N74-34603

COMPUTER PROGRAM FOR THIN-WIRE ANTENNA OVER
A PERFECTLY CONDUCTING GROUND PLANE

J. H. Richmond

CR-140,622

(NASA-CR-140622) COMPUTER PROGRAM FOR THIN-WIRE ANTENNA OVER A PERFECTLY CONDUCTING GROUND PLANE (Ohio State Univ.) N74-34603

The Ohio State University

ElectroScience Laboratory

Department of Electrical Engineering
Columbus, Ohio 43212

TECHNICAL REPORT 2902-19

October 1974

Grant No. NGL 36-008-138

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23365

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

COMPUTER PROGRAM FOR THIN-WIRE ANTENNA OVER A PERFECTLY CONDUCTING GROUND PLANE

J. H. Richmond

TECHNICAL REPORT 2902-19

October 1974

Grant No. NGL 36-008-138

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23365

ABSTRACT

A computer program is presented for a thin-wire antenna over a perfect ground plane. The analysis is performed in the frequency domain, and the exterior medium is free space. The antenna may have finite conductivity and lumped loads. The output data includes the current distribution, impedance, radiation efficiency and gain. The program uses sinusoidal bases and Galerkin's method.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Reference 1 presents a computer program and reference 2 presents the theory for thin-wire antennas and scatterers in a homogeneous conducting medium. The present program differs from reference 1 only in the following details:

a. The exterior medium is free space.
b. The antenna is situated over a perfect ground plane.
c. The wires have no dielectric sleeves.
d. The frequency is real.
e. Scattering problems are not considered.

To avoid unnecessary duplication, it is assumed the reader is familiar with the program in reference 1.

The program handles antennas constructed of straight wire segments. One or more segments may connect to the ground plane, or the antenna may be situated some distance away from the ground plane. No segment has both endpoints on the ground plane. This program can readily be modified to handle more general situations as in reference 1. The program uses the delta-gap model for the generators.

The method of images is employed to reduce the problem to a thin-wire structure in free space. The user sets up the geometry of the real wire configuration, and the program automatically sets up the image. If unlimited storage were available, one might set up a large mutual-impedance matrix for the wire antenna and its image in free space. Instead, this program takes advantage of the ground-plane symmetry and sets up the compressed matrix C(I,J). Only the currents on the real segments are treated as independent unknown quantities, and the image currents are dependent. All the currents, however, are influenced by the mutual couplings among all the segments including the images. In taking advantage of the mirror symmetry, we lose the advantage of having a symmetric matrix. However, the net gain in computational speed and storage is substantial.

In practice, many wire antennas operate over a ground plane with finite conductivity and finite extent. In many cases, however, one may substitute a perfectly conducting ground plane of infinite extent without unduly disturbing the antenna current distribution or impedance. After these quantities have been calculated, one may then take into account the finite ground plane in calculating the efficiency and patterns. The present program, however, assumes an idealized ground plane throughout.

The remaining sections present the computer program with enough explanation to enable an experienced engineer to use it.

II. THE INPUT DATA AND SUBROUTINE IWIRE

Fig. 1 is a Fortran listing of subroutine IWIRE. This subroutine is used to set up the input data. The following data must be read or programmed in IWIRE:

AL	wire radius a/λ
CMM	wire conductivity in megamhos/m
DPH	increment in far-field angle ϕ in degrees
FMC	frequency in MHz
TH	elevation angle θ in degrees for far-field pattern

To define the shape and size of the wire antenna, the input data includes the coordinates XC(I), YC(I), and ZC(I) of the wire endpoints, terminals and other current sampling stations along the wire axis. The unit of length is selected by the user, and SCALE is the conversion factor such that XC(I)*SCALE gives the coordinate of point I in meters. NPGP denotes the number of points on the ground plane, and NRP is the number of real points including those on the ground plane. Coordinates are supplied only for the real points. The ground plane coincides with the xy plane.

NRS denotes the number of real segments, and NSGP is the number of real segments having an endpoint on the ground plane. For each real segment J, the input data specifies the endpoints IA(J) and IB(J). In assigning index numbers to the segments, the lowest numbers must refer to those having an endpoint on the ground plane. In assigning index numbers to the points, the lowest numbers must refer to those on the ground plane.

Set IWRCJ = 1 to obtain a writeout of the antenna currents; otherwise IWRCJ = -1. Set IWRITE = 1 to obtain a writeout of the antenna geometry; otherwise IWRITE = -1. If INT = 0, the rigorous closed-form expressions will be used for the mutual impedance of sinusoidal dipoles and the calculations will tend to be slow but accurate. If INT is a positive integer, Simpson's rule will be used for the mutual-impedance calculations. The closed-form expressions are always used automatically for the most critical impedances. We usually use INT = 0 for multi-turn loop antennas with closely-spaced turns, and INT = 4 for general purpose. Simpson's rule uses INT integration intervals. Thus the accuracy and the execution time tend to increase with larger values of INT.

Set NLD = 0 if there are no lumped loads; otherwise NLD = 1. ZLD(J) denotes the impedance (in ohms) inserted in segment J at endpoint IA(J). A lumped load at endpoint IB(J) is denoted by ZLD(J + NRS). The user sets up only the real generators and lumped loads, and the program takes care of the images.

```
      SUBROUTINE IWIRE(IA,IB,INP,INS,INT,IWRCJ,IWRITE,NLD,NP,NS,NRP,
     2NRS,NPGP,NSGP,AL,CMM,DPH,FMC,SCALE,TH,VG,XC,YC,ZC,ZLD)
      DIMENSION IA(1),IB(1),XC(1),YC(1),ZC(1)
      COMPLEX VG(1),ZLD(1)
    4 FORMAT(4X,'INP=',I4,5X,'NP=',I4,5X,'INS=',I4,5X,'NS=',I4)
    5 FORMAT(1H0)
C  IA(J) AND IB(J) ARE ENDPOINTS OF SEGMENT J
C  XC(I),YC(I),ZC(I) ARE COORDINATES OF POINT I WITH ARBITRARY UNITS
C  NRP = NUMBER OF REAL POINTS, INCLUDING THOSE ON THE GROUND PLANE
C  NRS = NUMBER OF REAL SEGMENTS
C  NPGP = NUMBER OF POINTS ON THE GROUND PLANE
C  NSGP = NUMBER OF REAL SEGMENTS WITH ENDPOINT ON GROUND PLANE
C  NLD = NUMBER OF LUMPED LOADS
      DO 10 J=1,INS
      VG(J)=(.0,.0)
   10 ZLD(J)=(.0,.0)
C  SET UP THE REAL GENERATORS AND REAL LUMPED LOADS
      JGN=4
      VG(JGN)=(1.,0.)
      NLD=0
      INT=4
      IWRITE=1
      IWRCJ=1
      AL=.0001
      CMM=1.
      DPH=20.
      FMC=75.
      SCALE=1.
      TH=85.
      NSGP=2
      NPGP-2
      NRS=4
      NRP=5
      NS=2*NRS
      NP=2*NRP-NPGP
      WRITE(6,4)INP,NP,INS,NS
      IF(NS.GT.INS .OR. NP.GT.INP)GO TO 600
      DO 20 I=1,NRP
      XC(I)=.0
      YC(I)=.0
   20 ZC(I)=.0
C  NEXT SET UP THE REAL POINTS
      XC(1)=1.
      XC(3)=1.
      ZC(3)=.5
      ZC(4)=.5
      XC(5)=1.707
      ZC(5)=1.207
C  NEXT SET UP THE REAL SEGMENTS
      IA(1)=1
      IB(1)=3
      IA(2)=2
      IB(2)=4
      IA(3)=3
      IB(3)=4
      IA(4)=3
      IB(4)=5
  600 RETURN
      END
```

Fig. 1. Subroutine IWIRE.

The generator location is defined by JGN. The numbering system for the generators is the same as for lumped loads. If the generator is to be inserted in segment J at endpoint IA(J), the generator index JGN is the same as the segment index J. To insert a generator at endpoint IB(J), set JGN = J + NRS. VG(JGN) denotes the complex voltage of the generator. The reference direction for these voltages is from IA(J) toward IB(J). If the antenna is fed with several generators, delete JGN and merely input the generator voltages VG.

NP and NS denote the number of points and segments, respectively, for the complete system (antenna and image) in free space.

III. THE MAIN COMPUTER PROGRAM

The main computer program is listed in Fig. 2. This program calls subroutine IWIRE for the input data. Then it calls subroutine ISORT to generate and store the data for the image points and image segments and the length DC(J) of each segment. Then ISORT generates a list of sinusoidal dipole modes for the complete system (antenna and image) in free space. Dipole mode I has segments JA(I) and JB(I), terminals at point I2(I), and endpoints I1(I) and I3(I). This subroutine also generates the following information:

ND(J)	number of dipole modes sharing segment J
MD(J,K)	list of dipoles sharing segment J
NCM	size of the compressed matrix
N	number of dipole modes on the complete system

The following quantities must be specified in the main program:

ICC	dimension for the compressed matrix C(I,J)
ICJ	dimension related to number of dipole modes N
INP	dimension related to number of points NP
INS	dimension related to number of segments NS

In Fig. 2, all quantities having the same dimensions are dimensioned in the same or adjacent statements. The numerical values assigned to ICC, ICJ, INP and INS must agree with the dimensions actually reserved for the corresponding quantities in the COMPLEX and DIMENSION statements. ICC, ICJ, INP and INS must be at least as large as NCM, N, NP and NS, respectively. In Fig. 2, the main program is dimensioned for up to 150 modes, 90 points, 100 segments, and a compressed matrix as large as 30 by 30. If the wire antenna makes no contact with the ground plane, the compressed matrix will be exactly half as large as the full matrix. Otherwise NCM is somewhat larger than N/2.

```
C      INCLUDE ANTI:CROUT:EXPJ:IDANT:IFFLD:ISORT:IWIRE:ZFF:ZGMM:ZGS
C      INCLUDE POISS:ZSURF
C  THIN-WIRE ANTENNA OVER PERFECT GROUND PLANE
C  SINUSOIDAL-GALERKIN FREQUENCY-DOMAIN
C  PROGRAM ORIGINATED BY J. H. RICHMOND, OHIO STATE UNIVERSITY
      COMPLEX EPH,ETH,Y11,Z11,ZM
      COMPLEX C(30,30)
      COMPLEX CJ(150),EP(150),ET(150),EPP(150),ETT(150),VJ(150)
      COMPLEX VS(150),ZLD(150)
      DIMENSION XC(90),YC(90),ZC(90),X(90),Y(90),Z(90)
      DIMENSION D(100),DC(100)
      DIMENSION IA(100),IB(100),MD(100,4),ND(100),CDK(100),SDK(100)
      DIMENSION I1(150),I2(150),I3(150),JA(150),JB(150)
      DATA PI,TP/3.14159,6.28318/
  1   FORMAT(8X,'JPP=',I5,5X,'MAX=',I5,5X,'MIN=',I5,5X,'N=',I5,5X,
     2'NCM=',I5)
  2   FORMAT(8X,'AL=',F8.6,5X,'CMM=',F8.4,5X,'FMC=',F8.2)
  3   FORMAT(8X,'EFF=',F7.2,3X,'Y11=',2F10.6,3X,'Z11=',2F8.2)
  4   FORMAT(8X,'PH',8X,'TH',8X,'DBP',7X,'DBT',7X,'GPP',7X,'GTT')
  5   FORMAT(1H0)
  6   FORMAT(1X,2F10.0,4F10.2)
      ICC=30
      ICJ=150
      INP=90
      INS=100
C  THE GEOMETRY OF THE THIN-WIRE STRUCTURE IS SPECIFIED IN SUB. IWIRE
      CALL       IWIRE(IA,IB,INP,INS,INT,IWRCJ,IWRITE,NLD,NP,NS,NRP,
     2NRS,NPGP,NSGP,AL,CMM,DPH,FMC,SCALE,TH,VG,XC,YC,ZC,ZLD)
      IF(NSGP.LT.NPGP)GO TO 500
      IF(NS.GT.INS .OR. NP.GT.INP)GO TO 500
      CALL       ISORT(IA,IB,ICC,ICJ,INS,IWRITE,I1,I2,I3,JA,JB,
     2MAX,MIN,MD,N,NCM,ND,NP,NS,NRP,NRS,NPGP,NSGP,DC,XC,YC,ZC)
C  NCM =  SIZE OF COMPRESSED MATRIX C(I,J)
      JPP=NCM-NPGP
      WRITE(6,1)JPP,MAX,MIN,N,NCM
      WRITE(6,5)
      AK=TP*AL
      WAVM=300./FMC
      WRITE(6,2)AL,CMM,FMC
      WRITE(6,5)
      TPL=SCALE*TP/WAVM
      IF(N.LE.0 .OR. N.GT.ICJ)GO TO 500
      IF(NCM.GT.ICC)GO TO 500
      IF(MAX.LE.0 .OR. MIN.LE.0)GO TO 500
      DO 90 J=1,NS
 90   D(J)=TPL*DC(J)
      DO 100 I=1,NP
      X(I)=TPL*XC(I)
      Y(I)=TPL*YC(I)
100   Z(I)=TPL*ZC(I)
      CALL       IDANT(IA,IB,ICC,INS,INT,I1,I2,I3,JA,JB,JPP,MD,N,NCM,
     2ND,NLD,NP,NPGP,NRS,NS,AK,C,CMM,D,FMC,CDK,SDK,X,Y,Z,ZM,ZLD)
      IF(N.EQ.0)GO TO 500
      I12=1
      CALL       ANTI(IA,IB,I1,I2,I3,IWRCJ,IWRITE,I12,ICC,INS,JA,JB,
     2JPP,MD,N,NCM,ND,NLD,NPGP,NRS,NS,C,CDK,SDK,CJ,CMM,D,EFF,G,VG,VJ,
     3Y11,Z11,ZM,ZLD)
      IF(I12.NE.12)GO TO 500
      WRITE(6,3)EFF,Y11,Z11
      IF(G.EQ.1. .AND. EFF.EQ.0.)GO TO 500
      WRITE(6,5)
      LIM=1.5+360./DPH
```

Fig. 2. The MAIN computer program.

```
      WRITE(6,4)
      DO 300 IPH=1,LIM
      PH=DPH*(IPH-1)
      CALL          IFFLO(IA,IB,INS,I1,I2,I3,MD,N,ND,NS,COK,CJ,D,
     2 EPP,ETT,FPH,FTH,C,GPP,GTT,PH,SOK,TH,X,Y,Z)
      DBP=.0
      DBT=.0
      IF(GPP.GT.0.)DBP=10.*ALOG10(GPP)
      IF(GTT.GT.0.)DBT=10.*ALOG10(GTT)
  300 WRITE(6,6)PH,TH,DBP,DBT,GPP,GTT
  500 CALL EXIT
      END
```

Fig. 2. The MAIN computer program - continued

X(I), Y(I) and Z(I) denote kx, ky and kz for point I, where $k = 2\pi/\lambda$. If calculations are desired for a given antenna at several frequencies, the frequency DO LOOP will begin just below the call to ISORT.

The main program calls subroutine IDANT to generate the compressed open-circuit impedance matrix C(I,J). Then subroutine ANTI is called to obtain the current distribution CJ(I) and the radiation efficiency EFF. ANTI also calculates the complex power input to the antenna, denoted by Y11, and the time-average input power G. If the antenna has only one generator and VG(JCN) = (1.,0.), then Y11 and Z11 denote the antenna admittance and impedance, respectively.

Finally, the antenna pattern is obtained by calling subroutine IFFLD. TH and PH denote the spherical coordinates θ and ϕ (in degrees) of the distant observer. GPP and GTT denote the ϕ-polarized and θ-polarized power gains, respectively, and DBP and DBT are the decibel versions. The user may want to increment θ as well as ϕ, but this will require only a trivial change in the main program. IFFLD is called once for each look angle (θ, ϕ). When $\theta = 90°$, GPP will vanish if the program has set up a valid system of images.

IV. AN EXAMPLE

Fig. 3 shows a simple antenna and its image, with a dotted line to indicate the ground plane. In Fig. 1, subroutine IWIRE sets up the following input data for this antenna:

VG(4) = 1	unit voltage generator at endpoint IA of segment 4
NLD = 0	no lumped loads
CMM = 1.	the wire conductivity is 1 megamho/m
NSGP = 2	2 real segments connect to the ground plane
NPGP = 2	2 points on the ground plane
NRS = 4	4 real segments
NRP = 5	5 real points

This planar antenna has 8 points (NP = 8) and 8 segments (NS = 8), and the numbering system is shown in Fig. 3. Note that the lowest numbers are assigned to the two points on the ground plane and the two segments terminating on the ground plane. The points and segments must be labeled with consecutive positive integers 1, 2, 3, . . . For a given segment J, it makes no difference which end is labeled IA(J). In Fig. 3, each numeral located near a dot is the index I of that point. Each numeral located near the center of a line is the index J of that segment.

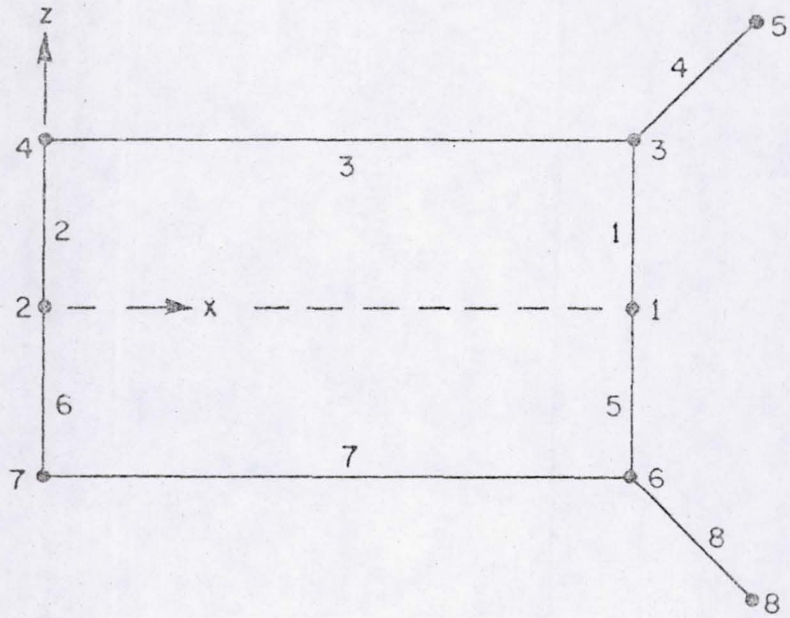

Fig. 3. Points and segments on a simple wire antenna lying in the xz plane.

Fig. 4 shows the same antenna and the eight dipole modes defined by subroutine ISORT. The arrows indicate the reference directions for the mode currents and voltages. The mode index number I is placed near the terminal point I2(I). Mode I is a sinusoidal basis function which vanishes at the endpoints I1(I) and I3(I) and has unit current at the terminal point. These are overlapping subsectional bases, and mode I extends over two intersecting segments JA(I) and JB(I). The reference direction for mode currents and voltages is from I1 to I2 toward I3. In Fig. 4, modes 1 and 2 have terminals at the ground plane, with segment JA above and segment JB below the ground plane. This type of mode has no image. Modes 3, 4 and 5 have images. The size of the compressed matrix is NCM = 5. If we did not take advantage of the ground-plane symmetry, the matrix size would be N = 8.

Table I presents some of the output data for this example and Table II lists the elements of the compressed matrix C(I,J) on return from subroutine IDANT. From Table I, the calculated impedance is Z11 = 959 + j 664 ohms. For the same antenna with perfect

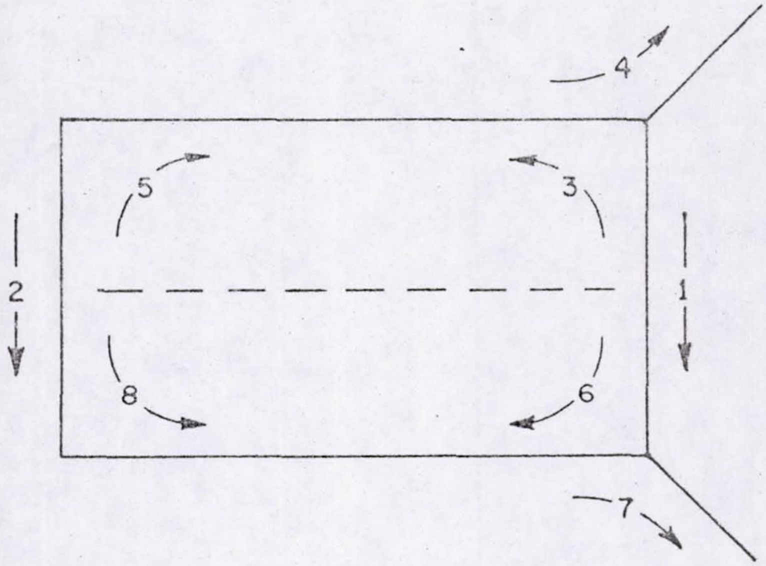

Fig. 4. Mode map for the antenna shown in Fig. 3.

conductivity, Z11 = 879 + j 749 ohms. The calculated results should not be considered accurate without checking the behavior as the wires are subdivided into more segments. The longest segment should not exceed $\lambda/4$. The thin-wire and delta-gap formulations are justified most readily if the wire radius does not exceed 0.007λ. Fortuitously, satisfactory results are often obtained for closed wire loops even when the wire radius is as large as 0.02λ. For dipoles, an upper limit of 0.007λ is recommended.

With 20-ohm resistive loads inserted in each end of each segment of the antenna in this example, the calculated impedance is 710 + j 206 ohms and the efficiency is 45.5 per cent with CMM = 1. With and without lumped loads, the results obtained with the present computer program show satisfactory agreement with those obtained with the program in reference 1. (A new version of subroutine SGANT was used for these tests because the original version in reference 1 does not always handle lumped loads properly.)

