDS EXCEEDS STORAGE ALL
     1OTTED)
  142 FORMAT  (///,10X,48HNUMBER OF NETWORK CARDS EXCEEDS STORAGE ALLOTT
     1ED)
  143 FORMAT  (///,10X, 79HWHEN MULTIPLE FREQUENCIES ARE REQUESTED, ONLY
     1 ONE NEAR FIELD CARD CAN BE USED -,/,10X,22HLAST CARD READ IS USED
     2)
  144 FORMAT (10X,25HSEE USERS MANUAL SECTION ,A4)
  145 FORMAT  (////,33X,33H- - - - - - FREQUENCY - - - - - -,//,36X,10HF
     1REQUENCY=,E11.4,4H MHZ,/,36X,11HWAVELENGTH=,E11.4,7H METERS)
  146 FORMAT  (///,30X,40H - - - STRUCTURE IMPEDANCE LOADING - - -)
  147 FORMAT  (/ ,35X,28HTHIS STRUCTURE IS NOT LOADED)
  148 FORMAT  (///,34X,31H- - - ANTENNA ENVIRONMENT - - -,/)
  149 FORMAT  (40X,21HMEDIUM UNDER SCREEN -)
  150 FORMAT  (40X,27HRELATIVE DIELECTRIC CONST.=,F7.3,/,40X,13HCONDUCTI
     1VITY=,E10.3,11H MHOS/METER)
  151 FORMAT  (  42X,14HPERFECT GROUND)
  152 FORMAT  (  44X,10HFREE SPACE)
  153 FORMAT  (///,32X,25H- - - MATRIX TIMING - - -,//,24X,5HFILL=,F9.3,
     1 15H SEC.,  FACTOR=,F9.3,5H SEC.)
  154 FORMAT  (///,40X,22H- - - EXCITATION - - -)
  155 FORMAT  (/,4X,10HPLANE WAVE,4X,6HTHETA=,F7.2,11H DEG,  PHI=,F7.2,
     1  11H DEG,  ETA=,F7.2,13H DEG,  TYPE -,A6,15H,  AXIAL RATIO=,F6.3)
  156 FORMAT  (/,31X,17HPOSITION (METERS),14X,18HORIENTATION (DEG)=/,28X
     1,1HX,12X,1HY,12X,1HZ,10X,5HALPHA,5X,4HBETA,4X,13HDIPOLE MOMENT,//
     2  ,4X,14HCURRENT SOURCE,1X,3(3X,F10.5),1X,2(3X,F7.2),4X,F8.3)
  157 FORMAT  (4X,4(I5,1X),6(3X,E11.4),3X,A6,A2)
  158 FORMAT  (///,44X,24H- - - NETWORK DATA - - -)
  159 FORMAT  (/,6X,18H- FROM -     - TO -,11X,17HTRANSMISSION LINE,15X,3
     16H-  -  SHUNT ADMITTANCES (MHOS)  -  -,14X,4HLINE,/,6X,21HTAG  SEG
     2.  TAG  SEG.,6X,9HIMPEDANCE,6X,6HLENGTH,12X,11H- END ONE -,17X,11
     3H- END TWO -,12X,4HTYPE,/    ,6X,21HNO.   NO.   NO.   NO.,9X,4HOHM
     4S,8X,6HMETERS,9X, 4HREAL,10X,5HIMAG.,9X,4HREAL,10X,5HIMAG.)
  160 FORMAT  (/,6X,8H- FROM -,4X,6H- TO -,26X,45H-  -  ADMITTANCE MATRI
     1X ELEMENTS (MHOS)  -  -,/    ,6X,21HTAG  SEG.   TAG  SEG.,13X,9H(O
     2NE,ONE),19X,    9H(ONE,TWO),19X,9H(TWO,TWO),/     ,6X,21HNO.   NO.
     3   NO.    NO.,8X,4HREAL,10X,5HIMAG.,9X,4HREAL,10X,5HIMAG.,9X,4HREAL
     4, 10X,5HIMAG.)
  161 FORMAT  (///,29X,33H- - - CURRENTS AND LOCATION - - -,//,33X,24HDI
     1STANCES IN WAVELENGTHS)
  162 FORMAT  (  //,2X,4HSEG.,2X,3HTAG,4X,21HCOORD. OF SEG. CENTER,5X,
     1  4HSEG.,12X,26H- - - CURRENT (AMPS) - - -,/,2X,3HNO.,3X,3HNO.,
     2  5X,1HX,8X,1HY,8X,1HZ,6X,6HLENGTH,5X,4HREAL,8X,5HIMAG.,7X,4HMAG.,
     3  8X,5HPHASE)
  163 FORMAT  (///,33X,40H- - - RECEIVING PATTERN PARAMETERS - - -,/  ,4
     13X,4HETA=,F7.2,8H DEGREES,/,43X,6HTYPE -,A6,/,43X,12HAXIAL RATIO=,
     2  F6.3,//    ,11X,5HTHETA,6X,3HPHI,10X,13H-  CURRENT  -,9X,3HSEG,/
     3  ,11X,5H(DEG),5X,5H(DEG),7X,9HMAGNITUDE,4X,5HPHASE,6X,3HNO.,/)
  164 FORMAT  (10X,2(F7.2,3X),1X,E11.4,3X,F7.2,4X,I5)
  165 FORMAT  (1X,2I5,3F9.4,F9.5,1X,3E12.4,F9.3)
  166 FORMAT  (///,40X,24H- - - POWER BUDGET - - -,//      ,43X,15HINPUT P
     1OWER   =,E11.4,6H WATTS,/ ,43X,15HRADIATED POWER=,E11.4,6H WATTS,/
     2  ,43X,15HSTRUCTURE LOSS=,E11.4,6H WATTS,/    ,43X,15HNETWORK LOSS
     3 =,E11.4,6H WATTS,/,43X,15HEFFICIENCY    =, F7.2,8H PERCENT)
  167 FORMAT  (///,39X,23H- - - NEAR FIELDS - - -,//    ,12X,14H-  LOCATI
     1ON  -,21X     ,8H-  EX  -,15X,8H-  EY  -,15X,8H-  EZ  -,/      ,8X,
     21HX,10X,1HY,10X,1HZ,10X,    9HMAGNITUDE,3X,5HPHASE,6X,9HMAGNITUDE,
     33X,5HPHASE,6X,9HMAGNITUDE,3X,5HPHASE,/     ,6X,6HMETERS,5X,6HMETER
     4S,5X,6HMETERS,8X,7HVOLTS/M,3X,7HDEGREES,6X,7HVOLTS/M,3X,7HDEGREES,
     56X,7HVOLTS/M,3X,7HDEGREES)
  168 FORMAT  (2X,3(2X,F9.4),1X,3(3X,E11.4,2X,F7.2))
  169 FORMAT  (///,31X,39H- - - FAR FIELD GROUND PARAMETERS - - -,//)
  170 FORMAT  (40X,25HRADIAL WIRE GROUND SCREEN,/,40X,    I5,6H WIRES,/,4
```

```
     1X,12HWIRE LENGTH=,F8.3,7H METERS,/,40X,13HWIRE RADIUS=,E10.3,7H M  WA 974
     2ETERS)                                                             WA 975
171  FORMAT  (40X,4H,5H CLIFF,/,40X,14HEDGE DISTANCE=,F9.2,7H METERS,/, WA 976
     1  40X,7HHEIGHT=,F8.2,7H METERS,/,                                  WA 977