TABLE I

INP= 90 NP= 8 INS= 100 NS= 8

J	IA(J)	IB(J)	K	IA(K)	IB(K)	DC(J)
1	1	3	5	1	6	.50000
2	2	4	6	2	7	.50000
3	3	4	7	6	7	1.00000
4	3	5	8	6	8	.99985

I	XC(I)	YC(I)	ZC(I)	J	XC(J)	YC(J)	ZC(J)
1	1.00000	0.00000	0.00000				
2	0.00000	0.00000	0.00000				
3	1.00000	0.00000	.50000	6	1.00000	0.00000	-.50000
4	0.00000	0.00000	.50000	7	0.00000	0.00000	-.50000
5	1.70700	0.00000	1.20700	8	1.70700	0.00000	-1.20700

I	JA	JB	I1	I2	I3	K	JA	JB	I1	I2	I3
1	1	5	3	1	6						
2	2	6	4	2	7						
3	1	3	1	3	4	6	5	7	1	6	7
4	3	4	4	3	5	7	7	8	7	6	8
5	2	3	2	4	3	8	6	7	2	7	6

JPP= 3 MAX= 3 MIN= 1 N= 8 NCM= 5

AL= .000100 CMM= 1.0000 FMC= 75.00

J= 4 VG(J)= 1.00 0.00

I	MAGNITUDE	PHASE	REAL	IMAGINARY
1	1.000	99.9	-.0005441	.0031180
2	.870	91.1	-.0000507	.0027537
3	.730	-77.8	.0004892	-.0022579
4	.271	-34.7	.0007048	-.0004880
5	.638	-86.1	.0001362	-.0020153

EFF= 90.44 Y11= .000705 -.000488 Z11= 959.07 664.07

PH	TH	DBP	DBT	GPP	GTT
0.	85.	0.00	.65	0.00	1.16
20.	85.	-47.44	1.24	.00	1.33
40.	85.	-39.49	2.63	.00	1.83
60.	85.	-33.35	4.17	.00	2.61
80.	85.	-29.46	5.31	.00	3.40
100.	85.	-27.93	5.77	.00	3.77

TABLE II

Compressed Impedance Matrix

I	J	C(I,J)	C(J,I)
1	1	16.1 -j 720.5	16.1 -j 720.5
1	2	7.6 -j 6.7	7.6 -j 6.7
1	3	-15.9 -j 1059.9	- 7.9 -j 529.9
1	4	-18.8 -j 16.8	- 9.4 -j 8.4
1	5	- 5.2 +j 83.0	- 2.6 +j 41.5
2	2	16.1 -j 720.5	16.1 -j 720.5
2	3	- 5.2 +j 83.0	- 2.6 +j 41.5
2	4	-12.7 -j 77.3	- 6.4 -j 38.6
2	5	-15.9 -j 1059.9	- 7.9 -j 529.9
3	3	21.2 -j 326.0	21.2 -j 326.0
3	4	- 9.2 -j 22.6	- 9.2 -j 22.5
3	5	- 9.2 -j 396.0	- 9.2 -j 396.0
4	4	51.3 +j 60.8	51.3 +j 60.8
4	5	22.1 +j 420.9	22.1 +j 420.9
5	5	21.2 -j 326.0	21.2 -j 326.0

V. SUMMARY AND CONCLUSIONS

A computer program is presented for a thin-wire antenna over a perfectly conducting ground plane of infinite extent. The analysis is performed in the frequency domain, and the exterior medium is free space. The antenna may have finite conductivity and lumped loads. The output data includes the current distribution, impedance, radiation efficiency and gain. The program uses sinusoidal bases and Galerkin's method and takes advantage of the ground-plane symmetry to reduce the storage requirements and computation costs. The subroutines are included in alphabetical order in the Appendices with a brief explanation.

REFERENCES

1. J. H. Richmond, "Computer Program for Thin-Wire Structures in a Homogeneous Conducting Medium," NASA Contractor Report CR-2399, June 1974, for sale by the National Technical Information Service, Springfield, Virginia 22151, price $3.75.

2. J. H. Richmond, "Radiation and Scattering by Thin-Wire Structures in the Complex Frequency Domain," NASA Contractor Report CR-2396, May 1974, for sale by the National Technical Information Service, Springfield, Virginia 22151, price $3.25.

APPENDIX 1. Subroutine ANTI

Subroutine ANTI is listed in Fig. 5. Between statements 14 and 30, this subroutine sets up the excitation voltages CJ(I) and VJ(I) with the aid of the delta-gap model and the input data for the generator voltages VG(J). ANTI calls CROUT to obtain a solution for the simultaneous linear equations. On return from CROUT, the dipole mode currents are stored in CJ(I). The image currents are stored in CJ(K) in the DO LOOP ending with statement 80. The DO LOOP ending with statement 90 calculates the complex power input Y11 and the time-average power input G. The power dissipated (DISS) in the lumped loads and the imperfectly conducting wire is obtained by calling PDISS. Finally, the radiation efficiency EFF is calculated.

If IWRCJ is positive, ANTI writes a list of the dipole mode currents CJ(I). This list includes the normalized current magnitude, the phase in degrees, and the real and imaginary parts of the current.

APPENDIX 2. Subroutine CROUT

CROUT, listed in Fig. 6, solves a system of simultareous linear equations with complex coefficients. This subroutine uses the method of P. D. Crout. Although this subroutine does not use pivoting, it is efficient and accurate in the present application. The input data are defined as follows:

C(I,J)	complex coefficients in the simultaneous equations
S(I)	excitation column
ICC	dimensions of C and S
ISYM	zero or one for symmetric or nonsymmetric matrix
IWR	one or zero to write or suppress the solution
I12	one or two if C is original or auxiliary matrix
N	size of the square matrix C

Of course, N must not exceed ICC. If IWR is a positive integer, the solution will be printed out with the following definitions:

I	index number of the solution S(I)
SNOR	normalized magnitude of S(I)
SA	absolute magnitude of S(I)
PH	phase of S(I) in degrees

On the first call to CROUT, C(I,J) contains the original matrix. I12 = 1 and CROUT generates the auxiliary square matrix, overlaying it in the same location C and destroying the original matrix. Then CROUT proceeds to generate the solution, storing it in S(I) and destroying the original excitation column.

Next we might want another solution of the same system of simultaneous linear equations but with a new excitation column. This could be obtained by recalculating the original matrix C(I,J) and the new excitation column and calling CROUT again with I12 = 1. However, there is no need to recalculate C(I,J). Instead generate the new excitation column, set I12 = 2 (or any integer other than 1) and call CROUT again. CROUT uses less computer time when I12 differs from 1.

APPENDIX 3. Subroutine EXPJ

Subroutine EXPJ, listed in Fig. 7, evaluates the exponential integral defined as follows:

$$W12 = \int_{V1}^{V2} \frac{e^{-v}\,dv}{v} = E_1(V1) - E_1(V2) + j\,2n\pi$$

where the integration path is the straight line from V1 to V2 on the complex v plane and

$$E_1(z) = \int_{z}^{\infty} \frac{e^{-t}\,dt}{t} .$$

$E_1(z)$ denotes the principal branch of the exponential integral. To generate W12, EXPJ calculates $E_1(V1)$, subtracts $E_1(V2)$ and adds $j2n\pi$. The integer n is zero unless the straight-line integration path intersects the negative real v axis at a point between V1 and V2. When there is such an intersection, n = 1 if V1 lies in quadrant 1 or 2 and n = -1 if V1 lies in quadrant 3 or 4.

APPENDIX 4. Subroutine IDANT

Subroutine IDANT is listed in Fig. 8. This subroutine stores the quantities CDK(J) = cos kd_j and SDK(J) = sin kd_j where d_j is the length of segment j. The program writes AK, DMAX and DMIN and aborts if

a. the length of the shortest segment is less than the wire radius, or
b. the longest segment has a length d such that kd exceeds 3, or
c. the wire radius a is such that ka exceeds 0.1.

IDANT calculates the elements in the compressed impedance matrix C(I,J) as follows. Select a source segment K and a receiving segment L, where K and L range from 1 to NS. The mutual impedances P11, P12, P21 and P22 between the two segments are obtained by calling ZGMM if K = L, ZGMM if the segments intersect, or ZGS if segments K and L do not intersect.

Now select a test dipole I sharing segment K, and an expansion dipole J sharing segment L. Add the appropriate segment-to-segment impedance to the dipole-to-dipole impedance C(I,J). When this procedure has been completed at statement 200, the impedances C(I,J) are appropriate for a perfectly conducting thin-wire system with no lumped loads.

Between statements 200 and 262, the impedance matrix C(I,J) is modified to account for the finite conductivity of the wire antenna. The surface impedance ZS is obtained by calling ZSURF. For each segment K, the program selects a test dipole I and an expansion dipole J sharing this segment. The contribution to C(I,J) associated with finite conductivity is ZSAM if dipoles I and J have terminals at the same end of segment K, and ZOPP if they have terminals at opposite ends. C(I,J) is not affected unless dipoles I and J share one or two segments.

Between statements 262 and 280, the impedance matrix is modified to account for the lumped loads. Each diagonal element C(I,I) is modified by adding the impedance of the lumped load inserted at the terminals of mode I. If modes I and J share a segment and have terminals at the same point, C(I,J) is modified by adding or subtracting the impedance of the lumped load inserted at the terminal end of this segment. (Add or subtract ZLD if mode currents I and J have the same or opposite reference directions on the shared segment.)

APPENDIX 5. Subroutine IFFLD

Subroutine IFFLD, listed in Fig. 9, calculates the far-zone field of the thin-wire antenna.

Let (r,θ,ϕ) denote the spherical coordinates of the distant observer, and let $E_\theta(I)$ and $E_\phi(I)$ denote the electric field intensities of dipole mode I with unit terminal current. Then

$$EPP(I) = (r/\lambda)\, e^{jkr}\, E_\phi(I)$$

$$ETT(I) = (r/\lambda)\, e^{jkr}\, E_\theta(I)$$

The field of sinusoidal dipole mode I may be regarded as the sum of the fields of each of its two segments. The field of segment K is obtained by calling subroutine ZFF, and EPP(I) and ETT(I) are generated by adding the appropriate numbers obtained from two different calls to ZFF. In the DO LOOP ending with statement 260, the antenna field is calculated as a weighted sum of the mode fields as follows:

$$EPH = \sum_{1}^{N} CJ(I)\ EPP(I)$$

$$ETH = \sum_{1}^{N} CJ(I)\ ETT(I)$$

where CJ(I) denotes the terminal current of mode I and EPH and ETH denote the dimensionless range-independent form of the antenna fields E_ϕ and E_θ.

G denotes the time-average input power to the antenna, and GPP and GTT are the ϕ-polarized and θ-polarized power gains. Subroutine IFFLD is called once for each angular direction. In the input data supplied to this subroutine, PH and TH denote ϕ and θ in degrees.

This subroutine is useful for wire antennas with or without a ground plane. For an antenna over a ground plane, IFFLD must be supplied with information on the complete system including the image.

APPENDIX 6. Subroutine ISORT

Subroutine ISORT, listed in Fig. 10, is described briefly in Section III. This subroutine sets up the image segments and points and calculates the segment lengths. Then it checks the input data for consistence. The data are considered inconsistent and the run is aborted if

a. NPGP is greater than zero but one of the points on the ground plane has no segment (with index J less than or equal to NSGP) connected to it, or
b. a real point situated above the ground plane has no segment (with index J less than or equal to NRS) connected to it.

Between statements 32 and 50, this program calculates the number of modes N on the complete structure. The run is aborted if the dimensions are inadequate.

Between statements 50 and 58, the program sets up the modes that will not have images. The number of modes of this type is NPGP, and these modes have the lowest index numbers. Mode I has terminal point I2(I) = I on the ground plane, endpoint I1(I) is above the ground plane, endpoint I3(I) is the corresponding image point below the ground plane, and segment JA(I) is the lowest-numbered real segment with endpoint I.

Between statements 58 and 65, the program sets up the rest of the real modes. Modes of this type have the terminal point I2 on or above the ground plane. Each of these real modes (with index I greater than NPGP) has an image which is established between statements 65 and 75.

Below statement 75, the last part of the program counts the number of dipole modes sharing segment J, denoted by ND(J). It also stores a list of the dipole modes sharing segment J, denoted by MD(J,K). A segment may be shared by as many as four modes.

APPENDIX 7. Subroutine PDISS

Subroutine PDISS is listed in Fig. 11. This subroutine calculates the time-average power (DISS) dissipated in the lumped loads and the imperfectly conducting wire. The power is calculated for one segment at a time, and the total power dissipated is the sum of the powers dissipated on the various segments. On segment K, CJA and CJB denote the currents at endpoints IA(K) and IB(K). RLA and RLB denote the lumped resistors inserted in segment K at endpoints IA and IB.

This subroutine is suitable for a wire structure in free space and also for a wire structure over a perfect ground plane. If there is no ground plane, the total number of segments NS must be supplied as the tenth calling parameter, and DISS denotes the power dissipated on the entire structure. If there is a ground plane, the number of real segments NRS is supplied instead, and DISS denotes the power dissipated on the real segments.

APPENDIX 8. Subroutine ZFF

Subroutine ZFF, listed in Fig. 12, calculates the far-zone field of a sinusoidal electric monopole in free space. The monopole has endpoints at (XA,YA,ZA) and (XB,YB,ZB). (These symbols denote kx, ky and kz.) Let E denote the electric field intensity. The dimensionless range-independent field is defined by

$$F = (r/\lambda)\ e^{jkr}\ E$$

EP1 and ET1 denote F_ϕ and F_θ for the mode with unit current at (XA,YA,ZA). EP2 and ET2 denote F_ϕ and F_θ for the mode with unit current at (XB,YB,ZB). The far field vanishes in the endfire direction where GK = 0.

APPENDIX 9. Subroutine ZGMM

Subroutine ZGMM, listed in Fig. 13, calculates the mutual impedance between two filamentary monopoles with sinusoidal current distributions. The dipole-dipole mutual impedance in Eq. 20 of reference 2 is the sum of four monopole-monopole mutual impedances. The monopole impedances are calculated by ZGS with Simpson's rule or by ZGMM with closed-form expressions in terms of exponential integrals.

If the monopoles are parallel, let the z axis be parallel with both monopoles. The coordinate origin may be selected arbitrarily. S1 and S2 denote the z coordinates of the endpoints of the test monopole, T1 and T2 are the z coordinates of the endpoints of the expansion monopole, and D is the perpendicular distance (displacement) between the monopoles. The mutual impedance of parallel monopoles is calculated in the last part of ZGMM below statement 110.

For skew monopoles, let the test monopole s lie in the xy plane and the expansion monopole t in the plane z = D. (D is the perpendicular distance between the parallel planes.) If the monopoles are viewed along a line of sight parallel with the z axis, the extended axes of the two monopoles will appear to intersect at a point on the xy plane. Let s measure the distance along the axis of the test monopole with origin at the apparent intersection. S1 and S2 denote the s coordinates of the endpoints of the test monopole. Similarly, let t measure distance along the axis of the expansion monopole with origin at the apparent intersection. T1 and T2 denote the t coordinates of the endpoints of the expansion monopole. Let $\hat{s}$ and $\hat{t}$ be unit vectors parallel with the positive s and t axes, respectively. Then CPSI = $\hat{s} \cdot \hat{t} = \cos \psi$. The monopole lengths are d_s and d_t, and the remaining input data are defined as follows:

$$\text{CGDS} = \cos kd_s$$
$$\text{SGD1} = \sin kd_s$$
$$\text{SGD2} = \sin kd_t$$

ZGMM calls EXPJ for the exponential integrals. ZGMM is specialized for sinusoidal monopoles in free space. In ZGMM the input data S1, S2, T1, T2 and D denote ks_1, ks_2, kt_1, kt_2 and kd, respectively. Otherwise, ZGMM is the same as GGMM.

The output data from ZGMM are the impedances P11, P12, P21 and P22. In defining these impedances, the reference direction is from S1 to S2 for the current on monopole s, and from T1 to T2 for the current on monopole t. In the impedance P_{ij}, the first subscript is 1 or 2 if the test dipole has terminals at S1 or S2 on monopole s. The second subscript is 1 or 2 if the expansion dipole has terminals at T1 or T2 on monopole t. The endpoint coordinates S1, S2, T1 and T2 may be positive or negative. The monopole lengths d_s and d_t are assumed positive in defining the input data CGDS, SGD1 and SGD2.

For parallel monopoles, CPSI = 1 or -1. S1, S2, T1 and T2 are cartesian coordinates for parallel monopoles and spherical coordinates for skew monopoles. For skew monopoles, the radial coordinates S1, S2, T1 and T2 tend to infinity as the angle ψ tends to zero or π. Therefore, if the monopoles are within 4.5 degrees of being parallel, they are approximated by parallel dipoles.

APPENDIX 10. Subroutine ZGS

Subroutine ZGS, listed in Fig. 14, calculates the mutual impedance between two filamentary monopoles with sinusoidal current distributions. (The dipole-dipole mutual impedance in Eq. 20 of reference 2 is the sum of four monopole-monopole mutual impedances.) The endpoints of the axial test monopole s are at (XA,YA,ZA) and (XB,YB,ZB), and the endpoints of the expansion monopole t are at (X1,Y1,Z1) and (X2,Y2,Z2). DS and DT denote the lengths of monopoles s and t. Dimensionless forms are used for the input data. For example, XA, AK, DS and DT denote kx_a, ka, kd_s and kd_t. CAS, CBS and CGS are the direction cosines of monopole s, and CA, CB and CG are the direction cosines of monopole t.

If INT = 0, ZGS calls ZGMM for the closed-form impedance calculations. Otherwise ZGS calculates the mutual impedance via Simpson's-rule integration with the following number of sample points: IP = INT + 1. If the monopoles are parallel with small displacement, ZGS calls ZGMM to avoid the difficulties of numerical integration.

For the fields of the test monopole, ZGS uses Eqs. 75 and 76 of reference 2. The current distribution on the expansion monopole is given by Eq. 74 of reference 2. With an origin at (X1,Y1,Z1), the coordinate T measures distance along the expansion monopole. Thus T is the integration variable.

Let the coordinate s measure distance along the test monopole with origin at (XA,YA,ZA). From any point T on monopole t, construct a line to the test monopole such that the line is perpendicular to the test monopole. SZ denotes the s coordinate of the intersection of this line with the test monopole. The length of the line is the radial coordinate ρ, and RS denotes ρ^2. R1 and R2 are the distances from (XA,YA,ZA) and (XB,YB,ZB) to the point T. C1 is the current at T for the mode with terminals at (X1,Y1,Z1), and C2 is the current at T for the other mode with terminals at (X2,Y2,Z2). C denotes the Simpson's-rule weighting coefficient.

Below statement 300, ZGS performs some analytic geometry in preparation for calling ZGMM. The remaining part of this Appendix concerns this last part of subroutine ZGS.

Let $\hat{s}$ denote a unit vector from (XA,YA,ZA) toward (XB,YB,ZB), and let $\hat{t}$ denote a unit vector from (X1,Y1,Z1) toward (X2,Y2,Z2). Then $\hat{s} \cdot \hat{t} = \cos \psi$ = CC where ψ is the angle formed by the axes of the two monopoles. Let monopole s lie in one plane P_s and monopole t in another parallel plane P_t. CAD, CBD and CGD are the direction cosines of the unit vector $\hat{d} = \hat{t} \times \hat{s} / \sin \psi$ which is perpendicular to both planes. To obtain the distance DK between the planes, we construct a vector $\underline{R}_{11}$ from (XA,YA,ZA) to (X1,Y1,Z1) and take DK = $\underline{R}_{11} \cdot \hat{d}$.

Construct a line from (X1,Y1,Z1) to the test monopole, such that the line is perpendicular to the test monopole. SZ denotes the s coordinate of the intersection of this line with the test monopole, and the cartesian coordinates of this intersection are XZ, YZ and ZZ. The direction cosines of $\hat{s} \times \hat{d}$ are CAP, CBP and CGP.

From the point (X1,Y1,Z1) in plane P_t, construct a perpendicular line to the point (XP1,YP1,ZP1) in the plane P_s. This line is parallel with $\hat{d}$ and has length DK. Let $\underline{R}$ represent a vector from (XZ,YZ,ZZ) to (XP1,YP1,ZP1). P1 denotes $\underline{R} \cdot (\hat{s} \times \hat{d})$. S1 and T1 are defined in Appendix 9.

Subroutine ZGS is essentially the same as GGS except the medium is specialized to free space in ZGS.

APPENDIX 11. Subroutine ZSURF

Subroutine ZSURF, listed in Fig. 15, calculates the surface impedance of a solid circular-cylindrical wire with exterior excitation. ZS denotes the surface impedance in ohms, and the input data are defined as follows:

AK	ka, where a is the wire radius
CMM	conductivity of the wire in megamhos/m
FMC	frequency, MHz

The surface impedance is defined by $Z_S = E_Z/H_\phi$ where the fields E and H are evaluated at the surface of the wire. This subroutine calculates the impedance for the lowest order cylindrical mode with fields E_Z and H_ϕ independent of ϕ. The wire is considered to be a good conductor in the sense that the displacement current is negligible in comparison with the conduction current. In the present application, we require the surface impedance appropriate for the current distribution $I(z) = \sin kz$. For a highly conducting wire, however, this impedance is considered to be the same as that for a uniformly distributed current. BER, BEI, BERP and BEIP denote the Kelvin functions ber, bei, ber' and bei' with argument x. When x is less than 8, BES and BES1 denote the Bessel functions J_0 and J_1 with argument

$$z = x\,e^{-j\pi/4}\ .$$

When x is greater than 8,

$$J_0/BES = J_1/BES1 = \frac{e^{.707x}}{\sqrt{2\pi x}}\ .$$

```
      SUBROUTINE ANTI(IA,IB,I1,I2,I3,IWRCJ,IWRITF,I12,ICC,INS,JA,JB,
     2JPP,MD,N,NCM,NO,NLD,NPGP,NRS,NS,C,CDK,SDK,CJ,CMM,D,EFF,G,VG,VJ,
     3Y11,Z11,ZH,ZLD)
      COMPLEX C(ICC,ICC),CJ(1),VG(1),VJ(1),ZLD(1)
      COMPLEX Y11,Z11,ZH,CJI,VJI,DYY
      DIMENSION IA(1),IB(1),NO(1),CDK(1),SDK(1),D(1),I1(1),I2(1),I3(1)
      DIMENSION MD(INS,4),JA(1),JB(1),IGEN(1),JGEN(1)
1     FORMAT(8X,'J=',I5,5X,'VG(J)=',2F10.2)
2     FORMAT(8X,'J=',I5,5X,'ZLD(J)=',2F10.2)
3     FORMAT(10X,'I',3X,'MAGNITUDE',3X,'PHASE',9X,'REAL',8X,'IMAGINARY')
4     FORMAT(1X,I110,1F10.3,1F10.1,2F15.7)
5     FORMAT(1H0)
      IF(I12.GT.0)I12=1
      IF(IWRITF.LE.0)GO TO 14
      DO 10 J=1,NRS
      AVG=CABS(VG(J))
10    IF(AVG.GT.0.01)WRITE(6,1)J,VG(J)
      WRITE(6,5)
      IF(NLD.LE.0)GO TO 14
      ZMAX=.0
      DO 12 J=1,NRS
      AZL=CABS(ZLD(J))
      IF(AZL.LT.0.1)GO TO 12
      ZMAX=1.
      WRITE(6,2)J,ZLD(J)
12    CONTINUE
      IF(ZMAX.GT.0.5)WRITE(6,5)
14    DO 30 I=1,NCM
      CJ(I)=(.0,.0)
      L=2
      FAC=1.
      K=JA(I)
      IF(I.GT.NPGP)GO TO 15
      L=1
      FAC=2.
      IF(JB(I).LT.K)K=JB(I)
15    DO 25 KK=1,L
      KA=IA(K)
      KB=IB(K)
      JJ=K
      FI=FAC
      IF(KB.EQ.I2(I))GO TO 22
      IF(KB.EQ.I1(I))FI=-FAC
      CJ(I)=CJ(I)+FI*VG(JJ)
      GO TO 25
22    IF(KA.EQ.I3(I))FI=-FAC
      JJ=K+NRS
      CJ(I)=CJ(I)+FI*VG(JJ)
25    K=JB(I)
      VJ(I)=CJ(I)
      K=I+JPP
30    IF(I.GT.NPGP)VJ(K)=-VJ(I)
      ISYM=1
      IF(N.EQ.NPGP)ISYM=0
      IWR=0
      CALL CROUT(C,CJ,ICC,ISYM,IWR,I12,NCM)
      I12=12
      CMAX=.0
      DO 80 I=1,NCM
      CA=CABS(CJ(I))
      K=I+JPP
      IF(I.GT.NPGP)CJ(K)=-CJ(I)
```

Fig. 5. Subroutine ANTI.

```
80    IF(CA.GT.CMAX)CMAX=CA
      IF(IWRCJ.GE.1)WRITE(6,3)
      Y11=(.0,.0)
      G=1.
      EFF=.0
      Z11=(.0,.0)
      IF(CMAX.LE.0.)GO TO 500
      G=.0
      DO 90 I=1,N
      CJI=CJ(I)
      VJI=VJ(I)
      DYY=CJI*CONJG(VJI)
      IF(I.LE.NCM)Y11=Y11+DYY
      G=G+REAL(DYY)
      IF(IWRCJ.LE.0)GO TO 90
      IF(I.GT.NCM)GO TO 90
      CA=CABS(CJI)/CMAX
      PH=.0
      IF(CA.GT.1.E-30)PH=57.29578*ATAN2(AIMAG(CJI),REAL(CJI))
      WRITE(6,4)I,CA,PH,CJI
90    CONTINUE
      IF(IWRCJ.GE.1)WRITE(6,5)
      G=G/2.
      Z11=1./Y11
      EFF=100.
      IF(CMM.LE.0. .AND. NLD.LE.0)GO TO 500
      CALL       PDISS(IA,IB,INS,I1,I2,I3,MD,ND,NLD,NRS,CJ,CMM,D,CDK,
     2SDK,DISS,ZH,ZLD)
      EFF=100.*(G-DISS)/G
500.  RETURN
      END
```

Fig. 5. Subroutine ANTI - continued

```
      SUBROUTINE CROUT(C,S,ICC,ISYM,IWR,I12,N)
      COMPLEX C(ICC,ICC),S(1)
      COMPLEX F,P,SS,T
    2 FORMAT(1X,I15,1F10.3,1F15.7,1F10.0)
    5 FORMAT(1H0)
      IF(I12.NE.1)GO TO 22
      IF(N.EQ.1)S(1)=S(1)/C(1,1)
      IF(N.EQ.1)GO TO 100
      IF(ISYM.NE.0)GO TO 8
      DO 6 I=1,N
      DO 6 J=1,N
    6 C(J,I)=C(I,J)
    8 F=C(1,1)
      DO 10 L=2,N
   10 C(1,L)=C(1,L)/F
      DO 20 L=2,N
      LLL=L-1
      DO 20 I=L,N
      F=C(I,L)
      DO 11 K=1,LLL
   11 F=F-C(I,K)*C(K,L)
      C(I,L)=F
      IF(L.EQ.I)GO TO 20
      P=C(L,L)
      IF(ISYM.EQ.0)GO TO 15
      F=C(L,I)
      DO 12 K=1,LLL
   12 F=F-C(L,K)*C(K,I)
      C(L,I)=F/P
      GO TO 20
   15 F=C(I,L)
      C(L,I)=F/P
   20 CONTINUE
   22 DO 30 L=1,N
      P=C(L,L)
      T=S(L)
      IF(L.EQ.1)GO TO 30
      LLL=L-1
      DO 25 K=1,LLL
   25 T=T-C(L,K)*S(K)
   30 S(L)=T/P
      DO 38 L=2,N
      I=N-L+1
      II=I+1
      T=S(I)
      DO 35 K=II,N
   35 T=T-C(I,K)*S(K)
   38 S(I)=T
      IF(IWR.LE.0) GO TO 100
      WRITE(6,5)
      CNOR=.0
      DO 40 I=1,N
      SA=CABS(S(I))
   40 IF(SA.GT.CNOR)CNOR=SA
      IF(CNOR.LE.0.)CNOR=1.
      DO 44 I=1,N
      SS=S(I)
      SA=CABS(SS)
      SNOR=SA/CNOR
      PH=.0
      IF(SA.GT.0.)PH=57.29578*ATAN2(AIMAG(SS),REAL(SS))
   44 WRITE(6,2)I,SNOR,SA,PH
      WRITE(6,5)
  100 RETURN
      END
```

Fig. 6. Subroutine CROUT.

```
      SUBROUTINE EXPJ(V1,V2,W12)
      COMPLEX EC,E15,S,T,UC,VC,V1,V2,W12,Z
      DIMENSION V(21),W(21),D(16),E(16)
      DATA V/   0.22284667E 00,
     20.11889321E 01,0.29927363E 01,0.57751436E 01,0.98374674E 01,
     20.15982874E 02,0.93307812E-01,0.49269174E 00,0.12155954E 01,
     20.22699495E 01,0.36676227E 01,0.54253366E 01,0.75659162E 01,
     20.10120228E 02,0.13130282E 02,0.16654408E 02,0.20776479E 02,
     20.25623894E 02,0.31407519E 02,0.38530683E 02,0.48026086E 02/
      DATA W/   0.45896460E 00,
     20.41700083E 00,0.11337338E 00,0.10399197E-01,0.26101720E-03,
     20.89854791E-06,0.21823487E 00,0.34221017E 00,0.26302758E 00,
     20.12642582E 00,0.40206865E-01,0.85638778E-02,0.12124361E-02,
     20.11167440E-03,0.64599267E-05,0.22263169E-06,0.42274304E-08,
     20.39218973E-10,0.14565152E-12,0.14830270E-15,0.16005949E-19/
      DATA D/   0.22495842E 02,
     2 0.74411568E 02,-0.41431576E 03,-0.78754339E 02, 0.11254744E 02,
     2 0.16021761E 03,-0.23862195E 03,-0.50094687E 03,-0.68487854E 02,
     2 0.12254778E 02,-0.10161976E 02,-0.47219591E 01, 0.79729681E 01,
     2-0.21069574E 02, 0.22046490E 01, 0.89728244E 01/
      DATA E/   0.21103107E 02,
     2-0.37959787E 03,-0.97489220E 02, 0.12900672E 03, 0.17949226E 02,
     2-0.12910931E 03,-0.55705574E 03, 0.13524801E 02, 0.14696721E 03,
     2 0.17949528E 02,-0.32981014E 00, 0.31028836E 02, 0.81657657E 01,
     2 0.22236961E 02, 0.39124892E 02, 0.81636799E 01/
      Z=V1
      DO 100 JIM=1,2
      X=REAL(Z)
      Y=AIMAG(Z)
      E15=(.0,.0)
      AB=CABS(Z)
      IF(AB.EQ.0.)GO TO 90
      IF(X.GE.0. .AND. AB.GT.10.)GO TO 80
      YA=ABS(Y)
      IF(X.LE.0. .AND. YA.GT.10.)GO TO 80
      IF(YA-X.GE.17.5.OR.YA.GE.6.5.OR.X+YA.GE.5.5.OR.X.GE.3.)GO TO 20
      IF(X.LE.-9.)GO TO 40
      IF(YA-X.GE.2.5)GO TO 50
      IF(X+YA.GE.1.5)GO TO 30
   10 N=6.+3.*AB
      E15=1./(N-1.)-Z/N**2
   15 N=N-1
      E15=1./(N-1.)-Z*E15/N
      IF(N.GE.3)GO TO 15
      E15=Z*E15-CMPLX(.577216+ALOG(AB),ATAN2(Y,X))
      GO TO 90
   20 J1=1
      J2=6
      GO TO 31
   30 J1=7
      J2=21
   31 S=(.0,.0)
      YS=Y*Y
      DO 32 I=J1,J2
      XI=V(I)+X
      CF=W(I)/(XI*XI+YS)
   32 S=S+CMPLX(XI*CF,-YA*CF)
      GO TO 54
   40 T3=X*X-Y*Y
      T4=2.*X*YA
      T5=X*T3-YA*T4
      T6=X*T4+YA*T3
```

Fig. 7. Subroutine EXPJ.

```
      UC=CMPLX(D(11)+D(12)*X+D(13)*T3+T5-E(12)*YA-E(13)*T4,
     2         E(11)+E(12)*X+E(13)*T3+T6+D(12)*YA+D(13)*T4)
      VC=CMPLX(D(14)+D(15)*X+D(16)*T3+T5-E(15)*YA-E(16)*T4,
     2         E(14)+E(15)*X+E(16)*T3+T6+D(15)*YA+D(16)*T4)
      GO TO 52
50    T3=X*X-Y*Y
      T4=2.*X*YA
      T5=X*T3-YA*T4
      T6=X*T4+YA*T3
      T7=X*T5-YA*T6
      T8=X*T6+YA*T5
      T9=X*T7-YA*T8
      T10=X*T8+YA*T7
      UC=CMPLX(D(1)+D(2)*X+D(3)*T3+D(4)*T5+D(5)*T7+T9-(E(2)*YA+E(3)*T4
     2+E(4)*T6+E(5)*T8),E(1)+E(2)*X+E(3)*T3+E(4)*T5+E(5)*T7+T10+
     3(D(2)*YA+D(3)*T4+D(4)*T6+D(5)*T8))
      VC=CMPLX(D(6)+D(7)*X+D(8)*T3+D(9)*T5+D(10)*T7+T9-(E(7)*YA+E(8)*T4
     2+E(9)*T6+E(10)*T8),E(6)+E(7)*X+E(8)*T3+E(9)*T5+E(10)*T7+T10+
     3(D(7)*YA+D(8)*T4+D(9)*T6+D(10)*T8))
52    EC=UC/VC
      S=EC/CMPLX(X,YA)
54    EX=EXP(-X)
      T=EX*CMPLX(COS(YA),-SIN(YA))
      E15=S*T
56    IF(Y.LT.0.)E15=CONJG(E15)
      GO TO 90
80    E15=.409319/(Z+.193044)+.421831/(Z+1.02666)+.147126/(Z+2.56788)+
     2.206335E-1/(Z+4.90035)+.107401E-2/(Z+8.18215)+.158654E-4/(Z+
     312.7342)+.317031E-7/(Z+19.3957)
      E15=E15*CEXP(-Z)
90    IF(JIM.EQ.1)W12=E15
100   Z=V2
      Z=V2/V1
      TH=ATAN2(AIMAG(Z),REAL(Z))-ATAN2(AIMAG(V2),REAL(V2))
     2+ATAN2(AIMAG(V1),REAL(V1))
      AB=ABS(TH)
      IF(AB.LT.1.)TH=.0
      IF(TH.GT.1.)TH=6.2831853
      IF(TH.LT.-1.)TH=-6.2831853
      W12=W12-E15+CMPLX(.0,TH)
      RETURN
      END
```

Fig. 7. Subroutine EXPJ - continued

```
      SUBROUTINE IDANT(IA,IB,ICC,INS,INT,I1,I2,I3,JA,JB,JPP,MD,N,NCM,
     2ND,NLD,NP,NPGP,NRS,NS,AK,C,CMM,D,FMC,CDK,SDK,X,Y,Z,ZH,ZLD)
      COMPLEX ZS,ZH,P(2,2),Q(2,2),CIJ,ZSAM,ZOPP
      COMPLEX C(ICC,ICC),ZLD(1)
      DIMENSION X(1),Y(1),Z(1),IA(1),IB(1),ND(1),CDK(1),SDK(1),D(1)
      DIMENSION I1(1),I2(1),I3(1),JA(1),JB(1),MD(INS,4)
      DATA TP/6.28318/
2     FORMAT(8X,'AK=',F8.6,5X,'DMAX=',F8.4,5X,'DMIN=',F8.4)
      DO 10 I=1,NCM
      DO 10 J=1,NCM
10    C(I,J)=(.0,.0)
      DMAX=.0
      DMIN=100.
      DO 20 J=1,NRS
      DJ=D(J)
      IF(DJ.GT.DMAX)DMAX=DJ
      IF(DJ.LT.DMIN)DMIN=DJ
      CDK(J)=COS(DJ)
      SDK(J)=SIN(DJ)
      K=J+NRS
      CDK(K)=CDK(J)
20    SDK(K)=SDK(J)
      IF(DMIN.LT.AK)GO TO 21
      IF(DMAX.GT.3.)GO TO 21
      IF(AK.GT.0.1)GO TO 21
      GO TO 22
21    WRITE(6,2)AK,DMAX,DMIN
      N=0
      RETURN
22    DO 200 K=1,NS
      NDK=ND(K)
      KA=IA(K)
      KB=IB(K)
      DK=D(K)
      DO 200 L=1,NS
      NDL=ND(L)
      LA=IA(L)
      LB=IB(L)
      DL=D(L)
      NIL=0
      DO 200 II=1,NDK
      I=MD(K,II)
      IF(I.GT.NCM)GO TO 200
      FI=1.
      IF(KB.EQ.I2(I))GO TO 36
      IF(KB.EQ.I1(I))FI=-1.
      IS=1
      GO TO 40
36    IF(KA.EQ.I3(I))FI=-1.
      IS=2
40    DO 200 JJ=1,NDL
      J=MD(L,JJ)
      IF(I.GT.J)GO TO 200
      FJ=1.
      IF(LB.EQ.I2(J))GO TO 46
      IF(LB.EQ.I1(J))FJ=-1.
      JS=1
      GO TO 50
46    IF(LA.EQ.I3(J))FJ=-1.
      JS=2
50    IF(NIL.NE.0)GO TO 168
      NIL=1
```

Fig. 8. Subroutine IDANT.

```
      IF(K.EQ.L)GO TO 120
      IND=(LA-KA)*(LB-KA)*(LA-KB)*(LB-KB)
      IF(IND.EQ.0)GO TO 80
C     SEGMENTS K AND L SHARE NO POINTS
      CALL ZGS(X(KA),Y(KA),Z(KA),X(KB),Y(KB),Z(KB),X(LA),Y(LA),Z(LA),
     2X(LB),Y(LB),Z(LB),AK,DK,CDK(K),SDK(K),DL,SDK(L),INT,
     3P(1,1),P(1,2),P(2,1),P(2,2))
      GO TO 168
C     SEGMENTS K AND L SHARE ONE POINT (THEY INTERSECT)
   80 KG=0
      JM=KB
      JC=KA
      KF=1
      IND=(KB-LA)*(KB-LB)
      IF(IND.NE.0)GO TO 82
      JC=KB
      KF=-1
      JM=KA
      KG=3
   82 LG=3
      JP=LA
      LF=-1
      IF(LB.EQ.JC)GO TO 83
      JP=LB
      LF=1
      LG=0
   83 SGN=KF*LF
      CPSI=((X(JP)-X(JC))*(X(JM)-X(JC))+(Y(JP)-Y(JC))*(Y(JM)-Y(JC))
     2+(Z(JP)-Z(JC))*(Z(JM)-Z(JC)))/(DK*DL)
      CALL ZGMM(.0,DK,.0,DL,AK,CDK(K),SDK(K),SDK(L),CPSI
     2,Q(1,1),Q(1,2),Q(2,1),Q(2,2))
      DO 98 KK=1,2
      KP=IABS(KK-KG)
      DO 98 LL=1,2
      LP=IABS(LL-LG)
      P(KP,LP)=SGN*Q(KK,LL)
   98 CONTINUE
      GO TO 168
C     K=L  (SELF REACTION OF SEGMENT K)
  120 S=.5
      IF(KA.NE.LA)S=-.5
      CALL ZGMM(.0,DK,DK*(.5-S),DK*(.5+S),AK,CDK(K),SDK(K),SDK(K),1.
     2,P(1,1),P(1,2),P(2,1),P(2,2))
  168 CIJ=FI*FJ*P(IS,JS)
      IF(J.GT.NCM)GO TO 190
      C(I,J)=C(I,J)+CIJ
      IF(I.NE.J)C(J,I)=C(J,I)+CIJ
      GO TO 200
  190 JG=J-JPP
      C(I,JG)=C(I,JG)-CIJ
  200 CONTINUE
      ZH=(.0,.0)
      IF(CMM.LE.0.)GO TO 262
      CALL ZSURF(AK,CMM,FMC,ZS)
      ZH=ZS/(4.*TP*AK)
      DO 260 K=1,NS
      NDK=ND(K)
      ZSAM=2.*ZH*(D(K)-SDK(K)*CDK(K))/SDK(K)**2
      DO 210 II=1,NDK
      I=MD(K,II)
  210 IF(I.LE.NCM)C(I,I)=C(I,I)+ZSAM
      IF(NDK.EQ.1)GO TO 260
```

Fig. 8. Subroutine IDANT - continued

```
      ZOPP=2.*ZH*(SDK(K)-D(K)*CDK(K))/SDK(K)**2
      KA=IA(K)
      KB=IB(K)
      DO 260 II=1,NDK
      I=MD(K,II)
      IF(I.GT.NCM)GO TO 260
      FI=1.
      IF(KB.EQ.I2(I))GO TO 236
      IF(KB.EQ.I1(I))FI=-1.
      IS=1
      GO TO 240
  236 IF(KA.EQ.I3(I))FI=-1.
      IS=2
  240 DO 260 JJ=1,NDK
      J=MD(K,JJ)
      IF(I.GE.J)GO TO 260
      FJ=1.
      IF(KB.EQ.I2(J))GO TO 246
      IF(KB.EQ.I1(J))FJ=-1.
      JS=1
      GO TO 250
  246 IF(KA.EQ.I3(J))FJ=-1.
      JS=2
  250 IF(IS.EQ.JS)CIJ=FI*FJ*ZSAM
      IF(IS.NE.JS)CIJ=FI*FJ*ZOPP
      IF(J.GT.NCM)GO TO 259
      C(I,J)=C(I,J)+CIJ
      C(J,I)=C(J,I)+CIJ
      GO TO 260
  259 JG=J-JPP
      C(I,JG)=C(I,JG)-CIJ
  260 CONTINUE
  262 IF(NLD.LE.0)GO TO 300
      DO 280 I=1,NCM
      JJA=JA(I)
      J1=JJA
      II2=I2(I)
      II1=I1(I)
      IF(II2.EQ.IB(J1))J1=J1+NRS
      IF(I.LE.NPGP)GO TO 270
      JJB=JB(I)
      J2=JJB
      IF(II2.EQ.IB(J2))J2=J2+NRS
      C(I,I)=C(I,I)+ZLD(J1)+ZLD(J2)
      JJJ=JJA
      DO 268 K=1,2
      NDJ=ND(JJJ)
      DO 266 JJ=1,NDJ
      J=MD(JJJ,JJ)
      IF(J.EQ.I)GO TO 266
      IF(I2(J).NE.II2)GO TO 266
      FI=1.
      IF(K.EQ.2)GO TO 264
      IF(I1(J).NE.II1)FI=-1.
      C(I,J)=C(I,J)+FI*ZLD(J1)
      GO TO 266
  264 IF(I3(J).NE.I3(I))FI=-1.
      C(I,J)=C(I,J)+FI*ZLD(J2)
  266 CONTINUE
  268 JJJ=JJB
      GO TO 280
  270 IF(IB(J1).LE.NPGP)J1=J1+NRS
```

Fig. 8. Subroutine IDANT - continued

```
      C(I,I)=C(I,I)+2.*ZLD(J1)
      NDJ=ND(JJA)
      DO 278 JJ=1,NDJ
      J=MD(JJA,JJ)
      IF(J.EQ.I)GO TO 278
      IF(I2(J).NE.II2)GO TO 278
      FI=1.
      IF(I1(J).NE.II1)FI=-1.
      C(I,J)=C(I,J)+2.*FI*ZLD(J1)
  278 CONTINUE
  280 CONTINUE
  300 RETURN
      END
```

Fig. 8. Subroutine IDANT - continued

```
      SUBROUTINE IFFLD(IA,IB,INS,I1,I2,I3,MD,N,ND,NS,CDK,CJ,D,
     2EPP,ETT,EPH,ETH,G,GPP,GTT,PH,SDK,TH,X,Y,Z)
      COMPLEX EPH,ETH,CJI,ET1,ET2,EP1,EP2
      COMPLEX CJ(1),EPP(1),ETT(1)
      DIMENSION IA(1),IB(1),ND(1),CDK(1),SDK(1),D(1),X(1),Y(1),Z(1)
      DIMENSION I1(1),I2(1),I3(1),MD(INS,4)
      DATA CJI/(.0,-.530888E-2)/
      THR=.0174533*TH
      CTH=COS(THR)
      STH=SIN(THR)
      PHR=.0174533*PH
      CPH=COS(PHR)
      SPH=SIN(PHR)
      DO 130 I=1,N
      ETT(I)=(.0,.0)
  130 EPP(I)=(.0,.0)
      DO 140 K=1,NS
      KA=IA(K)
      KB=IB(K)
      CALL ZFF(X(KA),Y(KA),Z(KA),X(KB),Y(KB),Z(KB),D(K)
     2,CDK(K),SDK(K),CTH,STH,CPH,SPH,ET1,ET2,EP1,EP2)
      NDK=ND(K)
      DO 140 II=1,NDK
      I=MD(K,II)
      FI=1.
      IF(KB.EQ.I2(I))GO TO 136
      IF(KB.EQ.I1(I))FI=-1.
      EPP(I)=EPP(I)+FI*EP1
      ETT(I)=ETT(I)+FI*ET1
      GO TO 140
  136 IF(KA.EQ.I3(I))FI=-1.
      EPP(I)=EPP(I)+FI*EP2
      ETT(I)=ETT(I)+FI*ET2
  140 CONTINUE
      EPH=(.0,.0)
      ETH=(.0,.0)
  200 DO 260 I=1,N
      ETH=ETH+CJ(I)*ETT(I)
  260 EPH=EPH+CJ(I)*EPP(I)
      APP=CABS(EPH)
      ATT=CABS(ETH)
      GPP=APP*APP/(30.*G)
      GTT=ATT*ATT/(30.*G)
      RETURN
      END
```