     2  40X,15HSECOND MEDIUM -   ,/,40X,27HRELATIVE DIELECTRIC CONST.=,F WA 978
     37.3,/,40X,13HCONDUCTIVITY=,E10.3,5H MHOS)                          WA 979
172  FORMAT  (  ///,48X,30H- - - RADIATION PATTERNS - - -,/)             WA 980
173  FORMAT  (   5X,5HRANGE=,E13.6,7H METERS,/,5X,16H2.*PI*EXP(-JKR)/   WA 981
     1,  5H(KR)=,E12.5,9H AT PHASE,F7.2,9H DEGREES,/)                    WA 982
174  FORMAT  (  4X,14H- - ANGLES - -,8X,2A6,7HGAINS -,7X,13H- - - POLAR  WA 983
     1IZ 11HATION - - -,20H- - E(THETA) - - -,1H- - - E(PHI) -           WA 984
     2-, 2H -,/,6X,5HTHETA,5X,3HPHI,7X,A6,3X,A6,4X,5HTOTAL,6X,5HAXIAL,   WA 985
     3   5X,4HTILT,3X,5HSENSE,5X,9HMAGNITUDE,4X,6HPHASE ),/,1X,          WA 986
     -   2(2X,7HDEGREES),3X,2HDB,8X,5HRATIO,5X,4HDEG.,8X,                WA 987
     5   2(6X,7HVOLTS/M,4X,7HDEGREES))                                   WA 988
175  FORMAT  (///,28X,42H- - - RADIATED FIELDS NEAR GROUND - - -,//,8X  WA 989
     1, 20H- - - LOCATION - - -,10X,14H- - E(THETA) - -,8X,14H- - E(PHI) WA 990
     2 - -,8X,17H- - E(RADIAL) - -,                                      WA 991
     3    /  ,7X,3HRHO,6X,3HPHI,9X,1HZ,12X,3HMAG,6X,5HPHASE,9X,3HMAG,6X  WA 992
     4  ,5HPHASE,9X,3HMAG,6X,5HPHASE,/,5X,6HMETERS,3X,7HDEGREES,4X,6HMET WA 993
     5ERS,8X  ,7HVOLTS/M,3X,7HDEGREES,6X,7HVOLTS/M,3X,7HDEGREES,  6X,7HV WA 994
     6OLTS/M,3X,7HDEGREES,/)                                             WA 995
176  FORMAT  (1X,2F9.2,2X,F8.3,2X,F9.5,F9.2,2X,A6,2(3X,E12.5,F9.2))      WA 996
177  FORMAT  (3X,F9.2,2X,F7.2,2X,F9.2,1X,3(3X,E11.4,2X,F7.2))            WA 997
178  FORMAT  (  //,3X,19HAVERAGE POWER GAIN=,E12.5,7X,16HSOLID ANGLE US WA 998
     1ED 15H IN AVERAGING=(,F7.4,16H)*PI STERADIANS.,//)                 WA 999
179  FORMAT  (  //,37X,31H- - - - NORMALIZED GAIN - - - -,//,37X,2A6,   WA1000
     1   4HGAIN,/,38X,22HNORMALIZATION FACTOR =,F9.2,3H DB,//,3(14X,     WA1001
     2   14H- - ANGLES - -,6X,4HGAIN,7X),/,3(4X,5HTHETA,5X,3HPHI,8X,     WA1002
     3   2HDB,8X),/,3(3X,7HDEGREES,2X,7HDEGREES,16X))                    WA1003
180  FORMAT  (3(1X,2F9.2,1X,F9.2,6X))                                    WA1004
181  FORMAT  (///,4X,51HRECEIVING PATTERN STORAGE TOO SMALL,ARRAY TRUNC  WA1005
     1ATED)                                                              WA1006
182  FORMAT  (///,32X,47H- - - NORMALIZED RECEIVING PATTERN - - -,/,41X  WA1007
     1, 21HNORMALIZATION FACTOR=,E11.4,/,41X,6HETA=,F7.2,8H DEGREES,/,41 WA1008
     2X,6HTYPE -,A6,/,41X,15HAXIAL RATIO=,F6.3,/,41X,12HSEGMENT NO.=,I5, WA1009
     3//,21X,5HTHETA,6X,3HPHI,13X,13H- PATTERN -,/,21X,5H(DEG),5X,5H(DE  WA1010
     4G),8X,2HDB,8X,9HMAGNITUDE,/)                                       WA1011
183  FORMAT  (20X,2(F7.2,3X),1X,F7.2,4X,E11.4)                           WA1012
184  FORMAT  (///,36X,32H- - - INPUT IMPEDANCE DATA - - -,/   ,45X,18HS  WA1013
     1OURCE SEGMENT NO.,I4,/  ,45X,21HNORMALIZATION FACTOR=,E12.5,//     WA1014
     2 ,7X,5HFREQ.,13X,34H- - UNNORMALIZED IMPEDANCE - -,21X,   32H-     WA1015
     3 - NORMALIZED IMPEDANCE  - -,/   ,19X,10HRESISTANCE,4X,9HREACT     WA1016
     4ANCE,6X,9HMAGNITUDE,4X,5HPHASE,7X,10HRESISTANCE,4X,9HREACTANCE,6X, WA1017
     5  9HMAGNITUDE,4X,5HPHASE,/    ,2X,3HMHZ,11X,4HOHMS,10X,4HOHMS,11X, WA1018
     6  4HOHMS,5X,7HDEGREES,47X,7HDEGREES,/)                             WA1019
185  FORMAT  (///,4X,62HSTORAGE FOR IMPEDANCE NORMALIZATION TOO SMALL,   WA1020
     1ARRAY TRUNCATED)                                                   WA1021
186  FORMAT  (4X,F9.3,2X,2(2X,E12.5),3X,E12.5,2X,F7.2,2X,2(2X,E12.5),3X WA1022
     1, E12.5,2X,F7.2)                                                   WA1023
187  FORMAT(  ////,20X,55HAPPROXIMATE INTEGRATION EMPLOYED FOR SEGMENTS
     1 MORE THAN,F8.3,18H WAVELENGTHS APART)
213  FORMAT(45H SYMMETRY DESTROYED BY JUNCTION SPECIFICATION)
     END                                                                 WA1024-
```

```
      SUBROUTINE  CABC
C
C     CABC COMPUTES COEFFICIENTS OF THE CONSTANT (A), SINE (B), AND
C     COSINE (C) TERMS IN THE CURRENT INTERPOLATION FUNCTIONS FOR THE
C     CURRENT VECTOR CUR.