Fig. 9. Subroutine IFFLD.

```
      SUBROUTINE ISORT(IA,IB,ICC,ICJ,INS,IWRITE,I1,I2,I3,JA,JB,
     2MAX,MIN,MD,N,NCM,ND,NP,NS,NRP,NRS,NPGP,NSGP,DC,XC,YC,ZC)
      DIMENSION JSP(20),DC(1),XC(1),YC(1),ZC(1)
      DIMENSION I1(1),I2(1),I3(1),JA(1),JB(1)
      DIMENSION IA(1),IB(1),ND(1),MD(INS,4)
    1 FORMAT(5X,'J',1X,'IA(J)',1X,'IB(J)',12X,'K',
     21X,'IA(K)',1X,'IB(K)',7X,'DC(J)')
    2 FORMAT(1X,3I5,1I15,2I5,2F15.5)
    3 FORMAT(1X,I15,3F10.5,I15,3F10.5)
    4 FORMAT(1X,6I5,1I15,6I5)
    5 FORMAT(1H0)
    6 FORMAT(5X,'I',4X,'XC(I)',5X,'YC(I)',5X,'ZC(I)',
     2 5X,'J',4X,'XC(J)',5X,'YC(J)',5X,'ZC(J)')
    7 FORMAT(5X,'I',3X,'JA',3X,'JB',3X,'I1',3X,'I2',3X,'I3',
     214X,'K',3X,'JA',3X,'JB',3X,'I1',3X,'I2',3X,'I3')
      IGPP=NPGP+1
      NPI=NRP-NPGP
C  NEXT SET UP THE IMAGE SEGMENTS
      DO 18 J=1,NRS
      K=J+NRS
      IA(K)=IA(J)
      IF(IA(J).GT.NPGP)IA(K)=IA(J)+NPI
      IB(K)=IB(J)
   18 IF(IB(J).GT.NPGP)IB(K)=IB(J)+NPI
C  NEXT SET UP THE IMAGE POINTS
      DO 20 I=IGPP,NRP
      J=I+NPI
      XC(J)=XC(I)
      YC(J)=YC(I)
   20 ZC(J)=-ZC(I)
C  NEXT CALCULATE THE SEGMENT LENGTHS DC(J)
      IF(IWRITE.LE.0)GO TO 22
      WRITE(6,5)
      WRITE(6,1)
   22 DO 25 J=1,NRS
      K=IA(J)
      L=IB(J)
      DX=XC(K)-XC(L)
      DY=YC(K)-YC(L)
      DZ=ZC(K)-ZC(L)
      DC(J)=SQRT(DX*DX+DY*DY+DZ*DZ)
      K=J+NRS
      DC(K)=DC(J)
   25 IF(IWRITE.GE.1)WRITE(6,2)J,IA(J),IB(J),K,IA(K),IB(K),DC(J)
      IF(IWRITE.LE.0)GO TO 32
      WRITE(6,5)
      WRITE(6,6)
      DO 30 I=1,NRP
      IF(I.GT.NPGP)GO TO 28
      WRITE(6,3)I,XC(I),YC(I),ZC(I)
      GO TO 30
   28 J=I+NPI
      WRITE(6,3)I,XC(I),YC(I),ZC(I),J,XC(J),YC(J),ZC(J)
   30 CONTINUE
      WRITE(6,5)
C  CHECK INPUT DATA FOR CONSISTENCE
   32 N=0
      MIN=100
      MAX=100
      IF(NPGP.LE.0)GO TO 40
      DO 38 I=1,NPGP
      L=0
```

Fig. 10. Subroutine ISORT.

```
      DO 35 J=1,NSGP
      K=(IA(J)-I)*(IB(J)-I)
   35 IF(K.EQ.0)L=L+1
      IF(L.LT.MAX)MAX=L
   38 N=N+2*L-1
   40 IF(NRP.LE.NPGP)GO TO 50
      DO 46 I=IGPP,NRP
      L=0
      DO 44 J=1,NRS
      K=(IA(J)-I)*(IB(J)-I)
   44 IF(K.EQ.0)L=L+1
      IF(L.LT.MIN)MIN=L
   46 N=N+2*(L-1)
   50 IF(N.LE.0 .OR. N.GT.ICJ)GO TO 500
      IF(MAX.LE.0 .OR. MIN.LE.0)GO TO 500
      IF(NPGP.LE.0)GO TO 58
C  SET UP THE MODES AT THE GROUND PLANE THAT WILL NOT HAVE IMAGES
      DO 56 I=1,NPGP
      J=0
   52 J=J+1
      IAJ=IA(J)
      IBJ=IB(J)
      KK=(IAJ-I)*(IBJ-I)
      IF(J.EQ.NSGP)GO TO 54
      IF(KK.NE.0)GO TO 52
   54 JA(I)=J
      JB(I)=J+NRS
      I2(I)=I
      I1(I)=IBJ
      IF(IBJ.EQ.I)I1(I)=IAJ
   56 I3(I)=I1(I)+NPI
   58 I=NPGP
      N=NPGP
      NCM=NPGP
      JPP=0
      IF(NRS.EQ.NPGP)GO TO 75
C  SET UP THE REST OF THE REAL MODES
      DO 65 K=1,NRP
      NJK=0
      DO 60 J=1,NRS
      IND=(IA(J)-K)*(IB(J)-K)
      IF(IND.NE.0)GO TO 60
      NJK=NJK+1
      JSP(NJK)=J
   60 CONTINUE
      MOD=NJK-1
      IF(MOD.LE.0)GO TO 65
      DO 62 IMD=1,MOD
      I=I+1
      IPD=IMD+1
      JAI=JSP(IMD)
      JA(I)=JAI
      JBI=JSP(IPD)
      JB(I)=JBI
      I1(I)=IA(JAI)
      IF(IA(JAI).EQ.K)I1(I)=IB(JAI)
      I2(I)=K
      I3(I)=IA(JBI)
   62 IF(IA(JBI).EQ.K)I3(I)=IB(JBI)
   65 CONTINUE
      NCM=I
      JPP=NCM-NPGP
```

Fig. 10. Subroutine ISORT - continued

```
C   SET UP THE IMAGE MODES
      DO 70 I=IGPP,NCM
      K=I+JPP
      JA(K)=JA(I)+NRS
      JB(K)=JB(I)+NRS
      IIA=I1(I)
      IIB=I2(I)
      IIC=I3(I)
      I1(K)=IIA
      IF(IIA.GT.NPGP)I1(K)=IIA+NPI
      I2(K)=IIB
      IF(IIB.GT.NPGP)I2(K)=IIB+NPI
      I3(K)=IIC
   70 IF(IIC.GT.NPGP)I3(K)=IIC+NPI
      N=2*NCM-NPGP
   75 MAX=0
      MIN=100
C   ND(J) = NUMBER OF DIPOLE MODES SHARING SEGMENT J
C   MD(J,K) = LIST OF DIPOLES SHARING SEGMENT J
      DO 100 J=1,NS
      DO 80 K=1,4
   80 MD(J,K)=0
      K=0
      DO 90 I=1,N
      JAI=JA(I)
      JBI=JB(I)
      L=(JAI-J)*(JBI-J)
      IF(L.NE.0)GO TO 90
      K=K+1
      MD(J,K)=I
   90 CONTINUE
      ND(J)=K
      IF(K.GT.MAX)MAX=K
  100 IF(K.LT.MIN)MIN=K
      IF(IWRITE.LE.0)GO TO 500
      WRITE(6,7)
      DO 110 I=1,NCM
      IF(I.GT.NPGP)GO TO 108
      WRITE(6,4)I,JA(I),JB(I),I1(I),I2(I),I3(I)
      GO TO 110
  108 K=I+JPP
      WRITE(6,4)I,JA(I),JB(I),I1(I),I2(I),I3(I),K,JA(K),JB(K),
     2I1(K),I2(K),I3(K)
  110 CONTINUE
      WRITE(6,5)
  500 RETURN
      END
```

Fig. 10. Subroutine ISORT - continued

```
      SUBROUTINE PDISS(IA,IB,INS,I1,I2,I3,MD,ND,NLD,NS ,CJ,CMM,D,CDK,
     2SDK,DISS,ZH,ZLD)
      COMPLEX CJ(1),ZH,CJA,CJB,ZLA,ZLB,ZLD(1)
      DIMENSION CDK(1),SDK(1),D(1),I1(1),I2(1),I3(1),IA(1),IB(1),ND(1)
      DIMENSION MD(INS,4)
      RH=REAL(ZH)
      DISS=.0
      DO 100 K=1,NS
      IF(CMM.LE.0.)GO TO 10
      FA=2.*RH*(D(K)-SDK(K)*CDK(K))/SDK(K)**2
      FB=4.*RH*(SDK(K)-D(K)*CDK(K))/SDK(K)**2
 10   KA=IA(K)
      KB=IB(K)
      CJA=(.0,.0)
      CJB=(.0,.0)
      NDK=ND(K)
      DO 40 II=1,NDK
      I=MD(K,II)
      FI=1.
      IF(KB.EQ.I2(I))GO TO 36
      IF(KB.EQ.I1(I))FI=-1.
      CJA=CJA+FI*CJ(I)
      GO TO 40
 36   IF(KA.EQ.I3(I))FI=-1.
      CJB=CJB+FI*CJ(I)
 40   CONTINUE
      IF(NLD.LE.0)GO TO 50
      AJA=CABS(CJA)**2
      BJB=CABS(CJB)**2
      KK=K+NS
      RLA=REAL(ZLD(K))
      RLB=REAL(ZLD(KK))
      DISS=DISS+AJA*RLA+BJB*RLB
 50   IF(CMM.LE.0.)GO TO 100
      DISS=DISS+FA*(CABS(CJA)**2+CABS(CJB)**2)
     2+FB*(REAL(CJA)*REAL(CJB)+AIMAG(CJA)*AIMAG(CJB))
 100  CONTINUE
      RETURN
      END
```

Fig. 11. Subroutine PDISS.

```
      SUBROUTINE ZFF(XA,YA,ZA,XB,YB,ZB,D
     2,CKD,SKD,CTH,STH,CPH,SPH,ET1,ET2,EP1,EP2)
      COMPLEX EJA,EJB,EP1,EP2,ES1,ES2,ET1,ET2
      CA=(XB-XA)/D
      CB=(YB-YA)/D
      CG=(ZB-ZA)/D
      G=(CA*CPH+CB*SPH)*STH+CG*CTH
      GK=1.-G*G
      ET1=(.0,.0)
      ET2=(.0,.0)
      EP1=(.0,.0)
      EP2=(.0,.0)
      IF(GK.LT..001)GO TO 200
      A=XA*STH*CPH+YA*STH*SPH+ZA*CTH
      B=XB*STH*CPH+YB*STH*SPH+ZB*CTH
      EJA=CMPLX(COS(A),SIN(A))
      EJB=CMPLX(COS(B),SIN(B))
      SGD=SIN(G*D)
      CGD=COS(G*D)
      ES1=30.*EJA*CMPLX(SGD-G*SKD,CKD-CGD)/GK/SKD
      ES2=30.*EJB*CMPLX(G*SKD-SGD,CKD-CGD)/GK/SKD
      T=(CA*CPH+CB*SPH)*CTH-CG*STH
      P=-CA*SPH+CB*CPH
      ET1=T*ES1
      ET2=T*ES2
      EP1=P*ES1
      EP2=P*ES2
  200 RETURN
      END
```

Fig. 12. Subroutine ZFF.

```
      SUBROUTINE ZGMM(S1,S2,T1,T2,D,CGDS,SGD1,SGD2,CPSI,P11,P12,P21,P22)
      DOUBLE PRECISION R1,R2,DPO,SIS,TS1,TS2,ST1,ST2,CD,BD,CPSS,SPSI,SK
     2,TL1,TL2,TD1,TD2,SDI,DPSI,DD,ZD
      COMPLEX E(2,2),F(2,2),GAM,P11,P12,P21,P22
      COMPLEX EB,EC,EK,EL,EKL,EGZ1,ES1,ES2,ET1,ET2,EXPA,EXPB
      COMPLEX EGZ(2,2),GM(2),GP(2)
      DATA ETA,GAM,PI/376.727,(.0,1.),3.14159/
      DSO=D*D
      SGDS=SGD1
      IF(S2.LT.S1)SGDS=-SGD1
      SGDT=SGD2
      IF(T2.LT.T1)SGDT=-SGD2
      IF(ABS(CPSI).GT.0.997)GO TO 110
      ES1=CEXP(GAM*S1)
      ES2=CEXP(GAM*S2)
      ET1=CEXP(GAM*T1)
      ET2=CEXP(GAM*T2)
      DD=D
      DPSI=CPSI
      TD1=T1
      TD2=T2
      CPSS=DPSI*DPSI
      CD=DD/DSQRT(1.D0-CPSS)
      C=CD
      BD=CD*DPSI
      B=BD
      EB=CEXP(GAM*CMPLX(.0,B))
      EC=CEXP(GAM*CMPLX(.0,C))
      DO 10 K=1,2
      DO 10 L=1,2
10    E(K,L)=(.0,.0)
      TS1=TD1*TD1
      TS2=TD2*TD2
      DPO=DD*DD
      SI=S1
      DO 100 I=1,2
      FI=(-1)**I
      SDI=SI
      SIS=SDI*SDI
      ST1=2.*SDI*TD1*DPSI
      ST2=2.*SDI*TD2*DPSI
      R1=DSQRT(DPO+SIS+TS1-ST1)
      R2=DSQRT(DPO+SIS+TS2-ST2)
      EK=EB
      DO 50 K=1,2
      FK=(-1)**K
      SK=FK*SDI
      EL=EC
      DO 40 L=1,2
      FL=(-1)**L
      EKL=EK*EL
      XX=FK*BD+FL*CD
      TL1=FL*TD1
      TL2=FL*TD2
      RR1=R1+SK+TL1
      RR2=R2+SK+TL2
      CALL EXPJ(GAM*CMPLX(RR1,-XX),GAM*CMPLX(RR2,-XX),EXPA)
      CALL EXPJ(GAM*CMPLX(RR1,XX),GAM*CMPLX(RR2,XX),EXPB)
      E(K,L)=E(K,L)+FI*(EXPA*EKL+EXPB/EKL)
40    EL=1./EC
50    EK=1./EB
      ZD=SDI*DPSI
```

Fig. 13. Subroutine ZGMM.

```
      ZC=ZD
      EGZI=CEXP(GAM*ZC)
      RR1=R1+ZD-TD1
      RR2=R2+ZD-TD2
      CALL EXPJ(GAM*RR1,GAM*RR2,EXPB)
      RR1=R1-ZD+TD1
      RR2=R2-ZD+TD2
      CALL EXPJ(GAM*RR1,GAM*RR2,EXPA)
      F(1,1)=2.*SGDS*(.0,1.)*EXPA/EGZI
      F(1,2)=2.*SGDS*(.0,1.)*EXPB*EGZI
  100 SI=S2
      CST=-ETA/(16.*PI*SGDS*SGDT)
      P11=CST*(( F(1,1)+E(2,2)*ES2-E(1,2)/ES2)*ET2
     A         +(-F(1,2)-E(2,1)*ES2+E(1,1)/ES2)/ET2)
      P12=CST*((-F(1,1)-E(2,2)*ES2+E(1,2)/ES2)*ET1
     B         +( F(1,2)+F(2,1)*ES2-E(1,1)/ES2)/ET1)
      P21=CST*((-F(2,1)-F(2,2)*ES1+E(1,2)/ES1)*ET2
     C         +( F(2,2)+E(2,1)*ES1-E(1,1)/ES1)/ET2)
      P22=CST*(( F(2,1)+E(2,2)*ES1-E(1,2)/ES1)*ET1
     D         +(-F(2,2)-E(2,1)*ES1+E(1,1)/ES1)/ET1)
      RETURN
  110 IF(CPSI.LT.0.)GO TO 120
      TA=T1
      TB=T2
      GO TO 130
  120 TA=-T1
      TB=-T2
      SGDT=-SGDT
  130 SI=S1
      DO 150 I=1,2
      TJ=TA
      DO 140 J=1,2
      ZIJ=TJ-SI
      R=SQRT(DSQ+ZIJ*ZIJ)
      W=R+ZIJ
      IF(ZIJ.LT.0.)W=DSQ/(R-ZIJ)
      V=R-ZIJ
      IF(ZIJ.GT.0.)V=DSQ/(R+ZIJ)
      IF(J.EQ.1)V1=V
      IF(J.EQ.1)W1=W
      EGZ(I,J)=CEXP(GAM*ZIJ)
  140 TJ=TB
      CALL EXPJ(GAM*V1,GAM*V,GP(I))
      CALL EXPJ(GAM*W1,GAM*W,GM(I))
  150 SI=S2
      CST=ETA/(8.*PI*SGDS*SGDT)
      P11=CST*(GM(2)*EGZ(2,2)+GP(2)/EGZ(2,2)
     2-CGDS*(GM(1)*EGZ(1,2)+GP(1)/EGZ(1,2)))
      P12=CST*(-GM(2)*EGZ(2,1)-GP(2)/EGZ(2,1)
     2+CGDS*(GM(1)*EGZ(1,1)+GP(1)/EGZ(1,1)))
      P21=CST*(GM(1)*EGZ(1,2)+GP(1)/EGZ(1,2)
     2-CGDS*(GM(2)*EGZ(2,2)+GP(2)/EGZ(2,2)))
      P22=CST*(-GM(1)*EGZ(1,1)-GP(1)/EGZ(1,1)
     2+CGDS*(GM(2)*EGZ(2,1)+GP(2)/EGZ(2,1)))
      RETURN
      END
```

Fig. 13. Subroutine ZGMM - continued

```
      SUBROUTINE ZGS(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,AK,
     2DS,CDS,SDS,DT,SDT,INT,P11,P12,P21,P22)
      COMPLEX CST,EJ1,EJ2,EJA,EJB,ER1,ER2,ET1,ET2,P11,P12,P21,P22,GAM
      COMPLEX SGDS,SGDT
      DATA ETA,GAM,PI/376.727,(.0,1.),3.14159/
      CA=(X2-X1)/DT
      CB=(Y2-Y1)/DT
      CG=(Z2-Z1)/DT
      CAS=(XB-XA)/DS
      CBS=(YB-YA)/DS
      CGS=(ZB-ZA)/DS
      CC=CA*CAS+CB*CBS+CG*CGS
      IF(ABS(CC).GT.0.997)GO TO 200
   20 SZ=(X1-XA)*CAS+(Y1-YA)*CBS+(Z1-ZA)*CGS
      IF(INT.EQ.0)GO TO 300
      CGDS=CDS
      SGDS=CMPLX(.0,SDS)
      SGDT=CMPLX(.0,SDT)
      INS=2*(INT/2)
      IF(INS.LT.2)INS=2
      IP=INS+1
      DELT=DT/INS
      T=.0
      DSZ=CC*DELT
      P11=(.0,.0)
      P12=(.0,.0)
      P21=(.0,.0)
      P22=(.0,.0)
      AKS=AK*AK
      SGN=-1.
      DO 100 IN=1,IP
      ZZ1=SZ
      ZZ2=SZ-DS
      XXZ=X1+T*CA-XA-SZ*CAS
      YYZ=Y1+T*CB-YA-SZ*CBS
      ZZZ=Z1+T*CG-ZA-SZ*CGS
      RS=XXZ**2+YYZ**2+ZZZ**2
      R1=SQRT(RS+ZZ1**2)
      EJA=CMPLX(COS(R1),-SIN(R1))
      EJ1=EJA/R1
      R2=SQRT(RS+ZZ2**2)
      EJB=CMPLX(COS(R2),-SIN(R2))
      EJ2=EJB/R2
      ER1=EJA*SGDS+ZZ1*EJ1*CGDS-ZZ2*EJ2
      ER2=-EJB*SGDS+ZZ2*EJ2*CGDS-ZZ1*EJ1
      FAC=.0
      IF(RS.GT.AKS)FAC=(CA*XXZ+CB*YYZ+CG*ZZZ)/RS
      ET1=CC*(EJ2-EJ1*CGDS)+FAC*ER1
      ET2=CC*(EJ1-EJ2*CGDS)+FAC*ER2
      C=3.+SGN
      IF(IN.EQ.1 .OR. IN.EQ.IP)C=1.
      C1=C*SIN(DT-T)
      C2=C*SIN(T)
      P11=P11+ET1*C1
      P12=P12+ET1*C2
      P21=P21+ET2*C1
      P22=P22+ET2*C2
      T=T+DELT
      SZ=SZ+DSZ
  100 SGN=-SGN
      CST=-(.0,1.)*ETA*DELT/(12.*PI*SGDS*SGDT)
      P11=CST*P11
```

Fig. 14. Subroutine ZGS.

```
      P12=CST*P12
      P21=CST*P21
      P22=CST*P22
      RETURN
  200 SZ1=(X1-XA)*CAS+(Y1-YA)*CBS+(Z1-ZA)*CGS
      RH1=SQRT((X1-XA-SZ1*CAS)**2+(Y1-YA-SZ1*CBS)**2+(Z1-ZA-SZ1*CGS)**2)
      SZ2=SZ1+DT*CC
      RH2=SQRT((X2-XA-SZ2*CAS)**2+(Y2-YA-SZ2*CBS)**2+(Z2-ZA-SZ2*CGS)**2)
      DDK=(RH1+RH2)/2.
      IF(DDK.GT.20.*AK .AND. INT.GT.0)GO TO 20
      IF(DDK.LT.AK)DDK=AK
      CALL ZGMM(.0,DS,SZ1,SZ2,DDK,CDS,SDS,SDT,1.,P11,P12,P21,P22)
      RETURN
  300 SS=SQRT(1.-CC*CC)
      CAD=(CGS*CB-CBS*CG)/SS
      CBD=(CAS*CG-CGS*CA)/SS
      CGD=(CBS*CA-CAS*CB)/SS
      DK=(X1-XA)*CAD+(Y1-YA)*CBD+(Z1-ZA)*CGD
      DK=ABS(DK)
      IF(DK.LT.AK)DK=AK
      XZ=XA+SZ*CAS
      YZ=YA+SZ*CBS
      ZZ=ZA+SZ*CGS
      XP1=X1-DK*CAD
      YP1=Y1-DK*CBD
      ZP1=Z1-DK*CGD
      CAP=CBS*CGD-CGS*CBD
      CBP=CGS*CAD-CAS*CGD
      CGP=CAS*CBD-CBS*CAD
      P1=CAP*(XP1-XZ)+CBP*(YP1-YZ)+CGP*(ZP1-ZZ)
      T1=P1/SS
      S1=T1*CC-SZ
      CALL ZGMM(S1,S1+DS,T1,T1+DT,DK,CDS,SDS,SDT,CC,P11,P12,P21,P22)
      RETURN
      END
```

Fig. 14. Subroutine ZGS - continued

```
      SUBROUTINE ZSURF(AK,CMM,FMC,ZS)
      COMPLEX BES,BES1,ZS
      DATA ETA,SQT,TP/376.72727,1.41421356,6.2831853/
      SQSWE=1.E6*SQRT(CMM/TP/FMC/8.85433)
      X=AK*SQSWE
      IF(X.GT.8.)GO TO 50
      T=X/8.
      T2=T*T
      T4=T2*T2
      BER=((((((-.901E-5*T4+.122552E-2)*T4-.08349609)*T4+2.641914)*T4
     2-32.363456)*T4+113.77778)*T4-64.)*T4+1.
      BEI=((((((.11346E-3*T4-.01103667)*T4+.52185615)*T4-10.567658)*T4
     2+72.817777)*T4-113.77778)*T4+16.)*T2
      BERP=X*T2*((((((-.394E-5*T4+.45957E-3)*T4-.02609253)*T4+.66047849)
     2*T4-6.0681481)*T4+14.222222)*T4-4.)
      BEIP=X*((((((.4609E-4*T4-.379386E-2)*T4+.14677204)*T4-2.3116751)*
     2T4+11.377778)*T4-10.666667)*T4+.5)
      BES=CMPLX(BER,BEI)
      BES1=.707107*CMPLX(BERP-BEIP,BERP+BEIP)
      GO TO 100
   50 XP=.7071068*X
      X1=1./X
      F=((-.0459205*X1+.390625E-2)*X1+.08838835)*X1+1.
      T=((-.04603559*X1-.0625)*X1-.08838835)*X1-.39269907+XP
      BES=F*CMPLX(COS(T),SIN(T))
      F=((.11290231*X1+.03515625)*X1-.26516505)*X1+1.
      T=((.1160097*X1+.1875)*X1+.26516505)*X1+1.1780972+XP
      BES1=F*CMPLX(COS(T),SIN(T))
  100 ZS=-CMPLX(1.,-1.)*ETA*BES/BES1/SQT/SQSWE
      RETURN
      END
```

Fig. 15. Subroutine ZSURF.

COMPUTER PROGRAM FOR THIN-WIRE ANTENNA OVER A PERFECTLY CONDUCTING GROUND PLANE

J. H. Richmond

TECHNICAL REPORT 2902-19

October 1974

Grant No. NGL 36-008-138

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23365

ABSTRACT

A computer program is presented for a thin-wire antenna over a perfect ground plane. The analysis is performed in the frequency domain, and the exterior medium is free space. The antenna may have finite conductivity and lumped loads. The output data includes the current distribution, impedance, radiation efficiency and gain. The program uses sinusoidal bases and Galerkin's method.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Reference 1 presents a computer program and reference 2 presents the theory for thin-wire antennas and scatterers in a homogeneous conducting medium. The present program differs from reference 1 only in the following details:

a. The exterior medium is free space.
b. The antenna is situated over a perfect ground plane.
c. The wires have no dielectric sleeves.
d. The frequency is real.
e. Scattering problems are not considered.

To avoid unnecessary duplication, it is assumed the reader is familiar with the program in reference 1.

The program handles antennas constructed of straight wire segments. One or more segments may connect to the ground plane, or the antenna may be situated some distance away from the ground plane. No segment has both endpoints on the ground plane. This program can readily be modified to handle more general situations as in reference 1. The program uses the delta-gap model for the generators.

The method of images is employed to reduce the problem to a thin-wire structure in free space. The user sets up the geometry of the real wire configuration, and the program automatically sets up the image. If unlimited storage were available, one might set up a large mutual-impedance matrix for the wire antenna and its image in free space. Instead, this program takes advantage of the ground-plane symmetry and sets up the compressed matrix C(I,J). Only the currents on the real segments are treated as independent unknown quantities, and the image currents are dependent. All the currents, however, are influenced by the mutual couplings among all the segments including the images. In taking advantage of the mirror symmetry, we lose the advantage of having a symmetric matrix. However, the net gain in computational speed and storage is substantial.

In practice, many wire antennas operate over a ground plane with finite conductivity and finite extent. In many cases, however, one may substitute a perfectly conducting ground plane of infinite extent without unduly disturbing the antenna current distribution or impedance. After these quantities have been calculated, one may then take into account the finite ground plane in calculating the efficiency and patterns. The present program, however, assumes an idealized ground plane throughout.

The remaining sections present the computer program with enough explanation to enable an experienced engineer to use it.

II. THE INPUT DATA AND SUBROUTINE IWIRE

Fig. 1 is a Fortran listing of subroutine IWIRE. This subroutine is used to set up the input data. The following data must be read or programmed in IWIRE:

AL	wire radius a/λ
CMM	wire conductivity in megamhos/m
DPH	increment in far-field angle ϕ in degrees
FMC	frequency in MHz
TH	elevation angle θ in degrees for far-field pattern

To define the shape and size of the wire antenna, the input data includes the coordinates XC(I), YC(I), and ZC(I) of the wire endpoints, terminals and other current sampling stations along the wire axis. The unit of length is selected by the user, and SCALE is the conversion factor such that XC(I)*SCALE gives the coordinate of point I in meters. NPGP denotes the number of points on the ground plane, and NRP is the number of real points including those on the ground plane. Coordinates are supplied only for the real points. The ground plane coincides with the xy plane.

NRS denotes the number of real segments, and NSGP is the number of real segments having an endpoint on the ground plane. For each real segment J, the input data specifies the endpoints IA(J) and IB(J). In assigning index numbers to the segments, the lowest numbers must refer to those having an endpoint on the ground plane. In assigning index numbers to the points, the lowest numbers must refer to those on the ground plane.

Set IWRCJ = 1 to obtain a writeout of the antenna currents; otherwise IWRCJ = -1. Set IWRITE = 1 to obtain a writeout of the antenna geometry; otherwise IWRITE = -1. If INT = 0, the rigorous closed-form expressions will be used for the mutual impedance of sinusoidal dipoles and the calculations will tend to be slow but accurate. If INT is a positive integer, Simpson's rule will be used for the mutual-impedance calculations. The closed-form expressions are always used automatically for the most critical impedances. We usually use INT = 0 for multi-turn loop antennas with closely-spaced turns, and INT = 4 for general purpose. Simpson's rule uses INT integration intervals. Thus the accuracy and the execution time tend to increase with larger values of INT.

Set NLD = 0 if there are no lumped loads; otherwise NLD = 1. ZLD(J) denotes the impedance (in ohms) inserted in segment J at endpoint IA(J). A lumped load at endpoint IB(J) is denoted by ZLD(J + NRS). The user sets up only the real generators and lumped loads, and the program takes care of the images.

```
      SUBROUTINE IWIRE(IA,IB,INP,INS,INT,IWRCJ,IWRITE,NLD,NP,NS,NRP,
     2NRS,NPGP,NSGP,AL,CMM,DPH,FMC,SCALE,TH,VG,XC,YC,ZC,ZLD)
      DIMENSION IA(1),IB(1),XC(1),YC(1),ZC(1)
      COMPLEX VG(1),ZLD(1)
    4 FORMAT(4X,'INP=',I4,5X,'NP=',I4,5X,'INS=',I4,5X,'NS=',I4)
    5 FORMAT(1H0)
C  IA(J) AND IB(J) ARE ENDPOINTS OF SEGMENT J
C  XC(I),YC(I),ZC(I) ARE COORDINATES OF POINT I WITH ARBITRARY UNITS
C  NRP = NUMBER OF REAL POINTS, INCLUDING THOSE ON THE GROUND PLANE
C  NRS = NUMBER OF REAL SEGMENTS
C  NPGP = NUMBER OF POINTS ON THE GROUND PLANE
C  NSGP = NUMBER OF REAL SEGMENTS WITH ENDPOINT ON GROUND PLANE
C  NLD = NUMBER OF LUMPED LOADS
      DO 10 J=1,INS
      VG(J)=(.0,.0)
   10 ZLD(J)=(.0,.0)
C  SET UP THE REAL GENERATORS AND REAL LUMPED LOADS
      JGN=4
      VG(JGN)=(1.,0.)
      NLD=0
      INT=4
      IWRITE=1
      IWRCJ=1
      AL=.0001
      CMM=1.
      DPH=20.
      FMC=75.
      SCALE=1.
      TH=85.
      NSGP=2
      NPGP=2
      NRS=4
      NRP=5
      NS=2*NRS
      NP=2*NRP-NPGP
      WRITE(6,4)INP,NP,INS,NS
      IF(NS.GT.INS .OR. NP.GT.INP)GO TO 600
      DO 20 I=1,NRP
      XC(I)=.0
      YC(I)=.0
   20 ZC(I)=.0
C  NEXT SET UP THE REAL POINTS
      XC(1)=1.
      XC(3)=1.
      ZC(3)=.5
      ZC(4)=.5
      XC(5)=1.707
      ZC(5)=1.207
C  NEXT SET UP THE REAL SEGMENTS
      IA(1)=1
      IB(1)=3
      IA(2)=2
      IB(2)=4
      IA(3)=3
      IB(3)=4
      IA(4)=3
      IB(4)=5
  600 RETURN
      END
```

Fig. 1. Subroutine IWIRE.

The generator location is defined by JGN. The numbering system for the generators is the same as for lumped loads. If the generator is to be inserted in segment J at endpoint IA(J), the generator index JGN is the same as the segment index J. To insert a generator at endpoint IB(J), set JGN = J + NRS. VG(JGN) denotes the complex voltage of the generator. The reference direction for these voltages is from IA(J) toward IB(J). If the antenna is fed with several generators, delete JGN and merely input the generator voltages VG.

NP and NS denote the number of points and segments, respectively, for the complete system (antenna and image) in free space.

III. THE MAIN COMPUTER PROGRAM

The main computer program is listed in Fig. 2. This program calls subroutine IWIRE for the input data. Then it calls subroutine ISORT to generate and store the data for the image points and image segments and the length DC(J) of each segment. Then ISORT generates a list of sinusoidal dipole modes for the complete system (antenna and image) in free space. Dipole mode I has segments JA(I) and JB(I), terminals at point I2(I), and endpoints I1(I) and I3(I). This subroutine also generates the following information:

ND(J)	number of dipole modes sharing segment J
MD(J,K)	list of dipoles sharing segment J
NCM	size of the compressed matrix
N	number of dipole modes on the complete system

The following quantities must be specified in the main program:

ICC	dimension for the compressed matrix C(I,J)
ICJ	dimension related to number of dipole modes N
INP	dimension related to number of points NP
INS	dimension related to number of segments NS

In Fig. 2, all quantities having the same dimensions are dimensioned in the same or adjacent statements. The numerical values assigned to ICC, ICJ, INP and INS must agree with the dimensions actually reserved for the corresponding quantities in the COMPLEX and DIMENSION statements. ICC, ICJ, INP and INS must be at least as large as NCM, N, NP and NS, respectively. In Fig. 2, the main program is dimensioned for up to 150 modes, 90 points, 100 segments, and a compressed matrix as large as 30 by 30. If the wire antenna makes no contact with the ground plane, the compressed matrix will be exactly half as large as the full matrix. Otherwise NCM is somewhat larger than N/2.

```
C      INCLUDE ANTI;CROUT:EXPJ;IDANT;IFFLD;ISORT;IWIRE;ZFF;ZGMM;ZGS
C      INCLUDE PDISS:ZSURF
C   THIN-WIRE ANTENNA OVER PERFECT GROUND PLANE
C   SINUSOIDAL-GALERKIN FREQUENCY-DOMAIN
C   PROGRAM ORIGINATED BY J. H. RICHMOND, OHIO STATE UNIVERSITY
      COMPLEX EPH,ETH,Y11,Z11,ZH
      COMPLEX C(30,30)
      COMPLEX CJ(150),EP(150),ET(150),EPP(150),ETT(150),VJ(150)
      COMPLEX VG(150),ZLD(150)
      DIMENSION XC(90),YC(90),ZC(90),X(90),Y(90),Z(90)
      DIMENSION D(100),DC(100)
      DIMENSION IA(100),IB(100),MD(100,4),ND(100),CDK(100),SDK(100)
      DIMENSION I1(150),I2(150),I3(150),JA(150),JB(150)
      DATA PI,TP/3.14159,6.28318/
    1 FORMAT(8X,'JPP=',I5,5X,'MAX=',I5,5X,'MIN=',I5,5X,'N=',I5,5X,
     2'NCM=',I5)
    2 FORMAT(8X,'AL=',F8.6,5X,'CMM=',F8.4,5X,'FMC=',F8.2)
    3 FORMAT(8X,'EFF=',F7.2,3X,'Y11=',2F10.6,3X,'Z11=',2F8.2)
    4 FORMAT(8X,'PH',8X,'TH',8X,'DBP',7X,'DBT',7X,'GPP',7X,'GTT')
    5 FORMAT(1H0)
    6 FORMAT(1X,2F10.0,4F10.2)
      ICC=30
      ICJ=150
      INP=90
      INS=100
C   THE GEOMETRY OF THE THIN-WIRE STRUCTURE IS SPECIFIED IN SUB. IWIRE
      CALL        IWIRE(IA,IB,INP,INS,INT,IWRCJ,IWRITE,NLD,NP,NS,NRP,
     2NRS,NPGP,NSGP,AL,CMM,DPH,FMC,SCALE,TH,VG,XC,YC,ZC,ZLD)
      IF(NSGP.LT.NPGP)GO TO 500
      IF(NS.GT.INS .OR. NP.GT.INP)GO TO 500
      CALL        ISORT(IA,IB,ICC,ICJ,INS,IWRITE,I1,I2,I3,JA,JB,
     2MAX,MIN,MD,N,NCM,ND,NP,NS,NRP,NRS,NPGP,NSGP,DC,XC,YC,ZC)
C   NCM =  SIZE OF COMPRESSED MATRIX C(I,J)
      JPP=NCM-NPGP
      WRITE(6,1)JPP,MAX,MIN,N,NCM
      WRITE(6,5)
      AK=TP*AL
      WAVM=300./FMC
      WRITE(6,2)AL,CMM,FMC
      WRITE(6,5)
      TPL=SCALE*TP/WAVM
      IF(N.LE.0 .OR. N.GT.ICJ)GO TO 500
      IF(NCM.GT.ICC)GO TO 500
      IF(MAX.LE.0 .OR. MIN.LE.0)GO TO 500
      DO 90 J=1,NS
   90 D(J)=TPL*DC(J)
      DO 100 I=1,NP
      X(I)=TPL*XC(I)
      Y(I)=TPL*YC(I)
  100 Z(I)=TPL*ZC(I)
      CALL        IDANT(IA,IB,ICC,INS,INT,I1,I2,I3,JA,JB,JPP,MD,N,NCM,
     2ND,NLD,NP,NPGP,NRS,NS,AK,C,CMM,D,FMC,CDK,SDK,X,Y,Z,ZH,ZLD)
      IF(N.EQ.0)GO TO 500
      I12=1
      CALL        ANTI(IA,IB,I1,I2,I3,IWRCJ,IWRITE,I12,ICC,INS,JA,JB,
     2JPP,MD,N,NCM,ND,NLD,NPGP,NRS,NS,C,CDK,SDK,CJ,CMM,D,EFF,G,VG,VJ,
     3Y11,Z11,ZH,ZLD)
      IF(I12.NE.12)GO TO 500
      WRITE(6,3)EFF,Y11,Z11
      IF(G.EQ.1. .AND. EFF.EQ.0.)GO TO 500
      WRITE(6,5)
      LIM=1.5+360./DPH
```

Fig. 2. The MAIN computer program.

```
      WRITE(6,4)
      DO 300 IPH=1,LIM
      PH=DPH*(IPH-1)
      CALL        IFFLD(IA,IB,INS,I1,I2,I3,MD,N,NO,NS,COK,CJ,D,
     2EPP,ETT,EPH,ETH,C,GPP,GTT,PH,SOK,TH,X,Y,Z)
      DBP=.0
      DBT=.0
      IF(GPP.GT.0.)DBP=10.*ALOG10(GPP)
      IF(GTT.GT.0.)DBT=10.*ALOG10(GTT)
  300 WRITE(6,6)PH,TH,DBP,DBT,GPP,GTT
  500 CALL EXIT
      END
```

Fig. 2. The MAIN computer program - continued

X(I), Y(I) and Z(I) denote kx, ky and kz for point I, where $k = 2\pi/\lambda$. If calculations are desired for a given antenna at several frequencies, the frequency DO LOOP will begin just below the call to ISORT.

The main program calls subroutine IDANT to generate the compressed open-circuit impedance matrix C(I,J). Then subroutine ANTI is called to obtain the current distribution CJ(I) and the radiation efficiency EFF. ANTI also calculates the complex power input to the antenna, denoted by Y11, and the time-average input power G. If the antenna has only one generator and VG(JGN) = (1.,0.), then Y11 and Z11 denote the antenna admittance and impedance, respectively.

Finally, the antenna pattern is obtained by calling subroutine IFFLD. TH and PH denote the spherical coordinates θ and ϕ (in degrees) of the distant observer. GPP and GTT denote the ϕ-polarized and θ-polarized power gains, respectively, and DBP and DBT are the decibel versions. The user may want to increment θ as well as ϕ, but this will require only a trivial change in the main program. IFFLD is called once for each look angle (θ,ϕ). When $\theta = 90°$, GPP will vanish if the program has set up a valid system of images.

IV. AN EXAMPLE

Fig. 3 shows a simple antenna and its image, with a dotted line to indicate the ground plane. In Fig. 1, subroutine IWIRE sets up the following input data for this antenna:

VG(4) = 1	unit voltage generator at endpoint IA of segment 4
NLD = 0	no lumped loads
CMM = 1.	the wire conductivity is 1 megamho/m
NSGP = 2	2 real segments connect to the ground plane
NPGP = 2	2 points on the ground plane
NRS = 4	4 real segments
NRP = 5	5 real points

This planar antenna has 8 points (NP = 8) and 8 segments (NS = 8), and the numbering system is shown in Fig. 3. Note that the lowest numbers are assigned to the two points on the ground plane and the two segments terminating on the ground plane. The points and segments must be labeled with consecutive positive integers 1, 2, 3, . . For a given segment J, it makes no difference which end is labeled IA(J). In Fig. 3, each numeral located near a dot is the index I of that point. Each numeral located near the center of a line is the index J of that segment.

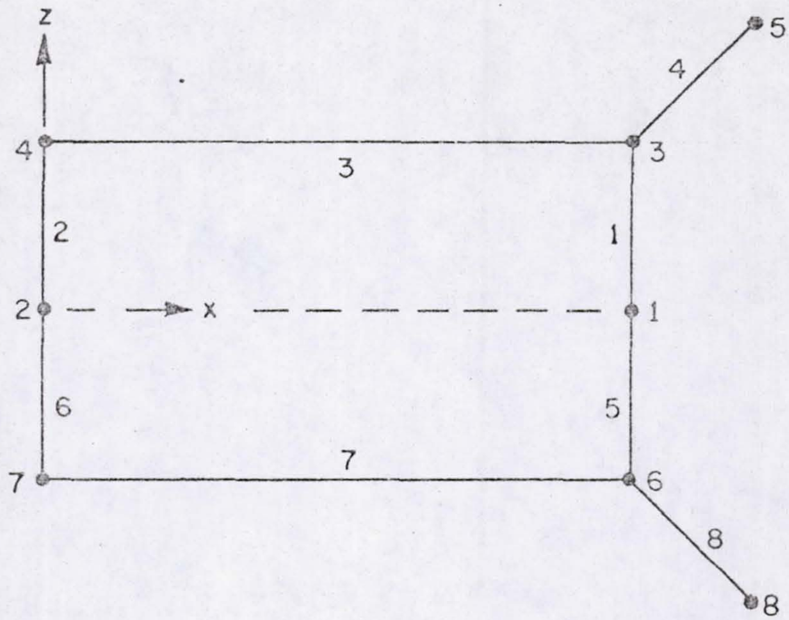

Fig. 3. Points and segments on a simple wire antenna lying in the xz plane.

Fig. 4 shows the same antenna and the eight dipole modes defined by subroutine ISORT. The arrows indicate the reference directions for the mode currents and voltages. The mode index number I is placed near the terminal point I2(I). Mode I is a sinusoidal basis function which vanishes at the endpoints I1(I) and I3(I) and has unit current at the terminal point. These are overlapping subsectional bases, and mode I extends over two intersecting segments JA(I) and JB(I). The reference direction for mode currents and voltages is from I1 to I2 toward I3. In Fig. 4, modes 1 and 2 have terminals at the ground plane, with segment JA above and segment JB below the ground plane. This type of mode has no image. Modes 3, 4 and 5 have images. The size of the compressed matrix is NCM = 5. If we did not take advantage of the ground-plane symmetry, the matrix size would be N = 8.

Table I presents some of the output data for this example and Table II lists the elements of the compressed matrix C(I,J) on return from subroutine IDANT. From Table I, the calculated impedance is Z11 = 959 + j 664 ohms. For the same antenna with perfect

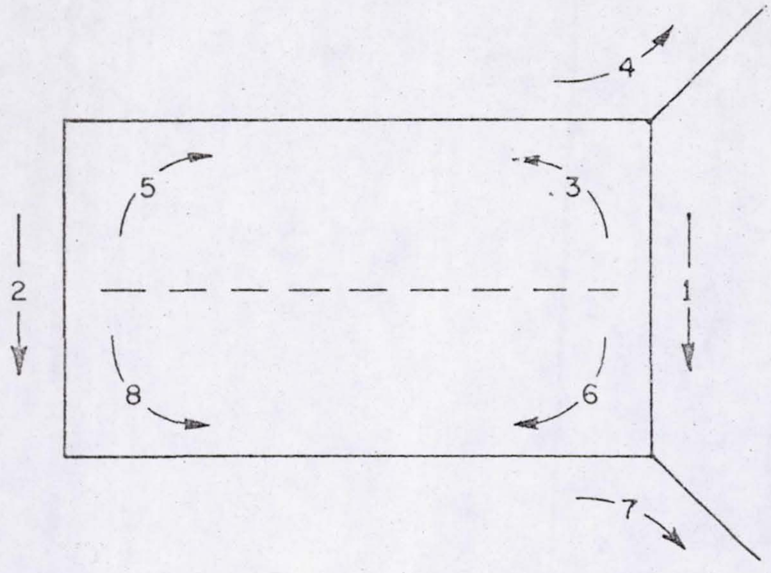

Fig. 4. Mode map for the antenna shown in Fig. 3.

conductivity, Z11 = 879 + j 749 ohms. The calculated results should not be considered accurate without checking the behavior as the wires are subdivided into more segments. The longest segment should not exceed $\lambda/4$. The thin-wire and delta-gap formulations are justified most readily if the wire radius does not exceed 0.007λ. Fortuitously, satisfactory results are often obtained for closed wire loops even when the wire radius is as large as 0.02λ. For dipoles, an upper limit of 0.007λ is recommended.

With 20-ohm resistive loads inserted in each end of each segment of the antenna in this example, the calculated impedance is 710 + j 206 ohms and the efficiency is 45.5 per cent with CMM = 1. With and without lumped loads, the results obtained with the present computer program show satisfactory agreement with those obtained with the program in reference 1. (A new version of subroutine SGANT was used for these tests because the original version in reference 1 does not always handle lumped loads properly.)