C
      COMMON /DATA/  N,NP,X(800),Y(800),Z(800),SI(800),BI(800),ALP(800),
     1 BET(800),ICON1(800),ICON2(800),ITAG(800),WLAM,IPSYM,JMAX
      COMMON /CRNT/  AIR(800),AII(800),BIR(800),BII(800),CIR(800),CII(80
     10) ,CUR(800)
      COMMON /JUNK/ NCOX,JOX(25),NCIX,JIX(25),NCOZ,JOZ(25),NCIZ,JIZ(25)
      COMPLEX  CUR,CLO,CLL,CLY,AX,BX,CX
      DATA  TP/6.283185308/
      DO 21 I=1,N
      CALL TPIO(I,JCO1,JCO2,DIL,DIK)
      CL=TP*DIL
      CK=TP*DIK
      SINL=SIN(CL)
      COSL=COS(CL)
      SINK=SIN(CK)
      COSK=COS(CK)
      IF(JCO1.GT.(-90000))GO TO 5
      JIXK=N-(JCO1+90000)
      CLO=CUR(JIXK)
      GO TO 8
5     IF(JCO1)1,6,7
1     CLO=(0.,0.)
      IF(NCIX.LT.1)GO TO 3
      DO 2 K=1,NCIX
      JIXK=JIX(K)
2     CLO=CLO+CUR(JIXK)
3     IF(NCOX.LT.1)GO TO 8
      DO 4 K=1,NCOX
      JOXK=JOX(K)
4     CLO=CLO-CUR(JOXK)
      GO TO 8
6     CLO=(0.,0.)
      GO TO 8
7     CLO=CUR(JCO1)
      IF(ICON2(JCO1).NE.I.AND.JCO1.NE.I)CLO=-CLO
8     CLL=CUR(I)
      IF(JCO2.GT.(-90000))GO TO 13
      JIXK=N-(JCO2+90000)
      CLY=CUR(JIXK)
      GO TO 16
13    IF(JCO2)9,14,15
9     CLY=(0.,0.)
      IF(NCOZ.LT.1)GO TO 11
      DO 10 K=1,NCOZ
      JOZK=JOZ(K)
10    CLY=CLY+CUR(JOZK)
11    IF(NCIZ.LT.1)GO TO 16
      DO 12 K=1,NCIZ
      JIZK=JIZ(K)
12    CLY=CLY-CUR(JIZK)
      GO TO 16
14    CLY=(0.,0.)
      GO TO 16
15    CLY=CUR(JCO2)
      IF(ICON1(JCO2).NE.I.AND.JCO2.NE.I)CLY=-CLY
16    IF(JCO1.LT.(-90000))GO TO 17
      IF(JCO2.LT.(-90000))GO TO 18
      SILK=SINL*COSK+COSL*SINK
      CELLO=SINL+SINK-SILK
      AX=(CLO*SINK-CLL*SILK+CLY*SINL)/CELLO
      BX=(CLO*(COSK-1.)+CLL*(COSL-COSK)+CLY*(1.-COSL))/CELLO
      GO TO 20
17    IF(JCO2.LT.(-90000))GO TO 19
      SILK=COSL*COSK-SINL*SINK
      CELLO=SILK-COSL
      AX=(CLO*SINK+CLL*SILK-CLY*COSL)/CELLO
      BX=(CLO*(COSK-1.)+(CLL-CLY)*SINL)/CELLO
      GO TO 20
18    SILK=COSL*COSK-SINL*SINK
      CELLO=COSK-SILK
      AX=(CLO*COSK-CLL*SILK+CLY*SINL)/CELLO
      BX=((CLL-CLO)*SINK+CLY*(1.-COSL))/CELLO
      GO TO 20
19    CELLO=2.*SINL*COSL
      AX=CLL+(CLY-CLO)*COSL/CELLO
      BX=(CLY+CLO)*SINL/CELLO
20    CX=CLL-AX
      AIR(I)=REAL(AX)
      AII(I)=AIMAG(AX)
      BIR(I)=REAL(BX)
      BII(I)=AIMAG(BX)
      CIR(I)=REAL(CX)
21    CII(I)=AIMAG(CX)
      RETURN
      END
```

```
      SUBROUTINE CMSET (NROW,NCOL,CM,NLOAD,RKH)
C
C     CMSET SETS UP THE COMPLEX STRUCTURE MATRIX IN THE ARRAY CM
C
      COMMON /DATA/  N,NP,X(800),Y(800),Z(800),SI(800),BI(800),ALP(800),
     1 BET(800),ICON1(800),ICON2(800),ITAG(800),WLAM,IPSYM,JMAX
      DIMENSION CM(NROW,NCOL)
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      COMMON /RESTRT/  IC1,IC2,IC3,NPES,NPRES,IHLCK
      COMMON /ANGL/  SALP(800)
      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
      COMMON /ZLOAD/  ZARRAY(800)
      COMMON /GND/ ZRATI,ZRATI2,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,IPERF
      DIMENSION CAB(1), SAB(1)
      DIMENSION ETR(3), ETI(3),EZE(2),EPE(2)
      COMPLEX  FJ,CM,ZARRAY,C1,EP,ET,EZ,ER,CE1,CE2,CE3
      COMPLEX  ZRATI,REFS,REFPS,ZRSIN,ZRATIS,T1,ZSCRN,ZRATI2
      EQUIVALENCE (CAB(1),ALP(1)), (SAB(1),BET(1)),(EZ,EZE),(EP,EPE)
      DATA ETA/376.73/,PI2/6.2831853/,PI/3.14159265/
C     FUNCTIONS - -
      JSEQ(J)=J+(J-1)/NP*JPMAX
      JJEQ(J)=-J-90000+((-J-90001)/JPMAX+1)*NP
C
      NOP=N/NP
      JPMAX=JMAX/NOP
      JEQN=0
      IF (ICASE.GT.2) REWIND  11
      FJ=(0.,1.)
      I2=2*NPBLK*NROW
      IBLCK=I2
      IT=NPBLK
      NI1=IC1+1
      IF ( IC1 .EQ. 0 ) GO TO 1
      IF (IC1.LE.(-1)) GO TO 22
      CALL BLCKIN (11,1,I2,IC1,1)
1     CONTINUE
      DO 21 IXBLK1=NI1,NBLOKS
      ISV=( IXBLK1 - 1)*NPBLK
      IF (IXBLK1.EQ.NBLOKS) IT=NLAST
      IF ( ICASE .LT. 3 ) IT=NCOL
      IF (NRADL.EQ.0) GO TO 2
      T1=FJ*2367.067/FLOAT(NRADL)
      T2=SCRWR*FLOAT(NRADL)
      ZRATIS=ZRATI
2     DO 3 I=1,IT
      DO 3 J=1,NROW
3     CM(J,I)=(0.,0.)
      ITL=IT
      IF(ISV.GE.NP)GO TO 60
      IF(ISV+IT.GT.NP)ITL=NP-ISV
C
C     SOURCE SEGMENT LOOP
C
      DO 18 J=1,N
      CALL TRIO(J,JCO1,JCO2,DIL,DIK)
      S=SI(J)
      B=BI(J)
      XJ=X(J)
      YJ=Y(J)
      ZJ=Z(J)
      CABJ=CAB(J)
      SABJ=SAB(J)
      SALPJ=SALP(J)
C
C     OBSERVATION SEGMENT LOOP
C
      DO 18 IPR = 1, ITL
      I=ISV+IPR
      XIJ=X(I)-XJ
      YIJ=Y(I)-YJ
      IJ=I-J
      CABI=CAB(I)
      SABI=SAB(I)
      SALPI=SALP(I)
      RFL=-1.
C
C     LOOP TO INCLUDE IMAGE OF SOURCE SEGMENT FOR STRUCTURE OVER GROUND.
C
      DO 18 IP=1,KSYMP
      RFL=-RFL
      SALPR=SALPJ*RFL
      ZIJ=Z(I)-RFL*ZJ
      ZP=XIJ*CABJ+YIJ*SABJ+ZIJ*SALPR
      DIJ=CABI*CABJ+SABI*SABJ+SALPI*SALPR
      RHOX=XIJ-CABJ*ZP
      RHOY=YIJ-SABJ*ZP
      RHOZ=ZIJ-SALPR*ZP
      RH=SQRT(RHOX*RHOX+RHOY*RHOY+RHOZ*RHOZ)
      IF (RH.GT.1.E-6) GO TO 4
      RHOX=0.
      RHOY=0.
      RHOZ=0.
      DIR=0.
      GO TO 5
```

```
4     RHOX=RHOX/RH
      RHOY=RHOY/RH
      RHOZ=RHOZ/RH
      DIR=RHOX*CABI+RHOY*SABI+RHOZ*SALPI
5     R=SQRT(XIJ*XIJ+YIJ*YIJ+ZIJ*ZIJ)
      IF(IP.NE.2)GO TO 10
      IJ=1
      IF (IPERF.EQ.1) GO TO 10
      RMAG=R
      XYMAG=SQRT(XIJ*XIJ+YIJ*YIJ)
C
C     SET PARAMETERS FOR RADIAL WIRE GROUND SCREEN.
C
      IF (NRADL.EQ.0) GO TO 7
      XSPEC=(X(I)*ZJ+Z(I)*XJ)/(Z(I)+ZJ)
      YSPEC=(Y(I)*ZJ+Z(I)*YJ)/(Z(I)+ZJ)
      RHOSPC=SQRT(XSPEC*XSPEC+YSPEC*YSPEC+T2*T2)
      IF (RHOSPC.GT.SCRWL) GO TO 6
      ZSCRN=T1*RHOSPC*ALOG(RHOSPC/T2)
      ZRATI=(ZSCRN*ZRATIS)/(ETA*ZRATIS+ZSCRN)
      GO TO 7
6     ZRATI=ZRATIS
7     IF (XYMAG.GT.1.E-6) GO TO 8
C
C     CALCULATION OF REFLECTION COEFFICIENTS WHEN GROUND IS SPECIFIED.
C
      PX=0.
      PY=0.
      CTH=1.
      ZRSIN=(1.,0.)
      GO TO 9
8     PX=-YIJ/XYMAG
      PY=XIJ/XYMAG
      CTH=ZIJ/RMAG
      ZRSIN=CSQRT(1.-ZRATI*ZRATI*(1.-CTH*CTH))
9     REFS=-(CTH-ZRATI*ZRSIN)/(CTH+ZRATI*ZRSIN)
      REFPS=(ZRATI*CTH-ZRSIN)/(ZRATI*CTH+ZRSIN)
      REFPS=REFPS-REFS
10    IF(R.GT.RKH)GO TO 58
      CALL INTG(R,S,RH,ZP,DIJ,DIR,ETR,ETI,DIL,DIK,IJ,IP,JCO1,JCO2,XM,YM,
     1 ZM)
C
C     FILL MATRIX ELEMENTS.  ELEMENT LOCATIONS DETERMINED BY CONNECTION
C     DATA.
C
      CE1=CMPLX(ETR(1),ETI(1))
      CE2=CMPLX(ETR(2),ETI(2))
      CE3=CMPLX(ETR(3),ETI(3))
      IF(JCO1.GT.0)JE1=JSEQ(JCO1)
      IF(JCO1.LT.(-90000))JE1=JJEQ(JCO1)
      JE2=JSEQ(J)
      IF(JCO2.GT.0)JE3=JSEQ(JCO2)
      IF(JCO2.LT.(-90000))JE3=JJEQ(JCO2)
      IF(JCO1.GT.(-90000))GO TO 24
      IF(JCO2.LT.(-90000))GO TO 26
      CE2=CE2-YM/XM*CE1
      CE3=CE3-ZM/XM*CE1
      CM(JE1,IPR)=CM(JE1,IPR)+CE1/XM
      GO TO 27
24    IF(JCO2.GT.(-90000))GO TO 27
      CE2=CE2-YM/XM*CE3
      CE1=CE1-ZM/XM*CE3
      CM(JE3,IPR)=CM(JE3,IPR)-CE3/XM
      GO TO 27
26    DEN=1./(XM*XM-ZM*ZM)
      CE2=CE2+CE1-CE3
      CM(JE3,IPR)=CM(JE3,IPR)+DEN*(ZM*CE1-XM*CE3)
      CM(JE1,IPR)=CM(JE1,IPR)+DEN*(XM*CE1-ZM*CE3)
27    IF(JCO1.LT.(-90000))GO TO 29
      IF(JCO1)11,29,12
11    CALL JMELS(CE1,NCIX,JIX,NCOX,JOX,IPR,CM,NROW,NCOL,NP,JPMAX)
      GO TO 29
12    IF(ICON2(JCO1).EQ.J)GO TO 28
      IF(JCO1.EQ.J)GO TO 28
      CM(JE1,IPR)=CM(JE1,IPR)-CE1
      GO TO 29
28    CM(JE1,IPR)=CM(JE1,IPR)+CE1
29    CM(JE2,IPR)=CM(JE2,IPR)-CE2
      IF(JCO2.LT.(-90000))GO TO 18
      IF(JCO2)13,18,14
13    CALL JMELS(CE3,NCOZ,JOZ,NCIZ,JIZ,IPR,CM,NROW,NCOL,NP,JPMAX)
      GO TO 18
14    IF(ICON1(JCO2).EQ.J)GO TO 30
      IF(JCO2.EQ.J)GO TO 30
      CM(JE3,IPR)=CM(JE3,IPR)-CE3
      GO TO 18
30    CM(JE3,IPR)=CM(JE3,IPR)+CE3
      GO TO 18
58    RKH1=PI2*R
      A0=ZP/R
      A1=SQRT(ABS(1.-A0*A0))
      C1=CMPLX(COS(RKH1),-SIN(RKH1))
      ER=S*ETA*A0*C1*CMPLX(1.,-1./RKH1)/(PI2*R*R)
      ET=S*ETA*A1*C1*CMPLX(1.,RKH1-1./RKH1)/(2.*PI2*R*R)
      EZ=ER*A0-ET*A1
      EP=ER*A1+ET*A0
      IF(IP.EQ.2)CALL GN(EZE(1),EZE(2),EPE(1),EPE(2))
      JE2=JSEQ(J)
      CM(JE2,IPR)=CM(JE2,IPR)+DIJ*EZ+DIR*EP
```

```
18    CONTINUE                                                          CM 145
      IF(IT.EQ.ITL)GO TO 59
      ITL=ITL+1
      GO TO 61
60    ITL=1
61    DO 66 IPR=ITL,IT
      JEON=JEON-1
      I=NP-JEON
      DO 66 J=1,N
      JCO1=ICON1(J)
      JCO2=ICON2(J)
      IF(JCO1+90000.NE.JEON)GO TO 62
      JCAS=2
      IF(JCO2.LT.(-90000))JCAS=4
      GO TO 63
62    IF(JCO2+90000.NE.JEON)GO TO 66
      JCAS=1
      IF(JCO1.LT.(-90000))JCAS=3
63    CALL TRIO(J,JCO1,JCO2,DIL,DIK)
      DIL=PI2*DIL
      DIK=PI2*DIK
      DEN=SIN(DIL)+SIN(DIK)-SIN(DIL+DIK)
      YM=COS(DIL+DIK)
      IF(DIK.LT.DIL)DIL=DIK
      ZM=COS(DIL)
      XM=(YM-ZM)/DEN
      YM=(1.-YM)/DEN
      ZM=(ZM-1.)/DEN
      GO TO (67,68,69,70),JCAS
67    IF(JCO1.GT.0)JE1=JSEQ(JCO1)
      JE2=JSEQ(J)
      CM(JE2,IPR)=CM(JE2,IPR)+YM/XM
      CM(I,IPR)=CM(I,IPR)+1./XM
      IF(JCO1)64,66,73
64    CE1=ZM/XM
      CALL JMELS(CE1,NCIX,JIX,NCOX,JOX,IPR,CM,N,NCOL,NP,JPMAX)