TABLE I

INP= 90 NP= 8 INS= 100 NS= 8

J	IA(J)	IB(J)	K	IA(K)	IB(K)	DC(J)
1	1	3	5	1	6	.50000
2	2	4	6	2	7	.50000
3	3	4	7	6	7	1.00000
4	3	5	8	6	8	.99985

I	XC(I)	YC(I)	ZC(I)	J	XC(J)	YC(J)	ZC(J)
1	1.00000	0.00000	0.00000				
2	0.00000	0.00000	0.00000				
3	1.00000	0.00000	.50000	6	1.00000	0.00000	-.50000
4	0.00000	0.00000	.50000	7	0.00000	0.00000	-.50000
5	1.70700	0.00000	1.20700	8	1.70700	0.00000	-1.20700

I	JA	JB	I1	I2	I3	K	JA	JB	I1	I2	I3
1	1	5	3	1	6						
2	2	6	4	2	7						
3	1	3	1	3	4	6	5	7	1	6	7
4	3	4	4	3	5	7	7	8	7	6	8
5	2	3	2	4	3	8	6	7	2	7	6

JPP= 3 MAX= 3 MIN= 1 N= 8 NCM= 5

AL= .000100 CMM= 1.0000 FMC= 75.00

J= 4 VG(J)= 1.00 0.00

I	MAGNITUDE	PHASE	REAL	IMAGINARY
1	1.000	99.9	-.0005441	.0031183
2	.870	91.1	-.0000537	.0027537
3	.730	-77.8	.0004892	-.0022579
4	.271	-34.7	.0007048	-.0004880
5	.638	-86.1	.0001362	-.0020153

EFF= 90.44 Y11= .000705 -.000488 Z11= 959.07 664.07

PH	TH	DBP	DBT	GPP	GTT
0.	85.	0.00	.66	0.00	1.16
20.	85.	-47.44	1.24	.00	1.33
40.	85.	-39.49	2.63	.00	1.83
60.	85.	-33.35	4.17	.00	2.61
80.	85.	-29.46	5.31	.00	3.40
100.	85.	-27.93	5.77	.00	3.77

TABLE II

Compressed Impedance Matrix

I	J	C(I,J)	C(J,I)
1	1	16.1 -j 720.5	16.1 -j 720.5
1	2	7.6 -j 6.7	7.6 -j 6.7
1	3	-15.9 -j 1059.9	- 7.9 -j 529.9
1	4	-18.8 -j 16.8	- 9.4 -j 8.4
1	5	- 5.2 +j 83.0	- 2.6 +j 41.5
2	2	16.1 -j 720.5	16.1 -j 720.5
2	3	- 5.2 +j 83.0	- 2.6 +j 41.5
2	4	-12.7 -j 77.3	- 6.4 -j 38.6
2	5	-15.9 -j 1059.9	- 7.9 -j 529.9
3	3	21.2 -j 326.0	21.2 -j 326.0
3	4	- 9.2 -j 22.6	- 9.2 -j 22.5
3	5	- 9.2 -j 396.0	- 9.2 -j 396.0
4	4	51.3 +j 60.8	51.3 +j 60.8
4	5	22.1 +j 420.9	22.1 +j 420.9
5	5	21.2 -j 326.0	21.2 -j 326.0

V. SUMMARY AND CONCLUSIONS

A computer program is presented for a thin-wire antenna over a perfectly conducting ground plane of infinite extent. The analysis is performed in the frequency domain, and the exterior medium is free space. The antenna may have finite conductivity and lumped loads. The output data includes the current distribution, impedance, radiation efficiency and gain. The program uses sinusoidal bases and Galerkin's method and takes advantage of the ground-plane symmetry to reduce the storage requirements and computation costs. The subroutines are included in alphabetical order in the Appendices with a brief explanation.

REFERENCES

1. J. H. Richmond, "Computer Program for Thin-Wire Structures in a Homogeneous Conducting Medium," NASA Contractor Report CR-2399, June 1974, for sale by the National Technical Information Service, Springfield, Virginia 22151, price $3.75.

2. J. H. Richmond, "Radiation and Scattering by Thin-Wire Structures in the Complex Frequency Domain," NASA Contractor Report CR-2396, May 1974, for sale by the National Technical Information Service, Springfield, Virginia 22151, price $3.25.

APPENDIX 1. Subroutine ANTI

Subroutine ANTI is listed in Fig. 5. Between statements 14 and 30, this subroutine sets up the excitation voltages CJ(I) and VJ(I) with the aid of the delta-gap model and the input data for the generator voltages VG(J). ANTI calls CROUT to obtain a solution for the simultaneous linear equations. On return from CROUT, the dipole mode currents are stored in CJ(I). The image currents are stored in CJ(K) in the DO LOOP ending with statement 80. The DO LOOP ending with statement 90 calculates the complex power input Y11 and the time-average power input G. The power dissipated (DISS) in the lumped loads and the imperfectly conducting wire is obtained by calling PDISS. Finally, the radiation efficiency EFF is calculated.

If IWRCJ is positive, ANTI writes a list of the dipole mode currents CJ(I). This list includes the normalized current magnitude, the phase in degrees, and the real and imaginary parts of the current.

APPENDIX 2. Subroutine CROUT

CROUT, listed in Fig. 6, solves a system of simultaneous linear equations with complex coefficients. This subroutine uses the method of P. D. Crout. Although this subroutine does not use pivoting, it is efficient and accurate in the present application. The input data are defined as follows:

C(I,J)	complex coefficients in the simultaneous equations
S(I)	excitation column
ICC	dimensions of C and S
ISYM	zero or one for symmetric or nonsymmetric matrix
IWR	one or zero to write or suppress the solution
I12	one or two if C is original or auxiliary matrix
N	size of the square matrix C

Of course, N must not exceed ICC. If IWR is a positive integer, the solution will be printed out with the following definitions:

I	index number of the solution S(I)
SNOR	normalized magnitude of S(I)
SA	absolute magnitude of S(I)
PH	phase of S(I) in degrees

On the first call to CROUT, C(I,J) contains the original matrix. I12 = 1 and CROUT generates the auxiliary square matrix, overlaying it in the same location C and destroying the original matrix. Then CROUT proceeds to generate the solution, storing it in S(I) and destroying the original excitation column.

Next we might want another solution of th ame system of simultaneous linear equations but with a new ex ation column. This could be obtained by recalculating the original matrix C(I,J) and the new excitation column and calling CROUT again with I12 = 1. However, there is no need to recalculate C(I,J). Instead generate the new excitation column, set I12 = 2 (or any integer other than 1) and call CROUT again. CROUT uses less computer time when I12 differs from 1.

APPENDIX 3. Subroutine EXPJ

Subroutine EXPJ, listed in Fig. 7, evaluates the exponential integral defined as follows:

$$W12 = \int_{V1}^{V2} \frac{e^{-v}\,dv}{v} = E_1(V1) - E_1(V2) + j\,2n\pi$$

where the integration path is the straight line from V1 to V2 on the complex v plane and

$$E_1(z) = \int_{z}^{\infty} \frac{e^{-t}\,dt}{t} \,.$$

$E_1(z)$ denotes the principal branch of the exponential integral. To generate W12, EXPJ calculates $E_1(V1)$, subtracts $E_1(V2)$ and adds $j2n\pi$. The integer n is zero unless the straight-line integration path intersects the negative real v axis at a point between V1 and V2. When there is such an intersection, n = 1 if V1 lies in quadrant 1 or 2 and n = -1 if V1 lies in quadrant 3 or 4.

APPENDIX 4. Subroutine IDANT

Subroutine IDANT is listed in Fig. 8. This subroutine stores the quantities CDK(J) = cos kd_j and SDK(J) = sin kd_j where d_j is the length of segment j. The program writes AK, DMAX and DMIN and aborts if

a. the length of the shortest segment is less than the wire radius, or
b. the longest segment has a length d such that kd exceeds 3, or
c. the wire radius a is such that ka exceeds 0.1.

IDANT calculates the elements in the compressed impedance matrix C(I,J) as follows. Select a source segment K and a receiving segment L, where K and L range from 1 to NS. The mutual impedances P11, P12, P21 and P22 between the two segments are obtained by calling ZGMM if K = L, ZGMM if the segments intersect, or ZGS if segments K and L do not intersect.

Now select a test dipole I sharing segment K, and an expansion dipole J sharing segment L. Add the appropriate segment-to-segment impedance to the dipole-to-dipole impedance C(I,J). When this procedure has been completed at statement 200, the impedances C(I,J) are appropriate for a perfectly conducting thin-wire system with no lumped loads.

Between statements 200 and 262, the impedance matrix C(I,J) is modified to account for the finite conductivity of the wire antenna. The surface impedance ZS is obtained by calling ZSURF. For each segment K, the program selects a test dipole I and an expansion dipole J sharing this segment. The contribution to C(I,J) associated with finite conductivity is ZSAM if dipoles I and J have terminals at the same end of segment K, and ZOPP if they have terminals at opposite ends. C(I,J) is not affected unless dipoles I and J share one or two segments.

Between statements 262 and 280, the impedance matrix is modified to account for the lumped loads. Each diagonal element C(I,I) is modified by adding the impedance of the lumped load inserted at the terminals of mode I. If modes I and J share a segment and have terminals at the same point, C(I,J) is modified by adding or subtracting the impedance of the lumped load inserted at the terminal end of this segment. (Add or subtract ZLD if mode currents I and J have the same or opposite reference directions on the shared segment.)

APPENDIX 5. Subroutine IFFLD

Subroutine IFFLD, listed in Fig. 9, calculates the far-zone field of the thin-wire antenna.

Let (r,θ,ϕ) denote the spherical coordinates of the distant observer, and let $E_\theta(I)$ and $E_\phi(I)$ denote the electric field intensities of dipole mode I with unit terminal current. Then

$$EPP(I) = (r/\lambda)\ e^{jkr}\ E_\phi(I)$$

$$ETT(I) = (r/\lambda)\ e^{jkr}\ E_\theta(I)$$

The field of sinusoidal dipole mode I may be regarded as the sum of the fields of each of its two segments. The field of segment K is obtained by calling subroutine ZFF, and EPP(I) and ETT(I) are generated by adding the appropriate numbers obtained from two different calls to ZFF. In the DO LOOP ending with statement 260, the antenna field is calculated as a weighted sum of the mode fields as follows:

$$EPH = \sum_{1}^{N} CJ(I)\ EPP(I)$$

$$ETH = \sum_{1}^{N} CJ(I)\ ETT(I)$$

where CJ(I) denotes the terminal current of mode I and EPH and ETH denote the dimensionless range-independent form of the antenna fields E_ϕ and E_θ.

G denotes the time-average input power to the antenna, and GPP and GTT are the ϕ-polarized and θ-polarized power gains. Subroutine IFFLD is called once for each angular direction. In the input data supplied to this subroutine, PH and TH denote ϕ and θ in degrees.

This subroutine is useful for wire antennas with or without a ground plane. For an antenna over a ground plane, IFFLD must be supplied with information on the complete system including the image.

APPENDIX 6. Subroutine ISORT

Subroutine ISORT, listed in Fig. 10, is described briefly in Section III. This subroutine sets up the image segments and points and calculates the segment lengths. Then it checks the input data for consistence. The data are considered inconsistent and the run is aborted if

a. NPGP is greater than zero but one of the points on the ground plane has no segment (with index J less than or equal to NSGP) connected to it, or
b. a real point situated above the ground plane has no segment (with index J less than or equal to NRS) connected to it.

Between statements 32 and 50, this program calculates the number of modes N on the complete structure. The run is aborted if the dimensions are inadequate.

Between statements 50 and 58, the program sets up the modes that will not have images. The number of modes of this type is NPGP, and these modes have the lowest index numbers. Mode I has terminal point I2(I) = I on the ground plane, endpoint I1(I) is above the ground plane, endpoint I3(I) is the corresponding image point below the ground plane, and segment JA(I) is the lowest-numbered real segment with endpoint I.

Between statements 58 and 65, the program sets up the rest of the real modes. Modes of this type have the terminal point I2 on or above the ground plane. Each of these real modes (with index I greater than NPGP) has an image which is established between statements 65 and 75.

Below statement 75, the last part of the program counts the number of dipole modes sharing segment J, denoted by ND(J). It also stores a list of the dipole modes sharing segment J, denoted by MD(J,K). A segment may be shared by as many as four modes.

APPENDIX 7. Subroutine PDISS

Subroutine PDISS is listed in Fig. 11. This subroutine calculates the time-average power (DISS) dissipated in the lumped loads and the imperfectly conducting wire. The power is calculated for one segment at a time, and the total power dissipated is the sum of the powers dissipated on the various segments. On segment K, CJA and CJB denote the currents at endpoints IA(K) and IB(K). RLA and RLB denote the lumped resistors inserted in segment K at endpoints IA and IB.

This subroutine is suitable for a wire structure in free space and also for a wire structure over a perfect ground plane. If there is no ground plane, the total number of segments NS must be supplied as the tenth calling parameter, and DISS denotes the power dissipated on the entire structure. If there is a ground plane, the number of real segments NRS is supplied instead, and DISS denotes the power dissipated on the real segments.

APPENDIX 8. Subroutine ZFF

Subroutine ZFF, listed in Fig. 12, calculates the far-zone field of a sinusoidal electric monopole in free space. The monopole has endpoints at (XA,YA,ZA) and (XB,YB,ZB). (These symbols denote kx, ky and kz.) Let E denote the electric field intensity. The dimensionless range-independent field is defined by

$$F = (r/\lambda)\ e^{jkr}\ E$$

EP1 and ET1 denote F_ϕ and F_θ for the mode with unit current at (XA,YA,ZA). EP2 and ET2 denote F_ϕ and F_θ for the mode with unit current at (XB,YB,ZB). The far field vanishes in the endfire direc where GK = 0.

APPENDIX 9. Subroutine ZGMM

Subroutine ZGMM, listed in Fig. 13, calculates the mutual impedance between two filamentary monopoles with sinusoidal cu distributions. The dipole-dipole mutual impedance in Eq. 20 o reference 2 is the sum of four monopole-monopole mutual impedan The monopole impedances are calculated by ZGS with Simpson's r by ZGMM with closed-form expressions in terms of exponential i

If the monopoles are parallel, let the z axis be paralle both monopoles. The coordinate origin may be selected arbitrar S1 and S2 denote the z coordinates of the endpoints of the test monopole, T1 and T2 are the z coordinates of the endpoints of t expansion monopole, and D is the perpendicular distance (displac between the monopoles. The mutual impedance of parallel monopo calculated in the last part of ZGMM below statement 110.

For skew monopoles, let the test monopole s lie in the xy plane and the expansion monopole t in the plane z = D. (D is the perpendicular distance between the parallel planes.) If the monop are viewed along a line of sight parallel with the z axis, the extended axes of the two monopoles will appear to intersect at a poi on the xy plane. Let s measure the distance along the axis of the test monopole with origin at the apparent intersection. S1 and S2 denote the s coordinates of the endpoints of the test monopole. Similarly, let t measure distance along the axis of the expansion monopole with origin at the apparent intersection. T1 and T2 denote the t coordinates of the endpoints of the expansion monopole. Let $\hat{s}$ and $\hat{t}$ be unit vectors parallel with the positive s and t axes, respectively. Then CPSI = $\hat{s} \cdot \hat{t} = \cos \psi$. The monopole lengths are d_s and d_t, and the remaining input data are defined as follows:

CGDS = $\cos kd_s$
SGD1 = $\sin kd_s$
SGD2 = $\sin kd_t$

ZGMM calls EXPJ for the exponential integrals. ZGMM is specialized for sinusoidal monopoles in free space. In ZGMM the input data S1, S2, T1, T2 and D denote ks_1, ks_2, kt_1, kt_2 and kd, respectively. Otherwise, ZGMM is the same as GGMM.

The output data from ZGMM are the impedances P11, P12, P21 and P22. In defining these impedances, the reference direction is from S1 to S2 for the current on monopole s, and from T1 to T2 for the current on monopole t. In the impedance P_{ij}, the first subscript is 1 or 2 if the test dipole has terminals at S1 or S2 on monopole s. The second subscript is 1 or 2 if the expansion dipole has terminals at T1 or T2 on monopole t. The endpoint coordinates S1, S2, T1 and T2 may be positive or negative. The monopole lengths d_s and d_t are assumed positive in defining the input data CGDS, SGD1 and SGD2.

For parallel monopoles, CPSI = 1 or -1. S1, S2, T1 and T2 are cartesian coordinates for parallel monopoles and spherical coordinates for skew monopoles. For skew monopoles, the radial coordinates S1, S2, T1 and T2 tend to infinity as the angle ψ tends to zero or π. Therefore, if the monopoles are within 4.5 degrees of being parallel, they are approximated by parallel dipoles.

APPENDIX 10. Subroutine ZGS

Subroutine ZGS, listed in Fig. 14, calculates the mutual impedance between two filamentary monopoles with sinusoidal current distributions. (The dipole-dipole mutual impedance in Eq. 20 of reference 2 is the sum of four monopole-monopole mutual impedances.) The endpoints of the axial test monopole s are at (XA,YA,ZA) and (XB,YB,ZB), and the endpoints of the expansion monopole t are at (X1,Y1,Z1) and (X2,Y2,Z2). DS and DT denote the lengths of monopoles s and t. Dimensionless forms are used for the input data. For example, XA, AK, DS and DT denote kx_a, ka, kd_s and kd_t. CAS, CBS and CGS are the direction cosines of monopole s, and CA, CB and CG are the direction cosines of monopole t.

If INT = 0, ZGS calls ZGMM for the closed-form impedance calculations. Otherwise ZGS calculates the mutual impedance via Simpson's-rule integration with the following number of sample points: IP = INT + 1. If the monopoles are parallel with small displacement, ZGS calls ZGMM to avoid the difficulties of numerical integration.

For the fields of the test monopole, ZGS uses Eqs. 75 and 76 of reference 2. The current distribution on the expansion monopole is given by Eq. 74 of reference 2. With an origin at (X1,Y1,Z1), the coordinate T measures distance along the expansion monopole. Thus T is the integration variable.

Let the coordinate s measure distance along the test monopole with origin at (XA,YA,ZA). From any point T on monopole t, construct a line to the test monopole such that the line is perpendicular to the test monopole. SZ denotes the s coordinate of the intersection of this line with the test monopole. The length of the line is the radial coordinate ρ, and RS denotes ρ^2. R1 and R2 are the distances from (XA,YA,ZA) and (XB,YB,ZB) to the point T. C1 is the current at T for the mode with terminals at (X1,Y1,Z1), and C2 is the current at T for the other mode with terminals at (X2,Y2,Z2). C denotes the Simpson's-rule weighting coefficient.

Below statement 300, ZGS performs some analytic geometry in preparation for calling ZGMM. The remaining part of this Appendix concerns this last part of subroutine ZGS.

Let $\hat{s}$ denote a unit vector from (XA,YA,ZA) toward (XB,YB,ZB), and let $\hat{t}$ denote a unit vector from (X1,Y1,Z1) toward (X2,Y2,Z2). Then $\hat{s} \cdot \hat{t} = \cos \psi$ = CC where ψ is the angle formed by the axes of the two monopoles. Let monopole s lie in one plane P_s and monopole t in another parallel plane P_t. CAD, CBD and CGD are the direction cosines of the unit vector $\hat{d} = \hat{t} \times \hat{s} / \sin \psi$ which is perpendicular to both planes. To obtain the distance DK between the planes, we construct a vector $\underline{R}_{11}$ from (XA,YA,ZA) to (X1,Y1,Z1) and take DK = $\underline{R}_{11} \cdot \hat{d}$.

Construct a line from (X1,Y1,Z1) to the test monopole, such that the line is perpendicular to the test monopole. SZ denotes the s coordinate of the intersection of this line with the test monopole, and the cartesian coordinates of this intersection are XZ, YZ and ZZ. The direction cosines of $\hat{s} \times \hat{d}$ are CAP, CBP and CGP.

From the point (X1,Y1,Z1) in plane P_t, construct a perpendicular line to the point (XP1,YP1,ZP1) in the plane P_s. This line is parallel with $\hat{d}$ and has length DK. Let $\underline{R}$ represent a vector from (XZ,YZ,ZZ) to (XP1,YP1,ZP1). P1 denotes $\underline{R} \cdot (\hat{s} \times \hat{d})$. S1 and T1 are defined in Appendix 9.

Subroutine ZGS is essentially the same as GGS except the medium is specialized to free space in ZGS.

APPENDIX 11. Subroutine ZSURF

Subroutine ZSURF, listed in Fig. 15, calculates the surface impedance of a solid circular-cylindrical wire with exterior excitation. ZS denotes the surface impedance in ohms, and the input data are defined as follows:

AK	ka, where a is the wire radius
CMM	conductivity of the wire in megamhos/m
FMC	frequency, MHz

The surface impedance is defined by $Z_S = E_Z/H_\phi$ where the fields E and H are evaluated at the surface of the wire. This subroutine calculates the impedance for the lowest order cylindrical mode with fields E_Z and H_ϕ independent of ϕ. The wire is considered to be a good conductor in the sense that the displacement current is negligible in comparison with the conduction current. In the present application, we require the surface impedance appropriate for the current distribution $I(z) = \sin kz$. For a highly conducting wire, however, this impedance is considered to be the same as that for a uniformly distributed current. BER, BEI, BERP and BEIP denote the Kelvin functions ber, bei, ber' and bei' with argument x. When x is less than 8, BES and BES1 denote the Bessel functions J_0 and J_1 with argument

$$z = x\, e^{-j\pi/4} .$$

When x is greater than 8,

$$J_0/BES = J_1/BES1 = \frac{e^{.707x}}{\sqrt{2\pi x}} .$$

```
      SUBROUTINE ANTI(IA,IB,I1,I2,I3,IWRCJ,IWRITE,I12,ICC,INS,JA,JB,
     2JPP,MD,N,NCM,NG,NLD,NPGP,NRS,NS,C,CDK,SDK,CJ,CMM,D,EFF,G,VG,VJ,
     3Y11,Z11,ZH,ZLD)
      COMPLEX C(ICC,ICC),CJ(1),VG(1),VJ(1),ZLD(1)
      COMPLEX Y11,Z11,ZH,CJI,VJI,DYY
      DIMENSION IA(1),IB(1),MD(1),CDK(1),SDK(1),D(1),I1(1),I2(1),I3(1)
      DIMENSION MD(INS,4),JA(1),JB(1),IGEN(1),JGEN(1)
 1    FORMAT(8X,'J=',I5,5X,'VG(J)=',2F10.2)
 2    FORMAT(8X,'J=',I5,5X,'ZLD(J)=',2F10.2)
 3    FORMAT(10X,'I',3X,'MAGNITUDE',3X,'PHASE',9X,'REAL',8X,'IMAGINARY')
 4    FORMAT(1X,1I10,1F10.3,1F10.1,2F15.7)
 5    FORMAT(1H0)
      IF(I12.GT.0)I12=1
      IF(IWRITE.LE.0)GO TO 14
      DO 10 J=1,NRS
      AVG=CABS(VG(J))
 10   IF(AVG.GT.0.01)WRITE(6,1)J,VG(J)
      WRITE(6,5)
      IF(NLD.LE.0)GO TO 14
      ZMAX=.0
      DO 12 J=1,NRS
      AZL=CABS(ZLD(J))
      IF(AZL.LT.0.1)GO TO 12
      ZMAX=1.
      WRITE(6,2)J,ZLD(J)
 12   CONTINUE
      IF(ZMAX.GT.0.5)WRITE(6,5)
 14   DO 30 I=1,NCM
      CJ(I)=(.0,.0)
      L=2
      FAC=1.
      K=JA(I)
      IF(I.GT.NPGP)GO TO 15
      L=1
      FAC=2.
      IF(JB(I).LT.K)K=JB(I)
 15   DO 25 KK=1,L
      KA=IA(K)
      KB=IB(K)
      JJ=K
      FI=FAC
      IF(KB.EQ.I2(I))GO TO 22
      IF(KB.EQ.I1(I))FI=-FAC
      CJ(I)=CJ(I)+FI*VG(JJ)
      GO TO 25
 22   IF(KA.EQ.I3(I))FI=-FAC
      JJ=K+NRS
      CJ(I)=CJ(I)+FI*VG(JJ)
 25   K=JB(I)
      VJ(I)=CJ(I)
      K=I+JPP
 30   IF(I.GT.NPGP)VJ(K)=-VJ(I)
      ISYM=1
      IF(N.EQ.NPGP)ISYM=0
      IWR=0
      CALL CROUT(C,CJ,ICC,ISYM,IWR,I12,NCM)
      I12=12
      CMAX=.0
      DO 80 I=1,NCM
      CA=CABS(CJ(I))
      K=I+JPP
      IF(I.GT.NPGP)CJ(K)=-CJ(I)
```