      GO TO 66
73    IF(ICON2(JCO1).EQ.J)GO TO 71
      IF(JCO1.EQ.J)GO TO 71
      CM(JE1,IPR)=CM(JE1,IPR)-ZM/XM
      GO TO 66
71    CM(JE1,IPR)=CM(JE1,IPR)+ZM/XM
      GO TO 66
68    JE2=JSEQ(J)
      IF(JCO2.GT.0)JE3=JSEQ(JCO2)
      CM(JE2,IPR)=CM(JE2,IPR)-YM/XM
      CM(I,IPR)=CM(I,IPR)+1./XM
      IF(JCO2)74,66,75
74    CE3=-ZM/XM
      CALL JMELS(CE3,NCOZ,JOZ,NCIZ,JIZ,IPR,CM,N,NCOL,NP,JPMAX)
      GO TO 66
75    IF(ICON1(JCO2).EQ.J)GO TO 72
      IF(JCO2.EQ.J)GO TO 72
      CM(JE3,IPR)=CM(JE3,IPR)+ZM/XM
      GO TO 66
72    CM(JE3,IPR)=CM(JE3,IPR)-ZM/XM
      GO TO 66
69    JE2=JSEQ(J)
      JE1=NP-(JCO1+90000)
      DEN=1./(ZM*ZM-XM*XM)
      CM(JE2,IPR)=CM(JE2,IPR)-1.
      CM(I,IPR)=CM(I,IPR)-XM*DEN
      CM(JE1,IPR)=CM(JE1,IPR)-ZM*DEN
      GO TO 66
70    JE2=JSEQ(J)
      JE1=NP-(JCO2+90000)
      DEN=1./(ZM*ZM-XM*XM)
      CM(JE2,IPR)=CM(JE2,IPR)+1.
      CM(I,IPR)=CM(I,IPR)-XM*DEN
      CM(JE1,IPR)=CM(JE1,IPR)-ZM*DEN
66    CONTINUE
59    IF (NRADL.NE.0) ZRATI=ZRATIS                                      CM 146
C                                                                       CM 147
C     MATRIX ELEMENTS MODIFIED BY LOADING                               CM 148
C                                                                       CM 149
      IF (NLOAD.EQ.0) GO TO 20                                          CM 150
      DO 19 I=1,IT                                                      CM 151
      J=ISV+I                                                           CM 152
      IF(J.GT.NP)GO TO 20
19    CM(J,I)=CM(J,I)-ZARRAY(J)                                         CM 153
20    IF (ICASE.LT.3) GO TO 21                                          CM 154
      CALL BLCKOT (11,1,I2+1,31)                                        CM 155
      IC1=IXBLK1                                                        CM 156
      IF (IC1.EQ.NBLOKS) GO TO 21                                       CM 157
      CALL  CHKPRT                                                      CM 158
21    CONTINUE                                                          CM 159
      IF (ICASE.LT.3) GO TO 22                                          CM 160
      REWIND 11                                                         CM 161
      IC1=-2                                                            CM 162
      IF (ICASE.EQ.3) IC1=-1                                            CM 163
      CALL  CHKPRT                                                      CM 164
22    RETURN                                                            CM 165
      END                                                               CM 166-
```

```
      SUBROUTINE  DATAGN
C
C     DATAGN IS THE MAIN ROUTINE FOR INPUT OF GEOMETRY DATA.
C
      COMMON /DATA/  N,NP,X(800),Y(800),Z(800),SI(800),BI(800),ALP(800),
     1 BET(800),ICON1(800),ICON2(800),ITAG(800),WLAM,IPSYM,JMAX
      DIMENSION X2(1), Y2(1), Z2(1)
      DIMENSION ATST(8), IFX(2), IFY(2), IFZ(2)
      INTEGER  GM,ATST
      EQUIVALENCE (X2(1),SI(1)), (Y2(1),ALP(1)), (Z2(1),BET(1))
      DATA  ATST/2HGW,2HGX,2HGR,2HGS,2HGE,2HGM,2HJX,2HJE/
      DATA  IFX/1H ,1HX/,IFY/1H ,1HY/,IFZ/1H ,1HZ/
      DATA  TA/0.01745329252/
      IPSYM=0
      NWIRE=0
      N=0
      PRINT 11
      PRINT 12
C
C     READ GEOMETRY DATA CARD AND BRANCH TO SECTION FOR OPERATION
C     REQUESTED
C
1     READ 13, GM,ITG,NS,XW1,YW1,ZW1,XW2,YW2,ZW2,RAD
      IF (GM.EQ.ATST(1)) GO TO 2
      IF (GM.EQ.ATST(2)) GO TO 3
      IF (GM.EQ.ATST(3)) GO TO 4
      IF (GM.EQ.ATST(4)) GO TO 6
      IF (GM.EQ.ATST(5)) GO TO 9
      IF (GM.EQ.ATST(6)) GO TO 8
      GO TO 10
C
C     GENERATE SEGMENT DATA FOR STRAIGHT WIRE.
C
2     NWIRE=NWIRE+1
      I1=N+1
      I2=N+NS
      PRINT 14, NWIRE,XW1,YW1,ZW1,XW2,YW2,ZW2,RAD,NS,I1,I2,ITG
      CALL WIRE (XW1,YW1,ZW1,XW2,YW2,ZW2,RAD,NS,ITG)
      GO TO 1
C
C     REFLECT STRUCTURE ALONG X,Y, OR Z AXES OR ROTATE TO FORM CYLINDER.
C
3     IY=NS/10
      IZ=NS-IY*10
      IX=IY/10
      IY=IY-IX*10
      IF (IX.NE.0) IX=1
      IF (IY.NE.0) IY=1
      IF (IZ.NE.0) IZ=1
      PRINT 15, IFX(IX+1),IFY(IY+1),IFZ(IZ+1),ITG
      GO TO 5
4     PRINT 16, NS,ITG
      IX=-1
5     CALL REFLC (IX,IY,IZ,ITG,NS)
      GO TO 1
C
C     SCALE STRUCTURE DIMENSIONS BY FACTOR XW1.
C
6     DO 7 I=1,N
      X(I)=X(I)*XW1
      Y(I)=Y(I)*XW1
      Z(I)=Z(I)*XW1
      X2(I)=X2(I)*XW1
      Y2(I)=Y2(I)*XW1
      Z2(I)=Z2(I)*XW1
7     BI(I)=BI(I)*XW1
      PRINT 17, XW1
      GO TO 1
C
C     MOVE STRUCTURE OR REPRODUCE ORIGINAL STRUCTURE IN NEW POSITIONS.
C
8     PRINT 18, ITG,NS,XW1,YW1,ZW1,XW2,YW2,ZW2,RAD
      XW1=XW1*TA
      YW1=YW1*TA
      ZW1=ZW1*TA
      CALL MOVE (XW1,YW1,ZW1,XW2,YW2,ZW2,INT(RAD+.5),NS,ITG)
      GO TO 1
C
C     TERMINATE STRUCTURE GEOMETRY INPUT.
C
9     CALL CONECT (ITG)
      CALL  CONVRT
      JMAX=0
      IF(NS.EQ.0)RETURN
      PRINT 90
111   READ 91,GM,IX,IY,IZ
      PRINT 93,GM,IX,IY,IZ
      IF(GM.EQ.ATST(7))GO TO 21
      IF(GM.NE.ATST(8))GO TO 212
      IF(JMAX.EQ.0)GO TO 208
      RETURN
212   PRINT 92
      STOP
21    IF(IX.EQ.0.AND.IY.EQ.0)GO TO 208
      I=ISEGNO(IX,IY)
```

```
      IF(IZ.EQ.2)GO TO 203
      IX=ICON1(I)
      IF(IX.LT.(-90000))GO TO 111
      IF(IX.EQ.0.OR.IX.EQ.I)GO TO 111
      JMAX=JMAX+1
      IZ=-(JMAX+90000)
      ICON1(I)=IZ
      GO TO 204
203   IX=ICON2(I)
      IF(IX.LT.(-90000))GO TO 111
      IF(IX.EQ.0.OR.IX.EQ.I)GO TO 111
      JMAX=JMAX+1
      IZ=-(JMAX+90000)
      ICON2(I)=IZ
204   IF(IX.LT.0)GO TO 205
      IF(ICON1(IX).EQ.I)GO TO 206
      ICON2(IX)=IZ
      GO TO 111
206   ICON1(IX)=IZ
      GO TO 111
205   DO 207 I=1,N
      IF(ICON1(I).EQ.IX)ICON1(I)=IZ
      IF(ICON2(I).EQ.IX)ICON2(I)=IZ
207   CONTINUE
      GO TO 111
208   IF(JMAX.GT.0)GO TO 212
      DO 210 I=1,N
      IX=0
      IY=-I
      IZ=-(JMAX+90001)
      DO 209 J=1,N
      IF(ICON1(J).NE.IY)GO TO 211
      IX=1
      ICON1(J)=IZ
      GO TO 209
211   IF(ICON2(J).NE.IY)GO TO 209
      IX=1
      ICON2(J)=IZ
209   CONTINUE
      IF(IX.EQ.0)GO TO 213
      JMAX=JMAX+1
210   CONTINUE
213   IF(GM.EQ.ATST(8))RETURN
      GO TO 111
10    PRINT 19                                                          DA  84
      PRINT 20, GM,ITG,NS,XW1,YW1,ZW1,XW2,YW2,ZW2,RAD                   DA  85
      STOP                                                              DA  86
C                                                                       DA  87
11     FORMAT (////,33X,35H- - - STRUCTURE SPECIFICATION - - -,//,37X,  DA  88
     1  28HCOORDINATES MUST BE INPUT IN,/,37X,                          DA  89
     2  29HMETERS OR BE SCALED TO METERS,/,37X,                         DA  90
     3  31HBEFORE STRUCTURE INPUT IS ENDED,//)                          DA  91
12     FORMAT (   2X,4HWIRE,79X,6HNO. OF,4X,5HFIRST,2X,4HLAST,5X,3HTAG, DA  92
     1 /,2X,3HNO.,8X,2HX1,9X,2HY1,9X,2HZ1,10X,2HX2,9X,2HY2,9X,2HZ2,6X,  DA  93
     2 6HRADIUS,3X,4HSEG.,5X,4HSEG.,3X,4HSEG.,5X,3HNO.)                 DA  94
13     FORMAT (A2,I3,I5,7F10.5)                                         DA  95
14     FORMAT (1X,I5,3F11.5,1X,4F11.5,2X,I5,4X,I5,1X,I5,3X,I5)          DA  96
15     FORMAT (6X,34HSTRUCTURE REFLECTED ALONG THE AXES,3(1X,A1),       DA  97
     1  22H.  TAGS INCREMENTED BY,I5)                                   DA  98
16     FORMAT (6X,30HSTRUCTURE ROTATED ABOUT Z-AXIS,I3,                 DA  99
     1 30H TIMES.  LABLES INCREMENTED BY,I5)                            DA 100
17     FORMAT (6X,26HSTRUCTURE SCALED BY FACTOR,F10.5)                  DA 101
18     FORMAT (6X,49HTHE STRUCTURE HAS BEEN MOVED. MOVE DATA CARD IS -/ DA 102
     1 6X,I3,I5,7F10.5)                                                 DA 103
19     FORMAT (25H GEOMETRY DATA CARD ERROR)                            DA 104
20     FORMAT (1X,A2,I3,I5,7F10.5)                                      DA 105
90     FORMAT(////,46H JUNCTIONS USING OPTIONAL INTERPOLATION METHOD)
91     FORMAT(A2,I3,2I5)
93     FORMAT(1X,A2,I3,2I5)
92     FORMAT(69H ERROR-- INVALID DATA CARD WHERE JUNCTION INTERPOLATION
     1CARD EXPECTED)
       END                                                              DA 106-
```

```
      SUBROUTINE FACTR (N.A.IP.NDIM)
C
C     SUBROUTINE TO FACTOR A MATRIX INTO A UNIT LOWER TRIANGULAR MATRIX
C     AND AN UPPER TRIANGULAR MATRIX USING THE GAUSS-DOOLITTLE ALGORITHM
C     PRESENTED ON PAGES 411-415 OF A. RALSTON--A FIRST COURSE IN
C     NUMERICAL ANALYSIS.  COMMENTS BELOW REFER TO COMMENTS IN RALSTONS
C     TEXT.     (MATRIX TRANSPOSE).