Fig. 5. Subroutine ANTI.

```
80   IF(CA.GT.CMAX)CMAX=CA
     IF(IWRCJ.GE.1)WRITE(6,3)
     Y11=(.0,.0)
     G=1.
     EFF=.0
     Z11=(.0,.0)
     IF(CMAX.LE.0.)GO TO 500
     G=.0
     DO 90 I=1,N
     CJI=CJ(I)
     VJI=VJ(I)
     DYY=CJI*CONJG(VJI)
     IF(I.LE.NCM)Y11=Y11+DYY
     G=G+REAL(DYY)
     IF(IWRCJ.LE.0)GO TO 90
     IF(I.GT.NCM)GO TO 90
     CA=CABS(CJI)/CMAX
     PH=.0
     IF(CA.GT.1.E-30)PH=57.29578*ATAN2(AIMAG(CJI),REAL(CJI))
     WRITE(6,4)I,CA,PH,CJI
90   CONTINUE
     IF(IWRCJ.GE.1)WRITE(6,5)
     G=G/2.
     Z11=1./Y11
     EFF=100.
     IF(CMM.LE.0. .AND. NLD.LE.0)GO TO 500
     CALL      PDISS(IA,IB,INS,I1,I2,I3,MD,ND,NLD,NRS,CJ,CMM,D,CDK,
    2SDK,DISS,ZH,ZLD)
     EFF=100.*(G-DISS)/G
500. RETURN
     END
```

Fig. 5. Subroutine ANTI - continued

```fortran
      SUBROUTINE CROUT(C,S,ICC,ISYM,IWR,I12,N)
      COMPLEX C(ICC,ICC),S(1)
      COMPLEX F,P,SS,T
    2 FORMAT(1X,I15,1F10.3,1F15.7,1F10.0)
    5 FORMAT(1H0)
      IF(I12.NE.1)GO TO 22
      IF(N.EQ.1)S(1)=S(1)/C(1,1)
      IF(N.EQ.1)GO TO 100
      IF(ISYM.NE.0)GO TO 8
      DO 6 I=1,N
      DO 6 J=I,N
    6 C(J,I)=C(I,J)
    8 F=C(1,1)
      DO 10 L=2,N
   10 C(1,L)=C(1,L)/F
      DO 20 L=2,N
      LLL=L-1
      DO 20 I=L,N
      F=C(I,L)
      DO 11 K=1,LLL
   11 F=F-C(I,K)*C(K,L)
      C(I,L)=F
      IF(L.EQ.I)GO TO 20
      P=C(L,L)
      IF(ISYM.EQ.0)GO TO 15
      F=C(L,I)
      DO 12 K=1,LLL
   12 F=F-C(L,K)*C(K,I)
      C(L,I)=F/P
      GO TO 20
   15 F=C(I,L)
      C(L,I)=F/P
   20 CONTINUE
   22 DO 30 L=1,N
      P=C(L,L)
      T=S(L)
      IF(L.EQ.1)GO TO 30
      LLL=L-1
      DO 25 K=1,LLL
   25 T=T-C(L,K)*S(K)
   30 S(L)=T/P
      DO 38 L=2,N
      I=N-L+1
      II=I+1
      T=S(I)
      DO 35 K=II,N
   35 T=T-C(I,K)*S(K)
   38 S(I)=T
      IF(IWR.LE.0) GO TO 100
      WRITE(6,5)
      CNOR=.0
      DO 40 I=1,N
      SA=CABS(S(I))
   40 IF(SA.GT.CNOR)CNOR=SA
      IF(CNOR.LE.0.)CNOR=1.
      DO 44 I=1,N
      SS=S(I)
      SA=CABS(SS)
      SNOR=SA/CNOR
      PH=.0
      IF(SA.GT.0.)PH=57.29578*ATAN2(AIMAG(SS),REAL(SS))
   44 WRITE(6,2)I,SNOR,SA,PH
      WRITE(6,5)
  100 RETURN
      END
```

Fig. 6. Subroutine CROUT.

```
      SUBROUTINE EXPJ(V1,V2,W12)
      COMPLEX EC,E15,S,T,UC,VC,V1,V2,W12,Z
      DIMENSION V(21),W(21),D(16),E(16)
      DATA V/   0.22284667E 00,
     20.11889321E 01,0.29927363E 01.0.57751436E 01.0.98374674E 01,
     20.15982874E 02,0.93307812E-01,0.49269174E 00,0.12155954E 01,
     20.22699495E 01,0.36676227E 01,0.54253366E 01,0.75659162E 01,
     20.10120228E 02,0.13130282E 02,0.16654408E 02,0.20776479E 02,
     20.25623894E 02,0.31407519E 02,0.38530683E 02,0.48026086E 02/
      DATA W/   0.45896460E 00,
     20.41700083E 00,0.11337338E 00,0.10399197E-01,0.26101720E-03,
     20.89854791E-06,0.21823487E 00,0.34221017E 00,0.26302758E 00,
     20.12642582E 00,0.40206865E-01,0.85638778E-02,0.12124361E-02,
     20.11167440E-03,0.64599267E-05,0.22263169E-06,0.42274304E-08,
     20.39218973E-10,0.14565152E-12,0.14830270E-15,0.16005949E-19/
      DATA D/   0.22495842E 02,
     2 0.74411568E 02,-0.41431576E 03,-0.78754339E 02, 0.11254744E 02,
     2 0.16021761E 03,-0.23862195E 03,-0.50094687E 03,-0.68487854E 02,
     2 0.12254778E 02,-0.10161976E 02,-0.47219591E 01. 0.79729681E 01,
     2-0.21069574E 02, 0.22046490E 01, 0.89728244E 01/
      DATA E/   0.21103107E 02,
     2-0.37959787E 03,-0.97489220E 02, 0.12900672E 03, 0.17949226E 02,
     2-0.12910931E 03,-0.55705574E 03,  .13524801E 02, 0.14696721E 03,
     2 0.17949528E 02,-0.32981014E 00, 0.31028836E 02, 0.81657657E 01,
     2 0.22236961E 02, 0.39124892E 02, 0.81636799E 01/
      Z=V1
      DO 100 JIM=1,2
      X=REAL(Z)
      Y=AIMAG(Z)
      E15=(.0,.0)
      AB=CABS(Z)
      IF(AB.EQ.0.)GO TO 90
      IF(X.GE.0. .AND. AB.GT.10.)GO TO 80
      YA=ABS(Y)
      IF(X.LE.0. .AND. YA.GT.10.)GO TO 80
      IF(YA-X.GE.17.5.OR.YA.GE.6.5.OR.X+YA.GE.5.5.OR.X.GE.3.)GO TO 20
      IF(X.LE.-9.)GO TO 40
      IF(YA-X.GE.2.5)GO TO 50
      IF(X+YA.GE.1.5)GO TO 30
   10 N=6.+3.*AB
      E15=1./(N-1.)-Z/N**2
   15 N=N-1
      E15=1./(N-1.)-Z*E15/N
      IF(N.GE.3)GO TO 15
      E15=Z*E15-CMPLX(.577216+ALOG(AB),ATAN2(Y,X))
      GO TO 90
   20 J1=1
      J2=6
      GO TO 31
   30 J1=7
      J2=21
   31 S=(.0,.0)
      YS=Y*Y
      DO 32 I=J1,J2
      XI=V(I)+X
      CF=W(I)/(XI*XI+YS)
   32 S=S+CMPLX(XI*CF,-YA*CF)
      GO TO 54
   40 T3=X*X-Y*Y
      T4=2.*X*YA
      T5=X*T3-YA*T4
      T6=X*T4+YA*T3
```

Fig. 7. Subroutine EXPJ.

```
      UC=CMPLX(D(11)+D(12)*X+D(13)*T3+T5-E(12)*YA-E(13)*T4,
     2         E(11)+E(12)*X+E(13)*T3+T6+D(12)*YA+D(13)*T4)
      VC=CMPLX(D(14)+D(15)*X+D(16)*T3+T5-E(15)*YA-E(16)*T4,
     2         E(14)+E(15)*X+E(16)*T3+T6+D(15)*YA+D(16)*T4)
      GO TO 52
50    T3=X*X-Y*Y
      T4=2.*X*YA
      T5=X*T3-YA*T4
      T6=X*T4+YA*T3
      T7=X*T5-YA*T6
      T8=X*T6+YA*T5
      T9=X*T7-YA*T8
      T10=X*T8+YA*T7
      UC=CMPLX(D(1)+D(2)*X+D(3)*T3+D(4)*T5+D(5)*T7+T9-(E(2)*YA+E(3)*T4
     2+E(4)*T6+E(5)*T8),E(1)+E(2)*X+E(3)*T3+E(4)*T5+E(5)*T7+T10+
     3(D(2)*YA+D(3)*T4+D(4)*T6+D(5)*T8))
      VC=CMPLX(D(6)+D(7)*X+D(8)*T3+D(9)*T5+D(10)*T7+T9-(E(7)*YA+E(8)*T4
     2+E(9)*T6+E(10)*T8),E(6)+E(7)*X+E(8)*T3+E(9)*T5+E(10)*T7+T10+
     3(D(7)*YA+D(8)*T4+D(9)*T6+D(10)*T8))
52    EC=UC/VC
      S=EC/CMPLX(X,YA)
54    EX=EXP(-X)
      T=EX*CMPLX(COS(YA),-SIN(YA))
      E15=S*T
56    IF(Y.LT.0.)E15=CONJG(E15)
      GO TO 90
80    E15=.409319/(Z+.193044)+.421831/(Z+1.02666)+.147126/(Z+2.56788)+
     2.206335E-1/(Z+4.90035)+.107401E-2/(Z+8.18215)+.158654E-4/(Z+
     312.7342)+.317031E-7/(Z+19.3957)
      E15=E15*CEXP(-Z)
90    IF(JIM.EQ.1)W12-E15
100   Z=V2
      Z=V2/V1
      TH=ATAN2(AIMAG(Z),REAL(Z))-ATAN2(AIMAG(V2),REAL(V2))
     2+ATAN2(AIMAG(V1),REAL(V1))
      AB=ABS(TH)
      IF(AB.LT.1.)TH=.0
      IF(TH.GT.1.)TH=6.2831853
      IF(TH.LT.-1.)TH=-6.2831853
      W12=W12-E15+CMPLX(.0,TH)
      RETURN
      END
```

Fig. 7. Subroutine EXPJ - continued

```
      SUBROUTINE IDANT(IA,IB,ICC,INS,INT,I1,I2,I3,JA,JB,JPP,MD,N,NCM,
     2ND,NLD,NP,NPGP,NRS,NS,AK,C,CMM,D,FMC,CDK,SDK,X,Y,Z,ZH,ZLD)
      COMPLEX ZS,ZH,P(2,2),Q(2,2),CIJ,ZSAM,ZOPP
      COMPLEX C(ICC,ICC),ZLD(1)
      DIMENSION X(1),Y(1),Z(1),IA(1),IB(1),ND(1),CDK(1),SDK(1),D(1)
      DIMENSION I1(1),I2(1),I3(1),JA(1),JB(1),MD(INS,4)
      DATA TP/6.28318/
2     FORMAT(8X,'AK=',F8.6,5X,'DMAX=',F8.4,5X,'DMIN=',F8.4)
      DO 10 I=1,NCM
      DO 10 J=1,NCM
10    C(I,J)=(.0,.0)
      DMAX=.0
      DMIN=100.
      DO 20 J=1,NRS
      DJ=D(J)
      IF(DJ.GT.DMAX)DMAX=DJ
      IF(DJ.LT.DMIN)DMIN=DJ
      CDK(J)=COS(DJ)
      SDK(J)=SIN(DJ)
      K=J+NRS
      CDK(K)=CDK(J)
20    SDK(K)=SDK(J)
      IF(DMIN.LT.AK)GO TO 21
      IF(DMAX.GT.3.)GO TO 21
      IF(AK.GT.0.1)GO TO 21
      GO TO 22
21    WRITE(6,2)AK,DMAX,DMIN
      N=0
      RETURN
22    DO 200 K=1,NS
      NDK=ND(K)
      KA=IA(K)
      KB=IB(K)
      DK=D(K)
      DO 200 L=1,NS
      NDL=ND(L)
      LA=IA(L)
      LB=IB(L)
      DL=D(L)
      NIL=0
      DO 200 II=1,NDK
      I=MD(K,II)
      IF(I.GT.NCM)GO TO 200
      FI=1.
      IF(KB.EQ.I2(I))GO TO 36
      IF(KB.EQ.I1(I))FI=-1.
      IS=1
      GO TO 40
36    IF(KA.EQ.I3(I))FI=-1.
      IS=2
40    DO 200 JJ=1,NDL
      J=MD(L,JJ)
      IF(I.GT.J)GO TO 200
      FJ=1.
      IF(LB.EQ.I2(J))GO TO 46
      IF(LB.EQ.I1(J))FJ=-1.
      JS=1
      GO TO 50
46    IF(LA.EQ.I3(J))FJ=-1.
      JS=2
50    IF(NIL.NE.0)GO TO 168
      NIL=1
```

Fig. 8. Subroutine IDANT.

```
      IF(K.EQ.L)GO TO 120
      IND=(LA-KA)*(LB-KA)*(LA-KB)*(LB-KB)
      IF(IND.EQ.0)GO TO 80
C     SEGMENTS K AND L SHARE NO POINTS
      CALL ZGS(X(KA),Y(KA),Z(KA),X(KB),Y(KB),Z(KB),X(LA),Y(LA),Z(LA),
     2X(LB),Y(LB),Z(LB),AK,DK,CDK(K),SDK(K),DL,SDK(L),INT,
     3P(1,1),P(1,2),P(2,1),P(2,2))
      GO TO 168
C     SEGMENTS K AND L SHARE ONE POINT (THEY INTERSECT)
   80 KG=0
      JM=KB
      JC=KA
      KF=1
      IND=(KB-LA)*(KB-LB)
      IF(IND.NE.0)GO TO 82
      JC=KB
      KF=-1
      JM=KA
      KG=3
   82 LG=3
      JP=LA
      LF=-1
      IF(LB.EQ.JC)GO TO 83
      JP=LB
      LF=1
      LG=0
   83 SGN=KF*LF
      CPSI=((X(JP)-X(JC))*(X(JM)-X(JC))+(Y(JP)-Y(JC))*(Y(JM)-Y(JC))
     2+(Z(JP)-Z(JC))*(Z(JM)-Z(JC)))/(DK*DL)
      CALL ZGMM(.0,DK,.0,DL,AK,CDK(K),SDK(K),SDK(L),CPSI
     2,Q(1,1),Q(1,2),Q(2,1),Q(2,2))
      DO 98 KK=1,2
      KP=IABS(KK-KG)
      DO 98 LL=1,2
      LP=IABS(LL-LG)
      P(KP,LP)=SGN*Q(KK,LL)
   98 CONTINUE
      GO TO 168
C     K=L  (SELF REACTION OF SEGMENT K)
  120 S=.5
      IF(KA.NE.LA)S=-.5
      CALL ZGMM(.0,DK,DK*(.5-S),DK*(.5+S),AK,CDK(K),SDK(K),SDK(K),1.
     2,P(1,1),P(1,2),P(2,1),P(2,2))
  168 CIJ=FI*FJ*P(IS,JS)
      IF(J.GT.NCM)GO TO 190
      C(I,J)=C(I,J)+CIJ
      IF(I.NE.J)C(J,I)=C(J,I)+CIJ
      GO TO 200
  190 JG=J-JPP
      C(I,JG)=C(I,JG)-CIJ
  200 CONTINUE
      ZH=(.0,.0)
      IF(CMM.LE.0.)GO TO 262
      CALL ZSURF(AK,CMM,FMC,ZS)
      ZH=ZS/(4.*TP*AK)
      DO 260 K=1,NS
      NDK=ND(K)
      ZSAM=2.*ZH*(D(K)-SDK(K)*CDK(K))/SDK(K)**2
      DO 210 II=1,NDK
      I=MD(K,II)
  210 IF(I.LE.NCM)C(I,I)=C(I,I)+ZSAM
      IF(NDK.EQ.1)GO TO 260
```

Fig. 8. Subroutine IDANT - continued

```
      ZOPP=2.*ZH*(SDK(K)-D(K)*CDK(K))/SDK(K)**2
      KA=IA(K)
      KB=IB(K)
      DO 260 II=1,NDK
      I=MD(K,II)
      IF(I.GT.NCM)GO TO 260
      FI=1.
      IF(KB.EQ.I2(I))GO TO 236
      IF(KB.EQ.I1(I))FI=-1.
      IS=1
      GO TO 240
  236 IF(KA.EQ.I3(I))FI=-1.
      IS=2
  240 DO 260 JJ=1,NDK
      J=MD(K,JJ)
      IF(I.GE.J)GO TO 260
      FJ=1.
      IF(KB.EQ.I2(J))GO TO 246
      IF(KB.EQ.I1(J))FJ=-1.
      JS=1
      GO TO 250
  246 IF(KA.EQ.I3(J))FJ=-1.
      JS=2
  250 IF(IS.EQ.JS)CIJ=FI*FJ*ZSAM
      IF(IS.NE.JS)CIJ=FI*FJ*ZOPP
      IF(J.GT.NCM)GO TO 259
      C(I,J)=C(I,J)+CIJ
      C(J,I)=C(J,I)+CIJ
      GO TO 260
  259 JG=J-JPP
      C(I,JG)=C(I,JG)-CIJ
  260 CONTINUE
  262 IF(NLD.LE.0)GO TO 300
      DO 280 I=1,NCM
      JJA=JA(I)
      J1=JJA
      II2=I2(I)
      II1=I1(I)
      IF(II2.EQ.IB(J1))J1=J1+NRS
      IF(I.LE.NPGP)GO TO 270
      JJB=JB(I)
      J2=JJB
      IF(II2.EQ.IB(J2))J2=J2+NRS
      C(I,I)=C(I,I)+ZLD(J1)+ZLD(J2)
      JJJ=JJA
      DO 268 K=1,2
      NDJ=ND(JJJ)
      DO 266 JJ=1,NDJ
      J=MD(JJJ,JJ)
      IF(J.EQ.I)GO TO 266
      IF(I2(J).NE.II2)GO TO 266
      FI=1.
      IF(K.EQ.2)GO TO 264
      IF(I1(J).NE.II1)FI=-1.
      C(I,J)=C(I,J)+FI*ZLD(J1)
      GO TO 266
  264 IF(I3(J).NE.I3(I))FI=-1.
      C(I,J)=C(I,J)+FI*ZLD(J2)
  266 CONTINUE
  268 JJJ=JJB
      GO TO 280
  270 IF(IB(J1).LE.NPGP)J1=J1+NRS
```

Fig. 8. Subroutine IDANT - continued

```
      C(I,I)=C(I,I)+2.*ZLD(J1)
      NDJ=ND(JJA)
      DO 278 JJ=1,NDJ
      J=MD(JJA,JJ)
      IF(J.EQ.I)GO TO 278
      IF(I2(J).NE.I12)GO TO 278
      FI=1.
      IF(I1(J).NE.I11)FI=-1.
      C(I,J)=C(I,J)+2.*FI*ZLD(J1)
  278 CONTINUE
  280 CONTINUE
  300 RETURN
      END
```

Fig. 8. Subroutine IDANT - continued

```
      SUBROUTINE IFFLD(IA,IB,INS,I1,I2,I3,MD,N,ND,NS,CDK,CJ,D,
     2EPP,ETT,EPH,ETH,G,GPP,GTT,PH,SDK,TH,X,Y,Z)
      COMPLEX EPH,ETH,CJI,ET1,ET2,EP1,EP2
      COMPLEX CJ(1),EPP(1),ETT(1)
      DIMENSION IA(1),IB(1),ND(1),CDK(1),SDK(1),D(1),X(1),Y(1),Z(1)
      DIMENSION I1(1),I2(1),I3(1),MD(INS,4)
      DATA CJI/(.0,-.530888E-2)/
      THR=.0174533*TH
      CTH=COS(THR)
      STH=SIN(THR)
      PHR=.0174533*PH
      CPH=COS(PHR)
      SPH=SIN(PHR)
      DO 130 I=1,N
      ETT(I)=(.0,.0)
  130 EPP(I)=(.0,.0)
      DO 140 K=1,NS
      KA=IA(K)
      KB=IB(K)
      CALL ZFF(X(KA),Y(KA),Z(KA),X(KB),Y(KB),Z(KB),D(K)
     2,CDK(K),SDK(K),CTH,STH,CPH,SPH,ET1,ET2,EP1,EP2)
      NDK=ND(K)
      DO 140 II=1,NDK
      I=MD(K,II)
      FI=1.
      IF(KB.EQ.I2(I))GO TO 136
      IF(KB.EQ.I1(I))FI=-1.
      EPP(I)=EPP(I)+FI*EP1
      ETT(I)=ETT(I)+FI*ET1
      GO TO 140
  136 IF(KA.EQ.I3(I))FI=-1.
      EPP(I)=EPP(I)+FI*EP2
      ETT(I)=ETT(I)+FI*ET2
  140 CONTINUE
      EPH=(.0,.0)
      ETH=(.0,.0)
  200 DO 260 I=1,N
      ETH=ETH+CJ(I)*ETT(I)
  260 EPH=EPH+CJ(I)*EPP(I)
      APP=CABS(EPH)
      ATT=CABS(ETH)
      GPP=APP*APP/(30.*G)
      GTT=ATT*ATT/(30.*G)
      RETURN
      END
```

Fig. 9. Subroutine IFFLD.

```
      SUBROUTINE ISORT(IA,IB,ICC,ICJ,INS,IWRITE,I1,I2,I3,JA,JB,
     2MAX,MIN,MD,N,NCM,ND,NP,NS,NRP,NRS,NPGP,NSGP,DC,XC,YC,ZC)
      DIMENSION JSP(20),DC(1),XC(1),YC(1),ZC(1)
      DIMENSION I1(1),I2(1),I3(1),JA(1),JB(1)
      DIMENSION IA(1),IB(1),ND(1),MD(INS,4)
    1 FORMAT(5X,'J',1X,'IA(J)',1X,'IB(J)',12X,'K',
     21X,'IA(K)',1X,'IB(K)',7X,'DC(J)')
    2 FORMAT(1X,3I5,1I15,2I5,2F15.5)
    3 FORMAT(1X,1I5,3F10.5,1I5,3F10.5)
    4 FORMAT(1X,6I5,1I15,6I5)
    5 FORMAT(1H0)
    6 FORMAT(5X,'I',4X,'XC(I)',5X,'YC(I)',5X,'ZC(I)',
     2 5X,'J',4X,'XC(J)',5X,'YC(J)',5X,'ZC(J)')
    7 FORMAT(5X,'I',3X,'JA',3X,'JB',3X,'I1',3X,'I2',3X,'I3',
     214X,'K',3X,'JA',3X,'JB',3X,'I1',3X,'I2',3X,'I3')
      IGPP=NPGP+1
      NPI=NRP-NPGP
C  NEXT SET UP THE IMAGE SEGMENTS
      DO 18 J=1,NRS
      K=J+NRS
      IA(K)=IA(J)
      IF(IA(J).GT.NPGP)IA(K)=IA(J)+NPI
      IB(K)=IB(J)
   18 IF(IB(J).GT.NPGP)IB(K)=IB(J)+NPI
C  NEXT SET UP THE IMAGE POINTS
      DO 20 I=IGPP,NRP
      J=I+NPI
      XC(J)=XC(I)
      YC(J)=YC(I)
   20 ZC(J)=-ZC(I)
C  NEXT CALCULATE THE SEGMENT LENGTHS DC(J)
      IF(IWRITE.LE.0)GO TO 22
      WRITE(6,5)
      WRITE(6,1)
   22 DO 25 J=1,NRS
      K=IA(J)
      L=IB(J)
      DX=XC(K)-XC(L)
      DY=YC(K)-YC(L)
      DZ=ZC(K)-ZC(L)
      DC(J)=SQRT(DX*DX+DY*DY+DZ*DZ)
      K=J+NRS
      DC(K)=DC(J)
   25 IF(IWRITE.GE.1)WRITE(6,2)J,IA(J),IB(J),K,IA(K),IB(K),DC(J)
      IF(IWRITE.LE.0)GO TO 32
      WRITE(6,5)
      WRITE(6,6)
      DO 30 I=1,NRP
      IF(I.GT.NPGP)GO TO 28
      WRITE(6,3)I,XC(I),YC(I),ZC(I)
      GO TO 30
   28 J=I+NPI
      WRITE(6,3)I,XC(I),YC(I),ZC(I),J,XC(J),YC(J),ZC(J)
   30 CONTINUE
      WRITE(6,5)
C  CHECK INPUT DATA FOR CONSISTENCE
   32 N=0
      MIN=100
      MAX=100
      IF(NPGP.LE.0)GO TO 40
      DO 38 I=1,NPGP
      L=0
```

Fig. 10. Subroutine ISORT.

```
      DO 35 J=1,NSGP
      K=(IA(J)-I)*(IB(J)-I)
   35 IF(K.EQ.0)L=L+1
      IF(L.LT.MAX)MAX=L
   38 N=N+2*L-1
   40 IF(NRP.LE.NPGP)GO TO 50
      DO 46 I=IGPP,NRP
      L=0
      DO 44 J=1,NRS
      K=(IA(J)-I)*(IB(J)-I)
   44 IF(K.EQ.0)L=L+1
      IF(L.LT.MIN)MIN=L
   46 N=N+2*(L-1)
   50 IF(N.LE.0 .OR. N.GT.ICJ)GO TO 500
      IF(MAX.LE.0 .OR. MIN.LE.0)GO TO 500
      IF(NPGP.LE.0)GO TO 58
C  SET UP THE MODES AT THE GROUND PLANE THAT WILL NOT HAVE IMAGES
      DO 56 I=1,NPGP
      J=0
   52 J=J+1
      IAJ=IA(J)
      IBJ=IB(J)
      KK=(IAJ-I)*(IBJ-I)
      IF(J.EQ.NSGP)GO TO 54
      IF(KK.NE.0)GO TO 52
   54 JA(I)=J
      JB(I)=J+NRS
      I2(I)=I
      I1(I)=IBJ
      IF(IBJ.EQ.I)I1(I)=IAJ
   56 I3(I)=I1(I)+NPI
   58 I=NPGP
      N=NPGP
      NCM=NPGP
      JPP=0
      IF(NRS.EQ.NPGP)GO TO 75
C  SET UP THE REST OF THE REAL MODES
      DO 65 K=1,NRP
      NJK=0
      DO 60 J=1,NRS
      IND=(IA(J)-K)*(IB(J)-K)
      IF(IND.NE.0)GO TO 60
      NJK=NJK+1
      JSP(NJK)=J
   60 CONTINUE
      MOD=NJK-1
      IF(MOD.LE.0)GO TO 65
      DO 62 IMD=1,MOD
      I=I+1
      IPD=IMD+1
      JAI=JSP(IMD)
      JA(I)=JAI
      JBI=JSP(IPD)
      JB(I)=JBI
      I1(I)=IA(JAI)
      IF(IA(JAI).EQ.K)I1(I)=IB(JAI)
      I2(I)=K
      I3(I)=IA(JBI)
   62 IF(IA(JBI).EQ.K)I3(I)=IB(JBI)
   65 CONTINUE
      NCM=I
      JPP=NCM-NPGP
```

Fig. 10. Subroutine ISORT - continued

```
C   SET UP THE IMAGE MODES
      DO 70 I=IGPP,NCM
      K=I+JPP
      JA(K)=JA(I)+NRS
      JB(K)=JB(I)+NRS
      IIA=I1(I)
      IIB=I2(I)
      IIC=I3(I)
      I1(K)=IIA
      IF(IIA.GT.NPGP)I1(K)=IIA+NPI
      I2(K)=IIB
      IF(IIB.GT.NPGP)I2(K)=IIB+NPI
      I3(K)=IIC
   70 IF(IIC.GT.NPGP)I3(K)=IIC+NPI
      N=2*NCM-NPGP
   75 MAX=0
      MIN=100
C   ND(J) = NUMBER OF DIPOLE MODES SHARING SEGMENT J
C   MD(J,K) = LIST OF DIPOLES SHARING SEGMENT J
      DO 100 J=1,NS
      DO 80 K=1,4
   80 MD(J,K)=0
      K=0
      DO 90 I=1,N
      JAI=JA(I)
      JBI=JB(I)
      L=(JAI-J)*(JBI-J)
      IF(L.NE.0)GO TO 90
      K=K+1
      MD(J,K)=I
   90 CONTINUE
      ND(J)=K
      IF(K.GT.MAX)MAX=K
  100 IF(K.LT.MIN)MIN=K
      IF(IWRITE.LE.0)GO TO 500
      WRITE(6,7)
      DO 110 I=1,NCM
      IF(I.GT.NPGP)GO TO 108
      WRITE(6,4)I,JA(I),JB(I),I1(I),I2(I),I3(I)
      GO TO 110
  108 K=I+JPP
      WRITE(6,4)I,JA(I),JB(I),I1(I),I2(I),I3(I),K,JA(K),JB(K),
     2I1(K),I2(K),I3(K)
  110 CONTINUE
      WRITE(6,5)
  500 RETURN
      END
```

Fig. 10. Subroutine ISORT - continued

```
      SUBROUTINE PDISS(IA,IB,INS,I1,I2,I3,MD,ND,NLD,NS ,CJ,CMM,D,CDK,
     2SDK,DISS,ZH,ZLD)
      COMPLEX CJ(1),ZH,CJA,CJB,ZLA,ZLB,ZLD(1)
      DIMENSION CDK(1),SDK(1),D(1),I1(1),I2(1),I3(1),IA(1),IB(1),ND(1)
      DIMENSION MD(INS,4)
      RH=REAL(ZH)
      DISS=.0
      DO 100 K=1,NS
      IF(CMM.LE.0.)GO TO 10
      FA=2.*RH*(D(K)-SDK(K)*CDK(K))/SDK(K)**2
      FB=4.*RH*(SDK(K)-D(K)*CDK(K))/SDK(K)**2
   10 KA=IA(K)
      KB=IB(K)
      CJA=(.0,.0)
      CJB=(.0,.0)
      NDK=ND(K)
      DO 40 II=1,NDK
      I=MD(K,II)
      FI=1.
      IF(KB.EQ.I2(I))GO TO 36
      IF(KB.EQ.I1(I))FI=-1.
      CJA=CJA+FI*CJ(I)
      GO TO 40
   36 IF(KA.EQ.I3(I))FI=-1.
      CJB=CJB+FI*CJ(I)
   40 CONTINUE
      IF(NLD.LE.0)GO TO 50
      AJA=CABS(CJA)**2
      BJB=CABS(CJB)**2
      KK=K+NS
      RLA=REAL(ZLD(K))
      RLB=REAL(ZLD(KK))
      DISS=DISS+AJA*RLA+BJB*RLB
   50 IF(CMM.LE.0.)GO TO 100
      DISS=DISS+FA*(CABS(CJA)**2+CABS(CJB)**2)
     2+FB*(REAL(CJA)*REAL(CJB)+AIMAG(CJA)*AIMAG(CJB))
  100 CONTINUE
      RETURN
      END
```

Fig. 11. Subroutine PDISS.

```
      SUBROUTINE ZFF(XA,YA,ZA,XB,YB,ZB,D
     2,CKD,SKD,CTH,STH,CPH,SPH,ET1,ET2,EP1,EP2)
      COMPLEX EJA,EJB,EP1,EP2,ES1,ES2,ET1,ET2
      CA=(XB-XA)/D
      CB=(YB-YA)/D
      CG=(ZB-ZA)/D
      G=(CA*CPH+CB*SPH)*STH+CG*CTH
      GK=1.-G*G
      ET1=(.0,.0)
      ET2=(.0,.0)
      EP1=(.0,.0)
      EP2=(.0,.0)
      IF(GK.LT..001)GO TO 200
      A=XA*STH*CPH+YA*STH*SPH+ZA*CTH
      B=XB*STH*CPH+YB*STH*SPH+ZB*CTH
      EJA=CMPLX(COS(A),SIN(A))
      EJB=CMPLX(COS(B),SIN(B))
      SGD=SIN(G*D)
      CGD=COS(G*D)
      ES1=30.*EJA*CMPLX(SGD-G*SKD,CKD-CGD)/GK/SKD
      ES2=30.*EJB*CMPLX(G*SKD-SGD,CKD-CGD)/GK/SKD
      T=(CA*CPH+CB*SPH)*CTH-CG*STH
      P=-CA*SPH+CB*CPH
      ET1=T*ES1
      ET2=T*ES2
      EP1=P*ES1
      EP2=P*ES2
  200 RETURN
      END
```

Fig. 12. Subroutine ZFF.

```
      SUBROUTINE ZGMM(S1,S2,T1,T2,D,CGDS,SGD1,SGD2,CPSI,P11,P12,P21,P22)
      DOUBLE PRECISION R1,R2,DPO,SIS,TS1,TS2,ST1,ST2,CD,BD,CPSS,SPSI,SK
     2,TL1,TL2,TD1,TD2,SDI,DPSI,DD,ZD
      COMPLEX E(2,2),F(2,2),GAM,P11,P12,P21,P22
      COMPLEX EB,EC,EK,EL,EKL,EGZ1,ES1,ES2,ET1,ET2,EXPA,EXPB
      COMPLEX FGZ(2,2),GM(2),GP(2)
      DATA ETA,GAM,PI/376.727,(.0,1.),3.14159/
      DSQ=D*D
      SGDS=SGD1
      IF(S2.LT.S1)SGDS=-SGD1
      SGDT=SGD2
      IF(T2.LT.T1)SGDT=-SGD2
      IF(ABS(CPSI).GT.0.997)GO TO 110
      ES1=CEXP(GAM*S1)
      ES2=CEXP(GAM*S2)
      ET1=CEXP(GAM*T1)
      ET2=CEXP(GAM*T2)
      DD=D
      DPSI=CPSI
      TD1=T1
      TD2=T2
      CPSS=DPSI*DPSI
      CD=DD/DSQRT(1.D0-CPSS)
      C=CD
      BD=CD*DPSI
      B=BD
      EB=CEXP(GAM*CMPLX(.0,B))
      EC=CEXP(GAM*CMPLX(.0,C))
      DO 10 K=1,2
      DO 10 L=1,2
   10 E(K,L)=(.0,.0)
      TS1=TD1*TD1
      TS2=TD2*TD2
      DPQ=DD*DD
      SI=S1
      DO 100 I=1,2
      FI=(-1)**I
      SDI=SI
      SIS=SDI*SDI
      ST1=2.*SDI*TD1*DPSI
      ST2=2.*SDI*TD2*DPSI
      R1=DSQRT(DPQ+SIS+TS1-ST1)
      R2=DSQRT(DPQ+SIS+TS2-ST2)
      EK=EB
      DO 50 K=1,2
      FK=(-1)**K
      SK=FK*SDI
      EL=EC
      DO 40 L=1,2
      FL=(-1)**L
      EKL=EK*EL
      XX=FK*BD+FL*CD
      TL1=FL*TD1
      TL2=FL*TD2
      RR1=R1+SK+TL1
      RR2=R2+SK+TL2
      CALL EXPJ(GAM*CMPLX(RR1,-XX),GAM*CMPLX(RR2,-XX),EXPA)
      CALL EXPJ(GAM*CMPLX(RR1,XX),GAM*CMPLX(RR2,XX),EXPB)
      E(K,L)=E(K,L)+FI*(EXPA*EKL+EXPB/EKL)
   40 EL=1./EC
   50 EK=1./EB
      ZD=SDI*DPSI
```

Fig. 13. Subroutine ZGMM.

```
      ZC=ZD
      EGZ1=CEXP(GAM*ZC)
      RR1=R1+ZD-TD1
      RR2=R2+ZD-TD2
      CALL EXPJ(GAM*RR1,GAM*RR2,EXPB)
      RR1=R1-ZD+TD1
      RR2=R2-ZD+TD2
      CALL EXPJ(GAM*RR1,GAM*RR2,EXPA)
      F(1,1)=2.*SGDS*(.0,1.)*EXPA/EGZ1
      F(1,2)=2.*SGDS*(.0,1.)*EXPB*EGZ1
  100 SI=S2
      CST=-ETA/(16.*PI*SGDS*SGDT)
      P11=CST*(( F(1,1)+E(2,2)*ES2-E(1,2)/ES2)*ET2
     A        +(-F(1,2)-E(2,1)*ES2+E(1,1)/ES2)/ET2)
      P12=CST*((-F(1,1)-E(2,2)*ES2+E(1,2)/ES2)*ET1
     B        +( F(1,2)+E(2,1)*ES2-E(1,1)/ES2)/ET1)
      P21=CST*((-F(2,1)-E(2,2)*ES1+E(1,2)/ES1)*ET2
     C        +( F(2,2)+E(2,1)*ES1-E(1,1)/ES1)/ET2)
      P22=CST*(( F(2,1)+E(2,2)*ES1-E(1,2)/ES1)*ET1
     D        +(-F(2,2)-E(2,1)*ES1+E(1,1)/ES1)/ET1)
      RETURN
  110 IF(CPSI.LT.0.)GO TO 120
      TA=T1
      TB=T2
      GO TO 130
  120 TA=-T1
      TB=-T2
      SGDT=-SGDT
  130 SI=S1
      DO 150 I=1,2
      TJ=TA
      DO 140 J=1,2
      ZIJ=TJ-SI
      R=SQRT(DSQ+ZIJ*ZIJ)
      W=R+ZIJ
      IF(ZIJ.LT.0.)W=DSQ/(R-ZIJ)
      V=R-ZIJ
      IF(ZIJ.GT.0.)V=DSQ/(R+ZIJ)
      IF(J.EQ.1)V1=V
      IF(J.EQ.1)W1=W
      EGZ(I,J)=CEXP(GAM*ZIJ)
  140 TJ=TB
      CALL EXPJ(GAM*V1,GAM*V,GP(I))
      CALL EXPJ(GAM*W1,GAM*W,GM(I))
  150 SI=S2
      CST=ETA/(8.*PI*SGDS*SGDT)
      P11=CST*(GM(2)*EGZ(2,2)+GP(2)/EGZ(2,2)
     2-CGDS*(GM(1)*EGZ(1,2)+GP(1)/EGZ(1,2)))
      P12=CST*(-GM(2)*EGZ(2,1)-GP(2)/EGZ(2,1)
     2+CGDS*(GM(1)*EGZ(1,1)+GP(1)/EGZ(1,1)))
      P21=CST*(GM(1)*EGZ(1,2)+GP(1)/EGZ(1,2)
     2-CGDS*(GM(2)*EGZ(2,2)+GP(2)/EGZ(2,2)))
      P22=CST*(-GM(1)*EGZ(1,1)-GP(1)/EGZ(1,1)
     2+CGDS*(GM(2)*EGZ(2,1)+GP(2)/EGZ(2,1)))
      RETURN
      END
```

Fig. 13. Subroutine ZGMM - continued

```
      SUBROUTINE ZGS(XA,YA,ZA,XB,YB,ZB,X1,Y1,Z1,X2,Y2,Z2,AK,
     2DS,CDS,SDS,DT,SDT,INT,P11,P12,P21,P22)
      COMPLEX CST,EJ1,EJ2,EJA,EJB,ER1,ER2,ET1,ET2,P11,P12,P21,P22,GAM
      COMPLEX SGDS,SGDT
      DATA ETA,GAM,PI/376.727,(.0,1.),3.14159/
      CA=(X2-X1)/DT
      CB=(Y2-Y1)/DT
      CG=(Z2-Z1)/DT
      CAS=(XB-XA)/DS
      CBS=(YB-YA)/DS
      CGS=(ZB-ZA)/DS
      CC=CA*CAS+CB*CBS+CG*CGS
      IF(ABS(CC).GT.0.997)GO TO 200
   20 SZ=(X1-XA)*CAS+(Y1-YA)*CBS+(Z1-ZA)*CGS
      IF(INT.EQ.0)GO TO 300
      CGDS=CDS
      SGDS=CMPLX(.0,SDS)
      SGDT=CMPLX(.0,SDT)
      INS=2*(INT/2)
      IF(INS.LT.2)INS=2
      IP=INS+1
      DELT=DT/INS
      T=.0
      DSZ=CC*DELT
      P11=(.0,.0)
      P12=(.0,.0)
      P21=(.0,.0)
      P22=(.0,.0)
      AKS=AK*AK
      SGN=-1.
      DO 100 IN=1,IP
      ZZ1=SZ
      ZZ2=SZ-DS
      XXZ=X1+T*CA-XA-SZ*CAS
      YYZ=Y1+T*CB-YA-SZ*CBS
      ZZZ=Z1+T*CG-ZA-SZ*CGS
      RS=XXZ**2+YYZ**2+ZZZ**2
      R1=SQRT(RS+ZZ1**2)
      EJA=CMPLX(COS(R1),-SIN(R1))
      EJ1=EJA/R1
      R2=SQRT(RS+ZZ2**2)
      EJB=CMPLX(COS(R2),-SIN(R2))
      EJ2=EJB/R2
      ER1=EJA*SGDS+ZZ1*EJ1*CGDS-ZZ2*EJ2
      ER2=-EJB*SGDS+ZZ2*EJ2*CGDS-ZZ1*EJ1
      FAC=.0
      IF(RS.GT.AKS)FAC=(CA*XXZ+CB*YYZ+CG*ZZZ)/RS
      ET1=CC*(EJ2-EJ1*CGDS)+FAC*ER1
      ET2=CC*(EJ1-EJ2*CGDS)+FAC*ER2
      C=3.+SGN
      IF(IN.EQ.1 .OR. IN.EQ.IP)C=1.
      C1=C*SIN(DT-T)
      C2=C*SIN(T)
      P11=P11+ET1*C1
      P12=P12+ET1*C2
      P21=P21+ET2*C1
      P22=P22+ET2*C2
      T=T+DELT
      SZ=SZ+DSZ
  100 SGN=-SGN
      CST=-(.0,1.)*ETA*DELT/(12.*PI*SGDS*SGDT)
      P11=CST*P11
```

Fig. 14. Subroutine ZGS.

```
      P12=CST*P12
      P21=CST*P21
      P22=CST*P22
      RETURN
  200 SZ1=(X1-XA)*CAS+(Y1-YA)*CBS+(Z1-ZA)*CGS
      RH1=SQRT((X1-XA-SZ1*CAS)**2+(Y1-YA-SZ1*CBS)**2+(Z1-ZA-SZ1*CGS)**2)
      SZ2=SZ1+DT*CC
      RH2=SQRT((X2-XA-SZ2*CAS)**2+(Y2-YA-SZ2*CBS)**2+(Z2-ZA-SZ2*CGS)**2)
      DDK=(RH1+RH2)/2.
      IF(DDK.GT.20.*AK .AND. INT.GT.0)GO TO 20
      IF(DDK.LT.AK)DDK=AK
      CALL ZGMM(.0,DS,SZ1,SZ2,DDK,CDS,SDS,SDT,1.,P11,P12,P21,P22)
      RETURN
  300 SS=SQRT(1.-CC*CC)
      CAD=(CGS*CB-CBS*CG)/SS
      CBD=(CAS*CG-CGS*CA)/SS
      CGD=(CBS*CA-CAS*CB)/SS
      DK=(X1-XA)*CAD+(Y1-YA)*CBD+(Z1-ZA)*CGD
      DK=ABS(DK)
      IF(DK.LT.AK)DK=AK
      XZ=XA+SZ*CAS
      YZ=YA+SZ*CBS
      ZZ=ZA+SZ*CGS
      XP1=X1-DK*CAD
      YP1=Y1-DK*CBD
      ZP1=Z1-DK*CGD
      CAP=CBS*CGD-CGS*CBD
      CBP=CGS*CAD-CAS*CGD
      CGP=CAS*CBD-CBS*CAD
      P1=CAP*(XP1-XZ)+CBP*(YP1-YZ)+CGP*(ZP1-ZZ)
      T1-P1/SS
      S1=T1*CC-SZ
      CALL ZGMM(S1,S1+DS,T1,T1+DT,DK,CDS,SDS,SDT,CC,P11,P12,P21,P22)
      RETURN
      END
```

Fig. 14. Subroutine ZGS - continued

```
      SUBROUTINE ZSURF(AK,CMM,FMC,ZS)
      COMPLEX BES,BES1,ZS
      DATA ETA,SQT,TP/376.72727,1.41421356,6.2831853/
      SQSWE=1.E6*SQRT(CMM/TP/FMC/8.85433)
      X=AK*SQSWE
      IF(X.GT.8.)GO TO 50
      T=X/8.
      T2=T*T
      T4=T2*T2
      BER=((((((-.901E-5*T4+.122552E-2)*T4-.08349609)*T4+2.641914)*T4
     2-32.363456)*T4+113.77778)*T4-64.)*T4+1.
      BEI=((((((.11346E-3*T4-.01103667)*T4+.52185615)*T4-10.567658)*T4
     2+72.817777)*T4-113.77778)*T4+16.)*T2
      BERP=X*T2*((((((-.394E-5*T4+.45957E-3)*T4-.02609253)*T4+.66047849)
     2*T4-6.0681481)*T4+14.222222)*T4-4.)
      BEIP=X*((((((.4609E-4*T4-.379386E-2)*T4+.14677204)*T4-2.3116751)*
     2T4+11.377778)*T4-10.666667)*T4+.5)
      BES=CMPLX(BER,BEI)
      BES1=.707107*CMPLX(BERP-BEIP,BERP+BEIP)
      GO TO 100
50    XP=.70710681*X
      X1=1./X
      F=((-.0459205*X1+.390625E-2)*X1+.08838835)*X1+1.
      T=((-.04603559*X1-.0625)*X1-.08838835)*X1-.39269907+XP
      BES=F*CMPLX(COS(T),SIN(T))
      F=((.11290231*X1+.03515625)*X1-.26516505)*X1+1.
      T=((.1160097*X1+.1875)*X1+.26516505)*X1+1.1780972+XP
      BES1=F*CMPLX(COS(T),SIN(T))
100   ZS=-CMPLX(1.,-1.)*ETA*BES/BES1/SQT/SQSWE
      RETURN
      END
```

Fig. 15. Subroutine ZSURF.

END

DATE

FILMED

DEC 6 1974