C
      DIMENSION A(NDIM.NDIM). IP(NDIM)
      COMMON /SCRATM/  D(500)
      COMPLEX  A.D.ARJ
      INTEGER  R.RM1.RP1.PJ.PR
      IFLG=0
      DO 9 R=1.N
C
C     STEP 1
C
      DO 1 K=1.N
      D(K)=A(R.K)
1     CONTINUE
C
C     STEPS 2 AND 3
C
      RM1=R-1
      IF (RM1.LT.1) GO TO 4
      DO 3 J=1.RM1
      PJ=IP(J)
      ARJ=D(PJ)
      A(R.J)=ARJ
      D(PJ)=D(J)
      JP1=J+1
      DO 2 I=JP1.N
      D(I)=D(I)-A(J.I)*ARJ
2     CONTINUE
3     CONTINUE
4     CONTINUE
C
C     STEP 4
C
      DMAX=REAL(D(R)*CONJG(D(R)))
      IP(R)=R
      RP1=R+1
      IF (RP1.GT.N) GO TO 6
      DO 5 I=RP1.N
      ELMAG=REAL(D(I)*CONJG(D(I)))
      IF (ELMAG.LT.DMAX) GO TO 5
      DMAX=ELMAG
      IP(R)=I
5     CONTINUE
6     CONTINUE
      IF (DMAX.LT.1.E-10) IFLG=1
      PR=IP(R)
      A(R.R)=D(PR)
      D(PR)=D(R)
C
C     STEP 5
C
      IF (RP1.GT.N) GO TO 8
      ARJ=1./A(R.R)
      DO 7 I=RP1.N
      A(R.I)=D(I)*ARJ
7     CONTINUE
8     CONTINUE
      IF (IFLG.EQ.0) GO TO 9
      PRINT 10.  R.DMAX
      IFLG=0
9     CONTINUE
      RETURN
C
10    FORMAT (1H .6HPIVOT(.I3.2H)=.E14.8)
      END
```

```
      SUBROUTINE INTG(B,S,RH,ZP,DIJ,DIR,ETR,ETI,DIL,DIK,IJ,IP,JCO1,JCO2,
     1 XM,YM,ZM)
C
C     INTG COMPUTES THREE COMPLEX FIELD COMPONENTS THAT MULTIPLY THE
C     THREE SEGMENT CURRENTS USED IN INTERPOLATING OVER A SEGMENT.
C     THESE COMPONENTS, ETR AND ETI, GO INTO THE INTERACTION MATRIX.
C
      DIMENSION ETR(3), ETI(3)
      DATA  TP/6.283185308/
C
C     COMPUTE TANGENTIAL FIELD ON OBSERVATION SEGMENT DUE TO SINE,
C     COSINE, AND CONSTANT CURRENTS ON SOURCE SEGMENT.
C
      CALL EFLD (B,S,RH,ZP,IJ,EZRS,EZIS,ERRS,ERIS,EZRC,EZIC,ERRC,ERIC,
     1 EZRK,EZIK,ERRK,ERIK)
      IF (IP.NE.2) GO TO 1
      CALL GN (EZRS,EZIS,ERRS,ERIS)
      CALL GN (EZRC,EZIC,ERRC,ERIC)
      CALL GN (EZRK,EZIK,ERRK,ERIK)
1     ETRS=EZRS*DIJ+ERRS*DIR
      ETIS=EZIS*DIJ+ERIS*DIR
      ETRC=EZRC*DIJ+ERRC*DIR
      ETIC=EZIC*DIJ+ERIC*DIR
      ETRK=EZRK*DIJ+ERRK*DIR
      ETIK=EZIK*DIJ+ERIK*DIR
C
C     COMPUTE INTERPOLATION COEFFICIENTS AND FORM THE COEFFICIENTS OF
C     THE THREE SEGMENT CURRENTS USED IN CURRENT INTERPOLATION.
C
      CL=TP*DIL
      CK=TP*DIK
      SINL=SIN(CL)
      COSL=COS(CL)
      SINK=SIN(CK)
      COSK=COS(CK)
      SILK=SIN(CL+CK)
      CONS=SINL+SINK-SILK
      ETR(1)=(SINK*ETRK+(COSK-1.)*ETRS-SINK*ETRC)/CONS
      ETI(1)=(SINK*ETIK+(COSK-1.)*ETIS-SINK*ETIC)/CONS
      ETR(2)=(-SILK*ETRK+(COSL-COSK)*ETRS+(SINL+SINK)*ETRC)/CONS
      ETI(2)=(-SILK*ETIK+(COSL-COSK)*ETIS+(SINL+SINK)*ETIC)/CONS
      ETR(3)=(SINL*ETRK+(1.-COSL)*ETRS-SINL*ETRC)/CONS
      ETI(3)=(SINL*ETIK+(1.-COSL)*ETIS-SINL*ETIC)/CONS
      IF(JCO1.LT.(-90000))GO TO 2
      IF(JCO2.LT.(-90000))GO TO 3
      RETURN
3     COSL=COSK
2     SILK=COS(CL+CK)
      XM=(SILK-COSL)/CONS
      YM=(1.-SILK)/CONS
      ZM=(COSL-1.)/CONS
      RETURN
      END
```

```
      SUBROUTINE JMELS(CMEL,NCP,JP,NCM,JM,I,CM,NROW,NCOL,NP,JPMAX)
C
C     JMELS SUMS THE CONTRIBUTIONS TO THE MATRIX ELEMENTS FOR SEGMENTS
C     CONNECTED TO JUNCTIONS OF THREE OR MORE SEGMENTS
C
      DIMENSION CM(NROW,NCOL)
      DIMENSION JP(25), JM(25)
      COMPLEX  CM,CMEL
      IF (NCP.LT.1) GO TO 2
      DO 1 J=1,NCP
      JPJ=JP(J)+(JP(J)-1)/NP*JPMAX
1     CM(JPJ,I)=CM(JPJ,I)+CMEL
2     CONTINUE
      IF (NCM.LT.1) GO TO 4
      DO 3 J=1,NCM
      JMJ=JM(J)+(JM(J)-1)/NP*JPMAX
3     CM(JMJ,I)=CM(JMJ,I)-CMEL
4     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LFACTR (A,NROW,NCOL,IX1,IX2,IP)
C
C     LFACTR PERFORMS GAUSS-DOOLITTLE MANIPULATIONS ON THE TWO BLOCKS OF
C     THE TRANSPOSED MATRIX IN CORE STORAGE.  THE GAUSS-DOOLITTLE
C     ALGORITHM IS PRESENTED ON PAGES 411-416 OF A. RALSTON -- A FIRST
C     COURSE IN NUMERICAL ANALYSIS.  COMMENTS BELOW REFER TO COMMENTS IN
C     RALSTONS TEXT.
C
      COMMON /MATPAR/  ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM
      COMMON /SCRATM/  D(800)
      DIMENSION A(NROW,NCOL), IP(NROW)
      COMPLEX   A,D,AJR
      INTEGER   R,R1,R2,PJ,PR
      LOGICAL   L1,L2,L3
      IFLG=0
C
C     INITIALIZE R1,R2,J1,J2
C
      L1=IX1.EQ.1.AND.IX2.EQ.2
      L2=(IX2-1).EQ.IX1
      L3=IX2.EQ.NBLOKS
      IF (L1) GO TO 1
      GO TO 2
1     R1=1
      R2=2*NPBLK
      J1=1
      J2=-1
      GO TO 5
2     R1=NPBLK+1
      R2=2*NPBLK
      J1=(IX1-1)*NPBLK+1
      IF (L2) GO TO 3
      GO TO 4
3     J2=J1+NPBLK-2
      GO TO 5
4     J2=J1+NPBLK-1
5     IF (L3) R2=NPBLK+NLAST
      DO 16 R=R1,R2
C
C     STEP 1
C
      DO 6 K = J1,NROW
      D(K)=A(K,R)
6     CONTINUE
C
C     STEPS 2 AND 3
C
      IF (L1.OR.L2) J2=J2+1
      IF (J1.GT.J2) GO TO 9
      IXJ=0
      DO 8 J=J1,J2
      IXJ=IXJ+1
      PJ=IP(J)
      AJR=D(PJ)
      A(J,R)=AJR
      D(PJ)=D(J)
      JP1=J+1
      DO 7 I = JP1,NROW
      D(I)=D(I)-A(I,IXJ)*AJR
7     CONTINUE
8     CONTINUE
9     CONTINUE
C
C     STEP 4
C
      J2P1=J2+1
      IF (L1.OR.L2) GO TO 11
      IF (NROW.LT.J2P1) GO TO 16
      DO 10 I = J2P1,NROW
      A(I,R)=D(I)
10    CONTINUE
      GO TO 16
11    DMAX=REAL(D(J2P1)*CONJG(D(J2P1)))
      IP(J2P1)=J2P1
      J2P2=J2+2
      IF (J2P2.GT.NROW) GO TO 13
      DO 12 I = J2P2, NROW
      ELMAG=REAL(D(I)*CONJG(D(I)))
      IF (ELMAG.LT.DMAX) GO TO 12
      DMAX=ELMAG
      IP(J2P1)=I
12    CONTINUE
13    CONTINUE
      IF (DMAX.LT.1.E-10) IFLG=1
      PR=IP(J2P1)
      A(J2P1,R)=D(PR)
      D(PR)=D(J2P1)
C
C     STEP 5
C
      IF (J2P2.GT.NROW) GO TO 15
      AJR=1./A(J2P1,R)
      DO 14 I=J2P2,NROW
      A(I,R)=D(I)*AJR
14    CONTINUE
```

```
15     CONTINUE
       IF (IFLG.EQ.0) GO TO 15
       PRINT 17,   J2.JMAX
       IFLG=0
16     CONTINUE
       RETURN
C
C
17     FORMAT  (1H .6HPIVOT(.I3.2H)=.F16.8)
       END
```



```
       SUBROUTINE SOLVES (NOP.A.IP.B.NROW.NCOL.IX.NP.N.JPMAX)
C
C      SUBROUTINE SOLVES. FOR SYMMETRIC STRUCTURES. HANDLES THE
C      TRANSFORMATION OF THE RIGHT HAND SIDE VECTOR AND SOLUTION OF THE
C      MATRIX EQ.
C
       COMMON /SMAT/ S(10.10)
       DIMENSION A(NROW.NCOL). IP(N). IX(N). B(N)
       COMMON /SCRATM/  Y(800)
       COMMON /MATPAR/ ICASE.NBLOKS.NPBLK.NLAST.NBLSYM.NPSYM.NLSYM
       COMPLEX  A.B.Y.SUM.S
       NPEQ=NP
       IF(JPMAX.EQ.0)GO TO 15
       NPEQ=NP+JPMAX
       NEQ=N+JPMAX*NOP
       IF(NOP.EQ.1)GO TO 25
       DO 16 I=NP.N
16     Y(I)=B(I)
       IB=NP
25     IA=NP
       DO 17 I=1.NOP
       IF(I.EQ.1)GO TO 26
       DO 18 J=1.NP
       IA=IA+1
       IB=IB+1
18     B(IA)=Y(IB)
26     DO 19 J=1.JPMAX
       IA=IA+1
19     B(IA)=(0..0.)
17     CONTINUE
15     IF(NOP.EQ.1)GO TO 5
       FNOP=NOP
       FNORM=1./FNOP
C
C      TRANSFORM MATRIX EQ. RHS VECTOR ACCORDING TO SYMMETRY MODES
C
       DO 4 I=1.NPEQ
       DO 1 K=1.NOP
       IA=I+(K-1)*NPEQ
1      Y(K)=B(IA)
       SUM=Y(1)
       DO 2 K=2.NOP
2      SUM=SUM+Y(K)
       B(I)=SUM*FNORM
       DO 4 K=2.NOP
       IA=I+(K-1)*NPEQ
       SUM=Y(1)
       DO 3 J=2.NOP
3      SUM=SUM+Y(J)*CONJG(S(K.J))
4      B(IA)=SUM*FNORM
5      IF (ICASE.LT.3) GO TO 6
       REWIND  15
       REWIND  16
C
C      SOLVE EACH MODE EQUATION
C
6      DO 10 KK=1.NOP
       IA=(KK-1)*NPEQ+1
       IB=IA
       IF (ICASE.NE.4) GO TO 9
       DO 7 I = 1. NPEQ
7      READ (15) (A(J.I).J=1.NPEQ)
```

```
      IB=1
8     IF (ICASE.EQ.3.OR.ICASE.EQ.5) GO TO 9
      CALL SOLVE (NPEQ,A(IB,1),IP(IA),B(IA),NROW)
      GO TO 10
9     CALL LTSOLV (A,NPEQ,NCOL,IX(IA),B(IA) )
10    CONTINUE
      IF (NOP.EQ.1) GO TO 20
C
C     INVERSE TRANSFORM THE MODE SOLUTIONS
C
      DO 14 I=1,NPEQ
      DO 11 K=1,NOP
      IA=I+(K-1)*NPEQ
11    Y(K)=B(IA)
      SUM=Y(1)
      DO 12 K=2,NOP
12    SUM=SUM+Y(K)
      B(I)=SUM
      DO 14 K=2,NOP
      IA=I+(K-1)*NPEQ
      SUM=Y(1)
      DO 13 J=2,NOP
13    SUM=SUM+Y(J)*S(K,J)
14    B(IA)=SUM
20    IF(JPMAX.EQ.0.OR.N.EQ.NP)RETURN
      DO 21 I=NP,NEQ
21    Y(I)=B(I)
      IA=NP
      IB=N
      K=NP
      DO 22 I=1,NOP
      IF(I.EQ.1)GO TO 27
      DO 23 J=1,NP
      IA=IA+1
      K=K+1
23    B(IA)=Y(K)
27    DO 24 J=1,JPMAX
      IB=IB+1
      K=K+1
24    B(IB)=Y(K)
22    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRIO (J,JCO1,JCO2,DIL,DIK)
C
C     SUBROUTINE TRIO DETERMINES WHICH SEGMENTS ARE CONNECTED TO SEGMENT
C     J.  SUBROUTINE JUNC IS CALLED TO FILL COMMON/JUNK/ FOR MULTIPLE
C     JUNCTIONS.
C
      COMMON /DATA/  N,NP,X(800),Y(800),Z(800),SI(800),BI(800),ALP(800),
     1 BET(800),ICON1(800),ICON2(800),ITAG(800),WLAM,IPSYM,JMAX
      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.LT.(-90000))GO TO 2
      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.LT.(-90000))GO TO 6
      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
```

REFERENCES

1. MBAssociates, "Antenna Modeling Program - Composite User's Manual", MBAssociates Report No. MB-R-74/46, 1974.

2. MBAssociates, "Antenna Modeling Program - Engineering Manual", MBAssociates Report No. MB-R-74/62, 1974.

3. MBAssociates, "Antenna Modeling Program - Systems Manual", MBAssociates Report No. MB-R-74/62, 1974.

4. MBAssociates, Antenna Modeling Program Supplementary Computer Manual (AMP2), MBAssociates Report No. MB-R-75/4, Jan. 1975.

5. Schelkunoff, S. A., and H. T. Friis, Antennas Theory and Practice, John Wiley & Sons, Inc., New York, 1952.

6. Harrington, R. F., Time-Harmonic Electromagnetic Fields, McGraw-Hill (1961), p. 79.